

Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 293

Routinen zur Auswertung von Simulationen mit ESPResSo zur virtuellen Weiterentwicklung von Innenraumfiltern

Dominik Lekar

| | |
|---------------------|--|
| Studiengang: | Softwaretechnik |
| Prüfer/in: | Prof. Dr. rer. nat. habil. Miriam Mehl |
| Betreuer/in: | Dipl.-Ing. Carolin Schober |
| Beginn am: | 7. Januar 2016 |
| Beendet am: | 8. Juli 2016 |
| CR-Nummer: | I.6.6 |

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich während der Anfertigung dieser Arbeit unterstützt und motiviert haben. Mein besonderer Dank gilt Prof. Dr. rer. nat. habil. Miriam Mehl und Dipl.-Ing. Carolin Schober für die Ermöglichung dieser Arbeit und der hilfsbereiten Unterstützung.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 9 |
| 2 | Grundlagen | 11 |
| 2.1 | Filtration | 11 |
| 2.2 | Molekulardynamik | 14 |
| 2.3 | ESPresSo | 16 |
| 3 | Implementierung | 19 |
| 3.1 | Funktionsumfang des Skripts zu Beginn der Arbeit | 19 |
| 3.2 | Erweiterung des Skripts | 20 |
| 3.3 | Aufbau des Skripts | 21 |
| 3.4 | Bedienung | 22 |
| 3.5 | Berechnung der Kollision von Partikeln | 23 |
| 4 | Simulationsergebnisse | 27 |
| 4.1 | Absoluter Abscheidegrad | 29 |
| 4.2 | zeitliche Entwicklung des Abscheidegrades | 35 |
| 4.3 | Strukturaufbau der abgeschiedenen Partikel und dessen Dichte | 38 |
| 5 | Performance | 43 |
| 5.1 | Einflüsse auf die Gesamtlaufzeit einer Simulation | 43 |
| 5.2 | Vergleich der Performance mit dem ursprünglichen Skript | 44 |
| 5.3 | Anteil der Umgebungskorrektur an der Gesamtlaufzeit | 44 |
| 6 | Zusammenfassung und Ausblick | 47 |
| | Literaturverzeichnis | 49 |

Abbildungsverzeichnis

| | | |
|------|---|----|
| 2.1 | konkurrierende Anforderungen an ein Filtermedium, die die Effizienz eines Filters beschreiben | 11 |
| 2.2 | graphische Darstellung der Abscheidemechanismen in der Luftfiltration . . . | 12 |
| 2.3 | graphische Darstellung des möglichen Verlaufs der Abscheidegrade von Diffusion und Trägheit, in Abhängigkeit von der Partikelgröße und der daraus resultierenden MPPS | 13 |
| 2.4 | graphischer Verlauf des Lennard-Jones-Potential [15] | 15 |
| 2.5 | D3Q19 Model der Lattice-Boltzmann-Methode [16] | 16 |
| 2.6 | Coarse-Graining-Modell von ESPResSo [19] | 16 |
| 2.7 | Aufbau und Architektur[20] von ESPResSo | 17 |
| 3.1 | visueller Aufbau einer Simulation, die mit dem ursprünglichen Skript erstellt wurde | 19 |
| 3.2 | Darstellung einer Faser, an der unterschiedliche Partikeltypen abgeschieden wurden | 21 |
| 3.3 | Darstellung der Aufteilung des Skripts und der strukturelle Aufbau des Inhalts beider Teile | 22 |
| 3.4 | Graphische Darstellung zur Bestimmung des Radius für ein Faserpartikel . . . | 24 |
| 3.5 | Veranschaulichung der Problematik bei der berechneten Umgebung von ESPResSo | 25 |
| 4.1 | Darstellung einer zylinderförmigen Faser durch Faserpartikel | 29 |
| 4.2 | Abscheidegrade für unterschiedliche Stokes-Zahlen, bei unterschiedlicher Partikelanzahl | 30 |
| 4.3 | Abscheidegrade des kleinsten Partikeltyps für unterschiedliche Modellierungen | 31 |
| 4.4 | Abscheidegrade des größten Partikeltyps für unterschiedliche Modellierungen | 32 |
| 4.5 | Abscheidegrad von Partikeln bei unterschiedlichen Faserstrukturen | 32 |
| 4.6 | Abscheidegrad für unterschiedliche Stokes-Zahlen, abhängig von unterschiedlichen Fluidgeschwindigkeiten | 33 |
| 4.7 | Verlauf der Effizienz unterschiedlicher Faserstrukturen, in Abhängigkeit von der zurückgelegten Simulationszeit | 36 |
| 4.8 | Effizienz der Einzelfaser für unterschiedlichen Fluidgeschwindigkeiten, in Abhängigkeit von der zurückgelegten Simulationszeit | 37 |
| 4.9 | Struktur-Aufbau mit zwei Fasern | 38 |
| 4.10 | Struktur-Aufbau einer Einzelfaser | 39 |
| 4.11 | Struktur-Aufbau einer Einzelfaser mit langsameren Fluid | 39 |

| | | |
|------|--|----|
| 4.12 | Dichte der Partikelkörper aus den Simulation zur zeitlichen Effizienz | 41 |
| 5.1 | Laufzeit einer Simulation, in Abhängigkeit von der Anzahl der zugegebenen Partikeln | 43 |
| 5.2 | zeitlicher Anteil der Umgebungskorrektur | 45 |
| 5.3 | Darstellung der benötigten Rechenzeit der Umgebungskorrektur, im Verhältnis zur Gesamtlaufzeit | 46 |

1 Einleitung

Die stetig steigende Luftverschmutzung ist schon seit längerem, besonders in großen Ballungsgebieten, zu einem ernstzunehmenden Thema geworden. Feinstaub kann sowohl direkt abgegeben werden (primärer Feinstaub), etwa durch Kraftstoffverbrennung oder Industrieverarbeitungsprozesse, als auch durch gasförmige Vorläufersubstanzen entstehen (sekundärer Feinstaub) [1]. Die Schadstoffkonzentration ist somit umgebungs- und klimaabhängig.

Feinstaub wird, abhängig von dem Durchmesser der entsprechenden Partikeln, in unterschiedliche Klassen gruppiert. Darunter befinden sich die Klassen $PM_{2.5}$ (Particulate Matter), deren zugehörige Partikeln einen Durchmesser von $2,5 \mu m$ oder kleiner besitzen und PM_{10} , deren Partikeldurchmesser zwischen $2,5 \mu m$ und $10 \mu m$ liegt. Die Größe der Partikel spielt vor allem eine gesundheitliche Rolle. Partikel der Klasse PM_{10} sind inhalierbar und können bis zu den Bronchien gelangen. $PM_{2.5}$ wird als lungengängiger Feinstaub bezeichnet, da er bis in die Lungenbläschen durchdringen kann. Noch kleinere Partikel ($PM_{0.1}$) gelangen sogar in den Blutkreislauf [2].

Besonders im asiatischen Raum liegen die $PM_{2.5}$ -Feinstaubwerte weit über dem empfohlenen Durchschnitt. In Peking erreicht beispielsweise der Feinstaubwert einen Jahresdurchschnitt von knapp $100 \frac{\mu g}{m^3}$ [3]. Im Vergleich dazu liegen die jährlichen Werte in europäischen Städten bei circa $10-20 \frac{\mu g}{m^3}$ [4] [5]. Das ist zwar innerhalb des gesetzlichen Rahmens, die Weltgesundheitsorganisation (WHO) empfiehlt jedoch einen jährlichen Durchschnitt von $10 \frac{\mu g}{m^3}$ [6].

Insassen von Fahrzeugen sind von der hohen Belastung besonders betroffen, da die angesaugte Luft sich im Innenraum sammeln kann und zu einer erhöhten Konzentration führen kann. Deshalb ist es notwendig, dass Fahrzeuge mit einem Innenraumfilter ausgestattet werden. Sie dienen nicht nur dazu, die Insassen, sondern auch das Lüftungssystem vor Staub, Pollen, Bakterien und sonstigen Partikeln zu schützen. Die derzeitige Entwicklung von Filtermedien erfolgt durch Tests, bei denen das Filtermedium in einer künstlichen Testumgebung mit Partikeln beladen wird und anschließend die Anzahl der ungefilterten Partikeln ermittelt wird [7]. Dieses Verfahren erfordert jedoch einen hohen Versuchsaufwand und lässt kaum Rückschlüsse über das Verhalten der Partikeln während der Beladung ziehen. Um die Mechanismen auf der Mikroskala besser verstehen zu können, wird ein vorhandenes Skript für die Software ESPReso erweitert, welches die Beladung von einzelnen Fasern mit mehreren Partikeln (in kleinen und einfachen Modellen) simuliert. Anschließend werden die, aus der Simulation gewonnenen, Daten ausgewertet und anschaulich dargestellt. Anhand der graphischen Darstellungen sollen Trends in Abhängigkeit von verschiedenen Parametern charakterisiert werden. Weiterhin kann die Visualisierung einzelner Partikeltrajektorien zum Verständnis verschiedener Abscheidemechanismen beitragen.

2 Grundlagen

2.1 Filtration

In der Filtration gibt es eine Unterscheidung von drei verschiedenen Filter-Kategorien:

- Grobstaubfilter (Partikeldurchmesser $> 10 \mu\text{m}$)
- Feinstaubfilter (Partikeldurchmesser 1 bis $10 \mu\text{m}$)
- Schwebstofffilter (Partikeldurchmesser $< 1 \mu\text{m}$)

Jede dieser Kategorien ist nochmals in mehrere Filterklassen unterteilt, welche sich durch den Abscheidegrad unterscheiden. Der Abscheidegrad beschreibt dabei den prozentualen Anteil von Partikeln, die von dem Filtermedium aufgefangen werden. In Bezug zur aufgegebenen Menge, ergeben sich laut den europäischen Normen insgesamt 17 verschiedene Partikelfilterklassen [8].

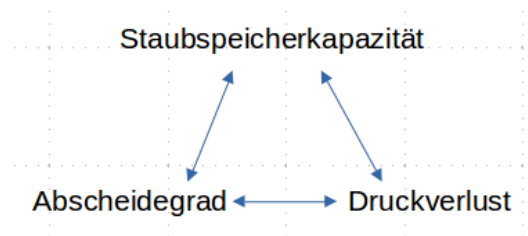


Abbildung 2.1: konkurrierende Anforderungen an ein Filtermedium, die die Effizienz eines Filters beschreiben

In Abbildung 2.1 sind die wichtigsten Anforderungen eines effizienten Filters dargestellt. Diese hängen unter anderem von dem verwendeten Filtermaterial und der Filterstruktur ab [9]. Der Abscheidegrad sollte möglichst hoch sein, um die Gesundheit der Insassen möglichst wenig zu beeinträchtigen. Die Staubspeicherkapazität sagt aus, wie viel Staub der Filter auffangen kann, bis er gewechselt werden muss. Unter dem Druckverlust versteht man die benötigte Energie der Luft, um in den Innenraum zu gelangen. Ein effizientes Filtermedium zu konzipieren ist daher nicht trivial, da die genannten Punkte konkurrierend sind. Beispielsweise kann die Filtereffizienz durch eine dichtere Faserstruktur gesteigert werden. Dies führt jedoch zu einem höheren Druckverlust und erhöht somit auch den Energiebedarf. Der Druckverlust kann

wiederm durch ein dünneres Filtermedium reduziert werden. In einem dünneren Filtermedium kann jedoch weniger Feinstaub angelagert werden, sodass der Filter öfters getauscht werden muss.

Heutzutage werden häufig mehrlagige Filtermedien aus synthetischem Vlies verwendet und somit eine Tiefenfiltration erreicht. Dies bedeutet, dass sich die Partikeln im Inneren des Mediums anlagern und im Gegensatz zu einem Oberflächenfilter (bei gleichem Druckverlust) mehr Staub aufgenommen werden kann [10]. Durch den stetig ansteigenden Anteil von $PM_{2.5}$ Partikeln, ändern sich jedoch auch die Anforderungen an die Filtermedien. Vor allem wird der Aufbau der Faserstruktur immer komplexer, um den Anforderungen gerecht zu werden. Auch die Wahl eines geeigneten Material wird schwieriger, da immer neuere Materialien zur Verfügung stehen.

2.1.1 Abscheidemechanismen in der Luftfiltration

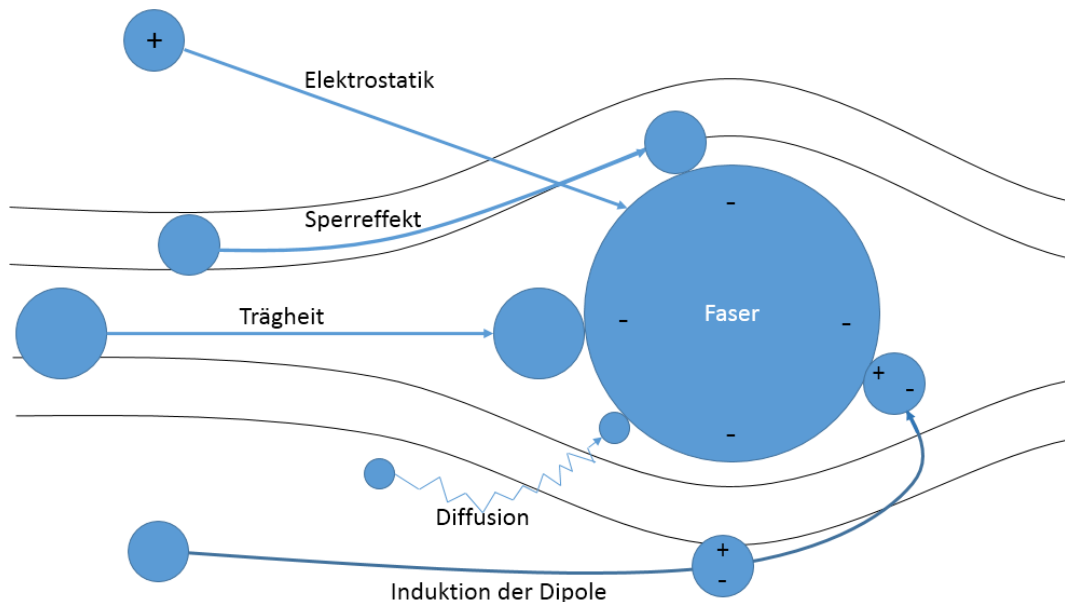


Abbildung 2.2: graphische Darstellung der Abscheidemechanismen in der Luftfiltration

Nähert sich ein Partikel einer Faser, so können unterschiedliche Effekte beobachtet werden. Diese sind in Abbildung 2.2 dargestellt. Abhängig von der Trägheit des Partikels, fließt dieses entweder entlang der Strömung um die Faser herum, oder aber kollidiert mit dieser. Besonders träge Partikel brauchen sehr lange, bis sie sich an die Richtungsänderung der Strömung angepasst haben. Diese bewegen sich daher direkt auf die Faser zu.

Als Sperreffekt wird die Kollision am äußeren Rand der Faser bezeichnet. Dies betrifft Partikel,

deren Trägheit gering genug ist, um sich der Strömung frühzeitig anzupassen. Diese können dennoch, aufgrund der geometrischen Ausdehnung der Partikel, an der Faser abgeschieden werden.

Zu den elektrostatischen Effekten gehören die Anziehung eines Partikels nach dem Coulomb'schen Gesetz und die Induktion von Dipolen eines Partikels. Falls Faser und Partikel eine elektrostatische Ladung besitzen, so wird das Partikel bei gleicher Ladung und gleicher Polarität von der Faser abgestoßen. Entsprechend wird es bei unterschiedlicher Ladung zur Faser angezogen. Bei einem ungeladenen Partikel können Dipole durch die geladene Faser induziert werden. Als Folge kann das Partikel zur Faser angezogen werden, nachdem es an dieser bereits vorbei geströmt ist.

Ein weiterer Effekt ist die Diffusion. Unter dem Begriff versteht man die Kollision mit den Molekülen aus dem Fluid, aufgrund von Temperaturfluktuationen. Sie beeinflusst besonders kleine Partikel. Diese haben eine dementsprechend geringe Masse und sind daher wenig träge. Die Kollision mit den Molekülen erhöht deren Abscheidegrad, da sie aus dem Strömungsverlauf in eine Faser gestoßen werden können [11].

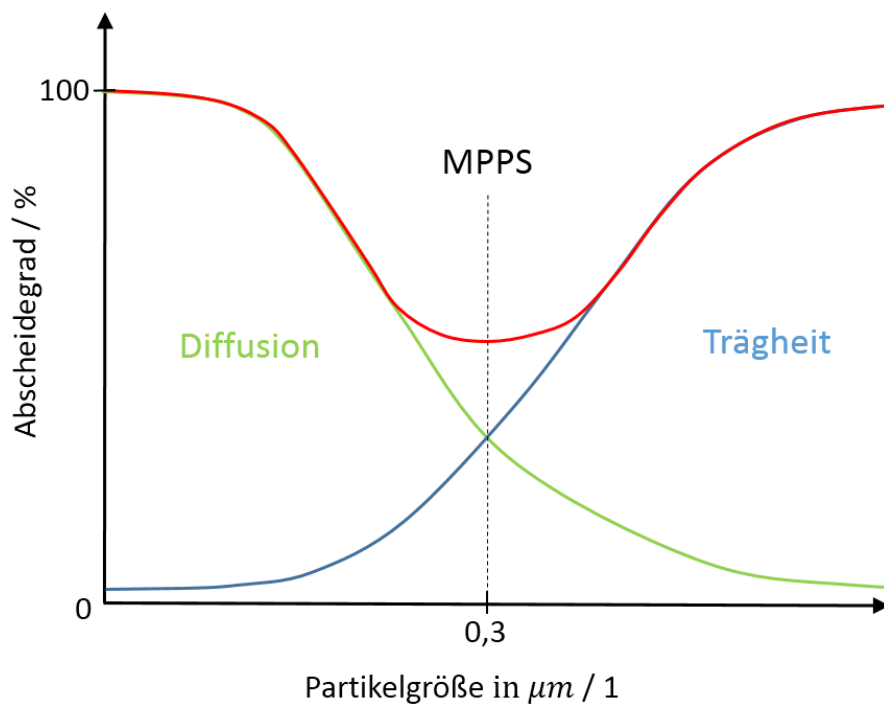


Abbildung 2.3: graphische Darstellung des möglichen Verlaufs der Abscheidegrade von Diffusion und Trägheit, in Abhängigkeit von der Partikelgröße und der daraus resultierenden MPPS

In Abbildung 2.3 ist der mögliche Verlauf der Abscheidegrade von Diffusion und Trägheit, in Abhängigkeit von der Partikelgröße, dargestellt. Der Abscheidegrad durch die Diffusion (grüne

Kurve) sinkt hierbei mit zunehmender Partikelgröße. Zur Trägheit (blaue Kurve) wurde hier sowohl die direkte Kollision von Partikeln aufgrund ihrer Trägheit, als auch die Abscheidung durch den Sperreffekt zusammengefasst. Entsprechend steigt der Abscheidegrad mit der Partikelgröße.

Aus der Kombination von Diffusion und Trägheit ergibt sich der reale Abscheidegrad (rote Kurve). Das daraus entstehende Minima beschreibt die MPPS (Most Penetrating Particle Size), sprich die Partikelgröße, die den geringsten Abscheidegrad hat. Der Wert von $0,3 \mu m$ dient hier nur als Richtwert. Sowohl der Verlauf der Kurve, als auch der MPPS-Wert variieren je nach gewählten Parametern [12] [13]. Da die Diffusion in dieser Arbeit jedoch nicht berücksichtigt wird, entspricht der erwartete Verlauf des Abscheidegrads, dem der Trägheit (blau Kurve).

2.2 Molekulardynamik

In der Molekulardynamik geht es um Computersimulationen von komplexen Systemen, welche auf molekularem Level modelliert werden. Ziel der Simulation ist ein besseres Verständnis der stattfindenden Mechanismen zu erlangen. Zu diesen gehören die Einflüsse von kinetischen, elektrostatischen und thermodynamischen Energien auf einzelne Atome und Moleküle, als auch die Wechselwirkungen untereinander. Im folgenden Abschnitt werden die relevanten Einflüsse erläutert, welche in dieser Arbeit von Bedeutung sind.

2.2.1 Lennard-Jones-Potential

Die Wechselwirkung zwischen Partikeln wird über deren Potentiale beschrieben. Hierzu gibt es sowohl kurzreichweitige, als auch langreichweitige Potentiale. Das verwendete Potential in dieser Arbeit wird als Lennard-Jones-(12,6)-Potential bezeichnet.

$$(2.1) \quad V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6\right]$$

Es beschreibt die Wechselwirkung zwischen zwei ungeladenen Partikeln. Der Wert r entspricht dem Abstand der Mittelpunkte der Kugeln, σ ergibt sich aus der Summe der beiden Radien der Partikel. Die Intensität der Wechselwirkung wird durch ϵ definiert. Die Herleitung des Potentials basiert dabei auf der Kombination von abstoßenden und anziehenden Kräften.

$$(2.2) \quad V_{Pauli}(r) = \frac{4\epsilon\sigma^{12}}{r^{12}}$$

$$(2.3) \quad V_{dW}(r) = -\frac{4\epsilon\sigma^6}{r^6}$$

Gleichung (2.2) beschreibt den abstoßenden Term nach dem Pauli-Prinzip und Gleichung (2.3) die Anziehung aufgrund der Van-der-Waals-Kräfte [14].

Gleichung (2.1) beschreibt den Verlauf des Lennard-Jones-Potentials, in Abhängigkeit von dem Abstand der Partikeln, welche in Abbildung 2.4 dargestellt ist. Für große Partikelabstände läuft das Lennard-Jones-Potential asymptotisch gegen null. Es findet somit nahezu keine Wechselwirkung zwischen den Partikeln statt. Das Lennard-Jones-Potential gehört deshalb zu den kurzreichweitigen Potentialen. Wird der Abstand der Partikeln geringer, dann ziehen sie sich gegenseitig an. Die größte Anziehung findet dabei in dem Punkt $r_m = 2^{1/6}\sigma$ statt. Verkürzt sich der Abstand noch weiter, dann wirkt sich der abstoßende Teil größer aus, als der Anziehende. Die Anziehung sinkt bis zum Punkt $r = \sigma$. Sobald sich die Partikel überlappen, stoßen sie sich gegenseitig ab.

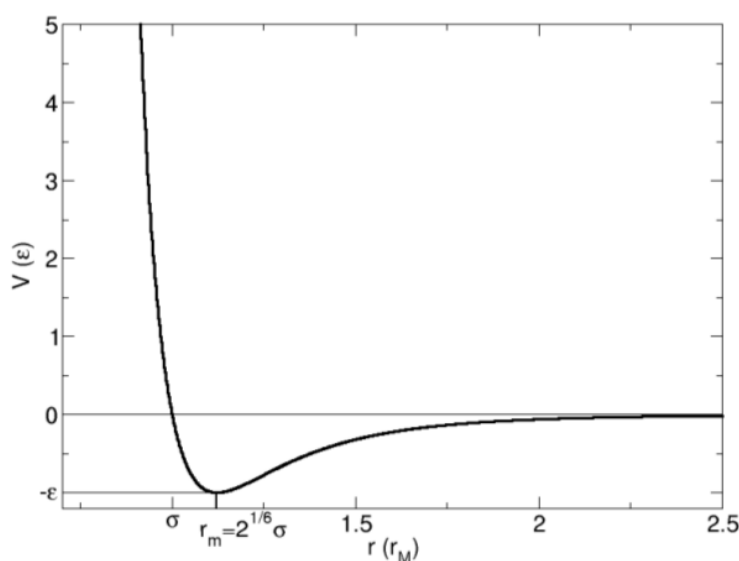


Abbildung 2.4: graphischer Verlauf des Lennard-Jones-Potential [15]

2.2.2 Lattice-Boltzmann-Methode

Die Lattice-Boltzmann-Methode dient zur Simulation der Hintergrundströmung der Partikeln. Die Grundidee liegt darin, dass sich die Gasmoleküle eines Fluids durch winzig kleine Partikeln modellieren lassen. Das Verhalten des Fluids wird durch den kontinuierlichen Transport der Gasmoleküle auf den Gitterknoten simuliert. ESPResSo verwendet das in Abbildung 2.5 dargestellte D3Q19 Modell. Es ist ein dreidimensionales Gitter, bestehend aus 19 Geschwindigkeitsvektoren \vec{c}_i . Alle Vektoren haben ihren Ursprung im zentralen Gitterknoten. Der Vektor \vec{c}_0 entspricht dabei dem Nullvektor ($v(\vec{c}_0) = 0$). \vec{c}_1 bis \vec{c}_6 zeigen zu den 6 nächsten und 12 übernächsten Nachbarknoten. Die Werte der Vektoren sind dabei abhängig von der Dichte und der Geschwindigkeit des Fluids, sowie sonstigen, von außen wirkenden Kräften, wie etwa der Gewichtskraft.

Die Berechnung der Fluidparameter findet in zwei Schritten statt. Im Strömungsschritt verteilen sich die Partikeln neu auf den Knoten, in Abhängigkeit von den jeweiligen Geschwindigkeitsvektoren. Während des Kollisionsschritts werden die Partikeldichten anschließend an den Gitterknoten aktualisiert, indem die Kollision zwischen den Gasmolekülen modelliert wird [16].

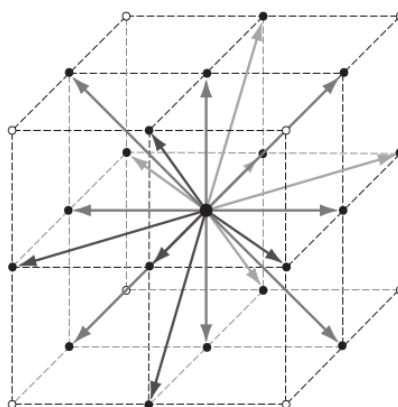


Abbildung 2.5: D3Q19 Model der Lattice-Boltzmann-Methode [16]

2.3 ESPResSo

Zur Simulation wird die Software **ESPResSo** (Extensible Simulation Package for **R**esearch on **S**oft matter) [17] [18] verwendet. Mit ESPResSo lassen sich Simulationen zur Molekulardynamik und Monte Carlo Simulationen durchführen. Sie bietet eine Vielzahl von Möglichkeiten um Systeme von partikulärer Materie zu modellieren. Darunter fällt beispielsweise die Simulation von biologischen Systemen, wie etwa der DNA [19].

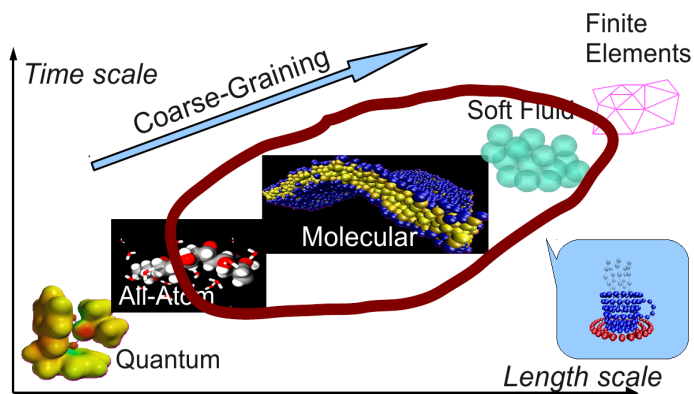


Abbildung 2.6: Coarse-Graining-Modell von ESPResSo [19]

Abbildung 4.12 beschreibt das von ESPResSo verwendete Coarse-Graining-Modell, welches zur Modellierung der Simulation benutzt wird. Im Gegensatz zum All-Atom-Modell, bei dem alle Informationen gespeichert und verwendet werden, wird beim Coarse-Graining-Modell nur ein gewisser Grad an Informationen verarbeitet. Hierbei werden Atome und Moleküle zu einzelnen Partikeln gruppiert und als Ganzes betrachtet. Dies hat den Vorteil, dass deutlich größere, aber dafür vereinfachte Simulationen realisierbar sind [20].

Der Kern von ESPResSo ist in der Programmiersprache C geschrieben, einige Bauteile bauen aber auch auf C++ auf. ESPResSo unterstützt sowohl das Message Passing Interface (MPI), um die Berechnung parallel auf mehreren Prozessoren zu verteilen, als auch die Schnittstelle CUDA für Nvidia Grafikkarten. Gesteuert wird die Software per Tcl(Tool command language)-Skript. Die Tcl-Schnittstelle (Abbildung 2.7) wird von C nativ unterstützt, da sie ursprünglich als Framework für C gedacht war. Es ist jedoch ein weiterer Tcl-Interpreter in ESPResSo enthalten, der nicht Standard-Tcl-Code einließt. Darunter fallen alle Funktionen, die in ESPResSo implementiert, aber durch das Tcl-Skript aufgerufen werden können.

Die Simulation wird durch den Aufruf von ESPResSo, mit dem Skript als zusätzlichen Parameter, gestartet. Das Skript wird anschließend eingelesen und dient zur Steuerung der Simulation. Simulationsergebnisse können während und am Ende der Simulation in externe Dateien geschrieben werden. Diese werden anschließend durch den Benutzer durch eine Visualisierungssoftware dargestellt, oder mit einem Analyseprogramm weiterverarbeitet.

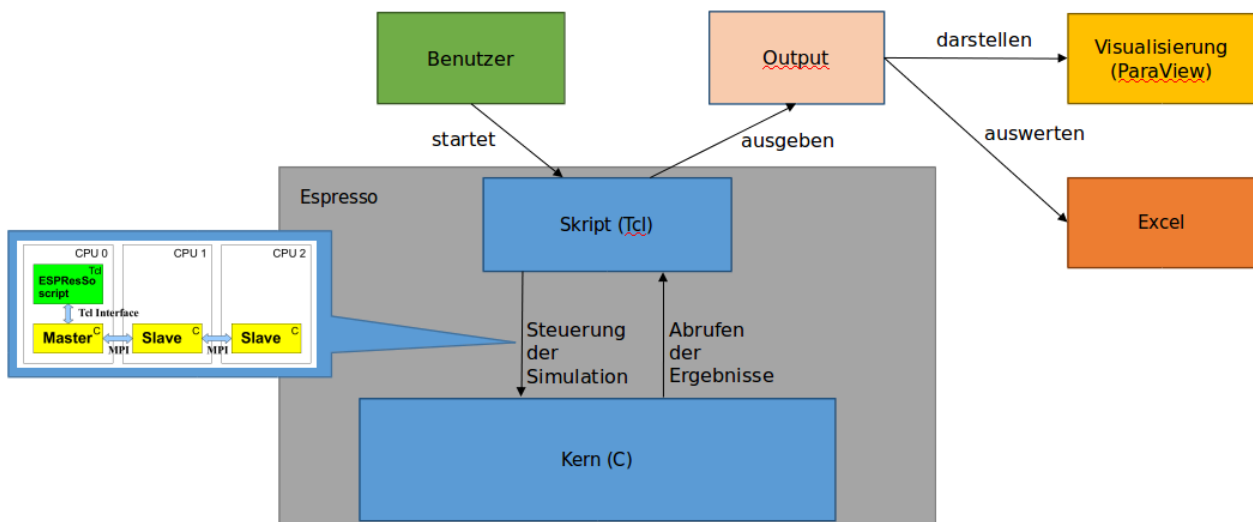


Abbildung 2.7: Aufbau und Architektur[20] von ESPResSo

3 Implementierung

3.1 Funktionsumfang des Skripts zu Beginn der Arbeit

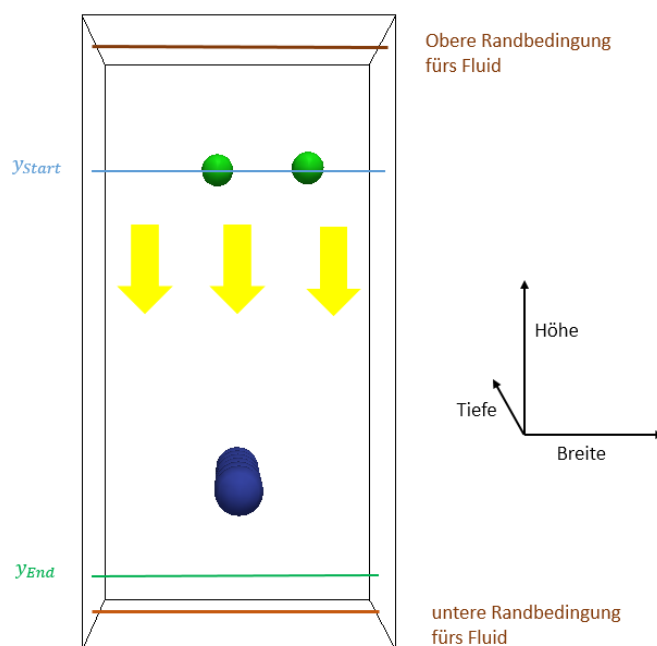


Abbildung 3.1: visueller Aufbau einer Simulation, die mit dem ursprünglichen Skript erstellt wurde

Abbildung 3.1 zeigt den visuellen Aufbau einer Simulation, die mithilfe des ursprünglichen Skripts erstellt wurde. Das Ziel der Simulation war, dass eine ungeladene Faser mit einzelnen, ungeladenen Partikeln beladen wird, um deren Interaktion beobachten zu können. Der schwarze Quader entspricht der Domaingröße, welche vom Benutzer frei definierbar ist. Die Domain ist dabei periodisch. Sollte ein Partikel während der Simulation beispielsweise über den rechten Rand der Domain ($Partikel_{x-coord} > Box_x$) hinausgehen, dann tritt dieses Partikel wieder am linken Rand hinein. Auch das Fluid fließt periodisch. Es wird am oberen und unteren Rand der Domain durch Randbedingungen eingegrenzt und fließt in negativer Y-Richtung (gelbe Pfeile).

Die Partikelzugabe erfolgt nur zu einem Zeitpunkt. Die Anzahl der Partikel ist dabei frei wählbar, Partikeltyp bleibt jedoch derselbe. Der Y-Wert y_{Start} , an dem die Partikel hinzugefügt werden, wird von dem Nutzer bestimmt. Alle Partikel werden entsprechend auf derselben Höhe hinzugefügt. Die X- und Z-Koordinaten werden dabei per Zufallsgenerator bestimmt. Die Ausgangsgeschwindigkeit eines Partikels ist ebenfalls dreidimensional definiert. Die Geschwindigkeit in X- und Z-Richtung beträgt 0. In Y-Richtung wird die Geschwindigkeit des Fluids gewählt.

Um die simulierten Ergebnisse darstellen zu können, werden während der Simulation folgende Informationen zur Visualisierung mitgeschrieben:

1. Position der Faserpartikel
2. Informationen zum Fluid, alle i Integrationsschritte
3. Position der hinzugefügten Partikel, alle i Integrationsschritte

Die Simulation verläuft in drei Phasen. Zu Beginn wird eine gewisse Anzahl von Schritten integriert, damit sich das Fluid einfließen kann. Die genaue Anzahl hängt dabei von der Domaingröße ab. In der zweiten Phase (hier nur ein Zeitpunkt) erfolgt die Partikelzugabe. Die Partikel werden am oberen Rand der Domain hinzugefügt und fließen dann in Richtung des Fluids. Falls ein frei bewegliches Partikel mit einem Faserpartikel kollidiert, dann wird dieses gelöscht und mit einem fixen Partikel ersetzt. Ein fixer Partikel hat dieselbe Größe wie das ursprüngliche Partikel und die Position bleibt nach der Fixierung immer gleich. Sollte ein Partikel den unteren Rand der Domain (y_{Ende}) erreichen, wird es auch hier wieder mit einem fixen Partikel ersetzt. In der dritten Phase werden keine neuen Partikel mehr hinzugefügt. Sie dient als Auslauf für die vorhandenen Partikel. Die Anzahl der Integrationsschritte hängt auch hier von der Domaingröße und von der Partikelgeschwindigkeit ab. Sie sollte hoch genug sein, damit die zuletzt eingefügten Partikeln das untere Ende der Domain erreichen können.

3.2 Erweiterung des Skripts

Abbildung 3.2 zeigt das Ergebnis einer Simulation, welche mit dem erweiterten Skript durchgeführt wurde. Partikel können nun kontinuierlich hinzugefügt werden. Die Kombination aus Partikeltyp, Anzahl und Zeitpunkt ist dabei frei wählbar. Eine genauere Erläuterung der Partikelzugabe wird in Abschnitt 3.4 erläutert. Sollte die Struktur der abgeschiedenen Partikeln den oberen Rand der Domain erreichen, sodass kein Platz zur Platzierung von neuen Partikeln vorhanden ist, dann wird die Simulation vorzeitig beendet.

Die Kollisionsberechnung wurde so angepasst, dass unterschiedliche Partikelgrößen unterstützt werden. Die betreffenden Abstände werden nun automatisch bestimmt, statt vom Nutzer angegeben werden zu müssen. Außerdem wird nun auch die Kollision zwischen frei beweglichen Partikeln und bereits abgesetzten Partikeln berücksichtigt. Die genaue Kollisionsberechnung ist in Abschnitt 3.5 erklärt.

Wird ein Partikel abgeschieden, erhält dieses ein zusätzliches, kugelförmiges Constraint für

das Fluid an derselben Stelle. Dies ist notwendig, da das Constraint dem Fluid die Existenz des Partikels signalisiert. Die im Verlauf einer Simulation entstehende Struktur hätte ansonsten keine Auswirkung auf das Fluid.

Partikel, die den unteren Rand der Domain erreichen, werden nur gelöscht und nicht mit einem fixen Partikel ersetzt. Dies hat den Vorteil, dass die Simulation performanter ist, da fixe Partikel bei der Integration mit Einfluss haben. Wird ein Partikel gelöscht, so bleibt der Eintrag in der internen Partikelliste von ESPResSo weiterhin mit dem Inhalt *n.a.* bestehen. Dies muss berücksichtigt werden, wenn beispielsweise über die Partikeltypen iteriert werden soll, denn *n.a.* wird hier als eigener Typ interpretiert.

Außerdem wurde das Skript mit Routinen zur Auswertung erweitert, welche alle zusätzlichen Information berechnen und in eine Datei schreiben, um diese anschließend für eines der gewünschten Schaubilder zu verwenden. Die Erfassung des Fluids wurde dahingehend eingeschränkt, dass nur noch die beiden Zeitpunkte nach der Aufwärmphase und am Ende der Simulation erfasst werden. Grund hierzu war die Dateigröße. Auf der großen Domain, beträgt die Dateigröße etwas über 100 MB pro erfasstem Zeitschritt. Die entsprechende Ausgabe wurde im Skript nur auskommentiert. Sollte die kontinuierliche Veränderung des Fluids eine wesentliche Rolle spielen, so kann der Benutzer den entsprechenden Code im Skript wieder aktivieren.

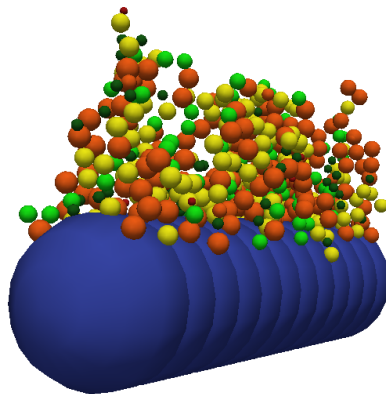


Abbildung 3.2: Darstellung einer Faser, an der unterschiedliche Partikeltypen abgeschieden wurden

3.3 Aufbau des Skripts

Abbildung 3.3 zeigt den sinnlichen Aufbau des Skripts. Ziel der Aufteilung auf zwei verschiedene Dateien ist die bessere Bedienbarkeit für den Benutzer, indem alle Variablen und Constraints, welche eine Simulation charakterisieren, in einer separaten Benutzerdatei übersichtlich ausgelagert sind. Der Tcl-Befehl *source* am Ende der Benutzerdatei bindet dann automatisch die Berechnungsdatei mit ein.

In der Berechnungsdatei befindet sich alle Prozeduren für die Simulation, darunter auch die Kollisionsberechnung. Eine universelle Berechnungsdatei hat außerdem den Vorteil, dass diese einfacher erweitert werden kann, da alle Änderungen automatisch in den verschiedenen Benutzerdateien beinhaltet sind.

Um die Simulation zu starten, muss die Benutzerdatei mit ESPReso aufgerufen werden. Die Berechnungsdatei wird dabei von Tcl als String innerhalb der Benutzerdatei eingelesen. Hilfsfunktion, welche die Werte der Variablen übergeben sind deshalb nicht notwendig, da alle Variablen in der Benutzerdatei auch in der Berechnungsdatei global sind.

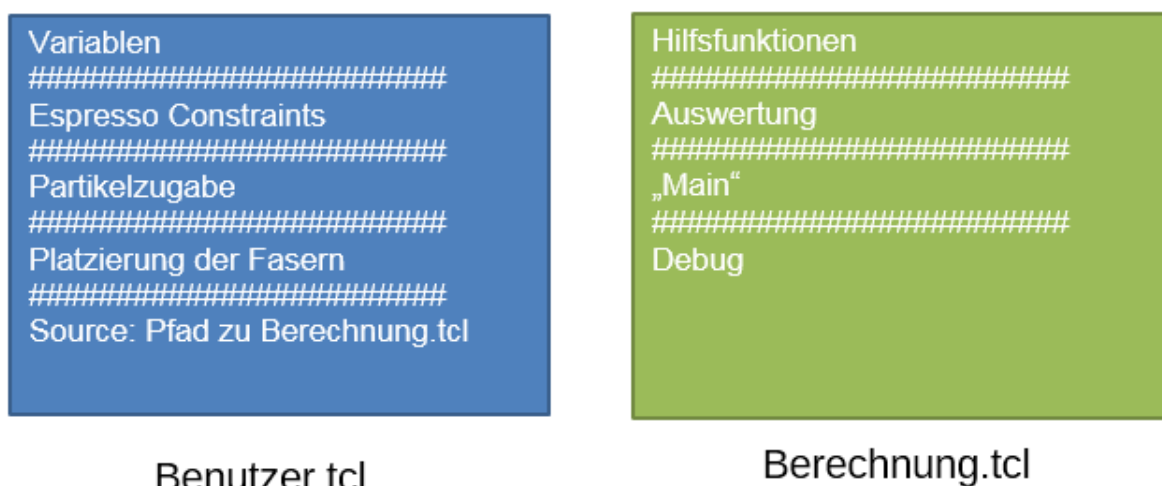


Abbildung 3.3: Darstellung der Aufteilung des Skripts und der strukturelle Aufbau des Inhalts beider Teile

3.4 Bedienung

Platzierung einer Faser

Eine Faser wird in dieser Arbeit immer zylinderförmig definiert. Die Platzierung besteht aus drei Elementen:

- einem Array von Faserpartikeln
- eine zylinderförmige Randbedingung für das Fluid
- ein zylinderförmiges Hilfs-Constraint

Die Faser wird als eine Kette von Faserpartikeln modelliert, wie in Abbildung 3.1 dargestellt (blaue, überlagernde Partikeln). Um eine möglichst vollständige Überdeckung der Faser zu erreichen, sollten die Faserpartikeln dicht übereinander liegen. Die Randbedingung für das

Fluid dient dazu, dem Fluid die Existenz der Faser zu signalisieren. Das zweite Hilfs-Constraint ist optional. Es dient nur der Kontrolle, indem von ESPResSo eine Fehlermeldung geworfen wird, sobald ein frei bewegliches Partikel in den Bereich des Constraints eindringt.

Partikelzugabe

Die Partikelzugabe erfolgt über eine mehrfach verkettete Tcl-Liste. Ein Eintrag definiert dabei eine Zugabe. Dieser besteht aus den Attributen Partikeltyp, Anzahl, Zeitschritt und einem Wahrheitswert zur kontinuierlichen Partikelzugabe. Der Partikeltyp ist ein separates Objekt, welches ebenfalls per Tcl-Liste mit den Werten ID, Masse und Radius des entsprechenden Partikels definiert ist. Die Anzahl und der Zeitschritt bestimmen wie viele Partikel hinzugefügt werden und zu welchem Zeitpunkt während der Simulation dies geschieht. Der Wahrheitswert sagt aus, ob die Zugabe einmalig sein soll, daher nur zu dem genannten Zeitpunkt, oder aber zu jedem Vielfachen des Zeitpunkts stattfinden soll. Die Aufwärmphase für das Fluid wird dabei berücksichtigt.

Der Eintrag $[Typ_0 \ 2 \ 4000 \ 1]$ beschreibt beispielsweise, dass nach der Aufwärmphase immer zwei Partikel vom Typ Typ_0 alle 4000 Zeitschritte hinzugefügt werden.

3.5 Berechnung der Kollision von Partikeln

Algorithmus 3.1 Kollisionsberechnung der Partikel

```
procedure BERECHNUNG
  for all  $part \in Partikel$  do
    if  $part \in Faserpartikel \vee p \in abgesetztes\ Partikel$  then
      Umgebung  $\leftarrow$  BERECHNE UMGEBUNG( $part, r_{Umgebung}$ )
      Umgebung  $\leftarrow$  KORRIGIERE UMGEBUNG( $part, Umgebung$ )
      FIXIERE PARTIKEL( $Umgebung$ )
    end if
  end for
end procedure
```

Algorithmus 3.1 zeigt den Pseudocode der Berechnung. Nachdem die ersten Partikel hinzugefügt wurden, wird zu jedem Integrationsschritt die Umgebung jedes Faser- und jedes abgeschiedenen Partikels berechnet. Berechnet wird die Umgebung mittels des ESPResSo-Befehls *analyze nbhood*. Der gewählte Radius für die Umgebung $r_{Umgebung}$ ergibt sich aus der

3 Implementierung

Summe der Radien des aktuellen Partikels $r_{current}$ und dem Radius des größten Partikeltyps r_{max} in der Simulation, der kein Faserpartikel ist.

$$r_{Umgebung} = r_{current} + r_{max}$$

Falls das aktuelle Partikel ein Faserpartikel ist, dann wird der gewählte Radius mittels dem Satz des Pythagoras berechnet. Dieser ist in Abbildung 3.5 graphisch dargestellt und hängt zusätzlich von dem Abstand der Faserpartikeln d_{Faser} ab.

$$r_{Umgebung} = \sqrt{\left(\frac{d_{Faser}}{2}\right)^2 + (r_{current} + r_{max})^2}$$

Es wird dabei vorausgesetzt, dass alle Faserpartikel innerhalb einer Faser den selben Abstand zum Nachbarn haben. Falls unterschiedliche Abstände gewünscht sind, müsste das Skript so erweitert werden, dass d_{Faser} für jedes Faserpartikel einzeln berechnet wird.

Durch die Berechnung der Hypotenuse aus dem Abstand der Faserpartikeln und der ursprünglichen Summe aus $r_{current}$ und r_{max} , soll sichergestellt werden, dass Partikel nicht in den Bereich zwischen zwei benachbarten Faserpartikel gelangen können, da es sonst zu einer Verletzung des Hilfs-Constraint führen würde. Idealerweise werden die Faserpartikel so dicht zu einander platziert, dass auch für die Umgebung der Faser gilt:

$$r_{Umgebung} \approx r_{Faser} + r_{max}$$

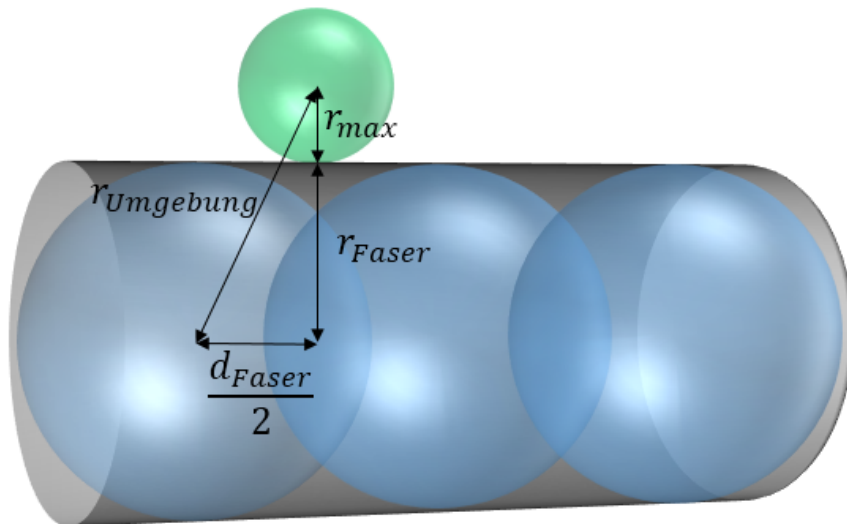


Abbildung 3.4: Graphische Darstellung zur Bestimmung des Radius für ein Faserpartikel

Korrektur der Umgebung

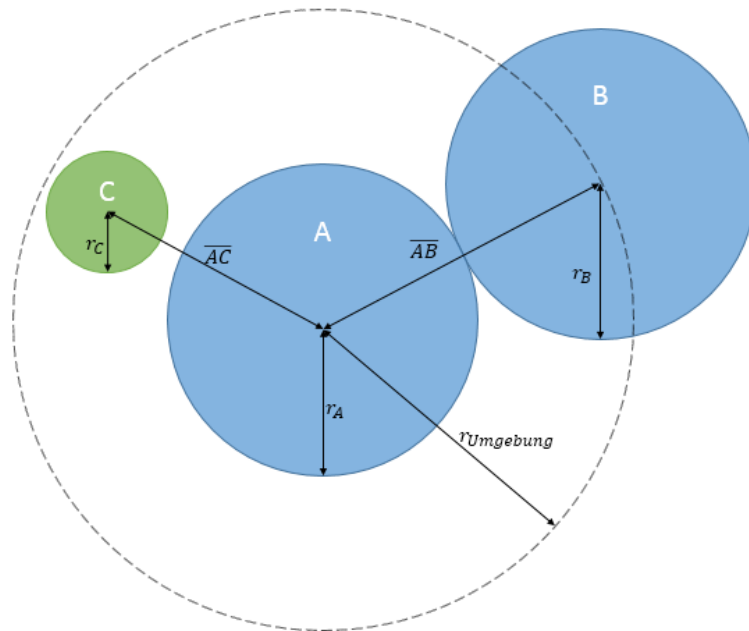


Abbildung 3.5: Veranschaulichung der Problematik bei der berechneten Umgebung von ES-PResSo

Abbildung 3.4 zeigt eine mögliche Situation bei der Kollisionsberechnung, wenn der in Abschnitt 3.5 verwendete Wert $radius_{Umgebung}$ benutzt wird. In dem Beispiel sollen die Partikel A und B demselben Partikeltyp entsprechen. Partikel A ist jedoch bereits abgeschieden worden, Partikel B ist noch frei beweglich. Partikel C ist ebenfalls noch frei beweglich und soll einen zweiten Partikeltyp symbolisieren, dessen Radius kleiner ist, als von den Partikeln A und B. An diesem Beispiel ist zu sehen, dass die Korrektur der Umgebung notwendig ist, da ansonsten Partikel abgeschieden werden, obwohl es zu keiner Kollision untereinander kommt. In diesem Fall wird die Umgebung von Partikel A bestimmt. Partikel C liegt auch darin, da der Abstand von Partikel A und C (\overline{AC}) geringer ist, als der verwendete Radius $r_{Umgebung}$. Der Abstand \overline{AC} ist jedoch größer als die Summe der Radien r_A und r_C der Partikel A und C. Folglich muss Partikel C aus der Umgebung von Partikel A gelöscht werden. Partikel B wird in diesem Beispiel abgeschieden, da der Abstand \overline{AB} der Partikel A und B, der Summe aus den Radien r_A und r_B entspricht.

Algorithmus 3.2 beschreibt die Korrektur der berechneten Umgebung. Zusätzlich zu dem Kriterium, dass der Abstand zwischen zwei Partikel nicht größer sein darf als die Summe derer Radien, werden zusätzlich alle Faser- und bereits abgeschiedene Partikeln ebenfalls aus der Umgebung gelöscht, darunter auch das Partikel selbst, von dem die Umgebung berechnet werden soll.

Algorithmus 3.2 Korrektur der Umgebung

```
procedure KORRIGIERE UMGEBUNG( $\text{part}_{\text{current}}$ , Umgebung)
  for all  $p \in \text{Umgebung}$  do
    Distanz $_{p, \text{part}_{\text{current}}}$   $\leftarrow$  BERECHNE DISTANZ( $p$ ,  $\text{part}_{\text{current}}$ )
    if Distanz $_{p, \text{part}_{\text{current}}} > (\text{radius}_p + \text{radius}_{\text{current}}) \vee p_{\text{Typ}} \in (\text{Faser} \cup \text{fixiert})$  then
      Lösche  $p$  aus Umgebung
    end if
  end for
  return Umgebung
end procedure
```

4 Simulationsergebnisse

Im folgenden Kapitel werden Simulationsergebnisse vorgestellt. Die dafür verwendeten Parameter können aus Tabelle 4.2 bis Tabelle 4.5 entnommen werden. Sofern Abweichungen von diesen Werten bestehen, sind sie zu dem entsprechenden Schaubild angemerkt.

Einheiten sind in ESPResSo nicht fest definiert, sondern können, je nach benötigter Größenordnung selbst bestimmt werden.

| reduzierte Einheit | Wert | Einheit |
|--------------------|----------|---------|
| [x] | 1.00E-05 | m |
| [m] | 1.76E-10 | kg |
| [E] | 1.76E-08 | J |
| [t] | 1.00E-06 | s |
| [q] | 1.60E-19 | As |

Tabelle 4.1: verwendete Reduzierung der Einheiten

| Beschreibung | Parametername | Wert | reduzierte Einheit |
|---------------------|----------------|------|--------------------|
| Domainhöhe | box_y | 15 | [x] |
| Domainbreite | box_x | 6 | [x] |
| Domaintiefe | box_z | 4 | [x] |
| Zeitschritt | time_step | 0.01 | [t] |
| Partikel- Starthöhe | position_start | 13 | [x] |
| Partikel- Endhöhe | position_end | 0.5 | [x] |

Tabelle 4.2: allgemeine Parameter

Die Maße der Domain, sowie die Start- und Endhöhe der Partikeln sind in Abbildung 3.1 abgebildet. *position_start* und *position_end* entsprechen hier y_{Start} und y_{End} .

| Beschreibung | Parametername | Wert | reduzierte Einheit |
|---------------------------|---------------|-----------|-----------------------|
| Geschwindigkeit | v_part | -0.0381 | [x]/[s] |
| (kinematische) Viskosität | visc | 0.152 | [x] ² /[t] |
| Dichte | dense | 0.0000068 | [m]/[x] ³ |
| Reibung | friction | 0.000007 | - |
| Gitterkonstante | agrid | 0.1 | [x] |
| LB-Zeitschritt | tau | 0.01 | [t] |

Tabelle 4.3: Parameter für das Fluid

| Typ | Radius[x] | Masse[m] |
|-----|-----------|-----------|
| 1 | 0.04665 | 6.403E-06 |
| 2 | 0.07185 | 2.339E-05 |
| 3 | 0.0958 | 5.545E-05 |
| 4 | 0.1106 | 8.533E-05 |
| 5 | 0.12775 | 1.315E-4 |
| 6 | 0.1475 | 2.024E-4 |

Tabelle 4.4: Partikeltypen

| Beschreibung | Parameter | Wert | reduzierte Einheit |
|---------------------------|--------------|------|--------------------|
| X-Position | posx | 4 | [x] |
| Y-Position | posy | 3 | [x] |
| Radius | fiberRadius | 1 | [x] |
| Abstand der Faserpartikel | dist_circles | 0.67 | [x] |
| Anzahl Faserpartikel | n_circles | 6 | - |

Tabelle 4.5: Position und Eigenschaften der Faser

Die ersten beiden Variablen in Tabelle 4.5 beschreiben die Position der Faser in der Domain. Der Radius der Faser wird durch die Variable *fiberRadius* definiert. Der Abstand der Faserpartikeln bestimmt, wie weit die Mittelpunkte zweier Faserpartikel entfernt sind. Die Anzahl der Faserpartikeln ergibt sich auch der Tiefe der Domain und dem Abstand der Faserpartikeln. Radius und Abstand der Faserpartikeln sind zusätzlich in Abbildung 4.1 dargestellt.

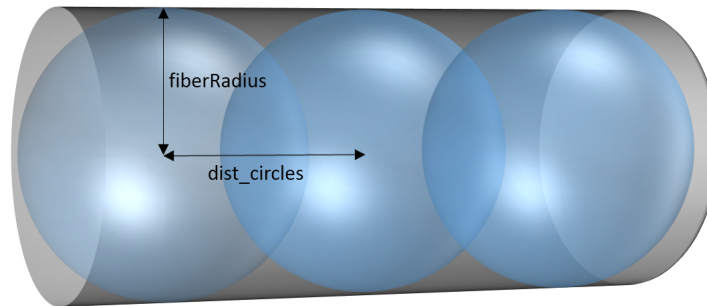


Abbildung 4.1: Darstellung einer zylinderförmigen Faser durch Faserpartikel

4.1 Absoluter Abscheidegrad

Der Abscheidegrad beschreibt die Anzahl der abgeschiedenen Partikeln im Verhältnis zur Gesamtmenge. Dieser hängt von mehreren Faktoren ab, wie beispielsweise der Trägheit eines Partikels oder der Struktur der Faser. Die Trägheit eines Partikels lässt sich durch die Stokes-Zahl (St) beschreiben. Sie sagt aus wie schnell ein Partikel auf eine Strömungsänderung reagiert und ist von dem Durchmesser des Partikels P_D , dem Faserdurchmesser F_D , der Dichte ρ_{Fluid} , der Strömungsgeschwindigkeit v_{Fluid} und von der kinematischen Viskosität ν_{Fluid} des Fluids abhängig.

$$(4.1) \quad St = \frac{\rho_{Fluid} * P_D^2 * v_{Fluid}}{18 * \nu_{Fluid} * F_D}$$

In Abbildung 4.2 sind die Abscheidegrade von unterschiedlichen Partikeltypen zu sehen. In den Simulationen wurde die Partikelgröße und somit, bei gleicher Partikeldichte, die Masse der Partikeln variiert. Es wurden hierbei fünf Simulationen durchgeführt, bei denen alle n -Zeitschritte eine unterschiedliche Anzahl an Partikeln hinzugefügt wurde, sodass die Gesamtzahl variiert. Die Anzahl pro Partikeltyp blieb jeweils gleich. Aufgrund der zufälligen Partikelplatzierung schwanken vor allem die Werte mit einer geringen Anzahl an Partikeln. Wenn diese steigt, wird der Verlauf der Kurve glatter. Dennoch lässt sich der Trend erkennen, dass der Abscheidegrad für alle Simulationen mit der Stokes-Zahl steigt.

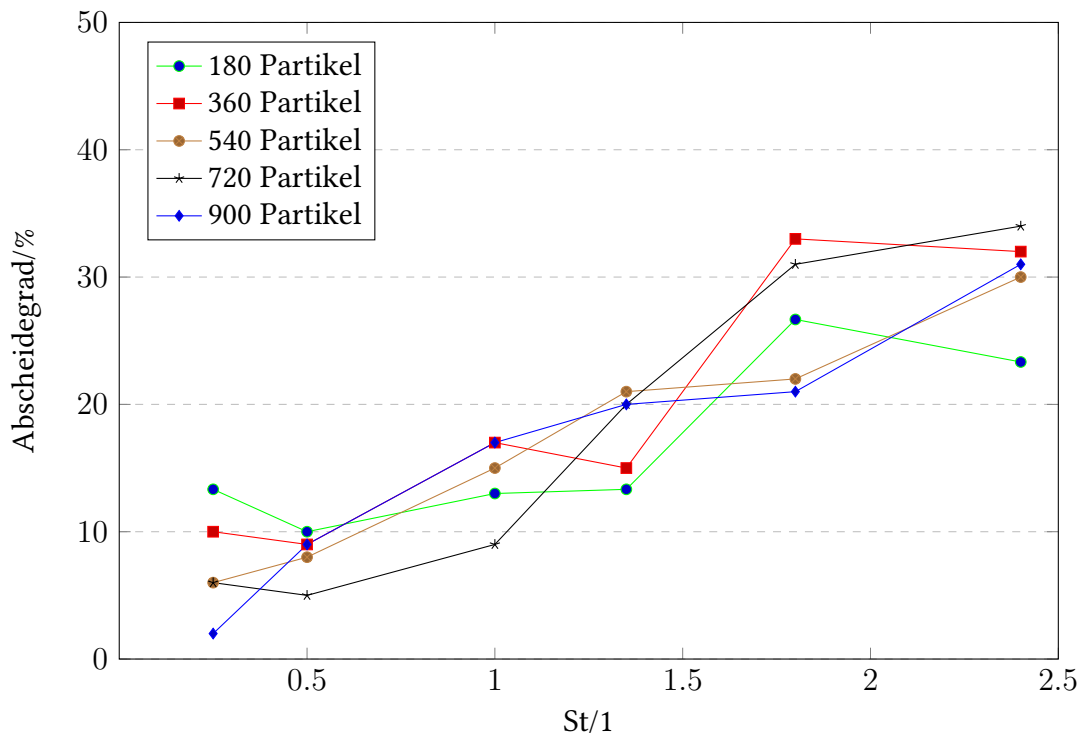


Abbildung 4.2: Abscheidegrade für unterschiedliche Stokes-Zahlen, bei unterschiedlicher Partikelanzahl

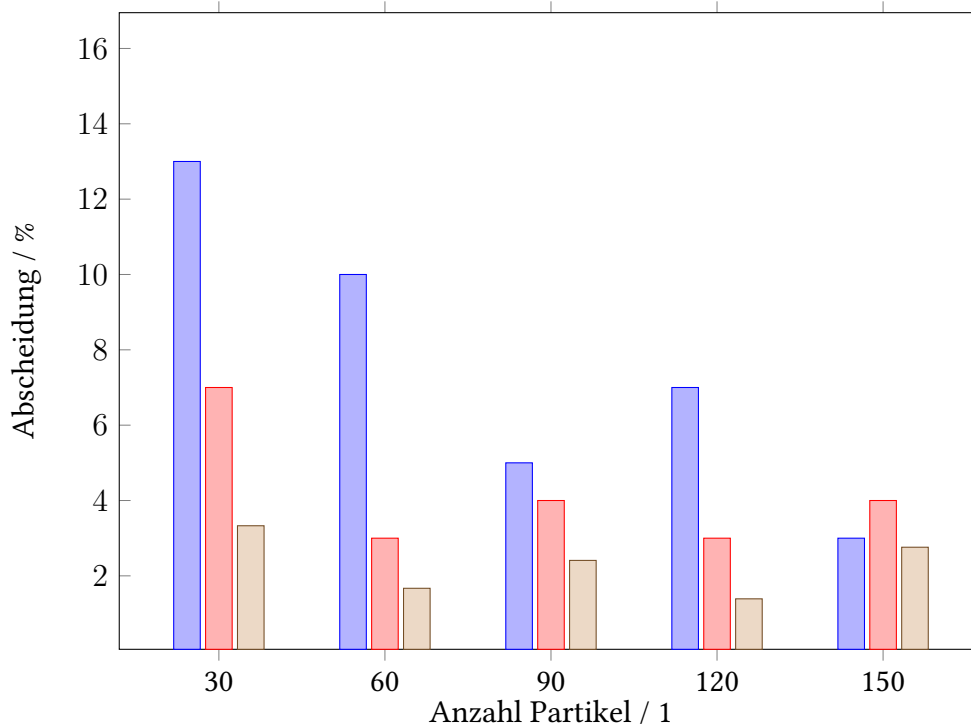
Nun soll die Wechselwirkung zwischen den einzelnen Partikeltypen genau untersucht werden. Hierfür werden der kleinste und größte Partikeltyp genauer betrachtet.

In Abbildung 4.3 ist der Abscheidegrad des kleinsten Partikeltyps mit dem Radius $0,04665[x]$ dargestellt. Die sich daraus ergebende Stokes-Zahl entspricht dem Wert $0,25$. Jedes Balken-Tripel steht für drei verschiedene Simulationen. Der blaue Balken entspricht den Werten aus Abbildung 4.2. Bei dem jeweils roten Balken wurde nur der entsprechende Typ in der angegebenen Anzahl hinzugefügt. Dies bedeutet, dass die Gesamtzahl der zugegebenen Partikel nur noch ein sechstel der ursprünglichen Simulation beträgt. Hieraus soll der direkte Einfluss durch andere Partikeln sichtbar werden. In der dritten Messreihe (beige) wurde ebenfalls nur der entsprechende Partikeltyp hinzugefügt. Dieser wurde diesmal mit sechs multipliziert, sodass die Gesamtzahl der Partikeln wieder den ursprünglichen Wert erreicht. Dies soll den Einfluss durch andere Partikeltypen veranschaulichen.

Anhand der Betrachtung des ursprünglichen Abscheidegrads mit einem der beiden anderen Werten innerhalb eines Tripels wird deutlich, dass der Abscheidegrad von Partikeln mit einer geringen Trägheit stark durch größere Partikeln beeinflusst wird. Die beiden Werte fallen dementsprechend deutlich geringer aus, da in diesen Simulationen keine größeren Partikeln zugegeben wurden, die den Abscheidegrad hätten beeinflussen können.

Dies liegt daran, dass Partikeln mit einer geringen Trägheit entlang der Strömung an der Faser vorbei fließen. Trifft dieses Partikel jedoch auf abgeschiedene Partikeln, ist die Wahrscheinlich-

keit höher, dass es an ihnen hängen bleibt. Grund hierfür ist die Strömung, die nicht komplett außen herum, sondern auch durch Hohlräume hindurch fließt. Ein weiterer Grund könnte an der Modellierung liegen. Das Fluid registriert Partikel erst, wenn diese abgeschieden werden, indem ein Constraint für das Fluid an dieser Stelle gesetzt wird. Abhängig von der Dauer, bis sich die Strömung an dieser Stelle angepasst hat, fließen die, sich unmittelbar oberhalb befindlichen Partikeln in falscher Richtung.



■ Werte aus Abbildung 4.2 ■ Einfluss durch andere Partikel ■ Einfluss durch andere Partikeltypen

Abbildung 4.3: Abscheidegrade des kleinsten Partikeltyps für unterschiedliche Modellierungen

Nach der Betrachtung des kleinsten Partikeltyps, werden nun die Abscheidegrade des größte Typs mit dem Radius $0,1475[x]$ analysiert. Diese sind in Abbildung 4.4 dargestellt. Die Stokes-Zahl entspricht diesmal dem Wert 2,4.

Im Gegensatz zum vorherigen Partikeltyp sind hier keine aufschlussreichen Unterschiede zu erkennen. Dies liegt daran, dass kleine Partikel wenig Einfluss auf größere Partikel haben. Der Größenunterschied der Balken fällt entsprechend gering aus. Erwartungsgemäß liefert hier die dritte Messreihe die größten Abscheidegrade, da der durchschnittliche Partikeldurchmesser gestiegen ist und somit auch das Volumen der abgeschiedenen Partikel schneller ansteigt.

4 Simulationsergebnisse

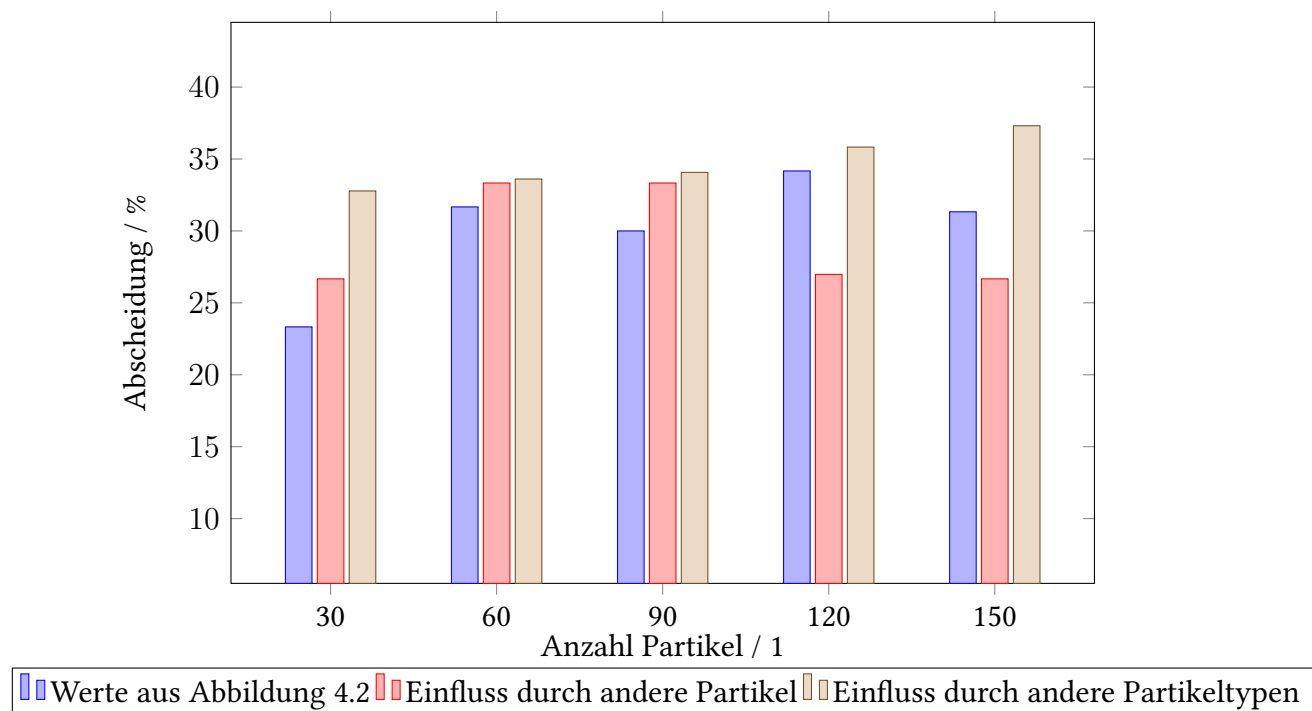


Abbildung 4.4: Abscheidegrade des größten Partikeltyps für unterschiedliche Modellierungen

Nachdem nun erste Erkenntnisse über die Wechselwirkungen zwischen den Partikeln gewonnen werden konnten, werden nun weitere Faktoren untersucht, welche einen Einfluss auf den Abscheidegrad haben können.

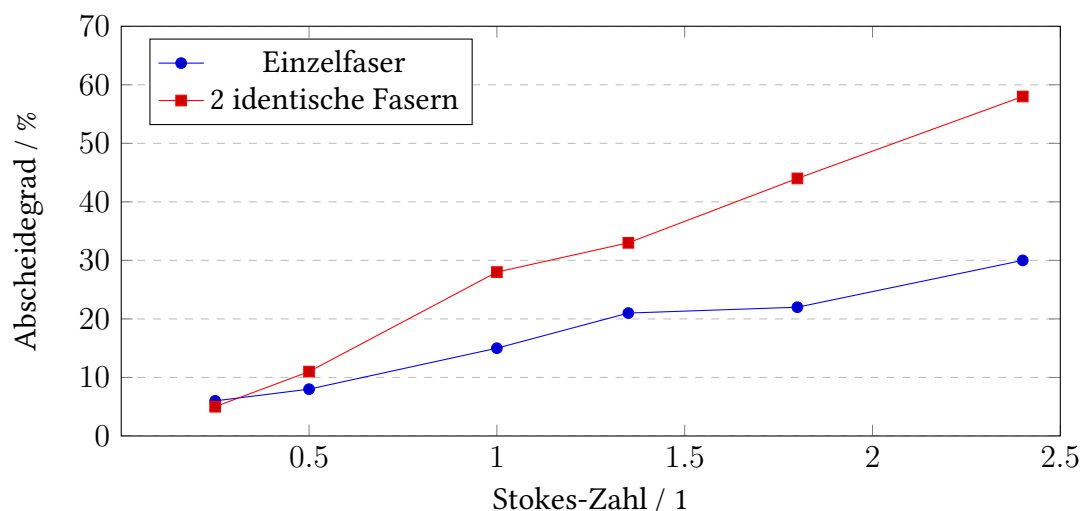


Abbildung 4.5: Abscheidegrad von Partikeln bei unterschiedlichen Faserstrukturen

In Abbildung 4.5 ist der Abscheidegrad von unterschiedlichen Partikeltypen zusehen, welche sich aus den gewählten Faserstrukturen ergibt. Die blaue Kurve entspricht der Simulation mit 540 Partikeln aus Abbildung 4.2. Anstatt eine Faser mittig zu platzieren, wurde nun eine zweite Faser, parallel zur ersten Faser, hinzugefügt. Diese ist höhenversetzt und liegt neben der Ersten. Eine ähnliche Struktur ist in Abbildung 4.9 zu sehen. Zu beobachten ist, dass die Filtereffizienz nicht im Allgemeinen höher ist, wie vermutet werden könnte. Der gewählte Strukturaufbau ist nur für größere Partikeln effizienter. Dies liegt daran, dass diese Struktur keinerlei zusätzlichen Einflüsse für weniger Träge Partikel mit sich bringt. Es ändert sich lediglich die Auftrefffläche (doppelter Faserdurchmesser), von der nur trägere Partikeln profitieren.

Nachdem der Einfluss einer anderen Faserstruktur betrachtet wurde, wir nun die Änderung der Fluidgeschwindigkeit analysiert.

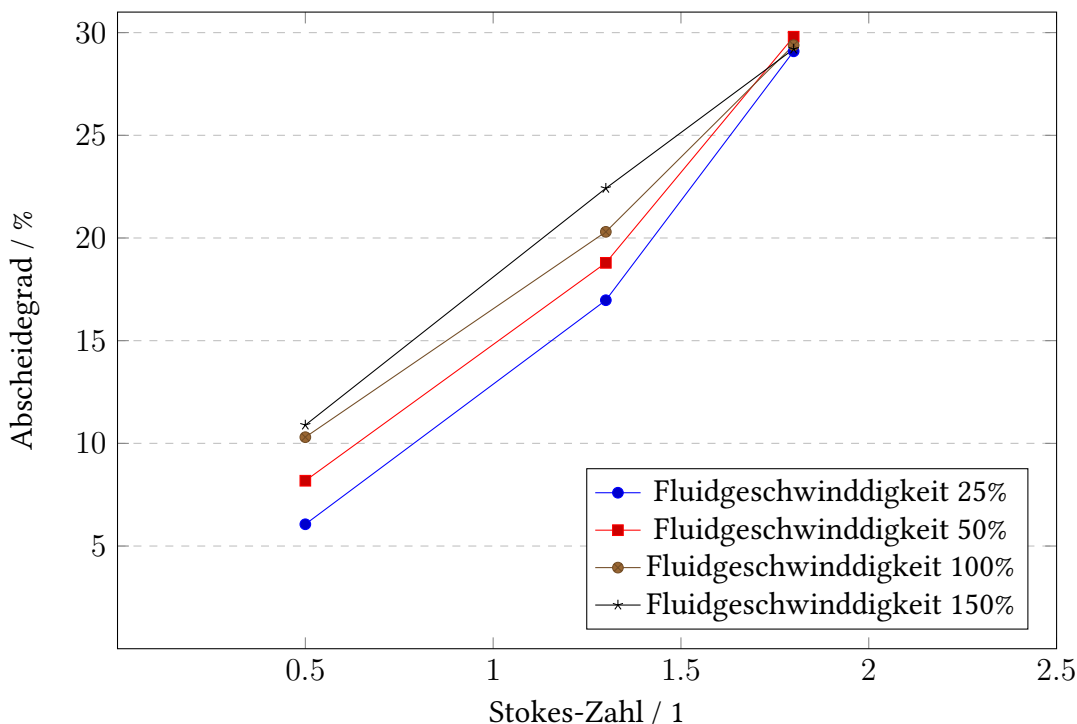


Abbildung 4.6: Abscheidegrad für unterschiedliche Stokes-Zahlen, abhängig von unterschiedlichen Fluidgeschwindigkeiten

Abbildung 4.6 stellt den Abscheidegrad für unterschiedliche Stokes-Zahlen dar, in Abhängig von unterschiedlichen Fluidgeschwindigkeiten. Anders als in den Simulationen zuvor, wurde hier ein fester Partikelradius für alle Typen festgelegt, anstelle derselben Dichte. Der Radius beträgt $0,1[x]$. Die Massen wurden entsprechend angepasst, um dieselben Stokes-Werte zu erlangen. Außerdem wurden nur drei statt sechs Partikeltypen verwendet, sodass die Anzahl der Partikel pro Typ erhöht werden kann, ohne dass die Gesamtlaufzeit der Simulation steigt. Die höhere Anzahl pro Typ soll hierbei eine höhere Genauigkeit bei der Auswertung erbringen. Es ist zu erkennen, dass eine geringere Geschwindigkeit des Fluids den Abscheidegrad von weniger trägen Partikeln reduziert. Wie bereits erklärt wurde, beschreibt die Stokes-Zahl,

4 Simulationsergebnisse

wie schnell ein Partikel auf eine Strömungsänderung reagiert. Folglich hat ein Partikel bei einer geringeren Strömungsgeschwindigkeit mehr Zeit, um auf eine Änderung zu reagieren. Entsprechend ist es genau umgekehrt für eine höhere Geschwindigkeit.

Lediglich für den Partikeltyp mit der Stokes-Zahl 1,8 hatte die Variation der Fluidgeschwindigkeit keinen Einfluss auf den Abscheidegrad. Grund hierfür war die hohe Trägheit des Partikeltyps. Diese war zu hoch, als dass die Änderung der Fluidgeschwindigkeit einen bemerkbaren Einfluss hätte auswirken können.

4.2 zeitliche Entwicklung des Abscheidegrades

Im folgenden Abschnitt wird auf die zeitliche Effizienz der Filtration eingegangen. Um anschaulichere Ergebnisse zu erlangen, wurden die gewählten Parameter der Simulation grundlegend geändert. Grund hierfür war die hohe Rechenzeit, die für Simulationen mit einer großen Anzahl an Partikeln benötigt wird. Selbst bei Simulationen mit knapp 6000 Partikeln konnte das Volumen der angesammelten Partikeln keinen konstanten Einfluss auf den zeitlichen Abscheidegrad ausüben. Größten Einfluss hatte hier weiterhin die zufällige Partikelplatzierung. Die geänderten Werte können aus Tabelle 4.6 entnommen werden.

Im Allgemeinen sollte noch angemerkt werden, dass zu den folgenden Abbildungen eine zeitliche Verschiebung auf der X-Achse mitberechnet werden muss, die hier nicht berücksichtigt ist. Werden n Partikeln zu einem Zeitpunkt i hinzugefügt, so ist der Abscheidegrad von den n Partikeln in i dargestellt. Die tatsächliche Abscheidung erfolgt jedoch erst zu einem späteren Zeitpunkt, dann wenn ein Partikel abgeschieden wird, oder es den unteren Rand der Domain erreicht hat.

Zudem wurden für diese Simulationsreihe zwei Partikeltypen mit gleichem Radius und unterschiedlichen Massen verwendet. Da beide Typen dieselben Abscheidegrade (sowohl total, als auch zeitlich) aufweisen, wird auf die Darstellung der Ergebnisse, der einzelnen Typen nicht genauer eingegangen.

| Parametername | Wert |
|----------------|-----------|
| Allgemein | |
| box_x | 2.5 |
| box_y | 18 |
| box_z | 0.7 |
| Fluid | |
| v_part | -9.16E-03 |
| dens | 8.85E-06 |
| visc | 0.133 |
| friction | 1.962E-06 |
| Partikel | |
| Radius | 0.075 |
| $Masse_{Typ1}$ | 6,699E-05 |
| $Masse_{Typ2}$ | 9,571E-06 |
| Faser | |
| fiberRadius | 0.3 |
| posx | 1.25 |
| posy | 1.3 |
| dist_circles | 0.1 |

Tabelle 4.6: geänderte Parameter zur Simulation der Effizienz

4 Simulationsergebnisse

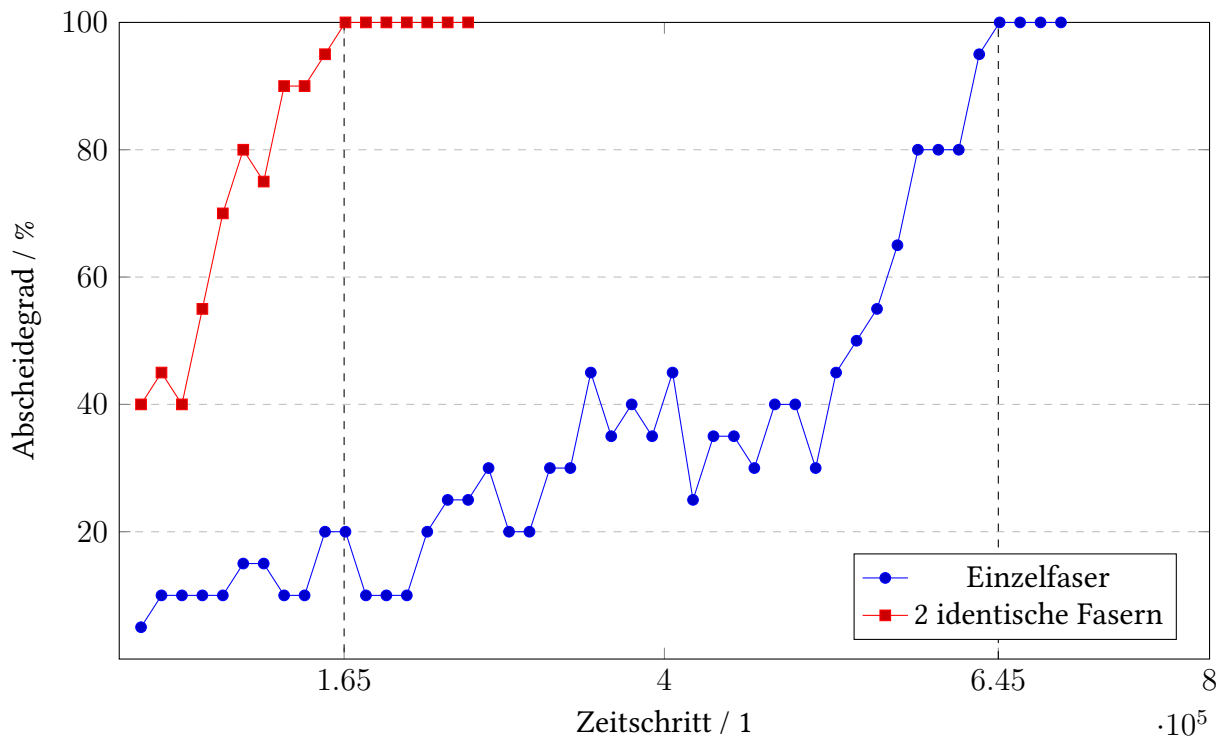


Abbildung 4.7: Verlauf der Effizienz unterschiedlicher Faserstrukturen, in Abhängigkeit von der zurückgelegten Simulationszeit

In Abbildung 4.7 ist der zeitliche Abscheidegrad in Abhängigkeit der Faserstruktur dargestellt. Die Struktur ist ähnlich, wie in der Simulation zu Abbildung 4.9. Es wurden hierbei alle n Zeitschritte die gleiche Anzahl an Partikeln hinzugefügt.

Zu sehen ist, dass die Abscheidung an zwei Fasern höher ist und auch schneller steigt, als an einer einzelnen Faser. In diesem Fall liegt das daran, dass die gewählten Partikeltypen sehr träge sind. Entscheidend ist hier wieder die verdoppelte Auftrefffläche durch eine zweite Faser. Auch die Effizienz steigt mit der Zeit schneller an, da das Volumen der zwei lokalen Strukturen schneller steigt.

In der Simulation mit einer Faser wurde zu dem Zeitpunkt $i = 6.45E10^5$ eine Effizienz von 100% erreicht. Dies bedeutet, dass alle zugegebenen Partikeln zu diesem Zeitpunkt abgeschieden wurden. Unter Umständen bedeutet dies jedoch nicht, dass alle weiteren Partikeln, welche zu einem späteren Zeitpunkt hinzugefügt wurden, auch zu 100% abgeschieden werden. Dies hängt vor allem von der zugegeben Anzahl zu einem Zeitpunkt ab. Im Extremfall von nur einem Partikel ergibt sich eine Effizienz von 0% oder 100%.

Im Gegensatz zur Struktur mit zwei Fasern, die eine Effizienz von 100% schon zum Zeitpunkt $i = 1.65E10^5$ erreicht, braucht die Einzelfaser in dieser Modellierung fast vier Mal so lange, um dasselbe Niveau zu erreichen.

Wie schon bei der Berechnung des absoluten Abscheidegrads für verschiedene Stokes- Zahlen, soll auch hier wieder der Einfluss einer anderen Fluidgeschwindigkeit veranschaulicht werden.

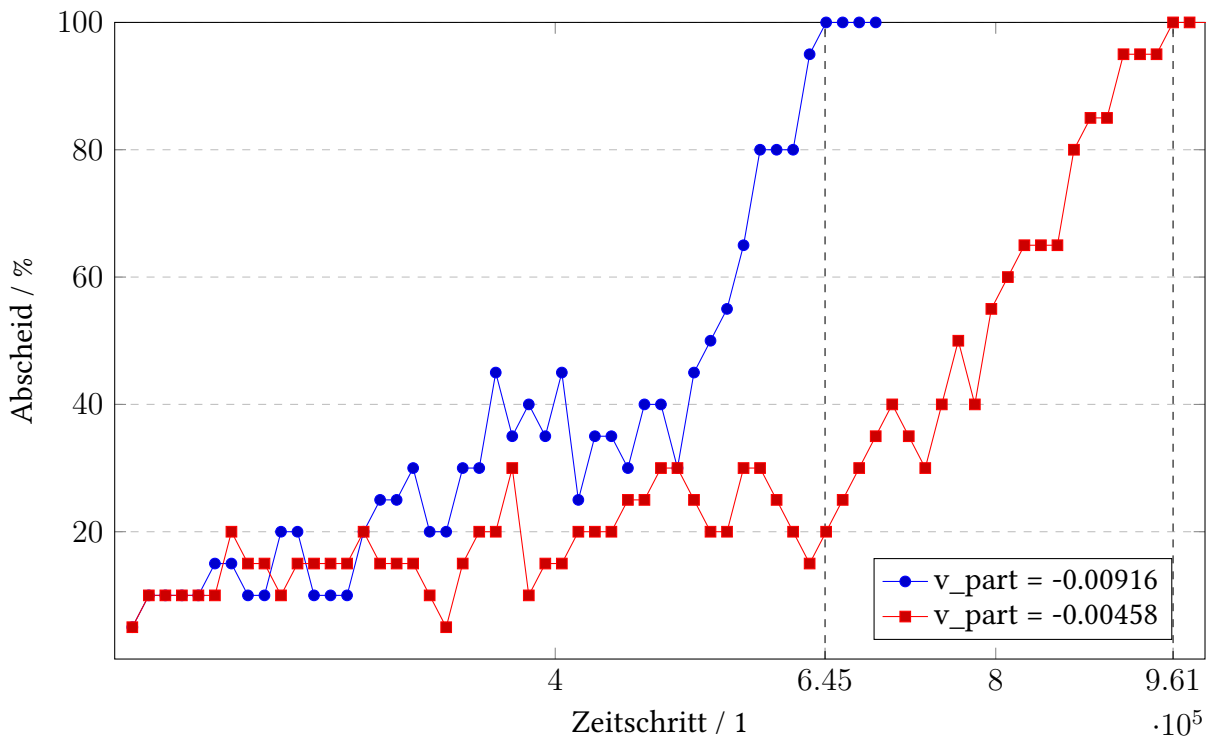


Abbildung 4.8: Effizienz der Einzelfaser für unterschiedlichen Fluidgeschwindigkeiten, in Abhängigkeit von der zurückgelegten Simulationszeit

In diesem Fall wurde die Geschwindigkeit um 50% reduziert. Das Ergebnis ist in Abbildung 4.8 zu sehen.

Zu Beginn der Partikelzugabe liegen die Abscheidegrade beider Simulationen in demselben Bereich. Im Verlauf der Simulation steigt dann der Abscheidegrad bei einer höheren Fluidgeschwindigkeit schneller an. Es hat in dieser Modellierung knapp 50% mehr Integrationsschritte gebraucht, bis bei der Simulation mit der geringeren Fluidgeschwindigkeit eine Effizienz von 100% erreicht wurde.

Die Fluidgeschwindigkeit hat einen bemerkbaren, wenn auch geringen Einfluss auf den Abscheidegrad. Die Werte zu Beginn der Partikelzugabe können durch die zufällige Platzierung der Partikeln verursacht werden. Dies soll bedeuten, dass der gewählte Partikeltyp träge genug ist (selbst bei einer langsameren Fluidgeschwindigkeit), um an der Faser abgeschieden zu werden, wenn ein Partikel zentral über der Faser platziert wird. Gleichzeitig ist das Partikel selbst bei der höheren Fluidgeschwindigkeit nicht träge genug, um an der Faser abgeschieden zu werden, wenn die Platzierung des Partikel zu weit außen über der Faser geschieht.

4.3 Strukturaufbau der abgeschiedenen Partikel und dessen Dichte

Aus den vorhergehenden Simulation aus Abschnitt 4.2 soll nun die finale Struktur der abgeschiedenen Partikeln analysiert werden. Hierbei wird der Aufbau der abgeschiedenen Partikeln am Ende der Simulation betrachtet.

In Abbildung 4.9 ist die finale Struktur der abgeschiedenen Partikeln mit zwei Fasern zu sehen. Aufgrund der nahen Platzierung beider Fasern, konnten sich die lokalen Strukturen frühzeitig zu einer „Brücke“ zusammenschließen. Die Auftrefffläche hat sich dadurch deutlich vergrößert. Entsprechend liegen die abgeschiedenen Partikeln dicht zusammen. Auch die Höhe der finalen Struktur fällt gering aus, da die weitere Partikelzugabe frühzeitig abgebrochen wurde. Als Abbruchkriterium wurde in den Simulationen eine 100 prozentige Effizienz der letzten drei Partikelzugaben gewählt.

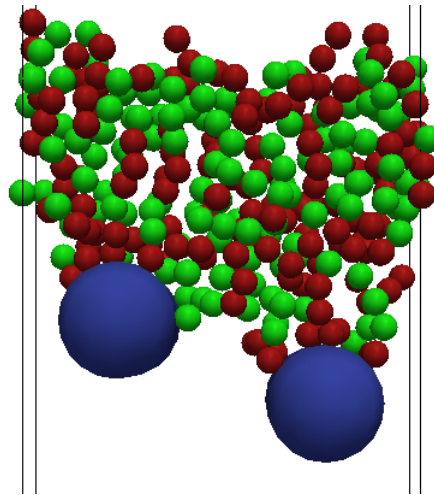


Abbildung 4.9: Struktur-Aufbau mit zwei Fasern

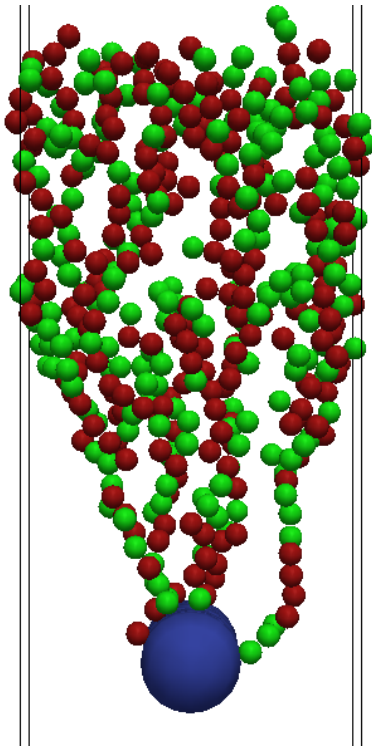


Abbildung 4.10: Struktur-Aufbau einer Einzelfaser

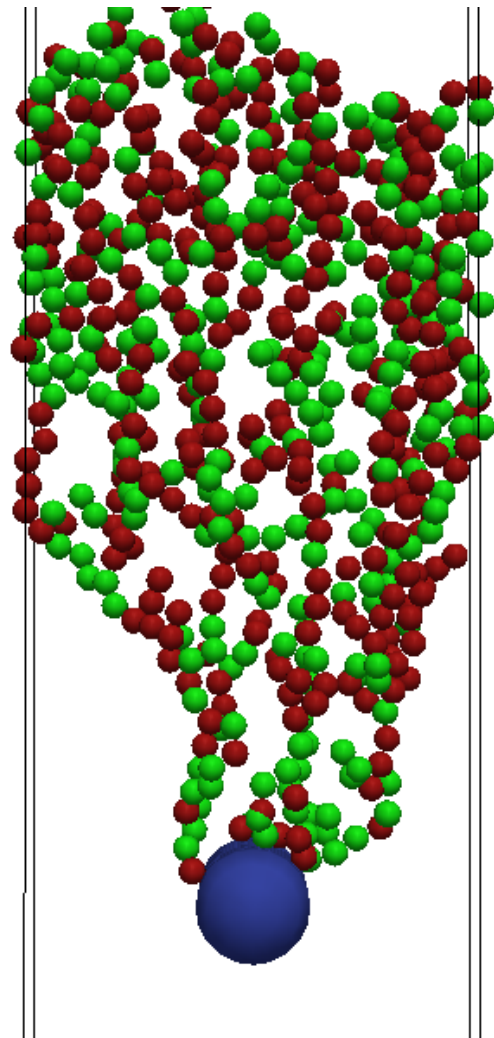


Abbildung 4.11: Struktur-Aufbau einer Einzelfaser mit langsamerem Fluid

In Abbildung 4.10 und Abbildung 4.11 sind die finalen Strukturen der Einzelfaser für unterschiedliche Fluidgeschwindigkeiten dargestellt. Sie sind deutlich höher und weniger dicht als bei zwei Fasern. Vergleicht man die Strukturen untereinander, so sehen sie sich sehr ähnlich. Grund hierfür ist der geringe Einfluss der unterschiedlichen Fluidgeschwindigkeiten auf die gewählten Partikeltypen. Lediglich die Höhe und der zeitliche Verlauf des Aufbaus unterscheiden sich geringfügig. Die Höhe ist unterschiedlich groß, da in der Simulation mit dem langsameren Fluid mehr Simulationsschritte nötig waren, um eine 100 prozentige Effizienz zu erlangen. Folglich ist auch die Anzahl der zugegeben Partikeln höher und somit die Anzahl der abgeschiedenen Partikeln. Der zeitliche Aufbau der Struktur unterscheidet sich dahingehend, dass sich die Struktur der abgeschiedenen Partikel bei einer höheren Fluidgeschwindigkeit schneller nach oben ausbreitet. Dies ist zwar auch bei dem langsameren Fluid der Fall, hier ist

der Unterschied jedoch nicht so stark ausgeprägt. Erkennen lässt sich dies an den einzelnen „Pfad“ der abgeschiedenen Partikeln, welche ihren Ursprung an der Faser haben und sich danach unterschiedlich ausbreiten

Der Strukturaufbau hängt aber nicht nur von den gewählten Parametern, wie beispielsweise der Fluidgeschwindigkeit oder der Masse eines Partikels ab. Es kommt auch auf die Modellierung der Simulation an. So breitet sich beispielsweise die Struktur deutlich steiler nach oben aus, wenn die Zeitschritte der Partikelzugabe dichter aufeinander folgen. Grund hierfür ist die Implementierung des Fluid Constraints für abgeschiedene Partikel. Dieses wird erst definiert, wenn ein Partikel abgeschieden wurde. Folgen die nächsten Partikeln unmittelbar darauf, kann es vorkommen, dass sich das Fluid nicht rechtzeitig auf das neue Constraint ausrichten kann. Die folgenden Partikel wirken dadurch träger und der Sperrereffekt fällt entsprechend gering aus.

Eine weitere Charakterisierung der finalen Struktur erfolgt über die Analyse der Dichte der Struktur. Diese beschreibt das Verhältnis zwischen dem Volumen der abgeschiedenen Partikeln V_P und dem Gesamtvolumen des Körpers V_K , in dem sie sich befinden. Aktuell wird dieser Körper im Skript als Quader modelliert. Hierbei wird am Ende der Simulation die Position der abgeschiedenen Partikel betrachtet und der jeweils kleinste und größte Koordinatenwert jeder Achse ermittelt ($Min_X, Max_X, Min_Y, Max_Y, Min_Z, Max_Z$). Aus den Werten ergibt sich das Volumen des Körpers V_K

$$(4.2) \quad V_K = (Max_X - Min_X) * (Max_Y - Min_Y) * (Max_Z - Min_Z)$$

und daraus die Dichte der Struktur

$$(4.3) \quad Dichte = \frac{V_P}{V_K}$$

Aus Gleichung (4.3) ergeben sich die Dichten der Partikelstrukturen, die in Abschnitt 4.2 dargestellt sind. Die Dichte für zwei Fasern ist, im Vergleich zu den beiden anderen Werten, deutlich höher. Dies liegt an der deutlich kleineren und dichteren Struktur der abgeschiedenen Partikeln. Die beiden Werte der Einzelfaser liegen aufgrund des geringen Einflusses der Fluidgeschwindigkeit im selben Bereich. Hier spielt unter anderem die zufällige Partikelplatzierung einen Einfluss auf das Resultat.

4.3 Strukturaufbau der abgeschiedenen Partikel und dessen Dichte

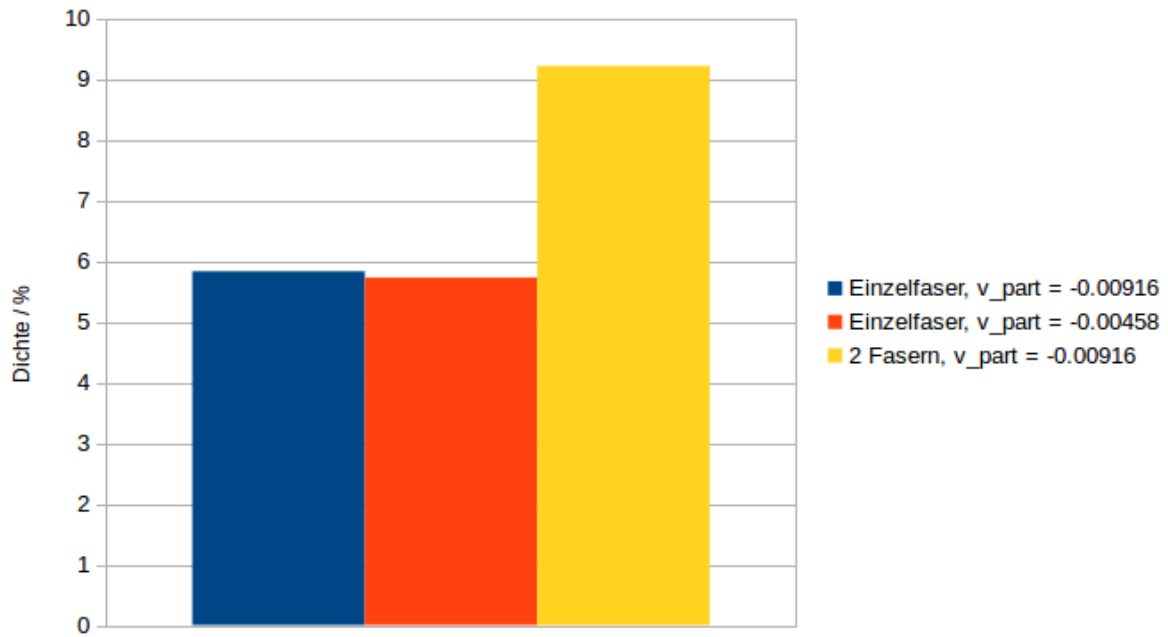


Abbildung 4.12: Dichte der Partikelkörper aus den Simulation zur zeitlichen Effizienz

5 Performance

Abschließend werden noch relevante Performance-Daten analysiert und erläutert worauf der Benutzer achten sollte.

5.1 Einflüsse auf die Gesamtlaufzeit einer Simulation

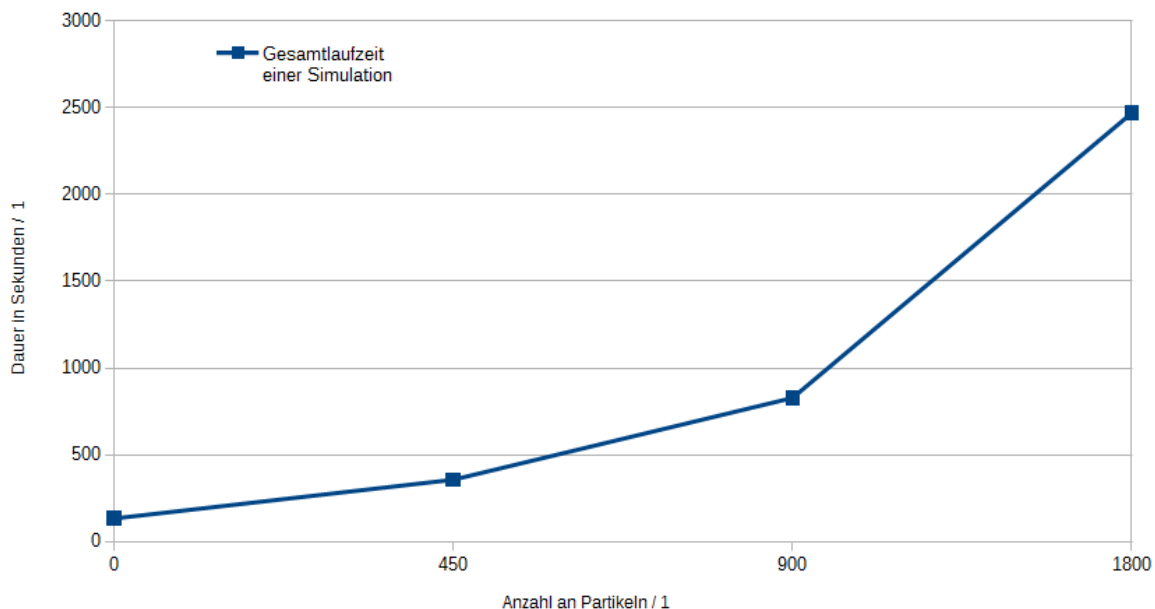


Abbildung 5.1: Laufzeit einer Simulation, in Abhängigkeit von der Anzahl der zugegebenen Partikeln

In Abbildung 5.1 ist die Laufzeit der Simulation in Abhängigkeit zur Anzahl der zugegeben Partikeln veranschaulicht. Hierbei blieb die Anzahl der Integrationschritte und das Intervall der Partikelzugabe gleich. Nur die Anzahl der zugegeben Partikel pro Batch wurde variiert. Die Laufzeit steigt prinzipiell linear mit der Anzahl der Partikeln an. Der Grund für den exponentiellen Verlauf der Kurve ist, dass mit der Anzahl der zugegeben Partikeln auch die Anzahl der abgeschiedenen Partikeln steigt. Diese wiederum verbrauchen dauerhaft bis zum Ende der Simulation Rechenzeit. Die Dauer eines Integrationschritts ist daher abhängig von

der Anzahl der Partikeln, die sich zu diesem Zeitpunkt in der Domain befinden. Die Anzahl wird wiederum von mehreren Faktoren beeinflusst. Neben der Modellierung, wann und wie viele Partikeln zugegeben werden, ist vor allem die Höhe der Domain und die Fluidgeschwindigkeit ausschlaggebend. Die Höhe legt die Länge der Trajektorie für die Partikel fest und die Geschwindigkeit des Fluids beeinflusst die Partikelgeschwindigkeit. Beides ist entscheidend, wie lange ein Partikel in der Domain existiert, bis es den unteren Rand der Domain erreicht und gelöscht wird.

5.2 Vergleich der Performance mit dem ursprünglichen Skript

Nachdem die allgemeine Laufzeit einer Simulation erläutert wurde, soll nun die Performance der Implementierung genauer analysiert werden. Zunächst soll die Optimierung der Kollisionsberechnung mit dem ursprünglichen Skript verglichen werden. Aufgrund der eingeschränkten Funktionalität und der deutlich höheren Laufzeit, wurde eine simple Simulation mit der einmaligen Zugabe von 30 Partikeln durchgeführt. Um den Einfluss durch andere Faktoren, wie etwa der Berechnung des Fluids, gering zu halten, wurde eine entsprechend kleinere Domain gewählt. Die Laufzeit des aktuellen Skripts betrug 17 Sekunden. Die Laufzeit des ursprünglichen Skripts lag bei 398 Sekunden und hat somit circa 20 Mal so lange gebraucht. Grund hierfür ist die geringere Anzahl der Aufrufe zur Bestimmung der Umgebung. Diese wurde pro Integrationsschritt von $n^2 - mn$ auf n reduziert, wobei n für die Gesamtzahl der Partikeln und m für die Anzahl der Faserpartikeln steht.

5.3 Anteil der Umgebungskorrektur an der Gesamtlaufzeit

Eine weitere Anpassung an dem Skript ist die Korrektur der Umgebung, welche in Kapitel 3 erläutert wurde. Den tatsächlichen Einfluss zu messen stellt sich jedoch nicht als einfach heraus. Um den exakten Mehraufwand zu bestimmen, müsste im Voraus der Verlauf einer Simulation bekannt sein. Dies bedeutet, dass auch ohne Korrektur der Umgebung, jedes Partikel zum identischen Zeitpunkt an der identischen Stelle abgeschiedenen werden muss, wie mit Korrektur. Ansonsten wären die Simulationen nicht identisch und die daraus resultierende Laufzeit könnte durch andere Faktoren beeinflusst werden. Eine weitere Möglichkeit wäre die Verwendung von nur einem Partikeltypen. Dann wäre die Simulation auch ohne Korrektur identisch. Dies macht die Korrektur jedoch überflüssig, da die Umgebung, welche von ESPResSo berechnet wird, bereits korrekt ist. Die Laufzeiten wären so gut wie identisch. Daher wird die Laufzeit per Tcl-Timer gemessen, indem man die Uhrzeit in Millisekunden vor und nach dem Prozeduraufruf misst und anhand dieser die Dauer ermittelt. Diese Methode ist zwar genau, es

5.3 Anteil der Umgebungskorrektur an der Gesamtlaufzeit

fehlt jedoch ein Vergleichswert, da eine korrekte Berechnung der Umgebung durch ESPResSo auch nicht bei Null läge. Nichts desto trotz sind in Abbildung 5.2 Messwerte dargestellt, bei denen die Anzahl der zugegeben Partikeln variiert.

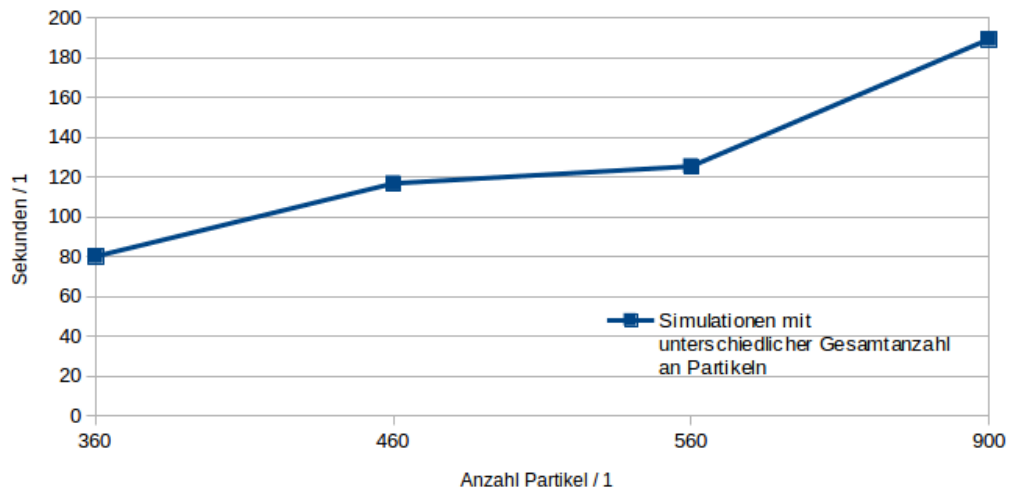


Abbildung 5.2: zeitlicher Anteil der Umgebungskorrektur

Wie in Abbildung 5.2 zu sehen ist, steigt die benötigte Rechenzeit der Prozedur mit der Anzahl der Partikeln an. Grund hierfür ist auch die steigende Anzahl der abgeschiedenen Partikeln. Auf diese wird die Prozedur wiederum aufgerufen.

Ein weiterer Faktor der eine Rolle spielt, ist die Differenz zwischen dem kleinsten und größten Partikeltyp in der Simulation. Der günstigste Fall ist dementsprechend, wenn es nur einen Partikeltyp gibt, oder alle Partikel den gleichen Radius haben. Dann ist die Korrektur überflüssig, da die von ESPResSo berechnete Umgebung bereits ausreichend ist. Hier haben alle Partikeln den gleichen Abstand, der nötig ist, um abgeschieden zu werden.

Der teuerste Fall wäre hier eine Simulation in der überwiegend kleine Partikeln zugeben werden, aber auch ein deutlich größerer Partikeltyp existiert. Die Liste der naheliegenden Partikeln wäre verhältnismäßig groß und ein Großteil müsste jedes Mal aus der Liste entfernt werden.

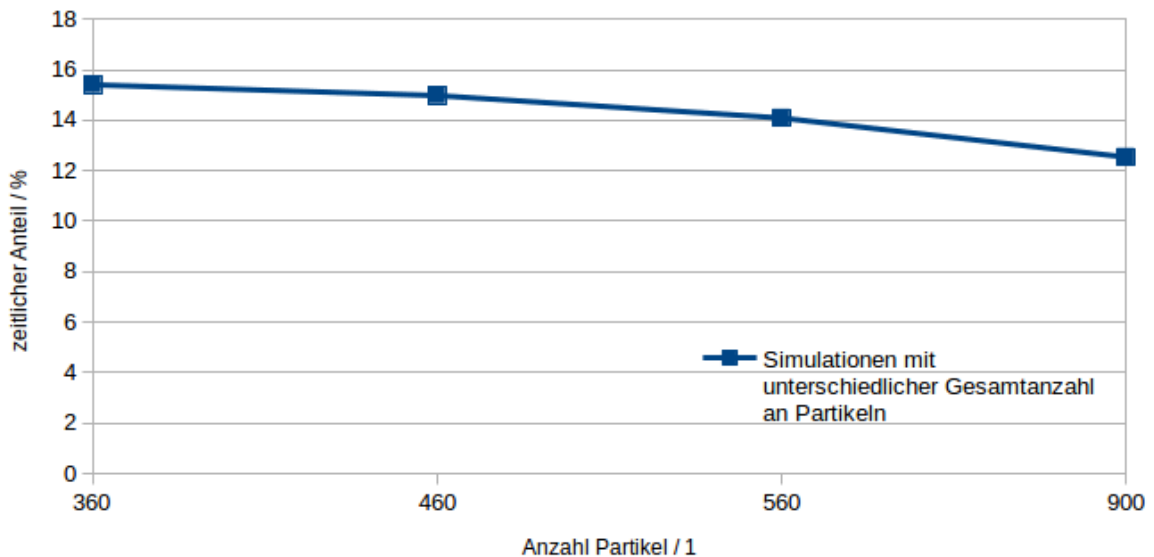


Abbildung 5.3: Darstellung der benötigten Rechenzeit der Umgebungskorrektur, im Verhältnis zur Gesamtlaufzeit

In Abbildung 5.3 ist der prozentuale Anteil der Umgebungskorrektur, im Verhältnis zur Gesamtlaufzeit dargestellt. Hierbei fällt der Anteil mit steigender Anzahl an Partikeln. Dies liegt daran, dass die benötigte Rechenzeit zur Simulation der Partikeln schneller steigt, als der Aufwand für die Korrektur der Umgebung.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Skript für die Software für Molekulardynamik ESPResSo weiterentwickelt, um die kontinuierliche Beladung von einzelnen Fasern mit unterschiedlichen Partikeltypen zu simulieren. Zunächst werden grundlegende Effekte und Anforderungen zur Filtration erläutert. Anschließend werden die gewählten Modelle zur Beschreibung der Wechselwirkungen zwischen Partikeln und der Simulation des Fluids vorgestellt. Außerdem wird die Software ESPResSo vorgestellt.

In Kapitel 3 wird das entwickelte Skript näher beschrieben. Als Erstes wird der Funktionsumfang des ursprünglichen Skripts mit der fertigen Erweiterung verglichen. Danach wird das erweiterte Skript genauer vorgestellt. Hierbei wird der Aufbau und die Bedienung des Skripts, sowie die implementierte Kollisionsberechnung der Partikeln erläutert.

Im letzten Kapitel wird auf Simulationsergebnisse eingegangen, die mit dem Skript erstellt wurden. Es werden der absolute und zeitliche Abscheidegrad, sowie die Struktur und die Dichte der abgeschiedenen Partikeln, in Abhängigkeit von unterschiedlichen Parametern, vorgestellt und die Zusammenhänge erklärt. Wir sind zu der Erkenntnis gekommen, dass der Abscheidegrad mit steigender Stokes-Zahl ansteigt. Partikeln mit geringer Trägheit bleiben vor allem an anderen, bereits abgeschiedenen Partikeln hängen. Träge Partikeln werden sowohl an der Faser, als auch an anderen Partikeln abgeschieden. Unterschiedliche Geschwindigkeiten des Fluids müssen nicht zwingend Einfluss auf alle Partikeltypen haben. In diesem Fall wurde nur der Abscheidegrad von weniger trägen Partikeln beeinflusst. Dies gilt ebenfalls für die Wahl der Faserstruktur. Die Platzierung einer weiteren, identischen Faser hat sich überwiegend auf den Abscheidegrad von trägen Partikeln ausgewirkt. Auch der Aufbau der abgeschiedenen Partikeln hängt von der Fluidgeschwindigkeit und von der Faserstruktur ab. Während die Struktur der abgeschiedenen Partikeln bei zwei Fasern klein und sehr dicht ausfällt, breitet sie sich bei einer einzelnen Faser pyramidenförmig nach oben aus.

Abschließend wird die Performance des Skripts analysiert. Hierbei wird auf die Modellierung einer Simulation eingegangen und erläutert, welche Faktoren einen generellen Einfluss auf die Laufzeit der Simulation haben. Wir konnten feststellen, dass die Rechenzeit für einen Integrationsschritt vor allem mit der Anzahl der, sich in der Domain befindlichen, Partikeln steigt. Zuletzt werden Ergebnisse vorgestellt, welche die Optimierung des ursprünglichen Skripts, als auch die gestiegene Komplexität des erweiterten Skripts, veranschaulichen.

Ausblick

In diesem Abschnitt werden einige Punkte betrachtet, die für eine zukünftige Weiterentwicklung in Betracht gezogen werden können. Zunächst sollen Aspekte erläutert werden, die unabhängig von der Entwicklung von ESPResSo sind. Hierzu gehören das Abscheidemodell und die Berechnung der Strukturdichte. Das bisherige Abscheidemodell scheidet ein Partikel ab, sobald es eine bestimmte Distanz zu einem Faserpartikel, oder einem abgeschiedenen Partikel hat. Dieses könnte beispielsweise mit einem zusätzlichen Wahrscheinlichkeitswert, in Abhängigkeit des Winkels, erweitert werden. Beispielsweise muss ein sehr träges Partikel nicht unbedingt abgeschiedenen werden, wenn es die Faser am äußeren Rand streift.

Zur Berechnung der Strukturdichte der abgeschiedenen Partikeln wird ein vereinfachtes Modell in Form eines Quaders verwendet, um die Struktur zu modellieren. Das resultierende Volumen fällt in den meisten Fällen zu hoch aus, da die Struktur pyramidenförmig ist. Hier könnte einerseits die Form genauer ermittelt werden, als auch die Erkennung von lokalen Strukturen bei mehreren Fasern berücksichtigt werden.

Abhängig von der Entwicklung von ESPResSo und dem Simulationsmodell, könnte die Kollisionsberechnung in speziellen Fällen optimiert werden. Der in Kapitel 5 beschriebene „teuerste Fall“ der Umgebungskorrektur könnte etwa beseitigt werden, wenn ESPResSo die Umgebung eines Partikels nicht nur nach Radius, sondern zusätzlich auch nach einem bestimmten Typ ermitteln könnte. Eine, im Kern von ESPResSo implementierte Berechnung, wäre aber auch im Allgemeinen performanter, da diese parallelisiert durchgeführt werden kann.

Abschließend sollen die von ESPResSo berechneten Partikelkräfte erwähnt werden. Diese können bei der Wahl von falschen Parametern unter Umständen so stark steigen, dass die Simulation mit einer Fehlermeldung abbricht. Ursache hierfür können viele Faktoren eine Rolle spielen, wie beispielsweise

- die Reibung des Fluids
- die Fluidgeschwindigkeit
- das Verhältnis zwischen Gitter- und Partikelgröße
- das Verhältnis zwischen Gittergröße und Partikelgeschwindigkeit
- Wechselwirkungen zwischen den Partikeln

Um letztere Fehlerquelle auszuschließen ist es daher ratsam, das Potential zwischen den Partikeln zu beschränken.

Literaturverzeichnis

- [1] Thomas Schartner. „Sensitivitätsstudien zum meteorologischen Einfluss auf die Modellierung des Feinstaubtransportes“. diploma thesis. Institut für Meteorologie Fachbereich Geowissenschaften Freie Universität Berlin, 2007.
- [2] *Gesundheitliche Wirkungen von Feinstaub und Stickstoffdioxid im Zusammenhang mit der Luftreinhalteplanung*. Landesamt für Natur, Umwelt und Verbraucherschutz Nordrhein-Westfalen, 2010.
- [3] Aaron van Donkelaar. *Global Estimates of Fine Particulate Matter from Satellite*. International Conference on High Performance Computing & Simulation. 2014.
- [4] *Feinstaub (PM 2,5)*. Umweltbundesamt. 2016. URL: <http://www.umweltbundesamt.at/umweltsituation/luft/luftschadstoffe/staub/pm25/>.
- [5] *Feinstaub - Jährliche Auswertungen*. Stand 21.06.2016. Umweltbundesamt. 2016. URL: <http://www.umweltbundesamt.de/themen/luft/luftschadstoffe/feinstaub>.
- [6] *WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide- global update 2005*. 2006. URL: http://apps.who.int/iris/bitstream/10665/69477/1/WHO_SDE_PHE_OEH_06.02_eng.pdf.
- [7] H-P. Baumgarner und F. Löffler. *The collection performance of electret filtern in the particle size range 10nm-10µm*. Techn. Ber. J. Aerosol Sci. Vol. 17. Institut für Mechanische Verfahrenstechnik und Mechanik, Universität Karlsruhe, 1986.
- [8] *Filterklassen nach DIN EN 779:2012*. 2012. URL: <http://www.freudenberg-filter.com/de/luftfiltration/standards-und-zertifizierungen/filterklassifizierungs-norm-en-7792012/>.
- [9] Oleg Iliev, Ralf Kirsch, Zahra Lakdawala, Stefan Rief and Konrad Steiner. *Modellierung und Simulation von Filtrationsprozessen*. 2015.
- [10] *Laboratory filtration- Principles and chemical compatibility chart*. GE Healthcare UK Limited. GE Healthcare Life Sciences, März 2013.
- [11] A. Hellmann, K. Schmidt, S. Ripperge, C. Thelen, C. Möhlmann. „Bestimmung der Abscheidewirkung von faserförmigen Filtermedien gegenüber ultrafeinen Partikeln“. In: *Gefahrstoffe- Reinhaltung der Luft*. Springer VDI Verlag, 2012, S. 477–483.
- [12] Prof. Dr. Thomas Nussbaumer. *Staubabscheidesysteme- Eine Einführung*. Techn. Ber. BAFU-Fachtagung Holzverbrennung und Feinstaub, Bern 8.11.11. Hochschule Luzern-Technik und Architektur Fachgruppe Bioenergie, Nov. 2011.

- [13] K. W. Lee, B. Y. H. Liu. „On the Minimum Efficiency and the Most Penetrating Particle Size for Fibrous Filters“. In: *Journal of the Air Pollution Control Association* 30.4 (1980), S. 377–381. eprint: <http://dx.doi.org/10.1080/00022470.1980.10464592>. URL: <http://dx.doi.org/10.1080/00022470.1980.10464592>.
- [14] Vytautas Gapsys, Daniel Seeliger, und Bert L. de Groot. „New Soft-Core Potential Function for Molecular Dynamics Based Alchemical Free Energy Calculations“. In: *Journal of Chemical Theory and Computation (JCTC)* (2012).
- [15] Olaf Lenz. „Technical Tricks of Coarse-Grained MD Visualization with VMD“. Institut für Computerphysik, Universität Stuttgart Stuttgart, Germany.
- [16] Georg Rempfer. „Lattice-Boltzmann Simulations in Complex Geometries“. Diss. Institute for Computational Physics , University of Stuttgart, 2010.
- [17] A. Arnold, O. Lenz, S. Kesselheim, R. Weeber, F. Fahrenberger, D. Roehm, P. Košovan & C. Holm. „ESPResSo 3.1 – Molecular Dynamics Software for Coarse-Grained Models“. In: *Meshfree Methods for Partial Differential Equations VI*. Hrsg. von M. Griebel, M. A. Schweitzer. Bd. 89. Lecture Notes in Computational Science and Engineering. Springer, 2013, S. 1–23. URL: <http://www.springer.com/mathematics/computational+science+%26+engineering/book/978-3-642-32978-4>.
- [18] H. J. Limbach, A. Arnold, B. A. Mann & C. Holm. „ESPResSo – An Extensible Simulation Package for Research on Soft Matter Systems“. In: *Comp. Phys. Comm.* 174.9 (Mai 2006), S. 704–727.
- [19] *Coarse-graining models*. URL: <http://espressomd.org/wordpress/about/summary/>.
- [20] Olaf Lenz. „Introduction to ESPResSo and Tcl“. Institut für Computerphysik, Universität Stuttgart Stuttgart, Germany.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift