Numerical simulation of destabilizing heterogeneous suspensions at vanishing Reynolds numbers

zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN
(Dr.–Ing.)

der Fakultät für Chemieingenieurwesen und Verfahrenstechnik des
Karlsruher Instituts für Technologie (KIT)

genehmigte
DISSERTATION

von
Dipl.–Math. techn. Fabian Bülow
aus Rheinfelden

Referent:	Prof. Dr.–Ing. H. Nirschl
Korreferent:	Prof. Dr. W. Dörfler

Tag der mündlichen Prüfung: 10.12.2015

Acknowledgements

First, I would like to express my gratitude to my two advisors, Professor Nirschl and Professor Dörfler, for giving me the opportunity to pursue my PhD studies in the field of process engineering. Both were always willing to help when I had questions. They complemented each other very well, each supporting me with knowledge in their field of expertise. Professor Nirschl taught me an engineering approach to problems and that the process of writing successful proposals for research projects takes time. Professor Dörfler helped me improving my manuscripts by very detailed corrections, sharpening my eye for details, and answered any math-related question I approached him with. Most of all, I thank my advisors for giving me the immense freedom I had with respect to my research. I highly acknowledge this.

Worth mentioning is definitely the financial support from the German Research Foundation (DFG) and the computational support from the Steinbuch Centre for Computing (SCC), which provided the necessary computational power.

Of course I thank all my colleagues, foremost my office mates, for making the last years very enjoyable. I was never running out of chocolate or someone to discuss any, not necessarily research-related subject with. Everyone in the two workgroups I was working in was always very helpful and open for discussions. I would like to thank our technicians, my colleagues from the particle measurement group and Kerstin Bertsch for making conducted experiments possible.

I also thank the students, who contributed to this work and who trusted me as their supervisor during their thesis or their job as research assistant. Namely, these are Philipp Hamberger, Nicolas Hesse, Daniel Knapp, Jonas Bretz, Amirah Bajawi, Sebastian Bäum, Marc Neumann, Adam Zietak and Jonas Hansert.

Special thanks go to my family, who always supports me. Without my father, my mother and my sister I would not be where I am now. Last but not least, I thank my wife for her constant support and for her great patience during the time in which this work has been written.

Abstract

The present work deals with the investigation of destabilizing suspensions. The stability of such is determined by two basic processes: Clustering and sedimentation. Elementary effects, which influence suspension stability, are explained in an introductory part. Because numerical simulations of destabilizing suspensions rely on the mathematical description of these effects, modeling approaches are addressed as well. After a discussion of currently used hindered settling functions, an overview of the most widely used methods for suspension simulation is given. The methods are described in such a way that similarities between different methods become clear. A rough characterization of existing methods can be made by means of the modeling approaches for different phases. This results in a classification of methods in Eulerian-, Lagrangian- and Eulerian–Lagrangian methods.

Results presented in this work have been obtained using a Lagrangian method, the so-called Stokesian Dynamics (SD) method. This method allows for a simple inclusion of models for particle-particle interactions of non-hydrodynamic nature. Moreover, hydrodynamic interactions of particles are accounted for without a direct solution of the basic equations describing the fluid phase. This is an advantage but also a limitation of the SD method. Especially the inclusion of external geometries, for instance the geometries of settling tanks, centrifuges, or CSTRs (continuously stirred tank reactors) pose extreme difficulties for the method. On the other hand, hydrodynamic interparticle interactions in polydisperse systems can be computed very efficiently. The results section of this work is confined to investigations of problems on the particle level, such that the described drawback plays no role.

Certain effects in suspensions can only be observed by means of numerical experiments when the number of simulated particles is high enough. Especially when it comes to polydisperse particle systems, reported direct numerical simulations, i.e. simulations in which hydrodynamic interactions are described by as few modeling as possible, are often limited to systems of less than a thousand particles. This number does not suffice to show e.g. for isolated settling hydrodynamic particle clusters, also known as particle clouds, an experimentally and numerically verified behaviour. Even though the SD Method is one of the most efficient methods for the

simulation of such problems, a sequential implementation requires simulation times of several days up to weeks and is typically limited to not more than one or two thousand particles. This limitation gave rise to the parallelization of the open-source simulation framework RYUON as part of this work. The aim of this parallelization is the numerical simulation of destabilizing suspensions on CPU-based supercomputers in a reasonable time span.

Results concerning this aim are presented in the first part of the results section of this work. It is shown that a parallelization pushes the limits of particle numbers which can be simulated within days from about a thousand to a million. The exact number of particles in systems which can be simulated within a given time span depends on the number of used CPUs. Computational time is also for a parallelized SD method quadratically dependent on the number of simulated particles. To reduce this complexity, the SD method has been coupled with the so-called Barnes–Hut method in the present work. The Barnes–Hut method has been developed for the solution of stellar dynamics problems and is nowadays applied to simulations in all fields where long-ranged interparticle interactions play a role. In the context of this method, particles can be stars, particles met in the field of process engineering, plasma ions, etc. The hybrid method resulting from the coupling of the SD and the Barnes–Hut method is described in the second part of the results section of this work. Apart from a detailed description of the method, results are presented, which show that the resulting method can indeed exhibit a better scaling than the parallel method, when it comes to increasing particle numbers. The resulting SSD method spatially groups particles during computation of hydrodynamic interactions according to a fixed criterion, which yields the reduction in computational time. Because of the motion of particles it is difficult to make a clear statement about the resulting complexity of the method. Yet, it is demonstrated in this work that complexity lies somewhere between linear complexity and the complexity of the parallelized SD method. Due to the dynamical grouping of particles, the timestep width is reduced when only hydrodynamic interparticle interactions are considered (and if an adaptive time-stepping scheme is applied). When additionally interactions due to electrostatics and London–van–der–Waals attraction are considered, no reduction is observable.

In the third part of the results section of this work, results concerning the numerical investigation on the settling behaviour of polydisperse particle clouds in viscous fluids are presented. The gained possibility to simulate large particle systems allows for the demonstration that polydisperse particle clouds undergo the same characteristic motion as monodisperse clouds of equal size. Because of the mentioned limitation of particle numbers in earlier works, consensus up to this work was that polydisperse systems would not be stable enough to show the typical behaviour. Apart from this, numerical experiments conducted in the context of this work indicate that the radius ratio of particles in an initially unmixed bidisperse cloud has a strong influence on the effect of small particles being dragged along by larger particles forming a hydrodynamic cluster. If both fractions are initially mixed, a larger number of small particles is dragged along by the fraction of large particles which determines the overall motion of the cluster. Another result of the described investigations is an analytical expression, which can be used to determine the maximum settling velocity of a polydisperse cloud. This expression yields a substantially better prediction than thus far applied formulas, which do not consider polydispersity.

The description of these results concerning hydrodynamic clustering are followed by an analysis of clustering due to heterocoagulation induced by hydrodynamic and non-hydrodynamic interactions which can be described by means of DLVO theory. On the example of orthokinetic heterocoagulation of two aluminium-oxide particles in water, it is set out how the standard procedure for the derivation of the so-called capture efficiency can be accelerated. The accurate determination of the capture efficiency is fundamental to a successful simulation of processes on a technical scale by means of population balance equations, e.g. in a crystallizer or in a settling tank. In this work, an expression for the capture efficiency of orthokinetic heterocoagulation is derived. For more complex problems, in which several effects (e.g. shear and settling) are superimposed, the standard procedure requires an extremely large number of experiments. Several thousand experiments have to be conducted before an expression for the capture efficiency can be gained. A statistical analysis of simulations of large systems circumvents several of the problems of the standard two-particle procedure: The formation and growth of clusters comprised of more than two particles can be examined. Moreover, the influence of other particles and of the solid volume fraction on the processes of coagulation and cluster growth are considered. In order to do so, a cluster detection algorithm has been implemented. The resulting post-processing tool

yields time-dependent information on the distribution of cluster settling velocities, of maximum Feret-diameters, and of the number of primary particles per cluster, as well as their radius of gyration and their fractal dimension. The named selection of values presents only the implemented ones. After a simulation time-resolved information on particle positions, -velocities, as well as -sizes is available, such that miscellaneous data can be obtained from this treatment.

In the course of this thesis, approaches for the simulation of non-spherical particles and aggregates by means of the SD method, as well as a coupling with a continuum mechanics approach for the solution of convection-diffusion equations describing the evolution of solved substances have been implemented in the overall simulation framework. The mentioned coupling has, with simultaneous consideration of growth kinetics and the effect of polymers, been applied to the numerical simulation of the dynamics in biofilms. Results were promising, yet not enough data has been collected in order to publish the results and to present them here. Apart from that, the settling behaviour of polydisperse particle systems in suspensions has been examined experimentally as well as numerically. During these investigations, an effect has been observed, which is not covered by thus far proposed hindered settling functions. The resulting deviation is attributed to the formation of hydrodynamic clusters at moderate solid volume fractions. Because research in this observation is ongoing, results have not been included in this work.

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Untersuchung von sich destabilisierenden Suspensionen. Bei solchen sind zwei grundlegende Vorgänge von besonderer Bedeutung: Clusterbildung und Sedimentation. Wesentliche Effekte, welche diese Vorgänge und damit die Suspensionsstabilität beeinflussen, werden zu Beginn dieser Arbeit vorgestellt und erklärt. Dabei wird auch auf Ansätze zur Modellierung eingegangen, denn erst die mathematische Beschreibung zugrundeliegender Prozesse ermöglicht deren Berücksichtigung in numerischen Simulationen, welche im Bereich der Verfahrenstechnik immer mehr an Bedeutung gewinnen. Nach einer

Diskussion der gebräuchlichsten Formeln zur Vorhersage der mittleren Sedimentationsgeschwindigkeit von Partikeln in Suspensionen werden daraufhin zur Zeit verwendete Methoden der numerischen Simulation von Suspensionen vorgestellt und so beschrieben, dass Zusammenhänge zwischen unterschiedlichen Methoden erkennbar sind. Eine grobe Charakterisierung existierender Methoden ist anhand des Ansatzes zur Beschreibung der unterschiedlichen Phasen möglich. Die resultierende Unterteilung ergibt Euler-, Lagrange- und Euler–Lagrange-Methoden.

Für die in dieser Arbeit präsentierten Ergebnisse wird eine Lagrange-Methode, die sogenannte Stokesian Dynamics (SD) Methode verwendet. Diese Methode erlaubt die einfache Einbindung von Modellen für Partikel–Partikel-Wechselwirkungen nicht-hydrodynamischer Natur. Darüber hinaus werden hydrodynamische Wechselwirkungen zwischen Partikeln berücksichtigt, ohne die hydrodynamischen Grundgleichungen direkt zu lösen. Dies stellt einen Vorteil, aber auch eine Einschränkung dar. Insbesondere das Einbeziehen externer Geometrien, wie zum Beispiel die für gewöhnlich abgerundeten Behälterwände eines Sedimentationsbehälters, einer Zentrifuge, oder eines Rührkessels stellt sich als extrem problematisch dar. Dem gegenüber steht die äußerst effiziente Berechnung hydrodynamischer Partikel–Partikel-Wechselwirkungen polydisperser Systeme, auf deren Untersuchung sich der Ergebnisteil dieser Arbeit beschränkt.

Gewisse Effekte in Suspensionen lassen sich in numerischen Untersuchungen erst beobachten, wenn die Anzahl an simulierten Partikeln hoch genug ist. Gerade bei polydispersen Suspensionen beschränkten sich Untersuchungen mit Hilfe von direkten numerischen Simulationen, d.h. mit möglichst genauer Beschreibung hydrodynamischer Wechselwirkungen sowie Auflösung von Strukturen, für gewöhnlich auf Partikelsysteme von unter eintausend Partikeln. Diese Anzahl ist nicht ausreichend, um zum Beispiel bei isolierten sedimentierenden hydrodynamischen Clustern, sogenannten Partikelwolken, ein experimentell und numerisch nachgewiesenes typisches Verhalten zu beobachten. Obwohl die SD Methode eine der effizientesten Methoden für solche Probleme darstellt, benötigt eine sequentielle Implementierung dieser Methode für eine Simulation Rechenzeiten von mehreren Tagen bis hin zu Wochen und ist typischer Weise beschränkt auf Partikelanzahlen von unter ein- bis zweitausend. Aufgrund dieser Einschränkung wurde die Open-Source Simulationsplattform RYUON im Rahmen dieser Arbeit parallelisiert, mit dem Ziel die Simulation von sich destabilisierenden Suspensionen mit Hilfe von CPU-basierten Supercom-

putern in einer vertretbaren Zeit zu ermöglichen.
Im ersten Ergebnisteil der vorliegenden Arbeit wird gezeigt, dass eine Parallelisierung die Grenzen der innerhalb von Tagen simulierbaren Partikelzahlen stark nach oben verschiebt — von etwa tausend zu einer Million Partikeln. Die genaue Anzahl der bei gleicher Rechenzeit simulierbaren Partikel hängt dabei von der Anzahl der verwendeten CPUs ab.
Da die Rechenzeit allerdings auch bei einer parallelen Simulation noch quadratisch von der Anzahl der Partikel abhängt, wurde die SD Methode im Rahmen dieser Arbeit mit der Barnes–Hut Methode verknüpft. Letztere wurde für Probleme in der Astrophysik entwickelt und findet überall dort Anwendung, wo langreichweitige Interaktionen zwischen simulierten Partikeln (Sterne, Partikel im verfahrenstechnischen Sinne, Plasma-Ionen, etc.) auftreten. Im zweiten Ergebnisteil dieser Arbeit wird die resultierende Hybridmethode präsentiert. Des Weiteren werden Ergebnisse vorgestellt, welche belegen dass die resultierende SSD Methode ein besseres Verhalten bei einer Erhöhung der Partikelanzahl aufweist, als die parallelisierte SD Methode. Die SSD Methode gruppiert Partikel für die Berechnung hydrodynamischer Interaktionen dynamisch nach einem festen Kriterium, wodurch die Rechenzeitreduktion erzielt wird. Aufgrund des dynamischen Verhaltens der Partikel ist es schwer, eine Aussage über die resultierende Komplexität zu treffen. Allerdings wird gezeigt, dass die Komplexität der SSD Methode zwischen linearer Komplexität und der der parallelisierten SD Methode liegt. Aufgrund der dynamischen Gruppierung von Partikeln sinkt der Zeitschritt bei reiner hydrodynamischer Interaktion zwischen Partikeln, was allerdings bei zusätzlicher Berücksichtigung von elektrostatischen und van–der–Waals-Interaktionen zwischen Partikeln (DLVO-Interaktionen) nicht mehr der Fall ist.
In einem dritten Ergebnisteil dieser Arbeit werden Ergebnisse vorgestellt, welche durch numerische Untersuchungen des Sedimentationsverhaltens polydisperser Partikelwolken in viskosen Flüssigkeiten erzielt wurden. Die Möglichkeit der Simulation größerer Partikelsysteme erlaubt den Nachweis, dass polydisperse Wolken das gleiche charakteristische Verhalten aufweisen wie monodisperse Wolken gleicher Größe. Aufgrund der begrenzten Möglichkeit zur Simulation größerer Partikelsysteme herrschte bis zu dieser Arbeit die Meinung, dass polydisperse Systeme dafür zu instabil seien. Des Weiteren wird in dieser Arbeit durch numerische Untersuchungen nachgewiesen, dass das Radienverhältnis in anfangs nicht durchmischten bidispersen Wolken einen großen Einfluss auf den Mitnah-

meeffekt von kleinen Partikeln hat. Mischt man beide Fraktionen, so wird ein größerer Anteil an kleinen Partikeln durch die dominierende Fraktion an großen Partikeln in der Partikelwolke eingeschlossen. Über diese Betrachtungen hinaus wird eine analytische Formel zur Vorhersage der maximalen Sinkgeschwindigkeit einer polydispersen Wolke hergeleitet, welche für polydisperse Systeme eine wesentlich bessere Vorhersage liefert, als bisher verwendete Formeln, die Polydispersität nicht berücksichtigen.
Auf die beschriebenen Ergebnisse von Untersuchungen hinsichtlich des Verhaltens hydrodynamischer Cluster folgt in einem vierten Ergebnisteil die Beschreibung von Resultaten der Analyse von Clusterbildung durch Heterokoagulation aufgrund von hydrodynamischen und DLVO-Partikel-Partikel-Wechselwirkungen. Am Beispiel der orthokinetischen Heterokoagulation zweier Aluminiumoxidpartikel in Wasser wird dargelegt, wie das Standardvorgehen zur Bestimmung der sogenannten Kollisionseffizienz beschleunigt werden kann. Die korrekte Beschreibung der Kollisionseffizienz ist essentiell für Simulationen in technischem Maßstab, zum Beispiel für Simulationen von Vorgängen in Kristallisatoren oder Sedimentationsbecken mittels Populationsbilanzgleichungen. Im Zuge dieser Nachforschungen wird daher ein analytischer Ausdruck für die Kollisionseffizienz für orthokinetische Heterokoagulation bestimmt. Dieses Vorgehen erfordert jedoch bei komplexeren Problemen, bei denen unterschiedliche Einflüsse überlagert sind (z.B. Scherung und Sedimentation), eine extrem hohe Anzahl an Experimenten. Es müssen mehrere tausend Experimente mit zwei Partikeln durchgeführt werden, um einen Ausdruck für die Kollisionseffizienz bestimmen zu können. Eine statistische Auswertung von Simulationen großer Partikelsysteme umgeht mehrere der Probleme von zwei-Partikel Experimenten: Es kann die Bildung von Clustern, welche mehr als zwei Partikel umfassen, sowie deren Wachstum untersucht werden. Darüber hinaus wird der Einfluss anderer Partikel, sowie der Einfluss des Feststoffvolumenanteils auf den Koagulationsvorgang und das Clusterwachstum berücksichtigt. Hierfür wurde ein entsprechender Clustererkennungsalgorithmus implementiert, welcher anhand von Simulationsdaten zeitlich aufgelöste Informationen wie die Verteilung von Clustersinkgeschwindigkeiten, maximalen Feretdurchmessern, der Anzahl der Primärpartikel pro Cluster und deren Gyrationsradius sowie deren fraktale Dimension liefert. Bei den hier aufgelisteten Attributen handelt es sich lediglich um die implementierten. Da nach einer Simulation zeitlich aufgelöste Informationen über Partikelpositionen, -geschwindigkeiten, und

-größen bekannt sind, lassen sich auch diverse andere Daten ableiten.
Im Rahmen dieser Arbeit wurden auch Ansätze zur Simulation nichtsphärischer Partikel und Aggregate mit der SD Methode, sowie eine Kopplung mit einem kontinuumsmechanischen Ansatz zum Transport von gelösten Stoffen in der entwickelten Simulationsplattform implementiert.
Letzterer wurde dazu verwendet, unter Berücksichtigung von Wachstumskinetiken die Dynamik in Biofilmen zu untersuchen. Der resultierende Ansatz ist sehr vielversprechend, allerdings wurden noch nicht genügend Ergebnisse gesammelt, um diese im Rahmen dieser Arbeit zu präsentieren.
Darüber hinaus wurde das Sedimentationsverhalten von polydispersen Suspensionen sowohl experimentell als auch numerisch untersucht, wobei eine von bisher entwickelten Formeln zur Beschreibung der mittleren Sedimentationsgeschwindigkeit nicht berücksichtigte Überhöhung aufgrund von hydrodynamischer Clusterbildung festgestellt wurde. Da sich auch diese Nachforschungen noch in unveröffentlichem Zustand befinden, wird darüber im Rahmen dieser Arbeit ebenfalls nicht berichtet.

Contents

Chapter 1

Introduction, basics and overview

The present work deals with the investigation and numerical simulation of the destabilization behaviour of heterogeneous particle systems in suspensions. The focus is set on the deployment of a numerical framework capable to simulate comparably large particle systems while taking all significant effects into account. With the resulting framework, the settling behaviour of particles in suspensions and the destabilisation due to coagulation are investigated.

In this chapter we first give an overview of the field of heterogeneous destabilizing suspensions, where we focus on destabilization and settling — two major effects in solid-liquid separation. This is done in sections 1.1, 1.2, 1.3, and 1.4. Subsequently, in section 1.5, we give an introductory overview of currently used simulation methods in the field of suspensions. We conclude this chapter by giving an outline of the following chapters, which describe done research and obtained results.

1.1 Heterogeneous suspensions

Heterogeneous systems of particles and the separation of such from a continuous liquid phase can be found in almost every field in process engineering. In paper recycling used paper is broken down, chopped up and heated to become the so-called pulp. This pulp is a highly heterogeneous

suspension containing glue, plastic and ink particles, as well as cellulose fibres in varying proportions. In wastewater treatment one can find a large size and density spectrum of particles. Ranging from trash, which is filtered out by large grills, over sand and other particles of relatively high density to different kinds bacteria, growing primarily in the aeration tank, and colloids, which are particles of about 100nm to 10μm in size. Mineral processing is another area where heterogeneous suspensions and the correct treatment of such can play an important role. After a size reduction, particles are sized and, if in suspension, concentrated and separated from the liquid. Usually the intermediate product is a metal in oxide or sulfide form and is further processed involving another operation of solid-liquid separation. In chemical production of polymers, suspension polymerization is a widely used process employed to produce e.g. PMMA (polymethyl methacrylate) or PVC (polyvinyl chloride) particles. Before solid-liquid separation, PVC particles grow suspended in water from monomers to particles of different sizes, mostly ranging from 50μm to 200μm.
These are just a few examples which illustrate the vast spectrum of heterogeneity in suspensions of industrial relevance. To sum up, particle systems can be heterogeneous due to:

1. shape
2. size
3. material
4. state of matter
5. phase
6. other physical properties (density, elasticity, electric charge, etc.)
7. chemical properties (enthalpy of formation, chemical stability, etc.),

where these points can depend on one another.
Because of the broad spectrum of heterogeneity in particle suspensions in process engineering in general and in solid-liquid separation in particular, the points in the above list are treated each in different and to limited extent in this work.
We confine ourselves to the consideration of a single continuous liquid phase, which can act as solvent for several solute substances. Because of the limitations of the method which we use to compute hydrodynamic

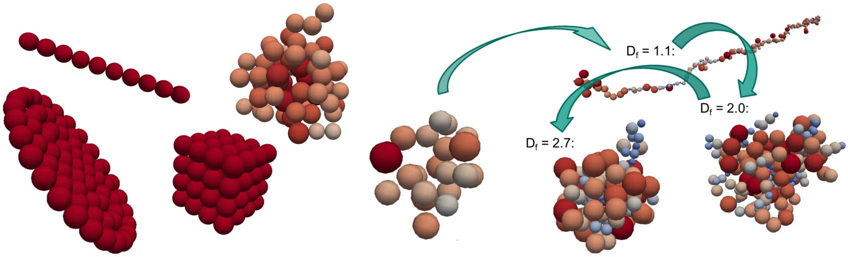

Figure 1.1: Illustration of the possible aggregate and agglomerate structures in the developed simulation platform. D_f denotes the corresponding fractal dimension.

interactions among particles, all simulated primary particles are of spherical shape. Examples of implemented shapes of particles, agglomerates and aggregates built of spherical primary particles are shown in figure 1.1. The fractal aggregates on the right hand side of figure 1.1 are constructed by means of an imlementation of the tunable dimension method presented in [Kätzel et al., 2008] for polydisperse primary particles. In this work we follow the German industry norm DIN 53206 in which an aggregate is defined as sintered cluster of primary particles which has a smaller surface area than the sum of the surface areas of the primary particles it consists of. According to this norm an agglomerate is a non-sintered cluster of primary particles or aggregates, whose surface area does not significantly deviate from the sum of the surface areas of the particles it consists of. A floc is defined as agglomerate, which can be broken easily, e.g. by low shear forces. One point considered important for this work is that all used models consider a possible polydispersity of the investigated particle system. This is due to the fact that real particle systems always exhibit polydispersity, however narrow the size distribution may be. The particle size is theoretically not limited in the framework presented in the following, but due to the limitation of the simulation method yielding hydrodynamic interactions among particles we confine ourselves to a consideration of suspensions for which the particle Reynolds number as well as other Reynolds numbers, such as an aggregate or cloud Reynolds number, are so small that they can be considered vanishing. This means that the Stokes equations and the mass conservation equation for an in-

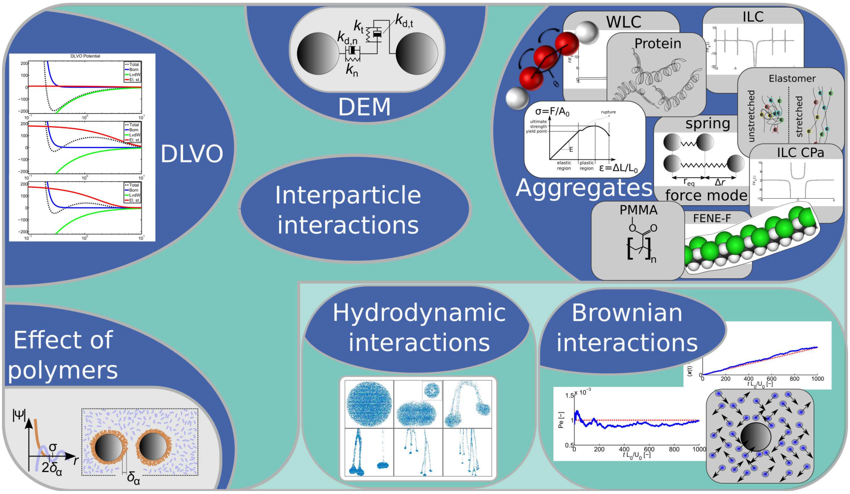

Figure 1.2: Overview of implemented models for interparticle interactions considering physical properties of particles.

compressible Newtonian fluid suffice to describe the motion of the fluid. Due to nondimensionalization of the basic equations (cf. e.g. section 5), the settling of millimeter-sized glass beads in a highly viscous oil can be considered equally well as that of colloidal Al_2O_3 particles in water. An extension to the used simulation code which allows for the numerical simulation of biological particles (bacteria with attached EPS), makes simulations of growing, dividing and dissolving particles possible. Our simulation-based investigations are therefore not limited to particle systems with a fixed size spectrum, as long as the underlying kinetics are known. Likewise, the material of the simulated particles is not limited to one material. A mixture of different particles such as biological and non-biological particles can be investigated. Other physical properties of particles are considered in different models for interparticle interactions, as depicted in figure 1.2. Chemical properties are considered inasmuch as they are part of the metabolism of simulated bacterial species. Other reactions as well as chemical properties are not considered but possible.

1.2 Reasons for destabilization

Heterogenoeus suspensions, especially such containing colloids or nanoparticles, tend to destabilize. Destabilization is the process when otherwise non-settling dispersed particles in a suspension form clusters which can be large enough to settle due to external body forces such as gravity or centrifugal forces. This can happen naturally due to the hydrodynamic or physico-chemical conditions in the suspension, or it can be induced by changing these conditions (cf. figure 1.3). Depending on the type of formed clusters and the way the suspension was destabilized, the process can be reversed. This is then called stabilization, as can also be seen in figure 1.3. Usually additional work has to be done on the system in order to stabilize it again, e.g. by ultrasonic dispersion. Mixing can promote destabilization or re-stabilization, depending on the mixing speed, i.e. the resulting shear rate [Russel et al., 1992]. In the following we describe the main reasons for destabilization.

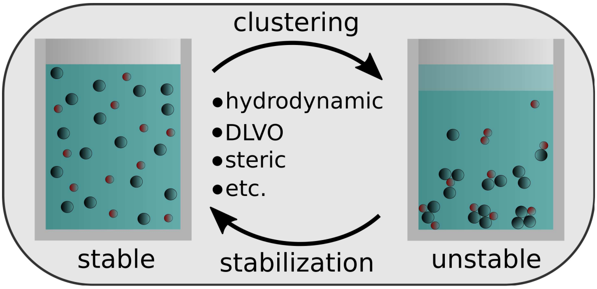

Figure 1.3: Reasons for destabilization and methods to stabilize a suspension. Left: Stable suspension. Right: Destabilized/unstable suspension.

1.2.1 Hydrodynamic clustering

Destabilization can happen because of hydrodynamic clustering, which is the formation of structures (clusters) of particles due to purely hydrodynamic interactions. Caused by hydrodynamic clustering, particle fractions which would actually separate after a certain time stay together for a prolonged period of time and can form more stable clusters. This can be beneficial or disadvantageous for the respective process. So are particle clusters in bubbling fluidized beds desired in processes of fluidized bed granulation, and not wanted in case of reaction processes such as fluid catalytic cracking (FCC) or fluidized bed combustion. In solid-liquid separation, hydrodynamic clustering can be advantageous, namely when a fluid shall be purified and particles which settle under the given conditions drag smaller particles along in formed clusters. Applications here are e.g. settling ponds or decanter centrifuges. In classification, e.g. in a solid bowl centrifuge or in an (air) classifier, where particles are to be separated by means of surface forces and body forces, this effect evidently worsens the process result. Owing to the hydrodynamics as cause for clustering, purely hydrodynamic clusters are usually of changing shape, size and duration, differing from the conventional image of clusters. Furthermore, hydrodynamic forces can facilitate clustering of relatively large particles, i.e. particles of a size above 100 microns. Because of the long ranged nature of hydrodynamic interactions, which decay in a non-turbulent flow as $1/r$, where r is the interparticle distance [Brady and Bossis, 1988], hydrodynamic clustering can promote longer lasting clustering due to other driving forces. Examples of hydrodynamic clustering are shown in figure 1.4, where arrows indicate characteristic flow behaviour. Hydrodynamic clustering is investigated for polydisperse clouds of particles in chapter 4 of this work.

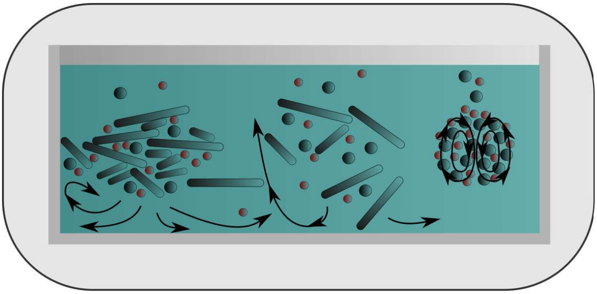

Figure 1.4: Examples of hydrodynamic clustering in a heterogeneous suspension of settling particles. Arrows indicate the flow direction.

1.2.2 Non-hydrodynamic forces as cause for clustering

A main cause for the destabilization of suspensions and the formation of more stable clusters such as flocs, agglomerates or aggregates are direct and indirect interparticle interactions of non-hydrodynamic nature. We start this subsection by a more in-depth explanation of the so-called DLVO interactions, followed by other major causes for destabilization.

DLVO interactions

A theory developed for colloidal particles called DLVO theory, named after Derjaguin, Landau, Verwey and Overbeek [Verwey et al., 1999], states that interparticle potentials arising from London–van der Waals (LvdW) attraction and electrostatic repulsion can be superimposed to give a combined interaction potential, the so-called DLVO potential. This potential is often combined with the Born potential [Feke et al., 1984], to consider the extremely short ranged repulsion that developes as particles begin to contact. With an inclusion of the Born potential, the appropriate examination of reversible coagulation or adsorbtion phenomena and the simulation of structuring and ordering within colloidal systems is possible [Feke et al., 1984]. This is due to the fact that with an inclusion of Born repulsion three different kinds of interparticle potentials are possible: Potentials that exhibit only a primary minimum, potentials that exhibit only a secondary minimum and potentials that exhibit both, a primary and a secondary minimum. In this context 'primary minimum' means the minimum located close to a surface-to-surface distance of $r_0 = 0$ (touching particles). If a potential barrier resulting from electrostatic repulsion exists, the primary minimum is situated between $r_0 = 0$ and the potential barrier. The secondary minimum is per definition the potential minimum located at a point for which r_0 is larger than the location of the potential barrier (cf. figure 1.5). Without a consideration of Born repulsion, interparticle attraction would become infinitely strong for interparticle distances approaching zero, which is not physical [Russel et al., 1992]. Apart from the works giving the expressions we use for the different contributions to the DLVO potential [Feke et al., 1984, Hamaker, 1937, Hogg et al., 1966, Bell et al., 1970], [Bhattacharjee et al., 1998] includes

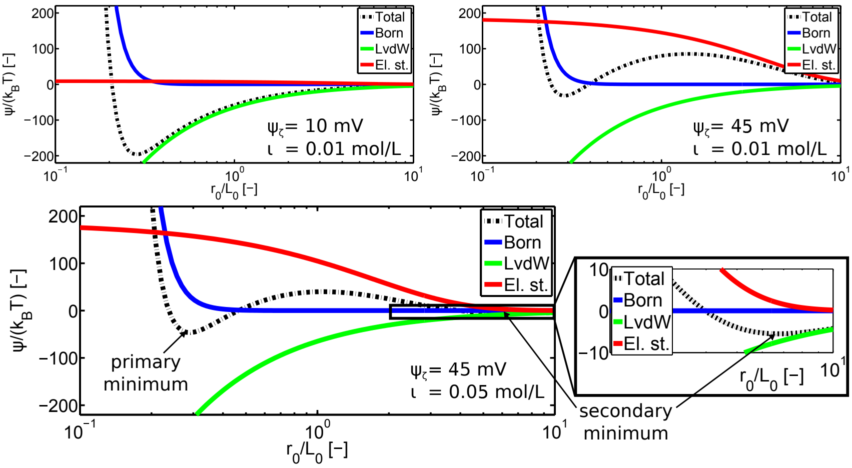

Figure 1.5: Three DLVO potentials arising from exemplarily values for the model parameters with differing surface potential ψ_ζ and ionic strength ι.

a good comparison of different models for electrostatic and LvdW interactions with respect to their validity. As one can see in figure 1.5, DLVO theory can give an explanation for the stabilizing or destabilizing effect of added salts (ions). In figure 1.5 top left, the resulting or total DLVO potential exhibits a deep primary minimum and no potential barrier. The corresponding suspension is unstable, as particles attract each other until the reach their potential minimum. Because this minimum is comparably deep, the resulting agglomerates are generally stable. If particle roughness allows for it, they can undergo a compaction due to the strong forces attracting the primary particles in the agglomerate. Figure 1.5 top right shows the potential for a suspension of the same ionic strength ι, but with particles having a higher surface potential ψ_ζ. Here one can see a potential barrier, as the ions in the suspension are not sufficient to shield the stronger electric field. If no other influences, such as strong Brownian motion, shear, or other forces like magnetic forces affect the motion of the particles, this potential barrier will not be overcome and the suspension is stable. For the potentials depicted in the plot in figure 1.5 bottom, the ionic strength has been increased from $\iota = 0.01$mol/L to $\iota = 0.05$mol/L, compared to figure 1.5 top right. The result is a stronger decrease of the electrostatic potential yielding a lower potential barrier and an emerging secondary minimum. Particles in such a suspension form weaker agglomerates or flocs. The stronger decrease in the electrostatic potential, which in this case is also called Coulomb potential or double layer potential, is a result of the thicker hull of ions around the charged particles. This hull is usually called electrical double layer (EDL) or diffuse double layer. The currently most widely accepted model for the electrical double layer around a charged particle in an electrolyte, and for the corresponding potential, is the BDM model by Bockris, Devanathan and Müller [Bockris et al., 1963], which is illustrated in figure 1.6. Apart from proposing the BDM model, Bockris et al. [1963] give a review on former models. Among them that by Grahame [1947] who stresses that besides solvent molecules some ionic or uncharged molecules can penetrate the Stern layer and adsorb specifically to the surface of a particle. The adsorbtion is mostly driven by London–van der Waals attraction. During penetration of the Stern layer ions lose their solvation shell [Stojek, 2010] and can therefore adsorb closer to the particle surface than ions which still have a shell of solvent molecules [Grahame, 1947]. Bockris et al. [1963] extend the model by Grahame [1947] for a contribution of field dependendly oriented solvent dipoles and the contribution from adsorbed solvent dipoles to the double-layer capacity. In the BDM model, the boundary of the Stern layer is also called outer Helmholtz plane (OHP, cf. figure 1.6), the plane through the centers of specifically adsorbed ions and field orented solvent molecules is named inner Helmholtz plane (IHP). The layer beyond the IHP up to the slipping plane is called diffuse layer due to the loose association of the molecules in it with the charged particle. More information on the electrical double layer can be found in literature on colloidal dispersions, e.g. in [Russel et al., 1992], and in literature on electrochemical capacitors, e.g. [Stojek, 2010].

Important for engineering applications is the zeta potential ψ_{zeta} at the location of the slipping plane of the electrical double layer, κ^{-1}. The Debye length κ^{-1} can be computed for colloidal suspensions and depends, among other paramters, on the square root of the inverse ionic strength ι [Russel et al., 1992]. This dependency is considered in DLVO theory and can be directly seen when comparing the plots of the elctrostatic pontential in figure 1.5 top right and bottom.

As indicated in figure 1.6, the electric surface potential of a particle, ψ_{ζ}, as well as its Stern potential ψ_{Stern} usually do not equal the zeta potential

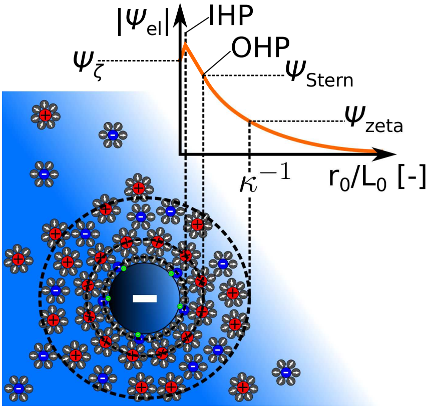

Figure 1.6: Illustration of the electrostatic potential ψ_{el} of a negatively charged particle in a suspension according to the BDM model. IHP and OHP denote the locations of the inner and outer Helmholtz plane, ψ_{Stern} and ψ_{zeta} the Stern potential and the zeta potential, respectively. The Debye length κ^{-1} indicates the location of the slipping plane. Ions and solvent molecules have been enlarged compared to the particle for better visibility. Arrows on the solvent molecules show their orientation.

ψ_{zeta}, which is also emphasized by Kirby [2010]. Nevertheless, the zeta potential is often the most easily accessible or the only available way to quantify the electric potental of charged particles in an electrolyte and it is common practice to identify these potentials with one another [Hecht et al., 2006]. For electrical double layers which are thin compared to the particle radius a, i.e. $\kappa a \gg 1$, this assumption can be made. Therefore, we usually set $\psi_\zeta \approx \psi_{\text{zeta}}$ in the following. Generally, the absolute value of the zeta potential for colloidal suspensions is between 0 and 100mV. As a rule of thumb, a zeta potential with high absolute value corresponds to a stable suspension, whereas a value of ψ_{zeta} around zero, i.e. around the point of zero charge or iso-electric point (IEP), means that the considered suspension can destabilize easily due to negligible electrostatic repulsion. There are other ways to stabilize a suspension (cf. figure 1.3), so $\psi_{\text{zeta}} \approx 0$ does not necessarily mean a suspension is unstable, but a zeta potential around the IEP is a good indicator.

A quantitative determination of the zeta potential can be done indirectly utilizing electrokinetic or electroacoustic phenomena. For this, an external electric field is applied to the investigated suspension, and a constant viscosity of the fluid, a low Reynolds number and a moderate electric field strength are assumed. Ohms law can be interpreted and applied to the drift velocity of an electron, and more generally to that of a particle carrying a non-zero charge. Solving for the electrical mobility, one gets an expression for the latter, which is given by the ratio of particle drift velocity to the magnitude of the applied electric field. This ratio can then be related to the relative permittivity of the suspending medium, the vacuum permittivity and the zeta potential divided by the dynamic viscosity of the fluid [Smoluchowski, 1924]. So, roughly speaking, the zeta potential can be determined experimentally from the magnitude of the applied electric field, the dielectric constant and the dynamic viscosity of the fluid, provided that the drift velocity of a particle can be measured. The described method is called electrophoresis and utilizes electrokinetic phenomena (the drift of a particle in an applied electric field and the appearing retardation force due to the EDL). When electroacoustic phenomena are exploited, not the drift velocity but an emerging current or generated ultrasound waves are measured. The resulting current is proportional to the volume fraction of particles, the scaled mass density difference and the electric mobility. In general, the described Smoluchowski relation or similar relations between the electrical mobility and the zeta potential are used [O'Brien et al., 1995]. The Smoluchowski relation is only valid for electrical double layers which are thin compared to the particle radius a, i.e. $\kappa a \gg 1$. Another correlation with the same set of parameters developed by Hückel [1924] is valid for $\kappa a < 1$. There is a large number of correlation equations, taking different effects such as e.g. surface conductivity of the particles into account [Henry, 1948]. Current research shows that excess electrical charge on the surface of TiO_2 particles can be responsible for surface conductivity, which can significantly lower electrophoretic measurements and thus the apparent zeta potential [Leroy et al., 2011]. A dimensionless number relating the surface conductivity to the bulk electrical conductivity is the so-called Dukhin number Du. The Smoluchowski relation is only valid for low values of Du, i.e. negligible surface conductivity. For further reading, the interested reader may be advised to read the work by Dukhin and Goetz [2010] or Lyklema [2005].

In general, surface conductivity is neglected by commercially available systems for zeta potential measurements. The two systems we made use of in order to determine the zeta potentials taken for our numerical experiments considering agglomeration are the Zetasizer Nano ZS [zet, 2015] and the Acoustosizer II [aco, 2015]. Both systems measure the electric mobility of the particles, but they rely on different phenomena. The Zetasizer uses electrokinetic phenomena and measures the electric mobility of the considered particles by dynamic light scattering (DLS), also called photon correlation spectroscopy (PCS), where the electric mobility can be related to the Doppler shift frequency of an incident laser beam [Russel et al., 1992]. The Acoustosizer relies on electroacoustic phenomena, which are the change in an applied electric field due to ultrasound waves and vice versa. The respective effects are called colloid vibration current (CVI) and electric sonic amplitude (ESA). While for CVI the particles in the considered suspension are moved by ultrasound and a change in the electric field is measured, ESA is the generation of ultrasound due to an oscillating electric field [Dukhin and Goetz, 2010]. The Acoustosizer utilizes ESA, as is well explained in [O'Brien et al., 1995]. Because of the different measuring principles and the assumptions made to relate the measured signal to the electric mobility of the particles, both systems operate at different regimes of solid volume fractions in the measured suspensions. Where the Zetasizer provides reliable results only in the very dilute regime with solid volume fractions $\phi \ll 0.001$, the Acoustosizer gives good results only for more concentrated suspensions with $\phi \geq 0.01$. Both systems can work with different correlations between electrical mobility and zeta potential, but usually rely on the Smoluchowski relation described earlier. Besides information on the particle size, measurements yield the value of the zeta potential together with electric mobility, conductivity and pH of the suspension. The Acoustosizer further gives a value for κa, from which the ionic strength can be computed. There are theoretical approaches which correlate the pH value and the zeta potential [Hecht et al., 2006], but a practical test showed that the prediction of the zeta potential from the pH value alone, at least by the tested correlation given in [Hecht et al., 2006], yields poor results.

Comparisons of results obtained from measurements using electrokinetic and electroacoustic phenomena should be handled with care. When the ionic strength is not adjusted accordingly, the double layer thickness and thus the zeta potential varies with solid volume fraction and with particle size. This is actually hardly surprising if one recalls the BDM model illus-

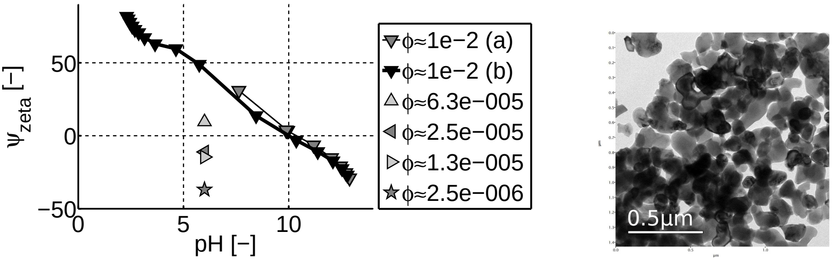

Figure 1.7: Left: Exemplarily comparison of results from measurements for the zeta potential ψ_{zeta} over pH value of Al_2O_3 particles in water. The data for the solid volume fraction $\phi = 0.01$ was obtained using the Acoustosizer, whereas the single data points are from measurements using the Zetasizer. Right: TEM image of the considered Al_2O_3 particles.

trated in figure 1.6, but it is often neglected in practice. Figure 1.7, left, shows exemplary results of measurements of Al_2O_3 particles in Milli-Q water at a solid volume fraction of $\phi = 0.01$ and at lower values of ϕ. The temperature was 293K, suspension samples have been treated by ultrasonic dispersion before measurements. The pH value has been adjusted by NaOH for an increase in pH and by HCl addition for a decrease in pH, respectively. The original pH was 4.5 for $\phi = 2.5\text{e}{-6}$ and around 6 for the other solid volume fractions analyzed with the Zetasizer. In case of $\phi = 0.01$, the original pH value was about 7.6. The measured mean particle size was for the Zetasizer approximately 255nm, the Acoustosizer gave a particle size range between 60nm and 400nm, which is in agreement with sample measurements using transmission electron microscopy (TEM), cf. figure 1.7, right. As one can see, the measured value of the zeta potential depends, apart from the ionic strength and the pH value, on the solid volume fraction in the tested suspension. Line (a) shows the zeta potential for an addition of NaOH, for line (b) HCl has been added to decrease the pH value. The main objective of this work is the numerical simulation of suspensions as the one exemplarily considered here. Therefore, even though it is an interesting field, we will not go into further details on dependencies of the measured zeta potential. Note that non-spherical and inhomogeneous particles carry a non-uniform surface charge, which can even differ in sign when comparing corners and even surfaces [Zhu, 2014].

As we consider polydispersity, but mostly simulate spherical particles, we assume a uniform surface charge for all particles. Inhomogeneities can be modeled by an assignment of individual surface charges to the primary particles of aggregates, agglomerates and flocs.
There are more forces influencing suspension stability. Depending on the considered system, these can be dominant or negligible. For the sake of completeness we describe them, but we do not go too deep into details.

Hydrophobicity

Apart from DLVO interactions, cluster formation can be caused by the tendency of hydrophobic (non-polar) particles in water or another polar fluid to minimize interfacial area. This effect can be modeled by a Lewis acid-base (AB) model [Van Oss, 1989, 1993]. The DLVO model which includes AB interactions is typically called extended DLVO model and gives in some applications a better description of interparticle interactions than the standard DLVO model [Hermansson, 1999].

Hydration forces

Forces resulting from the formation of a primary and a secondary hydration hull around charged particles are called structural or hydration forces [Feke et al., 1984, Grasso et al., 2002]. Here, primary hydration means the direct interaction of particle surface and water molecules, secondary hydration refers to the interaction of molecules adsorbed to the particle surface and those around solutes [Parsegian and Zemb, 2011]. Structural or hydration forces can be attractive or repulsive and play a role when surface-to-surface distances are of the order of a few nanometers. They can be dominant for macromolecules and small nanoparticles, where they can lead to spontaneous assembly and structuring [Todd et al., 2008]. These forces can also be taken into account when explaining strong repulsive forces before particle contact, which can be attributed to the disruption of hydrogen bonding networks between particle surfaces. A precise definition of (primary) hydration forces is given in [Zhang, 2013], where they are described as strong short-range repulsive forces that act between polar surfaces separated by a thin layer (< 3nm) of water, and which decay quasi-exponentially with the surface spacing. In case of other polar solvents, these forces are also called solvation forces. For further reading we refer to the given references. Zhang [2013] provides an expression for a model of hydration forces. We do not employ this model in our work because hydration forces are in our framework partially covered by the used model for lubrication forces [Bülow et al., 2016].

The effect of added polymers

Direct and indirect interparticle forces due to adsorbed and free polymers can be described by models for steric and depletion or osmotic forces [Feke et al., 1984, Lewis, 2000]. These come into play when a suspension contains soluble or non-soluble polymers. The influence of polymers can lead to stabilization or to destabilization. Both effects are superimposed and stability diagrams are usually plotted over the polymer concentration. Let us briefly clarify why both effects are superimposed.

Steric forces arising from adsorbed polymers are to a great extent repulsive. When two particles with adsorbed polymers approach each other such that their polymer layers overlap, the polymers are usually bent and exert a repulsive force on the approaching particles, which depends on the polymer stiffness, their concentration in the layer and other parameters. When the layers interpenetrate each other, mixing of the polymers can occur and forces such as LvdW forces and such arising from ionic and hydrogen bonds can result in attractive interparticle interactions. Depletion or osmotic forces result from the effect of solute or free polymers. The repulsive part of this effect comes from an occuring negative gradient in the polymer concentration between the bulk fluid and the volume of the interparticle gap, which lets polymers diffuse into the gap. The more important attractive part results from a concentration gradient of polymers as well. If the interparticle gap is so small that no polymer can enter it, an increased osmotic pressure in the surrounding solution presses the particles together [Russel et al., 1992]. From this description it is already clear that depletion forces mainly depend on the concentration and the size of non-adsorbed polymers.

There are two ways to include polymers in a suspension simulation. Either they are resolved as particles, or treated implicitly. Let us discuss the latter method first: If polymers are not resolved as particles, concentration transport equations have to be solved in order to account for non-adsorbed polymers (cf. subsection 1.5.1). This makes the introduction of an extra numerical background grid and the interpolation between particle positions and grid nodes necessary. To account for adsorbed polymers, individual polymer layers of dynamic thickness and concentration have to be included in the simulation. The temporal change in layer thickness and -concentration also requires an interpolation between moving particles and fixed background grid. This is additional numerical effort, as the model for hydrodynamic interactions chosen for this work, as well as most models for non-hydrodynamic interactions do not depend on a fixed numerical grid and do not require the solution of partial differential equations, such as time-dependent convection-diffusion equations for polymer transport. Furthermore, it complicates preservation of load balance. Roughly speaking, load balance means that during a simulation running on several interacting processors, all processors have an approximately even amount of work such that none of them is slower and others do not have to wait at an inevitable synchronization point. For grid free methods, data can be distributed among processors with distributed memory according to the spatial distribution of the particles [Bülow et al., 2014]. When using methods with a fixed background grid, data is usually assigned to grid nodes. The background grid is partitioned and corresponding memory is allocated in the memory of the respective processors. During a simulation particles can cross the boundaries of the domains assigned to different processors, which can result in a spatially anisotropic distribution of particles. If one wants to preserve load balance, this leads to extra communication. Either due to data transfer, because the data related to particles in a new domain is transfered to another processor, or due to the interpolation between particle locations and background grid, because particles in a domain have their data in the memory of a processor responsible for another domain. For now this should be enough information on implementation details, and we refer to [Bülow et al., 2014, 2016] for further reading. Yet it should have become clear that an inclusion of the effect of polymers without resolving them as particles is not for free. Nevertheless, this method is computationally cheaper than a resolution of all polymeric substances by means of simulated particles, as we show in the following.

When polymeric substances are resolved for suspension simulation, they are usually modeled as chains of spheres connected by bonds. Depending on the type of polymer, these bonds are modeled differently. This approach can be seen as coarse-grained molecular dynamics simulation [Hsieh et al., 2006, Somasi et al., 2002] and is widely used in Brownian dynamics (cf. subsections 1.5.1 and 1.5.2). The most applied bond models are: The worm-like chain (WLC) model [Marko and Siggia, 1995, Shaqfeh et al., 2004], the inverse Langevin chain (ILC) model [Kuhn and Grün, 1942, Arruda and Boyce, 1993], it's extension — Cohen's Pade approximation to the ILC [Cohen, 1991], the finitely extensible nonlinear elastic (FENE) model [Warner Jr, 1972], the FENE-Fraenkel model [Hsieh et al., 2006] and the discrete WLC model [Cifra and Bleha, 2007]. A simulation of particle suspensions including resolved polymers is possible with the developed framework, as all named models are implemented. Yet, this would be computationally expensive. Simulation time increases more than linearly with an increasing number of simulated particles (including the polymer segements of size of the respective persistence- or Kuhn length). For example one timestep of a hypothetical simulation of two particles with 100 adsorbed polymer chains each, where the chains consist of 10 discrete segments, would take about half an hour instead of the 7 milliseconds one timestep without a consideration of polymers would take. For this example we have assumed an increase in simulation time of $\mathcal{O}(N^{1.8})$, with the real increase being somewhere between linear and quadratic with respect to the number of particles, N [Bülow et al., 2014, 2016]. Not included in this rough estimate is the reduction in timestep width, resulting from the combination of effects on two different length scales in one simulation, namely that of large colloids and that of macromolecules. A treatment of polymers in the earlier described fashion only increases the simulation time by a few percent, as the solution of the concentration transport equations by means of a finite difference method and a linear interpolation between grid nodes and particle positions is computationally cheap. Because of this, we have included a model for unresolved polymeric substances in the simulation framework. The model for effects on particles is given in [Lewis, 2000] and accounts for steric and depletion interaction of particles due to adsorbed and free polymers. The resulting method is described in [Bülow et al., 2016].

1.3 Coagulation

According to the IUPAC definition, agglomeration, flocculation and coagulation are synonymous [McNaught, 2005]. Therefore, when speaking in general terms, we use the term coagulation for a process in which aggregates, agglomerates or flocs are formed. Coagulation can be classified by means of the driving force, which lets particles approach each other close enough to form the respective structures. In the case of Brownian motion being the dominant effect leading to coagulation, the process is called perikinetic. If, which is probably in technical processes the more frequently encountered case, hydrodynamic motion (e.g. convection or sedimentation) is the main cause for coagulation, the process is refered to as orthokinetic coagulation [Somasundaran, 2006]. Apart from this, there is a distinction between rapid and slow coagulation [Feke and Schowalter, 1983], as well as between primary and secondary coagulation. The latter classification is simply a more precise definition of the type of coagulation by refering to the preferred minimum of the interaction potential (cf. figure 1.5). Rapid and slow coagulation are loosely defined terms for primary and secondary coagulation. The primary minimum of the pair interaction potential of the particles is usually much deeper than the secondary minimum, resulting in stronger attractive forces and thus a faster pace of coagulation. In this work we confine ourselves to an investigation of orthokinetic slow coagulation, as this is usually the prevailing case in solid-liquid separation.

In the early stages of destabilization particles form so-called doublets. These are pairs of coagulated particles. The first systematic studies on coagulation and doublet formation, which led to a mathematical description of the so-called capture frequency, have been carried out by Smoluchowski [1917]. The assumptions made in his work, such as spheres of action outside of which particles do not interact, simplify the underlying problem strongly (see [Zhu, 2014] for a review discussing assumptions made by Smoluchowski and how they are commonly dealt with). Consequently there is a large number of works investigating orthokinetic as well as perikinetic coagulation of two particles under less assumptions [Curtis and Hocking, 1970, Van de Ven and Mason, 1976, 1977, Adler, 1981, Wang, 1992, Vanni and Baldi, 2002, Kobayashi, 2008, Balakin et al., 2012].

1.3.1 Orthokinetic coagulation

For the orthokinetic case, usually the interception of two particles in a simple shear flow is investigated. From this, one can obtain the capture efficiency α_o, which is defined as ratio of actual capture frequency β_{ij} to the capture frequency proposed by Smoluchowski $\beta_{ij,\mathrm{S}}$. The inverse of α_o is called stability ratio. With an inclusion of the capture efficiency, the adjusted expression for the orthokinetic capture frequency of particles of size classes i and j reads as [Feke and Schowalter, 1983, Van de Ven and Mason, 1977]

$$\beta_{ij} = \alpha_o \beta_{ij,\mathrm{S}} = \alpha_o \frac{4}{3} \dot{\gamma} (a_i + a_j)^3, \tag{1.1}$$

with the particle radii a_i, a_j and the shear rate $\dot{\gamma}$. Figure 1.8 (a) illustrates the typical setup for investigations in the orthokinetic coagulation of two particles. Figure 1.8 (b) shows the numerically obtained trajectories of two equal Al_2O_3 particles in water in the plane spanned by the flow direction and the direction of the velocity gradient of a simple shear flow with dimensionless shear rate $\dot{\gamma} = 1$. In this case, the initial particle separation

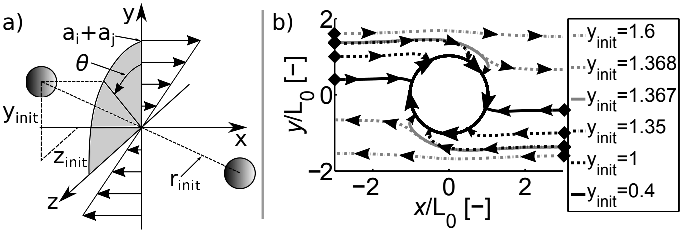

Figure 1.8: a) Illustration of the setup used for numerical investigations of the orthokinetic coagulation behaviour of two particles. The initial positions in the plane spanned by the flow direction and the direction of the velocity gradient of the investigated simple shear flow are rotated around the axis of the flow direction by an angle θ in order to cover all possible configurations. b) Exemplary trajectories of two equal Al_2O_3 particles in water for a dimensionless shear rate $\dot{\gamma} = 1$ and $\theta = 0$, for which the dimensionless initial distance from the x-axis is y_{init} [Bülow et al., 2016].

perpendicular to the flow direction is equal to two times y_{init}. As one can see, the trajectories are influenced by hydrodynamic and DLVO forces and the capture efficiency differs from unity (otherwise only trajectories with $y_{\text{init}} \leq 1$ would lead to coagulation).

Under the assumption that the corresponding Reynolds number is vanishing, electrostatic forces are negligible and the considered suspension is sufficiently dilute, Curtis and Hocking [1970] were among the first to present results from numerical investigations along with values for the capture efficiency for two equal spheres. The capture efficiency has been defined in [Curtis and Hocking, 1970] as normalized integral over all squared initial particle separations perpendicular to the flow direction, at which contact occurs. The sphere of action assumed in [Smoluchowski, 1917] is commonly called collision sphere and the quarter cross-section highlighted in figure 1.8 (a) consequently quarter collision cross-section. Because of the effect of hydrodynamic interactions, the capture cross section σ_c computed by Curtis and Hocking [1970] differs from the collision cross section insofar as it is not circular and its area differs from that of a circle with radius $a_i + a_j$. This is a direct consequence of the effect of hydrodynamics and other interparticle interactions.

Van de Ven, Mason et al. published a series of works called *The microrheology of colloidal dispersions*, in which they also investigate the formation of doublets (see e.g. [Van de Ven and Mason, 1976, 1977]). They state that the definition of the capture efficiency α_o in [Curtis and Hocking, 1970] would be incorrect because Curtis and Hocking [1970] equated α_o with the ratio of capture cross-section to collision cross-section. Interestingly, 25 years later, in [Vanni and Baldi, 2002], the same definition of the capture efficiency as in [Curtis and Hocking, 1970] is used. This might be a reason for the observed overestimation of 30% when using their formula for α_o, compared to experimentally obtained values for α_o. In order to give a more precise expression for the capture efficiency, Van de Ven and Mason [1977] start from the number of particles passing per second through one quadrant of the capture cross-section between y and $y + \mathrm{d}y$, with y being the direction of the flow velocity gradient. Adapted to polydisperse particle systems, for a simple shear flow with velocity in x-direction, $u_x = \dot{\gamma} y$, the particle flux through the whole capture cross-section is given by the expression

$$J_{ij}^{o} = \left(4\dot{\gamma} L_0^3 \int_0^{\max(\hat{y}_c)} \hat{y}\hat{Z}(\hat{y})\mathrm{d}\hat{y} \right) N_i N_j. \tag{1.2}$$

L_0 is the characteristic length scale of the considered problem, N_i and N_j are the number concentration of particles of the respective size classes, $\max(y_c)$ is the maximum extent of the capture cross-section in positive y-direction, and Z the function describing the boundary of the capture cross-section in z-direction. Variables denoted with $\hat{\cdot}$ are made dimensionless by L_0. From equations (1.1) and (1.2) it follows that

$$\alpha_o = 3\frac{L_0^3}{(a_i + a_j)^3} \int_0^{\max(\hat{y}_c)} \hat{y}\hat{Z}(\hat{y})\mathrm{d}\hat{y}. \tag{1.3}$$

For equally sized spherical particles with $L_0 = a_i = a_j$, expression (1.3) equals the expression given in [Van de Ven and Mason, 1977]. From our numerical results we found that Z not only depends on y, but also on the radius ratio λ, usually and in this work always defined as ratio of the radius of the small particle to the radius of the large particle. Furthermore, Z depends on the shear rate $\dot{\gamma}$. This can also be seen in [Van de Ven and Mason, 1977]. The apparent dependency solely on y results from the fact that the capture cross-section is obtained from experiments, where it is given as curve through pairs of coordinates (y_c, z_c).

Apart from the already introduced measures, Adler [1981] introduced a so-called global capture efficiency, which can be defined by

$$\alpha_{o,g} = \frac{(a_i + a_j)^3}{8L_0^3}\alpha_o, \tag{1.4}$$

and equals for $L_0 = a_i = a_j$ the capture efficiency α_o. From equations (1.3) and (1.4) we get

$$\alpha_{o,g} = \frac{3}{8}\int_0^{\max(\hat{y}_c)} \hat{y}\hat{Z}(\hat{y})\mathrm{d}\hat{y}. \tag{1.5}$$

Adler [1981] states that, contrary to α_o, the global capture efficiency accounts for the influence of the radius ratio λ fully (Z is indirectly dependent on λ). In fact both capture efficiencies, α_o as well as $\alpha_{o,g}$, have their justification.
If one wants to correct the Smoluchowski capture frequency $\beta_{ij,\mathrm{S}}$ in equation (1.1), the capture efficiency α_o given by expression (1.3) is more convenient. $\alpha_{o,g}$, obtained from equation (1.5), first needs to be converted to the capture efficiency α_o by means of equation (1.4) and can then be plugged into equation for the capture frequency (1.1). After that, the binary orthokinetic collision rate J^o_{ij}, used in population balance equation (PBE) modeling can be computed through

$$J^o_{ij} = \beta_{ij} N_i N_j, \tag{1.6}$$

for all of the size classes i, j among the N_f size fractions in the system.
On the other hand, the global capture efficiency $\alpha_{o,g}$ is more handy when determining the more probable type of coagulation — homo- or hetero-coagulation. For this, recall equation (1.1) for the capture frequency β_{ij}. Inserting the global capture efficiency by means of the discussed relations yields

$$\beta_{ij} = \alpha_{o,g}\frac{4}{3}\dot{\gamma}(2L_0)^3. \tag{1.7}$$

In case of a monodisperse system, i.e. for $L_0 = a_i = a_j$ for all particle size fractions in the system, this equals expression (1.1). The radius ratio λ for which the ratio β_{ij}/β_{kl}, with $i, j, k, l \in [1, N_f]$, is maximized, has the highest likelihood to coagulate. From equation (1.7) one sees directly that the λ with the highest $\alpha_{o,g}$ yields the highest value for β_{ij}/β_{kl}, as other terms do not depend on λ. This makes the global capture efficiency $\alpha_{o,g}$

a more transparent measure for an analysis of the coagulation probability in polydisperse systems. $\alpha_{o,g}$ includes all radius-dependent terms, while α_o needs to be scaled first. For an exemplarily comparison we refer to chapter 5 or to [Bülow et al., 2016].

1.3.2 Perikinetic coagulation

In case of perikinetic coagulation, the general procedure in deriving a mathematical model is similar and also based on the theory by Smoluchowski [1917]. The main difference is the fact that the capture efficiency is determined differently, namely in most cases by solving the diffusion equation for dispersed Brownian particles. With the resulting perikinetic capture efficiency α_p, one gets

$$J_{ij}^{p} = \beta_{ij}^{p} N_i N_j = \alpha_p \frac{2k_{\mathrm{B}}T}{3\mu_f}(a_i + a_j)\left(\frac{1}{a_i} + \frac{1}{a_j}\right) N_i N_j, \tag{1.8}$$

with the Boltzmann constant k_{B}, the temperature T and the dynamic viscosity μ_f of the fluid. Possible expressions for α_p are discussed in [Feke and Schowalter, 1983, 1985]. An approach based on an investigation of particle trajectories obtained from numerical experiments (Stokesian Dynamics simulations), as it is usually done for orthokinetic coagulation, is presented in [Nazockdast and Morris, 2013] for solid volume fractions $\phi \geq 0.2$ considering Brownian motion, but neglecting DLVO interactions. Mohammadi et al. [2015] follow a similar path, using Brownian Dynamics simulations. In our opinion, because of the stochastic nature of Brownian motion, a continuum-mechanics-like approach or an approach considering the motion of many particles at once is much more promising and less tedious. Notable is that in [Mohammadi et al., 2015] the authors investigate perikinetic as well as orthokinetic coagulation and also investigate the coagulation behaviour of Janus particles.

1.3.3 Remarks on coagulation

Sometimes coagulation due to differential settling (also known as gravitational coagulation [Qiao et al., 1998]) is considered a third type of coagulation and is investigated separately [Davis, 1984, Han and Lawler, 1991]. In technical applications all types of coagulation are superimposed to varying parts and influence each other. Coagulation can lead to settling of otherwise non-settling particles. Differential settling and resulting hydrodynamic clustering then can promote further coagulation, which in turn affects the distribution of settling rates [Farley and Morel, 1986]. Brownian motion can increase coagulation rates for Brownian particles in a quiescent fluid, which can lead to the described process. During settling on the other hand, Brownian motion can reduce the coagulation rate [Qiao et al., 1998]. Low-shear mixing can promote coagulation, as it increases the probability of particle interception. Rapid mixing (high shear rates) also increases the probability of particle interception, but it reduces the capture efficiency [Bülow et al., 2016]. [Feke and Schowalter, 1983] and a few related works address a combination of shear- and Brownian motion induced coagulation. Based on their results, Feke and Schowalter [1983] state that Brownian motion can act to increase or decrease the coagulation rate. Such a statement is not satisfying from an engineering point of view, yet it shows that a derivation of simple rules is not straightforward when more than one influencing effect is investigated simultaneously. From test simulations we found that the effect of Brownian motion on shear-induced coagulation depends strongly on the solid volume fraction of particles. This parameter is usually not considered by classical two-particle approaches leading to a coagulation (capture) efficiency. Yet, as we did not pursue these studies, we cannot give any more quantifiable statement here.

The example of superimposed Brownian motion and shear shows that so far used methods might not be optimal approaches for an investigation in superimposed influences on coagulation. In chapter 5 and in [Bülow et al., 2016] we present first results obtained with a statistical method, with which the problem-dependent superposition of the different effects can be accounted for in the determination of a collision rate. This can yield e.g. a better applicability of PBE based methods to solid-liquid separation problems.

1.4 Prediction of settling rates

The knowledge of settling rates of particles in suspensions is at least as important as a reliable prediction of the destabilization behaviour. Settling is the basic mechanism in unit operations such as separation or fluidization, but it can also play a role in suspension transport, filtration or mixing. Therefore, substantial effort has been made in finding ways to describe settling rates of particles in suspensions by means of universally applicable formulas. These are denoted as hindered settling functions, as an increase in the solid volume fraction usually hinders particles from settling. During derivation most works identify the mean settling velocity of the particles in a suspension with the ensemble average of particle velocities. Therefore we denote both expressions by $\langle U \rangle$.

1.4.1 Infinite dilution

The first to be named and most applied formula is the velocity of a single sphere settling at vanishing Reynolds number in an unbounded quiescent Newtonian fluid. The so-called Stokes velocity results from an equilibrium of an applied centrifugal force and the hydrodynamic drag force, which is also known as Stokes law,

$$U_0 = \frac{2(\rho_\mathrm{p} - \rho_\mathrm{f}) a^2 c_\mathrm{g} g}{9\mu_f}. \tag{1.9}$$

Here, ρ_p and ρ_f are the mass density of the particle and the fluid, respectively. a denotes the particle radius, μ_f the dynamic viscosity of the fluid, and $c_\mathrm{g} g$ the magnitude of the centrifugal force. For the latter we have assumed that the magnitude of the centrifugal acceleration $\omega^2 r$ can be expressed by the gravitational constant g scaled by the relative centrifugal force c_g. This approximation yields good estimates if a change in the distance r between particle position and the center of rotation has a negligible effect on the centrifugal force. The assumptions made during derivation of equation (1.9) are strongly limiting, mostly due to neglecting interactions with other particles and boundaries. Nevertheless, the Stokes velocity is widely used, not only to predict the settling of single spheres but also for suspensions at low solid volume concentrations ϕ (e.g. [Farley and Morel, 1986], [Cunha et al., 2008]).

1.4.2 High dilution

An extension to the Stokes velocity including the influence of the solid volume fraction ϕ is the formula for the ensemble average $\langle U \rangle$ of particle velocities,

$$\frac{\langle U \rangle}{U_0} = 1 - 6.55\phi + \mathcal{O}(\phi^2), \tag{1.10}$$

derived by Batchelor [1972]. The assumptions are, besides a Newtonian fluid, that the particle system is comprised of rigid spherical monodisperse particles with random positions, that the mean volume flux across any stationary surface is zero, and that the container geometry does not affect the settling behaviour. Furthermore, it is crucial that $\phi \ll 1$ not only because then the error term in equation (1.10) is small, but because it would otherwise not be $\mathcal{O}(\phi^2)$ [Russel et al., 1992]. Moreover, when neglecting the error term and ϕ is larger than approximately 0.15, $\langle U \rangle / U_0$ becomes negative. A comparison with other results shows that equation (1.10) gives good estimates for $\phi < 0.025$. A detailed derivation can also be found in [Russel et al., 1992]. In [Batchelor and Wen, 1982] the authors give expressions for polydisperse suspensions based on a similar analysis using pairwise additivity of hydrodynamic interactions between particles.

1.4.3 Medium dilution

Equation (1.10) has limited applicability, therefore Brady and Durlofsky [1988] proposed the formula,

$$\frac{\langle U \rangle}{U_0} - \langle u \rangle = 1 + \phi - 0.2\phi^2 - 1.2\phi\,\frac{5 - \phi + 0.5\phi^2}{1 + 2\phi}, \tag{1.11}$$

with the mean of the background fluid flow velocity, $\langle u \rangle$. This expression is theoretically not limited in ϕ and gives good results for values of ϕ up to approximately 0.2 [Brady and Durlofsky, 1988, Cunha et al., 2002].

1.4.4 Medium to high concentrations

Particle settling is closely related to fluidization. In case the investigated process is fluidization, then not the mean settling velocity but the fluidization velocity of the particle bed is sought for. If no bubbling occurs, or hydrodynamic clustering in case of settling, both velocities are related. We show this on a simple example before we discuss the probably most well-known formula for both the settling velocity and the fluidization velocity.

Sangani and Acrivos [1982] give an expression for the hydrodynamic force acting on stationary spheres in an ordered periodic array, which is an extension to the result in [Hasimoto, 1959] for higher values of ϕ. The expression is given by

$$\begin{aligned}\frac{6\pi\mu_f aU}{F^{\mathrm{h}}} =& 1 - 1.7601\phi^{1/3} + \phi - 1.5593\phi^2 + 3.9799\phi^{8/3} \\ & - 3.0734\phi^{10/3} + \mathcal{O}(\phi^{11/3}),\end{aligned} \tag{1.12}$$

with the magnitude F^{h} of the hydrodynamic force acting on a sphere of radius a. Together with the choice of the characteristic length scale $L_0 = a$ and an expansion of the left-hand side by the characteristic velocity U_0, rearrangement leads to an expression for the ratio U/U_0. As the spheres are not moving relative to another, the value U, which is in [Sangani and Acrivos, 1982] the magnitude of the flow velocity, equals the mean particle velocity $\langle U \rangle$. Note that the velocity obtained from this method is a terminal velocity, as only at steady state F^{h} can be expressed by other acting forces, such as centrifugal forces or gravity.

Richardson and Zaki [1997] made extensive use of dimensional analysis and experiments to find their famous equation,

$$\frac{\langle U \rangle}{U_\mathrm{i}} = (1-\phi)^n, \tag{1.13}$$

which yields not only the mean settling velocity in a suspension but also the fluidization velocity. According to [Richardson and Zaki, 1997], the exponent n depends only on the ratio a/a_tube, with the radius a_tube of the tube in which the particles are suspended. In case the Reynolds number is small, $\mathrm{Re} \leq 0.2$, $n = 4.65 + 19.5a/a_\mathrm{tube}$.
It is notable that in [Richardson and Zaki, 1997] the authors give a relation for $\langle U \rangle / U_\mathrm{i}$ instead of $\langle U \rangle / U_0$. A distinction is made between U_0, the terminal particle velocity in an infinite fluid, and U_i, the particle velocity at infinite dilution. While Richardson and Zaki [1997] state that for their experiments on settling $U_\mathrm{i} = U_0$, the two velocites do not equal in their experiments on fluidization. As the exponent n, the ratio U_i/U_0 depends on the ratio a/a_tube. This difference is attributed to the velocity gradient resulting from fluid friction at the walls and a correction is given.
The particles in [Richardson and Zaki, 1997] were uniform and larger than 100μm, furthermore the velocity of the sludge line at the interface between suspended particles and supernatant has been tracked. There are works investigating non-uniformity (e.g. [Funamizu and Takakuwa, 1995]) and the fluidization of nanoparticles (e.g. [Valverde and Castellanos, 2006]). As can be expected, the latter tend to coagulate and form agglomerates, which grow during fluidization but are broken down by higher flow rates. In [Ham and Homsy, 1988] results from experiments are presented, in which single particles in a settling suspension instead of the sludge line are tracked. All named works observed mean settling velocites which can be expressed by functions of the same type as equation (1.13). Based on their experiments, Ham and Homsy [1988] give a relation of $\langle U \rangle / U_0 = 1 - 4\phi + 8\phi^2$, which differs from equation (1.10) substantially, as the authors note. What they might have overlooked is that equation (1.10) is only valid for $\phi \ll 1$. The values of ϕ in [Ham and Homsy, 1988] ranged from 0.025 to 0.1, which supports our earlier statement, that equation (1.10) should only be used when $\phi < 0.025$. The experiments in [Richardson and Zaki, 1997] leading to equation (1.13) have been performed for $0.1 < \phi < 0.45$.

Using a theoretical approach, equation (1.13) is derived in [Russel et al., 1992] without limitation to ϕ.

1.4.5 Settling rates considering heterogeneity and remarks

Davis and Gecol [1994] proposed a combination of equation (1.13) and the theory in [Batchelor and Wen, 1982], which seems to give good results for polydisperse suspensions. As the other presented formulas in section 1.4, this proposed hindered settling function does not apply for suspensions in which lateral segregation occurs so that vertical fingers and blobs form [Davis and Gecol, 1994].

In [Cuthbertson et al., 2008] hindered settling of heterogeneous particle systems, namely a mixture of cohesive particles (mud flocs) and non-cohesive particles (sand grains), is investigated and expressions for hindered settling functions considering the properties of flocs are discussed. So far most research reports a monotonous dependency of the settling velocity on ϕ. But there is evidence that there is a maximum at low to medium concentrations, which which can be attributed to hydrodynamic clustering [Bhatty, 1986, Kuusela et al., 2003]. We could name many more works in this field, which shows that settling rates are still a relevant field of research. Most current works come from the fields of research in fluidized beds and sediment transport in river basins or coastal areas. With the growing possibilities of numerical simulations, and especially with the tremendously increasing popularity of CFD-DEM simulations (cf. subsection 1.5.2), a rapid determination of the drag force on suspended particles, which as we can see from equation (1.12) is closely related to the settling- or fluidization velocity, is becoming important. Worth mentioning here in particular is the work by Di Felice [1994], in which a so-called voidage function and its application to the computation of the drag coefficient is presented. This function resembles remarkably equation (1.13), whereby the exponent depends only on the Reynolds number.

In chapter 4 we present results from our investigations in the settling behaviour of polydisperse particle clouds. As already adressed by the examples [Davis and Gecol, 1994, Bhatty, 1986, Kuusela et al., 2003], inhomogeneities in the spatial distribution of particles in a suspension can lead to hydrodynamic clustering. We show that particle clouds, sometimes also called blobs, can reach much higher settling velocities than equation (1.9) or hindered settling functions predict, and we provide an analytic equation for the computation of the settling velocity of polydisperse clouds. The investigation of isolated particle clouds is a rather academic example. Yet it shows that hydrodynamic clusters can be quite stable and that the occurence of many clusters in a suspension can increase the mean settling velocity substantially. Furthermore, it shows that size segregation is strongly reduced when hydrodynamic clusters are formed, even when the overall solid volume fraction in the suspension might be low.

1.5 The numerical simulation of suspensions

Even though computational power is still increasing and code parallelization allows for the numerical simulation of increasingly large problems, industrial scale problems met in chemical and process engineering are difficult to simulate and often flow-sheet simulations, heavily relying on background data, are used for proof of concept calculations, design decisions and investigations in effects occuring during unit operations [Nikoo and Mahinpey, 2008, Doherty et al., 2009] as well as for whole plant simulations [Weifeng et al., 2006]. The background data used for such simulations is either produced through experiments, yielding the parameters of the used model equations [Wooley and Putsche, 1996], or it can be generated by means of numerical simulations [Dosta et al., 2014]. For the latter approach, one can use simplified models [Lackner et al., 2008, Dosta et al., 2010] which are usually not universal but give the parameters of interest directly. Because of the fact that in particulate flow problems usually many time- and length scales are covered, another way to obtain the necessary data has become popular recently. This method is called multiscale simulation [Dosta et al., 2014, Deen et al., 2007, Chakrabarty and Cagin, 2010, Van der Hoef et al., 2008, Ge et al., 2011, Lu et al., 2011, Li et al., 2013, Dosta et al., 2013] and uses the fact that problems on different scales are tackled best by different methods. Furthermore, a very detailed simulation of a specific problem has higher numerical costs (time,

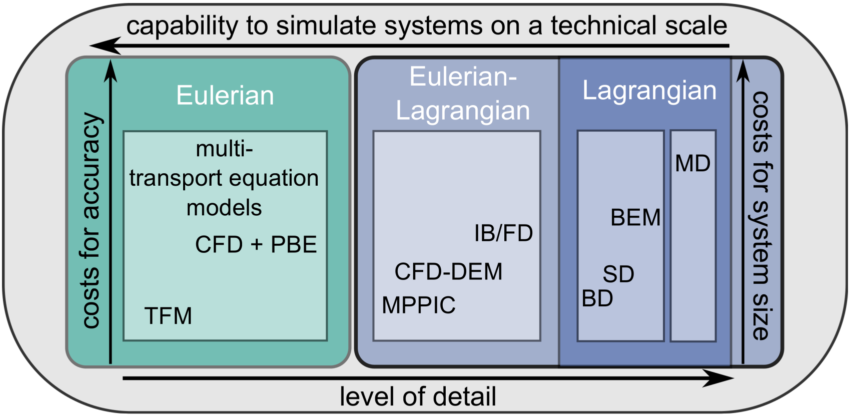

Figure 1.9: Classification of numerical methods for the simulation of particulate flows.

CPU power and memory requirements) than the usage of a more approximate, stronger model-based or less-dimensional method. Such methods often rely heavily on made assumptions and the validity of used models, but they are more capable of simulating systems on a technical scale (cf. figure 1.9).

As depicted exemplarily in figure 1.10, in multiscale simulations there can be two, three or more levels of detail below the technical scale or flow-sheet level. A method used at a certain level gets its initial and boundary conditions passed down from the next higher level method and provides that level with results. The higher level then uses the results of the lower level in order to obtain data for unresolved processes. This procedure is then repeated until the highest level in the multiscale simulation is reached. The higher the level the more modeling is required, for which the lower levels provide data or relations. In the former case the different levels are linked directly and simulations usually run simultaneously, connected by interfaces. Lower levels simply produce data for a current state of the next higher levels. When the lower level simulations or experiments yield relations over a range of parameters, the levels do not have to be coupled as tightly. Lower level simulations can be carried out first, giving the required relations which are then used during the simulation at higher

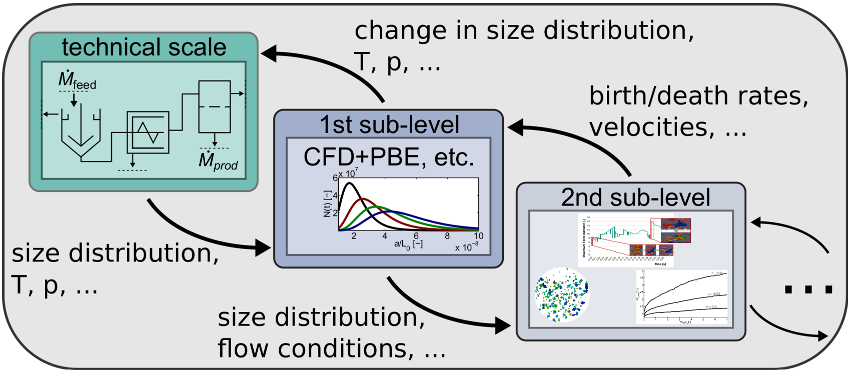

Figure 1.10: Illustration of the idea of multiscale simulations.

levels. Generally but not necessarily always, at the lowest level there is a direct numerical simulation. This means that physical phenomena and effects are resolved as accurately as possible.

There is a large number of methods for the numerical simulation of particulate flow problems. Each of them has its pros and cons and is usually better suited for the solution of a specific type of problem than for others. A complete discussion of all methods would lead beyond the scope of this work. Nevertheless, we outline the main methods and point out similarities between different approaches. In order to give the reader a better understanding of the field, we present the basic approaches for solid-fluid simulations first.

1.5.1 Basic approaches

When the temporal behaviour of suspensions is simulated, the motion of the fluid phase as well as that of the solid phase need to be described appropriately. There are several approaches for the modeling of the two phases. Each phase can either be considered as continuum, or as discrete particulate phase. Furthermore, one can follow the evolution of probability density functions describing the state of the particles a phase consists of. The typical approach is to model the fluid phase as continuum and the solid phase as discrete particle phase. In theory, any combination of the three approaches is possible. To give an overview, we introduce the basic approaches before we proceed to a presentation of the most common used methods in suspension simulation.

Continuum mechanics

Processes met in engineering, and in process engineering in particular, can often be described by balance- or transport equations. These give a mathematical description of the conservation of mass, momentum, energy and other properties in a control volume placed in a phase modeled as continuum. A typically met type is the so-called convection–diffusion equation, which describes the evolution or transport of a quantity c in time t,

$$\frac{\partial}{\partial t} c = -\nabla \cdot (\boldsymbol{u} c) + \nabla \cdot (\lambda \nabla c) + f. \tag{1.14}$$

Convective transport depends on the velocity $\boldsymbol{u}$ of the transporting medium. Diffusion is dependent on the diffusion coefficient λ of the quantity c in the surrounding medium. f accounts for sources or sinks, which can be chemical reactions or another cause increasing or reducing c. An intuitive example for c is the concentration of a solute in a solution. The variable c can be substituted by any transported quantity defined at a point $\boldsymbol{x}$ and time t, for example by the temperature T.

An example for equation (1.14) occuring in particulate flow problems is the Fokker–Planck equation. In this case c is substituted by the probability density function $p(\boldsymbol{x}, \boldsymbol{U}, t)$ giving the probability that at a point $\boldsymbol{x}$ and at a time t a particle with velocity $\boldsymbol{U}$ is found. In the context of Brownian particles, i.e. particles for which Brownian motion is dominant, the Fokker–Planck equation is also called Smoluchowski equation [Dhont, 1996]. The list of parameters for p can be extended for other attributes, such as particle volume, etc.

In the Boltzmann limit [Cercignani, 1988], the Boltzmann equation can be derived from the Liouville equation, which is equivalent to the Fokker–Planck equation without diffusion term [Dhont, 1996, Cercignani, 1988].

Even though the asumptions made during derivation seem to be strongly limiting (e.g. molecules are modeled as hard spheres with range of the interaction potential tending to zero), fluids can be simulated by means of a solution of the discretized (lattice) Boltzmann equation [Chen and Doolen, 1998]. The order of convergence of the corresponding lattice Boltzmann method (LBM) is in general first order in time and second order in space [Tölke and Krafczyk, 2008].
A solution of equation (1.14) for the case of transported quantities requires knowledge of the velocity $\boldsymbol{u}$ of the transporting medium. In case of particles suspended in a fluid, $\boldsymbol{u}$ equals the fluid velocity. Substitution of c in equation (1.14) by the (vectorial) momentum of a fluid results in the Navier–Stokes equations, substitution by the mass of the fluid yields the continuity equation (both with respect to infinitesimally small control volumes). In the field of computational fluid dynamics (CFD) these equations are solved numerically in order to describe a given flow problem. This means that the Navier–Stokes equations,

$$\frac{\partial}{\partial t}\boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho_{\mathrm{f}}}\nabla p + \frac{\mu_f}{\rho_{\mathrm{f}}}\Delta \boldsymbol{u} + \frac{1}{\rho_{\mathrm{f}}}\boldsymbol{f}, \tag{1.15}$$

together with the continuity equation,

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1.16}$$

are solved for the fluid velocity $\boldsymbol{u}$ and the pressure p. For the given forms of equations (1.15) and (1.16) it is assumed that the fluid is an incompressible Newtonian fluid with constant mass density ρ_{f} and dynamic viscosity μ_f. The variable $\boldsymbol{f}$ accounts for external body forces such as gravity. If the energy in the system is of interest, equation (1.14) is solved for the energy or the temperature in the system (see e.g. [Oertel et al., 2009, Versteeg and Malalasekera, 2007, Ferziger and Peric, 2012]). Partial differential equations (PDEs) like equations (1.14), (1.15) and (1.16) can be solved numerically by means of the finite element method (FEM) [Turek, 1999], the finite volume method (FVM) [Versteeg and Malalasekera, 2007] or the finite difference method (FDM) [Ferziger and Peric, 2012], to name the most common methods. Numerically, the three methods FEM, FVM and FDM are related and it can be shown that the resulting system of linear equations fundamental to the solution when using a certain FDM is also obtained when using a corresponding FVM or FEM. Furthermore, FVM can be seen as a low order FEM [Eymard et al., 2000]. Unstructured grids, time-dependent grid refinement, etc. can be handled by all three methods with varying efficiency. Basic comparisons of the methods can be found in [Ferziger and Peric, 2012, Eymard et al., 2000]. Currently, but without any particular reason, the FVM is by far the most popular method in CFD in chemical and process engineering.

Discrete particle dynamics

The motion of a particle α in a system of N particles can be described by Newton's second law [Newton, 1729]. This results in Euler's laws of motion [Rao, 2005], which are the well-known translational and angular momentum balance equations,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(m_{p,\alpha}\boldsymbol{U}_\alpha\right) = \sum_l \boldsymbol{F}^l_\alpha, \quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{I}_\alpha\boldsymbol{U}_\alpha\right) = \sum_l \boldsymbol{T}^l_\alpha. \tag{1.17}$$

The variable $\boldsymbol{U}_\alpha$ denotes the velocity of a single particle α, $m_{p,\alpha}$ its mass, $\boldsymbol{I}_\alpha$ its moment of inertia tensor and the sums of $\boldsymbol{F}_\alpha$ and $\boldsymbol{T}_\alpha$ the sums of the acting forces and torques, respectively.

In molecular dynamics (MD), equations (1.17) are applied to simulate the motion of atoms and molecules of a fluid or solid phase (see [Schlick, 2010] for a comprehensive introduction). A combination of several atoms to molecules, where each molecule is usually assumed to be spherical, or the combination of larger particles to groups of particles is called coarse-graining. This technique reduces numerical costs and allows for simulations of processes at larger timescales, as the number of particles for which trajectories and interactions have to be computed, can be reduced immensely [Chakrabarty and Cagin, 2010, Rudd and Broughton, 1998]. On the other hand coarse-graining introduces the need for extra modeling, which can be quite challenging.

An application of equations (1.17) to the simulation of solid particles larger than molecules is the discrete element method (DEM) [Cundall and Strack, 1979], which has very early been applied to the simulation of non-spherical particles [Cundall, 1988] and is currently the most popular method in the field of simulation of particulate flows with a wide variety of applications on industrial scale [Munjiza and Cleary, 2009]. The initial model considering only particle contact [Cundall and Strack, 1979] has been extended by a large number of models for other effects, such as adhesion [Moreno et al., 2003], solid bridges [Antonyuk et al., 2006], and liquid bridges [Muguruma et al., 2000, Hsiau and Yang, 2003, Soulie et al., 2006]. A current trend is clearly the development of fast DEM methods for the simulation of non-spherical particles [Fraige et al., 2008, Ferellec and McDowell, 2010], especially in combination with a coupled solution of equations (1.15) and (1.16) [Zhu et al., 2008, Kruggel-Emden and Oschmann, 2014].

Other methods

Methods arising from discrete particle dynamics generally do not require a background grid. Methods based on continuum mechanics on the other hand usually rely on a numerical grid which is spatially fixed or which rotates with its reference frame. This is a result from the discretization of equations (1.14), (1.15) and (1.16).

An exception from this is e.g. the smoothed particle hydrodynamics (SPH) method, where the Navier–Stokes equations (1.15) and the continuity equation (1.16) describe the motion of Lagrangian particles [Gingold and Monaghan, 1977]. A review on the method is given in [Liu and Liu, 2010]. If one wants to get a brief description of the method together with an application, we refer to [Dalrymple and Rogers, 2006]. The SPH method is suited best for simulations of two-phase flows where the two phases have such a large density difference that the lower-density phase can be neglected, e.g. for the simulation of waves and sprays [Dalrymple and Rogers, 2006, Hoefler et al., 2013].

The method of multi-particle collision dynamics (MPC), also known as stochastic rotation dynamics (SRD) [Gompper et al., 2009] originates from coarse-graining of molecular dynamics [Malevanets and Kapral, 2000], where fluid motion is divided into a streaming step and a collision step, during which the whole interaction of fluid particles takes place. In the collision step, fluid particles are mapped to the cells of a numerical background grid, in which they collide. In order to preserve Galilean invariance (invariance of the simulation result on the positions of the grid nodes), the grid used in SRD is shifted during a simulation [Ihle and Kroll, 2001]. Macroscopic variables such as the fluid velocity $\boldsymbol{u}$ or its density ρ_f are obtained by weighted integrations of the transported variables [Ihle and Kroll, 2003]. In [Malevanets and Kapral, 1999] the connection between the equations solved by SRD and equations (1.15) and (1.16) is shown. The derivation is similar to that showing the connection between the lattice Boltzmann equations and equations (1.15) and (1.16) [Chen and Doolen, 1998].

When referring to CFD in the following, we do not refer to a specific method, but to a direct or indirect numerical solution of equations (1.15), (1.16) or their counterpart for a specific problem.

1.5.2 Simulation methods for suspensions

Methods for the simulation of suspensions are usually based on the solution of a combination of equations (1.15), (1.16) and (1.17). As depicted in figure 1.9, they can be classified by means of the description of the two phases — Eulerian-, Lagrangian- or Eulerian–Lagrangian. Therefore, we now give the mainly used methods grouped by means of the description of the phases.

The two-phase fluid model (TFM) – an Eulerian method

Assuming no effect of particles on the fluid and no particle-particle interactions, the motion of inertialess particles in a suspension can be described by a combination of equations (1.14), (1.15), (1.16), where equation (1.14) describes the transport of the particle concentration [Ilina et al., 2008]. As is easy to see, these assumptions are strongly limiting the range of application.

The simulation of suspensions at scales close to industrial scale and at high solid volume fractions, considering fluid-particle, particle-particle and particle-fluid interactions, is still a very challenging task — the fluid-solid as well as the solid-solid interaction has to be captured accurately while the number of particles is extremely high. Therefore, the two-phase fluid model (TFM) [Anderson and Jackson, 1967, Drew, 1982, Enwald et al., 1996] finds broad application both in the investigation of suspension dynamics in fluidization [Wang et al., 2009, 2010] and in settling and sediment consolidation [Auzerais et al., 1988, Gustavsson and Oppelstrup, 2000].

In the TFM, the fluid and the solid phase are treated as continua, for which mass, momentum and energy conservation have to hold [Wang et al., 2009, Gustavsson and Oppelstrup, 2000]. Both phases interpenetrate each other, where the solid volume fraction ϕ determines the amount of particles in a computational grid cell. Momentum exchange of the two phases is considered by additional terms in the two Navier–Stokes equations [Shuai et al., 2012]. Critical points are the correct modeling of the usually ϕ-dependent viscosities, the pressure in the solid phase and, if considered, contributions to the energy equation resulting from particle interactions.

When this model is applied, the assumptions for the used models need to be kept in mind, as non-valid assumptions can lead to wrong predictions [Ge et al., 2011, Wang et al., 2009].

Population balance equations – another Eulerian method

The population balance equations (PBE) formulated by Smoluchowski [1917] gave rise to the PBE method, also known as population balance modeling (PBM). As the TFM, the PBM can be applied to the numerical simulation on industrial scale. If the kinetics (birth-, coagulation-, breakup- and death-rate of particles) in a considered process unit can be described by overall efficiencies (cf. section 1.3), and the dynamics can be simplified [Camp and Stein, 1943], an overall PBE (mass conservation) suffices to describe the change of a particle size fraction in time. The particle size distribution can be approximated by discrete size intervals, for which coupled PBEs are solved. In technical applications especially the shear gradient of the flow varies spatially, such that this technique is often not applicable [Kramer and Clark, 2000]. In this case local flow conditions can be obtained by CFD, which results in the widely used CFD-PBE method [Li et al., 2013, Cheng et al., 2012]. The number of coupled PBEs is very high for a broad size distribution, limiting the applicability of CFD-PBE methods. Because of this, methods such as QMOM (quadrature method of moments) [McGraw, 1997], and the DQMOM (direct QMOM), which accounts for more than one moment [Fan et al., 2004], are currently the most widely used CFD-PBE methods.

Lagrangian methods

While Eulerian methods are capable of simulating large systems and dense suspensions, they rely heavily on accurate modeling. Lagrangian methods usually treat the effect of the fluid implicitly, which results in the fact that these models do not need a numerical background grid, and the variables describing the fluid motion do not have to be computed in order to obtain hydrodynamic interactions. Furthermore, Lagrangian methods are capable of capturing interparticle interactions by means of more generally valid models (e.g. such describing the non-hydrodynamic interactions presented in subsection 1.2.2). The resulting forces or torques can simply be added to other acting forces and torques in the equations of motion (1.17). As illustrated in figure 1.9, the disadvantage of this detailed description of the particle motion is the fact that the numerical costs increase with increasing system size, i.e. number of particles. In chapter 4 we show that techniques exist, which can reduce the general costs of $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log(N))$ or in special cases to $\mathcal{O}(N)$, where N is the number of simulated particles, but the dependency on the number of particles persists. Another restriction for Lagrangian methods is the problem of an inclusion of external geometries (non-flat container walls, mixing elements, etc.) in a simulation. This is the downside of the fact that hydrodynamic interactions can be obtained purely from the knowledge of particle positions and geometries.

A typical example for a Lagrangian method is the already introduced MD method, in which all atoms or molecules are resolved [Schlick, 2010]. Other methods solve the Langevin equation [Ichiki and Hayakawa, 1995], which is formally equivalent to equation (1.17) and describes the motion of particles in a suspension. Mathematically, the Langevin equation differs from the general expression (1.17), because of the stochastic nature of the occurring Brownian force/torque term, which makes it a stochastic differential equation instead of an ordinary differential equation. Hydrodynamic interactions among particles are obtained from a mathematical treatment of the integral expression for the fluid velocity of a Stokes flow with present particles (cf. for example [Kim and Karrila, 2013]). The main methods are the boundary element method (BEM) [Nishimura, 2002, Wrobel, 2002, Pozrikidis, 2007], the Stokesian Dynamics (SD) method [Brady and Bossis, 1988, Durlofsky et al., 1987, Sierou and Brady, 2002] and the Brownian Dynamics (BD) method [Ermak and McCammon, 1978, Heyes and Melrose, 1993, Hütter, 2000]. The level of detail with respect to the treatment of hydrodynamic interactions is decreasing with increasing importance of Brownian motion from BEM over SD to BD.

Suspensions [Pozrikidis, 2007, Sekine, 2000] are only a minor field of application for BEM, which is a general method to solve boundary integral equations [Nishimura, 2002] occuring in heat transfer, accoustics, electrochemistry and fluid mechanics [Wrobel, 2002]. BD as well as SD have been developed for the simulation of suspensions, where both have their main field of application in the investigation of structure, diffusion and rheology of suspensions [Hsieh et al., 2006, Foss and Brady, 2000, Noguchi and Takasu, 2001]. We do not go into further details of the methods here, because the mainly used method in this work is the SD method, which is described roughly in all following chapters and more detailed in chapters 2 and 4, where we present our works [Bülow et al., 2014, 2016] which deal with numerical improvements of the method.

A good review comparing particle-based approaches for the numerical simulation of colloidal suspensions is given in [Bolintineanu et al., 2014]. The authors refer to the simulation platform LAMMPS [Plimpton, 1995], which originates from MD simulation but now includes a large number of simulation algorithms for the simulation of colloids [Bolintineanu et al., 2014].

Eulerian–Lagrangian methods

Eulerian methods are capable of simulating large systems, while Lagrangian methods are usually more accurate and can be extended for models of additional effects easily. Therefore it seems natural to combine methods from both classes in order to apply the ease of modeling to larger systems. Eulerian–Lagrangian methods are currently extremely popular, which is why we present the three mainly used methods. All these methods have in common that the fluid motion in the whole simulation domain is described by a continuum mechanics approach, i.e. the underlying equations, e.g. equations (1.15) and (1.16), are solved on an Eulerian grid. Particle motion is usually described by means of equations (1.17) for a moving Lagrangian grid, which is defined by the particles. All methods have in common that coupling between the different phases is done via an interpolation between the two grids, as is exemplarily depicted in figure 1.11. The methods differ in the way particles are represented and in the coupling of the solved equations.

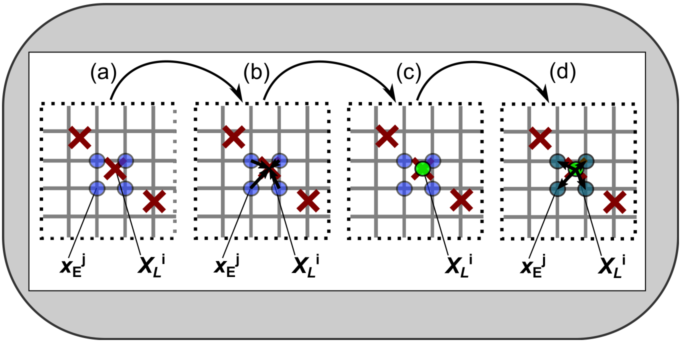

Figure 1.11: General idea of Eulerian–Lagrangian methods: Computation of fluid properties at the Eulerian grid nodes x_{E} (a), interpolation to Lagrangian grid with nodes X_{L} (b), computation of Lagrangian variables (c), interpolation of the effect of particles from Lagrangian grid nodes to Eulerian grid (d).

Immersed boundary/ficticious domain methods

In Immersed boundary (IB) methods as well as ficticious domain (FD) methods the Eulerian background grid is so fine that the particle geometry is more or less resolved (for spherical particles by typically between 10 and 30 grid cells per diameter). Rigid body motion is imposed either at the particle boundaries (IB) or in the whole particle domain (FD) by means of a ficticious body force or by modifying the fluid stress tensor.

There are several schemes differing in the way this rigidity constraint is imposed on the fluid at the respective Eulerian grid nodes [Glowinski et al., 2001, Peskin, 2002, Mittal and Iaccarino, 2005, Uhlmann, 2005, Sharma and Patankar, 2005, Münster et al., 2012]. In most of them a direct forcing fractional stepping scheme is applied. In this scheme, fluid variables (usually velocities) are computed first. Subsequently, these are interpolated to the Lagrangian grid nodes, where the ficticious force is computed by means of the difference between interpolated fluid velocity and the velocity of the Lagrangian grid nodes. The resulting body force is then interpolated to the Eulerian grid, where it accounts for the extra momentum resulting from the relative motion of the particles. Both methods, IB and FD, are not restricted in the shape of the simulated particles [Sharma and Patankar, 2005, Yang and Stern, 2012], can handle deformable bodies [Deen et al., 2009, Shirgaonkar et al., 2009] and are currently used to simulate suspensions of up to several thousand particles [Münster et al., 2012, Uhlmann, 2008]. Originally, these methods have been developed for fluid-structure interaction problems, for which they are still widely used (see e.g. [Borazjani et al., 2008]).

CFD-DEM

The currently by far most popular method for simulations of fluid-particle interactions is CFD-DEM, also known as discrete particle simulation(DPS) [Tsuji et al., 1993]. In this method, a CFD method is used to solve the fluid equations, while the motion of the particles is simulated by means of DEM (cf. subsection 1.5.1). Contrary to IB and FD methods, particles are assumed to be much smaller than the resolution of the Eulerian background grid. This results in the fact that the Lagrangian grid consists only of the particles' centers of mass instead of points describing their shape. The drag force acting on a single particle is assumed to be linearly dependent on the difference between particle velocity and the fluid velocity at the particle's center of mass. The standard relation is a combination of the Ergun equation [Ergun, 1952] and the correlation proposed by Wen and Yu [2014] (see e.g. [Deen et al., 2007, Xu et al., 2012]). As already pointed out in subsection 1.4.5, the determination of drag relations for hydrodynamically interacting particles is a current field of research, especially for polydisperse systems, coagulated and non-spherical particles, as well as wall-bounded systems. Exemplary works considering some of these aspects are [Di Felice, 1994, Chhabra et al., 1999, Kim and Lee, 2014, Tran-Cong et al., 2004, Loth, 2008, Hölzer and Sommerfeld, 2008]. Basic equations of CFD-DEM and a general flow diagram can be found e.g. in [Deen et al., 2007]. Typical numbers of simulated particles range between a few thousand and some hundred thousand [Xu et al., 2012, Fernández and Nirschl, 2013, Chu et al., 2009, Afkhami et al., 2015, Zhang et al., 2008]. Large-scale simulations on CPUs allow for simulations of up to some million particles [Yang et al., 2014], whereas a coupling of a parallel CFD code running on CPUs with a DEM code running on GPUs can yield particle numbers of 25 million [Jajcevic et al., 2013]. A review on applications of CFD-DEM is part of the review [Zhu et al., 2008].

The multiphase particle-in-cell method

Even though large numbers of particles can be simulated with CFD-DEM, realizable particle numbers in simulations for higher solid volume fractions within a timespan of a few days are still comparably low. The multiphase particle-in-cell method (MP-PIC) has been developed for rapid simulations of large-scale multi-phase flows that can handle dilute and dense particulate loading [Andrews and O'Rourke, 1996]. Through coarse graining, simulated particle numbers are much higher than for CFD-DEM while contrary to the TFM, the MP-PIC method considers particle size distributions. The equations describing the fluid motion by means of an Eulerian grid are the same as that in the TFM and CFD-DEM – conservation equations which take momentum coupling between the two phases by means of a drag relation into account. The particle motion is described by a Liouville equation (cf. subsection 1.5.1), where the transported probability density function accounts not only for particle positions and velocities in time, but also for their volume and density [Snider, 2001]. Similar to the SPH method (cf. subsection 1.5.1), the particle variables are defined on a Lagrangian grid which moves with the computational particles, i.e. with the coarse grained particulate phase. Each computational particle represents a group of particles of identical size, velocity and position [Patankar and Joseph, 2001]. Particle collisions are modeled by a particle stress tensor [Andrews and O'Rourke, 1996, Patankar and Joseph, 2001], which was later expanded for a BGK-type collision operator [O'Rourke and Snider, 2010, 2014]. The MP-PIC method relies on a lot of modeling, but when this is done correctly, industrial scale problems can be solved well (e.g. [Snider et al., 2004, Chen et al., 2013]).

1.6 Preview of the following chapters

The goal of this work is the investigation of the destabilization behaviour of suspensions with focus on the numerical simulation. This can theoretically be done by means of any of the simulation methods presented in subsection 1.5.2. Yet, results of Eulerian and Eulerian-Lagrangian methods rely strongly on the modeling of the relation for the momentum coupling between the two phases [Chen et al., 2013]. Accurate drag models are still under development (cf. the part on CFD-DEM in subsection 1.5.2) and so far used drag models usually do not explicitly account for destabilization. Moreover, effects described in section 1.2 can be modeled well at the particle level, which is the level of Lagrangian methods. At this level models such as that from DLVO theory have been tested extensively, such that their validity is largely known. Among the Lagrangian methods, the Stokesian Dynamics (SD) method is one of the most flexible ones. Its major limitation is the restriction to vanishing Reynolds numbers, but not the particle size or the solid volume fraction. As might have become clear in section 1.5, there is a fluid transition between different methods used for the simulation of suspensions. An SD simulation which takes Brownian motion into account, but does not include the standard lubrication scheme [Durlofsky et al., 1987], may easily be called Brownian Dynamics (BD) simulation with extensive modeling of hydrodynamic interactions. On the other hand, a BD simulation for which the full mobility matrix is used, can also be called SD simulation. Furthermore, the Langevin equation, which describes the motion of suspended particles in the SD and BD methods, can also be obtained from the corresponding Fokker–Planck equation [Brady and Bossis, 1988]. Therefore the SD method can be seen as Lagrangian method for the solution of the Fokker–Planck equation for spherical particles (in analogy to the SPH method, which is a Lagrangian method solving the Navier–Stokes equations).

Some values, such as the mean settling velocity or the agglomeration rate of particles in a suspension, can only be predicted well enough if the number of simulated particles is large enough [Bülow et al., 2014]. Therefore we have parallelized the open-source software RYUON [ryu, 2014], which served as basis for our simulation framework. Even though the current code does not resemble RYUON anymore, we still kept the name in honour of Kengo Ichiki, who provided us with the latest version of his code in 2012. Chapter 2 describes the utilized parallelization techniques and shows the strong speedup a parallel SD software can achieve. In this chapter we also present the models for DLVO interactions we use to investigate destabilizing suspensions.

In chapter 4 we introduce an improvement of the parallelized SD method. We show that a combination with the Barnes–Hut method [Barnes and Hut, 1986] can yield additional speedup of a simulation in case of non-homogeneous, clustered suspensions. Roughly speaking, this speedup results from combining particle groups for the computation of hydrodynamic interactions. This grouping is dynamic and is only done if the distance between two groups is large enough for the resulting error to be small. We called this hybrid method scalable SD method, because the criterion for the grouping can be chosen by the user, resulting in a scalable accuracy and speed of the simulation. The method is designed for parallel code execution and differs formally from the original SD method only in the computation of altered entries of the mobility matrix, leaving the theory behind the original method untouched [Bülow et al., 2016]. Also, the required software development effort for an inclusion in an already parallelized SD code is not high, which makes the method in this point more attractive than other fast schemes [Sierou and Brady, 2001, Ichiki, 2002]. This theoretical part is followed by a second part, in which we present results obtained with the developed simulation platform.

In chapter 4, we give insights into the settling behaviour of polydisperse particle clouds, which can also be seen as hydrodynamic clusters. In the considered case, hydrodynamic interactions are dominant, what leads to an interesting behaviour of the settling clouds. Before our work, only polydisperse clouds of up to a few hundred particles have been simulated. With the parallel SD method we were able to investigate the behaviour of clouds comprised of thousands of particles, which leads to new conclusions on the impact of polydispersity [Bülow et al., 2015]. In this chapter, we also show that the Hadamard-Rybczynski equation [Michaelides, 2006] can be used to predict the maximum settling velocity of a polydisperse settling cloud. The resulting equation can be used to estimate the maximum settling velocity in a suspension, e.g. for other simulations [Faletra et al., 2015].

Besides hydrodynamic clustering, destabilization due to coagulation is a major cause for increased settling rates. Population balance equations coupled to CFD are a valuable tool in describing destabilization behaviour in suspensions on a large scale, i.e. on technical or industrial scale [Cheng et al., 2012]. As stated earlier in this work (cf. subsections 1.3.3 and 1.5.2), this method relies strongly on the kernels used to describe changes in the particle size distribution. We used the developed simulation framework to obtain an analytical expression for a polydisperse orthokinetic capture efficiency. To our knowledge the relation presented in chapter 5 is the first relation, which takes the particle radius ratio fully into account. Nevertheless, the procedure involved is tedious, time-consuming, and it does not take other effects such as superimposed settling into account. Therefore we developed a cluster detection postprocessing, which gets the output of simulations of multi-particle systems as input and delivers all necessary data on the formed clusters. This data includes the cluster settling speed, the radius of gyration and the fractal dimension of all clusters over time. Among the data is also the number of primary particles per cluster over time. This data can be used to determine statistical cluster growth rates when effects such as shear and settling are superimposed. Furthermore, this way the influence of other particles is considered. In standard simulations yielding the capture efficiency, only two particles are considered. This can give wrong approximations to the capture efficiency in case of dense suspensions. For a further discussion we refer to chapter 5.

The next chapters include published results and such submitted for publication.

Chapter 2

A parallel implementation of the Stokesian Dynamics method applied to the simulation of colloidal suspensions

Abstract
We present a strategy for the numerical simulation of polydisperse colloidal suspensions on supercomputers using distributed memory. In order to simulate large colloidal systems we have parallelized the Stokesian Dynamics method and combined it with DLVO theory. Through an efficient parallelization using the message passing system MPI we are able to reduce the computational costs of the problem from $\mathcal{O}(N^2)$ to $\mathcal{O}(Pn_{p,\max}^2)$, where N is the number of particles in the system, P the number of used processes and $n_{p,\max} = \lceil \frac{N}{P} \rceil$, respectively. This allows the simulation of large colloidal systems, as the computational time now not only depends on the number of particles, but also on the number of cores, through which the CPU time can be reduced drastically.

This chapter follows our work of the same title published in the journal Computers&Fluids.

2.1 Introduction

The numerical simulation of colloidal suspensions, i.e. particles of diameter between 10nm and 10μm suspended in a fluid, is a challenging task. This is not only due to Brownian motion, which can play an important role at such length scales [Phung et al., 1996]. The interplay of other acting forces such as the London–van–der–Waals force and electrostatic repulsion often creates interparticle potentials which make the numerical treatment of such systems complex. Apart from these forces, hydrodynamic interactions play a very important role and cannot be neglected. For example in solid-liquid separation, agglomeration and subsequent settling of colloids is used to separate a solid product from the continuous phase. This is just one of the many examples in the fields of chemical, biological and environmental engineering. Due to the fact that physical experiments are often expensive or simply not possible, the numerical investigation of the suspension behaviour is a good alternative. The Stokesian Dynamics (SD) method originally developed by Brady and Bossis [Brady and Bossis, 1988] is an ideal method when it comes to the investigation of colloidal systems. The main advantage of this method is that an explicit computation of the fluid velocity is not necessary. Hydrodynamic interactions among particles are incorporated implicitly by a matrix-vector product, which results from the integral representation of the fluid velocity of a particulate Stokes flow [Durlofsky et al., 1987]. The SD method has successfully been applied to investigations on the effects of Brownian motion [Foss and Brady, 2000, Phung et al., 1996, Banchio and Brady, 2003, Banchio et al., 2006] and colloidal systems in general [Harshe et al., 2010, Harshe and Lattuada, 2011, Jones, 2001, Seto et al., 2012, Wagner and Brady, 2009]. It can be used to investigate microscopic [Cunha et al., 2002] or macroscopic properties of a suspension [Chang and Powell, 1994], giving rise to a wide spectrum of applications. Due to its efficiency when dealing with concentrated suspensions, the SD method has been applied successfully to numerical investigations in rheology, ordering and microstructure of colloidal and non-colloidal suspensions [Nazockdast and Morris, 2013, Sierou and Brady, 2002, Xu et al., 2013]. Studies in this field could give a better prediction of the macroscopic suspension behaviour if more than a few hundred particles are simulated. But even though the SD method and extensions to it [Sierou and Brady, 2001, Ichiki, 2002, Durlofsky and

Brady, 1989] are well established and the rapid evolution in computing power weakens the limitations of numerical simulations, apart from the work of Kopp and Höfling [Kopp and Höfling, 2012], the SD method does not seem to have been parallelized yet. Kopp and Höfling describe the parallelization of the simplest version of the SD method, the F-version, for GPU systems. The popularity of GPU systems is increasing but their major drawbacks persist. Their limited memory and SIMD architecture restricts their scope of application to specific problems. MIMD systems with distributed memory are far more common and have a wider range of application. In this work we present our implementation of the SD method applied to the simulation of colloidal suspensions for computer clusters using distributed memory. Our implementation is based on the message-passing system MPI mpi [2013], but the presented strategies can be transferred to any message-passing system or language for distributed memory machines. To clarify the terminology we give a short introduction to the SD method and we show how we model interparticle interactions of non-hydrodynamic nature. Different parallelization strategies have to be applied when it comes to the computation of hydrodynamic or non-hydrodynamic interactions. We present these strategies and show on examples how efficient a parallel SD method can be.

2.2 Methods

2.2.1 Stokesian Dynamics

We consider the motion of colloids in a fluid, thus the Stokes equations usually suffice to describe the fluid motion. The translational motion of a rigid colloidal particle α suspended in a fluid can be described by the Langevin equation [Brady and Bossis, 1988, Ichiki and Hayakawa, 1995],

$$m_\alpha \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_\alpha = \boldsymbol{F}^h_\alpha + \boldsymbol{F}^i_\alpha + \boldsymbol{F}^b_\alpha + \boldsymbol{F}^{ext}_\alpha. \tag{2.1}$$

For the angular momentum we have

$$\boldsymbol{I}_\alpha \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega_\alpha} = \boldsymbol{T}^{\mathrm{h}}_\alpha + \boldsymbol{T}^{i}_\alpha + \boldsymbol{T}^{b}_\alpha. \tag{2.2}$$

Here m_α is the mass of a particle, $\boldsymbol{U}_\alpha$ its translational velocity, $\boldsymbol{F}^h_\alpha$ the hydrodynamic force acting on the particle, $\boldsymbol{F}^i_\alpha$ interparticle forces, $\boldsymbol{F}^b_\alpha$ forces arising from Brownian motion and $\boldsymbol{F}^{ext}_\alpha$ external forces such as gravity. $\boldsymbol{I}_\alpha$ denotes the moment of inertia tensor of a particle, $\boldsymbol{\omega_\alpha}$ its angular velocity and $\boldsymbol{T}^{\mathrm{h}}_\alpha$ the hydrodynamic torque acting on the particle. $\boldsymbol{T}^i_\alpha$ and $\boldsymbol{T}^b_\alpha$ denote the torque arising from interparticle forces and Brownian forces, respectively. Let the background flow, i.e. the fluid velocity in the absence of any disturbing particles, at position $\boldsymbol{x}$ in the fluid domain

$$\boldsymbol{u}^\infty(\boldsymbol{x}) = \boldsymbol{U}^\infty + \boldsymbol{\Omega}^\infty \times \boldsymbol{x} + \boldsymbol{E}^\infty x,$$

where $\boldsymbol{U}^\infty$ and $\boldsymbol{\Omega}^\infty$ are the translational and angular velocity of the undisturbed Stokes flow and $\boldsymbol{E}^\infty$ the respective rate of strain. Then there is an integral expression for the fluid velocity with present particles [Durlofsky et al., 1987]. Through a truncated multipole expansion a linear relation between particle velocities $\boldsymbol{U}_\alpha$, $\boldsymbol{\omega_\alpha}$, $\alpha = 1, \ldots, N$, on the one hand, and hydrodynamic forces, hydrodynamic torques, stresslets $\boldsymbol{S}^h_\alpha$ and possibly considered higher moments on the other hand can be found [Durlofsky et al., 1987]. Considering the first two moments in the expansion yields the system of linear equations,

$$\begin{pmatrix} \boldsymbol{U}_1 - \boldsymbol{U}^\infty \\ \vdots \\ \boldsymbol{U}_N - \boldsymbol{U}^\infty \\ \boldsymbol{\omega}_1 - \boldsymbol{\Omega}^\infty \\ \vdots \\ \boldsymbol{\omega}_N - \boldsymbol{\Omega}^\infty \\ -\boldsymbol{E}^\infty \\ \vdots \\ -\boldsymbol{E}^\infty \end{pmatrix} = -\boldsymbol{\mathcal{M}}^\infty \begin{pmatrix} \boldsymbol{F}^{\mathrm{h}}_1 \\ \vdots \\ \boldsymbol{F}^{\mathrm{h}}_N \\ \boldsymbol{T}^{\mathrm{h}}_1 \\ \vdots \\ \boldsymbol{T}^{\mathrm{h}}_N \\ \boldsymbol{S}^{\mathrm{h}}_1 \\ \vdots \\ \boldsymbol{S}^{\mathrm{h}}_N \end{pmatrix}. \tag{2.3}$$

This formulation is valid for rigid particles, where the rate of strain of all particles can be assumed to be zero, i.e. $\boldsymbol{E}_\alpha = \boldsymbol{0}$ for all $\alpha \in [1, N]$. The so-called *grand mobility matrix* $\boldsymbol{\mathcal{M}}^\infty \in \mathbb{R}^{11N \times 11N}$ can be decomposed into sub-matrices,

$$\boldsymbol{\mathcal{M}}^\infty = \begin{pmatrix} \boldsymbol{M}_{UF} & \boldsymbol{M}_{UT} & \boldsymbol{M}_{US} \\ \boldsymbol{M}_{\omega F} & \boldsymbol{M}_{\omega T} & \boldsymbol{M}_{\omega S} \\ \boldsymbol{M}_{EF} & \boldsymbol{M}_{ET} & \boldsymbol{M}_{ES} \end{pmatrix}, \tag{2.4}$$

where e.g. the subscript of $\boldsymbol{M}_{UF} \in \mathbb{R}^{3N \times 3N}$ indicates that this sub-matrix links the hydrodynamic force on all N particles to their translational velocity. The tensors $\boldsymbol{E}^\infty$ and $\boldsymbol{S}_\alpha^{\mathrm{h}}$ are actually matrices in $\mathbb{R}^{3\times 3}$. But due to the symmetry of both tensors, together with the assumption that they have trace zero, they can be represented as vectors of length 5, so we have a dimension matching in equation (2.3). $\boldsymbol{S}_\alpha^h$ does not have to be traceless. Works dealing with particle pressure, which is defined through the trace of $\boldsymbol{S}_\alpha^h$ are e.g. [Jeffrey et al., 1993, Brady, 1993, Yurkovetsky and Morris, 2008]. Nevertheless, as in the original SD method [Brady and Bossis, 1988, Durlofsky et al., 1987] we found the assumption of a zero trace to be sufficient for our purposes. Since in the classical SD method particles are spheres, the entries of the mobility matrix depend only on the current particle positions and on their radius. The hydrodynamic effect of all particles on the velocity of a single particle thus only depends on the particle configuration, fluid velocities do not have to be computed. If equation (2.3) shall be solved for the velocities, it is called *mobility problem.* If it shall be solved for the forces, torques and stresslets, it is called *resistance problem.* A nondimensionalization of the equations leads to the definition of the Stokes number, which in our case is given by

$$\mathrm{St} = \frac{2}{9}\frac{\rho_p}{\rho_f}\mathrm{Re}. \tag{2.5}$$

ρ_p and ρ_f are the particle mass density and the fluid mass density, respectively. Re denotes the Reynolds number of the flow. If we can assume Stokes flow with vanishing Reynolds number, we can assume the Stokes number to vanish as well, provided that ρ_p and ρ_f are of the same order. With this assumption the left hand sides of the dimensionless form of equations (2.1) and (2.2) become zero. Plugging the resulting equations into equation (2.3), we obtain a system of linear equations where the hydrodynamic forces and torques do not have to be computed explicitly, which allows a very efficient simulation. In the derivation of (2.3) higher moments in the expansion of the force density in the expression for the fluid velocity field in Stokes flow are neglected [Brady and Bossis, 1988]. This leads to an error when particles are in close proximity, as it is usually the case for dense particle suspensions. In this case, or when the Stokes number (2.5) cannot be assumed to be zero, the system of linear equations (2.3) has to be solved. This is because the common schemes to include near-field lubrication effects are all based on the exact solution of the resistance problem for two or three particles [Jeffrey and Onishi, 1984, Jeffrey, 1992, Cichocki et al., 1999]. Formally, in these schemes the inverse of the mobility matrix is altered by the addition and subtraction of two matrices, which gives a better approximation to the original problem. This process involves more computational effort due to the construction of the two lubrication matrices and the solution of the linear system (2.3) which results in a substantially longer simulation time. Therefore, we advise to only use this scheme if a validation shows that it is necessary. We have parallelized both the simplified method for St ≈ 0 and the more complex method for St > 0 and show in section 2.3 the possible speedup one can achieve for both methods.

2.2.2 DLVO

When considering colloidal systems, interparticle forces of non-hydrodynamic nature usually have to be taken into account. We have employed the DLVO theory, named after Derjaguin, Landau, Verwey and Overbeek [Verwey et al., 1999], to model interparticle forces arising from London–van der Waals attraction and electrostatic repulsion. Further, we have included the Born force in our model, to consider the extremely short ranged repulsion covered by Born theory [Feke et al., 1984]. There are several expressions for the London–van der Waals potential and also for the potential resulting from the electrostatic repulsion caused by the formation of a double layer of counterions around colloids. The drawback of most expressions is that they are only valid for monodisperse particle systems. In the following we present the potentials we chose for the simulation of polydisperse particle systems when the mentioned interparticle forces cannot be neglected.

London–van der Waals Attraction

London–van der Waals forces arise from the interactions of permanent and/or induced dipoles between uncharged molecules in particles [Leite et al., 2012]. To model the corresponding London–van der Waals potential we apply the original expression by Hamaker [1937]. Hamaker derived an expression for pair interactions between particles of different sizes, which is given by

$$\Psi_{\mathrm{LvdW}}(r_{\alpha\beta}, a_\alpha, a_\beta) = -\frac{A_{\mathrm{h}}}{6}\left(\frac{2a_\alpha a_\beta}{r_{\alpha\beta}^2 - (a_\alpha + a_\beta)^2} + \frac{2a_\alpha a_\beta}{r_{\alpha\beta}^2 - (a_\alpha - a_\beta)^2}\right.$$
$$\left. + \ln\left(\frac{r_{\alpha\beta}^2 - (a_\alpha + a_\beta)^2}{r_{\alpha\beta}^2 - (a_\alpha - a_\beta)^2}\right)\right),$$

with the Hamaker constant A_{h} and the distance between the particles' centers of mass, $r_{\alpha\beta} := \left\|\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha\right\|_2$. The particle radii are a_α and a_β, respectively.

Electrostatic Repulsion

We have implemented two models for the electrostatic repulsion which take the effect of polydispersity into account. Utilizing the Debye–Hückel approximation for low surface potentials, as well as assuming that the potential-determining ions in the double layer around two considered particles are the same, the potential for spherical particles is given in [Hogg et al., 1966] by

$$\begin{aligned}\Psi_{\mathrm{el}}(r_0, a_\alpha, a_\beta) =& \pi\epsilon_r\epsilon_0 \frac{a_\alpha a_\beta}{a_\alpha + a_\beta}(\psi_\alpha^2 + \psi_\beta^2)\left(\frac{2\psi_\alpha\psi_\beta}{\psi_\alpha^2 + \psi_\beta^2}\ln\left(\frac{1+\exp(-\kappa r_0)}{1-\exp(-\kappa r_0)}\right)\right. \\ & \left. + \ln\left(1-\exp(-2\kappa r_0)\right)\right),\end{aligned}$$

where the shortest distance between the surfaces of the particles is given by $r_0 := r_{\alpha\beta} - (a_\alpha + a_\beta)$. The factors ϵ_r and ϵ_0 are the relative permittivity of the suspending medium and the vacuum permittivity, respectively. ψ_i, $i = \alpha, \beta$, are the total double layer potentials of the particles and κ is the Debye–Hückel reciprocal length parameter.
Using another approach (LSA instead of DA, cf. [Bhattacharjee et al., 1998]), Bell et al. [Bell et al., 1970] have derived a form of the electrostatic potential, which gives a better approximation for values of $\kappa a > 10$. It is given by

$$\begin{aligned}\Psi_{\mathrm{el}}(r_0, a_\alpha, a_\beta) =& \frac{64\pi\epsilon_0\epsilon_r a_\alpha a_\beta}{r_0 + a_\alpha + a_\beta}\left(\frac{k_{\mathrm{B}}T}{e}\right)^2 \tanh\left(\frac{e}{4k_{\mathrm{B}}T}\psi_\alpha\right) \\ & \tanh\left(\frac{e}{4k_{\mathrm{B}}T}\psi_\beta\right)\exp\left(-\kappa r_0\right).\end{aligned}$$

In this equation T is the absolute temperature of the suspension, k_{B} the Boltzmann constant and e the elementary charge.

Born Repulsion

The so-called Born potential models the strong short-ranged repulsion which two particles experience during overlap of the electron clouds of the atoms of which they consist. It is given in [Feke et al., 1984] as

$$\begin{aligned}\Psi_{\mathrm{Born}}(\overline{r},\lambda) = 4A_{\mathrm{h}} \left(\frac{\sigma}{a_\alpha}\right)^{n-6} &\frac{(n-8)!}{(n-2)!}\frac{1}{\overline{r}} \\ &\left(\frac{-\overline{r}^2-(n-5)(\lambda-1)\overline{r}-(n-6)(\lambda^2-(n-5)\lambda+1)}{(\overline{r}-1+\lambda)^{n-5}}\right. \\ &+\frac{-\overline{r}^2+(n-5)(\lambda-1)\overline{r}-(n-6)(\lambda^2-(n-5)\lambda+1)}{(\overline{r}+1-\lambda)^{n-5}} \\ &+\frac{\overline{r}^2+(n-5)(\lambda+1)\overline{r}+(n-6)(\lambda^2+(n-5)\lambda+1)}{(\overline{r}+1+\lambda)^{n-5}} \\ &\left.+\frac{\overline{r}^2-(n-5)(\lambda+1)\overline{r}+(n-6)(\lambda^2+(n-5)\lambda+1)}{(\overline{r}-1-\lambda)^{n-5}}\right).\end{aligned}$$

λ denotes the ratio of radii a_β and a_α, the distance $r_{\alpha\beta}$ between the centers of the two particles is made dimensionless by a_α yielding $\overline{r}$. σ is the separation at which the potential becomes zero and which is, according to Feke et al. [1984], typically $\mathcal{O}(5\mathring{A})$. A commonly used value for n is 12.

Resulting Force and Nondimensionalisation

The force acting on a particle α resulting from the different potentials can be computed as

$$\boldsymbol{F}^i_\alpha = -\sum_{\beta=1}^{N} \frac{\partial}{\partial r_{\alpha\beta}} \left(\Psi_{\mathrm{LvdW}} + \Psi_{\mathrm{el}} + \Psi_{\mathrm{Born}}\right) \boldsymbol{n}_{\alpha\beta}, \tag{2.6}$$

where $\boldsymbol{n}_{\alpha\beta}$ is the normalized vector from the center of particle α to the center of another particle β. To keep equation (2.6) simple, we dropped

the arguments of the potentials.
To account for the polydispersity of a given particle system we have implemented random number generators which give us uni- and bimodal normal and lognormal particle radius distributions. In case of polydisperse systems we have a particle radius distribution with a variance greater than zero. If we have a unimodal radius distribution, we use the expected value μ as characteristic length scale L_0, i.e. we define $L_0 := \mu$. In case of a bimodal distribution we take the major mode as characteristic length scale. For a nondimensionalization of velocities we use the corresponding Stokes velocity as characteristic velocity U_0. To make the total double layer potentials ψ_i, $i = \alpha, \beta$, dimensionless, we choose as characteristic value $\psi_c = k_{\mathrm{B}}T/e$.

Cutoff

As one can see in figure 2.1, all DLVO potentials and also the resulting forces decay rapidly but with differing slope. Furthermore, the potentials between large particles decay slower than the potentials between comparably small particles. To reduce computational costs we introduce different cutoff lengths for different potentials. At the beginning of every simulation we compute the DLVO forces between the two largest particles in the system. We determine where every force is smaller than a given threshold and use the resulting distance as maximum cutoff length $d_{\mathrm{c,max}}$ for all particles. Also shown in figure 2.1 is the extreme increase of the resulting DLVO potential at very short interparticle distances, which is the result of the incorporation of the Born potential. The extreme increase at these distances creates very strong repulsive forces, which demand a very small time step in a simulation. To deal with this numerical problem we have introduced a minimum cutoff length $d_{\mathrm{c,min}}$, which is determined in a similar way as the maximum cutoff length. Only this time we use the two smallest particles in the system and a maximum absolute value for the force. In reality, the repulsive force does not stop at a minimum length. To account for this fact, we have incorporated a contact model into our simulation.

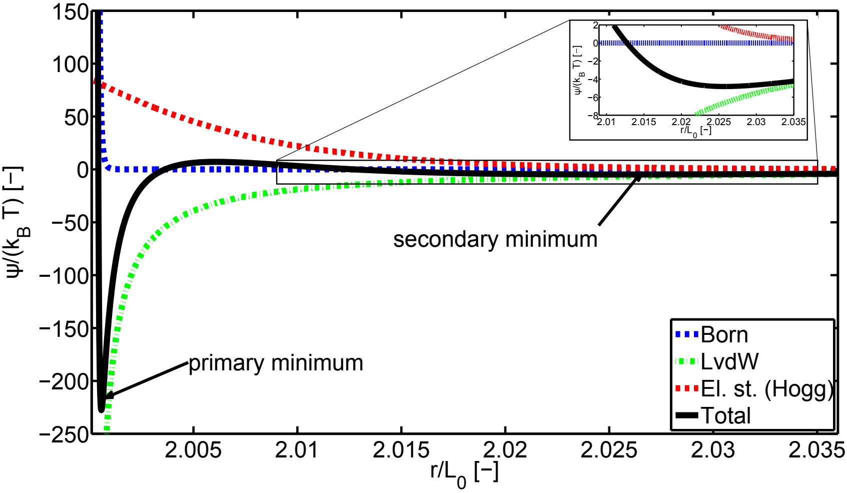

Figure 2.1: Dimensionless DLVO potentials with resulting total DLVO potential plotted over dimensionless interparticle distance r. Shown are the potentials for two particles with dimensionless radius 1. The inset figure shows a zoomed in view where the secondary minimum is visible.

2.2.3 Contact Model

The DLVO potentials have singularities at surface-to-surface distances $r_0 = 0$. Furthermore, the high gradients in the potentials (cf. figure 2.1) require a very small time step once two particles are in close contact. For the sake of allowing larger time steps in the simulation, Schäfer et al. [Schäfer et al., 2010] have replaced the Born potential by the less steep Hertz potential. To capture the case of particle distances shorter than the introduced minimum cutoff length, we make use of the Hertz potential as well. We do not replace the Born potential totally, but we combine it with the Hertz potential to get

$$\Psi_{\text{contact}}(r_{\alpha\beta}, \lambda) = \Psi_{\text{Born}}(d_{\text{c,min}}, \lambda) + k_{\text{H}}(a_\alpha + a_\beta - r_{\alpha\beta})^{2.5}, \qquad (2.7)$$

for all $r_{\alpha\beta}$ with $r_{\alpha\beta} < a_\alpha + a_\beta + d_{\mathrm{c,min}}$. As an additional simulation parameter we get the Hertz constant k_{H}. This constant is a material parameter which depends on the Young's modulus and Poisson's ratio of the particle material [Rosas and Lindenberg, 2003]. Not only DLVO forces but also classical lubrication force models exhibit a singularity at surface-to-surface distances $r_0 = 0$. This singularity arises from the assumptions of classical lubrication theory. These are negligible interparticle forces and smooth particle surfaces, separated by a thin, incompressible Newtonian fluid layer which behaves as a continuum [Davis et al., 1986]. When these assumptions hold, the thin fluid layer prevents the particles from actual contact. Real particles are not perfectly smooth and especially for colloids interparticle forces of non-hydrodynamic nature are non-negligible. Furthermore, at extremely small interparticle distances the interstitial fluid cannot be treated as continuum anymore. In [Zhang et al., 2005] a lubrication force model is derived, which accounts for rough particle surfaces, non-hydrodynamic forces and a non-continuum effect. In their numerical study on discontinuous shear thickening, Seto et al. [2013] also apply a lubrication model which accounts for surface roughness. Furthermore, they employ a model for frictional particle contact, which enables them to show discontinuous shear thickening in concentrated suspensions via a simplified SD method. These studies show that a contact model should be included, in order to simulate real suspension behaviour.

2.2.4 Parallelization

On distributed parallel computer systems multiple processes, that means instances of the same program, are running simultaneously. When using MPI, they can be distinguished by their unique rank within the group of processes. Due to the distributed memory, a process does not have constant access to all data, but only to data stored in its private memory. As most data fields have entries referring to particular particles, we store these fields according to the particle it refers to. When we say in the following that a particle is stored on a process, we mean all data related to this particle is stored in the memory related to the process. For an optimal load balance on P processes, $N \mathrm{mod} P$ processes get $\lceil \frac{N}{P} \rceil$ particles and the other processes receive the data of $\lfloor \frac{N}{P} \rfloor$ particles. The brackets $\lceil \cdot \rceil$

denote the smallest integer number greater than the value of a variable, the brackets $\lfloor \cdot \rfloor$ denote the integral part of its value. Another aspect we aimed at when designing the parallel code was to keep low redundancies, i.e. having data only stored in the memory of the processes that need it. Generally, reduced memory usage results in more communication, but as we want to simulate a large number of particles with a large number of processes, we accept these costs. When it comes to communication between processes on a distributed memory machine, different strategies have to be applied for the solution of equation (2.3) and for the computation of interparticle forces of non-hydrodynamic nature. We present the two strategies we found to be most effective for these two different types of problems. Subsequently we present the general algorithm we employ for the simulation of colloidal suspensions.

Communication Strategies

On a distributed memory machine all processes use private memory. The major difference to sequential or shared memory programming is the need to exchange data between processes. This can be done either using point-to-point communication between two processes, or as a collective operation, where all nodes communicate with each other. We deploy two extensions to these communication types using point-to-point communication, in order to perform collective operations on large data fields effectively.

Ring Communication

For some functions in our software, large distributed data fields need to be evaluated on every single process. Using collective communication would mean storing the whole data in the memory of every node. To avoid this we use so-called *ring communication.* Imagine the nodes to be ordered in a closed circle, each of them having a connection to its direct neighbours. Instead of all processes sending their data simultaneously to all other processes, the data packages are handed from one processes to the other in the ring. After P steps all data has again reached its origin and the communication process is finished. Between the communication steps the data can be processed at its current location. This type of communication has proven to be most effective when every process needs the data stored in the memory of every other process, as it is the case for the computation of the matrix-vector product in equation (2.3).

Sparse Communication

There are other cases, when every process has only few communication partners. To provide each process the possibility to communicate with an arbitrary group of other processes, we implemented another communication method, we call *sparse communication.* Each process may list the ranks of desired receivers and submit them to a coordinating routine, where the partners for all requesting processes are determined. Due to coinciding requests some processes might not obtain permission to send and have to wait for the next communication step. Then the data is communicated according to the priorities of the coordinating routine. This is repeated until the lists of all processes are executed. Analogous to the ring communication, the data can be handled on the receiving processes between the individual communication steps. Sparse communication is more effective than ring communication when the number of partners, i.e. the number of processes one process has to communicate with, is significantly lower than the total number of processes P. Thus it is ideal for the computation of short to medium ranged interparticle forces, such as DLVO or contact forces, provided that particles located in the same region of the simulation domain are stored on the same process. To achieve this we have implemented a merge-sort like parallel sorting algorithm which can be called at a chosen frequency and which preserves load balance. After the particle data has been sorted, the number of communication partners for each process is reduced to a minimum. The criterion used to determine whether two processes communicate is depicted in figure 2.2. Every process stores center and radius of the smallest sphere which contains all particles stored in its memory. If the surface-to-surface distance $r_{0,p_{i,j}}$ of two spheres which contain all particles stored on two different processes i, j is shorter than the maximum cutoff length $d_{\mathrm{c,max}}$, the two processes have to communicate.

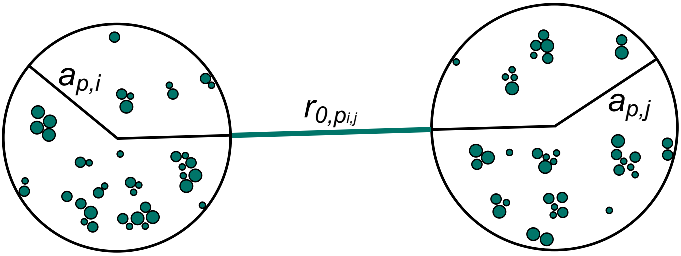

Figure 2.2: Particle configuration with bounding spheres of radius $a_{p,i}$ for particles on processes $i, j \in [1, P]$. The surface-to-surface distance between the spheres is denoted by $r_{0,p_{i,j}}$. Particles with data stored in the memory of other processes are not shown.

General Algorithm

The general algorithm of our simulation is shown in figure 2.3. After initialization and parameter input the particle data is sorted such that data of groups of particles in spatial proximity is in the memory of the same process [Bülow et al., 2016]. After this step forces acting on the particles are computed. These can be the described DLVO forces, centrifugal forces or forces arising from Brownian motion (see e.g. [Banchio and Brady, 2003]). Then the coupled system (2.1), (2.2) and (2.3) is solved. In case of $\mathrm{St} = 0$ this is done by a multiplication of the mobility matrix and the summed forces. The vector of stresslets $\boldsymbol{S}^h$ can be computed by a solution of $\boldsymbol{M}_{ES}\boldsymbol{S}^h = \boldsymbol{E}^\infty - \boldsymbol{M}_{EF}\boldsymbol{F}^\mathrm{h} - \boldsymbol{M}_{ET}\boldsymbol{T}^\mathrm{h}$, which is a subproblem of equation (2.3). In case the Stokes number cannot be assumed to vanish, the full system of linear equations (2.3) has to be solved. For the computation of the stresslets and the solution of (2.3) we use the open source library PETSc [Balay et al., 2013] which provides routines for the parallel solution of systems of linear equations. To allow for a large number of particles to be simulated we use iterative solvers exclusively, since the mobility matrix is dense and a complete storage of all matrix entries would result in immense memory usage. After this step the translational velocities are integrated in time to yield the new particle positions. For this we utilize our parallel implementation of the adaptive (4,5)-Runge-Kutta-Fehlberg method [Fehlberg, 1970]. In principle every process uses the embedded

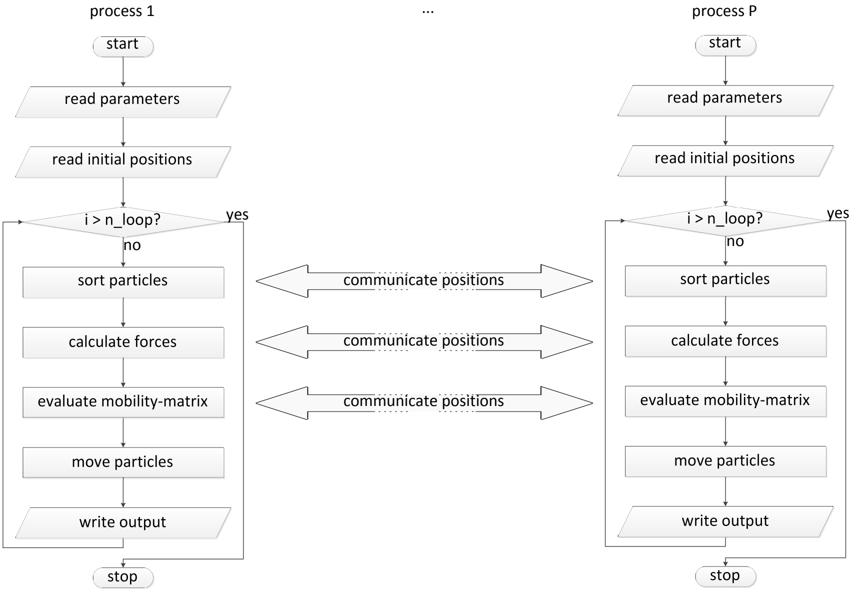

Figure 2.3: General algorithm of our parallel implementation of the SD method.

scheme for its data locally, where the global minimum of the proposed local time steps from the local solutions is taken as global time step for all processes. As we show in section 2.3.1 this method gives good results regarding speed and accuracy.

Complexity

For reasons of simplicity we now assume that the number of particles stored on a process is the same for every process $p \in [1, P]$. This means $N \mathrm{mod} P = 0$, which results in N/P particles per process. Then the complexity of our sorting algorithm is $\mathcal{O}\big(N \log(P) \log(N/P)/P + N\big)$. External forces such as centrifugal forces are independent from the positions of other particles and are computed at constant cost for each particle. The gravitational force acting on each particle can be computed once at the beginning of a simulation and does not contribute to the computational costs afterwards. For the interparticle forces acting on a particle we have to check the distances to all other particles. This implies costs of $\mathcal{O}(N^2/P)$ for each process. Exploiting the short range nature of interparticle forces and the fact that particles are spatially sorted, only the distance to other particle domains has to be checked (cf. 2.2.4). Only if the domain is inside the force's scope we calculate the distance to the single particles in a considered domain. Thus the complexity of this step is reduced to $\mathcal{O}(P + k\,(N/P)^2)$, with a constant $1 \leq k \leq P$, depending on the particle constellation. For the computation of Brownian forces as well as for the solution of the mobility problem or the resistance problem, the mobility matrix $\boldsymbol{\mathcal{M}}^\infty$ has to be calculated. In case of $\mathrm{St} = 0$ the matrix is applied once to the combined force-torque-stresslet vector. When the Stokes number is non-negligible we use iterative methods to solve equation (2.3) for this vector. In this case $\boldsymbol{\mathcal{M}}^\infty$ has to be applied multiple times. Calculating a single entry of the solution vector to the matrix-vector product is $\mathcal{O}(N)$. Hence for all entries computed by one process we have $\mathcal{O}(N^2/P)$. This is the complexity determining the overall complexity of the program. Increasing the number of used processes P means decreasing the number of particles per process, which reduces the CPU time of the simulation linearly by a factor of $1/P$. Of course, this is a theoretical result which has some limits as we show in section 2.3.

2.3 Results

2.3.1 Validation

We have validated our code on several examples. In the following we present some results, which are not only of interest for the validation, but also for the application of a parallel Stokesian Dynamics code. All simulations have been performed using the IC2 computer cluster [ic2, 2013]. Simulations run on another computer cluster at our university [hc3, 2014] show the same results with a slightly longer overall CPU time due to slower computing nodes.

DLVO Forces

The correct computation of DLVO forces was validated by means of a MATLAB code, which solves equation (2.1) neglecting all acting forces but the interparticle forces $\boldsymbol{F}^i_\alpha$, $\alpha = 1, \ldots, N$. Neglecting other forces such as $\boldsymbol{F}^b_\alpha$ and $\boldsymbol{F}^{ext}_\alpha$ yields a stationary configuration of two particles depending on their initial position and the form of the DLVO potential. Due to the order of magnitude of DLVO forces acting on particles close to a potential minimum, this configuration is the same as in the SD simulation, even with hydrodynamic interactions. For all initial configurations, the SD simulation agrees well with the test results from our MATLAB routine.

Error due to Truncation

As discussed in section 2.2.1, the use of equation (2.3) without any correction scheme leads to an error when two particles are in close proximity. We have examined the effect of solving the mobility problem with St = 0 without the application of a near-field lubrication scheme on the example of two equal spheres settling in a predefined configuration in an otherwise quiescent fluid ([Goldman et al., 1966]; [Bülow et al., 2015]). When we solve the resistance problem and apply a lubrication correction as in [Durlofsky et al., 1987], our results agree perfectly with the results of Goldman et al. [1966]. The particle settling velocity obtained from the solution of the mobility problem without lubrication correction starts to deviate by more than 1% at a distance of one particle diameter between the particle surfaces, where this deviation increases with shrinking gap size. The deviation is below 5% for all configurations, down to interparticle gaps of 1% of a particle diameter [Bülow et al., 2015]. Important for this work is the location of the secondary minimum of the DLVO potential exemplarily shown in figure 2.1. The resulting DLVO potential exhibits a secondary minimum at interparticle gaps of 1.3% of a particle diameter, which yields a deviation in the computation of the settling velocity below 5%. Without Brownian forces, this minimum is deep enough to keep particles from reaching the primary minimum. We show in section 2.3.4 that the solution of the mobility problem for St = 0 without lubrication scheme is computationally much more efficient than the solution of the problem when the system of linear equations has to be solved. For this reason we advise to solve the system for St = 0 whenever possible. But the resulting error should be kept in mind when the settling velocity of single particles is of importance.

Stability of the Integration Scheme

To examine the stability of our parallel implementation of the adaptive (4,5)-Runge-Kutta-Fehlberg method we have run simulations as in [Durlofsky et al., 1987]. Eight monodisperse particles placed at the corners of a cube (cf. figure 2.4, left) undergo a periodic motion when they settle at vanishing Stokes number. The top four particles settle faster than the four particles at the bottom, pass them and all particles return to the corners of the initial cube. Only this time the locations of top and bottom particles are switched. Now the formerly bottom particles settle faster and return to their original positions at the four bottom corners of the cube. This motion is periodic and theoretically repeats itself infinitely often. Due to numerical errors symmetry is broken after a while, which leads to a breakup of the whole configuration (cf. figure 2.4, right). Our results when using the FT level of approximation, i.e. considering hydrodynamic forces and torques in equation (2.3) but no stresslets, with additional lubrication scheme agree very well with the results in [Durlofsky et al., 1987]. If we solve the mobility problem for St = 0 without lubrication scheme, we still get good agreement with the results in [Durlofsky et al., 1987]. Initially, the particles are placed at the corners of a cube with center of mass at the origin. As in [Durlofsky et al., 1987] we investigated particle

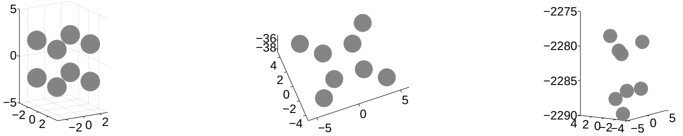

Figure 2.4: Left: Initial positions at the corners of a cube with dimensionless edge length 4. Middle: Configuration after 16 dimensionless time units, when particles first lie in a plane perpendicular to the settling direction. Right: Broken configuration after 1000 dimensionless time units.

configurations with dimensionless edge lengths $r_{\text{init}} = 3, 4, 5$. To prevent a possible particle overlap, we use the Hertz model given by the potential (2.7). We have run the simulations for St $= 0$ and for St $= 0.0001$, which gives similar results, at the FT level of approximation. Furthermore, we have run simulations with St $= 0$ at the FTS level of approximation. This level corresponds to the solution of the full system (2.3), including hydrodynamic stresslets $\boldsymbol{S}^h_\alpha$. To investigate the effect of the parallelization of the integration scheme we have run all simulations on a single process and on eight processes, where each process treats one particle. To determine the stability we determine a symmetry breaking point t_{break}. We define this point as that point in time when the projection of the center of mass $\boldsymbol{x}_{\text{CoM}}$ of the particle system on the x-y-plane deviates from the center of origin by more than 1% in the 2-norm. So symmetry is broken when $||\boldsymbol{x}_{\text{CoM}}||_2 > 0.01$. We use this criterion additionally to counting the number of periods n_{per} the particle system undergoes (cf. [Durlofsky et al., 1987]), because it is more sensitive. Even when symmetry is broken, the particle system continues the periodic movement for several periods. We determined the number of periods by the number of times the particles lie in the same plane (cf. figure 2.4, middle). The average time t_{per} for one period is determined by averaging all measured periods. Table 2.1 shows our results for different levels of approximation, different Stokes numbers St and the three different edge lengths r_{init} of the cube, which were also investigated in [Durlofsky et al., 1987]. The simulations using the FTS version of SD are slightly more stable than when the FT

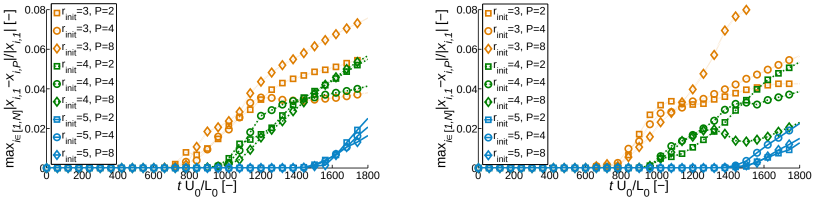

Figure 2.5: Maximum relative deviation of particle positions $X_{i,P}$, $i = 1, ..., N$, obtained from simulations when different numbers of processes $P > 1$ are used, from particle positions $X_{i,1}$ when $P = 1$. Left: Deviation over time for the solution of the mobility problem (St $= 0$) using the FT-approximation. Right: Deviation over time for the solution of the resistance problem (St > 0) using the FT-approximation.

Level of approximation	St	P	r_{init}	n_{per}	t_{break}	t_{per}
FT	0	1	3	20.5	581	31.9
FT	0	8	3	20	584	32.7
FT	0.0001	1	3	19	542	32.9
FT	0.0001	8	3	19.5	549	32.1
FTS	0	1	3	21	730	39.9
FTS	0	8	3	21.5	747	39.0
FT	0	1	4	14	798	61.9
FT	0	8	4	14.5	806	59.7
FT	0.0001	1	4	14	771	61.9
FT	0.0001	8	4	15	804	61.9
FTS	0	1	4	14.5	824	64.4
FTS	0	8	4	14	824	66.7
FT	0	1	5	14.5	1247	96
FT	0	8	5	14	1220	99.4
FT	0.0001	1	5	13.5	1195	95.4
FT	0.0001	8	5	14	1249	99.1
FTS	0	1	5	13	1208	103.5
FTS	0	8	5	13	1199	103.5

Table 2.1: Number of periods n_{per}, symmetry break point t_{break} and average time for one period t_{per} for the different considered setups.

approximation is used. Our parallel implementation of the adaptive (4,5)-Runge-Kutta-Fehlberg method performs as good as the sequential version, as one can see in table 2.1 when comparing the number of periods n_{per} and the symmetry break points t_{break} for $P = 1$ and $P = 8$. In general our integration scheme is quite stable, as our simulations are stable for at least 542 dimensionless time units and the values we obtain for n_{per} are always higher than 10. Durlofsky et al. [1987] name this value as approximate maximum number of periods in their simulations. Figure 2.5 shows the deviation of particle positions gained from simulations for which the number of used processes $P > 1$, compared with simulations for which $P = 1$. As one can see, the deviation is negligible until the breakup of the symmetric particle configuration (cf. table 2.1 for breakup times). The significant deviation after breakup is not surprising, as the breakup is triggered by slowly increasing numerical errors and particle trajectories are very sensitive to small deviations, such that the exact particle trajectories differ strongly after the breakup event. This fact has been reported in [Durlofsky et al., 1987]. Another problem where the sensitivity of particle trajectories plays a role can be found e.g. in [Sharma and Patankar, 2005]. In addition, we have run simulations with the same set of parameters, but with a maximum allowed time step $\Delta t_{\max}$ lower than any time step proposed by the time integration scheme ($\Delta t_{\max} = 10^{-5}$ instead of the originally used $\Delta t_{\max} = 0.1$). With this choice all simulation results agree perfectly over a chosen simulation time of 100000 dimensionless time steps and $P \in \{1, 2, 4, 8\}$.

2.3.2 Settling Cloud of Particles

Several works such as [Nitsche and Batchelor, 1997], [Metzger et al., 2007] or [Abade and Cunha, 2007] deal with the settling of particle clouds in a quiescent fluid. Depending on the level of approximation and the method used for the simulation, sequential codes are usually limited to the simulation of several hundred to a few thousand particles. A parallelized SD method is capable of simulating the dynamics of several thousand particles within hours. Even the simulation of a million particles is possible within days. To give an example of the evolution of a large particle system, we have simulated the dynamics of a settling polydisperse particle cloud comprised of 20000 particles over 4000 dimensionless time units within 7 days, using 256 processes. Different stages of the initially spherical particle cloud are shown in figure 2.6. The initial volume concentration of particles inside the cloud is 1%, the radius distribution is a lognormal distribution with expected dimensionless radius 1 and a standard deviation of 0.065 dimensionless length scales. The chosen level of approximation is FT, we assume $\mathrm{St} \approx 0$. As shown in [Bülow et al., 2015], polydisperse particle clouds behave like monodisperse particle clouds, if the number of particles in the initial cloud is large enough. The initially spherical cloud (figure 2.6 a)) constantly loses particles as it settles. It forms a torus (figure 2.6 b)), which eventually breaks up into smaller clouds (figure 2.6 c)). This evolution continues until the number of particles per cloud is too small to form a coherent cloud and the clouds dissipate. As one can see in figures 2.6 d), e) and f), the breakup of polydisperse clouds does not have to result in two sub-clouds, as it is the case for monodisperse particle clouds settling at vanishing Reynolds number. The facts that also polydisperse particle clouds undergo this characteristic evolution, and that they

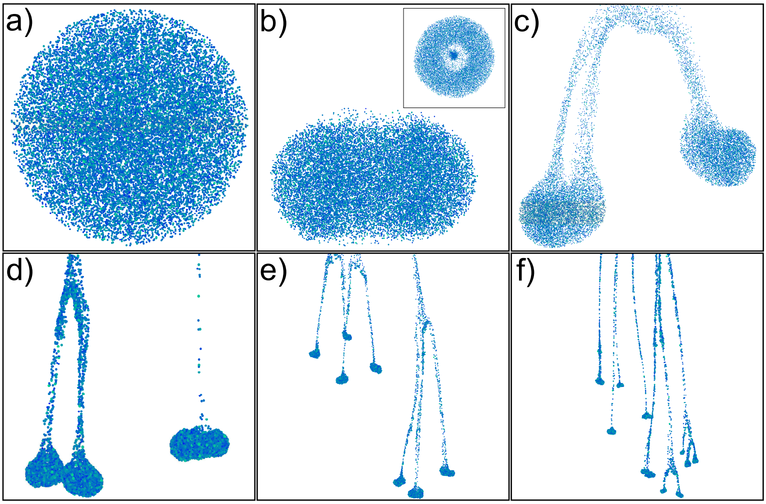

Figure 2.6: Different stages of a polydisperse particle cloud comprised of 20000 particles settling under the influence of gravity. Subfigure b) shows additionally to the side view a view from above, where the torus form of the cloud is visible. The tail of leaked particles is not shown in subfigures b)-f). For better visibility particles have been enlarged by a factor of 5 in subfigures d) and e) and by a factor of 10 in f).

can break up into more than two sub-clouds per breakup event are only visible if the number of particles is large enough. This is just one example which shows that due to parallelization it is not only possible to simulate large particle systems, but it is also necessary in order to see effects which are not visible when small particle systems are simulated.

2.3.3 Settling Cloud of Particles with Non-Hydrodynamic Interactions

As discussed in section 2.2.4, a parallelization can reduce the complexity of the simulation algorithm immensely. This makes computationally expensive simulations more feasible. Exemplarily, we have simulated the dynamics of 1000 colloids which interact not only hydrodynamically, but also through forces arising from the interparticle interactions described in sections 2.2.2 and 2.2.3. The particles are initially placed randomly in a spherical domain with an initial particle volume fraction of 1% and the same radius distribution as in section 2.3.2. The parameters for the interparticle forces were chosen such that the resulting potential was the potential depicted in figure 2.1. The chosen level of approximation is again FT. While a dimensionless time step is usually about 0.1 for purely hydrodynamic problems such as the simulation presented in section 2.3.2, simulations including DLVO interactions typically require a dimensionless time step of 1e-5. We have simulated 100 dimensionless time units corresponding to 10 million time steps on 32 processes within 11 days. Snapshots from the simulation are shown in figure 2.7. As one can see in figure 2.7 in the middle, where the particle cloud after 0.9 dimensionless time units is depicted, and on the right hand side, where the configuration after 100 dimensionless time units is shown, particles form aggregates which grow with time and become more compact. The aggregation of particles can be examined further through appropriate postprocessing, for which we use a cluster detection algorithm based on a R-tree data structure. With this tool the evolution of cluster size and -structure can be examined in detail and the effects of different conditions such as an increase in gravitational acceleration or a change in the ionic strength, which directly influences the Debye–Hückel reciprocal length κ, can be investigated.

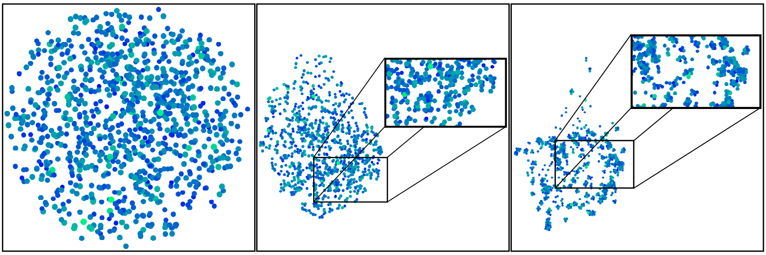

Figure 2.7: Left: Initial configuration of 1000 particles. Middle: Configuration after 0.9 dimensionless time units. Small fractal aggregates are visible. Right: Configuration after 100 dimensionless time units. Aggregates have grown in size and in their fractal dimension. The tail of leaked small aggregates and particles is not shown.

2.3.4 Efficiency of the Parallelization

The SD method itself already is an effective method for the simulation of colloidal systems, as the fluid motion is implicitly treated and the fluid velocity field does not have to be computed explicitly. In the following we show that a code parallelization can increase the number of particles which can be simulated by several orders of magnitude, while keeping the CPU time low. To determine the efficiency of a parallelization, a value called speedup is used. When no data of a sequential program is available the incremental speedup is used to measure the efficiency of a parallelization.

Speedup

The speedup $S(P)$ of a parallel algorithm running on P processes is defined as

$$S(P) := \frac{T(1)}{T(P),}$$

where $T(1)$ and $T(P)$ are the execution time for a single process and for P processes, respectively. Figures 2.8 and 2.9 show the speedup we achieved for the solution of different problems using different numbers of processes P and different numbers of particles N. The dashed lines in the graphs indicate linear speedup, where $S(P) = P$. A speedup higher than this is usually explained by cache effects. A speedup much lower than linear speedup is not desirable, as a further increase in the number of processes would not significantly decrease the CPU time. Due to increased communication between processes, a higher number of processes can actually increase CPU time, compared to a lower number of processes. We see this effect in figure 2.8, where the speedup we achieve for the solution of the mobility problem at the FT-level of approximation for a monodisperse system of 10000 particles is around 44 when we use 64 processes, whereas if we double the number of processes, the speedup decreases to a value of 36. The speedup in case of a polydisperse particle system shows the same behaviour as in the monodisperse case. As one can see in figure 2.8,

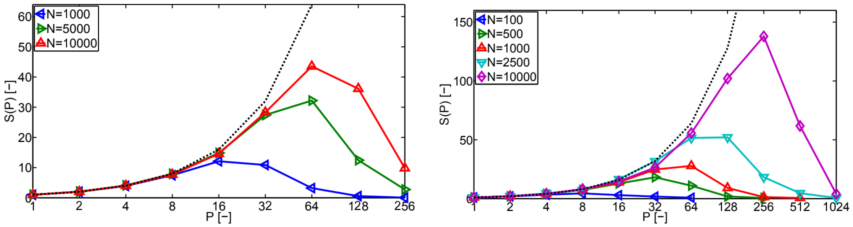

Figure 2.8: Left: Speedup $S(P)$ for the solution of the mobility problem (St $= 0$) using the FT-approximation for a monodisperse particle system. Right: Speedup $S(P)$ for the solution of the resistance problem (St > 0) using the FT-approximation for a polydisperse particle system.

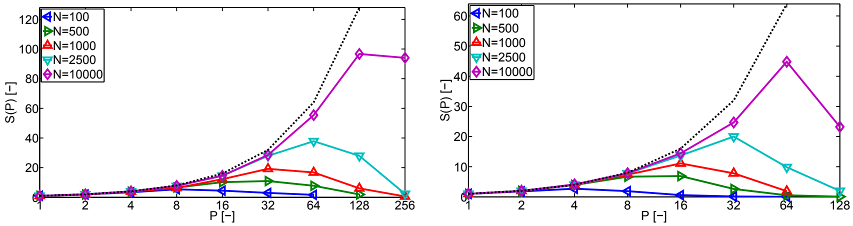

Figure 2.9: Left: Speedup $S(P)$ for the solution of the resistance problem (St > 0) using the FT-approximation for a monodisperse particle system with lubrication scheme as in [Durlofsky et al., 1987]. Right: Speedup $S(P)$ for the solution of the mobility problem (St $= 0$) using the FT-approximation for a polydisperse particle system with DLVO interactions.

right, the parallelized SD algorithm scales quite good, even when we solve the work-intensive resistance problem. When we additionally include a lubrication correction scheme as in [Durlofsky et al., 1987], we obtain a similarly good speedup (cf. figure 2.9, left). For the solution of the linear system we use the standard conjugate gradient method provided by PETSc. Figure 2.9 also shows the speedup which can be achieved when solving the mobility problem for a polydisperse particle system considering DLVO interactions with potentials as shown in figure 2.1. The achievable speedup is comparable to that of the solution of the mobility problem without DLVO interactions (cf. 2.8, left), but due to the steep DLVO potential for short interparticle distances, the time step is much smaller and thus the overall CPU time for a simulation over the same time span is much higher.

Incremental Speedup

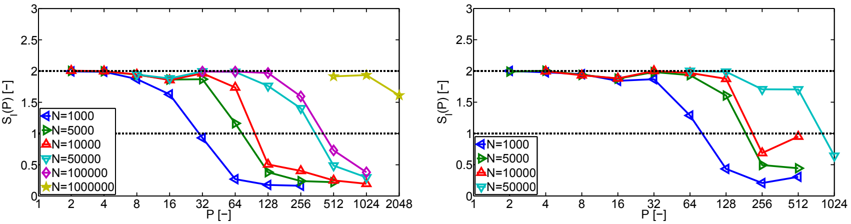

Figure 2.10: Left: Incremental speedup $S_{\mathrm{I}}(P)$ for the solution of the mobility problem (St = 0) using the FT-approximation for a polydisperse particle system. Right: Incremental speedup $S_{\mathrm{I}}(P)$ for the solution of the mobility problem (St = 0) using the FTS-approximation for a polydisperse particle system.

By definition, the computation of the speedup $S(P)$ depends on the execution time using one process. When a simulation cannot be run on a single process anymore, e.g. because the number of simulated particles becomes too large, the efficiency of the parallelization can be measured by the incremental speedup,

$$S_{\mathrm{I}}(P) := \frac{T(P/2)}{T(P)}.$$

It is defined as the ratio of the execution times of two computations when the number of processes is doubled. Figure 2.10 shows the incremental speedup for selected applications. The upper dashed line in the graphs represents optimal incremental speedup. This is reached when the execution time is halved by doubling the number of used processes. Data points below the lower dashed line in figure 2.10 show that the program becomes slower when doubling the number of processes. As one can see in figure 2.10, the mobility problem can be solved very efficiently in parallel, even for a large number of particles (figure 2.10, left) or at a higher level of approximation (figure 2.10, right). As a rule of thumb, the more particles shall be simulated, the more processes can be used with optimal performance for this type of problem. Even the use of over one thousand processes still yields a speedup for relatively large problems.

2.4 Conclusions

We have shown that a parallelized Stokesian Dynamics code running on a distributed memory machine is a very effective tool for the simulation of large particle systems. The achievable speedup can attain values of over 100, making the parallel algorithm more than 100 times faster than a sequential version. Depending on the problem to be solved, for large particle systems even more than a thousand processes can be used to their full capacity. This allows the simulation of the dynamics of one million particles within days. For supercomputers with more available processes, even larger numbers of particles should be doable. The resulting savings in time give rise to the simulation of bigger particle systems where not only hydrodynamic interactions play a role. We have incorporated attractive London–van–der–Waals forces, repulsive electrostatic forces and forces arising from particle contact for polydisperse particle systems into our simulation. The SD method allows incorporating such interparticle interactions in an elegant way as the additional forces can simply be added to other external forces acting on the particles. For the communication between processes, different strategies have to be applied for hydrodynamic interactions on the one hand and for non-hydrodynamic interactions on the other hand. Theoretically, the proposed strategies can also be applied to other simulation methods, as usually the computation of fluid velocities involves the solution of a system of linear equations, which can be done best with ring communication, whereas non-hydrodynamic interactions require sparse communication due to the limited number of communication partners. The SD method applied to infinite systems can be parallelized in essentially the same way when the standard Ewald summation is used to compute hydrodynamic interactions among particles.

Chapter 3

A scalable parallel Stokesian Dynamics method for the simulation of colloidal suspensions

Abstract

We have developed a new method for the efficient numerical simulation of colloidal suspensions. This method is designed and especially well-suited for parallel code execution, but it can also be applied to single-core programs. It combines the Stokesian Dynamics method with a variant of the widely used Barnes–Hut algorithm in order to reduce computational costs. This combination and the inherent parallelization of the method make simulations of large numbers of particles within days possible. The level of accuracy can be determined by the user and is limited by the truncation of the used multipole expansion. Compared to the original Stokesian Dynamics method the complexity can be reduced from $\mathcal{O}(N^2)$ to linear complexity for dilute suspensions of strongly clustered particles, N being the number of particles. In case of non-clustered particles in a dense suspension, the complexity depends on the particle configuration and is between $\mathcal{O}(N)$ and $\mathcal{O}(Pn_{p,\mathrm{max}}^2)$, where P is the number of used processes and $n_{p,\mathrm{max}} = \lceil N/P \rceil$, respectively.

This chapter follows our work of the same title under consideration for publication in the journal Computer Physics Communications.

3.1 Introduction

Colloidal suspensions, which means particles of diameter between 10 nm and 10 μm suspended in a fluid, can be found in a vast number of industrial processes. Even though currently research in this field is blooming, the behaviour of colloids is not fully understood yet. Physical experiments are either time consuming, costly, or simply not possible at the moment. This is where numerical simulations provide a valuable aid to experimental research. Since in numerical experiments influencing factors can be turned on or off easily, e.g. the nature of interparticle interactions and their impact on the overall behaviour of a suspension can be understood better. Due to the long-range nature of hydrodynamic forces which decay at a rate of $1/r$, where r is the interparticle distance, this type of problem can be seen as so-called N-body problem [Ichiki, 2002]. Such problems are characterized by long-range interactions of the N involved bodies. These interactions can be of any nature and appear on any length scale. On the atomic level [Board et al., 1992], [Winkel et al., 2012] the Coulomb force, which decays at rate $1/r^2$, acts between molecules. On a larger length scale, e.g. between toner particles which have typically the size of some microns, hydrodynamic forces and Coulomb forces cause a non-negligible N-body interaction [Hoffmann, 2006]. Even the simulation of the collision of galaxies is an N-body problem due to the long-range gravitational force acting between stars, which can also be seen as particles [Dubinski, 1996]. Due to the fact that every particle in the system interacts with every other particle, these problems inherently have a complexity of $\mathcal{O}(N^2)$ if no approximations are made. The two most prominent methods to reduce complexity are the Barnes–Hut algorithm [Barnes and Hut, 1986], [Barnes, 1990] and the fast multipole method (FMM) by Greengard and Rohklin [Greengard and Rohklin, 1987], [Carrier et al., 1988]. The latter has typically a complexity $\mathcal{O}(\log(1/\epsilon)^3 N)$, where ϵ is the desired accuracy of the approximation [Machu et al., 2001]. It can be parallelized and has been applied successfully to a wide range of N-body problems [Ichiki, 2002], [Ogata et al., 2003], [Kurzak and Pettitt, 2005] as well as to mathematically similar problems [Darve, 2000]. The Barnes–Hut algorithm is a tree-based method like the FMM. Although its complexity of $\mathcal{O}(N \log(N))$ is asymptotically higher than that of the FMM, the associated constants are smaller, particularly for simulations in three dimensions [Grama et al., 1998]. Because of its intuitivity and its ease of implementation, besides the FMM, the Barnes–Hut algorithm has become the most popular fast particle method [Koumoutsakos, 2005]. Utilizing this method and code parallelization, extremely large problems can be solved efficiently [Dubinski, 1996], [Winkel et al., 2012]. Apart from its original purpose of reducing the numerical costs for the computation of far-field interactions, it can also be used for collision detection in discrete element simulations [Hoffmann, 2006]. A direct comparison of FMM and Barnes–Hut can be found in [Singh et al., 1995]. Both methods make use of clustering of particles. The FMM depends on the interparticle potential to be expandable around the cluster centers, where the expansion is cut off at a desired order of approximation. The Barnes–Hut algorithm does not have this restriction but also makes use of particle clustering and a tree data structure, where the level of approximation can be controlled by the tree-depth and a chosen parameter.

Mesh free techniques like the Stokesian Dynamics (SD) method [Durlofsky et al., 1987], [Brady and Bossis, 1988] are an elegant way to solve hydrodynamic N-body problems arising from the simulation of colloidal suspensions. They do not rely on the explicit computation of the fluid velocity at a fixed computational grid. Instead, the fluid is treated implicitly and no solver for the fluid phase is necessary. This reduces computational costs and fast particle methods such as FMM or Barnes–Hut can be used to their full effect. SD has been coupled with several fast particle methods in order to reduce its generic complexity of $\mathcal{O}(N^2)$ resulting from the evaluation of a matrix-vector product. For example in the widely used accelerated Stokesian Dynamics method by Sierou and Brady [2001], a particle-mesh Ewald (PME) method has been applied to the original SD method in order to achieve a $\mathcal{O}(N\log(N))$-scheme for periodic domains. Saintillan et al. [2005] use a similar technique to simulate non-Brownian rigid fibres by means of slender-body theory [Butler and Shaqfeh, 2004]. Ichiki [Ichiki, 2002] has successfully utilized the FMM to obtain a SD method with complexity between $\mathcal{O}(N)$ and $\mathcal{O}(N^{3/2})$. This seems to be a favourable method, but there is a big additional overhead, which makes this scheme superior to the standard SD method only for large numbers of particles. Furthermore it is nontrivial to code, not speaking of a possible parallelization. Due to its advantages and its inherent suitability for parallel code execution on distributed memory machines, the Barnes–Hut algorithm is our method of choice in order to accelerate the already computationally efficient SD method. We have employed the concept of the Barnes–Hut algorithm to the original SD method, focussing on an execution on supercomputers which become more and more the standard tool for simulations of large particle systems. In this work we present a modified version of Stokesian Dynamics which we call scalable Stokesian Dynamics (SSD). We call it scalable because the level of approximation can be chosen via input parameters and the method is designed for an execution on parallel machines, where the size of the simulated particle system can grow with an increasing number of computing cores, which at the same time makes the method more efficient [Bülow et al., 2014]. First we briefly introduce the original SD method in order to clarify the terminology and the nature of the problem. After that we show how we have merged the Barnes–Hut algorithm with SD to obtain the load balanced SSD. We investigate the error resulting from the approximation and show how we group particles and their data in order to achieve optimal load balance and low computational costs. There are other works which propose methods to simulate particle suspensions, such as [Sangani and Mo, 1996], [Hernández-Ortiz et al., 2006], [Lomholt and Maxey, 2003] or [Tornberg and Greengard, 2008]. The authors of the latter rewrite Stokeslet and Stresslet summation formulas in such a way that the harmonic FMM can be used in "black-box" fashion, which is an interesting point for users who do not want to write their code from scratch. As Maxey and Patel [2001] state in their work, multipole expansions have, in one form or another, become a standard technique for the simulation of Stokes suspensions. The main differences between existent approaches lie in the method used to assign the multipole strengths either in terms of force densities on the particle surface or the boundary conditions, as well as in the inclusion of lubrication forces and the order of the multipole expansion. Even though other methods have achieved a considerable reduction of the overall numerical costs, they have not been designed for parallel code execution. This is the main advantage of our method. It scales with the number of cores which are used for the simulation. If a P-fold amount of cores is used, roughly the P-fold amount of particles can be simulated and our proposed method can be applied more effectively [Bülow et al., 2014].

3.2 The method

We present the SSD method based on the original Stokesian Dynamics method by Brady and Bossis [Brady and Bossis, 1988] without any near-field correction. There are several schemes to include near-field lubrication effects [Jeffrey and Onishi, 1984, Jeffrey, 1992], [Cichocki et al., 1999], [Nitsche and Batchelor, 1997], [Abade and Cunha, 2007] and [Cunha et al., 2002]. An inclusion of the schemes described in the works of Jeffrey and Onishi [1984] and that of Cichocki et al. [1999] in the described SSD method is possible by modifying the lubrication matrix in a similar manner as we alter the original mobility matrix to obtain a simplified mobility matrix. Yet this is not necessary. The effects arising from lubrication decrease rapidly with increasing interparticle distance. Due to this behaviour and the high numerical costs of the computation of the entries of the lubrication matrix one usually uses a cutoff length after which the entries are set to zero. This length is typically chosen to be smaller than 10 particle radii. For such distances the resulting mobility matrix proposed in this work reduces in most cases to the standard SD mobility matrix used in a parallel implementation of the original SD method [Bülow et al., 2014], making a reformulation of the lubrication matrix futile. Nevertheless, the tree structure resulting from the proposed grouping of particles (cf. section 3.3.2) can be used to minimize communication among processes during the computation of the lubrication matrix, which reduces numerical costs. Schemes which model lubrication as additional force can be included in our method without additional effort, yet they often show non-physical behaviour as most formulations model the repulsive part of lubrication, but not the attractive part.

3.2.1 Stokesian Dynamics

The motion of a spherical colloidal particle α suspended in a fluid can be described by the Langevin equation and the corresponding equation for the angular velocity,

$$\begin{aligned} m_\alpha \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_\alpha &= \boldsymbol{F}^{\mathrm{h}}_\alpha + \boldsymbol{F}^{\mathrm{b}}_\alpha + \boldsymbol{F}^{\mathrm{i}}_\alpha + \boldsymbol{F}^{\mathrm{ext}}_\alpha, \\ \boldsymbol{I}_\alpha \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega}_\alpha &= \boldsymbol{T}^{\mathrm{h}}_\alpha + \boldsymbol{T}^{\mathrm{b}}_\alpha + \boldsymbol{T}^{\mathrm{i}}_\alpha + \boldsymbol{T}^{\mathrm{ext}}_\alpha, \end{aligned} \tag{3.1}$$

where m_α is the mass of particle α, $\boldsymbol{I}_\alpha$ its moment of inertia tensor, $\boldsymbol{U}_\alpha$ the translational velocity and $\boldsymbol{\omega}_\alpha$ the angular velocity. $\boldsymbol{F}^{\mathrm{h}}_\alpha$ is the hydrodynamic force, $\boldsymbol{F}^{\mathrm{b}}_\alpha$ forces arising from Brownian motion, $\boldsymbol{F}^{\mathrm{i}}_\alpha$ interparticle forces, such as the London-van-der-Waals force, and $\boldsymbol{F}^{\mathrm{ext}}_\alpha$ external forces like gravity. $\boldsymbol{T}_\alpha$ denote the corresponding torques acting on the particle. When considering rigid particles the assumption $\boldsymbol{E}_\alpha = \boldsymbol{0}$, where $\boldsymbol{E}_\alpha$ is the rate of strain of particle α, is reasonable. As in [Durlofsky et al., 1987], the SD method is based on the integral representation for the velocity field in Stokes flow with rigid particles,

$$\boldsymbol{u}^{\mathrm{D}}(\boldsymbol{x}) := \boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}^\infty = -\frac{1}{8\pi\mu_f}\sum_{\alpha=1}^{N}\int_{\boldsymbol{\sigma}_\alpha} \mathfrak{G}(\boldsymbol{x}-\boldsymbol{\xi})\boldsymbol{f}_h(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\sigma}(\boldsymbol{\xi}). \tag{3.2}$$

Here, μ_f is the fluid viscosity, N the number of particles, $\boldsymbol{\sigma}_\alpha$ the surface of particle α, $\boldsymbol{f}_h$ the hydrodynamic force density and $\mathfrak{G}$ Green's function for Stokes flow vanishing as $||\boldsymbol{x}||_2 \to \infty$, the so-called Oseen tensor, defined by

$$\mathfrak{G}(\boldsymbol{x}) := \frac{1}{r}\left(\mathbf{Id} + \frac{1}{r^2}\boldsymbol{x}\otimes\boldsymbol{x}\right), \tag{3.3}$$

with $r := ||\boldsymbol{x}||_2$ and the identity matrix $\mathbf{Id}$. We have defined the disturbance velocity $\boldsymbol{u}^{\mathrm{D}}$ as difference of fluid velocity $\boldsymbol{u}$ with particles and the fluid velocity $\boldsymbol{u}^\infty$ without disturbance by any particles. This velocity is generally given by

$$\boldsymbol{u}^\infty(\boldsymbol{x}) := \boldsymbol{U}^\infty + \boldsymbol{\Omega}^\infty \times \boldsymbol{x} + \boldsymbol{E}^\infty \boldsymbol{x},$$

where $\boldsymbol{x}$ is a point in the fluid domain, $\boldsymbol{U}^\infty$ and $\boldsymbol{\Omega}^\infty$ are the translational and angular velocity of the undisturbed flow, $\boldsymbol{E}^\infty$ is the corresponding rate of strain. After a multipole expansion and together with Faxen's laws a linear relation between the particle velocities $\boldsymbol{U}_\alpha$, $\boldsymbol{\omega}_\alpha$, $\alpha = 1, ..., N$, and the hydrodynamic forces, hydrodynamic torques, stresslets $\boldsymbol{S}_\alpha$ and possibly considered higher moments can be found. Considering the first two moments in the multipole expansion [Durlofsky et al., 1987] this relation is given by the mobility problem [Ichiki, 2002]

$$\begin{pmatrix} \boldsymbol{U}_1 - \boldsymbol{U}^\infty \\ \vdots \\ \boldsymbol{U}_N - \boldsymbol{U}^\infty \\ \boldsymbol{\omega}_1 - \boldsymbol{\Omega}^\infty \\ \vdots \\ \boldsymbol{\omega}_N - \boldsymbol{\Omega}^\infty \\ -\boldsymbol{E}^\infty \\ \vdots \\ -\boldsymbol{E}^\infty \end{pmatrix} = -\boldsymbol{\mathcal{M}}^\infty \begin{pmatrix} \boldsymbol{F}_1^{\mathrm{h}} \\ \vdots \\ \boldsymbol{F}_N^{\mathrm{h}} \\ \boldsymbol{T}_1^{\mathrm{h}} \\ \vdots \\ \boldsymbol{T}_N^{\mathrm{h}} \\ \boldsymbol{S}_1^{\mathrm{h}} \\ \vdots \\ \boldsymbol{S}_N^{\mathrm{h}} \end{pmatrix}. \tag{3.4}$$

The grand mobility matrix $\boldsymbol{\mathcal{M}}^\infty \in \mathbb{R}^{11N \times 11N}$ can be decomposed into sub-matrices,

$$\boldsymbol{\mathcal{M}}^\infty = \begin{pmatrix} \boldsymbol{M}_{UF} & \boldsymbol{M}_{UT} & \boldsymbol{M}_{US} \\ \boldsymbol{M}_{\omega F} & \boldsymbol{M}_{\omega T} & \boldsymbol{M}_{\omega S} \\ \boldsymbol{M}_{EF} & \boldsymbol{M}_{ET} & \boldsymbol{M}_{ES} \end{pmatrix}, \tag{3.5}$$

where e.g. the subscript of $\boldsymbol{M}_{UF} \in \mathbb{R}^{3N \times 3N}$ indicates that this sub-matrix links the hydrodynamic force on all N particles to their translational velocity. In fact $\boldsymbol{E}^\infty, \boldsymbol{S}_\alpha^{\mathrm{h}} \in \mathbb{R}^{3 \times 3}$, but due to their symmetry and as they both can be assumed to have zero trace, they can be written as vectors of length 5, so the vectors match the dimension of $\boldsymbol{\mathcal{M}}^{\infty}$. The mobility matrix $\boldsymbol{\mathcal{M}}^{\infty}$ is symmetric and positive definite, where the symmetry can be verified by computing the entries of $\boldsymbol{\mathcal{M}}^{\infty}$. Briggs et al. [Briggs et al., 1991] provide a proof for the positive definiteness for all approximations to the mobility matrix gained by multipole expansions. This is an important property of the matrix and is a great advantage for the numerical treatment of the problem. Durlofsky et al. [Durlofsky et al., 1987] calculated the entries of the mobility matrix of a system of monodisperse spheres, i.e. spheres of the same radius, using scalar mobility functions as introduced by Jeffrey and Onishi [Jeffrey and Onishi, 1984]. The exact form of the entries of the mobility matrix $\boldsymbol{\mathcal{M}}^{\infty}$ for polydisperse systems is given in 3.5. Solving (3.4) formally for the hydrodynamic forces, torques and stresslets and plugging those into the Langevin equations (3.1) yields a system of nonlinear second order differential equations for the particle positions. Nonlinear, because the mobility matrix depends on the latter and its entries are nonlinear functions. We solve this system by employing a parallel implementation of an adaptive (4,5)-Runge–Kutta–Fehlberg method [Fehlberg, 1970] which yields good results, as shown in [Bülow et al., 2014]. The linear system (3.4) can be solved efficiently e.g. by a conjugate gradient method, which does not require the storage of all matrix entries. This is an advantage especially for large problems, where $N \gg 1000$. In order to explain our new method we restrict ourselves to the F-version of the SD method [Durlofsky et al., 1987], where $\boldsymbol{\mathcal{M}}^{\infty} = \boldsymbol{M}_{UF}$. An extension to the FT- and FTS-version or possibly higher versions of the SD method [Ichiki, 2002] can be done in an analogous way, which we explain at the end of section 3.2.2. We have implemented the SSD method in the F-, FT- and in the FTS-version. To keep this work somewhat short and focused on the method itself we restrict ourselves to showing most results only for the FT-version. As one can see in subsection 3.4.4, the FTS-version with and without lubrication correction scheme shows similar performance. But it is less economic when it comes to CPU time and memory.

3.2.2 Scalable Stokesian Dynamics

To model and simulate gravitational N-body problems in stellar dynamics a technique of hierarchical subdivision of space into cells was introduced by Barnes and Hut [Barnes and Hut, 1986]. For each cell all bodies within are combined and the cell is divided further into sub-cells. Then the interactions between single bodies can be replaced by cell interactions, if the resulting error is tolerable. Even though the long-range interactions in astronomy and hydrodynamics are of different nature, they share the same asymptotics. So we utilize this approach for the development of an accelerated method for the solution of the mobility problem, where the bodies are particles. From the derivation of the mobility problem (3.4) it becomes clear that the hydrodynamic force acting on particle α arises from the disturbance velocity caused by all other particles. If we think of a particle β far away from a considered particle α, slightly altering the position of particle β should only have little effect on the disturbance velocity $\boldsymbol{u}^{\mathrm{D}}$ for points close to the center $\boldsymbol{x}_\alpha$ of particle α. As the mobility matrix $\boldsymbol{\mathcal{M}}^\infty$ emerges directly from the disturbance velocity, we transfer the above heuristic to the matrix representation. Assume that there is a point $\boldsymbol{x}_{\overline{\beta}}$ such that $\boldsymbol{x}_{\beta_j} - \boldsymbol{x}_{\alpha_i} \approx \boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\alpha_i}$ for two given groups of particles, $\mathcal{A} := \{\alpha_1, \ldots, \alpha_m\}$ and $\mathcal{B} := \{\beta_1, \ldots, \beta_n\}$. When calculating $\boldsymbol{\mathcal{M}}^\infty$ we can therefore replace the distance vector $\boldsymbol{x}_{\beta_j} - \boldsymbol{x}_{\alpha_i}$ for such groups by the respective averaged distance vector $\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\alpha_i}$. We define the resulting matrix

$$\overline{\boldsymbol{\mathcal{M}}}^\infty = \begin{pmatrix} \overline{\boldsymbol{M}}_{UF} & \overline{\boldsymbol{M}}_{UT} & \overline{\boldsymbol{M}}_{US} \\ \overline{\boldsymbol{M}}_{\omega F} & \overline{\boldsymbol{M}}_{\omega T} & \overline{\boldsymbol{M}}_{\omega S} \\ \overline{\boldsymbol{M}}_{EF} & \overline{\boldsymbol{M}}_{ET} & \overline{\boldsymbol{M}}_{ES} \end{pmatrix}, \tag{3.6}$$

and call it simplified mobility matrix. In each submatrix, the $(\mathcal{A}, \mathcal{B})$-block will have altered entries compared to the original matrix. To preserve the symmetry of the mobility matrix, the entries of the $(\mathcal{B}, \mathcal{A})$-block have to be altered accordingly. E.g. the first submatrix $\overline{\boldsymbol{M}}_{UF}$ has the form

$$\overline{\boldsymbol{M}}_{UF} = \begin{pmatrix} & & & & & & \vdots & & \vdots & \\ & & & & & \cdots & \overline{\boldsymbol{a}}_{\alpha_1\beta_1} & \cdots & \overline{\boldsymbol{a}}_{\alpha_1\beta_n} & \cdots \\ & & \ddots & & & & \vdots & & \vdots & \\ & & & & & \cdots & \overline{\boldsymbol{a}}_{\alpha_m\beta_1} & \cdots & \overline{\boldsymbol{a}}_{\alpha_m\beta_n} & \cdots \\ & & & & & & \vdots & & \vdots & \\ & \vdots & & \vdots & & & & & & \\ \cdots & \overline{\boldsymbol{a}}_{\beta_1\alpha_1} & \cdots & \overline{\boldsymbol{a}}_{\beta_1\alpha_m} & \cdots & & & & & \\ & \vdots & & \vdots & & & & \ddots & & \\ \cdots & \overline{\boldsymbol{a}}_{\beta_n\alpha_1} & \cdots & \overline{\boldsymbol{a}}_{\beta_n\alpha_m} & \cdots & & & & & \\ & \vdots & & \vdots & & & & & & \end{pmatrix}.$$

In the following we will show in detail how this submatrix is approximated block-wise and use it as prototype for the other submatrices of the mobility matrix. The entries are called mobility functions, which depend not only on the particle distance but also on the radii of the particles (cf. equations (3.16)–(3.21)). In the original Barnes–Hut algorithm not only the distances between different bodies are approximated, but also their radii. In our method the radii a_{α_i}, a_{β_j} of the particles are not approximated. The exact radii can be kept without loss of efficiency by extracting them from the original matrix. To achieve this, we split the matrix block into three parts. The first part for the summands not depending on the radii and one part each containing the radii of the particles in $\mathcal{A}$ and $\mathcal{B}$ respectively. Using $\mathbf{Id}_3$ for the unit matrix in $\mathbb{R}^{3\times 3}$ we get

$$
\begin{aligned}
(\overline{\boldsymbol{M}}_{UF})_{\mathcal{A}\mathcal{B}} = & \begin{pmatrix} \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & \cdots & \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} \\ \vdots & & \vdots \\ \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} & \cdots & \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \\
& + \begin{pmatrix} \frac{a_{\alpha_1}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & \cdots & \frac{a_{\alpha_1}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} \\ \vdots & & \vdots \\ \frac{a_{\alpha_m}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} & \cdots & \frac{a_{\alpha_m}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \\
& + \begin{pmatrix} \frac{a_{\beta_1}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & \cdots & \frac{a_{\beta_n}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} \\ \vdots & & \vdots \\ \frac{a_{\beta_1}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} & \cdots & \frac{a_{\beta_n}^2}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \\
= & \begin{pmatrix} \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & \cdots & \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} \\ \vdots & & \vdots \\ \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} & \cdots & \frac{3}{4}\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \\
& + \begin{pmatrix} a_{\alpha_1}^2\mathbf{Id}_3 & & 0 \\ & \ddots & \\ 0 & & a_{\alpha_m}^2\mathbf{Id}_3 \end{pmatrix} \begin{pmatrix} \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & \cdots & \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} \\ \vdots & & \vdots \\ \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} & \cdots & \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \\
& + \begin{pmatrix} \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & \cdots & \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} \\ \vdots & & \vdots \\ \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} & \cdots & \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \begin{pmatrix} a_{\beta_1}^2\mathbf{Id}_3 & & 0 \\ & \ddots & \\ 0 & & a_{\beta_n}^2\mathbf{Id}_3 \end{pmatrix},
\end{aligned}
$$

with the abbreviation $\boldsymbol{\mathfrak{G}}^{\alpha\beta} = \boldsymbol{\mathfrak{G}}(\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha)$. Having the radii extracted, all columns in each of the three dense matrices are equal. Thus the Oseen tensor terms can be extracted as well. We obtain

$$
\begin{aligned}
(\overline{\boldsymbol{M}}_{UF})_{\mathcal{AB}} = \frac{3}{4} & \begin{pmatrix} \boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \boldsymbol{J}_{3N} \\
& + \frac{1}{6} \begin{pmatrix} a_{\alpha_1}^2 \mathbf{Id}_3 & & 0 \\ & \ddots & \\ 0 & & a_{\alpha_m}^2 \mathbf{Id}_3 \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & & 0 \\ & \ddots & \\ 0 & & \Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \boldsymbol{J}_{3N} \\
& + \frac{1}{6} \begin{pmatrix} \Delta\boldsymbol{\mathfrak{G}}^{\alpha_1\overline{\beta}} & & 0 \\ & \ddots & \\ 0 & & \Delta\boldsymbol{\mathfrak{G}}^{\alpha_m\overline{\beta}} \end{pmatrix} \boldsymbol{J}_{3N} \begin{pmatrix} a_{\beta_1}^2 \mathbf{Id}_3 & & 0 \\ & \ddots & \\ 0 & & a_{\beta_m}^2 \mathbf{Id}_3 \end{pmatrix},
\end{aligned}
\tag{3.7}
$$

where we use the matrix of ones $\boldsymbol{J}_{3N} \in \mathbb{R}^{3N\times 3N}$,

$$
\boldsymbol{J}_{3N} := \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}.
$$

For the block $(\overline{\boldsymbol{M}}_{UF})_{\mathcal{B}\mathcal{A}}$ we gain analogously

$$
\begin{aligned}
(\overline{\boldsymbol{M}}_{UF})_{\mathcal{B}\mathcal{A}} = \frac{3}{4}\boldsymbol{J}_{3N} & \begin{pmatrix} \boldsymbol{\mathfrak{G}}^{\overline{\beta}\alpha_1} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{\mathfrak{G}}^{\overline{\beta}\alpha_m} \end{pmatrix} \\
& + \frac{1}{6} \begin{pmatrix} a_{\beta_1}^2 \mathbf{Id}_3 & & 0 \\ & \ddots & \\ 0 & & a_{\beta_m}^2 \mathbf{Id}_3 \end{pmatrix} \boldsymbol{J}_{3N} \begin{pmatrix} \Delta\boldsymbol{\mathfrak{G}}^{\overline{\beta}\alpha_1} & & 0 \\ & \ddots & \\ 0 & & \Delta\boldsymbol{\mathfrak{G}}^{\overline{\beta}\alpha_m} \end{pmatrix} \\
& + \frac{1}{6}\boldsymbol{J}_{3N} \begin{pmatrix} \Delta\boldsymbol{\mathfrak{G}}^{\overline{\beta}\alpha_1} & & 0 \\ & \ddots & \\ 0 & & \Delta\boldsymbol{\mathfrak{G}}^{\overline{\beta}\alpha_m} \end{pmatrix} \begin{pmatrix} a_{\alpha_1}^2 \mathbf{Id}_3 & & 0 \\ & \ddots & \\ 0 & & a_{\alpha_m}^2 \mathbf{Id}_3 \end{pmatrix} .
\end{aligned}
\tag{3.8}
$$

For the considered blocks the matrix is now split up in diagonal matrices and matrices of ones. Hence the matrix-vector products $\boldsymbol{U}_{\mathcal{A}}^{\mathcal{B}} = (\overline{\boldsymbol{M}}_{UF})_{\mathcal{A}\mathcal{B}}\boldsymbol{F}_{\mathcal{B}}$ and $\boldsymbol{U}_{\mathcal{B}}^{\mathcal{A}} = (\overline{\boldsymbol{M}}_{UF})_{\mathcal{B}\mathcal{A}}\boldsymbol{F}_{\mathcal{A}}$ can be computed with reduced effort. We refer to the vectors

$$
\boldsymbol{F}_{\mathcal{A}} = (\boldsymbol{F}_{\alpha_1}, \ldots, \boldsymbol{F}_{\alpha_m})^{\mathrm{T}} \in \mathbb{R}^{3m}, \quad \boldsymbol{F}_{\mathcal{B}} = (\boldsymbol{F}_{\beta_1}, \ldots, \boldsymbol{F}_{\beta_n})^{\mathrm{T}} \in \mathbb{R}^{3n}
$$

and

$$
\boldsymbol{U}_{\mathcal{A}}^{\mathcal{B}} = \left(\boldsymbol{U}_{\alpha_1}^{\mathcal{B}}, \ldots, \boldsymbol{U}_{\alpha_m}^{\mathcal{B}}\right)^{\mathrm{T}} \in \mathbb{R}^{3m}, \quad \boldsymbol{U}_{\mathcal{B}}^{\mathcal{A}} = \left(\boldsymbol{U}_{\beta_1}^{\mathcal{A}}, \ldots, \boldsymbol{U}_{\beta_n}^{\mathcal{A}}\right)^{\mathrm{T}} \in \mathbb{R}^{3n}
$$

as forces and velocities. The superscripts to the velocities shall denote that the result is the part of the velocity caused by the respective forces.

The overall velocity is then calculated by summing up those parts for all force blocks. Equation (3.7) yields the elements of $\boldsymbol{U}_{\mathcal{A}}^{\mathcal{B}}$,

$$\begin{aligned} \boldsymbol{U}_{\alpha_i}^{\mathcal{B}} =& \big(\frac{3}{4}\mathfrak{G}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\alpha_i}) + \frac{1}{6}a_{\alpha_i}^2 \Delta\mathfrak{G}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\alpha_i})\big)\boldsymbol{\mathcal{S}}_{\mathcal{B}}^{F} \\ &+ \frac{1}{6}\Delta\mathfrak{G}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\alpha_i})\boldsymbol{\mathcal{S}}_{\mathcal{B}}^{a^2F}. \end{aligned} \tag{3.9}$$

The summed force vectors $\boldsymbol{\mathcal{S}}_{\mathcal{B}}^{F}$, $\boldsymbol{\mathcal{S}}_{\mathcal{B}}^{a^2F} \in \mathbb{R}^3$ are defined as

$$\boldsymbol{\mathcal{S}}_{\mathcal{B}}^{F} := \sum_{j=1}^{n} \boldsymbol{F}_{\beta_j} \ , \quad \boldsymbol{\mathcal{S}}_{\mathcal{B}}^{a^2F} := \sum_{j=1}^{n} a_{\beta_j}^2 \boldsymbol{F}_{\beta_j} \ .$$

They are the same for all α_i and can be computed at linear cost. From equation (3.8) we gain for the elements of $\boldsymbol{U}_{\mathcal{B}}^{\mathcal{A}}$

$$\begin{aligned} \boldsymbol{U}_{\beta_j}^{\mathcal{A}} =& \sum_{i=1}^{m}\big(\frac{3}{4}\mathfrak{G}(\boldsymbol{x}_{\alpha_i} - \boldsymbol{x}_{\overline{\beta}}) + \frac{1}{6}a_{\alpha_i}^2 \Delta\mathfrak{G}(\boldsymbol{x}_{\alpha_i} - \boldsymbol{x}_{\overline{\beta}})\big)\boldsymbol{F}_{\alpha_i} \\ &+ a_{\beta_j}^2 \sum_{i=1}^{m} \frac{1}{6}\Delta\mathfrak{G}(\boldsymbol{x}_{\alpha_i} - \boldsymbol{x}_{\overline{\beta}})\boldsymbol{F}_{\alpha_i}. \end{aligned} \tag{3.10}$$

Note, that the sums in (3.10) are independent of β_j. So after a computation of the sums, each component of the velocity vector can be computed with the effort of one multiplication and one addition.

If the initial approximation $\boldsymbol{x}_{\beta_j} - \boldsymbol{x}_{\alpha_i} \approx \boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\alpha_i}$ also applies vice versa, i.e. there is a point $\boldsymbol{x}_{\overline{\alpha}}$ such that $\boldsymbol{x}_{\beta_j} - \boldsymbol{x}_{\alpha_i} \approx \boldsymbol{x}_{\beta_j} - \boldsymbol{x}_{\overline{\alpha}}$, we have $\boldsymbol{x}_{\beta_j} - \boldsymbol{x}_{\alpha_i} \approx \boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\overline{\alpha}}$ for all α_i, β_j, $(i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\}$. Then the Oseen tensor is constant for all pairs and the calculation costs are reduced even further. We then have

$$\begin{aligned}\boldsymbol{U}^{\mathcal{B}}_{\alpha_i} =& \frac{3}{4}\boldsymbol{\mathfrak{G}}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\overline{\alpha}})\boldsymbol{\mathcal{S}}^{F}_{\mathcal{B}} + \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\overline{\alpha}})\boldsymbol{\mathcal{S}}^{a^2F}_{\mathcal{B}} \\ &+ a^2_{\alpha_i}\frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\overline{\alpha}})\boldsymbol{\mathcal{S}}^{F}_{\mathcal{B}}, \\ \boldsymbol{U}^{\mathcal{A}}_{\beta_j} =& \frac{3}{4}\boldsymbol{\mathfrak{G}}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\overline{\alpha}})\boldsymbol{\mathcal{S}}^{F}_{\mathcal{A}} + \frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\overline{\alpha}})\boldsymbol{\mathcal{S}}^{a^2F}_{\mathcal{A}} \\ &+ a^2_{\beta_j}\frac{1}{6}\Delta\boldsymbol{\mathfrak{G}}(\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\overline{\alpha}})\boldsymbol{\mathcal{S}}^{F}_{\mathcal{A}}.\end{aligned}$$

The only variable values left in those formulas are a_{α_i} and a_{β_j} respectively. Thus the effort which determines the complexity of the computation is reduced to the computation of the weighted force sums. In higher order versions of the SD method all non-diagonal entries $\boldsymbol{s}_{\alpha\beta}$ of the submatrices of $\boldsymbol{\mathcal{M}}^{\infty}$ can be represented by a sum of the form $\boldsymbol{s}_{\alpha\beta} = \boldsymbol{L}_1(\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha) + a^2_\alpha \boldsymbol{L}_2(\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha) + a^2_\beta \boldsymbol{L}_3(\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha)$, where $\boldsymbol{s}$ stands for any of the submatrices $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{g}, \boldsymbol{h}$ or $\boldsymbol{m}$ (cf. equations (3.16)–(3.21)). The matrices $\boldsymbol{L}_k$, $k = 1, 2, 3$, are independent from particle radii. Therefore the simplifications we made for the submatrix $\boldsymbol{M}_{UF}$ and its entries $\boldsymbol{a}$ can be transferred directly to the other mobility functions in the grand mobility matrix $\boldsymbol{\mathcal{M}}^{\infty}$.

3.2.3 Error analysis

To estimate the error of the above approximation we consider two particles α and β with distance vector $\boldsymbol{r} = \boldsymbol{x}_\beta - \boldsymbol{x}_\alpha$. If the center $\boldsymbol{x}_\beta$ of particle β is shifted by $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \zeta_3]^{\mathrm{T}} \in \mathbb{R}^3$, the disturbed particle distance vector is $\tilde{\boldsymbol{r}} = \boldsymbol{r} + \boldsymbol{\zeta}$. We define $r_{\alpha\beta} := \|\boldsymbol{r}\|_2$, $\boldsymbol{e} := \boldsymbol{r}/r_{\alpha\beta}$ as well as for the disturbed values $\tilde{r}_{\alpha\beta} := \|\tilde{\boldsymbol{r}}\|_2$ and $\tilde{\boldsymbol{e}} := \tilde{\boldsymbol{r}}/\tilde{r}_{\alpha\beta}$. Let $\zeta_{\max} := \max_{i=1}^{3} |\zeta_i|$, then we have $\tilde{\boldsymbol{e}} = \boldsymbol{e} + \boldsymbol{\eta}$ for the disturbed unit vector $\tilde{\boldsymbol{e}}$ with

$$\begin{aligned} |\eta_k| = |\tilde{e}_k - e_k| = \left|\frac{r_k + \zeta_k}{\tilde{r}_{\alpha\beta}} - e_k\right| &= \left|\frac{r_{\alpha\beta}}{\tilde{r}_{\alpha\beta}} e_k - e_k + \frac{\zeta_k}{\tilde{r}_{\alpha\beta}}\right| \\ = \left|e_k \frac{r_{\alpha\beta} - \tilde{r}_{\alpha\beta}}{\tilde{r}_{\alpha\beta}} + \frac{\zeta_k}{\tilde{r}_{\alpha\beta}}\right| &\leq |e_k| \frac{\zeta_{\max}}{\tilde{r}_{\alpha\beta}} + \left|\frac{\zeta_k}{\tilde{r}_{\alpha\beta}}\right| \leq 2 \frac{\zeta_{\max}}{\tilde{r}_{\alpha\beta}} \end{aligned}$$

for the entries η_k and $r_k, k \in \{1, 2, 3\}$, of $\boldsymbol{\eta}, \boldsymbol{r} \in \mathbb{R}^3$. Thus the componentwise error in the Oseen tensor can be estimated as follows:

$$\begin{aligned} |\mathfrak{G}_{ij}(\boldsymbol{r}) - \mathfrak{G}_{ij}(\tilde{\boldsymbol{r}})| =& \left|\frac{\delta_{ij}}{r_{\alpha\beta}} + \frac{r_i r_j}{r_{\alpha\beta}^3} - \frac{\delta_{ij}}{\tilde{r}_{\alpha\beta}} - \frac{\tilde{r}_i \tilde{r}_j}{\tilde{r}_{\alpha\beta}^3}\right| \\ =& \left|\frac{\delta_{ij} + e_i e_j}{r_{\alpha\beta}} - \frac{\delta_{ij} + \tilde{e}_i \tilde{e}_j}{\tilde{r}_{\alpha\beta}}\right| \\ \leq& \left|\frac{\tilde{r}_{\alpha\beta}(\delta_{ij} + e_i e_j) - \tilde{r}_{\alpha\beta}(\delta_{ij} + \tilde{e}_i \tilde{e}_j)}{\tilde{r}_{\alpha\beta} r_{\alpha\beta}}\right| \\ &+ \left|\frac{\zeta_{\max}(\delta_{ij} + \tilde{e}_i \tilde{e}_j)}{\tilde{r}_{\alpha\beta} r_{\alpha\beta}}\right| \\ =& \left|\frac{e_i e_j - \tilde{e}_i \tilde{e}_j}{r_{\alpha\beta}}\right| + \left|\frac{\zeta_{\max}(\delta_{ij} + \tilde{e}_i \tilde{e}_j)}{\tilde{r}_{\alpha\beta} r_{\alpha\beta}}\right| \\ \overset{(*)}{\leq}& 4 \frac{\zeta_{\max}}{\tilde{r}_{\alpha\beta} r_{\alpha\beta}} + 4 \frac{\zeta_{\max}^2}{\tilde{r}_{\alpha\beta}^2 r_{\alpha\beta}} + 2 \frac{\zeta_{\max}}{\tilde{r}_{\alpha\beta} r_{\alpha\beta}} \\ =& \left(6 + 4 \frac{\zeta_{\max}}{\tilde{r}_{\alpha\beta}}\right) \frac{\zeta_{\max}}{\tilde{r}_{\alpha\beta} r_{\alpha\beta}}. \end{aligned}$$

For the estimate indicated by $(*)$ we utilize

$$\begin{aligned}
|e_i e_j - \tilde{e}_i \tilde{e}_j| &= \frac{1}{2}\Big|(e_i + \tilde{e}_i)(e_j - \tilde{e}_j) + (e_i - \tilde{e}_i)(e_j + \tilde{e}_j)\Big| \\
&= \frac{1}{2}\Big|(2e_i + \eta_i)\eta_j + (2e_j + \eta_j)\eta_i\Big| \\
&= \Big|e_i \eta_j + e_j \eta_i + \eta_i \eta\Big| \\
&\leq 4\frac{\zeta_{\max}}{\tilde{r}_{\alpha\beta}} + 4\frac{\zeta_{\max}^2}{\tilde{r}_{\alpha\beta}^2}.
\end{aligned}$$

Evaluating the rotlet $\boldsymbol{\mathfrak{R}}$ (cf. 3.5) for the altered particle positions leads to a componentwise error of

$$\begin{aligned}
|\boldsymbol{\mathfrak{R}}_{ij}(\boldsymbol{r}) - \boldsymbol{\mathfrak{R}}_{ij}(\tilde{\boldsymbol{r}})| &= \Big|\epsilon_{ijk}\frac{r_k}{r_{\alpha\beta}^3} - \epsilon_{ijk}\frac{\tilde{r}_k}{\tilde{r}_{\alpha\beta}^3}\Big| = \Big|\frac{e_k}{r_{\alpha\beta}^2} - \frac{\tilde{e}_k}{\tilde{r}_{\alpha\beta}^2}\Big| \\
&\leq \Big|\frac{r_{\alpha\beta}^2 e_k - r_{\alpha\beta}^2 \tilde{e}_k}{\tilde{r}_{\alpha\beta}^2 r_{\alpha\beta}^2}\Big| + \Big|\frac{2r_{\alpha\beta}\zeta_{\max}\tilde{e}_k}{\tilde{r}_{\alpha\beta}^2 r_{\alpha\beta}^2}\Big| + \Big|\frac{\zeta_{\max}^2 \tilde{e}_k}{\tilde{r}_{\alpha\beta}^2 r_{\alpha\beta}^2}\Big| \\
&\leq \frac{2\zeta_{\max}}{\tilde{r}_{\alpha\beta}^3} + \frac{2\zeta_{\max}\tilde{e}_k}{\tilde{r}_{\alpha\beta}^2 r_{\alpha\beta}} + \frac{2\zeta_{\max}^2 \tilde{e}_k}{\tilde{r}_{\alpha\beta}^2 r_{\alpha\beta}^2}
\end{aligned}$$

In 3.5 we give all mobility functions for the FTS-version of the Stokesian dynamics method applied to a polydisperse particle system. The mobility functions are the entries of $\boldsymbol{\mathcal{M}}^\infty$ and are compositions of the Oseen tensor, the rotlet and their derivatives. Since we gain additional factors of $1/r_{\alpha\beta}$ in the derivatives (cf. 3.6) and we imply $r_{\alpha\beta} \gg 1$ we obtain the upper bound

$$\mathrm{err}_{\alpha\beta} < C(\|\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha\|_2)\frac{\|\tilde{\boldsymbol{x}}_\beta - \boldsymbol{x}_\beta\|_2}{\|\tilde{\boldsymbol{x}}_\beta - \boldsymbol{x}_\alpha\|_2}, \tag{3.11}$$

for the error in each disturbed mobility function for particles α and β, where the factor $C(\xi) = \mathcal{O}\Big(\dfrac{1}{\xi}\Big)$ depends on the undisturbed particle distance.

3.2.4 Grouping particles

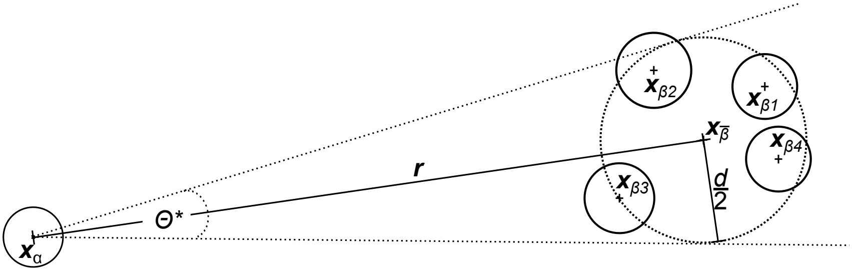

Figure 3.1: Grouping of particles proposed for the SSD method.

For the computation of the simplified mobility matrix $\overline{\boldsymbol{\mathcal{M}}}^\infty$ for a group of particles $\mathcal{B} = \{\beta_1, \ldots, \beta_n\}$, the vector $\boldsymbol{x}_{\overline{\beta}}$ has to be determined. Following Barnes and Hut, we choose the center of mass of a group as substitutionary center. As shown in figure 3.1 we define

$$\boldsymbol{x}_{\overline{\beta}} := \frac{\sum_{j=1}^n a_{\beta_j}^3 \boldsymbol{x}_{\beta_j}}{\sum_{j=1}^n a_{\beta_j}^3}$$

and the group diameter

$$d := 2 \max_{j=1}^{n} \|\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\beta_j}\|_2.$$

Considering the group's effect on a particle α, we have to set up a criterion to decide whether the averaged matrix block $\overline{\boldsymbol{\mathcal{M}}}^\infty_{\alpha\mathcal{B}}$ or no approximation to the mobility matrix is used. The error estimate (3.11) for the particle β_k furthest from the group's center states

$$\mathrm{err}_{\alpha\beta_k} < C(\|\boldsymbol{x}_{\beta_k} - \boldsymbol{x}_\alpha\|_2) \frac{\|\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_{\beta_k}\|_2}{\|\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_\alpha\|_2} \leq C(r-d)\frac{d}{r},$$

where r is the distance between the group's center and the center of particle α. Therewith the error $\mathrm{err}_{\alpha\mathcal{B}}$ arising from considering the whole group $\mathcal{B}$ instead of every single particle $\beta_j \in \mathcal{B}$, $j = 1, \ldots, n$, is

$$\mathrm{err}_{\alpha\mathcal{B}} < C_{\mathcal{B}}(r)\frac{d}{r},$$

with the factor $C_{\mathcal{B}}$ depending on the constellation of the particles in $\mathcal{B}$. This backs up the approach of Barnes and Hut to use the simplified method, if

$$\tan(\theta^*) = \frac{d}{r} < \theta,$$

for a given θ. As this criterion leads to larger errors when the center of mass is near the edge of the group Salmon and Warren [1994], we use a modified criterion

$$\frac{d}{r - \delta} < \theta, \tag{3.12}$$

where $\delta := \|\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}^{\circ}_{\mathcal{B}}\|_2$ is the distance from the group's center of mass to the center $\boldsymbol{x}^{\circ}_{\mathcal{B}}$ of the circumscribed sphere Dubinski [1996].

3.3 Implementation

3.3.1 Sorting algorithm

In order to achieve an optimal load balance we distribute the particle data equally over the used processes p_i, $i = 1, \ldots, P$. Due to the dynamic nature of the system, particles whose data is stored on the same process do not necessarily have to remain close to each other during a whole simulation, even though initially the data might have been assigned to the different processes according to the spatial distribution of the particles. Proximity of data for particles which are in proximity in the simulated domain is an advantage for the computation of interparticle forces, since introduced cutoff distances can reduce communication costs. Sorted data is not only an advantage for the SSD method, it is crucial for its efficiency. If a group of particles whose data is stored on the same process occupies a smaller region in the simulation domain, the error resulting from grouping is smaller and less data has to be sent between processes, as the simplified method can be applied more often during a simulation. Widespread particle groups only rarely meet the grouping criterion (3.12), which would make the method inefficient. Therefore we have implemented a parallel sorting algorithm that sorts particle data in distributed memory according to the spatial distribution of the particles. To achieve this we use a parallel merge-sort algorithm Sedgewick and Wayne [2011] using recursions. In every recursive step we require the processes to be arranged in groups having $m = 2^n, n \in \mathbb{N}$, members $p_1, \dots, p_m$, with the prerequisite that for processes in the subgroups $p_1 \dots p_{m/2}$ and $p_{m/2+1} \dots p_m$, respectively, the data is already sorted. This requirement is fulfilled by calling the algorithm for each of the two subgroups. Then the data is merged pairwise crossed. This means $p_1 \cup p_m, p_2 \cup p_{m-1}, \dots, p_{m/2} \cup p_{m/2+1}$, where the symbol $\cup$ stands for the merging process. After the merging routine for all values x stored on process p_i and all values y stored on p_j, the condition $x < y$ holds for $i < j$, where "$<$" can be any given ordering relation. In our case it is a comparison of the particle coordinates along a chosen sorting direction. Now every value in the subgroup assigned to p_i is smaller than all values assigned to p_j. Due to the structure of the merge function the data remains sorted locally on each process but is no longer guaranteed to be sorted within the subgroup. Thus the merge sort algorithm is called again for both subgroups. An example for one recursion of the merge sort algorithm is illustrated in figure 3.2. If the algorithm is applied to a number of processes unequal to a power of two, we use phantom processes that store no data. The merge routine will return immediately, when called for such a process. In the bottom level recursion nothing has to be done, as we presume that the local data on a process is always sorted. Therefore the initial step, before starting the recursion with the big group of all processes, is to sort the data process-wise with an arbitrary method, like the C standard quick-sort algorithm. Consider now N elements equally distributed over P processes. As every process stores N/P elements, the initial sorting is done in $\mathcal{O}\big((N/P)\log(N/P)\big)$ op-

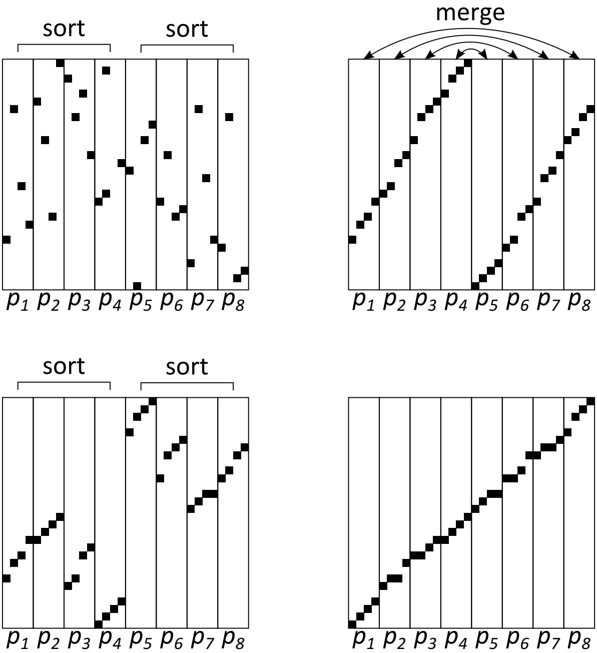
sort
sort
merge
p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8
p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8
sort
sort
p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8
p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8

Figure 3.2: Steps of the merge-sort algorithm for 8 processes p_i: Data-points are displayed as pixels, where the vertical position stands for their value and the horizontal position denotes their location in memory.

erations. The merging requires $O(N/P)$ operations, thus for an arbitrary recursion we have complexity

$$c_{\text{rec}}(P_g) = 2c_{\text{rec}}(\frac{P_g}{2}) + \mathcal{O}(\frac{N}{P}),$$

depending on the group size P_g. In general, the recursive sequence $(b_k)_{k\in\mathbb{N}}$ defined by

$$b_1 = C, \quad b_k := 2b_{k-1} + C \qquad \text{for } k \geq 2$$

with a constant parameter C, has the explicit form

$$b_k = C(2^k - 1).$$

The complexity of the algorithm is defined by this recursion as

$$c_{\text{rec}}(2^k) := b_k,$$

where the constant C is $\mathcal{O}(N/P)$. Therefore the complexity of the sorting is

$$c_{\text{rec}}(P_g) = (P_g - 1)\mathcal{O}(\frac{N}{P}).$$

Since the simulation space has 3 dimensions, we use the above sorting algorithm to construct Cartesian cuboidal domains around the particles, one for each process. In order to have an equal number of particles in each domain, which yields better load balancing, we relinquish giving these domains a fixed size. In every recursive step we split the fluid subdomain of the considered group into two subdomains and repeat this until there is a domain for every process. As cutting plane we choose the median of the positions of the particles in the group sorted along the longest cuboid edge. This means calling the sorting routine $\log_2(P)$ times with a group size of 2^k in the k-th call. Let $n_p := N/P$ the number of particles per process, the overall complexity for sorting c_{sort} is therewith

$$\begin{aligned} c_{\text{sort}}(N,P) &:= \sum_{k=1}^{\log(P)} \big(\mathcal{O}(n_p \log(n_p)) + c_{\text{rec}}(2^k)\big) \\ &= \log(P)\mathcal{O}(n_p \log(n_p)) + (2^{\log(P)+1} - 3)\mathcal{O}(n_p) \qquad (3.13) \\ &= \mathcal{O}\big(n_p \log(P) \log(n_p)\big) + \mathcal{O}(2Pn_p) - \mathcal{O}(3n_p). \end{aligned}$$

An example for 384 particles in a plane, distributed over 16 processes is shown in figure 3.3.

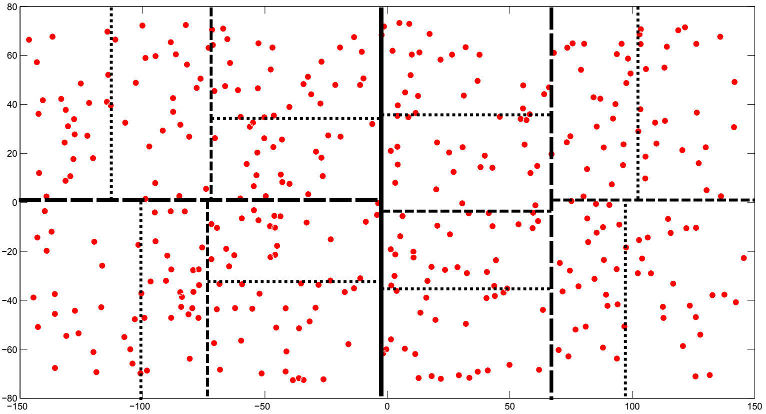

Figure 3.3: Domain decomposition in 2D for 16 processes after application of our parallel merge sort routine. Dots denote particle positions in space.

3.3.2 Tree structure

The SSD method uses the concept of trees, which is often used in computer science. In this context 'node' describes a data entry of the tree. This expression could be easily confused with the node of a computer cluster. Thus in the following we will use the term node exclusively for the nodes of a tree. We will take the sorting algorithm presented in subsection 3.3.1 as starting point to construct these nodes. The original method by Barnes and Hut Barnes and Hut [1986] is based on a hierarchical tree structure, where the N leaf nodes represent one particle each and their parent node contains the combined data of the two children. This is repeated up to the root node, which contains the information of all particles. For the parallel implementation, where the simulation is running on P processes, we cut this tree at the P-th level and use only the P remaining sub trees, one for each process. This way we waive the possibility to form nodes of more particles than stored on one process, in favour of a better fit to the parallel layout and reduced computational costs due to communication. In the grouping process, as illustrated in figure 3.4, we initially use the parallel merge sort routine described in subsection 3.3.1 to sort the particles according to their positions. Based on the sorted data, each process

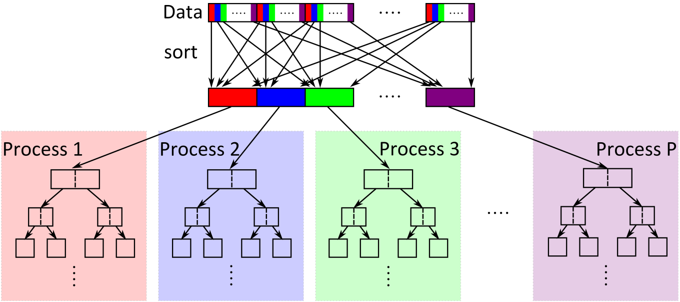

Figure 3.4: Illustration of the implemented grouping process.

builds a binary tree of a given depth. Note that this can be done completely in parallel without any latency due to communication. Adding a new level to the tree is done in three steps for each current leaf node:

a) Construct a cuboid with edges parallel to the Cartesian coordinates, circumscribing all particles on the node. Then determine the direction of the longest edge.

b) Sort the particles according to their scalar component in the determined direction.

c) Create two new sibling nodes, where the left node receives the first half of the sorted particles and the right one stores the rest.

We determine the groups as we successively sort the particles. All nodes, except the leaf nodes, pass the particle information to their children during this procedure and do no longer contain any information. When now the leaf nodes create a child for each of their particles we have a completely empty tree structure in the intermediate levels. This is depicted in figure 3.5 where the white circles denote empty nodes. Storing the new single particle leaves in a contiguous array, we have sorted indices $\alpha_1, \dots, \alpha_{n_p}$ on a process storing n_p particles and we can recognize the particle groups by their index range. Once we know which particles belong to a group, we can compute the matrix-vector product with the SSD method. Therefore we now describe the procedure from the point of view of an arbitrary process. When particles move, their indices α_i and therewith the groups stay the same, but the information relevant for the matrix calculation, the particle

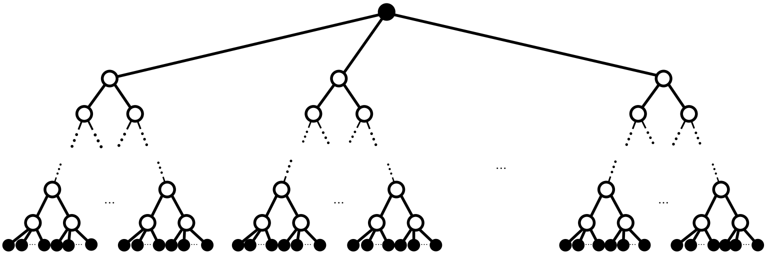

Figure 3.5: Tree structure after sorting and grouping. White circles denote empty nodes. Each leaf contains the data of one particle.

position, changes. Thus every time the matrix-vector product has to be computed we fill the empty nodes of a tree with the current information on the related particle positions. This means we calculate for each node the center and the radius of the smallest sphere surrounding all particle centers of the node's siblings $\alpha_i, \ldots, \alpha_j$. Additionally, we compute the center of mass and the sums of the forces and torques acting on those particles. Initially, a process has only information on the groups in its own tree. Thus before the aspect ratio condition (3.12) can be checked for the particles on this process, the geometrical information of all groups of the other processes must be collected. To reduce this communication we perform the check vice versa. This means a process will not only check if foreign groups match the criterion for its own particles, but it will examine whether there are processes that can make use of its groups. Therefore the data of the circumscribing spheres for the root node groups on all processes is gathered and a worst case position $\boldsymbol{x}^*$ is assumed as reference for each of those foreign groups. Worst case means choosing a point for which the aspect ratio assumes its maximum value, namely that point on the surface of the circumscribing sphere that is closest to the considered own group. Subsequently the groups in the tree are checked starting from the root node. If a group matches the desired aspect ratio the respective group is listed and the searched branch of the tree is closed, else the child nodes are tested. This is repeated until each branch is closed. Criterion (3.12) is met at the latest when a leaf node is reached, as they contain only a single particle and thus $d = 0$. We have integrated the described grouping process into the sorting routine, which can be skipped at a given frequency. This gives the possibility to reduce sorting costs in case of a well-known geometry. Note that not only the aspect ratio θ^* (cf. figure 3.1) is decreased when the groups take up less space, but also the likelihood is higher that the distance between a considered particle and a particle group stored on another process is larger than a cutoff distance introduced for possible interparticle forces of non-hydrodynamic nature. This leads to reduced costs for the calculation of these forces, which reduces again overall computational costs.

3.4 Numerical Tests and Discussion

The major target we aimed at during the development of the SSD method was a reduction of the CPU time while maintaining a good accuracy. In our implementation there are three input parameters which determine the level of approximation and the efficiency of the method. The efficiency is mainly determined by the frequency at which the particle data is sorted and assigned to corresponding processes (cf. section 3.3). Another parameter besides the sorting frequency is the maximum allowed ratio of a group's circumscribing sphere diameter to the distance between the center of mass of the sphere and another particle. This ratio used in criterion (3.12) determines whether a node in the constructed tree is taken for the computation of the simplified mobility matrix or a child node is tested and possibly taken. If no particle group meets the criterion, no approximation to the mobility matrix is made for the considered particle and the original SD method is recovered. The third parameter is the depth of the local trees. In this context local means with respect to a single process. So a local tree-depth of one means that either the simplified mobility matrix is computed with the group of particles whose data is stored in the memory of the same process, or no approximation is made and the SSD method collapses into the SD method. We have conducted all simulations using the IC2 supercomputer at our university [ic2, 2013]. In order to solve arising systems of linear equations we make use of the open source library PETSc [Balay et al., 2013]. The time-stepping scheme used to solve for the particle positions is for all shown simulation results our parallel implementation of the embedded (4,5)-Runge–Kutta–Fehlberg method [Bülow et al., 2014].

3.4.1 CPU time

The CPU time strongly depends on the choice of the tree-depth. Therefore we have investigated the effect of this parameter on the CPU time. As reference we have run simulations with the standard SD method. The chosen level of approximation is the FT-version of SD. We have simulated two polydisperse particle clouds of $N = 1000$ particles each, settling in a fluid under the influence of gravity. The particle radii are chosen according to a unimodal lognormal radius distribution with prescribed expected value of dimensionless radius 1.0 and standard deviation 0.065. Figure 3.6 shows the CPU time when using the SSD method with two different maximum tree-depths and when using the standard SD method. For this comparison we have chosen maximum local tree-depths of 1 and 5. We chose these values to show that, in an optimal configuration, at level 1 the CPU time can be halved, and that at a higher level in general more

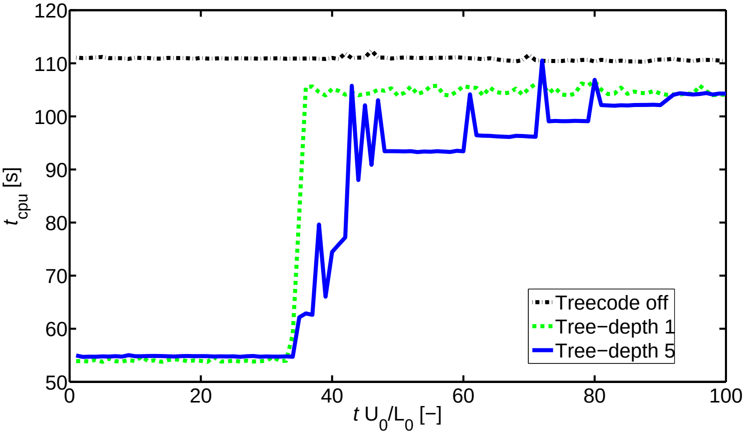

Figure 3.6: CPU time per timestep, t_{CPU}, over dimensionless simulated time for two different tree-depths.

savings in CPU time are possible. As described in [Nitsche and Batchelor, 1997] the clouds settle and lose particles in an emanating tail. After about 35 dimensionless time units, criterion (3.12) is not met anymore for all particles stored in the memory of different processes, which results in an increased CPU time. When a tree-depth of 1 is chosen, the SSD method becomes our parallel implementation of the standard SD method [Bülow et al., 2014] at this point, since the child nodes of the top level node only consist of single particles. The fact that the CPU time is still lower than when the standard SD method is used, arises from the more efficient implementation of our method. For a tree-depth of 5, the CPU time increases first when the highest level in the tree cannot be used anymore. After that it slowly increases as lower and lower levels have to be used, up to the point where the curve coincides with the curve displaying the CPU time when only a tree depth of 1 is used. We have triggered the sudden increase in CPU time by choosing the parameter for criterion (3.12) as 1. Since we have set the sorting frequency during the simulation to zero, particle data initially assigned to one process remains in the memory of this process. Using two processes in total, we have assigned the data of one particle cloud to one process. The effect of this choice is displayed in figure 3.7. During settling the particle clouds lose particles, which form a tail behind the cloud. Once this tail is long enough, criterion (3.12) is not met anymore for the whole group of particles on one process and the next level in the tree is used, resulting in a higher CPU time. If the top-level nodes can be used, the gain in speed is severe. In our example the CPU time is halved compared to the standard SD method (cf. figure 3.6).

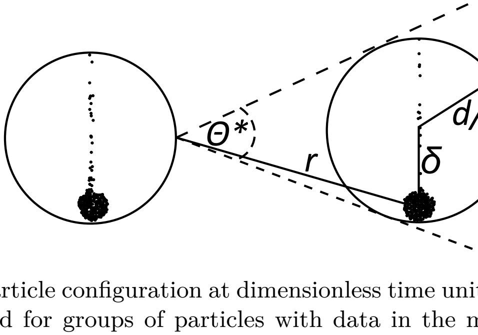

Figure 3.7: Particle configuration at dimensionless time unit 35, when criterion (3.12) is reached for groups of particles with data in the memory of different processes.

3.4.2 Accuracy

For a validation of the SSD method we have run several test runs using a particle configuration similar to the configuration shown in figure 3.1. We have placed a spherical cloud of $N_\mathrm{c} \in \{100, 500, 1000, 2000\}$ particles with a solid volume fraction of $\phi = 0.2$ in the initial cloud domain at a given distance r from a single particle α. Since we investigate the error arising from the application of the SSD method, we only consider hydrodynamic interactions among particles and no other interparticle interactions. The driving force of the particle motion is gravity. We have run the simulations once using the SSD method and once using the standard SD method, both in the FT-version. The approximation of the particle distances when using the SSD method induces a deviation in the settling velocity of the single particle α at distance r from the particle cloud. The entries of the mobility matrix arising from interactions among particles inside the cloud are not approximated, since the only particle for which the approximation criterion (3.12) is met, is particle α. The critical ratio is chosen as 1, the maximum tree-depth as well. This is because for the given setup a critical ratio of 1 ensures that the particles in the cloud form the group represented by the top-level node in the sub-tree of the corresponding process and thus a tree-depth of 1, which induces the largest error, suffices. Furthermore, for this test we have given up load balance and assigned the data of particles in the cloud to one process and the data of the single particle to another. To ensure the desired grouping, the sorting frequency is set to zero. For particle-to-cloud distances r in the range from 10 to 5000 the relative deviation of the settling velocity $U_{z,\alpha}^{\mathrm{SSD}}$ from the corresponding value when using the standard SD method, $U_{z,\alpha}^{\mathrm{SD}}$, is depicted in figure 3.8. Lengths have been nondimensionalized by the expected value L_0 of the particle radius distribution, velocities by the absolute value of the corresponding Stokes velocity U_0. Figure 3.8 shows that the error of the SSD method is bounded. When the ratio d/r used for criterion (3.12) becomes too large, e.g. for $N_c = 2000$ and $r = 20$, the next lower node in the constructed tree is taken. Since for the numerical experiments shown in this section the tree depth is chosen as 1, this means that then the leaves of the tree are taken for the computation, which is equal to using the standard SD method. Due to the fact that we want to preserve the symmetry of the mobility matrix, not only the entries of $\boldsymbol{\mathcal{M}}^{\infty}$ which yield

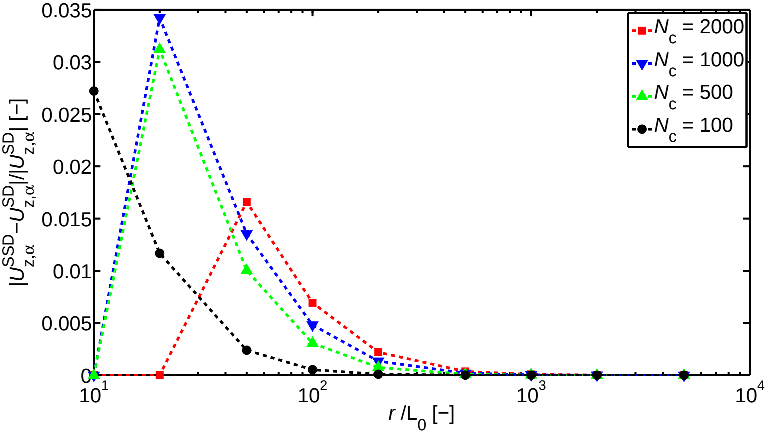

Figure 3.8: Relative deviation of the settling velocity $U^{\mathrm{SSD}}_{z,\alpha}$ of a single particle α from the corresponding value $U^{\mathrm{SD}}_{z,\alpha}$, when the standard SD method is used.

the velocity component $U^{\mathrm{SSD}}_{z,\alpha}$ are altered, but also the entries which are used to compute the velocities of other particles β. Instead of the exact distance vector $\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha$, the altered distance vector $\boldsymbol{x}_{\overline{\beta}} - \boldsymbol{x}_\alpha$ is used. The effect can be seen in the relative deviation of the overall settling velocity of all particles without particle α. We define this deviation as

$$\mathrm{err}_{\mathrm{rel}}\left(\langle U_{z,\beta\neq\alpha}\rangle\right) := \sum_{\beta\neq\alpha} \frac{|U^{\mathrm{SSD}}_{z,\beta} - U^{\mathrm{SD}}_{z,\beta}|}{N|U^{\mathrm{SD}}_{z,\beta}|}. \tag{3.14}$$

Figure 3.9 shows the value of (3.14) for the investigated setup. The overall deviation is low and decreasing with an increasing number of particles inside the cloud. Furthermore, it is bounded for the same reason as the single-particle deviation shown in figure 3.8 is.

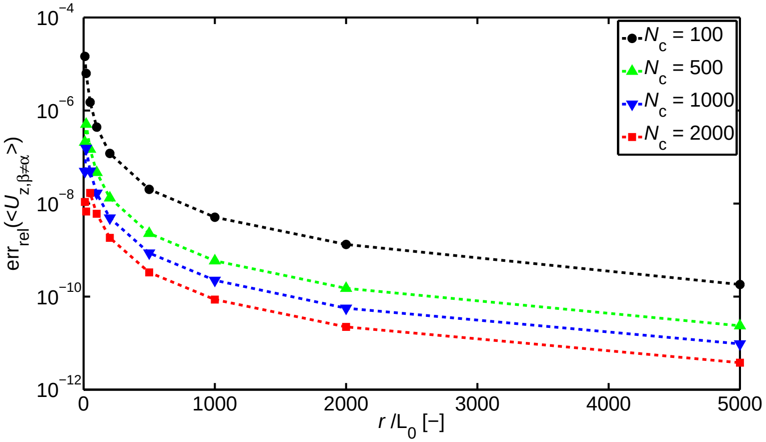

Figure 3.9: Relative deviation of the mean settling velocity of all particles inside a cloud of particles, $\langle U^{\mathrm{SSD}}_{z,\beta\neq\alpha}\rangle$, from the corresponding value $\langle U^{\mathrm{SD}}_{z,\beta\neq\alpha}\rangle$, when the standard SD method is used.

3.4.3 On the choice of parameters and the resulting performance

The performance of the SSD method may be measured in mean CPU time per timestep, $\langle t_{\mathrm{CPU}}\rangle$, relative deviation in the mean particle velocity from the standard SD method in the 2-norm, $\mathrm{err}_{\mathrm{rel}}\left(\langle \boldsymbol{U}\rangle\right)$, and depends on the number of used processes P. In [Bülow et al., 2014] we have shown that a parallel implementation of the standard SD method itself, and of simpler versions considering only mobility interactions on the F, FT or FTS-level, allow for a simulation of a large number of particles in a substantially shorter time with growing P. This advantage should not be lost when the SSD method is applied. The performance of the SSD method depends not only on P, but also on the angle θ used in criterion (3.12) and the depth of the local trees, which we call d_{tree}. The choice of the latter two parameters, θ and d_{tree}, is not very intuitive. In order to give the reader a better understanding of the effect of different choices for θ and d_{tree} on CPU time and accuracy, especially in combination with the number of used processes P, we have run numerical tests with a varying initial solid volume fraction ϕ in a cuboidal particle domain, in which monodisperse particles are placed at the nodes of an equidistant cubic grid. While for subsections 3.4.1 and 3.4.2 a particle configuration has been chosen which is ideal for the SSD method, an equidistant cubic grid is less than favourable for the SSD method. It this configuration, criterion (3.12) is not met very often for particle groups on different processes and the group diameters scale with inter-group distances. Thus this setup is ideal to show that even for a 'bad configuration' the SSD method performs well with respect to CPU time and resulting deviation $\mathrm{err}_{\mathrm{rel}}\left(\langle \boldsymbol{U} \rangle\right)$. The investigated parameters were all combinations of $\phi = \{5\mathrm{e}{-5}, 5\mathrm{e}{-3}, 0.25\}$, $\theta = \{1, 0.1, 0.01, 0.001\}$, $d_{\text{tree}} = \{1, 2, 3, 4, 5, 6\}$, and $P = \{32, 64, 128\}$. As references served the respective simulations with our standard parallel SD code [Bülow et al., 2014]. The number of particles was kept fixed at 50653, in order to be in a range in which a change in P still yields a good speedup [Bülow et al., 2014], and to simulate enough particles that the effect of a change in θ and d_{tree} is visible. The chosen level of accuracy is again FT. If not stated otherwise we used a maximum dimensionless timestep width of 0.001 for the time-stepping scheme. To ensure that the SSD method would not lose efficiency due to suboptimal grouping of particles we set the sorting frequency to 1, which means sorting in every of the 10 timesteps per simulation.

In general, the results for all investigated volume fractions ϕ were similar. For angles θ of 0.01 and 0.001 the proposed combination of SD and Barnes-Hut yields a decrease in mean CPU time per timestep, $\langle t_{\text{CPU}} \rangle$, of up to about 10% for $P = 32$, with decreasing gain in speed for an increasing number of processes. The resulting relative deviation in the mean particle velocity from the standard SD method, $\mathrm{err}_{\mathrm{rel}}\left(\langle \boldsymbol{U} \rangle\right)$, was negligible for all d_{tree} at a value around machine accuracy for these θ. At an angle of $\theta = 0.1$ we observed slight differences between the investigated ϕ. For most tree depths d_{tree} the mean CPU time was again between 0% and 10% lower than for the original SD method, again with no observable deviation in the mean velocity. For $P = 64$, $d_{\text{tree}} = 6$ and for $P = 128$, $d_{\text{tree}} = \{5, 6\}$ the relative deviation $\mathrm{err}_{\mathrm{rel}}\left(\langle \boldsymbol{U} \rangle\right)$ and, depending on ϕ, the mean CPU time were higher. Table 3.1 shows a comparison of the respective values. As one can see, the deviation $\mathrm{err}_{\mathrm{rel}}\left(\langle \boldsymbol{U} \rangle\right)$ is comparable for all investigated ϕ, and increasing with P and d_{tree}. Interesting is the extreme increase in CPU time for increasing ϕ, P and d_{tree}. It is a consequence of a reduction in

	$P = 64$,	$d_{\text{tree}} = 6$	$P = 128$,	$d_{\text{tree}} = 5$	$P = 128$,	$d_{\text{tree}} = 6$
ϕ	$\langle t_{\text{CPU}} \rangle$	$\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$	$\langle t_{\text{CPU}} \rangle$	$\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$	$\langle t_{\text{CPU}} \rangle$	$\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$
5e−5	−7.5%	7e−8	−10%	7e−7	−9%	3e−5
5e−3	−5%	1.6e−7	+2%	1e−6	+84%	3e−5
0.25	+38%	1.3e−7	+115%	9e−7	+464%	2.6e−5

Table 3.1: Increase in mean CPU time per timestep, $\langle t_{\text{CPU}} \rangle$, (first value in each column) and in deviation in the mean particle velocity from the standard SD method, $\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$, (second value in each column) for two different numbers of processes P and local tree depths d_{tree}. For all other parameters, the SSD method performs at least as good as the standard SD method.

timestep width during the simulation, resulting from the adaptive time-stepping scheme we use. The corresponding maximum, minimum and mean timestep widths can be seen in figure 3.10. While for $P = 64$ and $d_{\text{tree}} = 6$ we only have a small timestep reduction in one timestep, for $P = 128$ we see that the mean timestep width is considerably decreasing for increasing ϕ and d_{tree}. For $P = 128$, $d_{\text{tree}} = 6$ and $\phi = 0.25$ the set maximum timestep width of 0.001 is not reached anymore, thus resulting in the 464% increase in mean CPU time $\langle t_{\text{CPU}} \rangle$ for 10 timesteps of width 0.001. This reduction in timestep width is a result from the fact that new grouping of particles between different timesteps produces a right-hand side in equation (3.4) which is considered to be not smooth enough by the time-stepping scheme, resulting in a timestep reduction. We see this effect also in the peaks of the curve for $d_{\text{tree}} = 5$ in figure 3.6, when the method switches to lower levels. If we set the timestep width to a value of 1e−5 instead of 1e−3 the SSD method is always faster than the standard SD method. For $\phi = 0.25$ we get for all d_{tree} a decrease in mean CPU time of 6.5% for $P = 32$, 5.8% for $P = 64$ and $9 - 11\%$ for $P = 128$ instead of the discussed increase in $\langle t_{\text{CPU}} \rangle$. The deviation in the mean settling velocity, $\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$, remains as given in table 3.1 as it is a result of the grouping. As depicted exemplarily for $\phi = 0.25$ in figure 3.11 the mean CPU time for $\theta = 1$ is usually decreasing with increasing local tree depth d_{tree}. Only for $P = 32$ we see a slight increase up to $d_{\text{tree}} = 3$, followed by a shorter mean CPU time for higher tree depths such that for $d_{\text{tree}} = 6$ we still obtain a decrease in $\langle t_{\text{CPU}} \rangle$ of 6%. For $P = 64$ we get a decrease between 18% and 32%, for $P = 128$ of up to 7.5%. Furthermore, we see in figure 3.11

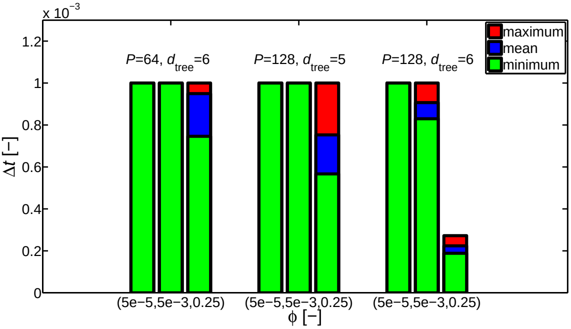

Figure 3.10: Maximum, mean and minimum timestep widths Δt over different volume fractions ϕ for the parameter combinations (P, d_{tree}) for which $\langle t_{\text{CPU}} \rangle$ and $\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$ are shown in table 3.1.

that $\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$ stays below 3.55% for all P and d_{tree}, as in subsection 3.4.2. Figure 3.11 also shows that we needed to decrease the maximum timestep width even further to not get increased CPU time due to timestep reduction. The increase (e.g. for $P = 32$ and $d_{\text{tree}} = 1$) is not as severe as shown in table 3.1, as the timestep width only goes down to $7.9e{-6}$ in one timestep, but it is noticeable. In general, for a more concentrated suspension (i.e. higher ϕ), more processes P and higher local tree depths d_{tree} an adaptive time-stepping scheme such as the applied embedded Runge-Kutta (4,5) method requires smaller timesteps. If the timesteps are chosen small enough, the proposed SSD method produces even for

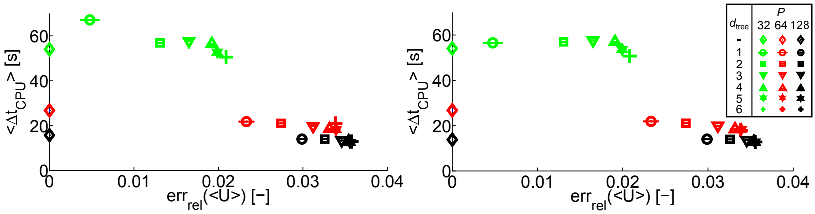

Figure 3.11: Mean CPU time $\langle t_{\text{CPU}} \rangle$ plotted against relative deviation $\text{err}_{\text{rel}}\left(\langle \boldsymbol{U} \rangle\right)$ for $\theta = 1$, $\phi = 0.25$ and $P = 32, 64, 128$. Left: For a maximum timestep width of 1e−5. Right: For a maximum timestep width of 1e−7.

unfavourable particle configurations better results when it comes to CPU time. The resulting error is with less than 4% acceptable and depends on the local tree depth and the angle θ. Because of the requirement of a small timestep, at least for adaptive methods, we propose to use the SSD method only for the simulation of colloidal suspensions. In that case DLVO potentials [Bülow et al., 2014] usually require a small timestep of the same order as in our numerical experiments. As colloidal particles often tend to form clusters, particle configurations as in subsection 3.4.1 are more likely than the one investigated in this section. In that case particle grouping is more effective, the SSD method is faster and the resulting error is smaller.

3.4.4 Speedup for random configurations

In the previous subsections of section 3.4 we have shown the performance of the SSD method on extreme examples: Strongly clustered suspensions and totally ordered unclustered suspensions. In this subsection we examine the performance on examples of suspensions of randomly distributed particles under the action of gravity. The initial positions of the particles are normally distributed random variables with the restriction that all particles are placed in a rectangular domain with side lenghts according to the given initial solid volume fraction and number of particles. Sorting is done in every time step to ensure optimal grouping of particles.
We have conducted speedup tests with initial volume fractions $\phi = 5\text{e}{-5}$, 0.005, 0.25, as in subsection 3.4.3, a fixed timestep width and local tree depth $d_{\text{tree}} = 6$, as well as angle $\theta = 1$. The choice of the latter two parameters is due to the fact that a higher d_{tree} and a larger θ result in lower CPU time (cf. subsection 3.4.3). As shown earlier, the resulting error is bounded. The radius distribution of the investigated particle systems was chosen as lognormal distribution with dimensionless expected value $\mu = 1$ and standard deviation 0.065. Figure 3.12 shows results of the speedup tests for the solution of the mobility problem ($\text{St} \approx 0$) using the FT-approximation for $N = 32768, 65536, 131072$. Most striking is the superlinear speedup compared with the sequential algorithm without Barnes–Hut coupling. This is not surprising, as the local tree depth of 6 allows for relatively small local groups of particles. Together with the

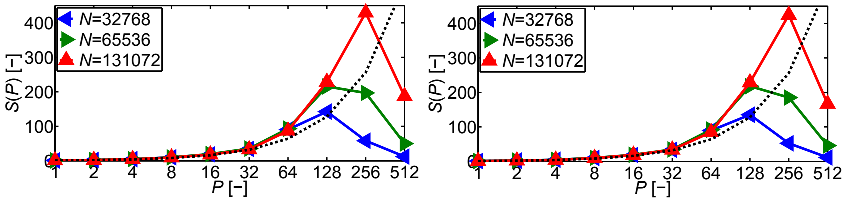

Figure 3.12: Speedup $S(P)$ for the solution of the mobility problem (St $\approx$ 0) using the FT-approximation for the investigated particle systems with random initial configuration and initial volume concentration $\phi = 0.005$ (left) and $\phi = 0.25$ (right) over used number of processes P.

relatively large angle θ, criterion (3.12) is met often for these groups. The higher P, the more local groups fulfill criterion (3.12). This effect is superimposed by the normal speedup of the parallel SD method, which we have presented in [Bülow et al., 2014], yielding the strongly superlinear speedup. A closer examination of the plots in figure 3.12 shows that the resulting speedup only weakly depends on ϕ. In order to rule out effects arising from the random initial placement of particles, we have conducted every simulation three times, each with the same parameters but with different resulting initial particle positions. Figure 3.12 shows the mean values for each parameter set. The deviation in CPU time for the same parameters (including P) due to different initial positions is usually below 6% and has a mean of 1.5%. To investigate the achievable speedup of the presented method for another level and an inclusion of the standard lubrication

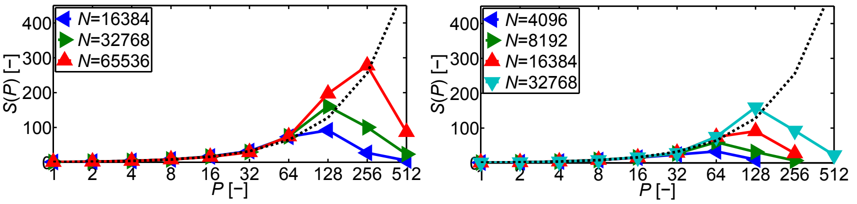

Figure 3.13: Speedup $S(P)$ using the FTS-approximation for the investigated particle systems with random initial configuration and initial volume concentration $\phi = 0.005$ over used number of processes P. Left: Without lubrication correction scheme. Right: With lubrication correction scheme.

scheme [Durlofsky et al., 1987], we have conducted speedup tests using the FTS-approximation once without and another time with standard lubrication correction scheme. The resulting speedup is shown in figure 3.13. The speedup is not as pronounced as for the FT-approximation (cf. figure 3.12). But as for the parallel implementation of the standard SD method [Bülow et al., 2014], a higher number of processes can be used and still give a noticeable speedup (cf. the plots for $N = 65536$ in figure 3.12 and in figure 3.13, left). The standard lubrication scheme [Durlofsky et al., 1987] with lookup table is especially for a polydisperse particle system very memory intensive, therefore we computed the speedup with lubrication correction only for up to $N = 32768$. For this number of particles the speedup is comparable with that of the FTS-approximation without lubrication correction scheme.

3.5 Conclusions

In this work we have presented the scalable Stokesian Dynamics method (SSD), a new method to compute the hydrodynamics in colloidal suspensions. We have coupled the Barnes–Hut algorithm, which is ideal for parallel code execution, with the Stokesian Dynamics method – an extremely efficient method to capture the dynamics of colloidal particulate systems. The computationally most expensive step of the SD method applied to such systems is the computation of the mobility matrix and its multiplication with the force vector. In the worst case, namely when no approximations can be made since particles are densely packed and the simulated system is too small for the application of the introduced simplified method, the usually inherent costs of the N-body problem persist. But this comes with no loss of accuracy since in that case the original SD method is recovered by the SSD method. Yet, due to parallel code execution the original costs of $\mathcal{O}(N^2)$ are reduced to $\mathcal{O}(Pn_p^2)$, where P is the number of processes and n_p the number of particles per process [Bülow et al., 2014]. In a computationally ideal case, where particles form clusters of approximately even size in an otherwise diluted suspension, the mobility matrix becomes constant for blocks of particle data stored in the memory of different processes. Then the main computational effort for the computation of particle velocities reduces to a local computation of radius weighted sums of the acting forces for each process. This yields not only a complexity of $\mathcal{O}(N)$ for a large number of particles and used processes but it also reduces communication between processes to a minimum, giving rise to a very efficient parallel code execution. In practical applications where local clusters are formed but the overall suspension is not dilute, the complexity of the SSD method lies somewhere between the two extremes. Since a parallelized SD method scales very well with P, one can choose P high, which results in a low quadratic term n_p^2. Furthermore, taking a higher number of processes increases the chances of application of the simplified method due to a smaller group size. E.g. in an average case, for a solid volume fraction of 1% in the initial particle domain, the dynamics of one million particles hydrodynamically interacting over ten dimensionless time units under the action of gravity with a simple Hertz force model for particle contacts can be simulated within less than three days. The simulation has been executed on a computer cluster using 2000 cores [ic2, 2013]. This example shows the efficiency of our new parallel method, which can in principle be applied to other problems of similar mathematical structure as well. Unfortunately we were not able to compare our method with the popular accelerated Stokesian Dynamics method (ASD), yet. This has two reasons: First, we have not yet extended our method to the simulation of infinite suspensions. The second and more important reason is that we are currently not aware of any parallel implementation of the ASD method. As we have shown, the SSD method is designed to run on many cores and can yield a very good speedup. It remains future work to compare the SSD method with a possible parallelized ASD method. For results showing the speedup of a parallelized standard SD method, we refer to our work [Bülow et al., 2014].

Appendix to chapter 3: Entries of the mobility matrix for polydisperse systems

The mobility matrix given in (3.5) can be subdivided into

$$
\boldsymbol{\mathcal{M}}^{\infty} = \begin{pmatrix}
\boldsymbol{a}_{11} & \cdots & \boldsymbol{a}_{1N_p} & \boldsymbol{b}_{11} & \cdots & \boldsymbol{b}_{1N_p} & \boldsymbol{g}_{11} & \cdots & \boldsymbol{g}_{1N_p} \\
\vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
\boldsymbol{a}_{N_p1} & \cdots & \boldsymbol{a}_{N_pN_p} & \boldsymbol{b}_{N_p1} & \cdots & \boldsymbol{b}_{N_pN_p} & \boldsymbol{g}_{N_p1} & \cdots & \boldsymbol{g}_{N_pN_p} \\
\tilde{\boldsymbol{b}}_{11} & \cdots & \tilde{\boldsymbol{b}}_{1N_p} & \boldsymbol{c}_{11} & \cdots & \boldsymbol{c}_{1N_p} & \boldsymbol{h}_{11} & \cdots & \boldsymbol{h}_{1N_p} \\
\vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
\tilde{\boldsymbol{b}}_{N_p1} & \cdots & \tilde{\boldsymbol{b}}_{N_pN_p} & \boldsymbol{c}_{N_p1} & \cdots & \boldsymbol{c}_{N_pN_p} & \boldsymbol{h}_{N_p1} & \cdots & \boldsymbol{h}_{N_pN_p} \\
\tilde{\boldsymbol{g}}_{11} & \cdots & \tilde{\boldsymbol{g}}_{1N_p} & \tilde{\boldsymbol{h}}_{11} & \cdots & \tilde{\boldsymbol{h}}_{1N_p} & \boldsymbol{m}_{11} & \cdots & \boldsymbol{m}_{1N_p} \\
\vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
\tilde{\boldsymbol{g}}_{N_p1} & \cdots & \tilde{\boldsymbol{g}}_{N_pN_p} & \tilde{\boldsymbol{h}}_{N_p1} & \cdots & \tilde{\boldsymbol{h}}_{N_pN_p} & \boldsymbol{m}_{N_p1} & \cdots & \boldsymbol{m}_{N_pN_p}
\end{pmatrix} . \tag{3.15}
$$

Due to the nature of $\boldsymbol{\mathcal{M}}^{\infty}$ it is sufficient to describe the respective block for two particles α and β. We define the distance vector between the centers of these particles as $\boldsymbol{r} := \boldsymbol{x}_\beta - \boldsymbol{x}_\alpha$. For the blocks in representation (3.15) of the mobility matrix we get

$$\boldsymbol{a}_{\alpha\beta} = \begin{cases} \dfrac{1}{a_\alpha}\mathbf{Id}, & \alpha = \beta \\ \dfrac{3}{4}\big(\boldsymbol{\mathfrak{G}}(\boldsymbol{r}) + \dfrac{1}{6}a_\alpha^2\Delta\boldsymbol{\mathfrak{G}}(\boldsymbol{r}) + \dfrac{1}{6}a_\beta^2\Delta\boldsymbol{\mathfrak{G}}(\boldsymbol{r})\big), & \alpha \neq \beta \end{cases} \tag{3.16}$$

$$\boldsymbol{b}_{\alpha\beta} = \begin{cases} 0, & \alpha = \beta \\ \dfrac{3}{4}\boldsymbol{\mathfrak{R}}(\boldsymbol{r}), & \alpha \neq \beta \end{cases} \tag{3.17}$$

$$\boldsymbol{c}_{\alpha\beta} = \begin{cases} \dfrac{3}{4a_\alpha^3}\mathbf{Id}, & \alpha = \beta \\ \dfrac{3}{8}(\nabla \times \boldsymbol{\mathfrak{R}})(\boldsymbol{r}), & \alpha \neq \beta \end{cases} \tag{3.18}$$

$$\boldsymbol{g}_{\alpha\beta} = \begin{cases} 0, & \alpha = \beta \\ \dfrac{3}{4}\big(\boldsymbol{\mathfrak{K}}(\boldsymbol{r}) + \dfrac{1}{6}a_\alpha^2\Delta\boldsymbol{\mathfrak{K}}(\boldsymbol{r}) + \dfrac{1}{10}a_\beta^2\Delta\boldsymbol{\mathfrak{K}}(\boldsymbol{r})\big), & \alpha \neq \beta \end{cases} \tag{3.19}$$

$$\boldsymbol{h}_{\alpha\beta} = \begin{cases} 0, & \alpha = \beta \\ \dfrac{3}{8}(\nabla \times \boldsymbol{\mathfrak{K}})(\boldsymbol{r}), & \alpha \neq \beta \end{cases} \tag{3.20}$$

$$\boldsymbol{m}_{\alpha\beta} = \begin{cases} \dfrac{9}{10a_\alpha^3}\mathbf{Id}, & \alpha = \beta \\ \tfrac{3}{8}\big[\boldsymbol{\nabla}\boldsymbol{\mathfrak{K}}(\boldsymbol{r}) + (\boldsymbol{\nabla}\boldsymbol{\mathfrak{K}}(\boldsymbol{r}))^{\mathrm{T}} & \\ \quad + \tfrac{1}{10}a_\alpha^2\big(\boldsymbol{\nabla}(\Delta\boldsymbol{\mathfrak{K}}(\boldsymbol{r})) + (\boldsymbol{\nabla}(\Delta\boldsymbol{\mathfrak{K}}(\boldsymbol{r})))^{\mathrm{T}}\big) & \\ \quad + \tfrac{1}{10}a_\beta^2\big(\boldsymbol{\nabla}(\Delta\boldsymbol{\mathfrak{K}}(\boldsymbol{r})) + (\boldsymbol{\nabla}(\Delta\boldsymbol{\mathfrak{K}}(\boldsymbol{r})))^{\mathrm{T}}\big)\big], & \alpha \neq \beta. \end{cases} \tag{3.21}$$

As $\boldsymbol{\mathcal{M}}^\infty$ is symmetric, we gain the remaining entries $\tilde{\boldsymbol{b}}$, $\tilde{\boldsymbol{g}}$ and $\tilde{\boldsymbol{h}}$ by transposing the respective blocks $\tilde{\boldsymbol{b}}_{\alpha\beta} = \boldsymbol{b}_{\beta\alpha}^{\mathrm{T}}$ etc. The tensors $\boldsymbol{\mathfrak{R}}$ and $\boldsymbol{\mathfrak{K}}$ are a result of the multipole expansion used to derive (3.4). The rotlet $\boldsymbol{\mathfrak{R}} \in \mathcal{R}^{3\times3}$ is given by

$$\boldsymbol{\mathfrak{R}}_{ij}(\boldsymbol{r}) := \sum_{k=1}^{3} \epsilon_{ijk} r_k r_{\alpha\beta}^{-3}, \tag{3.22}$$

where ϵ is the Levi-Civita symbol. The tensor $\boldsymbol{\mathfrak{K}} \in \mathcal{R}^{3\times3\times3}$ is defined by

$$\boldsymbol{\mathfrak{K}}_{ijk}(\boldsymbol{r}) := \frac{1}{2}\big(\nabla_k \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) + \nabla_j \boldsymbol{\mathfrak{G}}_{ik}(\boldsymbol{r})\big). \tag{3.23}$$

The entries of $\boldsymbol{\mathcal{M}}^{\infty}$ can be calculated explicitly at the considered level of approximation. Evaluating the derivatives in (3.16)–(3.21), we gain the scalar mobility functions. Using the notation in [Durlofsky et al., 1987] we get for the elements of the submatrices of the mobility matrix (3.15):

$$\begin{aligned}
(\boldsymbol{a}_{\alpha\beta})_{ij} &= x^a_{\alpha\beta} e_i e_j + y^a_{\alpha\beta}(\delta_{ij} - e_i e_j), && (3.24)\\
(\boldsymbol{b}_{\alpha\beta})_{ij} &= y^b_{\alpha\beta} \epsilon_{ijk} e_k, && (3.25)\\
(\boldsymbol{c}_{\alpha\beta})_{ij} &= x^c_{\alpha\beta} e_i e_j + y^c_{\alpha\beta}(\delta_{ij} - e_i e_j), && (3.26)\\
(\boldsymbol{g}_{\alpha\beta})_{ijk} &= x^g_{\alpha\beta}(e_i e_j e_k - \frac{1}{3}\delta_{jk} e_i) + y^g_{\alpha\beta}(\delta_{ij} e_k + \delta_{ik} e_j - 2 e_i e_j e_k), \\
& && (3.27)\\
(\boldsymbol{h}_{\alpha\beta})_{ijk} &= y^h_{\alpha\beta}(\epsilon_{ijl} e_k e_l + \epsilon_{ikl} e_j e_l), && (3.28)\\
(\boldsymbol{m}_{\alpha\beta})_{ijkl} &= \frac{3}{2} x^m_{\alpha\beta}(e_i e_j - \frac{1}{3}\delta_{ij})(e_k e_l - \frac{1}{3}\delta_{kl}) \\
&\quad + \frac{1}{2} y^m_{\alpha\beta}(e_i \delta_{jl} e_k + e_j \delta_{il} e_k + e_i \delta_{jk} e_l + e_j \delta_{ik} e_l - 4 e_i e_j e_k e_l) \\
&\quad + \frac{1}{2} z^m_{\alpha\beta}(\delta_{ik}\delta_{jl} + \delta_{jk}\delta_{il} - \delta_{ij}\delta_{kl} + e_i e_j \delta_{kl} + \delta_{ij} e_k e_l + e_i e_j e_k e_l \\
&\qquad - e_i \delta_{jl} e_k - e_j \delta_{il} e_k - e_i \delta_{jk} e_l - e_j \delta_{ik} e_l), && (3.29)
\end{aligned}$$

where we defined the unit vector $\boldsymbol{e} := (\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha)/r_{\alpha\beta}$ with the scalar particle distance $r_{\alpha\beta} := \|\boldsymbol{x}_\beta - \boldsymbol{x}_\alpha\|_2$. The corresponding mobility functions are

$$x^a_{\alpha\alpha} = \frac{1}{a_\alpha}, \qquad x^a_{\alpha\beta} = \frac{3}{2} r^{-1}_{\alpha\beta} - \frac{a^2_\alpha + a^2_\beta}{2} r^{-3}_{\alpha\beta}, \tag{3.30}$$

$$y^a_{\alpha\alpha} = \frac{1}{a_\alpha}, \qquad y^a_{\alpha\beta} = \frac{3}{4} r^{-1}_{\alpha\beta} + \frac{a^2_\alpha + a^2_\beta}{4} r^{-3}_{\alpha\beta}, \tag{3.31}$$

$$y^b_{\alpha\alpha} = 0, \qquad y^b_{\alpha\beta} = \frac{3}{4} r^{-2}_{\alpha\beta} \tag{3.32}$$

$$x^c_{\alpha\alpha} = \frac{3}{4a^3_\alpha}, \qquad x^c_{\alpha\beta} = \frac{3}{4} r^{-3}_{\alpha\beta}, \tag{3.33}$$

$$y^c_{\alpha\alpha} = \frac{3}{4a^3_\alpha}, \qquad y^c_{\alpha\beta} = -\frac{3}{8} r^{-3}_{\alpha\beta}, \tag{3.34}$$

$$x^g_{\alpha\alpha} = 0, \qquad x^a_{\alpha\beta} = -\frac{9}{4} r^{-2}_{\alpha\beta} + \frac{9}{4} a^2_\alpha r^{-4}_{\alpha\beta} + \frac{27}{20} a^2_\beta r^{-4}_{\alpha\beta}, \tag{3.35}$$

$$y^g_{\alpha\alpha} = 0, \qquad y^a_{\alpha\beta} = -\frac{3}{4} a^2_\alpha r^{-4}_{\alpha\beta} - \frac{9}{20} a^2_\beta r^{-4}_{\alpha\beta}, \tag{3.36}$$

$$y^h_{\alpha\alpha} = 0, \qquad y^h_{\alpha\beta} = -\frac{9}{8} r^{-3}_{\alpha\beta} \tag{3.37}$$

$$x^m_{\alpha\alpha} = \frac{9}{10a^3_\alpha} \qquad x^m_{\alpha\beta} = -\frac{9}{2} r^{-3}_{\alpha\beta} + \frac{27}{5}(a^2_\alpha + a^2_\beta) r^{-5}_{\alpha\beta} \tag{3.38}$$

$$y^m_{\alpha\alpha} = \frac{9}{10a^3_\alpha} \qquad y^m_{\alpha\beta} = \frac{9}{4} r^{-3}_{\alpha\beta} - \frac{18}{5}(a^2_\alpha + a^2_\beta) r^{-5}_{\alpha\beta} \tag{3.39}$$

$$z^m_{\alpha\alpha} = \frac{9}{10a^3_\alpha} \qquad z^m_{\alpha\beta} = -\frac{9}{10}(a^2_\alpha + a^2_\beta) r^{-5}_{\alpha\beta}. \tag{3.40}$$

3.6 Detailed derivatives

We will show in detail how the derivatives of $\boldsymbol{\mathfrak{G}}$, $\boldsymbol{\mathfrak{R}}$ and $\boldsymbol{\mathfrak{K}}$ are obtained. Note that for the vector $\boldsymbol{r}$, with $r_{\alpha\beta} = \left\|\boldsymbol{r}\right\|_2$ and the components $r_k, k = 1, 2, 3$,

$$\nabla_k r^{-p}_{\alpha\beta} = \nabla_k (r_1^2 + r_2^2 + r_3^2)^{-\frac{p}{2}} = -\frac{p}{2}(r_1^2 + r_2^2 + r_3^2)^{-\frac{p}{2}-1} 2r_k = -p r^{-(p+2)}_{\alpha\beta} r_k$$

holds. According to definition (3.3), the elements of the Oseen tensor are defined as

$$\boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) = \frac{1}{r_{\alpha\beta}}\Big(\delta_{ij} + \frac{1}{r_{\alpha\beta}^2} r_i r_j\Big).$$

We get the derivatives:

$$\begin{aligned}\nabla_k \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) &= -\frac{1}{r_{\alpha\beta}^3}\delta_{ij} r_k + \frac{1}{r_{\alpha\beta}^3}\delta_{ik} r_j + \frac{1}{r_{\alpha\beta}^3}\delta_{jk} r_i - \frac{3}{r_{\alpha\beta}^5} r_i r_j r_k \\ &= \frac{1}{r_{\alpha\beta}^3}(-\delta_{ij} r_k + \delta_{ik} r_j + \delta_{jk} r_i) - \frac{3}{r_{\alpha\beta}^5} r_i r_j r_k,\end{aligned}$$

$$\begin{aligned}\nabla_k \nabla_k \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) = &\frac{1}{r_{\alpha\beta}^3}(-\delta_{ij} + \delta_{ij}\delta_{jk} + \delta_{jk}\delta_{ik}) \\ &- \frac{3}{r_{\alpha\beta}^5} r_k(-\delta_{ij} r_k + \delta_{ik} r_j + \delta_{jk} r_i) \\ &- \frac{3}{r_{\alpha\beta}^5}(\delta_{ik} r_j r_k + \delta_{jk} r_i r_k + r_i r_j) + \frac{15}{r_{\alpha\beta}^7} r_i r_j r_k^2,\end{aligned}$$

$$\begin{aligned}\Delta \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) = &\sum_{k=1}^{3} \nabla_k \nabla_k \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) \\ = &\frac{1}{r_{\alpha\beta}^3}(-3\delta_{ij} + \delta_{ij} + \delta_{ij}) - \frac{3}{r_{\alpha\beta}^5}(-\delta_{ij} r_{\alpha\beta}^2 + r_i r_j + r_j r_i) \\ &- \frac{3}{r_{\alpha\beta}^5}(r_j r_i + r_i r_j + 3 r_i r_j) + \frac{15}{r_{\alpha\beta}^7} r_i r_j r_{\alpha\beta}^2 \\ = &\frac{2}{r_{\alpha\beta}^3}\delta_{ij} - \frac{6}{r_{\alpha\beta}^5} r_j r_i,\end{aligned}$$

$$\begin{aligned}\nabla_k \Delta \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) =& -\frac{6}{r_{\alpha\beta}^5} r_k \delta_{ij} - \frac{6}{r_{\alpha\beta}^5}(\delta_{ik} r_j + \delta_{jk} r_i) + \frac{30}{r_{\alpha\beta}^7} r_j r_i r_k \\ =& -\frac{6}{r_{\alpha\beta}^5}(r_k \delta_{ij} + \delta_{ik} r_j + \delta_{jk} r_i) + \frac{30}{r_{\alpha\beta}^7} r_j r_i r_k,\end{aligned}$$

$$\begin{aligned}\nabla_k \nabla_k \Delta \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) =& -\frac{6}{r_{\alpha\beta}^5}(\delta_{ij} + \delta_{ik}\delta_{jk} + \delta_{jk}\delta_{ik}) \\ &+ \frac{30}{r_{\alpha\beta}^7} r_k (r_k \delta_{ij} + \delta_{ik} r_j + \delta_{jk} r_i) \\ &+ \frac{30}{r_{\alpha\beta}^7}(\delta_{jk} r_i r_k + \delta_{ik} r_j r_k + r_j r_i) - \frac{210}{r_{\alpha\beta}^9} r_j r_i r_k^2,\end{aligned}$$

$$\Delta(\Delta \boldsymbol{\mathfrak{G}}_{ij})(\boldsymbol{r}) = \sum_{k=1}^{3} \nabla_k \nabla_k \Delta \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) = 0,$$

$$\begin{aligned}\nabla_l \nabla_k \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) =& \frac{1}{r_{\alpha\beta}^3}(-\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{jk}\delta_{il}) \\ &- \frac{3}{r_{\alpha\beta}^5} r_l(-\delta_{ij} r_k + \delta_{ik} r_j + \delta_{jk} r_i) \\ &- \frac{3}{r_{\alpha\beta}^5}(\delta_{il} r_j r_k + r_i \delta_{jl} r_k + r_i r_j \delta_{kl}) + \frac{15}{r_{\alpha\beta}^7} r_i r_j r_k r_l,\end{aligned}$$

$$\begin{aligned}\nabla_l \nabla_k \Delta \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) =& -\frac{6}{r_{\alpha\beta}^5}(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{jk}\delta_{il}) \\ &+ \frac{30}{r_{\alpha\beta}^7} r_l(\delta_{ij} r_k + \delta_{ik} r_j + \delta_{jk} r_i) \\ &+ \frac{30}{r_{\alpha\beta}^7}(\delta_{jl} r_i r_k + r_j \delta_{il} r_k + r_j r_i \delta_{kl}) - \frac{210}{r_{\alpha\beta}^9} r_i r_j r_k r_l,\end{aligned}$$

$$\begin{aligned}
(\boldsymbol{\nabla} \times \boldsymbol{\mathfrak{G}})_{ij}(\boldsymbol{r}) &= \sum_{l=1}^{3}\sum_{k=1}^{3} \epsilon_{ikl}\nabla_k \boldsymbol{\mathfrak{G}}_{lj} \\
&= \sum_{l=1}^{3}\sum_{k=1}^{3} \epsilon_{ikl}\Big(\frac{1}{r_{\alpha\beta}^{3}}(-\delta_{lj}r_k + \delta_{lk}r_j + \delta_{jk}r_l) - \frac{3}{r_{\alpha\beta}^{5}} r_l r_j r_k\Big) \\
&= \sum_{l=1}^{3}\sum_{k=1}^{3} \frac{1}{r_{\alpha\beta}^{3}}(-\epsilon_{ikj}r_k + \epsilon_{ijl}r_l) \\
&= \sum_{l=1}^{3}\sum_{k=1}^{3} \frac{1}{r_{\alpha\beta}^{3}}(\epsilon_{ijk}r_k + \epsilon_{ijl}r_l) = \sum_{k=1}^{3} \frac{2}{r_{\alpha\beta}^{3}}\epsilon_{ijk}r_k,
\end{aligned}$$

$$\begin{aligned}
(\boldsymbol{\nabla} \times \boldsymbol{\nabla}\boldsymbol{\mathfrak{G}})_{ijk}(\boldsymbol{r}) &= \sum_{m=1}^{3}\sum_{l=1}^{3} \epsilon_{ilm}\nabla_l\nabla_m \boldsymbol{\mathfrak{G}}_{jk} \\
&= \sum_{m=1}^{3}\sum_{l=1}^{3} \epsilon_{ilm}\Big(\frac{1}{r_{\alpha\beta}^{3}}(-\delta_{jk}\delta_{ml} + \delta_{jm}\delta_{kl} + \delta_{km}\delta_{jl}) \\
&\qquad\qquad - \frac{3}{r_{\alpha\beta}^{5}} r_l(-\delta_{jk}r_m + \delta_{jm}r_k + \delta_{km}r_j) \\
&\qquad\qquad - \frac{3}{r_{\alpha\beta}^{5}}(\delta_{jl}r_k r_m + r_j\delta_{kl}r_m + r_j r_k\delta_{ml}) \\
&\qquad\qquad + \frac{15}{r_{\alpha\beta}^{7}} r_j r_k r_m r_l\Big) \\
&= \sum_{m=1}^{3}\sum_{l=1}^{3}\Big(\frac{1}{r_{\alpha\beta}^{3}}(\epsilon_{ikj} + \epsilon_{ijk}) \\
&\quad - \frac{3}{r_{\alpha\beta}^{5}}(\epsilon_{ilj}r_l r_k + \epsilon_{ilk}r_l r_j - \epsilon_{ijm}r_k r_m - \epsilon_{ikm}r_j r_m)\Big) \\
&= \sum_{l=1}^{3} \frac{18}{r_{\alpha\beta}^{5}}(\epsilon_{ijl}r_k r_l + \epsilon_{ikl}r_j r_l),
\end{aligned}$$

$$\begin{aligned}(\boldsymbol{\nabla}\times\Delta\mathfrak{G})_{ij}(\boldsymbol{r}) =& \sum_{l=1}^{3}\sum_{k=1}^{3}\epsilon_{ikl}\Big(\frac{6}{r_{\alpha\beta}^{5}}(-r_k\delta_{lj}-\delta_{lk}r_j-\delta_{jk}r_l)+\frac{30}{r_{\alpha\beta}^{7}}r_jr_lr_k\Big)\\ =& \sum_{l=1}^{3}\sum_{k=1}^{3}\frac{6}{r_{\alpha\beta}^{5}}(-\epsilon_{ikj}-\epsilon_{ijl})\\ =& \sum_{l=1}^{3}\sum_{k=1}^{3}\frac{6}{r_{\alpha\beta}^{5}}(\epsilon_{ijk}r_k-\epsilon_{ijl}r_l)=0.\end{aligned}$$

The rotlet $\mathfrak{R}$ is defined in (3.22) as

$$\mathfrak{R}_{ij}(\boldsymbol{x})=\sum_{k=1}^{3}\epsilon_{ijk}r_kr_{\alpha\beta}^{-3}.$$

The derivatives required for the construction of the mobility matrix are

$$\begin{aligned}\nabla_k\mathfrak{R}_{ij}(\boldsymbol{x}) =& \sum_{l=1}^{3}\nabla_k(\frac{1}{r_{\alpha\beta}^{3}}\epsilon_{ijl}r_l)\\ =& \sum_{l=1}^{3}\epsilon_{ijl}(\frac{1}{r_{\alpha\beta}^{3}}\delta_{kl}-\frac{3}{r_{\alpha\beta}^{5}}r_kr_l)\\ =& \frac{1}{r_{\alpha\beta}^{3}}\epsilon_{ijk}-\sum_{l=1}^{3}\frac{3}{r_{\alpha\beta}^{5}}\epsilon_{ijl}r_kr_l,\end{aligned}$$

and

$$\begin{aligned}
(\boldsymbol{\nabla} \times \boldsymbol{\mathfrak{R}})_{ij}(\boldsymbol{x}) &= \sum_{k=1}^{3}\sum_{l=1}^{3} \epsilon_{ikl} \nabla_k \boldsymbol{\mathfrak{R}}_{lj} \\
&= \sum_{k=1}^{3}\sum_{l=1}^{3} \epsilon_{ikl} \Big(\frac{1}{r_{\alpha\beta}^3} \epsilon_{ljk} - \sum_{m=1}^{3} \frac{3}{r_{\alpha\beta}^5} \epsilon_{ljm} r_k r_m \Big) \\
&= \frac{2}{r_{\alpha\beta}^3}\delta_{ij} - \sum_{k=1}^{3}\sum_{l=1}^{3}\sum_{m=1}^{3} \frac{3}{r_{\alpha\beta}^5} \epsilon_{ikl}\epsilon_{jml} r_k r_m \\
&= \frac{2}{r_{\alpha\beta}^3}\delta_{ij} + \frac{3}{r_{\alpha\beta}^5}(1-\delta_{ij}) r_i r_j - \frac{3}{r_{\alpha\beta}^5}\delta_{ij} \sum_{k=1}^{3}\sum_{l=1}^{3} (\epsilon_{ikl})^2 r_k^2 \\
&= \frac{2}{r_{\alpha\beta}^3}\delta_{ij} + \frac{3}{r_{\alpha\beta}^5} r_i r_j + \frac{3}{r_{\alpha\beta}^5}\delta_{ij} r_i^2 - \frac{3}{r_{\alpha\beta}^5}\delta_{ij}(r_{\alpha\beta}^2 - r_i^2) \\
&= -\frac{1}{r_{\alpha\beta}^3}\delta_{ij} + \frac{3}{r_{\alpha\beta}^5} r_i r_j .
\end{aligned}$$

Since

$$\boldsymbol{\mathfrak{K}}_{ijk}(\boldsymbol{r}) = \frac{1}{2}\big(\nabla_k \boldsymbol{\mathfrak{G}}_{ij}(\boldsymbol{r}) + \nabla_j \boldsymbol{\mathfrak{G}}_{ik}(\boldsymbol{r})\big)$$

we gain all derivatives needed to calculate the mobility functions (3.16)–(3.21) given in 3.5 as combination of the above expressions.

Chapter 4

On the settling behaviour of polydisperse particle clouds in viscous fluids

Abstract
We investigate the behaviour of polydisperse particle clouds settling in a viscous fluid by means of numerical simulations. The sedimentation process is described on examples of bidisperse systems as well as fully polydisperse particle systems, pointing out parallels and differences to the monodisperse case. Based on our results we propose a simple formula which gives the maximum velocity of a polydisperse particle cloud settling in a viscous fluid. The proposed formula agrees with our numerical results even in case of systems with high polydispersity.

This chapter follows our work of the same title published in the European Journal of Mechanics - B/Fluids.

4.1 Introduction

Sedimentation is part of many industrial processes in the fields of chemical, biological and environmental engineering. Waste water treatment and mineral processing are only two examples where the treatment of highly polydisperse suspensions plays a major role. As opposed to the assumption of many models, particles are normally not spatially uniformly distributed in such suspensions. Due to hydrodynamic interactions and the polydisperse nature of the particulate system, clusters of different sizes and lifetimes are formed. Such clusters, or particle clouds, can undergo a remarkable motion. In the absence of turbulence, spherical clouds behave like drops of fluid settling in another fluid with lower mass density [Machu et al., 2001]. During sedimentation such a cloud evolves into a torus. The inhomogeneous distribution of particles in the torus causes oscillations which eventually lead to a break-up into two or more smaller clouds, which then undergo the same evolution. This process continues until the resulting clouds are so small that they disintegrate. In recent years such particle clouds have drawn a considerable amount of attention and have been investigated both experimentally as well as numerically. Different regimes of evolution for a sedimenting cluster of particles have been identified [Subramanian and Koch, 2008]. These depend on the particle Reynolds number, the cloud Reynolds number, the particle volume fraction inside the cluster and the size of the cluster. Research has mostly been concentrated on two regimes. On the one hand a regime where the cloud Reynolds number is small but cannot be assumed to vanish and thus the particle evolution can be described by Oseen interactions. On the other hand the Stokes regime, where the cloud Reynolds number is small enough to be assumed to be zero. In this regime, some particles can make an outward crossing of the settling cloud and leave it at its rear end in a vertical tail. The rate of leakage, which describes the rate of leaked particles per time, has been studied extensively [Nitsche and Batchelor, 1997, Metzger et al., 2007]. In case of a higher but still small cloud Reynolds number the leakage decreases to an insignificant number of particles, while the evolution of the shape of the cloud remains the same [Pignatel et al., 2011]. Apart from qualitative investigations and descriptions of the shape evolution of particle clouds, most works deal with the sedimentation velocity of particle clouds. Adachi et al. [1978] were among the first who examined the cloud evolution experimentally and gave a theoretical prediction for the sedimentation velocity of a cloud comprised of a large number of particles. With the increase in computational power, numerical simulations of settling clouds of many particles have become feasible. Nitsche and Batchelor [1997] proposed a simple formula for the sedimentation velocity of a spherical particle cloud, which was in good agreement with results from their numerical simulations of monodisperse particle clouds comprised of up to 320 particles. Bosse et al. [2005] have simulated particle clouds of several thousand particles, yielding results comparable to those of Nitsche and Bachelor and their proposed formula in case of a low cloud Reynolds number. Bosse et al. [2005] also found that for an increasing cloud Reynolds number, the number of emerging secondary clouds rises. Apart from this the overall process stays the same – a repeating cascade starting from an initially spherical cloud, which flattens, evolves into a torus eventually breaking up into smaller clouds, which then evolve in the same manner. A number of other works deal with experiments, numerical simulations or theoretical considerations on particle clouds in viscous fluids [Metzger et al., 2007, Alabrudzinski et al., 2009, Ekiel-Jezewska et al., 2006]. They all discuss the topics of cloud behaviour, initial cloud velocity and particle leakage. But they confine themselves to investigations on monodisperse particle clouds. Abade and Cunha [2007] have investigated the behaviour of slightly polydisperse particle clouds. They found that polydisperse clouds comprised of a small number of particles are less stable than comparable monodisperse clouds. Apart from clouds of spherical particles, some works consider clouds of fibres [Park et al., 2010] and fibre-particle mixtures [Feist et al., 2011]. Nevertheless, we are not aware of any works explicitly investigating the settling behaviour of polydisperse clouds comprised of more than 1000 particles, neither numerically nor experimentally. The present work deals with the behaviour of such clouds at vanishing Reynolds number and shall serve as a basis for further research in this area. Basis for our findings is our parallel Stokesian Dynamics code, which is based on the open source software RYUON [ryu, 2014]. Stokesian Dynamics is a numerical method developed and especially well-suited for the simulation of particle suspensions [Durlofsky et al., 1987, Brady and Bossis, 1988]. Our work is structured as follows. First we give a short introduction to the simulation method and we derive a formula yielding the maximum velocity of a polydisperse particle cloud settling at vanishing Reynolds number. This formula agrees well with our numerical simulations, as we show in the results section. Furthermore we present results of our investigations on the motion of polydisperse clouds on examples of bidisperse and fully polydisperse clouds.

4.2 Methods

In this section we describe our simulation method and we derive a formula which gives the settling velocity of a polydisperse spherical particle cloud. We confine ourselves to a brief discussion of the exact form of the numerical scheme we use, since the method and extensions to it are well known. For readers not familiar with the Stokesian Dynamics method, we refer to [Ichiki, 2002, Brady and Bossis, 1988, Sierou and Brady, 2001].

4.2.1 Numerical Scheme

We consider particles settling in a viscous fluid, thus the Stokes equations suffice to describe the fluid motion. When interparticle forces are negligible, the equations of motion for suspended spherical non-Brownian particles solely under the influence of gravity are

$$m_p \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{U}_\alpha = \boldsymbol{F}^h_\alpha + \boldsymbol{F_g}, \tag{4.1}$$

and

$$\boldsymbol{I}_\alpha \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\omega_\alpha} = \boldsymbol{T}^{\mathrm{h}}_\alpha. \tag{4.2}$$

Here m_p is the mass of a particle, $\boldsymbol{U}_\alpha$ its translational velocity, $\boldsymbol{F}^h_\alpha$ the hydrodynamic force acting on the particle, and $\boldsymbol{F_g}$ gravity. $\boldsymbol{I}_\alpha$ denotes the moment of inertia tensor of a particle, $\boldsymbol{\omega_\alpha}$ its angular velocity, and $\boldsymbol{T}^{\mathrm{h}}_\alpha$ the hydrodynamic torque acting on the particle. Based on the integral expression for the fluid velocity field in Stokes flow [Durlofsky et al., 1987], a linear relation between hydrodynamic force, hydrodynamic torque and the translational and angular velocities of particles suspended in a fluid can be found. This relation is given by

$$-\boldsymbol{\mathcal{M}}^\infty \begin{pmatrix} \boldsymbol{F_H} \\ \boldsymbol{T_H} \end{pmatrix} = \begin{pmatrix} \boldsymbol{U_P} \\ \boldsymbol{\omega_P} \end{pmatrix}, \tag{4.3}$$

where $\boldsymbol{F_H}, \boldsymbol{T_H}, \boldsymbol{U_P}, \boldsymbol{\omega_P} \in \mathbb{R}^{3N}$ are the same variables as in equations (4.1) and (4.2), with the difference that the vectors in equation (4.3) contain the components of all N particles. The mobility matrix $\boldsymbol{\mathcal{M}}^{\infty} \in \mathbb{R}^{6N \times 6N}$ depends only on the particle configuration and the particle radii explicitly. The fluid motion does not have to be computed, which is a major advantage over grid-based methods as for instance a coupling of a finite volume method and the discrete element method. A nondimensionalization of the equations leads to the definition of the Stokes number, which in our case is given by

$$\mathrm{St} = \frac{2}{9} \frac{\rho_p}{\rho_f} \mathrm{Re}. \tag{4.4}$$

ρ_p and ρ_f are the particle mass density and the fluid mass density, respectively. Re denotes the Reynolds number of the flow. If we can assume Stokes flow with vanishing Reynolds number, we can assume the Stokes number to vanish as well, provided that $\rho_p \approx O(\rho_f)$. With this assumption the left hand sides of the dimensionless form of equations (4.1) and (4.2) become zero, resulting in $\boldsymbol{F}^h_\alpha = -\boldsymbol{F_g}$ and $\boldsymbol{T}^{\mathrm{h}}_\alpha = \boldsymbol{0}$. Substitution into equation (4.3) now yields a linear relation between gravity on the one hand, and particle velocities on the other. The resulting algorithm reads as follows: (i) compute $\boldsymbol{\mathcal{M}}^{\infty}$ from particle positions and their radii, (ii) obtain the particle velocities from the substituted form of equation (4.3), (iii) integrate $\boldsymbol{U_P}$, which contains all translational velocities, to yield the new particle positions. The problem we solve – obtaining particle velocities from acting forces – is called mobility problem. For the construction of the mobility matrix $\boldsymbol{\mathcal{M}}^{\infty}$ in (4.3) higher moments in the expansion of the force density in the expression for the fluid velocity field in Stokes flow are neglected [Brady and Bossis, 1988]. We only keep the zeroth moment, the force $\boldsymbol{F_H}$, and the antisymmetric part of the first moment, the torque $\boldsymbol{T_H}$. This leads to an error when particles are in close proximity, as it usually is the case in dense particle suspensions. Closely related to the mobility problem is the so-called resistance problem. It has to be solved when particle velocities are given and the forces are unknown. This is usually the case when dense suspensions are considered and lubrication effects cannot be neglected. There are several ways to include near-field lubrication effects for this type of problem [Jeffrey and Onishi, 1984, Jeffrey, 1992, Cichocki et al., 1999]. We chose to solve the mobility problem without lubrication correction rather than solving the resistance problem with lubrication correction for the simple reason that the resulting algorithm is much faster than a solution of the resistance problem and yet exact enough to capture the dynamics of a particle cloud. It is also possible to include near field effects by an introduction of a short-range repulsive force, as in [Nitsche and Batchelor, 1997], [Abade and Cunha, 2007] and [Cunha et al., 2002]. This method of including hydrodynamic near-field interactions has the advantage that no inversion of the mobility matrix is necessary while near-field effects are still taken into account. However, our numerical tests of two particles interacting hydrodynamically as in [Goldman et al., 1966] and [Pozrikidis, 2007] have shown that, at least with the proposed choice of parameters, this treatment produces unrealistic results. In the following we will call an algorithm solving the mobility problem mobility scheme and a method for the solution of the resistance problem resistance scheme, respectively.

4.2.2 Cloud Settling Velocity of a Polydisperse Cloud

Nitsche and Batchelor [1997] have proposed a formula for the settling velocity U_c of a monodisperse spherical particle cloud at low Reynolds number,

$$U_c \approx U_{St} \left(\frac{6}{5} N \frac{a_p}{a_c} + 1 \right), \tag{4.5}$$

where U_{St} denotes the Stokes velocity of a single particle with radius a_p and a_c is the radius of the particle cloud formed by the N particles. Equation (4.5) has been derived from the Hadamard-Rybczynski equation (H-R equation). The H-R equation describes the terminal velocity of a spherical drop in a quiescent fluid under the action of gravity [Michaelides, 2006]. It is insignificant for the derivation whether the matter forming the spherical drop has a higher or lower mass density than the suspending fluid. Thus, based on the observation that particle clouds behave like settling drops of fluid [Machu et al., 2001], it seems natural to utilize the Hadamard-Rybczynski equation in order to derive a formula which gives the settling velocity of spherical particle clouds. In order to derive a formula yielding the maximum settling velocity of a polydisperse spherical particle cloud, we start from the Hadamard-Rybczynski equation as well. The cloud settling velocity U_c, as we derive it, is given by

$$U_c = \frac{6}{5} \sum_{i=1}^{N_f} N_{p,i} \frac{a_{p,i}}{a_c} U_{St,i}. \tag{4.6}$$

$U_{St,i}$ is the Stokes velocity of a spherical particle with radius $a_{p,i}$, N_f is the number of particle size fractions in the system forming a cloud of radius a_c. Fraction i contains $N_{p,i}$ particles. A detailed derivation of formula (4.6) is given in 4.4. The only assumption for this formula is that the particle cloud is dilute. This can be described by $\phi \ll 1$, where ϕ is the particle volume fraction in the spherical cloud. Because of this requirement, equation (4.6) seems to be valid only for cases when ϕ is very small. But as we will show in section 4.3, it gives a good approximation for a broad spectrum of particle volume fractions. As a comparison of equations (4.5) and (4.6) shows, assuming a monodisperse particle system, equation (4.5) contains an additional term equal to the settling velocity U_{St} of an isolated particle. Using a different approach, Ekiel-Jezewska et al. [2006] have derived a formula for the settling velocity of monodisperse particle clouds as well. Their formula also contains the additional term, which they attribute to a slip between particles and ambient fluid. Apart from this term, equation (4.6) coincides with equation (4.5) in the monodisperse case. We discuss this topic in section 4.3.4. If all fractions of the particulate system are known, e.g. in numerical experiments, the velocity resulting from equation (4.6) can be used as characteristic value for nondimensionalization. This yields a more stable simulation, which saves time and resources.

4.3 Results and Discussion

4.3.1 Validation

We have validated our code on several examples, obtaining good agreement. In this section we give some examples which are significant for this work. For all numerical experiments we assume that the particle density is the same for all particles. The driving force in all simulations is gravity, which scales as the third power of the particle radius.

Error Made in One Time Step

To determine the accuracy of our simulation, we chose the example of two equal spheres settling in a given configuration in a quiescent fluid ([Goldman et al., 1966]; figure 4.1). This choice arises from the fact that this example covers different particle configurations relative to the settling direction, as well as different interparticle distances. The particle settling velocity obtained from the solution of the mobility problem without lubrication correction starts to deviate from the results of Goldman et al. [1966] by more than 1% at a distance of one particle diameter between

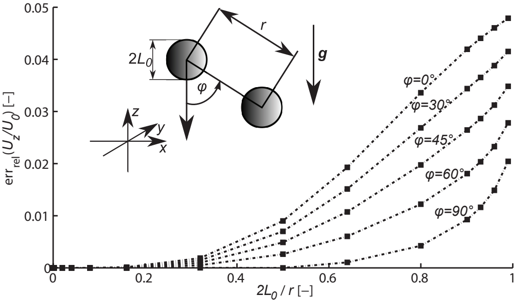

Figure 4.1: Particle configuration and relative deviation of our results using the mobility scheme from the results of Goldman et al. [1966] against the reciprocal dimensionless inter-particle spacing r.

the particle surfaces, where the deviation increases with shrinking gap size. Note that the deviation is smaller for larger angles φ, i.e. particles settling side by side (figure 4.1). The deviation is below 5% for all configurations, down to inter-particle gaps of 1% of a particle diameter.

Accuracy Over a Longer Time Span

In the previous example where we examined the error made in one time step, particles do not move relative to each other. To investigate the deviation of the solution of the mobility problem without lubrication correction from the solution of the resistance problem with lubrication correction after a longer time span, we have run numerical experiments as described in [Durlofsky et al., 1987]. Durlofsky et al. [1987] conducted numerical simulations of the settling of four and of eight spheres initially located at the corners of a square and a cube, respectively. Under the action of gravity and at vanishing Stokes number both configurations undergo a periodic motion. This motion has first been described by Hocking [1964] for the four-particle configuration. The spheres initially located at the top corners settle faster than the particles initially located at the bottom corners. After the faster settling particles pass the slower particles all particles return to the corners of the initial square or cube. Only this time the locations of top and bottom particles are switched. Now the particles initially located at the bottom corners settle faster and return to their original positions. As in [Durlofsky et al., 1987] we determined the vertical position when all particles form a horizontal line in case of four particles, and when they lie in a plane perpendicular to the settling direction in case of eight particles initially located at the corners of a cube. The considered dimensionless edge lengths are $r_{\text{init}} = 3, 4, 5$, the used level of approximation is the described FT-version. Our results using the resistance scheme with lubrication correction as in [Durlofsky et al., 1987] agree very well with their results. When we use the proposed mobility scheme without lubrication correction, our results deviate. This can be seen in figure 4.2. In case of eight settling particles (figure 4.2, right), the deviation is larger for particles with shorter initial distance. This is because lubrication effects play a bigger role for the configuration with shorter initial distance. In the graph on the left-hand side of figure 4.2,

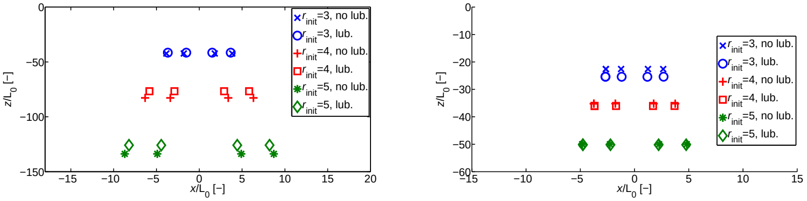

Figure 4.2: Left: Particle positions of four settling equal particles initially located at the corners of a square with initial edge length $r_{\text{init}} = 3, 4, 5$ when all particles form a line perpendicular to the settling direction. Right: Particle positions of eight settling equal particles initially located at the corners of a cube with initial edge length $r_{\text{init}} = 3, 4, 5$ when all particles lie in a plane perpendicular to the settling direction. Only four of the eight particles are shown because of the reflectional symmetry in the (x, z)- and (y, z)-planes. All lengths are given in dimensionless units.

where the vertical positions of four particles are displayed when they first form a horizontal line, one can see that the deviation is very small for the square with shortest edge length, whereas the square with longest edge length, where lubrication effects should play a smaller role, is larger. This results from the fact that the particles initially located at the corners of a square with edge length 5 settle over a longer distance until they form a line perpendicular to the settling direction. Thus the accumulated error is larger, even though the error in one time step is smaller than for a square with edge length 3. Solving the resistance problem with lubrication correction rather than the mobility problem without lubrication correction might lead to more precise results, but the computational time needed to achieve them is several orders of magnitude higher, especially for polydisperse systems. At the current state of the art, this restricts simulations using a lubrication correction scheme based on the resistance problem to the simulation of several hundreds to some thousands of particles within days or months. If we are interested in macroscopic properties such as the overall motion of a particle cloud, the solution of the mobility problem without lubrication correction suffices (see e.g. [Metzger et al., 2007] where a point force model was used). Provided that the code is parallelized, the method we chose allows for the simulation of the dynamics of several thousand up to a million particles within days [Bülow et al., 2014].

For such particle systems different effects can be observed compared to simulations of relatively small particle systems (cf. section 4.3.3).

Settling of a Monodisperse Particle Cloud

A qualitative comparison of our results from a simulation of the settling process of a monodisperse particle cloud of 3000 particles with the results of Metzger et al. [2007] shows good agreement, which supports our choice to solve the mobility problem rather than the time-consuming resistance problem. The characteristic stages of the settling process are shown in figure 4.3, where a) shows the initially spherical cloud, b) the formed torus (including top view), c) the particle system after the torus broke into smaller clouds, and finally d) even smaller clouds, formed after the secondary clouds broke up. To nondimensionalize the simulated system we use a characteristic length scale L_0 and a characteristic velocity U_0. For the simulation of bidisperse systems we chose the particle radius of the large size fraction as L_0. For simulations of fully polydisperse systems with a given radius distribution, L_0 was chosen such that it was slightly larger than the mean of the distribution. As U_0 served the Stokes velocity corresponding to the chosen L_0. To perform integration in time we applied an embedded Runge-Kutta-Fehlberg method with adaptive step-size control.

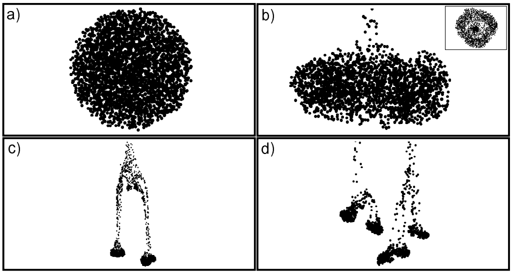

Figure 4.3: Characteristic stages during the sedimentation of a monodisperse particle cloud with initial particle volume fraction of 5%. Particles in c) and d) have been enlarged by a factor of 3 for better visibility. The vertical tail of leaked particles is not shown.

4.3.2 Evolution of a Bidisperse Cloud

In order to investigate the nature of the cloud evolution for a bidisperse particle system we have investigated two different kinds of initial conditions – unmixed bidisperse clouds, where the fraction of large particles is only distributed in the upper hemisphere of the cloud, and well-mixed clouds, where no restriction to the location of the particles inside the cloud has been imposed.

Unmixed Bidisperse Cloud

Figure 4.4 shows the characteristic first stages of a bidisperse particle cloud consisting of 1000 large and 1000 small particles, where the particles are placed according to a uniform random distribution and a given volume fraction ϕ inside the initially spherical cloud. The restriction in this study is that large particles are only placed in the upper hemisphere and small particles only in the lower hemisphere. We chose this initial configuration because it promotes mixing of the two fractions during sedimentation more than other comparable configurations. As depicted in figure 4.4 the fraction of large particles forms a cloud, dives through the

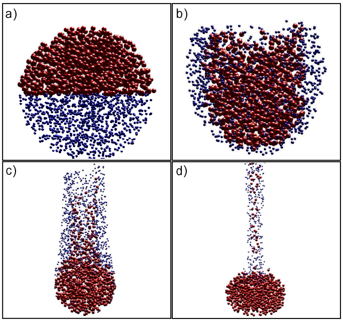

Figure 4.4: Initial particle configuration in an unmixed cloud and subsequent stages. The radius ratio of small to large particles is $\lambda = 0.5$, the initial volume fraction of particles inside the spherical cloud is $\phi = 0.05$ and the total number of particles is $N = 2000$.

portion of small particles, and takes a number of small particles with it. Figure 4.5 shows that the higher the radius ratio λ of small to large particles, the more small particles get carried along by the fraction of large particles. For values of λ lower than 0.5 the difference between the resulting curves is not as distinct as it is for higher radius ratios. The smaller a particle is, the less impact it has hydrodynamically on a larger particle. This effect is reflected by the entries of the mobility matrix $\boldsymbol{\mathcal{M}}^{\infty}$ in equation (4.3). Furthermore, as gravity scales with the third power of the particle radius, the driving force for the settling of particles is lower for a smaller particle size. These two factors are the main reasons leading to the described behaviour. Even though smaller particles have less hydrodynamical impact on large particles, the many small particles left behind for e.g. a radius ratio of $\lambda = 0.1$ can trap large particles, which then cannot settle with the cloud anymore. This effect can be seen in figure 4.6, where the evolution of the number of large particles $N_{\mathrm{p,large}}$ is shown. With increasing radius ratio more small particles stay in the cloud such that e.g. at $\lambda = 0.9$ no trapping occurs. Figures 4.5 and 4.6 show the number of particles inside the cloud until the first break-up event. At the point of break-up most small particles still trapped inside the particle cloud are freed and start to settle slower.

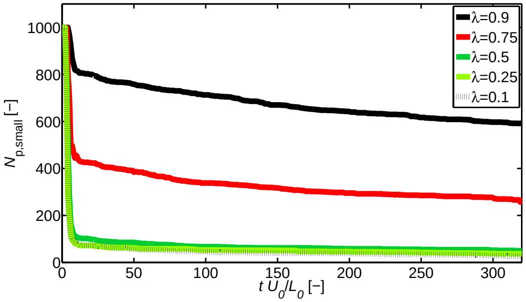

Figure 4.5: Number of small particles $N_{\mathrm{p,small}}$ inside an initially unmixed settling cloud of $N = 2000$ particles for an initial volume fraction $\phi = 0.075$.

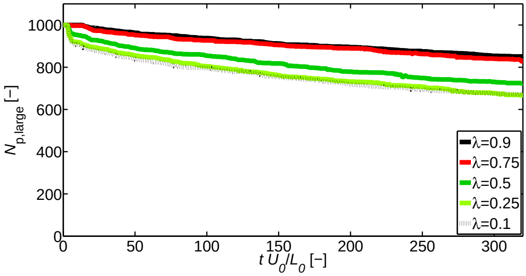

Figure 4.6: Number of large particles $N_{\mathrm{p,large}}$ inside an initially unmixed settling cloud of $N = 2000$ particles for an initial volume fraction $\phi = 0.075$.

Mixed Bidisperse Cloud

We have conducted numerical experiments on the probability of a break-up of a bidisperse particle cloud. This process is shown for the monodisperse case in figure 4.3 and has been reported in the mentioned literature ([Nitsche and Batchelor, 1997, Metzger et al., 2007]). Differing from the numerical experiments described in the last subsection, particles inside the initial cloud are now well mixed. We simulated the settling process of particle clouds of 1000 particles with a radius ratio of $\lambda = 0.5$ and a volume fraction of $\phi = 0.05$ in the initially spherical cloud, while increasing the percentage ν of small particles in the initial cloud in steps of 5% from 0% to 95%. In ten out of ten runs the described clouds form a torus and then break up into smaller clouds for all values of ν. The time of the break-up varies, depending on the exact initial particle configuration. Moreover, the break-up of the particle cloud itself depends on the exact initial positions of the particles. Figure 4.7 a) shows the initial configuration of a well-mixed particle cloud with radius ratio $\lambda = 0.5$ and an initial volume fraction $\phi = 0.05$. Figures 4.7 b)-d) show snapshots of the break-up for three different initial particle configurations. The only differences between the clouds shown in figures 4.7 b)-d) are the exact initial positions of the particles inside the spherical cloud domain, which are determined by a Monte Carlo method. Differing from the monodisperse case, a polydisperse particle cloud does not always break up into two smaller approximately evenly sized clouds. Instead it can break up into one large and another small cloud, into two similar clouds, or even into three clouds. The break-up into more than two clouds has thus far only been observed for break-up events at higher Reynolds numbers [Bosse et al., 2005] and in experiments as well as in numerical studies using a point-particle model [Myłyk et al., 2011]. In [Myłyk et al., 2011] polydispersity was not analyzed explicitly. In reality, particle systems are never truly monodisperse. Our numerical results show that the polydispersity seems to have an effect on the settling process. The fraction of large particles determines the motion of the cloud up to $\nu \approx 75\%$. At this value the dominating fraction was indeterminable in our numerical experiments. For percentages higher than 75%, the fraction of small particles seems to determine the cloud behaviour. By the expression 'determine the cloud behaviour' we mean that the respective fraction forms a cloud and drags the particles of the other fraction with it. The evolution of the number of small particles inside the settling cloud, $N_{\mathrm{p,small}}$, for the test runs for which the first break-up event is shown in figure 4.7 b)-d) is displayed on the right hand side of the figure. Compared with unmixed clouds (cf. figure 4.5) the loss of small particles in the initial phase is not as pronounced. More small particles remain in the cloud until the first break-up event.

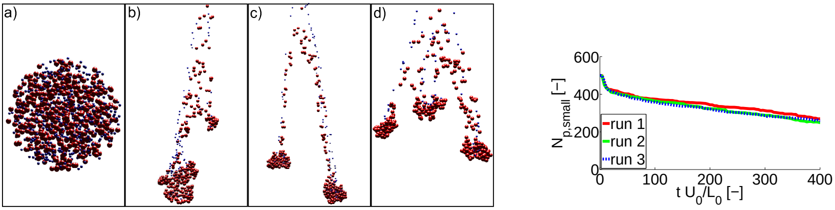

Figure 4.7: Left: Initial particle configuration for $\phi = 0.05$ in a mixed cloud (a) and snapshots of the break-up event for different initial configurations (b) – (d). Right: Number of small particles $N_{\mathrm{p,small}}$ inside the clouds shown in (b) – (d) until the first break-up. Run 1 corresponds to b), run 2 to c) and run 3 to d). The total number of particles is $N = 1000$.

4.3.3 Evolution of a Fully Polydisperse Cloud

In order to further investigate the behavior of polydisperse particle clouds, we have conducted numerical experiments on fully polydisperse systems with a particle radius distribution according to a given log-normal distribution. The considered clouds consisted of 1000, 2500 and 5000 particles, respectively, with particle and fluid properties such that the resulting Reynolds number and Stokes number were sufficiently small, i.e. the assumptions $\mathrm{Re} \approx 0$ and $\mathrm{St} \approx 0$ hold. For a particle volume fraction of $\phi = 0.01$ in a spherical cloud of 5000 particles, the characteristic stages of the cloud evolution are shown in figure 4.8. The depicted particle system has a radius distribution with resulting dimensionless expected value $\mu \approx 0.72$ and standard deviation $\sigma \approx 0.15$ (corresponding to the lognormal distribution $\mathcal{LN}(-0.35, 0.04)$). Abade and Cunha [2007] state that polydisperse particle clouds are less stable than comparable monodisperse clouds. They disintegrate before a torus can be formed. The clouds investigated in their work consisted of 300 particles. According to Metzger et al. [2007] this number of particles is also in the monodisperse case too low for the cloud to undergo the characteristic evolution, whereas the likelihood for clouds comprised of 1000 and more particles is about 100%. Our results show that also polydisperse clouds undergo this evolution, if they consist of enough particles.

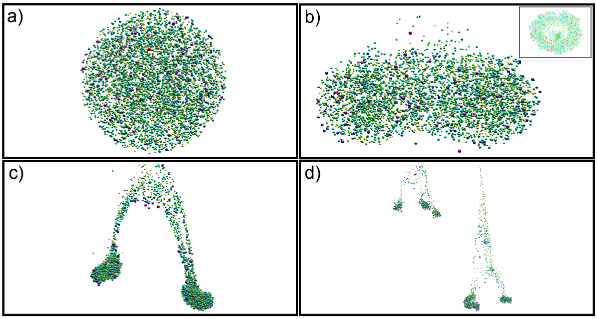

Figure 4.8: Characteristic stages during the sedimentation of a polydisperse particle cloud of $N = 5000$ particles. Particles in c) and d) have been enlarged by a factor of 3 for better visibility.

4.3.4 Cloud Settling Velocity

For a validation of formula (4.6) we have computed the mean of the particle velocities in our numerical experiments at a stage where the particle cloud still forms a sphere. At this time no particles have left the cloud and the cloud attains its maximum settling velocity. The error bars along the curves in figures 4.9, 4.10 and 4.12 result from the uniform random placement of particles inside the initial cloud, which influences the cloud's settling velocity. However, this influence is small compared to the total settling velocity of the cloud, as one can see by the width of the error bars. Their width goes to zero as the number of small particles in the cloud is increased, which indicates that the exact placement of particles in the cloud loses importance with a growing number of small particles. For every set of parameters we have carried out four simulations to obtain the error bars.

Velocity of Polydisperse Clouds

To cover a wide spectrum of possible particle systems, we have run extensive numerical tests on the already described well-mixed bi- and polydisperse particle systems. While we varied the percentage of small particles in bidisperse particle systems with fixed volume fraction, we varied ϕ in case of fully polydisperse particle systems. Tested values for ϕ are 0.01, 0.025, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.15 and 0.2. The radius distribution was chosen for all simulations such that the resulting dimensionless expected value was $\mu \approx 0.72$. Together with a standard deviation of $\sigma \approx 0.15$ this yields a reasonably polydisperse radius spectrum (the corresponding distribution is $\mathcal{LN}(-0.35, 0.04)$). A comparison of the cloud settling velocity of a bidisperse cloud, obtained from our numerical simulations, and the corresponding value from equation (4.6) is depicted in figure 4.9. The volume fraction of particles in the cloud was $\phi = 0.05$ with a total number of $N = 1000$ particles. The error bars showing the deviation due to different initial configurations in runs with the same set of parameters show that there is a statistical component. Yet, the agreement of prediction and numerical experiment is very good for all radius ratios λ and all values of ν. A comparison of our numerical results and

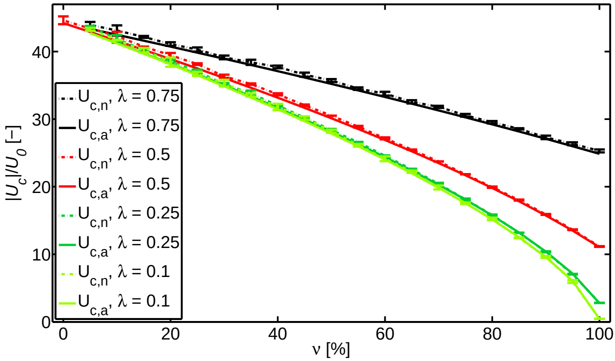

Figure 4.9: Comparison of values for the settling velocity of a bidisperse cloud obtained from our numerical simulations, $U_{c,n}$, and the prediction $U_{c,a}$ from formula (4.6). Here $\phi = 0.05$ and $N = 1000$. The settling velocities have been nondimensionalized by the Stokes velocity U_0 of a particle of the fraction of large particles.

the proposed prediction (4.6) for the case of a fully polydisperse cloud is depicted in figure 4.10. For these results we have varied the number of particles per cloud, N, and the solid volume fraction ϕ in the clouds. The deviation of the cloud settling velocity from the computed mean is extremely low for all tested N and ϕ. Furthermore, figure 4.10 shows that prediction and numerical results agree very well in case of low values of ϕ, what corresponds to a dilute cloud. The deviation between the corresponding curves increases for higher volume fractions. This is due to the assumption of a dilute cloud, which has been made in the derivation of formula (4.6). Nevertheless, this formula still gives a good prediction, even for solid volume fractions of 20%.

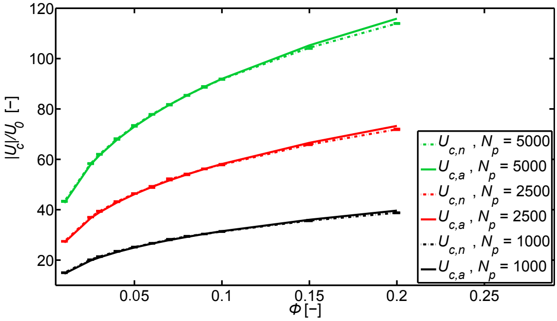

Figure 4.10: Comparison of the particle cloud settling velocity for varying ϕ and different N in a *fully polydisperse* cloud. $U_{c,n}$ denotes values from the numerical simulation, $U_{c,a}$ the prediction from formula (4.6). U_0 has been defined in section 4.3.1.

Velocity of a Monodisperse Cloud

In addition to the comparisons of analytical prediction and numerical results presented in figures 4.9 and 4.10 we have run simulations of monodisperse clouds with a varying number of particles N and a varying initial solid volume fraction ϕ in the clouds. The tested number of particles and initial volume fractions are the same as in the fully polydisperse case (cf. section 4.3.4, figure 4.10). As in the polydisperse case we obtain a good agreement of the prediction from formula (4.6) and our numerical results, as one can see in figure 4.11. Yet contrary to the polydisperse case, the additional U_{St}-term in equation (4.5) would increase the agreement of the analytical prediction by formula (4.6) and the numerically computed cloud settling velocity.

Using Formulas Proposed for Monodisperse Systems for Polydisperse Systems

The use of e.g. the mean particle radius together with a formula proposed for monodisperse clouds leads to an error, which can be significant when

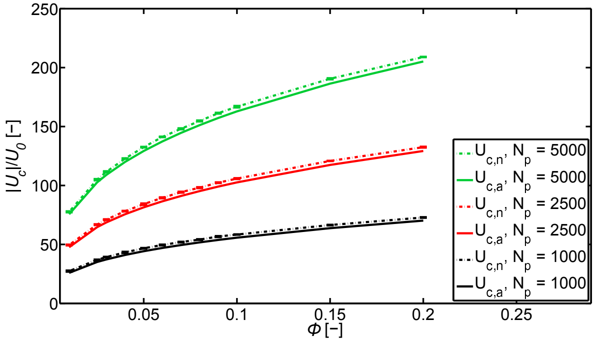

Figure 4.11: Comparison of the particle cloud settling velocity for varying ϕ and different N in a *monodisperse* cloud. $U_{c,n}$ denotes values from the numerical simulation, $U_{c,a}$ the prediction from formula (4.6). Particle velocities have been nondimensionalized by the Stokes velocity U_0 of a single particle.

polydisperse clouds are considered. Figure 4.12 shows our numerical results compared to values obtained from formula (4.6), once considering the given polydisperse radius distribution and another time assuming a monodisperse particle system. For the latter we use the mean particle radius of the polydisperse particle system described in section 4.3.4 together with formula (4.6) which in this case differs from equation (4.5) only in the additional term equal to the settling velocity of a single particle (cf. section 4.2.2). When polydispersity is not considered, the equation gives for a polydisperse system of 5000 particles a resulting cloud velocity with a relative deviation from the numerical value of 10% and more for all volume fractions ϕ, while equation (4.6) considering polydispersity agrees very well with the value obtained from the simulation. Therefore, an equation derived for monodisperse particle systems, like equation (4.5), should only be used for truly monodisperse particle clouds. If the particle size fractions of the system are known or can be approximated, formula (4.6) which considers polydispersity yields good estimates.

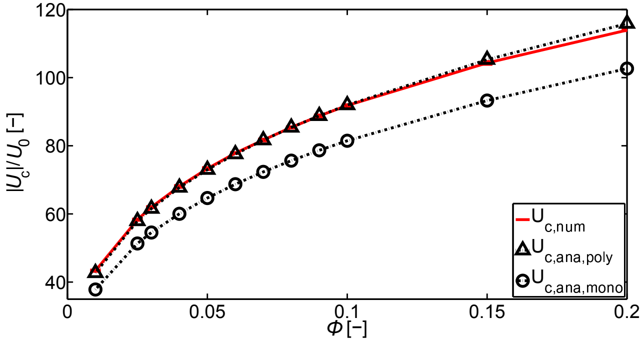

Figure 4.12: Comparison of different estimates for the settling velocity of a polydisperse cloud for $N = 5000$ and varying ϕ. $\mathrm{U}_{\mathrm{c,num}}$ denotes our numerical results, $\mathrm{U}_{\mathrm{c,ana,poly}}$ the use of formula (4.6) considering polydispersity, and $\mathrm{U}_{\mathrm{c,ana,mono}}$ the use of formula (4.6) assuming a monodisperse particle system. U_0 has been defined in section 4.3.1.

4.4 Conclusions

This work gives more insight into the sedimentation process of polydisperse particle clouds in viscous fluids. Until now research has been concentrated on the investigation and description of monodisperse particle clouds, drops or blobs, as they are sometimes called. Due to the fact that we have parallelized our algorithm, we are able to simulate particle systems of several thousand particles with any radius distribution in a short time. This enables us to investigate large polydisperse particle clouds, for which the evolution differs from that of relatively small particle clouds comprised of several hundred up to one thousand particles. One result is that, given that the considered particle cloud consists of a large number of particles, a polydisperse particle cloud undergoes the same remarkable motion as a sufficiently large monodisperse cloud does. An initially spherical cloud evolves into a torus that starts to oscillate until it breaks up into smaller clouds, which then undergo the same motion. For a smaller number of particles, namely about 1000 particles and less, polydisperse clouds are not as stable as their monodisperse counterpart. In an extreme case of polydispersity, a bidisperse system, the evolution is dominated by the fraction with the highest number of particles in the system. Due to a loss of particles during sedimentation, initially spherical particle clouds attain their maximum settling velocity while they still have their initial shape. Starting from the Hadamard-Rybczynski equation we have derived a formula which yields the settling velocity of such a particle cloud. By a comparison with numerical experiments we have shown that the proposed formula gives a very good estimate for the settling velocity of a polydisperse spherical particle cloud.

Appendix to chapter 4: Derivation of the Cloud Settling Velocity

We start from the Hadamard-Rybczynski equation (H-R equation),

$$U_c = \frac{2}{3} \frac{(\rho_f - \rho_c)\, a_c^2 g}{\mu_f} \frac{\lambda + 1}{3\lambda + 2}, \tag{4.7}$$

with the radius of the spherical cloud, a_c, the dynamic fluid viscosity μ_f and $\lambda = \mu_c/\mu_f$, the ratio of dynamic viscosity of the cloud to dynamic fluid viscosity. ρ_f and ρ_c are the mass density of the fluid and the mass density of the cloud, respectively. g denotes the gravitational constant. Utilizing $(\rho_f - \rho_c) = \phi(\rho_f - \rho_p)$, where ϕ is the volume fraction of the particles with mass density ρ_p in the spherical cloud, the H-R equation is equivalent to

$$U_c = \frac{2}{3} \frac{\phi\, (\rho_f - \rho_p)\, a_c^2 g}{\mu_f} \frac{\lambda + 1}{3\lambda + 2}. \tag{4.8}$$

If ϕ and λ are known, this is the equation we are looking for. Usually λ is not known a priori. Making the assumption $\phi \ll 1$ yields $\lambda \approx 1$. Together with

$$\phi = \sum_{i=1}^{N_f} N_{p,i} \left(\frac{a_{p,i}}{a_c} \right)^3, \tag{4.9}$$

this yields

$$U_c = \frac{6}{5}\sum_{i=1}^{N_f} N_{p,i}\frac{a_{p,i}}{a_c}U_{St,i}. \tag{4.10}$$

Here,

$$U_{St,i} = \frac{2}{9}\frac{(\rho_f - \rho_p)a_{p,i}^2 g}{\mu_f} \tag{4.11}$$

is the Stokes velocity of a spherical particle with radius $a_{p,i}$. $N_{p,i}$ is the number of particles in one of the N_f particle size fractions of the particle system.

Chapter 5

Numerical simulation of orthokinetic heterocoagulation and cluster growth in destabilizing suspensions

Abstract
In this work we investigate the coagulation behaviour of non-Brownian colloidal particles by means of direct numerical simulation on the example of Al_2O_3 particles. This yields the so-called capture efficiency, for which we give an analytical expression, as well as other time-dependent values such as the cluster growth rate. Instead of neglecting or strongly approximating hydrodynamic interactions between particles, we include hydrodynamic and non-hydrodynamic interactions in a Stokesian Dynamics approach and through comprehensive modeling of interparticle forces. The resulting parallelized simulation framework enables us to investigate the dynamics of polydisperse particle systems comprised of several hundred particles at the same high level of modeling we use for a close investigation of the coagulation behaviour of two unequal particles in shear flow. An appropriate cluster detection yields all the information about large destabilizing systems, which is needed for models used in flow-sheet simulations. Due to nondimensionalization the results can be generalized and applied to other systems tending to secondary coagulation.

This chapter follows our work of the same title under consideration for publication in the journal Particuology.

5.1 Introduction

Destabilization of suspensions in shear flow and during sedimentation are important topics in all kinds of suspension-related engineering, such as chemical engineering, process engineering, civil engineering, or wastewater treatment. Numerical simulations are a valuable tool for the investigation of fundamental mechanisms during destabilization, and for the design of coagulation-related processes, i.e. aggregation, agglomeration, flocculation. Due to progress in modeling, code parallelization and the usage of supercomputers, increasingly complex systems can be simulated within an acceptable timespan. Yet, the larger and the more complex the simulated system is, the more modeling has to be done. Population balance equation (PBE) based simulations are an appropriate technique for the simulation of coagulation-related processes on industrial scale [Gerstlauer et al., 2006]. They are mainly applied to investigate and to predict aggregation and agglomeration in stirred tank vessels [Hollander et al., 2001], but can also be applied to complex solid-liquid separation processes, such as the settling of activated sludge in a secondary clarifier [Torfs et al., 2012]. PBE models heavily rely on coagulation- and breakup kernels, which give the probability of coagulation and possible breakup of aggregates, agglomerates or flocs. If the used model does not describe the physics of the considered system well enough, the prediction by such a simulation is rather poor [Torfs et al., 2012]. Apart from giving a better understanding of the underlying physics, direct numerical simulations can provide the models for strongly model-based simulations, such as PBE, but also for analytic models used in flow-sheet simulations. There is a large number of recent works dealing with the simulation of agglomerate- and aggregate breakup and deformation [Dosta et al., 2013, Harshe and Lattuada, 2011, Seto et al., 2012, Conchuir et al., 2014, Higashitani et al., 2001, Zeidan et al., 2007, Manounou and Rémond, 2014], as well as several works considering the modeling and simulation of coagulation [Feke and Schowalter, 1983, Smoluchowski, 1917, Curtis and Hocking, 1970, Van de Ven and Mason, 1976], but only a limited number of works considering coagulation and settling at the same time [Qiao et al., 1998, Davis, 1984, Han and Lawler, 1991]. Work in the field of modeling and prediction of the coagulation probability can be classified by means of the level of approximation [Khadilkar et al., 2014], as well as by the investigated type of coagulation. There is a differentiation between perikinetic, orthokinetic and differential settling coagulation. Perikinetic coagulation is mainly driven by Brownian motion, orthokinetic coagulation by the flow. This means mostly shear flow, but depending on the definition it can also include settling [Somasundaran, 2006]. Coagulation due to differential settling is coagulation owing to the fact that in general a larger or denser particle settles faster than a smaller or less dense particle. Technically speaking, this effect differs from flow driven coagulation, as here the driving force is a volume force and not hydrodynamic forces, which are surface forces. A distinction by the Peclet number on the other hand yields only two groups — inertia dominated coagulation and Brownian motion dominated coagulation. Of course, there is a transition regime and on a technical scale coagulation due to flow and that due to differential settling are almost always superimposed. Therefore we leave it up to the reader to make an own distinction, as further discussion would go beyond the scope of this work. In this work we concentrate on non-Brownian coagulation and call it orthokinetic coagulation.

A classical way to predict orthokinetic coagulation in suspensions is the consideration of two particles in a simple shear flow, from which a derivation of the so-called orthokinetic collision rate is possible. This can be done e.g. based on conditional pair distribution functions (usually done if Brownian motion plays a role, see e.g. [Feke and Schowalter, 1983, Qiao et al., 1998]) or by a direct investigation of the trajectories of two intercepting particles. In his famous work, Smoluchowski [Smoluchowski, 1917] determined the collision rate of colloidal particles based on quite simplifying assumptions. Among them are the assumption of negligible hydrodynamic interactions and of attraction spheres around each particle, where two particles coagulate instantly when their spheres overlap. Nevertheless, the resulting collision rates are remarkably good and are in various forms widely in use. Usually the collision rates by Smoluchowski are modified by a capture efficiency, which is defined as ratio of observed collision rate to the collision rate proposed by Smoluchowski [Van de Ven and Mason, 1977]. There is quite a large number of works using a two-particle approach in bispherical coordinates in order to obtain the capture efficiency for two spheres in a simple shear flow [Van de Ven and Mason, 1977, Adler, 1981, Wang, 1992, Vanni and Baldi, 2002, Kobayashi, 2008, Balakin et al., 2012, Sato et al., 2004, Lin et al., 2006]. The resulting orthokinetic capture efficiency can be applied to compute the collision rate used in PBE simulations. But the applicability of simulations using bispherical coordinates is limited to a consideration of only two particles. Using a meshfree simulation approach for hydrodynamic interactions between particles [Brady and Bossis, 1988, Durlofsky et al., 1987] and detailed modeling of other occurring effects, such as lubrication [Davis et al., 1986] and interparticle interactions covered by DLVO theory [Verwey et al., 1999, Feke et al., 1984], allows us a close investigation of the coagulation behaviour of two particles as well as studies on destabilization of particle systems with a large number of settling particles [Bülow et al., 2014, 2015]. In this work we show that both ways yield results for the coagulation probability, the collision rate and the cluster growth rate, which are essentials for large-scale simulations by means of PBE and simplified methods such as flow-sheet simulations.

The structure of this work is as follows: First we give a quick overview of the method we use to simulate the behaviour of suspensions and we present the used models for particle-particle interactions. Here, we focus on a new model for lubrication forces between particles at almost-contact. The presented results from numerical investigations in the binary orthokinetic heterocoagulation of Al_2O_3 particles in shear flow lead to an analytical expression for the orthokinetic capture efficiency in convenient form. Furthermore, we show how effort can be reduced when doing similar numerical experiments. We conclude this work by giving results from a cluster detection applied to the simulation data from numerical experiments on destabilizing systems of suspended particles. This procedure yields, besides comprehensive insight into the process, useful data such as cluster growth rates and other information needed for flow-sheet simulations.

5.2 Methods

5.2.1 Underlying equations for the particle motion

We consider the motion of N particles for which the effects of Brownian motion is negligible, while the effect of interparticle forces as well as hydrodynamics is considerable. If these particles settle in an otherwise undisturbed fluid, or they are exposed to a simple shear flow with low to moderate shear rate $\dot{\gamma}$, the quasi-steady Stokes equations can describe the fluid motion appropriately. The translational motion of a spherical particle α suspended in a fluid can generally be described by the dimensionless Langevin equation [Brady and Bossis, 1988, Ichiki and Hayakawa, 1995],

$$\mathrm{St}\left(\frac{a_\alpha}{L_0}\right)^3 \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_\alpha = \boldsymbol{F}^h_\alpha - \left(\frac{a_\alpha}{L_0}\right)^3 \boldsymbol{e}_z + \boldsymbol{F}^{\mathrm{i,St}}_\alpha + \frac{1}{\mathrm{Pe}}\boldsymbol{F}^{\mathrm{i,Pe}}_\alpha, \tag{5.1}$$

with the Stokes number St defined as the mass density ratio scaled Reynolds number Re,

$$\mathrm{St} = \frac{2}{9}\frac{\rho_p}{\rho_f}\mathrm{Re}, \tag{5.2}$$

and the vector $\boldsymbol{e}_z$ pointing in positive z-direction (w.l.o.g. the applied centrifugal force points in negative z-direction). The dimensionless angular momentum balance is given by

$$\mathrm{St}\frac{2}{5}\left(\frac{a_\alpha}{L_0}\right)^5 \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega_\alpha} = \boldsymbol{T}^{\mathrm{h}}_\alpha + \boldsymbol{T}^{\mathrm{i,St}}_\alpha + \frac{1}{\mathrm{Pe}}\boldsymbol{T}^{\mathrm{i,Pe}}_\alpha. \tag{5.3}$$

In equations (5.1) and (5.3), $\boldsymbol{U}_\alpha$ is the translational velocity of particle α, $\boldsymbol{F}^h_\alpha$ the hydrodynamic force acting on the particle, $\boldsymbol{F}^{\mathrm{i,St}}_\alpha$ interparticle forces of the same scale as $\boldsymbol{F}^h_\alpha$, and $\boldsymbol{F}^{\mathrm{i,Pe}}_\alpha$ interparticle forces which need to be scaled by the Peclet number Pe. $\boldsymbol{\omega_\alpha}$ denotes the angular velocity of particle α and $\boldsymbol{T}^{\mathrm{h}}_\alpha$ the hydrodynamic torque acting on the particle. $\boldsymbol{T}^{\mathrm{i,St}}_\alpha$

and $\boldsymbol{T}_\alpha^{\mathrm{i,Pe}}$ are torques corresponding to the respective forces. Assuming a constant particle mass density ρ_p, the mass m_α of a spherical particle α is $m_\alpha = 4/3\pi\rho_\mathrm{p} a_\alpha^3$, with the particle radius a_α. With the same assumption the moment of inertia tensor $\boldsymbol{I}_\alpha$ writes as $\boldsymbol{I}_\alpha = 8/15\pi\rho_\mathrm{p} a_\alpha^5 \mathbf{Id}$, with the 3×3 identity matrix $\mathbf{Id}$. For nondimensionalization we used the characteristic length L_0, a characteristic force scale $F_0 = 6\pi\mu_f L_0 U_0$, and the characteristic velocity U_0, the Stokes velocity of a particle of radius L_0 in the applied centrifugal field.

The dimensionless equations (5.1) and (5.3) contain two dimensionless numbers, St and Pe. The Stokes number St of a monodisperse system of spherical particles, i.e. $a_\alpha = L_0$ for all $\alpha \in [1, N]$, is given in equation (5.2). In case of a polydisperse system, with L_0 chosen as the mean value of the radius distribution, one can define a Stokes number specific to a particle α,

$$\mathrm{St}_\alpha = \left(\frac{a_\alpha}{L_0}\right)^3 \mathrm{St} = \left(\frac{a_\alpha}{L_0}\right)^3 \frac{2}{9}\frac{\rho_p}{\rho_f}\mathrm{Re} = \frac{m_\alpha U_0^2}{6\pi\mu_f L_0^2 U_0}.$$

The variables ρ_f and μ_f are the mass density and the dynamic viscosity of the fluid, respectively. The Peclet number is defined by

$$\mathrm{Pe} = \frac{6\pi\mu_f L_0^2 U_0}{k_\mathrm{B} T},$$

with the Boltzmann constant k_B and the temperature T.

Let the background flow, i.e. the fluid velocity in the absence of particles, at position $\boldsymbol{x}$ in the fluid domain be given by the Stokes flow

$$\boldsymbol{u}^\infty(\boldsymbol{x}) = \boldsymbol{U}^\infty + \boldsymbol{\Omega}^\infty \times \boldsymbol{x} + \boldsymbol{E}^\infty \boldsymbol{x}, \tag{5.4}$$

where $\boldsymbol{U}^\infty$ and $\boldsymbol{\Omega}^\infty$ are the translational and angular velocity of the undisturbed Stokes flow and $\boldsymbol{E}^\infty$ the respective rate of strain. The Stokesian Dynamics (SD) method arises from a truncated multipole expansion of the integral representation for the fluid velocity in Stokes flow with present particles and background flow as given in equation (5.4). This multipole expansion yields a linear relation between particle velocities and hydrodynamic forces, torques and stresses. The linear relation between these variables is given by the so-called mobility matrix, the corresponding system of linear equations is called mobility problem [Bülow et al., 2014, Durlofsky et al., 1987]. The mobility matrix reflects the hydrodynamic interactions among all suspended particles and can be computed without explicit knowledge of the fluid velocities. These are treated implicitly by the multiplication of mobility matrix and the vector of forces, torques and stresses — in short FTS-vector. Together with equations (5.1) and (5.3) one obtains a method for the time-dependent simulation of the motion of particles suspended in a fluid. For a more detailed description of the method and the parallelization techniques we use, we refer to [Bülow et al., 2014]. For the problems investigated in this work we can assume a vanishing Stokes number [Curtis and Hocking, 1970]. For more information and a discussion on the consequences of the assumption St ≈ 0 we refer again to [Bülow et al., 2014].

5.2.2 Interparticle forces

For the results presented in this work we employ several models to account for interparticle forces of non-hydrodynamic nature and a model for the part of the hydrodynamic force, which is not considered by the SD approach for St ≈ 0. Together these models cover the almost-contact of particles, particle contact and interactions due to DLVO interactions. The implemented model for interactions included in DLVO theory [Verwey et al., 1999], which yields interparticle forces arising from London–van der Waals (LvdW) attraction, electrostatic repulsion and Born repulsion, as well as the applied Hertz-contact model can be found in [Bülow et al., 2014]. In the following we describe the adapted lubrication model we use. This model accounts for hydrodynamic effects between almost touching particles, which are not included in the mobility matrix.

Lubrication Model

The approximations of hydrodynamic forces and torques arising from the assumption of a vanishing Stokes number can yield a large error in particle velocities when particles are almost touching (see e.g. [Ichiki, 2002] for a method which includes higher moments in the previously mentioned multipole expansion, as well as [Bülow et al., 2015] for a quantitative evaluation of the resulting deviation). As illustrated in figure 5.1, the lubrication effect results from the squeezing motion of approaching particles exerted on the fluid in the gap between the particle surfaces as well as from the fact that separating particles need to overcome a negative pressure gradient between the surrounding fluid and the widening gap between their surfaces.

Classical lubrication models [Jeffrey and Onishi, 1984, Jeffrey, 1992, Cichocki et al., 1999] depend on the computation of an additional lubrication correction matrix of the same dimension as the mobility matrix. Furthermore they require the solution of the mobility problem, i.e. the solution of a system of linear equations with a dense matrix of dimension $\mathbb{R}^{c_{\text{lev}}N \times c_{\text{lev}}N}$, with $c_{\text{lev}} \in \{3, 6, 11\}$ depending on the level of approximation [Bülow et al., 2014]. Because of high numerical costs, the coefficients of the expansion yielding the entries of the additional lubrication correction matrix are often stored in a table prior to a simulation. Especially for systems of high polydispersity this technique produces an immense amount of data. The coefficients need to be computed for a large number of interparticle distances and all radius ratios. Moreover, storage of the coefficients produces additional communication costs in case of a parallelized implementation with dynamical sorting of particle data as in

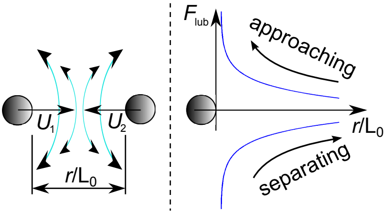

Figure 5.1: Left: Illustration of two particles squeezing fluid out of the gap between their surfaces as they approach each other. Right: Schematic progress of the resulting lubrication force for a constant relative velocity of the two particles.

[Bülow et al., 2014]. Therefore the coefficients should be computed on the fly, which would make a simulation with a classical lubrication scheme computationally inefficient.

Apart from the computational drawbacks, classical lubrication schemes assume the fluid in an interparticle gap to behave as an incompressible continuum for all surface-to-surface distances. This assumption results in an infinitely strong lubrication force (repulsive as well as attractive) for almost touching particle surfaces, what is sometimes called Stokes paradox [Zhang et al., 2005] and can be avoided by appropriate modeling. Especially for micron sized particles and nanoparticles separated by a thin interstitial gap, slip occurs due to non-continuum effects and as Hocking [1973] states, the Stokes equations are not valid when the interparticle gap approaches molecular dimensions. Zhang et al. [2005] discuss this topic and propose a lubrication model which includes slip effects in a more rigorous force-based scheme. In their model, lubrication effects are taken into account by an inclusion of a mathematically justified extra force, just as in [Davis et al., 1986]. Note that Davis et al. [1986] particularly mention the assumptions which give rise to the paradox of non-touching particles and which are tackled in [Zhang et al., 2005]. Besides the fact that the model in [Zhang et al., 2005] reflects the physical behaviour of almost touching particles in a more accurate way, a force-based lubrication scheme has lower numerical costs than classical SD lubrication schemes, it does not necessarily require the solution of a system of linear equations and has generally lower communication costs, as sparse communication can be applied [Bülow et al., 2014]. For these reasons we chose to modify the model given in [Zhang et al., 2005] such that it takes polydispersity of the considered particle system and a transition regime between continuum and non-continuum (or no-slip and slip) regimes into account. Considering a transition regime is not only physically justified, as the number of fluid molecules normally does not abruptly decrease with decreasing interstitial gap width, it also produces a smoother force which is numerically beneficial.

We distinguish four regimes, which are valid at specific surface-to-surface distances r_0. These regimes are: A continuum regime where the fluid is treated as incompressible Newtonian fluid (i), a regime where slip dominates (ii), a transition regime where the lubrication force gradually transitions from the continuum expression to the expression with occurring slip (iii), and a regime where surface roughness and DLVO forces dominate (iv). Figure 5.2 visualizes the resulting effective lubrication force $\boldsymbol{F}^{\text{lub, eff}}$ with its contributions $\boldsymbol{F}^{\text{lub, no-slip}}$ and $\boldsymbol{F}^{\text{lub, slip}}$ in the three regimes i)–iii). Also shown in figure 5.2 are the boundaries of the respective regimes. The no-slip regime (i) is valid for surface-to-surface distances larger than a specified distance $d_{\text{l,trans}}$. The transition regime (iii) lies between the lower bound $d_{\text{l,slip}}$ and the upper bound $d_{\text{l,trans}}$. Further indicated in figure 5.2 is the lower boundary $d_{\text{c,min}} < d_{\text{l,slip}}$ of the slip regime (ii) below which regime (iv) starts, where surface roughness and DLVO forces are dominant (cf. [Bülow et al., 2014]). In the following we give the expression for the lubrication force in the different regimes.

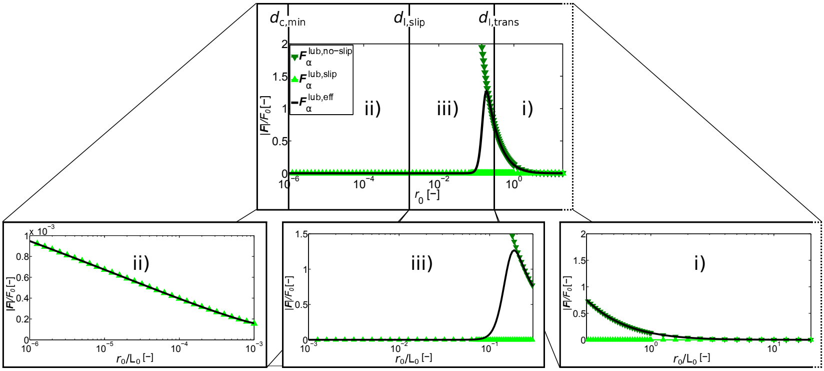

Figure 5.2: Dimensionless effective lubrication force $\boldsymbol{F}_\alpha^{\text{lub, eff}}$ with its contributions $\boldsymbol{F}_\alpha^{\text{lub, no-slip}}$ and $\boldsymbol{F}_\alpha^{\text{lub, slip}}$ in the three regimes i)–iii) for dimensionless values of $d_{\text{c,min}} = 10^{-6}$, $d_{\text{l,slip}} = 10^{-3}$, $d_{\text{l,trans}} = 0.3$ and a constant dimensionless relative velocity $\boldsymbol{v}_{\alpha\beta} = 1$.

i) Lubrication force in the no-slip regime, $r_0 \geq d_{\text{l,trans}}$:

Following [Davis et al., 1986, Zhang et al., 2005] we get an expression for the lubrication force acting on a particle α along the line through the centers of particle α and another particle β in the no-slip regime. This expression is given by

$$\boldsymbol{F}_\alpha^{\text{lub, no-slip}}(r_0, a_{\text{red}}, \boldsymbol{v}_{\alpha\beta}) = k_1^{\text{lub}} 6\pi\mu_f a_{\text{red}}^2 \left(\frac{1}{r_0} - \frac{1}{r_0 + 2a_{\text{red}}}\right) \boldsymbol{v}_{\alpha\beta}. \quad (5.5)$$

Here, the shortest distance $r_0 := r_{\alpha\beta} - (a_\alpha + a_\beta)$ between the surfaces of two interacting particles α and β with center-to-center distance $r_{\alpha\beta} = \left\| \boldsymbol{x}_\alpha - \boldsymbol{x}_\beta \right\|_2$ has been used. $\boldsymbol{v}_{\alpha\beta}$ is the relative velocity of the two particles. The reduced radius a_{red} is defined by

$$a_{\text{red}} := a_\alpha a_\beta / \left(a_\alpha + a_\beta\right), \tag{5.6}$$

where the radii a_α and a_β are the radii of two interacting particles α and β.

In order to improve the accuracy of their analytical expression for the lubrication force in the no-slip regime, Zhang et al. [2005] have carried out numerical computations where they solved the equation for the pressure distribution in the interparticle gap. As a result they obtained a correction factor k_1^{lub} for the analytical expression, with which the deviation between corrected analytical expression and numerical solution is below 1.5%. This factor is given by

$$k_1^{\text{lub}}(r_0, a_{\text{red}}) = 1.041 - 0.281 \log\left(\frac{r_0}{2a_{\text{red}}}\right) - 0.0351 \log^2\left(\frac{r_0}{2a_{\text{red}}}\right).$$

The resulting expression for the lubrication force acting on a particle α due to interaction with another particle β in the no-slip regime is then given by $\boldsymbol{F}_\alpha^{\text{lub, no-slip}}$ as defined in equation (5.5).

ii) Lubrication force in the slip regime, $d_{\text{c,min}} \leq r_0 < d_{\text{l,slip}}$:

As the model in [Zhang et al., 2005] we account for non-continuum effects in such a way that for surface-to-surface distances $d_{\text{c,min}} \leq r_0 < d_{\text{l,slip}}$ we use the expression

$$\begin{aligned}\boldsymbol{F}_\alpha^{\text{lub, slip}}&(r_0, a_{\text{red}}, \boldsymbol{v}_{\alpha\beta})\\ &= k_2^{\text{lub}} \frac{\pi\mu_f}{3d_{\text{l,slip}}} a_{\text{red}}^2 \Bigg((r_0 + 6d_{\text{l,slip}}) \ln(\frac{r_0 + 6d_{\text{l,slip}}}{r_0})\\ &-(r_0 + 2a_{\text{red}} + 6d_{\text{l,slip}}) \ln(\frac{r_0 + 2a_{\text{red}} + 6d_{\text{l,slip}}}{r_0 + 2a_{\text{red}}}) \Bigg) \boldsymbol{v}_{\alpha\beta},\end{aligned} \tag{5.7}$$

with the correction factor

$$k_2^{\text{lub}}(r_0, a_{\text{red}}) = 1.309 - 0.082 \log\left(\frac{r_0}{2a_{\text{red}}}\right) - 0.009 \log^2\left(\frac{r_0}{2a_{\text{red}}}\right).$$

Compared with [Zhang et al., 2005], the variable $d_{\text{l,slip}}$ does not stand for the mean free path of the fluid but for a value defining the slip regime, which can be seen as slip length. Equation (5.7) results from an inclusion of slip effects, which can have a non-negligible influence even in non-colloidal suspensions [Pieper et al., 2015].

iii) Lubrication force in the transition regime, $d_{\text{l,slip}} \leq r_0 < d_{\text{l,trans}}$:

In the interval where $d_{\text{l,slip}} \leq r_0 < d_{\text{l,trans}}$, we interpolate between $\boldsymbol{F}_\alpha^{\text{lub, no-slip}}$ and $\boldsymbol{F}^{\text{lub, slip}}$ by means of the error function erf and a function λ_{trans}, with

$$\lambda_{\text{trans}}(r_0) = 0.5 \left(\text{erf}(12 \frac{r_0 - d_{\text{l,slip}}}{d_{\text{l,trans}} - d_{\text{l,slip}}} - 6) + 1 \right), \tag{5.8}$$

such that we get

$$\boldsymbol{F}_\alpha^{\text{lub, trans}} = \lambda_{\text{trans}} \boldsymbol{F}_\alpha^{\text{lub, no-slip}} + (1 - \lambda_{\text{trans}}) \boldsymbol{F}_\alpha^{\text{lub, slip}}, \tag{5.9}$$

where we have dropped the arguments of the force functions for better readability.

The resulting effective lubrication force $\boldsymbol{F}^{\text{lub, eff}}$ is given by equations (5.5), (5.7) and (5.9), each valid in their interval. The function λ_{trans} has been defined such that in the transition regime the argument of the error function is a linear function of r_0 between -6 and 6. As one can see in figure 5.2 this results in a smooth interpolation with a deviation below machine accuracy for λ_{trans} at the boundaries of the transition regime. Without transition regime, the large difference in the values of $\boldsymbol{F}_\alpha^{\text{lub, no-slip}}$ and $\boldsymbol{F}_\alpha^{\text{lub, slip}}$ requires a strong reduction of the time-step width of the used time-stepping scheme, if an adaptive scheme such as the (4,5)-Runge-Kutta-Fehlberg method [Fehlberg, 1970] is used. For distances below $d_{\text{c,min}}$, which corresponds to r_0 in the regime (iv), we make use of the contact model

described in [Bülow et al., 2014]. The resulting force for every particle α is made dimensionless by the characteristic force scale F_0 and then added to the vector of interparticle forces $\boldsymbol{F}_{\alpha}^{\mathrm{i,St}}$ in equation (5.1).
The presented model extends the model derived in [Zhang et al., 2005], which is an extension of the model in [Davis et al., 1986]. That model can be seen as low level approximation to classical SD lubrication correction schemes as e.g. in [Jeffrey, 1992]. It takes into account the relative motion and the particle configuration and thus is superior to other force-based correction schemes, which only depend on particle positions (e.g. [Cunha et al., 2002, Nitsche and Batchelor, 1997]). The additional dependence on the relative velocity in equation (5.5) is not only mathematically justified [Davis et al., 1986], it reflects the attractive nature of the lubrication force during separation of almost touching particles. Even though a corresponding correction for the hydrodynamic torque and for higher moments is missing, this model yields a good lubrication correction as we show in section 5.3.

5.3 Results and discussion

In this section we present results dealing with basic coagulation behaviour and with the destabilization of large heterogeneous particle systems. All computations have been performed on the supercomputers HC3 [hc3, 2014] and IC2 [ic2, 2013] at our university. Used physical values are made dimensionless by a corresponding characteristic value. In case of a bimodal radius distribution the characteristic length L_0 is the amplitude of the right peak, for unimodal distributions we use the mean. The characteristic velocity U_0 is the corresponding Stokes velocity. Characteristic values for DLVO forces can be found in the subsection on nondimensionalization in [Bülow et al., 2014]. Validation and simulation results considering the motion of particles suspended in a fluid are given in [Bülow et al., 2014, 2015]. If not stated otherwise, we use the dimensionless parameters $d_{\mathrm{c,min}}$ = 1e-6, $d_{\mathrm{l,slip}}$ = 1e-3 and $d_{\mathrm{l,trans}} = 0.25$ for the lubrication model presented in subsection 5.2.2.
We present results yielding the orthokinetic capture efficiency α_o for polydisperse particle systems. As we will show, α_o depends strongly on the

shear rate $\dot{\gamma}$. The data is in dimensionless form to make the results more universal. A multiplication of the respective value of the shear rate by the problem-dependent inverse time scale U_0/L_0 yields the corresponding dimensioned value. The same holds for the already addressed lengths and velocities.

5.3.1 Lubrication effects and particle collision in a simple shear flow

We have tested the proposed lubrication model on examples given in [Pozrikidis, 2007], which have also been used in [Luo and Pozrikidis, 2007]. In the work by Pozrikidis [2007], the interception of two spherical particles in a simple shear flow is investigated by means of a boundary element method. Because of the assumption of an incompressible Newtonian fluid, which acts as continuum at all times, the considered particles never touch. In the work by Luo and Pozrikidis [2007] the authors allow for slip effects and find that there is a critical value of the slip coefficient (the scaled inverse slip length) below which the two intercepting particles collide after a finite interception time. This critical value is stated to depend on the relative initial particle positions in the considered numerical experiment. For particles initially almost lying in the plane of zero velocity of the simple shear flow, the authors of [Luo and Pozrikidis, 2007] expect the two particles to get in contact even for relatively high values of the slip coefficient. Since this coefficient is the inverse of the slip length, this means that particles get in contact even for small slip lengths. Luo and Pozrikidis [2007] further state that in reality attractive forces will cause the particles to form a temporary or a permanent doublet, which rotates around an axis perpendicular to the plane spanned by the vectors pointing in flow direction and in direction of the velocity gradient of the simple shear flow. Due to the lack of a contact model, the computations in [Luo and Pozrikidis, 2007] have been aborted before actual contact could occur. As we consider a lubrication model which accounts for slip effects, particle contact and attractive interparticle forces resulting from London–van der Waals attraction, we are able to investigate the effects described in [Luo and Pozrikidis, 2007]. In this specific subsection we take a close look at the effect of lubrication on particle collision with touching surfaces, not coagulation. Therefore we neglect forces arising from DLVO theory for now.

We simulated the behaviour of two intercepting particles in a simple shear flow with flow direction along the x-axis and gradient in y-direction. If not stated otherwise, the dimensionless shear rate is chosen as $\dot{\gamma} = 1$. The initial coordinates of the particles have been chosen as in Pozrikidis [2007] such that deviation from their initial paths due to particle interaction can be considered negligible (cf. figure 5.3). The considered radius ratios of the two spherical particles were $\lambda = a_2/a_1 = 0.5$ and $\lambda = 1$, where the particle with radius a_1 was placed above the plane of zero background flow

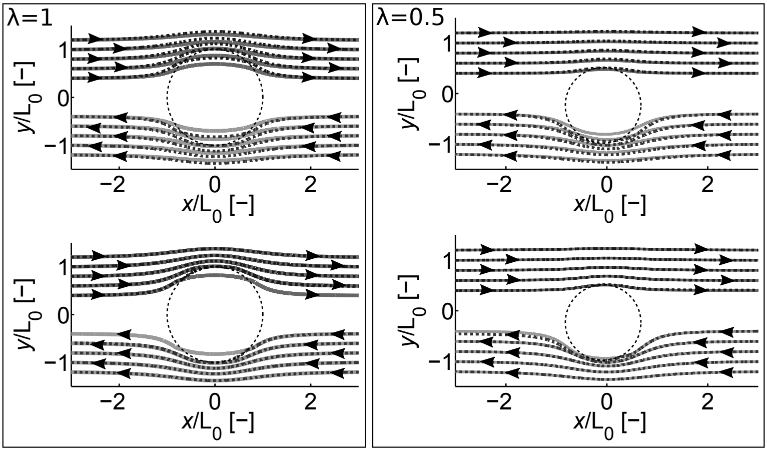

Figure 5.3: Trajectories of two intercepting particles in a simple shear flow. Top: Comparison of trajectories without lubrication scheme (solid lines) and with the proposed lubrication force model (dashed lines). Bottom: Comparison of the trajectories with lubrication force (solid lines) and with lubrication plus Hertz contact model (dashed lines). Left: Trajectories of two particles of equal radius. Right: Trajectories of two particles of unequal radius with radius ratio $\lambda = 0.5$. The particle with radius 1 started above $y = 0$, the particle with radius 0.5 below $y = 0$. Initial positions for $\lambda = 1$ are $x_{\text{init}} = \pm 3$, $y_{\text{init}} = \pm 0.4, 0.6, 0.8, 1.0, 1.2$, $z_{\text{init}} = 0$, for $\lambda = 0.5$ $x_{\text{init}} = \pm 6$ with the same y_{init} and z_{init}. Dashed circles indicate the region of particle contact, arrows show the direction of particle motion.

velocity and the particle with radius a_2 below this plane such that they move towards each other. Note that our simulations are in 3D, but as the flow is the described simple shear flow, the particle trajectories remain in the x-y-plane if particles start in this plane and if no Brownian motion is present.
In the two plots in figure 5.3, top, a comparison of the particle trajectories without lubrication scheme (solid lines) and with the proposed lubrication force model (dashed lines) is depicted. The lubrication model improves the SD solution without lubrication correction such that the particles only overlap for the innermost trajectories for both radius ratios, $\lambda = 1$ and $\lambda = 0.5$. In this case overlap is expected, as the lubrication force model proposed in subsection 5.2.2 does not totally prevent particle collision and in the simulations yielding figures 5.3, top, no contact model has been applied. Note that the lubrication force without additional contact model is an odd function and for all initial positions both considered particles return to their initial trajectories after interception. As predicted by Luo and Pozrikidis [2007], with the proposed lubrication model we observe particle contact for close enough particle trajectories and no contact if the particles are on trajectories, which result in interception but which are far enough apart such that hydrodynamic forces suffice to prevent actual contact. The bottom row of plots in figure 5.3 shows the trajectories obtained from simulations with additional contact model (dashed lines). No overlap can be observed, yet due to the purely repulsive Hertz contact force, particles are pushed from their original paths onto paths further away from the plane of zero background flow velocity. As the modeled lubrication force has no memory but always acts based on the current relative positions and velocities, the particles stay on their respective paths after contact. This modeling effect is more pronounced for particles of equal radius (cf. figure 5.3, bottom left) and almost does not affect the particles of unequal radius with radius ratio $\lambda = 0.5$ (cf. figure 5.3, bottom right). For both radius ratios, $\lambda = 1$ and $\lambda = 0.5$, we observe particle collision only for the innermost paths (initial positions $x_{\text{init}} = \pm 3$, $y_{\text{init}} = \pm 0.4, z_{\text{init}} = 0$ for $\lambda = 1$ and $x_{\text{init}} = \pm 6$, $y_{\text{init}} = \pm 0.4, z_{\text{init}} = 0$ for $\lambda = 0.5$). This shows that, if fundamental processes such as collision or coagulation of two particles are to be examined, hydrodynamic effects such as lubrication need to be taken into account to yield reliable results. A fact which was also noted in [Ernst et al., 2013], where the collision of two particles in a simple shear flow at a higher Reynolds number has been investigated.

5.3.2 Coagulation of two particles in a simple shear flow

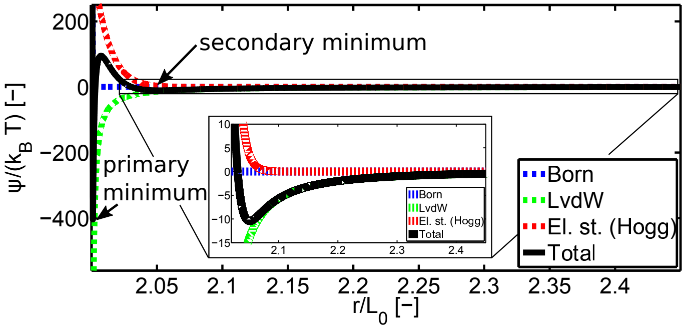

Figure 5.4: Dimensionless DLVO potentials with resulting total DLVO potential plotted over dimensionless interparticle distance r. Shown are the potentials for two particles of dimensionless radius 1. The inset figure displays a zoomed in view of the forces for $r \in [2.02, 2.45]$.

If we additionally consider DLVO forces with potentials as in figure 5.4, we observe the formation of permanent doublets. The depicted potential is the DLVO potential of Al_2O_3 particles in water, with Hamaker constant $A_{\mathrm{h}} = 4.76\mathrm{e}\text{-}20\mathrm{J}$, relative permittivity $\epsilon_r = 81$, temperature $T = 298\mathrm{K}$, an ionic strength of $\iota = 7\mathrm{mmol/l}$, and a separation $\sigma = 7\mathring{A}$, at which the Born potential can be assumed to vanish [Bülow et al., 2014]. The absolute value of the total double layer potential $|\psi_\zeta| = 37\mathrm{mV}$ has been measured for a concentration of 1mg/l at pH = 6 using a Malvern Instruments Zetasizer Nano ZS. The Debye-Hückel reciprocal length κ, which is a parameter of the used model for electrostatic repulsion, results from $\kappa = \sqrt{2N_{\mathrm{A}}e^2\iota/(\epsilon_r\epsilon_0 k_{\mathrm{B}}T)}$. Here, N_{A} denotes the Avogadro constant, e the elementary charge and ϵ_0 the vacuum permittivity.

The investigated setup is depicted in figure 5.5. Initial positions are chosen similar to that in subsection 5.3.1 with $x_{\mathrm{init}} = \pm 3$, for equal particles and $x_{\mathrm{init}} = \pm 6$ for unequal particles. W.l.o.g., for investigations on unequal particles we chose $y_{\mathrm{init}} \geq 0$ for the large particle, whereas for the small particle the respective y_{init} has a negative sign. In order to investigate the influence of the shear rate $\dot{\gamma}$ and the radius ratio λ on the capture efficiency α_o, we have performed simulations with dimensionless shear rates $\dot{\gamma} = 1, 2, 4, 8, 16, 32, 64, 100$ for the radius ratios $\lambda = 0.1, 0.5, 1$. Starting

from initial positions as in subsection 5.3.1, with the vector $\boldsymbol{r}_{\text{init}}$ lying totally in the x/y-plane, we redid the numerical experiments with differing s_{init} until we found the critical capture distance s_c, below which capture occurred. Subsequently we repeated the procedure with increasing angle $\theta = \arccos(y_{\text{init}}/s_{\text{init}})$, which is the angle between the y-axis and the projection of $\boldsymbol{r}_{\text{init}}$ into the y/z-plane, to cover the quarter cross-section highlighted in figure 5.5. Because of rotational symmetry this procedure suffices to cover all configurations which are necessary for a computation of the capture efficiency α_o. In case of equally sized particles, we have additionally tested the shear rates $\dot{\gamma} = 7, 48$.
Numerical integration during post-processing was performed using the trapezoidal rule with sufficiently small step size. If not stated otherwise, for curve fitting we have made use of the function 'fit' implemented in MATLAB [MATLAB, 2011].

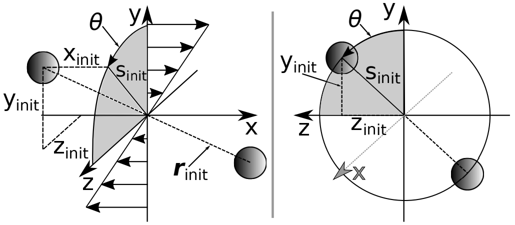

Figure 5.5: Investigated setup of two particles in a simple shear flow in x-direction with gradient in y-direction. Numerical experiments have been conducted with angles $\theta = 0, \pi/6, \pi/3, \pi/2.5, \pi/2.25$ and with values of $s_{\text{init}} = \sqrt{y_{\text{init}}^2 + z_{\text{init}}^2} \in [0.4, 2.0]$.

General observations for $\theta = 0$ and shear rate $\dot{\gamma} = 1$

Initial positions of particles are chosen as described in subsection 5.3.2, with $\theta = 0$, i.e. the initial z-coordinate of the particle positions is $z_{\text{init}} = 0$. In this case the initial distance from the x-axis is $s_{\text{init}} = y_{\text{init}}$, where the initial y-coordinate y_{init} is varied to study the particle behaviour due to differing particle separations. Figure 5.6 shows typical trajectories of two equally sized particles in the considered shear flow, when DLVO forces

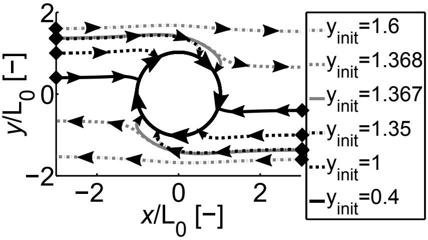

Figure 5.6: Trajectories of two intercepting particles of equal radius in the considered simple shear flow with models for lubrication, particle contact and DLVO forces. Arrows show the direction of particle motion and diamond markers their initial positions.

are taken into account. The particles follow the flow, but with decreasing interparticle distance the strength of the DLVO forces increases. If their initial separation is small enough, the particles leave their original path and approach each other more and more in a straight line through their centers until their center-to center separation is $r = 2.051$, which is the location of the secondary minimum of the DLVO potential for particles of equal size (cf. figure 5.4). This motion is superimposed by advection by the fluid, which from then on lets the particles rotate around the origin with constant surface-to-surface distance. As one can see in figure 5.6, the point of attachment to the closed-circle trajectory moves further downstream with increasing initial distance to the x-axis, y_{init}.
For unequal particles with radius ratio $\lambda = 0.5$ and initial y-positions of $y_{\text{init}} = \pm 0.4$, with the large particle starting at $(-6, 0.4, 0)^{\text{T}}$ and the small particle starting at $(6, -0.4, 0)^{\text{T}}$, the particle trajectories and snapshots of the configuration at different times are depicted in figure 5.7. The stage of approach is similar to the monodisperse case. If the initial distance is small enough, the particles approach each other until their center-to-center separation is $r = 1.552$, which corresponds to the location of the secondary minimum of the considered DLVO potential for two particles with radius ratio $\lambda = 0.5$. After the particles have reached the potential minimum, the small particle orbits the large particle, while this particle continues its path slightly influenced by the orbiting small particle. This motion happens for all investigated initial positions with y_{init} between $y_{\text{init}} = \pm 0.4$ and $y_{\text{init}} = \pm 1.1191$. The circling motion starts later and

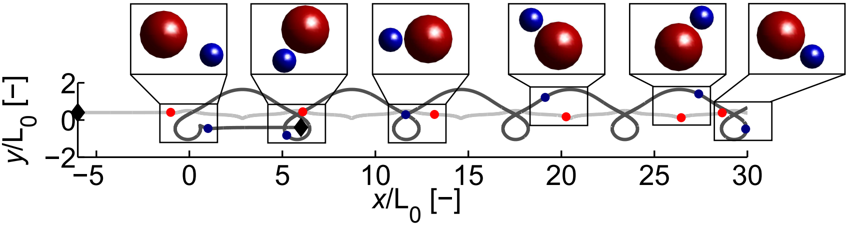

Figure 5.7: Trajectories of two intercepting particles of unequal radius with radius ratio $\lambda = 0.5$ in the considered simple shear flow with models for lubrication, particle contact and DLVO forces. The particle initial positions are indicated by diamond markers ($x_{\text{init}} = \pm 6$, $y_{\text{init}} = \pm 0.4, z_{\text{init}} = 0$). Snapshots visualize the relative positions of the particles on their trajectories.

the circle duration increases with increasing initial distance from the x-axis. For an initial separation slightly larger than this ($y_{\text{init}} = \pm 1.1192$), a strong influence of DLVO forces can be observed, but the forces are too weak to cause doublet formation (cf. figure 5.8). A further increase in initial distance in y-direction decreases the influence of DLVO forces, such that for $y_{\text{init}} = 1.4$ only 3.4% relative deviation from the original path is observed. Figure 5.8 shows selected trajectories of the small particle in the region of interception. Shown are trajectories for cases where the two particles pass each other before a possible coagulation. Differing from the case of equal particles, attractive interactions can alter the trajectory of

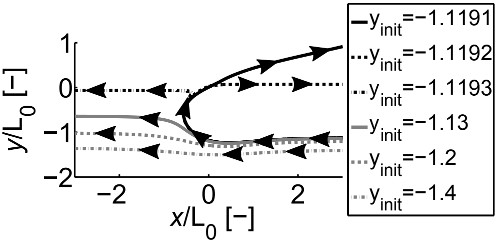

Figure 5.8: Close-up view of a part of the trajectories of a particle of dimensionless radius $a_2 = 0.5$ during interception with a particle of dimensionless radius $a_1 = 1$ in the considered simple shear flow with models for lubrication, particle contact and DLVO forces.

the smaller particle such that it is lifted from its trajectory below the x-axis to a trajectory above the x-axis, but without subsequent coagulation (cf. the trajectory for $y_{\text{init}} = -1.1192$). In that case, the small particle follows the large particle with increasing interparticle distance. If the initial distance is increased slightly ($y_{\text{init}} = -1.1193$), the trajectory of the small particle is altered strongly, but the particle persists moving downstream.

Influence of the angle θ and the shear rate $\dot{\gamma}$ on trajectories

We now describe observations concerning the influence of θ and the shear rate $\dot{\gamma}$ on the particle trajectories. For this we utilize results from simulations with dimensionless shear rates $\dot{\gamma} = 1, 2, 4, 8, 16, 32, 64, 100$ for the radius ratios $\lambda = 0.1, 0.5, 1$. Initial positions are as illustrated in figure 5.5, the procedure is described in subsection 5.3.2.

Especially for a low λ and a high $\dot{\gamma}$ we observed that the effect of lubrication plays an important role. Figure 5.9, left, shows the trajectories for $\lambda = 0.1$, $\dot{\gamma} = 32$ and initial y-positions with $y_{\text{init}} = \pm 0.5315$, and $\theta = 0$. As one can see, at the first encounter lubrication pushes the small particle off its original path. During interception it is attracted by the large particle, but the LvdW force is not strong enough to lead to direct coagulation. Crossing the plane of zero background flow velocity, the small particle follows the large particle for a while, closing the relatively large gap between their surfaces. This can be seen in figure 5.9, right, which shows a plot of the dimensionless surface-to-surface distance r_0 over the x-coordinate of the large particle, x_1. The inset figure shows a close-up view of r_0 for the time after coagulation. As stated in [Van de Ven and Mason, 1977] for equal particles, the surface separation r_0 oscillates with mean at the energy minimum during the orbiting motion.

The trajectories differ from the described cases when the motion does not lie completely in the plane spanned by the direction of the background flow and the direction of the shear gradient (in our case the x/y-plane). Figure 5.10 shows exemplary trajectories for the radius ratios $\lambda = 0.1, 1$, the shear rates $\dot{\gamma} = 8, 100$, different angles θ and initial separations s_{init} to the x-axis, at which subsequent capture occurs. In case of a small radius ratio, the large particle moves due to the background flow without strong

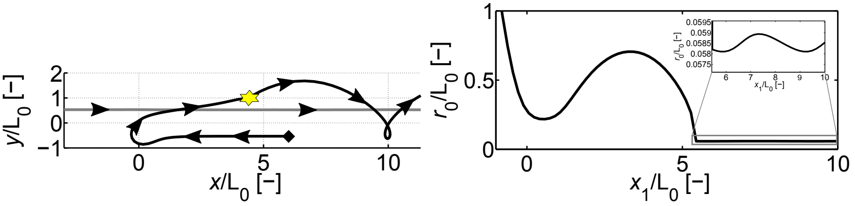

Figure 5.9: Left: Trajectories of two intercepting particles of radius ratio $\lambda = 0.1$ in the considered simple shear flow with models for lubrication, particle contact and DLVO forces. The starting position of the particle with radius $a_2 = 0.1$ is indicated by a diamond marker, the point of collision by a star marker. After collision the trajectories are periodic, as shown in figure 5.7. Right: Surface-to-surface distance r_0 over x-position x_1 of the large particle.

influence by the small particle. After interception, the small particle orbits the large particle in a trajectory parallel to the plane spanned by the direction of the background flow and the direction of the shear gradient. For equal particles, the orbiting motion after interception differs strongly. The particles do not only spin around the z-axis, as they do for $\theta = 0$. With increasing θ and growing $\dot{\gamma}$, the orbiting motion includes more and more a spin around the x- and y-axis, such that the trajectories cover the exclusion sphere more and more. As can be seen in figure 5.10 f), the exclusion sphere is completely covered for $\theta = \pi/2$ and respective values of $\dot{\gamma}$ and s_{init}. For $\lambda = 0.5$ we observed a behaviour similar to that when $\lambda = 0.1$, but with stronger influence of the small particle on the trajectory of the large particle (cf. figure 5.7). This results in an orbiting motion of the small particle around the large particle, which includes contributions of spins around the x- and y-axis, although not as pronounced as for $\lambda = 1$.

For the radius ratio $\lambda = 0.1$ and an angle $\theta = \pi/2$, we observed a kind of spiraling motion of the small particle towards the large particle (cf. figure 5.10 c)). This motion is best described by a superposition of the modes 1 and 2 given in [Feke and Schowalter, 1983]. The small particle swings back and forth in a plane parallel to the x/y-plane while it decreases the distance to the large particle. During approach, the curvature of the orbit increases until the small particle reaches the mutual secondary energy minimum. The higher the value of λ, the more the small

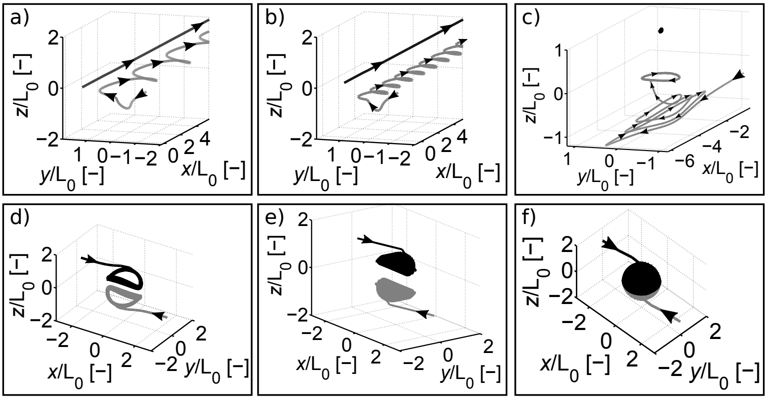

Figure 5.10: Trajectories of two particles in a simple shear flow for different radius ratios λ, shear rates $\dot{\gamma}$ and initial separations s_{init} to the x-axis, at which subsequent capture occurs. Top row: $\lambda = 0.1$, $\dot{\gamma} = 8$. Bottom row: $\lambda = 1$, $\dot{\gamma} = 100$. (a): $s_{\text{init}} = 0.66$, $\theta = \pi/3$, (b): $s_{\text{init}} = 0.7$, $\theta = \pi/2.25$, (c): $s_{\text{init}} = 1.0$, $\theta = \pi/2$, (d): $s_{\text{init}} = 0.83$, $\theta = \pi/3$, (e): $s_{\text{init}} = 0.88$, $\theta = \pi/2.5$, (f): $s_{\text{init}} = 1.0$, $\theta = \pi/2$.

particle influences the large particle. For equal particles, the motion is symmetric. It might be surprising that for $\theta = \pi/2$ the two particles approach each other even though they are too far apart for the LvdW force to move them. Furthermore, both particles have a zero y-component of their initial center coordinates, which would for point particles result in a zero drag force. But as stated in [Curtis and Hocking, 1970], the shear induces a rotation of the particles, which lets them move into the direction of the shear gradient. A single particle would just rotate without translation, but due to hydrodynamic interactions both particles experience a translational motion which lets them approach each other. This motion is initially very slow, so it is arguable whether the resulting capture efficiency reflects real conditions. In reality, the flow field usually changes too quickly for the particles to coagulate at such constellations. Other works define a maximum separation after interception, usually 10 dimensionless length scales (particle diameters), after which they declare a collision or capture as failed. E.g. in [Adler, 1981] or in [Vanni and Baldi, 2002] the authors state that they aborted simulations when particles are more than 10 particle radii apart after interception. Van de Ven and Mason [1977] say that theoretically particles can separate to infinity and return, but the error made by neglecting trajectories which exhibit a return after 10 dimensionless length units would be negligible for the computation of the capture cross-section. We address this topic in the following while we derive an expression for the orthokinetic capture efficiency for secondary (slow) coagulation of polydisperse particle systems on the example of Al_2O_3 particles in water. In similar works it is usually emphasized that the capture cross-section is not circular, as assumed by Smoluchowski [1917], therefore we include the assumption of a spherical collision/capture cross-section in our investigations. The outcome might be surprising and should be good news for everyone who is seeking for a good approximation to the *real* capture efficiency. Apart from the already given reasons we specifically concentrate on orthokinetic secondary coagulation. Investigation of a broader field, including perikinetic coagulation, would lead beyond the scope of this work. In case the reader might be interested in work which deals with the transition or interplay of orthokinetic and perikinetic coagulation, we refer to [Feke and Schowalter, 1983] and related work.

5.3.3 The capture efficiency of two unequal particles in a simple shear flow

Figure 5.11 shows obtained capture cross-sections for the investigated radius ratios $\lambda = 0.1, 0.5, 1$ and shear rates $\dot{\gamma} = 1, 2, 4, 8, 16, 32, 64, 100$. The top row shows the data points with a smoothing spline fit using smoothing parameters between 0.99 and 1, i.e. the fits are close to cubic spline fits. Values for the mean coefficient of determination $\overline{R}^2$ for the investigated λ are given in the captions to figure 5.11. The depicted capture cross-sections σ_c differ from circular collision cross-sections especially for increasing values of θ, i.e. for $y_{\text{init}} \to 0$. Furthermore, they correspond well to those presented in [Adler, 1981].

The more the particles lie initially in the $y = 0$ plane, which corresponds to $\theta = \pi/2$ in figure 5.5, the stronger is the deviation of the numerically obtained data yielding the capture cross-section from an assumption of

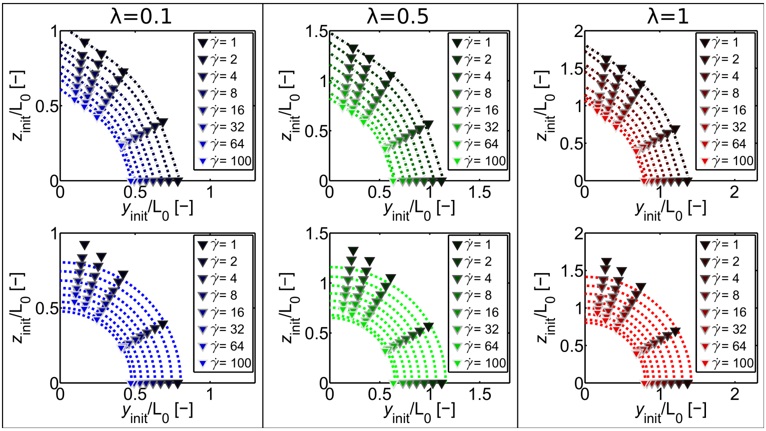

Figure 5.11: Capture cross-sections σ_c for the radius ratios $\lambda = 0.1, 0.5, 1$ (from left to right) and all investigated shear rates $\dot{\gamma}$. Top: Smoothing spline fit with $\overline{R}^2(\lambda) = 0.991, 0.987, 0.99$. Bottom: Circle fit only considering data points with $\theta = 0, \pi/6, \pi/3$.

a circular cross-section shown in the bottom row of plots in figure 5.11. To obtain the depicted circular capture cross-sections we fitted the numerical data using only values with $\theta = 0, \pi/6, \pi/3$. The circle fit can be performed using the standard Levenberg-Marquardt scheme of which numerous open-source implementations can be found (e.g. [cir, 2015]).
Apart from the already described observations, figure 5.11 shows that there is a dependency of the area of the capture cross section, usually also denoted by σ_c, on the radius ratio λ and a strong dependency on the shear rate $\dot{\gamma}$. The higher the shear rate, the smaller is the capture cross-section and thus the probability for two particles to coagulate under the given flow conditions.
The effect of $\dot{\gamma}$ can be seen well in figure 5.12, where the critical initial separation for capture, s_c, is plotted for the investigated λ over the shear rate. The value of s_c is decreasing with decreasing radius ratio and angle θ. For the smallest values of λ as well as for particles moving in the plane spanned by the direction of flow and the direction of the velocity gradient, the probability to coagulate is the lowest. This is a direct consequence of the facts that for decreasing θ the hydrodynamic force exerted by the

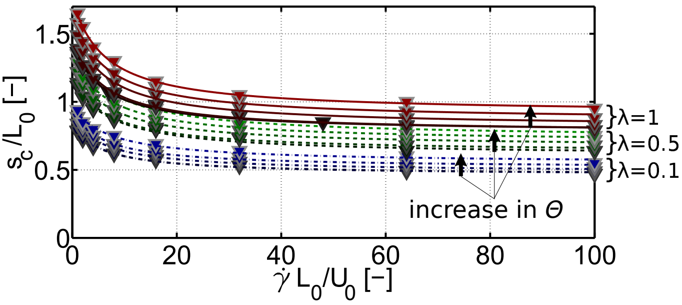

Figure 5.12: Values of the critical initial separations s_c from the x-axis, at which capture occurs, for all investigated angles θ as well as radius ratios λ over dimensionless shear rate $\dot{\gamma}$. The curves are hyperbolic fits with $\overline{R}^2(\lambda) = 0.988, 0.99, 0.993$.

background flow is increasing while for a decreasing λ the attractive LvdW force is decreasing (cf. [Hamaker, 1937] for the used form of the LvdW potential). Figure 5.12 further shows that the influence of the shear rate on s_c is decreasing hyperbolically, according to $(c_1\dot{\gamma} + c_2)/(\dot{\gamma} + c_3)$, with $c_1, c_2, c_3 \in \mathbb{R}$ depending on λ and θ. For dimensionless shear rates larger than about $\dot{\gamma} \approx 60$, the influence of an increase in the shear rate is small compared to the strong decrease for small $\dot{\gamma}$. All values of s_c have been determined using the method described earlier in this work. In order to keep the resulting errors in the capture cross-section σ_c and in the capture efficiency α_o small, numerical simulations have been stopped when the precision of s_c after a series of simulations was 1% or better. Former works [Van de Ven and Mason, 1976, Adler, 1981] state that they observed a later increase in s_c with increasing $\dot{\gamma}$. At high shear rates the hydrodynamic drag would be too strong to allow for coagulation in the secondary minimum, which causes the initial decrease in coagulation efficiency. After reaching a certain value, the potential barrier could be overcome which allows particles to coagulate in the primary minimum. Although this sounds plausible, we did not find any proof for such a behaviour of the system at hand. An investigation of the interparticle distances over time showed that even for the highest considered values of $\dot{\gamma}$ and λ we either found the particles to coagulate in the secondary minimum or to not reach it at all. For separations larger than the critical value of s_c, hydrodynamic forces (background flow as well as lubrication) suffice to keep particles from reaching an energy minimum and thus prevent coagulation. For smaller values of s_c, the LvdW force is too strong to be overcome. Note that s_c is the critical initial separation of the particle centers from the x-axis, for which capture occurs. Subtraction of the particle radius leads in case of equal particles and higher values of $\dot{\gamma}$ to negative values. This shows the strong effect of the hydrodynamic forces, which can prevent coagulation even if straight trajectories would lead to collision.

The area of the capture cross-section σ_c can be computed from

$$\sigma_c = 4 \int_0^{\max(y_c)} Z(y)\,\mathrm{d}y, \tag{5.10}$$

where $\max(y_c)$ is the maximum extent of the quarter of the capture cross-section with positive y- and z-values, and Z is the function describing the boundary of the capture cross-section in z-direction (cf. figure 5.11). Analogous to [Van de Ven and Mason, 1977] we further define the orthokinetic capture efficiency for polydisperse systems as

$$\alpha_o = 3 \frac{L_0^3}{(a_1 + a_2)^3} \int_0^{\max(\hat{y}_c)} \hat{y}\hat{Z}(\hat{y})\,\mathrm{d}\hat{y},$$

where variables denoted with $\hat{\cdot}$ have been made dimensionless by L_0. a_1 as well as a_2 are the radii of the two considered particle size fractions. In our numerical experiments considering two particles in a shear flow we have $L_0 = a_1, a_1 > a_2$, thus the orthokinetic capture efficiency can be written as

$$\alpha_o = \frac{3}{(1+\lambda)^3} \int_0^{\max(\hat{y}_c)} \hat{y}\hat{Z}(\hat{y})\,\mathrm{d}\hat{y}. \tag{5.11}$$

For $\lambda = 1$ this is in accordance with the derivation for equal particles in [Van de Ven and Mason, 1977].

The dependency of the capture cross-section on the shear rate is shown in figure 5.13 for the two different approximations to Z, which are depicted in figure 5.11. As one can see, for all radius ratios and shear rates, the

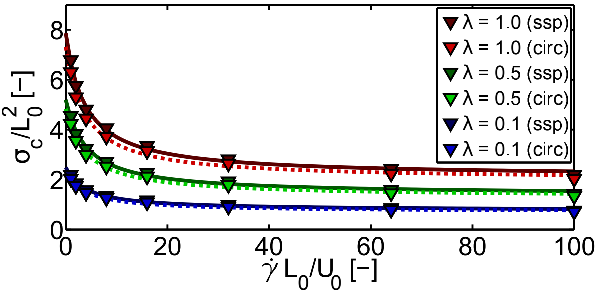

Figure 5.13: Areas of the capture cross-sections σ_c for the radius ratios $\lambda = 0.1, 0.5, 1$ over dimensionless shear rate $\dot{\gamma}$ once using a smoothing spline fit (ssp) to obtain the integral value and once using a circle fit (circ). The curves are hyperbolic fits to the respective data with $R^2(\lambda) = 0.99, 0.992, 0.995$ for the spline fit (solid line) and $R^2(\lambda) = 0.992, 0.992, 0.993$ for the circle fit (dashed line).

integration in equation (5.10) yields an almost matching area of the capture cross-section, no matter which assumption has been made for the shape of the capture cross-section. Furthermore, the behaviour of σ_c with a change in $\dot{\gamma}$ can be approximated well by a hyperbola for both shape-approximations. The former observation gives rise to two statements: First, the assumption of a spherical capture cross-section is reasonable. As already Van de Ven and Mason [1977] stated, despite the highly simplifying assumptions made by Smoluchowski [1917], the counteraction of hydrodynamic resistance and LvdW attraction yield a remarkably good prediction by Smoluchowski's theory. Second, when assuming a circular cross-section, one does not need to perform a bunch of experiments in order to obtain the arguable right shape of the capture cross-section. This renders time-consuming experiments for initial configurations with θ close to $\pi/2$ redundant. For these trajectories it is questionable whether the two particles touch under real conditions. Thus far all works have introduced an artificial maximum separation after interception of 10 dimensionless length units (particle radii), for which they considered an encounter as not leading to capture. During numerical experiments leading to the presented data we observed for $\lambda = 0.1$, $\dot{\gamma} = 8$, $s_{\text{init}} = 1.0$ and $\theta = \pi/2$ a slow spiraling motion of the small particle out of a plane

parallel to the plane of zero velocity of the background flow towards the almost stationary large particle. The initial major axis of the elliptical orbit increases with increasing s_{init} and was about 12 dimensionless length units for $s_{\text{init}} = 2.0$. In this case, the interparticle distance was temporarily larger than 10 dimensionless length units, yet the configuration led to capture. For a moderate value of $\dot{\gamma} = 8$, a radius ratio $\lambda = 0.1$, $s_c = 0.75$ and $\theta = \pi/2.25$, we observed an interesting behaviour of the small particle. At the first encounter hydrodynamic interactions prevent coagulation of the two particles. After interception, the small particle crossed the plane of zero background flow velocity and followed the large particle over a distance larger than 20 dimensionless length units, keeping a distance of about 5 dimensionless length units over a big part of the chase until it finally reached the common potential minimum and started to orbit the large particle (similar behaviour can be seen in figure 5.9 for a shorter initial distance s_{init}, a higher shear rate and $\theta = 0$). This motion is realistic, as explained in [Curtis and Hocking, 1970], but does it lead to capture in a technical system? We included it for the smoothing spline approximation to σ_c, but an inclusion is arguable. The good agreement of the values of σ_c for the two shape-approximations over a wide range of shear rates for all investigated λ suggests that this argument and a definition of a likewise questionable maximum separation are no longer required.
Figure 5.14 a) shows the behaviour of the capture efficiency α_o for the two

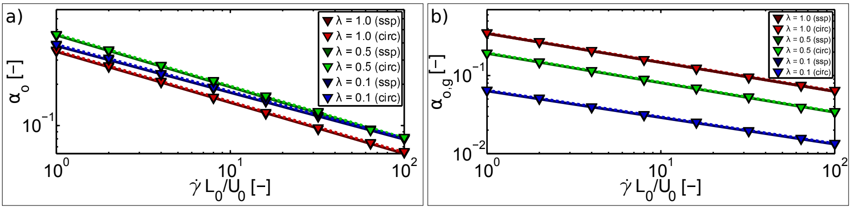

Figure 5.14: Values of the capture efficiency α_o (a) and the global capture efficiency $\alpha_{o,g}$ (b) for the radius ratios $\lambda = 0.1, 0.5, 1$ over dimensionless shear rate $\dot{\gamma}$ and curves from power-law fits to the approximated capture cross-sections from a smoothing spline fit (ssp) and a circle fit (circ). The fits to the respective data have a coefficient of determination above $R^2(\lambda) = 0.99$ for both the spline fit (solid line) and the circle fit (dashed line). Note that the data is plotted using log-log scaling.

discussed shape approximations. As one can see, the weighted integration in equation (5.11) giving α_o reduces the resulting deviation between α_o from the smoothing spline approximation and that from the circle approximation to the capture cross-section such that the corresponding data points as well as their fits (solid line for the smoothing spline approximation and dashed line for the circle approximation) are almost identical. So far we fitted data versus shear rate by means of a hyperbolic fit, which gives good agreement also for α_o. Yet, figure 5.14, in which the capture efficiency α_o is shown in a log-log plot, suggests a power-law dependency on $\dot{\gamma}$. This behaviour has already been observed earlier. Van de Ven and Mason [1977] propose for the case of negligible electrostatic repulsion, i.e. primary or fast coagulation, that $\alpha_o \approx f(\lambda_{\text{London}})\dot{\gamma}^{-0.18}$, with f being a function of the London wavelength λ_{London}, which is typically about 100nm [Zeichner and Schowalter, 1977]. Zeichner and Schowalter [1977] obtain $\alpha_o \propto \dot{\gamma}^{-0.33}$, provided the flow is sufficiently strong. Vanni and Baldi [2002] fit their data by $\alpha_o = 0.625\dot{\gamma}^{-0.186}a_1^{-0.7}$, with the dimensions of $\dot{\gamma}$ in $1/s$ and a_1 in μm. Unfortunately, their definition of the capture efficiency is questionable (see the statement in [Van de Ven and Mason, 1977] on the definition in [Curtis and Hocking, 1970]), so this expression should be handled with care. Adler [1981] gave an analytical expression for the global orthokinetic capture effciency $\alpha_{o,g}$ of particles subjected to an electric field, and assumed the global capture efficiency in case of DLVO interactions to behave like $\alpha_{o,g} = kH_{\text{A}}^n$, with $k, n \in \mathbb{R}$ and the dimensionless Hamaker constant $H_{\text{A}} = A_{\text{h}}/(144\pi\mu_f a_1^3\dot{\gamma})$. Here, A_{h} denotes the dimensioned Hamaker constant. The relation between $\alpha_{o,g}$ and α_o is given by

$$\alpha_{o,g} = \frac{(a_i + a_j)^3}{8L_0^3}\alpha_o. \tag{5.12}$$

For $L_0 = a_i = a_j$, the global capture efficiency $\alpha_{o,g}$ equals the capture efficiency α_o.

Care has to be taken when analyzing which type of coagulation is more probable for a certain $\dot{\gamma}$ — homo- or heterocoagulation. For studies on the coagulation probability, the capture efficiency alone can only be used in case of monodisperse systems. In case of polydisperse particle systems, one needs to compute the capture frequency β_{ij}, which can be extracted from the definition of the collision rate,

$$J_{ij}^{o} = \beta_{ij} N_i N_j = \alpha_o \frac{4}{3} \dot{\gamma} (a_i + a_j)^3 N_i N_j, \tag{5.13}$$

for the number concentrations N_i and N_j of the size fractions i, j among the N_f size fractions in the system. The ratio of capture frequencies, β_{ij}/β_{kl}, with the indices $i, j, k, l \in [1, N_f]$, then determines which λ yields the highest coagulation probability. As one can see in figure 5.14 (a), the values of α_o are often close for different radius ratios, while the capture efficiency α_o for $\lambda = 1$ is usually the lowest, but this does not reflect the coagulation probability. In order to give a qualitative statement on the coagulation probability for different λ, one needs to first scale the corresponding α_o by the other terms β_{ij} is dependent of. The same result can be obtained from a direct comparison of $\alpha_{o,g}$ for different λ, what is depicted in figure 5.14 (b). This makes the global collision efficiency $\alpha_{o,g}$ a better indicator for the coagulation probability in polydisperse systems. Figure 5.15 shows the behaviour of $\alpha_{o,g}$ over λ for varying $\dot{\gamma}$. The plots suggest a linear dependency of $\alpha_{o,g}$ on λ, but for low values of the shear rate, $\alpha_{o,g}$ is not linear in λ and a power-law relation between $\alpha_{o,g}$ and λ yields a better approximation. The so far made observations concerning

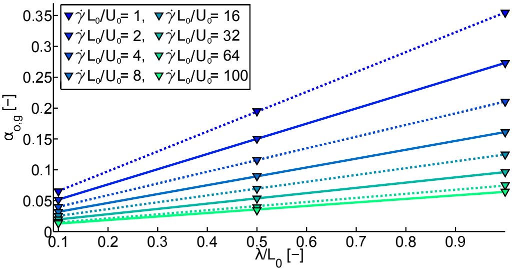

Figure 5.15: Global capture efficiency $\alpha_{o,g}$ over radius ratios $\lambda = 0.1, 0.5, 1$ for investigated dimensionless shear rates $\dot{\gamma}$.

$\alpha_{o,g}$ lead to the assumption

$$\alpha_{o,g}(\lambda, \dot{\gamma}) = c_1 \lambda^{k_1} \dot{\gamma}^{k_2}, \quad c_1, k_1, k_2 \in \mathbb{R}.$$

Using the Nelder–Mead simplex direct search [Lagarias et al., 1998], a minimization of the sum of squares of the residual with respect to the measured values yields the relation

$$\alpha_{o,g}(\lambda, \dot{\gamma}; L_0, U_0) = 0.33 \lambda^{0.69} \left(\frac{L_0}{U_0} \dot{\gamma} \right)^{-0.36} \tag{5.14}$$

for the global capture efficiency, where the shear rate $\dot{\gamma}$ is in dimensioned form and has been made dimensionless by the characteristic time scale L_0/U_0. Together with equation (5.12) and $\lambda = a_j/a_i$, $a_j \leq a_i$, we get the orthokinetic capture efficiency,

$$\alpha_o(a_i, a_j, \dot{\gamma}; L_0, U_0) = \frac{8L_0^3}{(a_i + a_j)^3} 0.33 \left(\frac{a_j}{a_i} \right)^{0.69} \left(\frac{L_0}{U_0} \dot{\gamma} \right)^{-0.36}, \tag{5.15}$$

which can be used as correction factor in equation (5.13) for the orthokinetic collision rate J_{ij}^o for heterocoagulation.

5.3.4 Coagulation and cluster growth in destabilizing particle systems

Our parallelized code, together with a cluster detection algorithm based on the R*-tree method [Beckmann et al., 1990], allows for an investigation of the destabilization behaviour of large particle systems within days [Bülow et al., 2014]. Therefore, to go beyond the consideration of two particles, we now consider the overall coagulation rate in a destabilizing system under the action of centrifugal forces, and we examine the applicability of a collision rate obtained from experiments considering only doublet formation.

In our simulations, the initial particle domain is determined by the polydisperse radius distribution, the number of simulated particles, and the solid volume fraction. The radius distribution was chosen as lognormal distribution with expected dimensionless radius 1 and a standard deviation of 0.065 dimensionless length scales. To rule out effects from the initial random placement of particles according to a uniform distribution, as well as from the chosen random radius distribution, we performed every experiment three times with fixed solid volume fraction ϕ and number of particles N. The deviation in the considered values due to the random placement and radius distribution was within a tolerance of 3% in all experiments. For the considered numbers of simulated particles and presented values, the effect of a change in N was negligible as well. For this, we performed every simulation using $N = 500, 1000, 2000$ and compared the temporal evolution of different characteristic values such as the relative number of clusters N_c/N over time. At the beginning of each simulation the initial number of clusters equals the number of primary particles, N.

Usually, it is assumed that doublet formation is the dominant case within the first few seconds of destabilization. Figure 5.16 shows the number of clusters over initial number of particles, N_c/N, for different initial solid volume fractions ϕ within this time. The dimensionless time over which N_c/N is plotted strongly depends on the particle size. For primary $\mathrm{Al_2O_3}$ particles of radius $a \approx 1\mu$m in water, one can assume a characteristic time L_0/U_0 of roughly 0.15s. This is the lower limit for the particle radius, if not already below the limit, for which the assumption of non-Brownian particles is justified. As one can see in figure 5.16, the interesting time range are indeed the first seconds of destabilization.

There are three main observations, which can be made in figure 5.16. First, the ratio N_c/N does not seem to depend on the simulated number of primary particles, N. The deviation between the curves for the same ϕ and differing N in the highlighted section is small for all investigated ϕ. The second observation is the expected fact that in more concentrated suspensions the overall coagulation rate is higher than in more dilute suspensions. This can be seen when comparing the slope of the curves corresponding to respective values of ϕ. Third, after a strong initial decrease, the relative number of clusters N_c/N approaches a limiting value, which is dependent on ϕ. Bearing in mind that large clusters usually settle faster than small clusters, the reason for this becomes evident: Classification due to

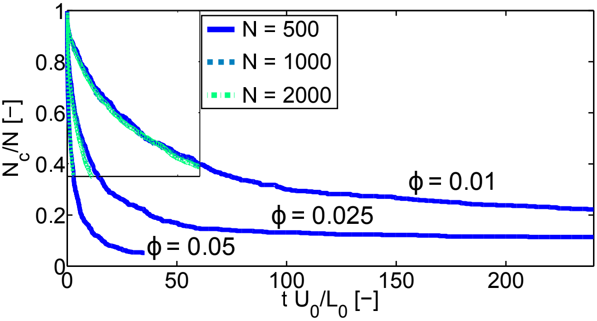

Figure 5.16: Dependency of the ratio of number of clusters N_c at time t to initial number of clusters N on the initial solid volume fraction ϕ for different values of N (highlighted section).

different settling speeds of clusters of different sizes and shapes limits coagulation with time.

If we take a look at the relative number of doublets $N_{\mathrm{c},2}/N$, triplets $N_{\mathrm{c},3}/N$ and quadruplets $N_{\mathrm{c},4}/N$ over time t for $\phi = 0.01$ and $\phi = 0.025$ (cf. figure 5.17 (a) and (b), respectively), we can see that doublet formation is indeed the dominant coagulation mechanism in the early stages of destabilization. Figure 5.17 shows that, while doublets are formed first and are the dominant type of coagulation in the beginning, larger clusters are formed close after, especially with increasing solid volume fraction ϕ. One can also see that for $\phi = 0.01$ doublets are the main fraction of clusters over a longer timespan. The relative number of doublets decreases

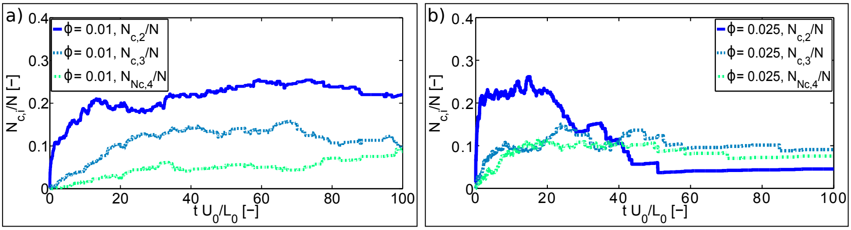

Figure 5.17: Relative number of doublets $N_{\mathrm{c},2}/N$ (solid line), triplets $N_{\mathrm{c},3}/N$ (dashed line) and quadruplets $N_{\mathrm{c},4}/N$ (dashed-dotted line) over time t for $\phi = 0.01$ (a) and $\phi = 0.025$ (b).

more rapidly for the higher solid volume fraction $\phi = 0.025$, where the initial formation rate of triplets and that of quadruplets is higher than for $\phi = 0.01$. For $\phi = 0.05$ we observed similar tendencies, with the formation rate of triplets and quadruplets being even closer to that of doublets, but the formation rate of doublets still being the highest in the early stages of destabilization.

The investigation of the temporal behaviour of the relative number of clusters of different numbers of primary particles leads to the conclusion that an investigation of doublet formation, may it be caused by the fluid flow as in the present work, or solely due to settling as e.g. in [Davis, 1984, Han and Lawler, 1991], yields an acceptable way to determine collision rates for early stages of destabilization. The prediction is better for dilute suspensions, as for these coagulation leading to larger clusters has a later onset. A consideration of two isolated particles and a derivation of a collision rate is a strongly simplifying assumption, best met by dilute suspensions. The more concentrated a suspension is, the more likely it is that at early stages clusters of more than two particles are formed. More importantly, the stronger is the influence of surrounding particles.

5.4 Conclusions

For the present work we investigated the orthokinetic coagulation dynamics of mono- and polydisperse particle systems. We presented a model for lubrication interactions including slip and showed that this model yields good results while keeping computational costs low. By means of direct numerical simulations of the behaviour of two particles with varying radius ratios we have derived an expression for the capture efficiency for orthokinetic secondary heterocoagulation (including the case of homocoagulation). We have shown that, while the capture efficiency is a good way to correct the collision rate obtained from Smoluchowski's theory, the global capture efficiency is a better indicator for the analysis of the more probable type of coagulation — homo- or heterocoagulation. Furthermore, due to the involved weighted integration during derivation of the capture efficiency, the assumption of a spherical capture cross-section leads to very similar results compared with the resolution of the capture cross-section by means of a large number of experiments. We have only shown this for the model system of Al_2O_3 particles in water and the considered shear rates. Yet due to nondimensionalization, the results are more general and hold at least for other systems tending to secondary coagulation. Furthermore, the part of the capture cross section which deviates the most from a spherical assumption is also for primary coagulation weighted the least during derivation of the capture efficiency. This supports the hypothesis that an assumption of a circular capture cross-section is reasonable for a broader range of systems. However, it is left to future experiments to further strengthen its validity. Nevertheless, one should always keep the practical case in mind. How much time is two particles given in a stirred tank or in a settler to coagulate under the exact same conditions as in the experiment? If particles lie in the plane of zero fluid velocity of an idealized simple shear flow and the conditions do not change, they coagulate with high probability. But this process takes a long time and it is questionable whether the two particles have this time before the flow conditions change.

In a second part of this work we discussed results from numerical experiments concerning the destabilization behaviour of polydisperse particle systems under the action of centrifugal forces. We showed that the solid volume fraction plays a role, both in the derivation of a global cluster growth rate as well as for the validity of coagulation rates based on experiments with two isolated particles. Investigations based on numerical experiments including more than two particles can yield a more accurate prediction of the coagulation rate in suspensions. If coagulation rates for the initial stages of destabilization are sought for, the total number of simulated particles does not necessarily need to be very high. In our experiments, simulations of the behaviour of 500 primary particles were enough to yield acceptable results. For the sake of brevity, we only showed some of the data one can obtain from a cluster detection applied to the data from numerical experiments. The implemented cluster detection gives the growth rates for all cluster size fractions over time. If a prediction of the overall cluster growth rate is not sufficient, this would yield excellent data for flow-sheet simulations, as well as for kernel-based methods such as numerical simulations involving the solution of population balance equations.

Simulations as the ones presented here can also give perikinetic coagulation rates, or such describing superimposed perikinetic and orthokinetic coagulation. One can either determine the growth rates as shown, or derive a size-dependent self-diffusion coefficient with which the coagulation rate can be determined.

Chapter 6

Summary and outlook

After giving an overview of destabilizing heterogeneous suspensions, we presented the different approaches and methods currently used for the numerical simulation of the behaviour of particles in fluids. Among these we chose the Stokesian Dynamics method for our research. We use this method because of its elegance when it comes to the computation of hydrodynamic interactions and because of its flexibility and extensibility with respect to additional models for interparticle interactions of non-hydrodynamic nature. Except for a simplified implementation for GPUs [Kopp and Höfling, 2012], there has not been any parallel implementation of the SD method. So far this limited the number of simulated particles to a few hundred, which restricted the applicability of the method with respect to the investigation of destabilizing suspensions [Bülow et al., 2015]. Therefore, our first step was to parallelize our base code and extend it for the feature of a consideration of DLVO interactions among particles [Bülow et al., 2014]. Because of the nature of the method, a parallelized SD method scales very well with the number of used CPUs, such that even one million hydrodynamically interacting particles can be simulated within an acceptable timespan.

To achieve a further reduction in numerical costs, we coupled the resulting parallel SD method with the Barnes–Hut method in order to obtain an even faster method [Bülow et al., 2016]. The Barnes–Hut method relies on particle grouping and is especially well-suited for parallel code execution. This is because on a distributed memory machine, what most supercomputers presently are, particle data is distributed among the memory of the used CPUs. In an extreme case, particles can be spatially grouped according to their distribution in the memory. This means that direct hydrodynamic interactions are considered only between the particles with data in the memory of the same CPU, and simplified hydrodynamic interactions are computed between groups with data in the memory of different CPUs. This does not only reduce computational but also communication costs. The combination of the two methods, which we called scalable SD method, yields very good results when the number of simulated particles is high, particles are clustered and the overall suspension is not very dense. In general, the better particles can be combined to groups which have large particle voids between them, the faster is the simulation. For a dense or very homogeneous suspension, the method falls back to the parallel implementation, with a small overhead due to the construction of the tree-structure coming from the Barnes–Hut part of the method. A drawback of the new method is the timestep reduction, which comes along with the dynamical grouping of particles [Bülow et al., 2016]. This reduction seems to be limited, with no effect for simulations of suspensions where coagulation due to DLVO interactions plays a role. In that case, the high gradients in the interparticle potential require such a small timestep that the grouping does not impose another reduction in timestep width. The method and the timestep problem are well described in chapter 4 and the corresponding work [Bülow et al., 2016]. Apart from the more detailed description in chapter 4 we did not go into further details on the theory behind the SD method, as it has been explicated sufficiently in other works such as [Brady and Bossis, 1988, Durlofsky et al., 1987, Ichiki, 2002] and [Kim and Karrila, 2013].

After the improvement of the used method with respect to simulations of large particle systems we investigated the settling behaviour of polydisperse particle clouds. The parallel framework enabled us to show that large polydisperse particle clouds settling at vanishing Reynolds number exhibit the same characteristic behaviour as their monodisperse counterpart [Bülow et al., 2015]. This has unintentionally already been shown by experiments, where slightly polydisperse particle systems were compared with simulations of monodisperse particle systems [Metzger et al., 2007, Myłyk et al., 2011]. In [Myłyk et al., 2011] the simulated particle clouds consisted of considerably less particles than were in the respective clouds in the experiments. By a comparison of settling velocities, we showed that both the assumption of a monodisperse system and the assumption of independence of the number of particles in the cloud lead to strong deviations [Bülow et al., 2015]. During our research we noticed that the deviation not only influences the settling velocity but also the evolution of the particle clouds. The more particles are in a cloud, the more stable it is, and the later it breaks up into possible daughter clouds. In our opinion this is the main factor leading to the 30% deviation between experiments and simulations, which has been reported in [Myłyk et al., 2011] for the destabilization length. Before our work, numerical experiments in this field have been carried out involving either monodisperse systems of up to a few thousand particles or polydisperse systems of only a few hundred particles. Clouds comprised of only a few hundred particles are per se not as stable as larger clouds [Metzger et al., 2007], which led in earlier works to the conclusion that polydisperse clouds would not undergo the same motion as monodisperse clouds and that the solid volume fraction would be the main influencing factor [Abade and Cunha, 2007]. In fact, the number of particles is the main influencing factor and if the particle clouds are large enough, polydisperse clouds undergo the same motion as monodisperse clouds [Bülow et al., 2015]. We also showed that in mixed bidisperse clouds the evolution is dominated by the fraction with the highest number of particles. Initially spherical clouds lose particles in an emanating tail and, depending on the intial number of particles, during settling they break up in a cascade into several daughter clouds. Recent research reports a preferential loss of small particles during settling [Faletra et al., 2015], which we did not observe as pronounced. This can most likely be attributed to the fact that the particle clouds investigated in [Faletra et al., 2015] consisted of only 300 particles, which are normally not enough to form a stable cloud [Bülow et al., 2015, Metzger et al., 2007]. During our investigations of particle clouds we also derived a formula, which can predict the settling velocity of a polydisperse particle cloud at vanishing Reynolds number. This formula can be useful when estimating the maximum settling velocity in a suspension (provided the size of formed hydrodynamic clusters is known), or when conducting numerical experiments on particle clouds. In that case, the predicted velocity can be utilized as characteristic velocity scale.

Hydrodynamic clustering is a result of particle motion and hydrodynamic interactions, which alone usually does not determine suspension stability. For example Brownian particles suspended in a quiescent fluid do not settle due to hydrodynamic clustering, but due to destabilization induced by perikinetic coagulation. The formed agglomerates can settle and cause hydrodynamic clustering, which facilitates stronger agglomeration and so on. In process units there is usually a flow with a shear gradient. Therefore, we have investigated the orthokinetic coagulation behaviour of two particles in a shear flow. This is a standard procedure for the derivation of the capture efficiency in suspensions (cf. section 1.3 or chapter 5). We found that the procedure of deriving the capture efficiency from two-particle experiments can be sped up, compared to the standard approach in which the capture cross-section is approximated by a vast number of numerical experiments. The effect of particle configurations with particles initially lying almost or totally in the plane of zero background flow velocity on the capture efficiency is small. This is a direct result of the mathematical definition of the capture efficiency, but also makes sense from a physical point of view: Particles lying in the plane of zero fluid velocity are not directly convected by the fluid, they first need to move out of this plane [Curtis and Hocking, 1970]. This leaves in the initial phase of approach only the London–van–der–Waals force as driving force for coagulation. This force is only strong for small interparticle distances. For larger distances, the particles will approach each other only very slowly. In a crystallizer or another process unit in which coagulation plays a role, the flow conditions usually change too quickly for the two particles to have the time to approach each other slowly while circling each other for a while until they coagulate (cf. chapter 5). This is equivalent to the earlier statement that such configurations do not influence the capture efficiency much.

Even though the classical two-particle approach can be sped up, the determination of the capture efficiency by these means can take a long time if several parameters have to be tested. Furthermore, the effect of the solid volume fraction and the influence of other particles are not considered by two-particle approaches. Moreover, for a superposition of different effects, such as shear, settling, Brownian motion, etc., the number of parameter combinations and arising experiments is exorbitantly high. Therefore we implemented a cluster detection algorithm, which gets the output of simulations of multi-particle systems as input and yields all necessary data of the formed clusters. Included are e.g. the cluster settling velocity, the radius of gyration and the fractal dimension of all clusters over time. Among the data is also the number of primary particles per cluster over time. This information can be used to determine statistical cluster growth rates when effects such as shear and settling are superimposed. Furthermore, this way the influence of other particles and that of the solid volume fraction are considered. Also, contrary to two-particle experiments, the evolution and interaction of triplets, quadruplets and larger clusters can be investigated. Solely considering two particles can give wrong approximations to the capture efficiency in case of dense suspensions. Based on our observations, doublet formation can be assumed to be the dominant type of coagulation for dilute suspensions and in the initial stages of coagulation. For a further discussion we refer to chapter 5.

A special type of destabilizing suspensions are biological suspensions. We have extended our simulation framework such that it includes solute transport and the change of particle size and number over time. The latter is based on kinetics, which are in wastewater treatment often described by the extended Monod equation [Klapper and Dockery, 2010, Xavier et al., 2005a,b]. As extra cellular polymeric substances (EPS) play a major role in biofilms, we included the model for the effect of unresolved polymers discussed in subsection 1.2.2. These extensions make the used method an Eulerian–Lagrangian method (cf. subsection 1.5.2), which introduces additional challenges for a parallel simulation. In this work we did not present any results obtained with this method, as it is not published yet. Nevertheless, preliminary results are promising, which is the reason why we mention it here. For further reading we refer to the planned publication [Bülow et al., 2016].

The presented framework can easily be extended to account for effects, which have not been considered in this work. For example Janus particles can be simulated by an inclusion of an interparticle distance- and orientation-dependent potential (e.g. that presented in [Vanakaras, 2006] for half hydrophobic and half hydrophilic core-shell Janus particles). The tracking of particle orientations is included in the part of the developed simulation framework involved in the simulation of elastic-plastic behaviour of aggregates. The corresponding method is similar to the method presented in [Seto et al., 2012], which we have implemented as well. From our experience, the method in [Seto et al., 2012] is not suitable for the simulation of purely elastic aggregates. Every tested choice of modeling parameters resulted in a strong deformation of the considered aggregates — even for single settling fibres, i.e simple aggregates experiencing only weak forces. After a short period of elasticity, the investigated agregates collapsed. The model proposed in [Kabanemi and Hétu, 2012] leads to very good results concerning the motion of flexible fibres in a shear flow and settling in a quiescent fluid, but has not yet been implemented for other types of aggregates. Compared to the model in [Seto et al., 2012], the model in [Kabanemi and Hétu, 2012] considers two neighbours of a primary particle in an aggregate during force and torque computation. The more neighbouring particles are considered for the computation of forces and torques in an aggregate, the more communication is required during a simulation running on a distributed memory machine. Therefore we suggest to use a three-particle method like that in [Kabanemi and Hétu, 2012] for the simulation of non-spherical particles and aggregates modeled by the multi-sphere approach [Ferellec and McDowell, 2010, Kruggel-Emden and Oschmann, 2014].

The Lewis acid-base (AB) model for hydrophobic and hydrophilic interactions among particles [Van Oss, 1989, 1993] seems to become a standard technique for a consideration of effects arising from these interactions in aqueous suspensions, especially in biological suspensions [Hermansson, 1999]. The latter are a promising field for future research. Fermentation-related production is a growing field and the presented framework can be used to simulate the growth behaviour of biofilms as well as their behaviour under stress — hydrodynamic as well as mechanical.

We did not present any results concerning non-spherical particles in this work. Figure 1.1 in chapter 1 shows implemented shapes of particles, agglomerates and aggregates. In case of agglomerates or flocs, the depicted shapes can be chosen for the initial placement of particles, whereas during a subsequent simulation DLVO interactions and particle contact models are used to compute interparticle interactions of non-hydrodynamic nature. In case of aggregated particles or non-spherical particles, the primary particles are connected by bonds which can be described by several implemented models for forces in normal direction [Hsieh et al., 2006, Shaqfeh et al., 2004], as well as by the discussed implemented methods for flexible aggregates [Seto et al., 2012, Kabanemi and Hétu, 2012]. The framework allows for a specification of the fractal dimension D_f of fractal aggregates, which can be obtained prior to a simulation from measurements or literature (see e.g. [Goertz et al., 2012]). Totally rigid aggregates can be implemented, e.g. following [Binder, 2012]. We did not implement this approach because it is not as efficient for parallel code execution and several works show that agregates can undergo restructuring [Seto et al., 2012, Conchuir et al., 2014, Higashitani et al., 2001, Selomulya et al., 2001], which cannot be captured well by the approach of totally rigid aggregates. In section 1.4 we discussed formulas, which have been proposed to yield the mean settling rate in suspensions. Apart from the presented investigations in the behaviour of hydrodynamic clusters, we have conducted experiments and numerical simulations concerning the settling behaviour of polydisperse suspensions and their mean settling velocity, which we did not report on in this work. In the experiments we have tracked the motion of micron-sized glass beads (SiLibeads by Sigmund Lindner GmbH, Germany) settling at vanishing Reynolds number in a rectangular container (acrylic glass) filled with paraffin oil (Ph.Eur., DAB, low viscosity by Carl Roth, Germany). Experiments were made with varying solid volume fractions and with particle systems exhibiting different size distributions. Particle size fractions ranged from 100μm to 1000μm. To track differing settling behaviour of distinct size fractions, each size fraction was coloured differently. A side effect of the colour coating is that the used glass beads are hydrophobic. This way we could assume that besides hydrodynamic interactions no other destabilizing effects described in section 1.2 had to be considered. After dispersion, the settling process of the suspension was filmed. The resulting videos were processed by means of the open-source software OpenCV [Bradski and Kaehler, 2008], which was used to track particle motion. From this we could extract the mean settling velocity, as well as the velocities of defined fractions of the fastest settling particles in the filmed suspensions. Additionally, we performed numerical simulations of the described suspensions and of other suspensions for which we have data of the settling rates. In both, the experiments as well as the numerical simulations, we observed good agreement with the formula proposed by Richardson and Zaki [1997]. Moreover, for certain solid volume fractions (usually between 1% and $10-15\%$) we observed an increase in the mean settling velocity, which is not predicted by the standard hindered settling functions (cf. section 1.4). This effect has been reported for nonspherical particles [Kuusela et al., 2003, Herzhaft and Guazzelli, 1999] as well as for sub-micron sized particles [Bickert and Stahl, 1997, 1996] and is attributed to the formation of (hydrodynamic) clusters. Further research on this observation is intended, as now we have the experimental as well as the numerical framework to conduct corresponding investigations.

The combination of the SD method with a solution of convection-diffusion equations for solved substances together with the possibility to simulate birth, growth, fission and death of particles allows not only for the simulation of biological suspensions, but also for simulations of precipitation processes such as crystallisation. This requires numerical schemes for the kinetics of particle formation and the treatment of Brownian motion. For the latter, two schemes have been parallelized and included into the presented framework [Banchio and Brady, 2003, Ball and Melrose, 1997]. Recently we implemented the schemes given in [Cunha et al., 2002, Swan and Brady, 2007] for use in the parallel framework presented in this work. This allows for numerical studies on hydrodynamic wall effects with respect to the settling rate, sediment buildup, or investigations in bacterial attachment to and growth at flat container walls.

Bibliography

U. Kätzel, R. Bedrich, M. Stintz, R. Ketzmerick, T. Gottschalk-Gaudig, H. Barthel, Particle & Particle Systems Characterization 25 (2008) 9–18.

W. B. Russel, D. A. Saville, W. R. Schowalter, Colloidal Dispersions, Cambridge University Press, 1992.

J. F. Brady, G. Bossis, Annual Review of Fluid Mechanics 20 (1988) 111–157.

E. E. J. W. Verwey, J. T. G. Overbeek, J. T. J. T. G. Overbeek, Theory of the Stability of Lyophobic Colloids, DoverPublications. com, 1999.

D. L. Feke, N. D. Prabhu, J. A. Mann Jr, J. A. Mann III, The Journal of Physical Chemistry 88 (1984) 5735–5739.

H. Hamaker, Physica 4 (1937) 1058–1072.

R. Hogg, T. Healy, D. Fuerstenau, Transactions of the Faraday Society 62 (1966) 1638–1651.

G. Bell, S. Levine, L. McCartney, Journal of Colloid and Interface Science 33 (1970) 335–359.

S. Bhattacharjee, M. Elimelech, M. Borkovec, Croatica Chemica Acta 71 (1998) 883–903.

J. Bockris, M. Devanathan, K. Muller, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 274, The Royal Society, pp. 55–79.

D. Grahame, Chemical Reviews 41 (1947) 441–501.

Z. Stojek, in: Electroanalytical Methods, Springer, 2010, pp. 3–9.

B. J. Kirby, Micro-and Nanoscale Fluid Mechanics: Transport in Microfluidic Devices, Cambridge University Press, 2010.

M. Hecht, J. Harting, M. Bier, J. Reinshagen, H. J. Herrmann, Physical Review E 74 (2006) 021403.

M. Smoluchowski, Pisma Mariana Smoluchowskiego 1 (1924) 403–420.

R. O'Brien, D. Cannon, W. Rowlands, Journal of Colloid and Interface Science 173 (1995) 406–418.

E. Hückel, Physikalische Zeitschrift 25 (1924) 204–210.

D. Henry, Transactions of the Faraday Society 44 (1948) 1021–1026.

P. Leroy, C. Tournassat, M. Bizi, Journal of Colloid and Interface Science 356 (2011) 442–453.

A. S. Dukhin, P. J. Goetz, Characterization of Liquids, Nano-and Microparticulates, and Porous Bodies using Ultrasound, Elsevier, 2010.

J. Lyklema, Fundamentals of Interface and Colloid Science: Soft Colloids, volume 5, Academic press, 2005.

Zetasizer Nano ZS, Malvern Instruments GmbH, http://www.malvern.com/en/products/product-range/zetasizer-range/zetasizer-nano-range/default.aspx, 2015.

Acoustosizer, Colloidal Dynamics, http://www.colloidal-dynamics.com/acoustosizer.php, 2015.

Z. Zhu, Journal of Geoscience and Environment Protection 2 (2014) 13.

C. J. Van Oss, Cell Biophysics 14 (1989) 1–16.

C. Van Oss, Colloids and Surfaces A: Physicochemical and Engineering Aspects 78 (1993) 1–49.

M. Hermansson, Colloids and Surfaces B: Biointerfaces 14 (1999) 105–119.

D. Grasso, K. Subramaniam, M. Butkus, K. Strevett, J. Bergendahl, Reviews in Environmental Science and Biotechnology 1 (2002) 17–38.

V. Parsegian, T. Zemb, Current Opinion in Colloid & Interface Science 16 (2011) 618–624.

B. A. Todd, N. Y. Sidorova, D. C. Rau, C. Stanley, Wiley Encyclopedia of Chemical Biology (2008).

C. Zhang, in: Encyclopedia of Tribology, Springer, 2013, pp. 1704–1708.

F. Bülow, H. Nirschl, W. Dörfler, Preprint submitted to Particuology (2016).

J. A. Lewis, Journal of the American Ceramic Society 83 (2000) 2341–2359.

F. Bülow, P. Hamberger, H. Nirschl, W. Dörfler, Computers & Fluids 103 (2014) 156–165.

F. Bülow, P. Hamberger, H. Nirschl, W. Dörfler, Under consideration for publication in Computer Physics Communications (2016).

C.-C. Hsieh, S. Jain, R. G. Larson, The Journal of Chemical Physics 124 (2006) 044911.

M. Somasi, B. Khomami, N. J. Woo, J. S. Hur, E. S. Shaqfeh, Journal of Non-Newtonian Fluid Mechanics 108 (2002) 227–255.

J. F. Marko, E. D. Siggia, Macromolecules 28 (1995) 8759–8770.

E. S. Shaqfeh, G. H. McKinley, N. Woo, D. Nguyen, T. Sridhar, Journal of Rheology (1978-present) 48 (2004) 209–221.

W. Kuhn, F. Grün, Kolloid-Zeitschrift 101 (1942) 248–271.

E. M. Arruda, M. C. Boyce, Journal of the Mechanics and Physics of Solids 41 (1993) 389–412.

A. Cohen, Rheologica Acta 30 (1991) 270–273.

H. R. Warner Jr, Industrial & Engineering Chemistry Fundamentals 11 (1972) 379–387.

P. Cifra, T. Bleha, in: Macromolecular Symposia, volume 256, Wiley Online Library, pp. 105–111.

F. Bülow, S. Bäum, S. Lackner, H. Nirschl, W. Dörfler, In preparation for submission to Computers and Fluids (2016).

A. D. McNaught, IUPAC Compendium of Chemical Terminology: The Gold Book, International Union of Pure and Applied Chemistry, 2005.

P. Somasundaran, Encyclopedia of Surface and Colloid Science, volume 1, CRC press, 2006.

D. Feke, W. Schowalter, Journal of Fluid Mechanics 133 (1983) 17–35.

M. Smoluchowski, Zeitschrift für physikalische Chemie (1917).

A. Curtis, L. Hocking, Transactions of the Faraday Society 66 (1970) 1381–1390.

T. Van de Ven, S. Mason, Journal of Colloid and Interface Science 57 (1976) 505–516.

T. Van de Ven, S. Mason, Colloid and Polymer Science 255 (1977) 468–479.

P. Adler, Journal of Colloid and Interface Science 83 (1981) 106–115.

Q. Wang, Journal of Colloid and Interface Science 150 (1992) 418–427.

M. Vanni, G. Baldi, Advances in Colloid and Interface Science 97 (2002) 151–177.

M. Kobayashi, Journal of Applied Mechanics 11 (2008) 517–523.

B. Balakin, A. C. Hoffmann, P. Kosinski, Chemical Engineering Science 68 (2012) 305–312.

D. Feke, W. Schowalter, Journal of Colloid and Interface Science 106 (1985) 203–214.

E. Nazockdast, J. F. Morris, Physics of Fluids (1994-present) 25 (2013) 070601.

M. Mohammadi, E. D. Larson, J. Liu, R. G. Larson, The Journal of Chemical Physics 142 (2015) 024108.

R.-L. Qiao, Z. Li, C.-S. Wen, Journal of Colloid and Interface Science 202 (1998) 205–207.

R. H. Davis, Journal of Fluid Mechanics 145 (1984) 179–199.

M. Han, D. F. Lawler, Journal of Hydraulic Engineering 117 (1991) 1269–1289.

K. J. Farley, F. M. Morel, Environmental Science & Technology 20 (1986) 187–195.

R. E. Cunha, M. Fortuny, C. Dariva, A. F. Santos, Industrial & Engineering Chemistry Research 47 (2008) 7094–7103.

G. Batchelor, Journal of Fluid Mechanics 52 (1972) 245–268.

G. Batchelor, C.-S. Wen, Journal of Fluid Mechanics 124 (1982) 495–528.

J. F. Brady, L. J. Durlofsky, Physics of Fluids (1958-1988) 31 (1988) 717–727.

F. R. Cunha, G. C. Abade, A. J. Sousa, E. J. Hinch, Journal of Fluids Engineering 124 (2002) 957–968.

A. Sangani, A. Acrivos, International Journal of Multiphase Flow 8 (1982) 343–360.

H. Hasimoto, Journal of Fluid Mechanics 5 (1959) 317–328.

J. Richardson, W. Zaki, Chemical Engineering Research and Design 75 (1997) S82–S100.

N. Funamizu, T. Takakuwa, Chemical Engineering Science 50 (1995) 3025–3032.

J. M. Valverde, A. Castellanos, AIChE Journal 52 (2006) 838–842.

J. Ham, G. Homsy, International Journal of Multiphase Flow 14 (1988) 533–546.

R. H. Davis, H. Gecol, AIChE Journal 40 (1994) 570–575.

A. Cuthbertson, P. Dong, S. King, P. Davies, Coastal Engineering 55 (2008) 1197–1208.

J. I. Bhatty, Separation Science and Technology 21 (1986) 953–967.

E. Kuusela, J. Lahtinen, T. Ala-Nissila, Physical Review Letters 90 (2003) 094502.

R. Di Felice, International Journal of Multiphase Flow 20 (1994) 153–159.

M. B. Nikoo, N. Mahinpey, Biomass and Bioenergy 32 (2008) 1245–1254.

W. Doherty, A. Reynolds, D. Kennedy, Biomass and Bioenergy 33 (2009) 1158–1167.

H. Weifeng, S. Hongye, H. Yongyou, C. Jian, Chinese Journal of Chemical Engineering 14 (2006) 584–591.

R. J. Wooley, V. Putsche, Development of an ASPEN PLUS Physical Property Database for Biofuels Components, Citeseer, 1996.

M. Dosta, S. Antonyuk, E.-U. Hartge, S. Heinrich, Chemie Ingenieur Technik 86 (2014) 1073–1079.

S. Lackner, A. Terada, B. F. Smets, Water Research 42 (2008) 1102–1112.

M. Dosta, S. Heinrich, J. Werther, Powder Technology 204 (2010) 71–82.

N. Deen, M. V. S. Annaland, M. Van der Hoef, J. Kuipers, Chemical Engineering Science 62 (2007) 28–44.

A. Chakrabarty, T. Cagin, Polymer 51 (2010) 2786–2794.

M. Van der Hoef, M. van Sint Annaland, N. Deen, J. Kuipers, Annual Review of Fluid Mechanics 40 (2008) 47–70.

W. Ge, W. Wang, N. Yang, J. Li, M. Kwauk, F. Chen, J. Chen, X. Fang, L. Guo, X. He, et al., Chemical Engineering Science 66 (2011) 4426–4458.

B. Lu, W. Wang, J. Li, Chemical Engineering Science 66 (2011) 4624–4635.

Z. Li, J. Kessel, G. Grünewald, M. Kind, Drying Technology 31 (2013) 1888–1896.

M. Dosta, S. Antonyuk, S. Heinrich, Industrial & Engineering Chemistry Research 52 (2013) 11275–11281.

J. K. Dhont, An iIntroduction to Dynamics of Colloids, Elsevier, 1996.

C. Cercignani, The Boltzmann Equation, Springer, 1988.

S. Chen, G. D. Doolen, Annual Review of Fluid Mechanics 30 (1998) 329–364.

J. Tölke, M. Krafczyk, International Journal of Computational Fluid Dynamics 22 (2008) 443–456.

H. Oertel, M. Böhle, T. Ehret, U. Dohrmann, Strömungsmechanik, Springer, 2009.

H. K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson Education, 2007.

J. H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer Science & Business Media, 2012.

S. Turek, Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach, volume 6, Springer Science & Business Media, 1999.

R. Eymard, T. Gallouët, R. Herbin, Handbook of Numerical Analysis 7 (2000) 713–1018.

I. Newton, The mathematical principles of natural philosophy, Benjamin Motte, 1729.

A. Rao, Dynamics of Particles and Rigid Bodies: A Systematic Approach, Cambridge University Press, 2005.

T. Schlick, Molecular Modeling and Simulation: An Interdisciplinary Guide, volume 21, Springer Science & Business Media, 2010.

R. E. Rudd, J. Q. Broughton, Physical Review B 58 (1998) R5893.

P. A. Cundall, O. D. Strack, Geotechnique 29 (1979) 47–65.

P. A. Cundall, in: International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, volume 25, Elsevier, pp. 107–116.

A. Munjiza, P. W. Cleary, Engineering Computations 26 (2009) 698–743.

R. Moreno, M. Ghadiri, S. Antony, Powder Technology 130 (2003) 132–137.

S. Antonyuk, M. Khanal, J. Tomas, S. Heinrich, L. Mörl, Chemical Engineering and Processing: Process Intensification 45 (2006) 838–856.

Y. Muguruma, T. Tanaka, Y. Tsuji, Powder Technology 109 (2000) 49–57.

S.-S. Hsiau, S.-C. Yang, Chemical Engineering Science 58 (2003) 339–351.

F. Soulie, F. Cherblanc, M. S. El Youssoufi, C. Saix, International Journal for Numerical and Analytical Methods in Geomechanics 30 (2006) 213–228.

F. Y. Fraige, P. A. Langston, G. Z. Chen, Powder Technology 186 (2008) 224–240.

J.-F. Ferellec, G. R. McDowell, Granular Matter 12 (2010) 459–467.

H. Zhu, Z. Zhou, R. Yang, A. Yu, Chemical Engineering Science 63 (2008) 5728–5770.

H. Kruggel-Emden, T. Oschmann, Powder Technology 268 (2014) 219–236.

R. A. Gingold, J. J. Monaghan, Monthly Notices of the Royal Astronomical Society 181 (1977) 375–389.

M. Liu, G. Liu, Archives of Computational Methods in Engineering 17 (2010) 25–76.

R. Dalrymple, B. Rogers, Coastal Engineering 53 (2006) 141–147.

C. Hoefler, S. Braun, R. Koch, H.-J. Bauer, Journal of Engineering for Gas Turbines and Power 135 (2013) 011503.

G. Gompper, T. Ihle, D. Kroll, R. Winkler, in: Advanced Computer Simulation Approaches for Soft Matter Sciences III, Springer, 2009, pp. 1–87.

A. Malevanets, R. Kapral, The Journal of Chemical Physics 112 (2000) 7260–7269.

T. Ihle, D. Kroll, Physical Review E 63 (2001) 020201.

T. Ihle, D. M. Kroll, Physical Review E 67 (2003) 066705.

A. Malevanets, R. Kapral, The Journal of Chemical Physics 110 (1999) 8605–8613.

T. Ilina, M. Panfilov, M. Buès, I. Panfilova, Transport in Porous Media 71 (2008) 311–329.

T. B. Anderson, R. Jackson, Industrial & Engineering Chemistry Fundamentals 6 (1967) 527–539.

D. A. Drew, Mathematical modeling of two-phase flow., Technical Report, DTIC Document, 1982.

H. Enwald, E. Peirano, A.-E. Almstedt, International Journal of Multiphase Flow 22 (1996) 21–66.

J. Wang, M. Van der Hoef, J. Kuipers, Chemical Engineering Science 64 (2009) 622–625.

J. Wang, M. Van der Hoef, J. Kuipers, Chemical Engineering Science 65 (2010) 3772–3785.

F. Auzerais, R. Jackson, W. Russel, Journal of Fluid Mechanics 195 (1988) 437–462.

K. Gustavsson, J. Oppelstrup, Computing and Visualization in Science 3 (2000) 39–45.

W. Shuai, L. Guodong, L. Huilin, X. Pengfei, Y. Yunchao, D. Gidaspow, Powder Technology 225 (2012) 176–189.

T. R. Camp, P. C. Stein, Journal of the Boston Society of Civil Engineers 85 (1943) 219–37.

T. A. Kramer, M. M. Clark, Journal of Colloid and Interface Science 227 (2000) 251–261.

J. Cheng, C. Yang, Z.-S. Mao, Chemical Engineering Science 68 (2012) 469–480.

R. McGraw, Aerosol Science and Technology 27 (1997) 255–265.

R. Fan, D. L. Marchisio, R. O. Fox, Powder technology 139 (2004) 7–20.

K. Ichiki, H. Hayakawa, Physical Review E 52 (1995) 658.

S. Kim, S. J. Karrila, Microhydrodynamics: Principles and Selected Applications, Courier Corporation, 2013.

N. Nishimura, Applied Mechanics Reviews 55 (2002) 299–324.

L. C. Wrobel, The Boundary Element Method, Applications in Thermo-fluids and Acoustics, volume 1, John Wiley & Sons, 2002.

C. Pozrikidis, Acta Mechanica 194 (2007) 213–231.

L. Durlofsky, J. F. Brady, G. Bossis, Journal of Fluid Mechanics 180 (1987) 21–49.

A. Sierou, J. Brady, Journal of Rheology (1978-present) 46 (2002) 1031–1056.

D. L. Ermak, J. McCammon, The Journal of Chemical Physics 69 (1978) 1352–1360.

D. Heyes, J. Melrose, Journal of Non-newtonian Fluid Mechanics 46 (1993) 1–28.

M. Hütter, Journal of Colloid and Interface Science 231 (2000) 337–350.

K. Sekine, Bioelectrochemistry 52 (2000) 1–7.

D. R. Foss, J. F. Brady, Journal of Fluid Mechanics 407 (2000) 167–200.

H. Noguchi, M. Takasu, Physical Review E 64 (2001) 041913.

D. S. Bolintineanu, G. S. Grest, J. B. Lechman, F. Pierce, S. J. Plimpton, P. R. Schunk, Computational Particle Mechanics 1 (2014) 321–356.

S. Plimpton, Journal of Computational Physics 117 (1995) 1–19.

R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, Journal of Computational Physics 169 (2001) 363–426.

C. S. Peskin, Acta Numerica 11 (2002) 479–517.

R. Mittal, G. Iaccarino, Annual Review of Fluid Mechanics 37 (2005) 239–261.

M. Uhlmann, Journal of Computational Physics 209 (2005) 448–476.

N. Sharma, N. A. Patankar, Journal of Computational Physics 205 (2005) 439–457.

R. Münster, O. Mierka, S. Turek, International Journal for Numerical Methods in Fluids 69 (2012) 294–313.

J. Yang, F. Stern, Journal of Computational Physics 231 (2012) 5029–5061.

N. G. Deen, M. van Sint Annaland, J. Kuipers, Chemical Engineering Science 64 (2009) 2186–2201.

A. A. Shirgaonkar, M. A. MacIver, N. A. Patankar, Journal of Computational Physics 228 (2009) 2366–2390.

M. Uhlmann, Physics of Fluids (1994-present) 20 (2008) 053305.

I. Borazjani, L. Ge, F. Sotiropoulos, Journal of Computational physics 227 (2008) 7587–7620.

Y. Tsuji, T. Kawaguchi, T. Tanaka, Powder technology 77 (1993) 79–87.

S. Ergun, Chemical engineering progress 48 (1952).

C. Wen, Y. Yu, in: Chem. Eng. Prog. Symp. Ser., volume 62, p. 100.

M. Xu, F. Chen, X. Liu, W. Ge, J. Li, Chemical Engineering Journal 207 (2012) 746–757.

R. Chhabra, L. Agarwal, N. Sinha, Powder Technology 101 (1999) 288–295.

J. Kim, S. Lee, Powder Technology 261 (2014) 22–32.

S. Tran-Cong, M. Gay, E. E. Michaelides, Powder Technology 139 (2004) 21–32.

E. Loth, Powder Technology 182 (2008) 342–353.

A. Hölzer, M. Sommerfeld, Powder Technology 184 (2008) 361–365.

X. R. Fernández, H. Nirschl, Chemical Engineering Science 94 (2013) 7–19.

K. Chu, B. Wang, A. Yu, A. Vince, Powder Technology 193 (2009) 235–247.

M. Afkhami, A. Hassanpour, M. Fairweather, D. Njobuenwu, Computers & Chemical Engineering 78 (2015) 24–38.

M. Zhang, K. Chu, F. Wei, A. Yu, Powder Technology 184 (2008) 151–165.

S. Yang, K. Luo, M. Fang, K. Zhang, J. Fan, Chemical Engineering Journal 236 (2014) 158–170.

D. Jajcevic, E. Siegmann, C. Radeke, J. G. Khinast, Chemical Engineering Science 98 (2013) 298–310.

M. Andrews, P. O'Rourke, International Journal of Multiphase Flow 22 (1996) 379–402.

D. Snider, Journal of Computational Physics 170 (2001) 523–549.

N. Patankar, D. Joseph, International Journal of Multiphase Flow 27 (2001) 1659–1684.

P. J. O'Rourke, D. M. Snider, Chemical Engineering Science 65 (2010) 6014–6028.

P. J. O'Rourke, D. M. Snider, Powder Technology 256 (2014) 39–51.

D. Snider, K. Williams, R. A. Johnson, in: ASME 2004 Heat Transfer/Fluids Engineering Summer Conference, American Society of Mechanical Engineers, pp. 777–785.

C. Chen, J. Werther, S. Heinrich, H.-Y. Qi, E.-U. Hartge, Powder Technology 235 (2013) 238–247.

RYUON, http://ryuon.sourceforge.net/, 2014.

J. Barnes, P. Hut, Nature 324 (1986) 446–449.

A. Sierou, J. F. Brady, Journal of Fluid Mechanics 448 (2001) 115–146.

K. Ichiki, Journal of Fluid Mechanics 452 (2002) 231–262.

F. Bülow, H. Nirschl, W. Dörfler, European Journal of Mechanics-B/Fluids 50 (2015) 19–26.

E. E. Michaelides, Particles, Bubbles & Drops, World Scientific, Singapore, 2006.

M. Faletra, J. S. Marshall, M. Yang, S. Li, Journal of Fluid Mechanics 769 (2015) 79–102.

T. N. Phung, J. F. Brady, G. Bossis, Journal of Fluid Mechanics 313 (1996) 181–207.

A. J. Banchio, J. F. Brady, The Journal of Chemical Physics 118 (2003) 10323.

A. J. Banchio, J. Gapinski, A. Patkowski, W. Häußler, A. Fluerasu, S. Sacanna, P. Holmqvist, G. Meier, M. P. Lettinga, G. Nägele, Physical Review Letters 96 (2006) 138303.

Y. M. Harshe, L. Ehrl, M. Lattuada, Journal of Colloid and Interface Science 352 (2010) 87–98.

Y. M. Harshe, M. Lattuada, Langmuir 28 (2011) 283–292.

R. Jones, The Journal of Chemical Physics 115 (2001) 5319.

R. Seto, R. Botet, G. K. Auernhammer, H. Briesen, The European Physical Journal E 35 (2012) 1–12.

N. J. Wagner, J. F. Brady, Physics Today 62 (2009) 27–32.

F. Cunha, A. Sousa, E. Hinch, Chemical Engineering Communications 189 (2002) 1105–1129.

C. Chang, R. L. Powell, Physics of Fluids 6 (1994) 1628.

X. Xu, S. A. Rice, A. R. Dinner, Proceedings of the National Academy of Sciences 110 (2013) 3771–3776.

L. J. Durlofsky, J. F. Brady, Journal of Fluid Mechanics 200 (1989) 39–67.

M. Kopp, F. Höfling, The European Physical Journal Special Topics 210 (2012) 101–117.

MPI: A Message-Passing Interface Standard, http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf, 2013.

D. Jeffrey, J. Morris, J. F. Brady, Physics of Fluids A: Fluid Dynamics (1989-1993) 5 (1993) 2317–2325.

J. F. Brady, The Journal of Chemical Physics 98 (1993) 3335–3341.

Y. Yurkovetsky, J. F. Morris, Journal of Rheology (1978-present) 52 (2008) 141–164.

D. Jeffrey, Y. Onishi, Journal of Fluid Mechanics 139 (1984) 261–290.

D. Jeffrey, Physics of Fluids A: Fluid Dynamics 4 (1992) 16.

B. Cichocki, M. Ekiel-Jeżewska, E. Wajnryb, The Journal of Chemical Physics 111 (1999) 3265.

F. L. Leite, C. C. Bueno, A. L. Da Róz, E. C. Ziemath, O. N. Oliveira, International Journal of Molecular Sciences 13 (2012) 12773–12856.

B. Schäfer, M. Hecht, J. Harting, H. Nirschl, Journal of Colloid and Interface Science 349 (2010) 186–195.

A. Rosas, K. Lindenberg, arXiv preprint cond-mat/0309070 (2003).

R. H. Davis, J.-M. Serayssol, E. Hinch, Journal of Fluid Mechanics 163 (1986) 479–497.

W. Zhang, R. Noda, M. Horio, Powder Technology 158 (2005) 92–101.

R. Seto, R. Mari, J. F. Morris, M. M. Denn, Physical Review Letters 111 (2013) 218301.

S. Balay, J. Brown, , K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Users Manual, Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

E. Fehlberg, Computing 6 (1970) 61–71.

IC2 user guide, http://www.scc.kit.edu/scc/docs/IC/ug/ugic2.pdf, 2013.

HC3 user guide, http://www.scc.kit.edu/scc/docs/HP-XC/ug/ug3k.pdf, 2014.

A. Goldman, R. Cox, H. Brenner, Chemical Engineering Science 21 (1966) 1151–1170.

J. Nitsche, G. Batchelor, Journal of Fluid Mechanics 340 (1997) 161–175.

B. Metzger, M. Nicolas, E. Guazzelli, Journal of Fluid Mechanics 580 (2007) 283–301.

G. C. Abade, F. R. Cunha, Computer Methods in Applied Mechanics and Engineering 196 (2007) 4597–4612.

K. Ichiki, Journal of Fluid Mechanics 452 (2002) 231–262.

J. A. Board, J. W. Causey, J. F. Leathrum, Chemical Physics Letters 198 (1992) 89–94.

M. Winkel, R. Speck, H. Hübner, L. Arnold, R. Krause, P. Gibbon, Computer Physics Communications 183 (2012) 880–889.

R. Hoffmann, Granular Matter 8 (2006) 151–157.

J. Dubinski, New Astronomy 1 (1996) 133–147.

J. E. Barnes, Journal of Computational Physics 87 (1990) 161–170.

L. Greengard, V. Rohklin, Journal of Computational Physics 73 (1987) 325–348.

J. Carrier, L. Greengard, V. Rohklin, SIAM J. Sci. Stat. Comput. 9 (1988) 669–686.

G. Machu, W. Meile, L. G. Nitsche, U. Schaflinger, Journal of Fluid Mechanics 447 (2001) 299–336.

S. Ogata, T. J. Campbell, R. K. Kalia, A. Nakano, P. Vashishta, S. Vemparala, Computer Physics Communications 153 (2003) 445–461.

J. Kurzak, B. M. Pettitt, Journal of Parallel and Distributed Computing 65 (2005) 870–881.

E. Darve, Journal of Computational Physics 160 (2000) 195–240.

A. Grama, V. Kumar, A. Sameh, Parallel Computing 24 (1998) 797–822.

P. Koumoutsakos, Annu. Rev. Fluid. Mech. 37 (2005) 457–487.

J. P. Singh, C. Holt, T. Totsuka, A. Gupta, J. Hennessy, Journal of Parallel and Distributed Computing 27 (1995) 118–141.

L. Durlofsky, J. F. Brady, G. Bossis, Journal of Fluid Mechanics 180 (1987) 21–49.

J. F. Brady, G. Bossis, Ann. Rev. Fluid Mech. 20 (1988) 111–157.

A. Sierou, J. F. Brady, Journal of Fluid Mechanics 448 (2001) 115–146.

D. Saintillan, E. Darve, E. S. Shaqfeh, Physics of Fluids (1994-present) 17 (2005) 033301.

J. E. Butler, E. S. Shaqfeh, The Journal of Chemical Physics 122 (2004) 014901.

A. S. Sangani, G. Mo, Physics of Fluids (1994-present) 8 (1996) 1990–2010.

J. P. Hernández-Ortiz, J. J. de Pablo, M. D. Graham, The Journal of Chemical Physics 125 (2006) 164906.

S. Lomholt, M. R. Maxey, Journal of Computational Physics 184 (2003) 381–405.

A.-K. Tornberg, L. Greengard, Journal of Computational Physics 227 (2008) 1613–1619.

M. Maxey, B. Patel, International Journal of Multiphase Flow 27 (2001) 1603–1626.

D. J. Jeffrey, Y. Onishi, Journal of Fluid Mechanics 139 (1984) 261–290.

D. J. Jeffrey, Phys. Fluids A 4 (1) (1992) 16–29.

B. Cichocki, M. L. Ekiel-Jezewska, E. Wajnryb, Journal of Chemical Physics 111 (1999) 3265–3273.

J. Nitsche, G. Batchelor, Journal of Fluid Mechanics 340 (1997) 161–175.

G. C. Abade, F. R. Cunha, Comput. Methods Appl. Mech. Engrg. 196 (2007) 4597–4612.

K. Briggs, E. Smith, I. Snook, W. van Megen, Physics Letters A 154 (1991) 149–153.

J. K. Salmon, M. S. Warren, Journal of Computational Physics 111 (1994) 136–155.

R. Sedgewick, K. Wayne, Algorithms, 4th Edition, Addison-Wesley, 2011.

G. Machu, W. Meile, L. G. Nitsche, U. Schaflinger, Journal of Fluid Mechanics 447 (2001) 299–336.

G. Subramanian, D. Koch, Journal of Fluid Mechanics 603 (2008) 63–100.

B. Metzger, M. Nicolas, E. Guazzelli, Journal of Fluid Mechanics 580 (2007) 283–301.

F. Pignatel, M. Nicolas, E. Guazzelli, Journal of Fluid Mechanics 671 (2011) 34–51.

K. Adachi, S. Kiriyama, N. Yoshioka, Chemical Engineering Science 33 (1978) 115–121.

T. Bosse, L. Kleiser, C. Härtel, Phys. Fluids 17 (2005) 1–17.

S. Alabrudzinski, M. L. Ekiel-Jezewska, D. Chehata-Gomez, T. A. Kowalewski, Phys. Fluids 21 (2009) 073302.

M. L. Ekiel-Jezewska, B. Metzger, E. Guazzelli, Phys. Fluids 18 (2006) 038104.

J. Park, B. Metzger, E. Guazzelli, J. Butler, Journal of Fluid Mechanics 648 (2010) 351–362.

M. Feist, F. Keller, H. Nirschl, W. Dörfler, The Canadian Journal of Chemical Engineering 89 (2011) 682–690.

A. J. Goldman, R. G. Cox, H. Brenner, Chemical Engineering Science 21 (1966) 1151–1170.

C. Pozrikidis, Acta Mechanica 194 (2007) 213–231.

L. Hocking, Journal of Fluid Mechanics 20 (1964) 129–139.

A. Myłyk, W. Meile, G. Brenn, M. L. Ekiel-Jeżewska, Physics of Fluids (1994-present) 23 (2011) 063302.

A. Gerstlauer, C. Gahn, H. Zhou, M. Rauls, M. Schreiber, Chemical Engineering Science 61 (2006) 205–217.

E. Hollander, J. Derksen, L. Portela, H. Van den Akker, AIChE Journal 47 (2001) 2425–2440.

E. Torfs, A. Dutta, I. Nopens, Chemical Engineering Science 70 (2012) 176–187.

B. O. Conchuir, Y. M. Harshe, M. Lattuada, A. Zaccone, Industrial & Engineering Chemistry Research 53 (2014) 9109–9119.

K. Higashitani, K. Iimura, H. Sanda, Chemical Engineering Science 56 (2001) 2927–2938.

M. Zeidan, B. Xu, X. Jia, R. Williams, Chemical Engineering Research and Design 85 (2007) 1645–1654.

A. K. Manounou, S. Rémond, Physica A: Statistical Mechanics and its Applications 412 (2014) 66–83.

A. Khadilkar, P. L. Rozelle, S. V. Pisupati, Powder Technology 264 (2014) 216–228.

D. Sato, M. Kobayashi, Y. Adachi, Journal of Colloid and Interface Science 272 (2004) 345–351.

W. Lin, M. Kobayashi, M. Skarba, C. Mu, P. Galletto, M. Borkovec, Langmuir 22 (2006) 1038–1047.

L. Hocking, Journal of Engineering Mathematics 7 (1973) 207–221.

S. Pieper, N. Kirchhoff, H.-J. Schmid, Rheologica Acta 54 (2015) 69–75.

H. Luo, C. Pozrikidis, Journal of Fluid Mechanics 581 (2007) 129–156.

M. Ernst, M. Dietzel, M. Sommerfeld, Acta Mechanica 224 (2013) 2425–2449.

MATLAB, version 7.12.0.635 (R2011a), The MathWorks Inc., Natick, Massachusetts, 2011.

Collection of circle fit functions by Nikolai Chernov, http://people.cas.uab.edu/ mosya/cl/MATLABcircle.html, last access: 2015.

G. Zeichner, W. Schowalter, AIChE Journal 23 (1977) 243–254.

J. C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, SIAM Journal on Optimization 9 (1998) 112–147.

N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger, The R*-tree: An Efficient and Robust Access Method for Points and Rectangles, volume 19, ACM, 1990.

I. Klapper, J. Dockery, SIAM Review 52 (2010) 221–265.

J. B. Xavier, C. Picioreanu, M. Van Loosdrecht, Environmental Microbiology 7 (2005a) 1085–1103.

J. B. Xavier, C. Picioreanu, S. A. Rani, M. C. van Loosdrecht, P. S. Stewart, Microbiology 151 (2005b) 3817–3832.

A. G. Vanakaras, Langmuir 22 (2006) 88–93.

K. K. Kabanemi, J.-F. Hétu, The Journal of Chemical Physics 136 (2012) 074903.

V. Goertz, A. Gutsche, N. Dingenouts, H. Nirschl, The Journal of Physical Chemistry C 116 (2012) 26938–26946.

C. Binder, Settling of Fractal Aggregates in Viscous Media, Ph.D. thesis, KIT, 2012.

C. Selomulya, R. Amal, G. Bushell, T. D. Waite, Journal of Colloid and Interface Science 236 (2001) 67–77.

G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, " O'Reilly Media, Inc.", 2008.

B. Herzhaft, É. Guazzelli, Journal of Fluid Mechanics 384 (1999) 133–158.

G. Bickert, W. Stahl, Particle & Particle Systems Characterization 14 (1997) 142–147.

G. Bickert, W. Stahl, Chemie Ingenieur Technik 68 (1996) 1459–1462.

R. Ball, J. R. Melrose, Physica A: Statistical Mechanics and its Applications 247 (1997) 444–472.

J. W. Swan, J. F. Brady, Physics of Fluids (1994-present) 19 (2007) 113306.