

**Gemeinsamer Abschlussbericht
des BMBF-Projekts**

SCAFACOS

Scalable Fast Coulomb Solvers

L. ARNOLD, H. DACHSEL, P. GIBBON, R. HALVER, I. KABADSHOW,
M. WINKEL, UND G. SUTMANN
Jülich Supercomputing Cente (JSC), Forschungszentrum Jülich GmbH

F. HEBER, R. WILDENHUES, UND M. GRIEBEL
Institut für Numerische Simulation, Rheinische Friedrich-Wilhelms-Universität Bonn

M. PIPPIG, UND D. POTTS
Fakultät für Mathematik, Technische Universität Chemnitz

M. HOFMANN, UND G. RÜNGER
Fakultät für Informatik, Technische Universität Chemnitz

A. ARNOLD, F. FAHRENBERGER, O. LENZ, UND C. HOLM
Institute for Computational Physics, Universität Stuttgart

F. GÄHLER
Institut für Theoretische und Angewandte Physik, Universität Stuttgart
jetzt: Fakultät für Mathematik, Universität Bielefeld

Das diesem Bericht zugrundeliegende Vorhaben SCAFACOS wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 01 IH 08001 A-D im Rahmen des ersten Calls „HPC-Software für skalierbare Parallelrechner“ von Januar 2009 bis Dezember 2011 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Inhaltsverzeichnis

1	Kurze Darstellung des Projekts	1
1.1	Aufgabenstellung	2
1.2	Randbedingungen und Voraussetzungen	3
1.3	Planung und Ablauf des Vorhabens	4
1.4	Wissenschaftlicher und technischer Stand	4
1.4.1	Parallele Sortierung	5
1.4.2	Nichtäquidistante schnelle Fourier-Transformationen	6
1.4.3	Schnelle Multipol Methode	6
1.4.4	Barnes-Hut Treecode	8
1.4.5	Periodische Summationsverfahren	8
1.4.6	Mehrgitter-Verfahren	9
1.4.7	Lokale Gitterverfahren	10
1.4.8	Molekulardynamik-Code IMD	11
1.5	Zusammenarbeit mit anderen Stellen	11
2	Eingehende Darstellung des Projekts	15
2.1	Verwendung der Zuwendung und erzielte Ergebnisse	15
2.1.1	Entwicklung des gemeinsamen Interfaces	15
2.1.2	WP 1 – Entwicklung und Parallelisierung der Basisbibliotheken	16
2.1.3	WP 2 – Entwicklung und Parallelisierung der Teilmodule	24
2.1.4	WP 3 – Benchmark und Evaluierung	44
2.1.5	Erzielte Fortschritte in der entwickelten Löser-Bibliothek	54
2.2	Wichtigste Positionen des zahlenmäßigen Nachweises	61
2.2.1	Nachweis Universität Bonn	61
2.2.2	Nachweis Universität Chemnitz	62
2.2.3	Nachweis Jülich Supercomputing Centre	63
2.2.4	Nachweis Universität Stuttgart	64
2.3	Notwendigkeit und Angemessenheit der geleisteten Arbeit	67
2.4	Voraussichtlicher Nutzen, insbesondere Verwertbarkeit der Ergebnisse im Sinne des fortgeschriebenen Verwertungsplans	71
2.5	Während der Durchführung des Vorhabens bekannt gewordene Fortschritte bei anderen Stellen	75

2.6	Erfolgte oder geplante Veröffentlichungen der Ergebnisse	77
2.6.1	Veröffentlichungen Bonn	77
2.6.2	Veröffentlichungen Chemnitz	77
2.6.3	Veröffentlichungen Jülich	81
2.6.4	Veröffentlichungen Stuttgart	86
2.6.5	Veröffentlichungen Wuppertal	88

1 Kurze Darstellung des Projekts

Computersimulationen von atomistischen Systemen spielen in vielen Bereichen der Wissenschaft und industriellen Forschung sowie der Anwendung eine wichtige Rolle. Als komplementäre Verfahren haben sich dabei die Monte Carlo Methode und die molekulardynamischen Simulationen etabliert. Beiden Verfahren liegt zugrunde, dass Teilchen bzw. atomistische Einheiten durch Kraftfelder miteinander wechselwirken. Die Kraftfelder beinhalten dabei alle Informationen, die es möglich machen, verschiedene Teilchenarten durch unterschiedliche Eigenschaften zu charakterisieren. Während man bei Monte Carlo Verfahren die Potentiale auswertet, um Akzeptanzwahrscheinlichkeiten für Konfigurationen zu bestimmen, braucht man bei molekulardynamischen Simulationen die Kräfte zwischen Teilchen, um Bewegungen zu beschreiben. Diese Kräfte können entweder durch Bindungen innerhalb eines Moleküls oder durch nichtgebundene Wechselwirkungen zwischen Molekülen und Teilchen untereinander hervorgerufen werden. Besitzen Teilchen Partialladungen oder elektrische Dipole, hat man es mit langreichweitigen Wechselwirkungen zu tun, was einen gegenseitigen Einfluss aller sich im System befindlichen Teilchen impliziert. Um komplexe Systeme mit Partialladungen effektiv simulieren zu können, wurden verschiedene Algorithmen entwickelt, die die numerische Komplexität von $\mathcal{O}(N^2)$ auf $\mathcal{O}(N \log N)$ oder sogar $\mathcal{O}(N)$ reduzieren.

Hier haben sich für periodische Teilchensysteme die Particle-Particle Particle-Mesh Methode (P3M) sowie für offene Systeme der Barnes-Hut Algorithmus bzw. die schnelle Multipol Methode (FMM) etabliert. Allerdings gibt es aufgrund traditioneller Entwicklungen noch eine ganze Reihe anderer Methoden, die in aktuellen Simulationsprogrammen verwendet werden, die teilweise aber weder die Performance noch die numerische Genauigkeit der erstgenannten Methoden erreichen.

Aufgrund der teilweise enormen Rechenzeitanforderungen zur Bewältigung der gestellten Probleme ist es unabdingbar effiziente, parallele Programme zu entwickeln und auf hoch skalierenden Rechnerarchitekturen auszuführen. Da die Berechnung der interatomaren Wechselwirkungen den größten Anteil der numerischen Arbeit einer Simulation darstellt und dabei insbesondere die langreichweitigen Wechselwirkungen eine herausragende Rolle spielen, ist es notwendig, skalierbare Algorithmen zu entwickeln, die in der Lage sind, auf heutigen und zukünftigen Rechnerarchitekturen effizient eingesetzt zu werden. Dies bedeutet insbesondere, parallele Algorithmen mit speziellen Methoden zu verknüpfen, die das Verhältnis von Kommunikation und Rechnung minimieren, um auf Tausenden von Prozessoren effizient eingesetzt werden zu können. Erst dies erlaubt zum

einen eine schnellere Berechnung und zum anderen den Schritt zu größeren Systemen, die Zugang zu einer realistischeren Beschreibung liefern und Phänomene beschreiben können, die erst auf großen Längen- und Zeitskalen auftreten.

1.1 Aufgabenstellung

Ausgangspunkt für das vorliegende Projekt war die serielle Version einer Bibliothek für Coulomb-Löser, die am JSC entwickelt wurde und die FMM, den Barnes-Hut Algorithmus sowie eine Ewald-Summation beinhaltet. In dem Projekt sollten dann zum einen die bestehenden Verfahren effizient parallelisiert werden und zum anderen noch weitere Verfahren mit integriert werden. Das Ziel dabei war, durch eine Vielfalt von Methoden, die für verschiedene Simulationsszenarien eingesetzt werden können, eine weitest mögliche Verbreitung der Bibliothek zu erreichen. Die verwendeten Verfahren können dabei teilweise als komplementär zu einander betrachtet werden. So hat das Barnes-Hut Verfahren Vorteile bei inhomogenen, unstrukturierten Systemen, schnelle Multipol-Verfahren sind für homogene Verteilungen sehr gut anzuwenden und Mehrgitterverfahren können beispielsweise bei dielektrischen Diskontinuitäten angewendet werden, für die es in den anderen Verfahren keine Entsprechung gibt.

Für den Anwender, bzw. das Anwendungsprogramm, sollte die detaillierte parallele Implementierung der Algorithmen keine Rolle spielen. Die Schnittstelle zwischen Programm und Bibliothek sollte dabei soweit ausgebaut werden, daß es einerseits möglich sein sollte, dass die verschiedenen Algorithmen sowohl feste als auch optionale Parameter erhalten sollten und zusätzlich alle Erfordernisse einer parallelen Implementierung abgedeckt sein sollten. Die zugrunde liegenden Algorithmen sollten weitestgehend autonom arbeiten in sofern, als dass Lastbalance, evtl. notwendige Sortierung der Daten sowie Kommunikation von abhängigen Daten bzw. Lösungsvektoren intern kontrolliert und verwaltet werden sollten.

Die parallele Zielplattform für die Bibliothek sollten massiv parallele Architekturen sein, wie beispielsweise das System IBM Blue Gene/P in Jülich mit ≈ 300.000 Prozessoren. Ein Ziel war es, die Bibliothek für Probleme vorzubereiten und bereit zu stellen, die eine sehr große Anzahl von Prozessoren erfordern, d.h. es sollte eine Skalierung auf Prozessorzahlen $\mathcal{O}(10^5)$ erreicht werden. Um die Effizienz der Bibliothek in einem realistischen Rahmen zu testen, sollten für mehrere existierende Molekulardynamik-Codes (z.B. IMD[104], ESPRESSO[63, 93]), die unter anderem im industriellen und akademischen Umfeld verbreitet sind, eine Anbindung an die Bibliothek erstellt und Skalierungsmessungen durchgeführt werden.

1.2 Randbedingungen und Voraussetzungen

Methoden für langreichweitige Wechselwirkungen zwischen Teilchen in komplexen Systemen werden aufgrund ihrer großen Bedeutung und Verbreitung in verschiedenen internationalen Gruppen entwickelt. Dabei haben sich die Gruppen zumeist auf bestimmte, einzelne Verfahren spezialisiert, die für verschiedene Systeme angewendet werden und entsprechend der Aufgabenstellung weiter entwickelt werden. Da der zeitliche Aufwand zur Entwicklung, Implementierung und Optimierung aber in der Regel sehr groß ist, ist die Community von Benutzern auf existierende Methoden angewiesen, die häufig in einzelnen Simulationsprogramme integriert sind. Für den flexiblen Gebrauch von verschiedenen Methoden, die das Skalierungsverhalten und die Fehlertoleranz für ein gegebenes System optimieren oder verschiedene Randbedingungen zulassen, gibt es bisher keine Lösung. Hierfür bietet sich die Bereitstellung einer Bibliothek an, über die verschiedene Methoden adressiert werden können und die dem Benutzer eigene aufwendige Implementierungsarbeiten erspart.

Das Projekt SCAFACOS hatte daher zum Ziel, eine Bibliothek zu erstellen, die mehrere moderne Methoden beinhaltet, die auf verschiedene Simulationsszenarien angewendet werden können und sich in ihrem Skalierungsverhalten und der erreichbaren Fehler-schranken unterscheiden können. Da aufgrund der erwähnten Implementierungsprobleme ein systematischer Test zwischen den verschiedenen Methoden noch aussteht, hatte das Projekt auch zum Ziel, die Voraussetzung für einen nahezu objektiven Vergleich zu schaffen. Hierbei ist natürlich immer zu beachten, dass die Performance in jedem Fall abhängig von der jeweiligen Implementierung ist. Bzgl. der schnellen Multipolmethode existieren z.B. Angaben für Teilchenzahlen in der Literatur, für die die Methode effizient einzusetzen ist, zwischen $N \approx 10^3$ und $N \approx 10^6$.

Die Partner, die in das Projekt involviert waren, brachten Expertise in der Entwicklung und Optimierung von den verschiedenen Methoden mit, die in die Bibliothek integriert werden sollten. Dazu kam Expertise für parallele Sortierverfahren, die essentiell für die Umsetzung der Bibliothek war, da Teilcheninformationen von einem aufrufenden Programm innerhalb der Bibliotheksfunktionen teilweise auf die Prozessoren umverteilt werden müssen. Die Methoden zur Behandlung der langreichweitigen Wechselwirkungen waren überwiegend bei den Projektpartnern bereits in serieller Form verfügbar und sollten effizient parallelisiert und in eine gemeinsame Bibliothek integriert werden, um auf die Bedürfnisse der Simulations-Community einzugehen. Zu den Entwicklungsarbeiten war es daher auch notwendig, ein Interface zwischen Simulationsprogramm und Bibliothek zu entwickeln, das mit verschiedenen Datentypen, unterschiedlicher Anzahl von festen und optionalen Parametern und verschiedenen Programmiersprachen umgehen konnte.

Das Ziel war daher, eine parallele, skalierbare Bibliothek zu erstellen, die in verschiedenen Simulations-Szenarien eingesetzt werden kann und auf einfache Weise durch eine

Schnittstelle in bestehende Simulations-Codes eingebunden werden kann.

1.3 Planung und Ablauf des Vorhabens

Das Projekt SCAFACOS wurde in der Zeit vom 1.1.2009 bis zum 31.12.2011 finanziell gefördert. Da aufgrund von Besetzungsproblemen bei einigen Partnern während der Projektphase finanzielle Mittel nicht vollständig ausgegeben und die Arbeitspakete nicht in vollem Umfang bearbeitet werden konnten, wurde das Projekt bis 30.04.2012 kostenneutral verlängert. Die Arbeiten gliederten sich in drei Arbeitspakete:

- WP 1: Entwicklung von Basis-Modulen
- WP 2: Entwicklung von Kern-Modulen für lang- und kurzreichweitige Wechselwirkungen
- WP 3: Evaluierung und Benchmarks der Methoden

WP 1 beinhaltet dabei die Erstellung von optimierten Implementierungen für die nichtäquidistante schnelle Fourier-Transformation, sowie die Entwicklung und Implementierung von schnellen Sortierverfahren. Diese Verfahren werden in mehreren anderen Methoden verwendet und stellen daher generische Komponenten der Bibliothek dar. In WP 2 wurden die verschiedenen Verfahren teilweise neu parallelisiert bzw. weiter entwickelt oder modularisiert. Die Methoden, die dabei bearbeitet wurden, waren die schnelle Multipol Methode, multipol-basierte Baum-Verfahren, Summationsverfahren, die auf der schnellen Fourier-Transformation basieren, Mehrgitterverfahren sowie ein lokales gitterbasiertes Verfahren zur Lösung der Maxwell-Gleichungen. In WP 3 wurden die Methoden in verschiedenen Szenarien auf Stabilität, Genauigkeit und Skalierbarkeit getestet. Ziel war es hier unter anderem einen Vergleich zwischen den Methoden zu ermöglichen, indem für definierte Systeme die Ausführungszeit und Skalierbarkeit bei vorgegebener tolerierbarer Fehlerschranke betrachtet wurde. Durch Einbindung der Bibliothek in bestehende Molekulardynamik-Programme konnte die Stabilität von System-Trajektorien untersucht werden, indem die Bewegungsgleichungen für Teilchensysteme gerechnet wurden unter Einbeziehung von langreichweitigen Wechselwirkungen, die mit den verschiedenen implementierten Methoden bestimmt wurden.

1.4 Wissenschaftlicher und technischer Stand

Das Projekt hatte zum Ziel, mehrere Methoden zur Behandlung langreichweitiger Wechselwirkungen in einer gemeinsamen skalierbaren Bibliothek zu integrieren. Basis-Methoden, die in mehreren Verfahren eingesetzt werden, sind dabei schnelle Fourier-Transformationen und die parallele Sortierung. Bei der schnellen Fourier-Transformation stand eine Version einer nichtäquidistanten schnellen Fourier-Transformation zur Verfügung, die

sowohl für ein neues schnelles Summationsverfahren, als auch für Ewald-Summationsverfahren eingesetzt werden sollte. Als Verfahren zur Behandlung der langreichweitigen Wechselwirkungen zwischen Teilchen wurden von den Partner-Gruppen die schnelle Multipol Methode, der Barnes-Hut Baumalgorithmus, Mehrgitterverfahren und Fourierbasierte Summationsverfahren in das Projekt eingebracht. Ein lokales Gitterverfahren sollte neu implementiert und parallelisiert werden. Um die Methoden und die Bibliothek in einer realistischen Simulationsumgebung testen zu können, stand der Molekulardynamik-Code IMD des Stuttgarter Partners zur Verfügung. Im folgenden wird die Ausgangssituation für die einzelnen Komponenten beschrieben, die in dem Projekt bearbeitet wurden.

1.4.1 Parallele Sortierung

Sowohl für sequentielles als auch paralleles Sortieren existierten zu Beginn des Projekts bereits eine Vielzahl an verschiedenen Sortierverfahren in der Literatur. Des Weiteren wurden zunehmend Verfahren entwickelt, welche gezielt moderne Multicore-Prozessoren, beispielsweise in Form von Grafikprozessoren, ausnutzen. Je nach Anwendungsgebiet und verwendetem Rechnersystem musste eine Auswahl geeigneter Sortierverfahren zumeist individuell durchgeführt werden. Praktische Ansätze zu hochskalierenden parallelen Sortierverfahren waren zudem nur wenig untersucht worden.

In Zusammenarbeit mit dem Partner im Forschungszentrum Jülich war vor Projektbeginn bereits ein erstes paralleles Sortierverfahren für die vorhandene FMM Implementierung realisiert worden. Zwar konnte dies seine Arbeitsweise bereits adaptiv an den jeweils zur Verfügung stehenden Hauptspeicher anpassen kann, unterlag dabei aber gleichzeitig deutlichen Einschränkungen hinsichtlich der Verteilung der zu sortierenden Partikeldaten vor und nach der parallelen Sortierung. Weiterer Ausgangspunkt für die Arbeiten zur parallelen Sortierung an der TU Chemnitz war ein bereits vorhandener Bibliotheksansatz für parallele Sortierverfahren [74]. Hiermit sollte die Verfügbarkeit effizienter Implementierungen paralleler Sortierverfahren für Anwendungen im wissenschaftlichen Rechnen allgemein erreicht werden. Die zum Projektbeginn vorhandene Umsetzung der Sortierbibliothek enthielt bereits die verschiedenen Algorithmenteile des parallelen Sortierverfahrens der FMM und erlaubte es, diese je nach Einsatzgebiet und Anforderung flexibel zu konfigurieren und kombinieren. Die geplante Entwicklung hochskalierender Sortierverfahren und der erweiterte Einsatz in zusätzlichen Anwendungen sollten durch Weiterentwicklung des vorhandenen Bibliotheksansatzes erreicht werden. Auf Grund der Zusammenarbeit mit dem Partnern im Forschungszentrum Jülich bestanden zudem bei Projektbeginn bereits Erfahrungen mit hoch skalierenden Plattformen und den darauf vorhandenen Kommunikationsoperationen.

1.4.2 Nichtäquidistante schnelle Fourier-Transformationen

Für die Berechnung der schnellen Fourier-Transformation existierten zu Projektbeginn eine Vielzahl von seriellen und parallelen Implementierungen. Allerdings waren nur die Bibliotheken [80, 99, 100] auf Grund ihrer zwei-dimensionalen Datenaufteilung als Basis massiv paralleler Algorithmen geeignet. Ein Vergleich dieser Bibliotheken bezüglich Performance und Flexibilität der Schnittstellen stand nicht zur Verfügung. Des Weiteren stand eine detaillierte Anforderungsanalyse aus, um festzustellen welche Funktionalität eine parallele FFT für die effiziente parallele Berechnung von Coulomb-Wechselwirkungen bieten muss.

Mit der an der TU Chemnitz entwickelten NFFT-Softwarebibliothek [89, 90] stand eine etablierte serielle Implementation der nichtäquidistanten schnellen Fourier-Transformation zur Verfügung. Diese Bibliothek beinhaltet außerdem eine serielle Implementation der nicht-periodischen schnellen Summation für kubische Geometrien. Erweiterungen der seriellen schnellen Summation auf periodische Randbedingungen und nicht-kubische Geometrien waren theoretisch unbekannt bzw. praktisch unzureichend erprobt. Erste Erfahrungen mit der Parallelisierung der NFFT für moderate Prozessorenanzahlen wurden in der Diplomarbeit [11] erarbeitet. Weder für die nichtäquidistante Fourier-Transformation noch für die schnelle Summation waren massiv parallele Algorithmen mit verteiltem Speicher untersucht worden. Aufgrund des modularen Aufbaus "FFT→NFFT→schnelle Summation" sollte die Parallelisierung in abgeschlossenen Teilschritten erfolgen. Obwohl die Parallelisierung der schnellen Summation zur schnellen Berechnung der Coulomb-Wechselwirkungen im Fokus des Projektes stand, sind die parallele FFT und die parallele NFFT wichtige Grundbausteine vieler weiterer schneller Algorithmen. Daher wurden als Teilziele die Parallelisierung der FFT und der NFFT und deren Publikation in Form öffentlich verfügbarer Softwarebibliotheken deklariert.

1.4.3 Schnelle Multipol Methode

Im Rahmen des BMBF geförderten Projekts "High-Performance Computing in Chemistry" (2001-2004) wurde mit der Entwicklung einer schnellen Multipol Methode (engl. Fast Multipole Method – FMM) für Ladungsverteilungen in der Quantenchemie begonnen. Weite Teile des Quellcodes konnten erfolgreich in die Entwicklung einer FMM für Punktladungen übernommen werden. Die FMM als schneller Coulomb Löser weist eine optimale Komplexität $\mathcal{O}(N)$ bezüglich der Teilchenanzahl N auf. Durch Verwendung des rotationsbasierten Ansatzes [108] skaliert die Methode mit der Anzahl der Multipolmomente p wie $\mathcal{O}(p^3)$.

Bei der Entwicklung der sequentiellen Version wurde besonderer Wert auf a-priori Fehlerkontrolle und automatische Laufzeitminimierung gelegt. Somit ist es möglich, die FMM-Parameter Baumtiefe, Multipolanzahl und Separationskriterium automatisch ein-

zustellen [73]. Dafür muss der Nutzer lediglich den von ihm geforderten absoluten oder relativen Energiefehler angeben. Der 2-stufige Fehlerschätzer ermittelt dann automatisch die dazu passenden FMM Parameter. Ein aufwendiges und vor allem ungenaues Einstellen der FMM Parameter von Hand entfällt hierbei vollständig. Die Reduzierung der einzustellenden Parameter auf nur eine einzige physikalische Größe erlaubt es, eine FMM für verschiedene Genauigkeitsanforderungen bereitzustellen. In der für das Projekt verwendeten FMM ist es möglich, einen relativen Energiefehler bei Verwendung von doppelt-genauen Gleitkommazahlen von $\Delta E_{\text{rel}} = 10^{-1}$ bis $\Delta E_{\text{rel}} = 10^{-15}$ zu nutzen. Somit kann die FMM in verschiedenen Forschungsbereichen Anwendung finden.

Weiterhin konnte für die sequentielle Implementierung gezeigt werden, dass die Methode für homogen verteilte Punktladungen bei einer Fehlerschranke von $\Delta E_{\text{rel}} = 10^{-5}$ bereits für Teilchenzahlen $N > 500$ effizienter eingesetzt werden kann als direkte paarweise Summation.

Um das Anwendungsfeld zu vergrößern, wurde neben den offenen Randbedingungen auch die Möglichkeit geschaffen periodische Systeme in ein, zwei und drei Dimensionen zu berechnen. Die Anwendbarkeit der a-priori Fehlerkontrolle und automatischen Laufzeitminimierung, sowie die optimale Komplexität behalten auch hierbei ihre Gültigkeit.

Neben diesen anwenderfreundlichen Funktionen besitzt die Implementierung weitere Merkmale, die für die Erstellung einer Softwarebibliothek für schnelle Coulomb-Löser von Vorteil sind:

- **Unterstützung verschiedener Gleitkomma-Datentypen:**
Die FMM erlaubt die Verwendung von einfach-, doppelt- sowie vierfach-genauen Gleitkommazahlen für die Ein- und Ausgabedaten. Hierdurch ergibt sich je nach Anwendung ein Geschwindigkeits- bzw. Genauigkeitsvorteil.
- **Separate Speicherverwaltung:**
Besonders für inhomogene Teilchenverteilungen kann es vorkommen, dass der Speicherbedarf der FMM selbst für relativ kleine Teilchenzahlen den vorhandenen Arbeitsspeicher übersteigt. Daher wurde eine unabhängige vierstufige Speicherverwaltung implementiert, die automatisch, je nach Clusterung der Teilchen, die passende Datenstruktur zur Speicherung wählt und somit ineffiziente Speicherzugriffe durch Fragmentierung vermeidet.
- **Effiziente FMM Baumverwaltung:**
Neben einer maßgeschneiderten Speicherverwaltung ist es außerdem unverzichtbar, die Verwaltungskosten der Datenstrukturen (Nachbarschaftssuche, Schlüsselgenerierung) im Nah- und Fernfeld auf ein Minimum zu reduzieren. Durch die Nutzung einer raumfüllenden Kurve zur Abbildung der 3d-Partikeldaten auf eine 1d-Kurve und den daraus resultierenden Eigenschaften ist dieser Aufwand kleiner 1% bezüglich der Gesamtrechenzeit.

1.4.4 Barnes-Hut Treecode

Der am JSC entwickelte molekulardynamische parallele Baumalgorithmus PEPC beinhaltet als Verfahren für langreichweitige Wechselwirkungen ein Barnes-Hut Schema, das auf einem Gebietszerlegungsalgorithmus durch Morton-Kurven basiert. Hierdurch wurde ein gutes Lastbalance-Verhalten ermöglicht, welches das Programm zum Beginn des Projektes bis zu ca. 4096 Prozessoren skalierbar machte. Der bereits modulare Aufbau dieses Codes erlaubte es, die physikalische Anwendung (Versionen für einfache Elektrostatik, Plasmaphysik und Astrophysik waren zu diesem Zeitpunkt verfügbar) vom BH-Kernel weitgehend zu trennen. Durch Vorgabe einer Fehlerschranke konnten die Parameter eingestellt werden, die zwischen Multipolentwicklung und direkter Summation entscheiden. Mehrere solcher „Multipol-Akzeptanzkriterien“ wurden im Rahmen einer Diplomarbeit implementiert und untersucht. Das Verfahren wurde zunächst für offene Systeme entwickelt. Eine Ewald-Multipol-Summation für periodische Systeme wurde im Rahmen des JSC-Gaststudenten-Programmes ebenfalls provisorisch implementiert, hatte aber begrenzte Genauigkeit.

1.4.5 Periodische Summationsverfahren

Ewald-Gitter-Summation

In den letzten Jahrzehnten wurden diverse Algorithmen zur Berechnung von elektrostatischen Wechselwirkungen in periodischen Teilchensystemen entwickelt, die auf der Ewald-Summe [82] und der schnellen Fourier-Transformation [101] eines Gitters beruhen, beispielsweise die Algorithmen P3M [87], PME [75] oder SPME [81], die eine Komplexität von $\mathcal{O}(N \log N)$ aufweisen. Einer dieser Algorithmen kommt in den meisten Simulationsprogrammen zum Einsatz, in denen elektrostatische Wechselwirkungen eine Rolle spielen.

In den letzten Jahren wurde deutlich, dass alle diese Algorithmen im Wesentlichen als ein Algorithmus mit verschiedenen Komponenten aufgefasst werden können, die prinzipiell ausgetauscht werden können [76]. Die genaue Wahl der Komponenten beeinflusst die Genauigkeit, die Geschwindigkeit sowie die Parallelisierbarkeit des Algorithmus.

Die konkrete Auswahl der Komponenten des Algorithmus in verschiedenen Simulationsprogrammen hat in der Regel historische Gründe; eine systematische Untersuchung der optimalen Kombination von Komponenten insbesondere im Hinblick auf die Geschwindigkeit lag zu Projektbeginn nicht vor.

Der Projektpartner Universität Stuttgart verfügte bereits zu Beginn des Projektes über eine Implementierung des P3M-Algorithmus als Teil der Simulationssoftware ESPRESSO [63, 93]. Diese war parallelisiert, effizient und genau und bereits seit mehreren Jahren in Gebrauch. Mit Hilfe von approximativen analytischen Fehlerformeln war es möglich,

die verschiedenen Parameter der Methode automatisch auf die gewünschte Genauigkeit zu optimieren.

Darüberhinaus war der Algorithmus bereits seit Jahren Forschungsthema in der Arbeitsgruppe in Stuttgart, und es wurden verschiedene Verbesserungen und Erweiterungen des Algorithmus entwickelt [64, 76, 77]. Insbesondere wurde der Algorithmus um die Fähigkeit zur Berechnung von Wechselwirkungen zwischen punktförmigen Dipolen erweitert [70].

Der Projektpartner TU Chemnitz verfügte über Erfahrung bei der Berechnung nicht-äquidistanter schneller Fourier-Transformationen in Systemen mit nicht-, teil- und voll-periodischen Randbedingungen.

Summationsverfahren für teilperiodische Systeme

Die Berechnung von elektrostatischen Wechselwirkungen in 1d- und 2d-quasiperiodischen Teilchensystemen stellt eine größere Herausforderung dar, als die Berechnung in vollperiodischen oder offenen Systemen. Zu Beginn der Projektlaufzeit waren nur wenige, sehr ineffiziente Algorithmen bekannt, die dazu in der Lage sind [109]. In den Jahren vor Projektbeginn wurden beim Projektpartner Universität Stuttgart zwei neuartige Algorithmen dafür entwickelt: MMM2D [61] für 2d- und MMM1D [62] für 1d-quasiperiodische Systeme.

Von diesen Algorithmen existierten bereits parallele, effiziente und jahrelang erprobte Implementierungen in der Simulationssoftware ESPRESSO des Projektpartners Universität Stuttgart [63, 93].

1.4.6 Mehrgitter-Verfahren

Zu Beginn des Vorhabens existierten seitens des Unterprojektpartners Wuppertal in Kooperation mit dem Forschungszentrum Jülich im Rahmen eines Promotionsprojektes entwickelte Mehrgitter-Verfahren, die auf einer Reformulierung des Coulomb-Problems als partielle Differentialgleichung basieren. Die Diskretisierung dieser partiellen Differentialgleichung erfolgt hierbei mittels finiter Differenzen oder finiter Volumina. Die auf der rechten Seite der Gleichung stehenden Singularitäten werden hierbei mittels Näherung durch andere Distributionen ersetzt, wodurch die Divergenzen der Potenzialoberfläche regularisiert werden. Durch Verwendung von Funktionen mit beschränktem Support kann dieser Eingriff später wieder im Nahfeld korrigiert werden.

Neben einem parallelen Mehrgitter-Verfahren für periodische Randbedingungen existierte zudem ein serieller Code für offene Randbedingungen. Die Mehrgitterverfahren *PP3MG* waren als C-Code verfügbar. Den Mehrgittercode galt es zunächst durch eine Anbindung an das Bonner Moleküldynamikpaket zu evaluieren und gegebenenfalls zu optimieren.

Anschließend sollte mit der Implementation einer neuen Mehrgitterbibliothek *VMG* begonnen werden, die nicht nur die parallele Behandlung von periodischen, offenen und quasi-periodischen Strukturen erlaubt, sondern auch die Behandlung von variierenden Dielektrizitätskonstanten gestattet.

Entscheidend ist dabei die Lastbalancierung und Optimierung der Bibliothek, damit sie auf Rechnersystemen mit sehr vielen Kernen effizient arbeitet.

1.4.7 Lokale Gitterverfahren

Das Konzept des Algorithmus „Maxwell Equations Molecular Dynamics“ (MEMD) wurde 2004 von Dünweg und Pasichnyk entwickelt [97], basierend auf der Idee von Maggs [94]. Anstelle eines elektrostatischen Potentials wird hierbei die komplette Elektrodynamik des Systems berechnet. Dabei werden folgende Vereinfachungen vorgenommen:

- Nach dem Vorbild der Car-Parinello-Methode [68] wird die Propagationsgeschwindigkeit der Magnetfelder im System stark reduziert. Solange nach wie vor eine ausreichende Separation von den Teilchengeschwindigkeiten gegeben ist, erhält man identische Resultate im statistischen Mittel.
- Die Zeit wird nicht nur für Teilchen-, sondern auch für Feldbewegungen diskretisiert. Entsprechend ist die Angabe eines Zeitschrittes im Algorithmus notwendig.
- Der Raum wird durch ein Gitter diskretisiert, auf das die Ladungen linear interpoliert werden zur Berechnung der induzierten Felder.

Durch diese Vereinfachungen war es möglich, einen an Geschwindigkeit zu herkömmlichen Methoden konkurrenzfähigen Algorithmus zu entwickeln. Die Vorteile des Algorithmus liegen hierbei vor allem in seiner intrinsischen Lokalität (die Maxwell-Gleichungen sind lokal). Dadurch ist eine sehr effiziente Parallelisierung möglich und es bietet sich die Möglichkeit für lokal veränderliche Eigenschaften des Systems. Der Algorithmus war bei Projektbeginn in einer einfachen Art, mit Fehlern behaftet und ohne lokal veränderliche dielektrische Eigenschaften im Programmpaket ESPRESSO implementiert.

Die „Induced Charge Compensation“ (ICC*, [53]) ist eine Methode, dielektrische Grenzflächen in einem System zu simulieren. Hierbei induziert jede echte Ladung im System auf jeder Grenzfläche Bildladungen, die die dielektrische Verschiebung jenseits der Grenzfläche und ihren Einfluss auf das externe Feld simulieren. Berechnet werden diese Grenzflächenladungen durch ein iteratives Schema und sie bieten die einzige Möglichkeit, dielektrische Grenzflächen in beliebigen Elektrostatik-Algorithmen einzuführen. Durch Implementation dieses Algorithmus wurde die Ausgangssituation geschaffen, um die Korrektheit der dielektrischen Variationen in MEMD überprüfen zu können und für bestimmte Spezialfälle eine alternative Routine bereitzustellen.

1.4.8 Molekulardynamik-Code IMD

IMD [104] ist ein hochskalierbarer, massiv-paralleler Molekulardynamik-Code, welcher von zahlreichen Gruppen weltweit vorwiegend für materialwissenschaftliche Simulationen eingesetzt wird. Design und Datenstrukturen von IMD sind ganz auf Wechselwirkungen kurzer Reichweite ausgelegt, von denen eine reiche Auswahl zur Verfügung steht.

Wechselwirkungen langer Reichweite werden dagegen nicht nativ unterstützt, obschon solche bei oxidischen Materialien durchaus eine wichtige Rolle spielen können. Als Notbehelf für solche Fälle wurde die sogenannte Wolf-Summation [110] implementiert [67], welche die langreichweitige Coulomb-Wechselwirkung durch eine effektive Wechselwirkung kurzer Reichweite ersetzt. Für Systeme mit einer hinreichenden internen Abschirmung funktioniert dies recht gut und lässt sich auch theoretisch begründen. Wie gut die effektive Wechselwirkung der Wolf-Summation die Coulomb-Wechselwirkung tatsächlich reproduziert, ist aber oft unklar, und es bleibt eine Frage der physikalischen Intuition, wann die Wolf-Summation eingesetzt werden kann.

Der Plan war deshalb, die SCAFACOS-Bibliothek an IMD anzubinden, um auch in IMD langreichweitige Coulomb-Wechselwirkungen zur Verfügung zu haben. Damit wird es möglich, auch Systeme zu simulieren, für welche die Wolf-Summation nicht geeignet ist, oder überhaupt zu untersuchen, in welchen Fällen die Wolf-Summation hinreichend genaue Ergebnisse liefert.

Des Weiteren stellt IMD eine ideale Testumgebung dar, mit dem die SCAFACOS-Bibliothek an realen Beispielen und mit einem realen, produktiv eingesetzten MD-Code getestet werden kann. Insbesondere können damit auch erste Erfahrungen mit der Anbindung der Bibliothek an einen MD-Code gesammelt werden.

1.5 Zusammenarbeit mit anderen Stellen

Chemnitz und Stuttgart Eine enge Zusammenarbeit der Projektpartner Chemnitz und Stuttgart ermöglichte neue theoretische Erkenntnisse, welche in Form einer gemeinsamen Publikation [2] öffentlich zur Verfügung gestellt werden.

Chemnitz Die separat veröffentlichten parallelen Softwarebibliotheken der TU Chemnitz führten zu Kooperationen mit verschiedenen Nutzergruppen. So mündete z.B. die Einbindung der parallelen FFT-Module in der OCTOPUS-Softwarebibliothek [60] in einer gemeinsamen Veröffentlichung [4]. Die Kooperation mit einer weiteren Nutzergruppe führte zur Erweiterung des parallelen FFT-Moduls auf allgemeinere Datenaufteilungen und ermöglichte somit den Einsatz der Bibliothek in Simulationen für die Veröffentlichung [69].

Jülich Die SCAFACOS-Bibliothek wurde bereits im Entwicklungsstadium den Teilnehmern der beiden Jülicher Sommerschulen zum Thema “Fast algorithms for long-range interactions” zur Verfügung gestellt. So konnten die Teilnehmer der Heraeus Sommerschule 2010 und CECAM Sommerschule 2011 frühzeitig die Funktionalität bereits vor Veröffentlichung testen. Auf Grund dieser Entwickler–Nutzer Interaktion war es möglich, Wünsche und Anregungen direkt in den Code einfließen zu lassen. Neben den Trainingsmöglichkeiten im Rahmen des Workshops stand die Bibliothek außerdem den Nutzern des BlueGene Skalierungsworkshops 2011 zur Verfügung. Dabei wurde die Bibliothek erstmals vollständig auf 300k CPUs in der TDDFT Anwendung OCTOPUS getestet.

Speziell für die **FMM** wurde auch zur Verbesserung des Datenaustausches die Kommunikationsbibliothek OSPRI entwickelt. Erste Benchmarks mit etablierten Molekulardynamik-Codes wie DL_POLY oder GROMACS zeigten vielversprechende Ergebnisse, so dass für die nahe Zukunft eine direkte Anbindung der SCAFACOS-Bibliothek an beide Codes geplant ist.

Der Barnes-Hut Code **PEPC** bildet auch als Stand-Alone Software die Basis vieler Kooperationen [25]. Dadurch sind Entwicklungen und Verbesserungen am Coulombkernel entstanden, die später in die Bibliotheksversion geflossen sind. Insbesondere sind folgende Zusammenarbeiten zu erwähnen:

- Institut für Energie und Klimaforschung IV (Plasmaphysik), Forschungszentrum Jülich: Anwendung auf Plasma-Wand-Wechselwirkungen in Tokamaks [22, 19].
- Helmholtz Allianz EMMI, Gesellschaft für Schwerionenforschung (GSI) Darmstadt: Transporteigenschaften stark korrelierter Plasmen [46].
- University of Lugano, Institute of Computational Science: Entwicklung eines gitterfreien Vortex-Modells zur Modellierung von Strömungsturbulenzen [33, 36].
- Universität Alberta, Canada: Modellierung der Kelvin-Helmholtz Instabilität an einem Plasma-Vakuum Interface; Vergleich mit Gitter-basierten Rechnungen.
- Institut für Physik, Charles University Prague: Entwicklung eines Baumalgorithmus für magnetisierte Plasmen [31].

Stuttgart In Zusammenarbeit mit Jens Harting (TU Eindhoven, Niederlande und Universität Stuttgart) wurde der SCAFACOS-Code an einen internen Programmcode zur Berechnung von Lattice-Boltzmann-Flüssigkeiten angebunden.

Zu Beginn der Projektlaufzeit waren Dr. Axel Arnold vom Fraunhofer-Institut SCAI in Skt. Augustin und Dr. Olaf Lenz vom Max-Planck-Institut für Polymerforschung in

Mainz in beratender Funktion am Projekt beteiligt. Im Verlauf des Projektes wurde Dr. Arnold zum Juniorprofessor an das Institut für Computerphysik der Universität Stuttgart berufen und konnte seine beratende Tätigkeit damit intensivieren. Dr. Olaf Lenz kam als Postdoc und Nachfolger des verstorbenen Dr. Alexey Neelov als eigener Projektnehmer ans ICP.

Dr. Franz Gähler war als Vertreter des ITAP, Universität Stuttgart, bei der Konzeption und Planung des Projekts beteiligt. Vor dem Projektstart wechselte er an die Universität Bielefeld, führte dort (unbezahlt) die Arbeiten zur Anbindung der Simulationssoftware IMD an die SCAFACOS-Bibliothek fort und stand auch weiter als beratender Mitarbeiter zur Verfügung.

Bonn In Zusammenarbeit mit Dr. Jan Hamaekers (Fraunhofer SCAI) wurde die Bibliothek an die Software TREMOLO-X angepasst.

Cognis (jetzt BASF) ist Industriepartner in diesem Projekt. Hier erfolgte eine Zusammenarbeit mit Dr. Matthias Hloucha zur Simulation von Tensiden in ionenhaltiger, wässriger Lösung.

Weiterhin bestand enger Kontakt mit Dr. Alex Bittner (nanoGUNE, Spanien) zur Validierung der Simulation des Tabakmosaikvirus in Wasser.

2 Eingehende Darstellung des Projekts

2.1 Verwendung der Zuwendung und erzielte Ergebnisse

2.1.1 Entwicklung des gemeinsamen Interfaces

Im Laufe der Entwicklung der Löserbibliothek stellte sich heraus, dass es aufwändiger als ursprünglich angenommen ist, ein gemeinsames Interface zu entwickeln, das eine einheitliche Bedienung aller implementierten Löser ermöglicht. Daher wurde beschlossen, dass die Entwicklung des Interfaces zentral von einem der Projektpartner übernommen wird. Diese Aufgabe fiel dem JSC zu.

WP 0: Interface

Das Interface der SCAFACOS-Bibliothek sollte es ermöglichen, Parameter für verschiedene Methoden auf einheitliche Weise zu setzen. Eine weitere Anforderung an das Interface war eine möglichst große Reduktion der zur Benutzung der Bibliothek notwendigen Befehle. In der derzeitigen Implementierung des Interface werden, bei Verwendung von Standardparametern der Löser, nur sechs Befehle zur Berechnung eines gegebenen Coulomb-Problems benötigt. Diese Standardparameter erlauben es Benutzern der Bibliothek sämtliche Methoden zu verwenden, ohne die Eigenschaften aller methodenspezifischen Parameter im Detail kennen zu müssen.

Das Interface SCAFACOS-Bibliothek ist in C programmiert. Ein Fortranwrapper für die Verwendung der Bibliothek in Fortrancodes wird ebenfalls bereitgestellt. Der Fortranwrapper verwendet dabei die gleichen Funktionsbezeichnungen wie die C-Implementierung. Sowohl in C als auch in Fortran werden Datentypen und Konstanten zur Verfügung gestellt, die zur Benutzung der Bibliothek notwendig sind bzw. diese vereinfachen (z.B. Rückgabewerte von Funktionen oder Parameterwerte für methodenspezifische Parameter).

Bei den Parametern wird zwischen unabhängigen Parametern, die von allen Lösern benötigt werden, und solchen, die spezifisch von einzelnen Lösern benötigt werden, unterschieden. Die unabhängigen Parameter beschreiben die allgemeinen Eigenschaften des zu berechnenden Coulomb-Problems, wie Basisvektoren der Box, in der die Ladungen liegen, oder die Periodizität dieser Box. Spezifische Parameter einzelner Löser beschreiben beispielsweise die Anzahl der verwendeten Knotenpunkte bei Mehrgitterverfahren. Die Parameter werden in einer Struktur gespeichert, die durch sämtliche Benutzungsschritte der Bibliothek hindurch verwendet wird. So wird es ermöglicht, mehrere Berechnungen mit gleichen Parametereinstellungen hintereinander durchzuführen. Des Weiteren können somit innerhalb eines Programms auch mehrere Methoden unabhängig voneinander genutzt werden.

Das Interface ermöglicht es ebenfalls, die verschiedenen Löser auf einfache Weise miteinander zu koppeln oder zu vergleichen. Die Bereitstellung eines Integrators wird hingegen nicht geleistet und war in der Projektplanung auch nicht vorgesehen. Neben dem gemeinsamen Interface enthält die Bibliothek noch einen gemeinsamen Nahfeldlöser, welcher zur Berechnung des Nahfeldanteils in verschiedenen Lösern der SCAFACOS-Bibliothek eingesetzt wird.

2.1.2 WP 1 – Entwicklung und Parallelisierung der Basisbibliotheken

WP 1-1: Parallele schnelle Fourier-Transformation

Implementierung der Methode

Ziel dieses Arbeitspakets war es, eine hochskalierende parallele FFT-Implementation in Form eines Moduls innerhalb und außerhalb der Coulomb-Löser-Bibliothek zur Verfügung zu stellen. Es sollte der Ansatz aus [79] auf seine Skalierbarkeit hin untersucht, parallel implementiert und mit bereits vorhandenen parallelen FFT-Bibliotheken verglichen werden. Dabei sollte das Hauptaugenmerk auf der Ermittlung der FFT-Bibliothek liegen, welche sich am besten für die Parallelisierung der NFFT und der schnellen Summation eignet. Zu Projektbeginn wurde gemäß Arbeitspaket 1.2 eine Evaluation bereits bestehender paralleler FFT-Implementationen durchgeführt. Sie ergab, dass keine der vorhandenen FFT-Bibliotheken alle Anforderungen für die effiziente Parallelisierung der NFFT und der schnellen Summation erfüllt. Eine eingehende Untersuchung der Arbeit [79] ergab, dass auch die approximative FFT nicht für massiv parallele Architekturen geeignet sein würde. Somit war im Hinblick auf die Parallelisierung der NFFT und der schnellen Summation die Implementation einer genügend flexiblen und hochskalierbaren, parallelen FFT-Bibliothek unumgänglich. Die parallele FFT stellt ein wichtiges Kernmodul der Coulomb-Löser-Bibliothek dar, welches auch von anderen Projektpartnern verwen-

det wird. Wir entschieden uns daher, an Stelle der approximativen FFT eine hochskalierende parallele FFT zu entwickeln, welche all Voraussetzungen für die effiziente Parallelisierung der NFFT und der schnellen Summation erfüllt und darüber hinaus für alle Anwender des High-Performance-Computings in Form einer frei zugänglichen Bibliothek zur Verfügung steht. Die entstandene Softwarebibliothek [14] zur Berechnung der parallelen schnellen Fourier-Transformation (PFFT) ist in ihrer Flexibilität einzigartig. Sie wurde stetig weiterentwickelt, getestet und öffentlich zur Verfügung gestellt. Algorithmische Details und Performancetests wurden in [16] publiziert. Das Interface ist an die weit verbreitete FFT-Softwarebibliothek FFTW [84] angelehnt, nutzerfreundlich gestaltet und beinhaltet eine Fortran-Schnittstelle. Die Anpassung der Datenaufteilung an Nutzerbedürfnisse und die Erweiterung der parallelen Implementation auf reelle Eingabedaten erschloss neue Nutzerkreise, wie z.B. durch die Einbindung in der OCTOPUS-Softwarebibliothek [60] belegt wird. Die Anwendung der parallelen FFT innerhalb der OCTOPUS-Bibliothek wird in der gemeinsamen Veröffentlichung [4] beschrieben.

Parallelisierung durch eine Blockzerlegung der Daten

Der parallele FFT-Algorithmus basiert auf einer hochskalierbaren zwei-dimensionalen Datenaufteilung. Die Datenaufteilung erfolgt in Blöcken, welche allerdings nicht für alle Prozesse die gleiche Größe besitzen müssen. Dies ermöglicht eine flexiblere Benutzung und die Berechnung paralleler FFTs beliebiger Größe, d.h. die Gittergröße der FFT muss nicht durch die Anzahl der parallelen Prozesse teilbar sein. Innerhalb der lokalen Datenblöcke werden serielle FFTs entlang der nicht zerteilten, letzten Dimension berechnet. Anschließend werden die Datenblöcke zweimal umverteilt, um die Berechnung der seriellen eindimensionalen FFTs entlang der beiden ersten Dimensionen zu ermöglichen. Die konsequente Erweiterung dieser Parallelisierungsstrategie führt zu d -dimensionalen FFTs mit $(d - 1)$ -dimensionaler Datenaufteilung, welche innerhalb unserer parallelen FFT-Implementation erstmals umgesetzt wurde. Ein Beispiel für die Notwendigkeit dreidimensionaler Datenaufteilung für vier-dimensionale FFTs liefert der Einsatz unserer FFT-Bibliothek für Simulationen in [69].

Für die vereinfachte Anbindung an bestehende Molekülsimulationssoftware wurde ebenfalls eine drei-dimensionale Datenaufteilung für drei-dimensionale FFTs implementiert. Die Daten werden dann innerhalb der Bibliothek auf eine passende zweidimensionale Blockzerlegung umverteilt, während außerhalb der Bibliothek eine für Molekülsimulationen typische dreidimensionale Blockzerlegung verwendet werden kann.

Entwicklung eines Fehlerschätzers

Der entwickelte parallele FFT-Algorithmus basiert auf der Berechnung serieller, eindimensionaler schneller Fourier-Transformationen. Es ergeben sich daher keine Approxi-

mationsfehler und die Entwicklung eines neuen Fehlerschätzer ist nicht notwendig.

Performance Analyse und Optimierung

Die Performance der entwickelten FFT-Implementation wurde mehrfach getestet und optimiert. Für die Bestimmung optimalen Laufzeit wird analog zur FFTW-Softwarebibliothek ein Planungsschritt vor der Berechnung der FFT aufgerufen. Dieser bestimmt in einem einmaligen Optimierungsschritt die beste Kombination aus seriellen FFTs, seriellen Datenumverteilungen und parallelen Datenumverteilungen für die aktuelle Problemgröße und Hardware. Dabei kann die Zeit, welche für die Planung verwendet werden soll, stufenweise durch Flags ausgewählt werden.

Performance- und Skalierungs-Tests gegen Standard-FFT Bibliotheken

Die Skalierbarkeit der neuen PFFT Softwarebibliothek wurde während des Scaling Workshops 2010 auf bis zu 262144 Prozessen einer BlueGene/P bestätigt. Zum Performance Vergleich wurde die Standard-FFT Bibliothek P3DFFT [99] verwendet. Ein Vergleich mit der renommierten FFTW Softwarebibliothek [84] wurde im Zuge eines studentischen Software-Praktikums durchgeführt. Die Ergebnisse wurden durch die Publikationen [12, 13, 16] öffentlich zugänglich gemacht.

WP 1-2: Nichtäquidistante schnelle Fourier-Transformation

Evaluierung geeigneter FFT Implementierungen

Ziel dieses Arbeitspunktes war, vorhandene parallele FFT Standardbibliotheken bezüglich ihrer Nutzbarkeit auf hochskalierbaren Rechnern und ihren Schnittstellen zu vergleichen. Insbesondere muss eine hinreichende Flexibilität der Bibliothek gewährleistet sein, um den vielfältigen Anwendungsmöglichkeiten innerhalb der zu entwickelnden Coulomb-Solver-Bibliothek gerecht zu werden.

Gegenstand der Untersuchungen waren die FFT Bibliotheken von M. Eleftheriou et al. [80] und S. Plimpton et al. [100] sowie die FFTW3-Softwarebibliothek [84] und die P3DFFT-Softwarebibliothek [99]. Um eine direkte Vergleichbarkeit der Bibliotheken zu ermöglichen, wurden die verschiedenen Schnittstellen und die berechneten FFTs für reelle bzw. komplexe Eingabedaten angeglichen. Im Zuge von Laufzeitvergleichen der Bibliotheken stellte sich heraus, dass die parallele Performance der P3DFFT allen anderen Bibliotheken überlegen ist.

Des Weiteren wurden die Anforderungen an die Flexibilität bezüglich der Datenverteilungen und berechenbaren Transformationen mit den Möglichkeiten der FFT Bibliotheken verglichen. Dabei wurden große Unterschiede zwischen den Bibliotheken festgestellt. Die FFTW3-Softwarebibliothek bietet die größte Flexibilität unter den vergliche-

nen parallelen FFT-Bibliotheken. Allerdings erfüllte keine der evaluierten FFT-Implementierungen alle Anforderungen für eine effiziente Parallelisierung der NFFT und schnellen Summation.

Einbindung der PFFT

Für die effiziente Berechnung des Fernfeldes wurde erfolgreich die in Arbeitspaket 1 entwickelte parallele FFT Bibliothek in die parallele NFFT integriert. Dabei war die Flexibilität PFFT-Softwarebibliothek von entscheidender Bedeutung. Nur durch die Verwendung spezialisierter Datenaufteilungen und die effiziente Implementierung von parallelen FFTs mit verkürzten Ein- und Ausgabedaten konnten Lastbalancierung und gleichmäßige Speicheranforderung in der parallelen NFFT erreicht werden.

Parallelisierung der Hin- und Rücktransformation der NFFT

Innerhalb dieses Arbeitspakets wurden Algorithmen zur Berechnung der parallelen nicht-äquidistanten schnellen Fourier-Transformation (PNFFT) entwickelt und innerhalb einer öffentlichen Softwarebibliothek [15] implementiert. Ihre Skalierung wird maßgeblich von der PFFT Bibliothek aus Arbeitspaket 1.1 bestimmt. Details der angewandten parallelen Algorithmen und Performancetests wurden Bestandteil der Publikation [18]. Ein großer Teil dieses Arbeitspunktes wurde bereits durch die Entwicklung einer flexiblen und skalierbaren FFT-Bibliothek vorweggenommen. Die NFFT Hin- und Rücktransformation wurden in Form einer öffentlich verfügbaren Software Bibliothek [15] implementiert. Die Schnittstelle orientiert sich an der bereits weit verbreiteten seriellen NFFT Bibliothek [89]. Damit wird eine einfache Portierung NFFT-basierter serieller Algorithmen auf parallele Architekturen ermöglicht. Das Interface wurde analog zur parallelen FFT sehr flexibel gestaltet, um die Parallelisierung der schnellen Summation problemlos zu ermöglichen. Besonders Augenmerk lag wiederum auf der Datenaufteilung. Die gute Skalierbarkeit der parallelen NFFT wurde in [18] nachgewiesen.

Berücksichtigung von 1d-, 2d-, 3d-periodischen Daten

Eine Erweiterung der schnellen Summation auf 3d-periodische Randbedingungen wurde erfolgreich implementiert und in die SCAFACOS-Bibliothek integriert. Eine Zusammenfassung des Algorithmus wurde in [17] publiziert. In Zusammenarbeit der Projektpartner Stuttgart und Chemnitz wurden die theoretischen Zusammenhänge zwischen der 3d-periodischen schnellen Summation und der P3M Methode erarbeitet. Die Ergebnisse werden in Form einer gemeinsamen Publikation [2] öffentlich zur Verfügung gestellt. Für die Behandlung von 1d- und 2d-periodischen Randbedingungen wurden innerhalb der Diplomarbeit [10] entscheidende Fortschritte erzielt.

Schnelle Summation für verschiedene Kerne

Die NFFT-basierte schnelle Summation wurde erfolgreich implementiert und an das Interface der SCAFACOS-Bibliothek angebunden. Sie folgt einem sehr modularen Aufbau, so dass durch Austausch einzelner Bausteine viele verschiedene Variationen der schnellen Summation verglichen werden können. Dies beinhaltet unter anderem auch die Berechnung verschiedener Kernfunktionen. Zur Berechnung des Nahfeldes wurde das Modul für kurzreichweitige Wechselwirkungen (WP 2-6) erfolgreich angebunden. Für die effiziente Berechnung des Fernfeldes wurde die in Arbeitspunkt 3 entwickelte parallele NFFT Softwarebibliothek verwendet. Durch die Anbindung des parallelen Sortiermoduls aus Arbeitspaket 1.3 ist es gelungen die technischen Details der parallelen Datenaufteilung vor dem Nutzer der Bibliothek zu verbergen.

WP 1-3: Parallele Sortierung

Anforderungsanalyse anhand verschiedener Coulomb-Löser

Ziel dieses Arbeitspunktes war es, die verschiedenen Anforderungen an parallele Sortier- und Datenumverteilungsverfahren zu analysieren. Hierfür wurden die in der Bibliothek implementierten Löser FMM, P2NFFT, P3M, PEPC und PP3MG sowie zusätzlich die parallelen Simulationsanwendungen ESPRESSO, IMD, LAMMPS und NAMD untersucht. Innerhalb der Löser haben sich parallele Sortierverfahren hierbei als wesentlich für die effiziente Organisation der Partikeldaten für die anschließende Verarbeitung herausgestellt. Dies dient der Erhöhung der Lokalität von Speicherzugriffen und ist ein integraler Bestandteil der eingesetzten Lastbalancierungsverfahren. Als zusätzliche Anforderung hat sich insbesondere die Speichereffizienz der eingesetzten Verfahren und die Berücksichtigung von individuellen Partikelgewichten zur Lastbalancierung herausgestellt. Als weitere wichtige Aspekte haben sich der Umfang der Partikeldaten und das verwendete Abspeicherungsformat ergeben. Neben verschiedenartigen Schlüsselwerten für die Sortierung und gängigen Partikeldaten, wie Positionen, Ladungen oder Massen, müssen oft noch weitere Eigenschaften, wie Geschwindigkeiten oder Kräfte, als Teil der Partikeldaten mit sortiert werden. Die ermittelten Abspeicherungsformate für Partikeldaten gliedern sich in (1) ein einzelnes Array mit strukturierten Datentypen, (2) separate Arrays für jede Eigenschaft der Partikel (Position, Ladung, Masse, ...) und (3) mehrere separate Arrays für einzelne dimensionsbehaftete Partikeleigenschaften (Position, Geschwindigkeit, ...). Ergebnisse der Anforderungsanalysen wurden am Beispiel der Löser PEPC und FMM in [5] veröffentlicht. Darüber hinaus wurde speziell für Partikelsimulationen eine neue kollektive MPI-Kommunikationsoperation vorgestellt, mit der eine feingranulare Datenumverteilung realisiert wird und einzelne Partikel gezielt zwischen Prozessoren ausgetauscht werden können [7].

Entwicklung neuartiger, hochskalierender Sortierverfahren

Durch die Kombination verschiedener paralleler Sortierstrategien sollten neue hochskalierende Verfahren entwickelt werden. Hierfür wurde eine Kategorisierung existierender paralleler Sortierverfahren durchgeführt, gemeinsame Bestandteile in verschiedenen Sortierverfahren identifiziert und generisch implementiert, so dass sie je nach Situation und Einsatzgebiet flexibel kombinierbar sind. Detaillierte Ergebnisse dieser Untersuchungen sind in [6] dargestellt. In Zusammenarbeit mit dem Projektpartner im Forschungszentrum Jülich wurde für den Löser PEPC ein neues paralleles Sortierverfahren entwickelt. Kern des Verfahrens ist ein paralleler Partitionierungsalgorithmus, welcher lokal sortierte Folgen von Elementen so aufteilt, dass daraus eine globale Sortierung der Elemente resultiert [9]. Durch die Angabe individueller Workloads pro Prozessor sind beliebige homogene und heterogene Zielverteilungen definierbar, die sowohl hinsichtlich Anzahl als auch Gewichtung der Partikel spezifiziert werden können. Zur Verbesserung des Laufzeitverhaltens bei hohen Prozessoranzahlen wurde eine mehrstufige Realisierung der Partitionierung entwickelt. Diese zerlegt das Partitionierungsproblem nacheinander in mehrere Teilprobleme, die unabhängig voneinander auf kleineren Prozessorgruppen gelöst werden können. Laufzeitmessungen haben gezeigt, dass sich die Skalierbarkeit der parallelen Sortierung dadurch deutlich verbessern kann. Insgesamt konnte der parallele Sortierschritt in PEPC insbesondere für hohe Prozessorzahlen deutlich verbessert und somit auch die Gesamtlaufzeit von PEPC verringert werden. Dies wurde sowohl durch die Laufzeitverbesserung der parallelen Sortierung als auch durch eine gleichmäßigere Lastverteilung innerhalb der parallelen Berechnungen von PEPC erreicht [5]. Im weiteren Projektverlauf wurde das neu entwickelte parallele Sortierverfahren auch für die Sortierung gewichteter Partikeldaten im Löser FMM eingesetzt. Weiterhin wurde ein In-place-Algorithmus zur effizienten Datenumverteilung entwickelt, der den Speicherverbrauch für die Durchführung von Datenumverteilungen deutlich reduziert [8]. Der Algorithmus wurde so implementiert, dass er in Form des Interfaces der kollektiven Datenumverteilungsoperation `MPI_Alltoallv` genutzt werden kann. Im Hinblick auf hochskalierbare Plattformen wurde ebenfalls der Speicherverbrauch in Abhängigkeit von der Prozessoranzahl berücksichtigt, um auch hierbei eine hohe Skalierbarkeit zu gewährleisten. Innerhalb der SCAFACOS-Bibliothek wird die neu entwickelte In-place-Datenumverteilung dazu verwendet, parallele Sortierverfahren, wie das vorgestellte parallele Sortieren basierend auf Partitionieren, bei Bedarf auch speichereffizient, d.h. mit nur geringem zusätzlichem Speicherbedarf, durchführen zu können.

Implementierungsstrategien für hochskalierende parallele Plattformen

Die Auswahl und Realisierung der parallelen Sortierverfahren sollte speziell im Hinblick auf die Eigenschaften moderner hochskalierender Plattformen durchgeführt wer-

den. Für die entwickelte mehrstufige Partitionierung wurden Implementierungsstrategien realisiert und untersucht, welche gezielt im Hinblick auf die vorhandene Mesh- bzw. Torus-Topologie des Kommunikationsnetzwerkes der Blue Gene/P Plattform von Vorteil sind. Hierdurch konnten jedoch keine weiteren Laufzeitverbesserungen innerhalb des Partitionierungsalgorithmus erreicht werden. Im Zuge der Integration der parallelen Sortierverfahren in die Löser PEPC und FMM wurden die Funktionsfähigkeit und das Laufzeitverhalten auf der Blue Gene/P Plattform überprüft, wobei verschiedene Anpassungen notwendig waren. Dies betraf im wesentlichen die Hauptspeichernutzung in Abhängigkeit von der Anzahl verwendeter Prozessoren bzw. Kerne sowie die eingesetzten Kommunikationsoperationen. Mit zufällig generierten Daten wurde die Funktionsfähigkeit der parallelen Sortierverfahren bis ca. 128.000 Prozessorkerne und innerhalb der Löser PEPC und FMM von den Projektpartnern im Forschungszentrum Jülich auf dem gesamten JUGENE System getestet. Weiterer Bedarf für gezielte Optimierungsstrategien ergab sich bei den Alltoall-Datenumverteilungsoperationen innerhalb der MPI-Implementierung der Blue Gene/P Plattform. So konnte der oft benötigte Austausch einzelner Integerwerte mit einem geänderten Algorithmus gegenüber der vorhandenen `MPI_Alltoall`-Operation bei hohen Prozessoranzahlen signifikant verbessert werden. Darüber hinaus konnte durch das gepackte Versenden verschiedener Partikeldaten mit einer einzelnen `MPI_Alltoallv`-Operation (anstatt mehrere) insbesondere im Falle weniger Partikel pro Prozessor eine deutliche Laufzeitverringerung erzielt werden.

Datenmanagement und Sortierstrategien für unterschiedliche Datenspezifika

Spezifischen Anforderungen an parallele Sortierverfahren beim Einsatz in Partikelsimulationen wurden insbesondere hinsichtlich der mit umzusortierenden Datenelemente, der Laufzeitkomplexität, dem Speicherverbrauch, der Verteilung der Eingabe- und Ausgabedaten, anwendungsspezifischer Gewichte, charakteristischer Schlüsselwerte und vorhandener Vorsortierungen identifiziert [6]. Diese boten zumeist auch Ansatzpunkte zur Optimierung, welche dann gezielt für einzelne Sortierverfahren und Löser ausgenutzt wurden. Hervorzuheben sind insbesondere Anpassungen hinsichtlich des Speicherverbrauchs für die parallele Sortierung der Partikeldaten innerhalb des Löser FMM. So wird in dessen parallele Partikelsortierung nur noch Speicher verwendet, welcher von der FMM Implementierung selbst bereitgestellt wird und es werden ebenfalls die verschiedenen Kompressionsmethoden zur Reduzierung des Speicherverbrauchs der FMM unterstützt. Im Ergebnis lassen sich damit insbesondere auch auf Plattformen mit relativ begrenztem Hauptspeicher deutlich größere Partikelsysteme berechnen als zuvor. In Zusammenarbeit mit den Projektpartnern aus Chemnitz und Stuttgart wurde die parallele Sortierung für die gitterbasierten Löser P2NFFT, P3M und MEMD realisiert. Dies umfasst die Sortierung der Partikel anhand ihrer dreidimensionalen Koordinaten und die Verteilung der

Partikeldaten auf ein jeweils bereitgestelltes Prozessgitter. Grundlage hierfür ist die neu entwickelte, feingranulare Datenumverteilungsoperation [7]. Nach der Erweiterung um eine Duplizierung im Zuge der Datenumverteilung wurde mit der feingranularen Datenumverteilungsoperation auch die Erzeugung von Geisterpartikeln für den bibliotheksinternen Nahfeldlöser umgesetzt. Weiterhin besteht auf Grund des Bibliotheksansatzes für alle Löser der SCAFACOS-Bibliothek die Notwendigkeit, die berechneten Ergebnisse wieder in die ursprüngliche Reihenfolge der Eingabedaten zurückzusortieren. Hierfür wurde ebenfalls ein einheitliches Sortierverfahren entwickelt, welches von mehreren Lösern genutzt wird.

Evaluation anhand der Bibliothek schneller Coulomb-Löser

Zur Evaluation der gesamten SCAFACOS-Bibliothek wurde zusammen mit dem Projektpartner aus Stuttgart ein generisches Testprogramm entwickelt, welches die zu berechnenden Partikelsysteme über ein XML-Format einliest und über das Bibliotheksinterface mit dem jeweils ausgewählten Löser berechnet. Zur Durchführung von Benchmarkläufen mit hochskalierbaren Plattformen wurden dabei verschiedene Generatoren zur automatischen Erzeugung großer Partikelsysteme integriert. Das Basismodul zur parallelen Sortierung wird innerhalb des generischen Testprogramms genutzt, um verschiedene Ausgangsverteilungen der Partikeldaten bereitzustellen und deren Einfluss auf die Laufzeiten der parallelen Löser zu untersuchen. Insgesamt wurden mit dem generischen Testprogramm ein Großteil der Benchmarkrechnungen innerhalb des Projekts durchgeführt. Darüber hinaus wurde es für vielfältige Arbeiten zur Optimierung und Fehlerbehebung in enger Kooperation zwischen den verschiedenen Partnern des Projekts eingesetzt.

Automatisierte und interaktive Softwareumgebung

Zur weitgehenden Automatisierung von Anpassungen der parallelen Sortierung und der SCAFACOS-Bibliothek insgesamt an die jeweilige parallel Plattform, sollten Verfahren zur Selbstadaptierung und zum automatischen Tuning entwickelt werden. Hierfür wurden zum einen verschiedene Benchmarkprogramme innerhalb eines separaten Tuning-Schritts der Sortierbibliothek realisiert, um plattformabhängige und performance-kritische Parameter automatisch zu ermitteln. Zum anderen wurde in Zusammenarbeit mit dem Projektpartner im Forschungszentrum Jülich eine integrierte Performancevorhersage für den Löser FMM entwickelt, mit dem die sowohl vom Partikelsystem als auch von der Hardwareplattform abhängige Optimierung des Verhältnisses zwischen Nah- und Fernfeldberechnungen automatisiert werden kann [3]. Darüber hinaus wurden wesentliche Teile des Buildsystems der SCAFACOS-Bibliothek inklusive der Integration zusätzlicher Softwarepakete (z.B. FFTW, ARCFI, OSPRI) beigetragen. Hierdurch wird eine weit-

gehend automatisierte Übersetzung und somit eine benutzerfreundliche Verwendung der SCAFACOS-Bibliothek auf unterschiedlichsten parallelen Plattformen ermöglicht.

2.1.3 WP 2 – Entwicklung und Parallelisierung der Teilmodule

WP 2-1: Die schnelle Multipol Methode

Der algorithmische Aufbau der FMM setzt sich aus fünf voneinander unabhängigen Teilen zusammen. Für die Parallelisierung ergeben sich hiermit verschiedene Herausforderungen. Um eine hohe Effizienz und Skalierbarkeit auf mehr als hunderttausend Prozessoren zu erreichen, wurde großer Wert auf eine separate Modellierung der einzelnen Teile (Pass 1 bis Pass 5) gelegt.

Der Parallelisierungsansatz nutzt neben MPI für die globale Kommunikation (`all-reduce`, `allgather`) und Synchronisation (`barrier`) auch zusätzliche Softwarebibliotheken zur Punkt-zu-Punkt Kommunikation. Die Nutzung von ARMCI [96] ermöglicht es, einseitige `put` Send-Operationen zwischen zwei Prozessoren nicht-blockierend durchzuführen. Für die Verwendung der SCAFACOS-Bibliothek auf der Bluegene Architekturen wurde aus Effizienzgründen für Punkt-zu-Punkt Kommunikation die OSPRI [85] Bibliothek gewählt. Für den gesamten Datentransfer zwischen Prozessoren wurden ausschließlich die erwähnten Kommunikationsfunktionen erlaubt, um auf modernen Rechnerarchitekturen die höchste Bandbreite und geringste Latenz für das Senden von verschieden großen Nachrichten (Daten) zu erhalten. Weiterhin ermöglicht dieser Ansatz die gezielte Kontrolle des verwendeten Speichers in der Kommunikationsbibliothek.

Da bereits für die sequenzielle FMM eine raumfüllende Morton-Kurve erfolgreich für die Baumverwaltung im Einsatz ist, wurde dieses Konzept in die parallele Version übertragen. Die Abbildung der 3d-verteilten Teilchen auf eine eindimensionale Kurve erlaubt der FMM eine einfache Aufteilung der Teilchen auf die Prozessoren. Auf die Nutzung von kartesischen Kommunikatoren und auf Beschränkung der Prozessorzahlen $P = 2^n$, $n \geq 0$ kann somit vollständig verzichtet werden. Dieses Schema ist zwar wesentlich aufwendiger zu implementieren, vermeidet aber automatisch Lastungleichheiten bei homogenen Systemen und erlaubt eine präzise Lastumverteilung bei geclusterten Teilchensystemen.

Durch die derzeit begrenzten Speicher-Ressourcen von Bluegene-Architekturen wurde bei der Parallelisierung darauf geachtet Datenstrukturen, die mit der Anzahl der Prozessoren skalieren zu vermeiden. Dadurch ist es möglich, den begrenzten Speicher effizienter zu nutzen. Um den durch die FMM benutzten Speicher weiter zu reduzieren, wurden rechenzeitunkritische Datenstrukturen und Felder komprimiert gespeichert. Damit ist es möglich auch sehr große Teilchenzahlen für eine Simulation zu nutzen. Der durch die

Kompression entstandene Mehraufwand (Packen/Entpacken) beträgt weniger als 5% der Gesamtrechenzeit.

Im folgenden werden die fünf unabhängigen Teile der FMM mit der jeweiligen Parallelisierungsstrategie beschrieben:

Parallelisierung von Pass 1 (Translation der Multipolentwicklungen)

Um Wechselwirkungsbeiträge im Fernfeld zu berechnen (Pass 2), müssen die Teilcheninformationen (Position, Ladung) als Multipole dargestellt werden. Mit Hilfe der parallelen Sortierung werden zuerst alle Teilchen Boxen auf einem bestimmten Level im FMM Baum zugeordnet. Diese Zuordnung erlaubt nun eine einfache Erzeugung der Multipol-Informationen. Dafür wird jedes Teilchen im Mittelpunkt der zugehörigen Box entwickelt. Dieser Schritt beinhaltet keine Kommunikation und skaliert – gleichmäßige Datenverteilung vorausgesetzt – ideal mit der Anzahl der Prozessoren. Aus Effizienzgründen wird nun jede Box einem Prozessor zugeordnet. D.h. in diesem Schritt findet Kommunikation zwischen benachbarten Prozessoren statt und Multipolentwicklungen für geteilte Boxen werden ausgetauscht. Anschließend werden diese Multipole vom untersten Level bis auf erste Level transferiert auf dem Wechselwirkungen stattfinden können. Dieser Schritt enthält sowohl lokale Anteile ohne Kommunikation als auch Anteile die durch Punkt zu Punkt Kommunikation via `put` zwischen den Prozessoren verteilt werden. Der Nachrichtenaustausch findet dabei über eine eigens geschaffene Sendepufferstruktur statt. Damit ist sichergestellt, dass für diese Operationen nur ein kleiner zusätzlicher Speicher verwendet wird. Der Transfer im Baum findet dabei wie folgt statt:

- Anlegen von Empfangspuffern
- Ermitteln der Zielprozessoren für die `put` Operation
- Nicht-blockierendes Senden mittels `put`
- Berechnung der lokalen Verschiebe-Operationen im Baum
- Abschließen der `put` Operationen
- Übernehmen der empfangenen Daten in die lokalen Datenstrukturen
- Löschen der Empfangspuffer

Durch den nicht-blockierenden Datentransfer findet eine maximale Überlagerung zwischen lokaler Berechnung und Kommunikation statt.

Parallelisierung von Pass 2 (Fernfeld-Wechselwirkung)

Die Berechnung der Wechselwirkungen im Fernfeld trägt fast 50% zur Gesamtrechenzeit der FMM bei. Zusätzlich ist in diesem Pass der Speicherbedarf der FMM maximal. Aus diesen beiden Gründen ist es besonders wichtig, eine geeignete Parallelisierungsstrategie umzusetzen. Da sich bereits aus Pass 1 eine Überlagerung von Rechnung und

Kommunikation als erfolgreich erwiesen hat, wurde dieses Schema auch hier beibehalten. Durch den bereits hohen Speicherbedarf im sequenziellen Code musste aber darauf geachtet werden, die Sende- und Empfangspuffer möglichst klein zu halten. Aus diesem Grund wurden die Sende- und Empfangspuffer für jedes Level separat alloziert. Da die zu sendenden Daten verteilt im Speicher liegen, werden diese vor der Sendeoperation kompakt in einen Sendepuffer geschrieben. Die Parallelisierung erfolgt level-unabhängig und kann wie folgt beschrieben werden:

- Ermitteln der möglichen Sendeprozessoren auf Empfängerseite
- Ermitteln der möglichen Empfängerprozessoren auf Senderseite
- Ermitteln der zu sendenden Datenpaketgröße pro Prozessor
- Anlegen der Empfangspuffer in der entsprechenden Größe
- Schreiben der zusammenhängenden Sendepuffer (Multipolmomente)
- Nicht-blockierendes Senden mittels `put`
- Berechnung der lokalen Fernfeld-Wechselwirkungen
- Abschließen der `put` Operationen
- Berechnung der verbliebenen Fernfeld-Wechselwirkungen mit Hilfe der empfangenen Multipolmomente
- Zurücksenden der berechneten Taylorentwicklungen
- Übernehmen der empfangenen Daten in die lokalen Datenstrukturen
- Löschen der Empfangspuffer

Durch das hier beschriebene Schema ist es möglich, den Speichermehrbedarf für die parallele Version der FMM auf 14% zu begrenzen.

Parallelisierung von Pass 3 (Translation der Taylorentwicklungen)

Die Parallelisierung der Translation der Taylorentwicklung im Baum erfolgt nach den gleichen Prinzipien, die bereits für Pass 1 angewendet wurden, allerdings in umgekehrter Reihenfolge. Im Gegensatz zur Translation der Multipole werden hierbei die erzeugten Taylorentwicklungen aus Pass 2 nun den FMM-Baum bis auf das tiefste Level verschoben. Die Kommunikation findet ausschließlich über Punkt-zu-Punkt Kommunikation via `put` statt. Die Zuordnung auf die Teilchen erfolgt wieder vollständig lokal. Allerdings muss, wie bereits im Pass 1, die Information evtl. noch auf geteilte Boxen auf Nachbarprozessoren mit einem abschließenden Kommunikationsschritt verteilt werden.

Parallelisierung von Pass 4 (Berechnung des Fernfeldanteils)

Die Berechnung des Fernfeldanteils (Potentiale, Kräfte) kann durch die vorgestellte Parallelisierung von Pass 3 vollständig lokal, ohne Punkt-zu-Punkt Kommunikation stattfinden. Lediglich die Berechnung der Gesamtenergie erfordert eine global Reduktion mittels `allreduce`.

Parallelisierung von Pass 5 (Berechnung des Nahfeldanteils)

Die Berechnung der direkten Wechselwirkungen nimmt ebenfalls fast 50% der Gesamt-Rechenzeit in Anspruch.

Der Ablauf der einzelnen Kommunikations- und Rechenschritte findet analog zu dem in Pass 2 beschriebenen Schema statt. Im Gegensatz zur Fernfeld-Wechselwirkung findet die Nahfeld-Wechselwirkung nur auf dem untersten Baumlevel statt. Allerdings müssen hier für jede Box-Box Wechselwirkung Teilcheninformationen statt Multipol-Informationen übertragen werden. Die Datenmenge kann daher bei geclusterten Verteilungen starke Unterschiede zwischen den Prozessoren aufweisen. Eine Überlagerung von Rechnung und Kommunikation stellt auch hier sicher, dass die Effizienz erhalten bleibt.

Loadbalancing der Passes

Für die Lastbalancierung der FMM wurden nur die zwei rechenzeitaufwendigsten Teile (Pass 2 und Pass 5) betrachtet. Diese tragen ca. 95% zur Gesamt-Rechenzeit bei. Die Lastbalancierung beider Teile ist ähnlich, erfordert aber unter Umständen eine erneute Umsortierung von Teilen der Simulationsdaten (Teilchen oder Multipole). Eine zusätzliche Lastbalancierung der Verschiebung von Multipolmomenten und Taylorkoeffizienten im Baum (Pass 1 und Pass 3) ist nicht nötig und wird bereits in ausreichendem Maße durch die Balancierung des Fernfeldes (Pass 2) erreicht. In den meisten untersuchten Fällen ist sogar nur der Lastausgleich im Nahfeld (Pass 5) ausreichend, um die Effizienz zu verbessern.

Durch die Nutzung der raumfüllenden Kurve wurde ein Kostenmodell (Wechselwirkungsliste) für jedes einzelne Teilchen aufgestellt. Mit Hilfe der parallelen Sortierung und Partitionierung können dann die Gesamtkosten im System gleichmäßig auf die Prozessoren verteilt werden. Für geclusterte Systeme konnte damit eine Reduzierung der Rechenzeit um bis zu 50% erreicht werden.

Entwicklung eines Fehlerabschätzers für 1d- und 2d-periodische Systeme

Bereits die sequenzielle Version der FMM enthielt die Möglichkeit, periodische Systeme in ein und zwei Dimensionen zu berechnen. Im Rahmen dieses Projektes wurde auch der Fehlerschätzer auf 1d- und 2d-periodische Systeme erweitert. Dabei mussten neben den zusätzlich auftretenden Fehlertermen aus dem periodischen Gitter auch die Fehlerterme zwischen Gitter und Simulationsbox einbezogen werden. Der einheitliche Ansatz für alle Periodizitäten (Renormalisierung) kommt ohne die Nutzung FFT-basierter Methoden aus, erhält die lineare Komplexität $\mathcal{O}(N)$, benötigt keine zusätzlichen Parameter und hat nur einen geringen Einfluss auf die Rechenzeit. Die so vorberechneten Fehlerterme wurden wie auch für offene und 3d-periodische Systeme als Look-up Tabelle für

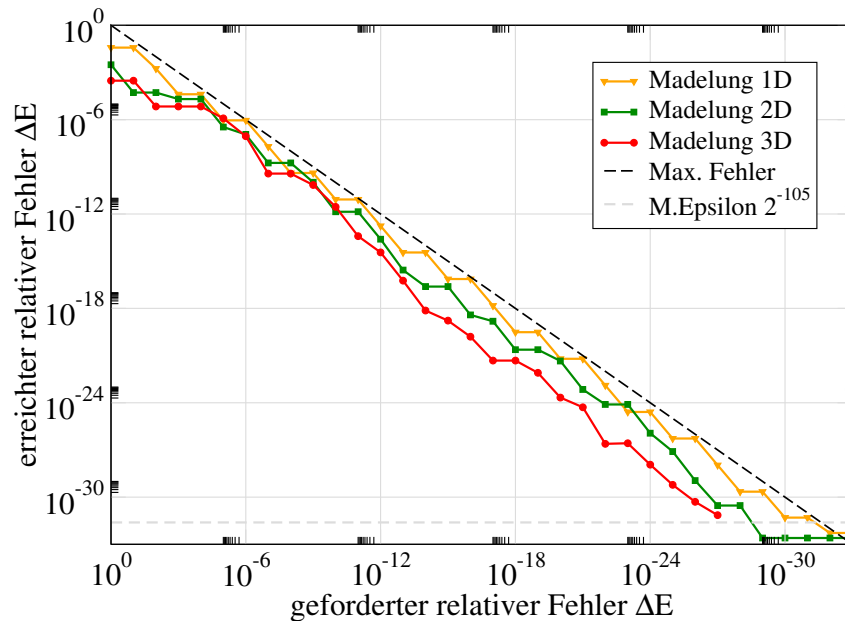


Abbildung 2.1: FMM Genauigkeit für 1d, 2d und 3d periodische Systeme

die Nutzung in der FMM abgespeichert. Damit ist es jetzt möglich, die Genauigkeit mit nur einem physikalischen Parameter im Bereich $\Delta E_{\text{rel}} = 10^{-1} \dots 10^{-15}$ auch für teilperiodische Systeme zu steuern.

Aufstellung eines Minimierungsproblems zur Optimierung der Laufzeit unter Berücksichtigung des Fehlers

Neben der a-priori Fehlerkontrolle ist die Laufzeitminimierung ein wichtiger Schlüsselpunkt für die Anwendung schneller Coulomb-Löser. Speziell für parallele Anwendungen mit Millionen von Zeitschritten ist es von hoher Bedeutung, eine effiziente Lösung für beide Probleme zu finden, ohne die eigentliche Simulation zu beeinträchtigen. Daher wurde im Rahmen des Projektes ein analytischer Fehlerschätzer und Laufzeitminimierer für homogene Teilchensystem entworfen und implementiert.

Neben dem vorhandenen zweistufigen Fehlerschätzer für beliebige Teilchensysteme wird die analytische Berechnung entweder vorgeschaltet oder ersetzt den komplexen Fehlerschätzer vollständig, falls es sich um ein homogenes System handelt.

Damit erhält man die Möglichkeit, für einen geforderten Fehler die Laufzeit zu minimieren, ohne dass die Laufzeitminimierung signifikant zur Gesamtlaufzeit beiträgt. Die Funktionalität, Stabilität und Genauigkeit wurde anhand einer Rechnung mit über 3 Billionen Teilchen gezeigt [88].

WP 2-2: Parallele Baum Algorithmen

Skalierbare Sortierung Wie bereits in WP 1-3 beschrieben, wurde am Anfang des Projekts eine intensive Zusammenarbeit mit der Chemnitzer Gruppe geführt, um einen Engpass in der Teilchensortierung aufzugreifen. Mit der Implementierung dieses neuen Sortierers in PEPC konnte eine signifikante Steigerung der Effizienz erzielt werden [35]. Darüber hinaus bietet der neue Sortierer die Möglichkeit, eine vorgegebene Lastbalance unter Berücksichtigung der vorhandenen Speicherressourcen einzuhalten.

Gebietszerlegung Zur Erhaltung einer möglichst idealen Lokalität der jeweils zu einem Prozessor gehörigen Teilchen wurde bisher die Morton-Kurve als Projektion der dreidimensionalen Koordinaten in den eindimensionalen Schlüsselraum genutzt. Tatsächlich enthält sie aber beliebig große Sprünge. Diese resultieren in teilweise fragmentierten Prozessordomains, was zu einer großen *branch*-Zahl sowie großem Kommunikationsbedarf führen kann, da die Oberfläche der Domains, die Sprünge in der raumfüllenden Kurve enthalten im Verhältnis zu ihrem Volumen unideal groß ist. Im Rahmen einer Master-Arbeit [29] wurden alternative Kurven (u.a. HILBERT) untersucht. Sie versprechen gegenüber der *Z*-Kurve noch bessere Lokalitätseigenschaften.

Optimierung der Baumstruktur Während des Projekts wurden umfangreiche Verbesserungen der Baumstruktur vorgenommen, vor allem um den Speicherbedarf der Multipolinformation zu minimieren:

Einen wesentlichen Teil des Parallelisierungskonzeptes von PEPC bilden die sogenannten *branch*-Knoten. Diese Knoten kennzeichnen Gebiete im globalen Baum, welche ausschließlich von einem MPI Prozess verwaltet werden. Mit Hilfe der *branches* werden prozessübergreifende Informationen ausgetauscht, wobei die *branches* aller Prozesse auf allen Prozessen gespeichert werden. Im Allgemeinen steigt die Anzahl dieser Knoten mit der Anzahl der verwendeten Prozesse. Daraus resultiert zum einen eine kollektive *all-to-all* Kommunikationsstruktur mit wachsendem Volumen und zum anderen ein steigender Speicherplatzbedarf, welcher a priori nicht abgeschätzt werden kann [34].

Um den Bedarf an Speicherplatz zu reduzieren, welcher bei gegebener Teilchenzahl letztendlich die maximale Anzahl an verwendbaren Prozessen limitiert, wurde eine Reduktion der benötigten bzw. gespeicherten *branch*-Knoten entwickelt. Der dabei verwendete Filterungsmechanismus arbeitet auf folgende Weise: Nachdem die *branch*-Knoten verschickt werden, wird für jeden dieser Knoten anhand einer geometrischen Abschätzung bestimmt, ob er auf der Interaktionsliste stehen wird und falls nicht, welcher *fill*-Knoten auf gröberen Stufen seine Multipolinformationen benötigt. Dabei richtet sich die Abschätzung nach dem *multipole acceptance criterion* (MAC). Es wird durch ein verschärftes Kriterium sichergestellt, dass ein abgelehnter Knoten nicht auf die Wechselwir-

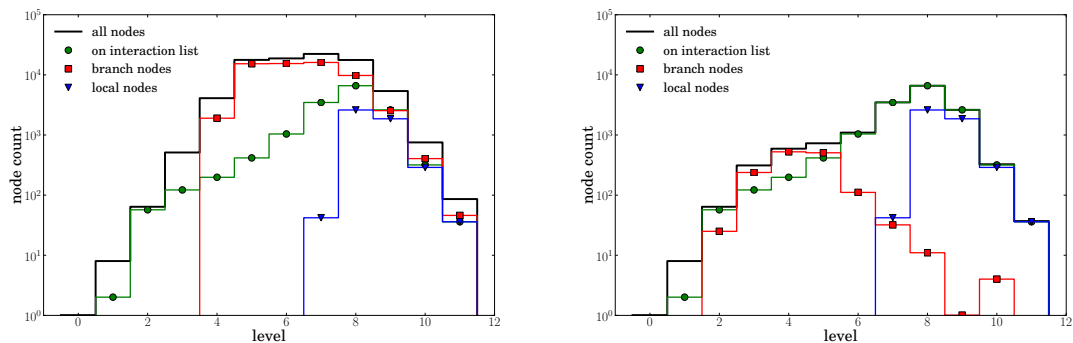


Abbildung 2.2: Anzahl der Knoten auf jeder Stufe, dargestellt für einen MPI Prozess, Nummer 1000 von 2048, ohne den *branch*-Filter (links) und mit eingeschaltetem Filter (rechts). Unterschieden wird zwischen allen Knoten (schwarz), Knoten auf der Wechselwirkungsliste (grün), *branch*-Knoten (rot) und lokalen Blättern (blau).

kungsliste kommen kann. Durch die Verschärfung wird die Effektivität der Filterung nur minimal reduziert [32].

In einem typischen Lauf mit zehn Millionen Teilchen auf 2048 Rechenkernen ergibt sich eine Reduktion der gespeicherten *branches* von etwa 60000 auf 1200. Die Effektivität und Ausmaße der Filterung sind in Abbildung 2.2 visualisiert. Sichtbar wird, dass nun im Wesentlichen die Wechselwirkungsliste die Anzahl der gesamten gespeicherten Knoten dominiert. Einzig auf den kleineren Stufen, hier 3 und 4, gibt es Knoten, welche nicht explizit benötigt werden; deren Anzahl ist jedoch vernachlässigbar.

Eine weitere Optimierung der *branch* Nodes, basierend auf den bereits nach der Gebietszerlegung auf jedem MPI Rank lokal verfügbaren Schlüsselgrenzen, wurde implementiert und verifiziert [29]. Hierzu wurde einerseits das *branch*-Level einstellbar implementiert und dabei weiter zu kleinerem Level, also in Richtung der Wurzel des Baumes verschoben werden um ihre Anzahl weiter zu reduzieren. Andererseits wurden Informationen für *branches*, die aufgrund geometrischer Abschätzungen für die Kraftberechnung auf höherem Level definitiv benötigt werden, direkt angefordert um latenzbedingte Wartezeiten während des Baumdurchlaufes zu vermeiden.

Implementierung neuer Tree-Walk Algorithmen: Hybride Parallelisierung mit POSIX Threads

Zur weiteren Optimierung der parallelen Effizienz des Codes wurde für den Baumdurchlauf, der die tatsächliche Kraftberechnung durchführt und damit der zeitintensivste Teil des Codes ist, eine hybride Parallelisierung implementiert [46]. Im ursprünglichen Ansatz zur Parallelisierung wurde jedem Prozessor ein MPI-Rank und

damit eigene Verwaltungsstrukturen, die ausserordentlich viel Speicher belegen, zugewiesen. Mit steigender Prozessorzahl und damit sinkender Teilchenzahl je Prozessor ist der Anteil der Verwaltungsstrukturen am Gesamtspeicherverbrauch pro Teilchen folglich der beherrschende Anteil.

Der native Ansatz, um den resultierenden Speicherengpass zu umgehen ist also, die Anzahl der MPI-Ranks möglichst klein zu halten. Dies wurde erreicht, indem beispielsweise auf der Blue Gene/P ein MPI-Rank pro Node, der jeweils vier Prozessoren enthält, genutzt wird. Die vier Prozessoren eines Nodes teilen sich gemeinsamen Speicher, können sich also auch den Großteil ihrer Verwaltungsstrukturen teilen. Der Baumdurchlauf wird dann für die Teilchen eines Nodes von den vier Prozessoren parallel durchgeführt. Da zukünftige Supercomputer eine noch deutlich größere Prozessorzahl je Node aufweisen werden (z.B. Blue Gene/Q: 16 Cores pro Node), ist dieser Ansatz auch mit dem Blick auf die Skalierung auf kommenden Generationen von High-Performance-Computer vielversprechend.

An Stelle des sehr einfach nutzbaren OpenMP Frameworks für die Parallelisierung innerhalb eines Rechenknotens wurden POSIX Threads genutzt. Sie sind ein standardisiertes API für multithreaded Anwendungen, also solche, die in Programmteile gespalten sind, die unabhängig voneinander ausgeführt werden können. Diese Programmteile können im Gegensatz zum OpenMP Framework sehr einfach verschiedene Aufgaben übernehmen. Dies wurde im Sinne einer bei dieser Gelegenheit noch deutlich verbesserten Lastverteilung und Speicherausnutzung getan. Für den Baumdurchlauf wird eine wählbare Anzahl an Sub-Threads auf den einzelnen Prozessoren gestartet, während der Haupt-Thread jegliche Kommunikationsanfragen von und zu anderen Nodes bearbeitet. Hierdurch ist die Notwendigkeit zur Synchronität aller MPI-Ranks für den Datenaustausch während des Baumdurchlaufes nicht mehr gegeben. Damit entfällt auch der Bedarf für Wechselwirkungslisten zur Kraftberechnung *nach* dem Baumdurchlauf. Stattdessen kann der Ausdruck für die Kraft zwischen Teilchen und Clustern direkt ausgewertet werden nachdem der Bedarf für eine Wechselwirkung festgestellt wurde.

Wie Abbildung 2.3 zeigt, erweist sich der hybride Ansatz zur Parallelisierung als äusserst erfolgreich. Der Baumdurchlauf skaliert mit Teilchenzahlen ab ca. 64.000.000 ideal bis zur vollen JUGENE-Maschine mit 294.912 Prozessoren. Weiterhin wurde es mit der hybriden Parallelisierung aufgrund der signifikanten Speicherersparnis möglich, Systeme mit bis zu 2.000.000.000 Teilchen zu simulieren. Solche Simulationen können nun mit dem Code bereits mit 32.768 Prozessoren behandelt werden. Dies stellt einen deutlichen Fortschritt gegenüber der vormaligen Beschränkung auf ca. 400.000.000 Teilchen auf bis zu 16.384 Prozessoren dar.

Erweiterung von PEPC für astrophysikalische Anwendungen Zur Vorbereitung der geplanten SCAFACOS-Anwendung PEPC-G mit SPH (Smooth Particle Hydro-

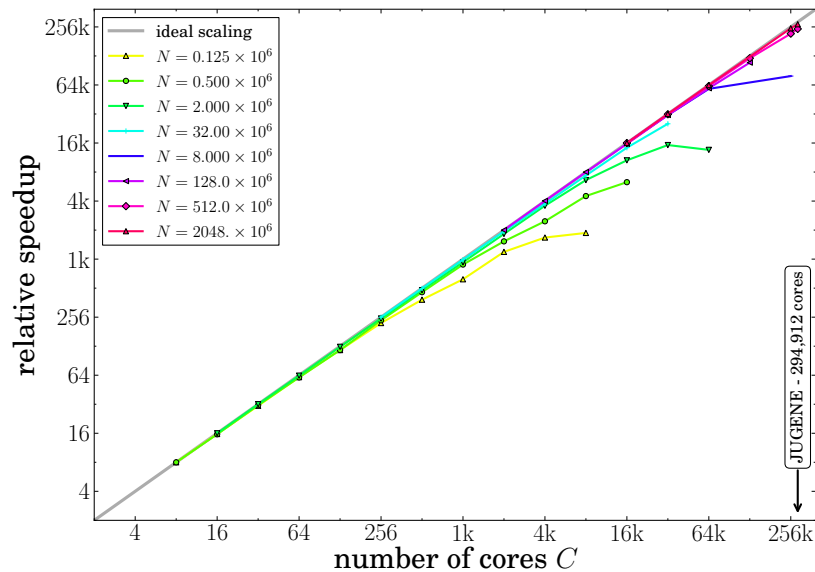


Abbildung 2.3: Skalierungsverhalten der Kraftberechnung für verschiedene Teilchenzahlen bis zur vollen Blue Gene/P Maschine JUGENE mit 294.912 Prozessoren.

dynamics), mit der Simulationen von protostellaren Staubscheiben durchgeführt werden sollen, wurde basierend auf der Baumstruktur des Barnes-Hut Algorithmus eine Nachbarschaftssuche für die Teilchen implementiert und an ersten Minimalbeispielen (1D Schall- und Schockwellen) erfolgreich getestet [21]. Derzeit wird für die Umsetzung des SPH-Teiles dieser Simulationen sehr detaillierter Zugriff auf interne methodenspezifische Datenstrukturen und Abläufe benötigt. Dieser kann im Rahmen der Bibliothek noch nicht zur Verfügung gestellt werden. Deswegen wird derzeit die Verwendbarkeit des Nahfeldalgorithmus hierfür untersucht. Alternativ müsste der Algorithmus zur Berechnung des Gasdynamik-Teiles der PEPC-G Anwendung im Anwendungsprogramm selbst umgesetzt werden. Beide Varianten erfordern konzeptionelle Modifikationen, die nicht in der Laufzeit des Projekts durchführbar waren. Dennoch ist das prinzipielle Ziel, den Barnes-Hut Algorithmus einzubinden und mit einem astrophysikalischen Frontend zu koppeln nach derzeitigem Stand weiterhin realisierbar.

WP 2-3: Periodische Summations-Verfahren

Anmerkung vom Projektpartner Universität Stuttgart: Der plötzliche Tod unseres Mitarbeiters Dr. Alexey Neelov im August 2009 hat das Projekt schwer getroffen. Für nahezu ein Jahr konnte kein Ersatz für ihn gefunden werden. In der darauffolgenden Zeit konn-

te durch geschickte Verteilung der Arbeiten innerhalb der Arbeitsgruppe der Zeitplan dennoch eingehalten werden. Die hohe Anzahl an Mitarbeitern im Projekt ist diesem Umstand geschuldet.

Parallelisierung der Ewald-Summation

Vor allem zu Test- und Validierungszwecken wurde vom Projektpartner Stuttgart auch das originale Ewald-Summationsverfahren[82] (mit kontinuierlicher Fourier-Transformation) implementiert, parallelisiert und in die Bibliothek eingebaut. Dabei wurde ein Schwerpunkt nicht auf die Effizienz gelegt, sondern auf die Lesbarkeit des Codes, da die Komplexität des Algorithmus von $\mathcal{O}(N^{\frac{3}{2}})$ in keinem Fall mit den anderen Methoden kompetitiv sein kann. Mit genügend hohem Rechenaufwand kann die Methode Kräfte und Potentiale mit prinzipiell beliebig hoher Genauigkeit berechnen, daher ist sie gut geeignet zur Erzeugung von Referenzkräften für die bibliotheksinterne Testumgebung.

Kombination von paralleler Ewald-Summation und paralleler NFFT

Die NFFT-basierte schnelle Ewald-Summation gemäß [86] wurde vom Projektpartner Chemnitz erfolgreich parallelisiert, implementiert und in die SCAFACOS-Bibliothek integriert. Für die Anbindung der parallelen NFFT an die parallele Ewald-Summation war eine Anpassung der parallelen Datenaufteilung notwendig. Zusätzlich wurde die periodische schnelle Summation auf nichtkubische Boxen erweitert.

Modularisierung der elektrostatischen Methoden aus dem Programmpaket ESPRESSO

Die in der Simulationssoftware ESPRESSO des Projektpartners Stuttgart enthaltenen Methoden P3M, MMM1D und MMM2D wurden modularisiert, aus der Software herausgelöst und in die Bibliothek integriert. Zur besseren Lesbarkeit und Modularität wurde der Code refaktoriert. Die Parallelisierung der Methoden wurde verallgemeinert und an die Vorgaben der Bibliothek angepasst.

Während der Projektlaufzeit stellte sich heraus, dass die Methoden MMM1D und MMM2D bezüglich ihrer Komplexität und Effizienz nicht mit den anderen Methoden des Projektes konkurrieren können. Daher wurde der Fokus der Weiterentwicklung auf die Methode P3M gelegt.

Weiterentwicklungen der P3M-Methode

Zu Beginn der Projektlaufzeit war der in ESPRESSO enthaltene P3M-Algorithmus nur in der Lage, Kräfte und Potentiale in kubischen Geometrien zu berechnen. Um die Anwendung des Algorithmus in mehr Systemen zu ermöglichen, wurde der Algorithmus

auf nichtkubische Geometrien erweitert. Dies erforderte nicht nur die Anpassung des eigentlichen Algorithmus, sondern auch die Entwicklung der entsprechenden analytischen approximativen Fehlerformel und die Anpassung des Algorithmus zum automatischen Tuning der Parameter.

Während der Projektlaufzeit implementierte und untersuchte Dr. Alexey Neelov die Wechsellmethode für den P3M-Algorithmus [52], die erstmals von Hockney und Eastwood [87] beschrieben wurde, jedoch nirgends verwendet wurde. Durch diese Methode wird mit exakt doppeltem Arbeitsaufwand eine um einen Faktor 8 erhöhte Genauigkeit der Methode erzielt. Man kann also für eine vorgegebene Präzision die Rechenzeit signifikant verbessern.

Neben dem P3M-Algorithmus gibt es weitere, eng verwandte Methoden mit der gleichen Komplexität $\mathcal{O}(N \log N)$ die auf der Ewald-Summe [82] und der schnellen Fourier-Transformation [101], beispielsweise PME [75] oder SPME [81]. In den letzten Jahren wurde deutlich, dass alle diese Algorithmen im Wesentlichen als ein Algorithmus mit verschiedenen Komponenten (Funktion zur Ladungsvermischung, Wechsellmethode, Influenzfunktion und Methode der Ableitung der Kräfte) aufgefasst werden können, die prinzipiell ausgetauscht werden können [76]. Die genaue Wahl der Komponenten beeinflusst die Genauigkeit, die Geschwindigkeit sowie die Parallelisierbarkeit des Algorithmus. Die Auswahl der Komponenten in den Varianten P3M, PME und SPME hat in der Regel historische Gründe; eine systematische Untersuchung der optimalen Kombination von Komponenten insbesondere im Hinblick auf die Geschwindigkeit lag zu Projektbeginn nicht vor. Eine solche systematische Untersuchung wurde an einer eigenständigen sequentiellen Implementation der Methoden durchgeführt [54]. Als optimale Kombination im Hinblick auf Geschwindigkeit und Genauigkeit stellte sich die P3M-Methode mit Wechsellmethode und analytischer Ableitung (P3M-ad-i) heraus, die einen Geschwindigkeitszuwachs von bis zu Faktor 3 bringen kann (siehe Abbildung 2.4). Obwohl die Untersuchung an einem sequentiellen Code durchgeführt wurde, so ist sie auf den parallelen Fall übertragbar, da die Parallelität des Algorithmus vollständig in der Umsetzung der schnellen Fourier-Transformationen zu sehen ist. Die gefundene optimale Methode verwendet weniger Fourier-Transformationen als die traditionelle P3M-Methode, daher ist die parallele Skalierung mindestens so gut. Ergänzend zur Verbesserung des eigentlichen Algorithmus war es notwendig, die analytische approximative Fehlerformel und den Algorithmus zum automatischen Tuning entsprechend anzupassen.

Der neue, optimierte Algorithmus wurde in die SCAFACOS-Bibliothek eingefügt. Damit verfügt die Bibliothek über die Implementation der modernsten und schnellsten Variante eines Ewald-Gitter-Algorithmus.

In der Zusammenarbeit zwischen den Projektpartnern TU Chemnitz und Universität Stuttgart wurde festgestellt, dass die in Chemnitz entwickelte Methode der Ewald-Summation mit Hilfe der nichtäquidistanten schnellen Fourier-Transformation im vollperiodischen Fall mathematisch äquivalent ist zur P3M-Methode [2].

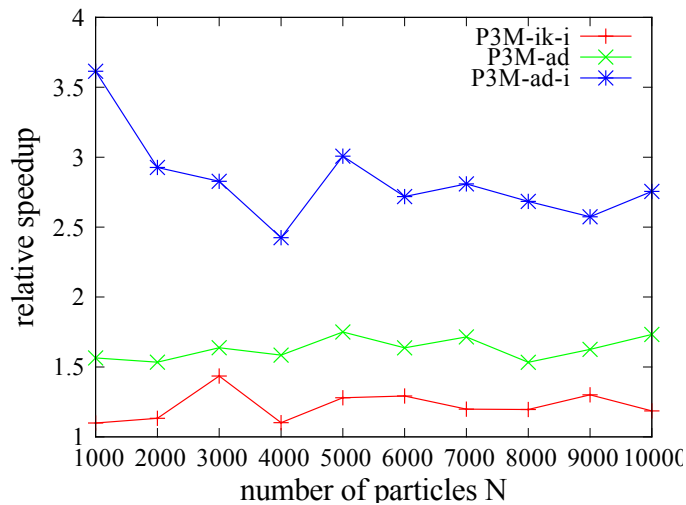


Abbildung 2.4: Speedup der verschiedenen P3M-Algorithmen mit unterschiedlichen Komponenten im Vergleich zum „traditionellen“ P3M mit ik-Ableitung für verschiedene Teilchenanzahlen N .

Des Weiteren wurde festgestellt, dass die softwaretechnisch ungünstig gestaltete parallele Implementation der schnellen Fourier-Transformation, die im P3M-Algorithmus des Projektpartners Universität Stuttgart verwendet wird, algorithmisch äquivalent ist zur effizient parallelisierte und lastbalancierte Bibliothek PFFT des Projektpartners TU Chemnitz. Dies ermöglichte es dem Projektpartner Universität Stuttgart, seine Implementation gegen die Chemnitzer Implementation auszutauschen. Da die Fourier-Transformation der einzige Teil des Algorithmus ist, der kritisch in Bezug auf die Parallelisierung und Lastbalancierung sein kann, verschwand damit die Notwendigkeit für den Projektpartner Stuttgart, sich mit der Lastbalancierung des Algorithmus zu beschäftigen. Dies erlaubte dem Projektpartner Stuttgart, die oben genannten Verbesserungen am P3M-Algorithmus (Wechselmethode und Optimierung der Komponenten) vorzunehmen, die nicht Teil des ursprünglichen Projektantrages waren, sondern deutlich darüber hinausgehen.

P3M-Methode für Systeme mit punktförmigen Dipolen in allgemeinen periodischen Geometrien

Auch die sequentielle Implementation des P3M-Algorithmus für punktförmige Dipole wurde vom Projektpartner Universität Stuttgart aus dem Simulationsprogramm ESPRESSO herausgelöst und nach dem Vorbild der P3M-Methode für Monopole modularisiert und refaktoriert. Mit Hilfe der Testfälle in ESPRESSO wurde die Implementation validiert.

Auf einem Projekttreffen wurde diskutiert, wie der Algorithmus in die SCAFACOS-Bibliothek eingefügt werden könne. Das Interface der Bibliothek ist nur für Teilchensysteme mit (Monopol-)Ladungen ausgelegt. Die Methode zur Berechnung von Wechselwirkungen zwischen punktförmigen Dipolen erfordert ein anderes Interface als alle anderen Methoden, da neben den Ladungen für die Teilchen auch die Dipolmomente benötigt werden, und neben den Kräften auch das Drehmoment zurückgegeben werden muss. Eine entsprechende Erweiterung des Interfaces würde es schwieriger zu benutzen machen, ohne die Hauptfunktionalität der Bibliothek wesentlich zu erweitern.

Daher wurde beschlossen, den dipolaren P3M-Algorithmus nicht zum integralen Bestandteil der SCAFACOS-Bibliothek zu machen, sondern ihn stattdessen in einer eigenständigen Bibliothek zu veröffentlichen, deren Interface jedoch an die SCAFACOS-Bibliothek angelehnt ist.

Parallelisierung der P3M Methode für dipolare Systeme

In Stuttgart wurde die dipolare P3M-Methode analog zur P3M-Methode für Punktladungen mit Hilfe der beim Projektpartner Chemnitz entwickelten PFFT-Bibliothek parallelisiert.

Lastbalancierung der Methoden

Ein besonderes Problem der nicht-periodischen schnellen Summation liegt in der ungleichmäßigen Verteilung der Stützstellen in Bezug auf die Simulationsbox. Bedingt durch Skalierungen der Teilchendaten konzentrieren sich die Stützstellen auf weniger als ein Achtel der Box. Daher werden in typischen Simulationen lediglich etwa 3% der FFT-Ausgabedaten für die weitere Berechnung benötigt, siehe [18]. Dies bedeutet eine stark ungleichmäßige Verteilung des Speicher- und Berechnungsbedarfs bei Verwendung einer herkömmlichen Blockzerlegung der Daten. Durch gezielte Erweiterung der parallelen FFT- und NFFT-Algorithmen auf abgeschnittene Ein- und Ausgabedaten konnte beim Projektpartner TU Chemnitz dennoch nahezu eine Gleichverteilung des Speicher- und Berechnungsaufwandes auf alle Prozesse erreicht werden.

Wegen der Entwicklung der parallelen FFT-Bibliothek in Chemnitz hat sich Projektpartner Stuttgart nicht mit der Lastbalancierung der Methode auseinandergesetzt. Stattdessen wurden in Stuttgart die oben genannten Verbesserungen am P3M-Algorithmus für Punktladungen gemacht.

WP 2-4: Mehrgitter-Verfahren

Optimierung des geometrischen parallelen MG für offene und 3d-periodische Systeme

Zunächst wurde der bestehende parallele Mehrgitter-Code *PP3MG* des Kooperationspartners in Wuppertal in das Bonner Moleküldynamiksoftwarepaket integriert, um ein optimales Interface zwischen Moleküldynamik-Software und Mehrgitter-Löser zu entwickeln und auch die grundsätzlichen Skalierungseigenschaften des Mehrgitter-Lösers bei Einsatz in einer fremden Software zu untersuchen.

Danach wurde diese Anknüpfung an die Molekülsoftware im Hinblick auf parallele Datenverteilung der einzelnen Programmkomponenten und auf Wiederverwendung von Lösungen vorheriger Zeitschritte im iterativen Mehrgitter-Löser optimiert.

Der parallele Mehrgitter-Code selbst wurde als eigenständiges Modul *PP3MG* in die gemeinsame Bibliothek *SCAFACOS* des Projekts integriert. Die darin schon enthaltene Behandlung des kurzreichweitigen Anteils konnte extrahiert und verallgemeinert werden. Sie wird dadurch auch den anderen Modulen zur Verfügung gestellt, da einige ebenfalls auf dem Splitting-Ansatz der getrennten Berechnung von Nah- und Fernfeld basieren. Außerdem wurde der Mehrgitter-Code auf die Behandlung von Gebieten beliebiger Größe und auch von quaderförmigen, nicht-kubischen Gebieten erweitert. Schließlich wurde die Behandlung offener Randbedingungen, welche im parallelen Mehrgitter-Code soweit nur seriell implementiert war, erfolgreich parallelisiert.

Eine Möglichkeit für eine genauere Berechnung der Kräfte, welche sich als negativer Gradient des Potentials auf dem Gitter ergeben, ergibt sich aus der Verwendung von versetzter Gitter (*interlaced grid*, *staggered mesh*), die im Rahmen der P3M-Methode vom Projektpartner in Stuttgart untersucht worden sind. Hierbei kann der Diskretisierungsfehler in der Kraftauswertung, der durch das diskrete, ortsfeste Gitter entsteht, durch Berechnung auf zwei, zueinander um eine halbe Gittermaschenweite versetzten Gittern gedämpft werden. Dies entspricht der Elimination des Fehlerterms höchster Ordnung mittels Extrapolation.

Dazu wurde die *interlaced grid* Methode für das in dem Bonner Moleküldynamikpaket vorhandenen SPME-Verfahren zum StME (*Staggered Mesh Ewald*) [71] Verfahren implementiert. Die Reduktion des Fehlers in der Kraftauswertung ist dabei über einen weiten Bereich des Splittings vorhanden.

Erweiterung des geometrischen parallelen Mehrgitterverfahrens für offene und 3d-periodische Systeme auf 1d- und 2d-periodische Systeme

Es wurde ein geometrisches paralleles Mehrgitter-Verfahren zunächst für periodische Systeme entwickelt, sowie die existierenden Verfahren für offene und quasi-periodische Systeme in der Literatur evaluiert.

Anschließend wurde die Implementation in Tests auf großen Prozessorzahlen verbessert und auf hohe Effizienz optimiert.

In PP3MG existiert bereits eine sequentielle Implementierung der Behandlung von offenen Randbedingungen. Diese basiert ursprünglich auf der Methode von Washio und Osterlee [107], einer systematischen Vergrößerung des Diskretisierungsgitters nach außen bis auf ein gegebenes Level und der Anwendung von Dirichlet-Null-Randwerten. Da bei dieser Methode der numerische Fehler stark von der Zahl der Vergrößerungsschritte abhängt und nicht gut beschränkt werden kann, wurde stattdessen eine Modifikation vorgeschlagen, die am Rand des vergrößerten Gitters die Werte der Lösung annimmt. Diese werden mit Hilfe einer Greens-Funktionsentwicklung berechnet [65]. Desweiteren wurde von Bolten gezeigt, dass sich die Zahl der Freiheitsgrade der vergrößerten Gitter optimal beschränken lässt [66].

Für die Behandlung von 1d- und 2d-periodischen Systemen ließ sich diese Methode übertragen, da dort eine analoge schnelle Berechnung der entsprechenden Greensfunktionen für quasiperiodische Gebiete gelingt. Diese wurden in [95] realisiert. Im Rahmen des neuen Mehrgitter-Verfahrens wurde nun eine effiziente parallele Implementierung dieser Verfahren entwickelt.

Die Greens-Funktionsentwicklung wurde in MatLab auf den Abfall ihrer Reihendarstellung untersucht. Anschließend wurde dieses Mehrgitter-Verfahren nach C++ übertragen.

Formulierung von Mehrgitterverfahren für variierende dielektrische Konstanten in 1d-, 2d- und 3d-periodischen Systemen

Grundsätzlich ist die Behandlung von auf dem Gesamtgitter ausreichend *glatten* variierenden dielektrischen Konstanten unproblematisch, da diese Konstanten einfach in das Aufstellen der rechten Seite des Poisson-Problems integriert werden können. Schwierig, aber in der Anwendung wesentlich bedeutsamer, sind *springende* Dielektrizitätskonstanten.

Die Schwierigkeit besteht darin, dass verschiedene Mehrgitter-Komponenten die Stetigkeit des des Gradienten des Potentials verlangen, siehe [106]. Grundsätzlich ist bekannt, dass sich Unstetigkeiten in der Dielektrizitätskonstante und in dem Gradienten des Potentials sich genau wegheben, so dass der resultierende Fluss stetig ist. Die betroffenen Komponenten sind einerseits die Interpolation auf das feinere Gitter und andererseits die Darstellung des Problems auf dem gröberen Gitter. Sie müssen beide so modifiziert werden, dass sie auf dem stetigen Fluss operieren. Aufgrund des verspäteten Projektbeginns seitens der Universität Bonn, des Weggangs von Herrn Ralf Wildenhues und der nicht vollständig erteilten Restfinanzierung wurde die Planung für diesen Arbeitspunkt entsprechend reduziert werden.

Optimierung und Lastbalancierung des parallelen MG zur Behandlung von inhomogenen Verteilungen

Zur Behandlung inhomogener Partikelverteilungen sollte untersucht werden, inwiefern sich adaptive Gitterverfeinerungen effizient parallelisieren lassen. Hierfür stehen mittlerweile baum-artige Verfahren in der Bibliothek zur Verfügung, die für inhomogene Problemstellungen gut geeignet sind und deren Verwendung sich als positiv auf die Gesamteffizienz des Codes herausgestellt hat.

Weiterhin wurden verschiedene Optimierungen vorgenommen. So wurden lokale Spline-Funktionen zur Interpolation der Punktladungen auf das Gitter verwendet, so dass der Aufwand hier aufgrund des kompakten Trägers unabhängig von der Gittergröße ist. Weiterhin wurde das Mehrstellenverfahren nach Collatz zusätzlich zum Standard-Differenzenstern integriert, welcher mit einem größeren Gitter eine vergleichbare Genauigkeit erreicht, so dass sich so auch Inhomogenitäten effizienter berechnen lassen.

WP 2-5: Lokale Gitterverfahren

Entwicklung eines Randelementverfahrens zur iterativen Behandlung der elektrostatischen Randbedingungen für Systeme mit dielektrischen Einschlüssen

Da die Theorie zu diesem Verfahren bereits vorhanden und ausformuliert zur Verfügung stand, verlief die Implementierung relativ problemlos. Eine Parallelisierung war hierbei unnötig, da die Kraftberechnung selber in einem bereits vorhandenen Algorithmus (in unserem Fall P3M) durchgeführt wird. Die iterative Berechnung der Randladungen ist zwar im Prinzip parallelisierbar, nimmt aber einen sehr kleinen Teil der gesamten Rechenzeit ein.

Zur leichteren Benutzung wurde außerdem ein Interface entwickelt, um mit einfachen Kommandos Ebenen, Kugeln, Poren und andere geometrische Objekte zu erzeugen.

Herleitung des Formalismus einer Coulomb-Wechselwirkung mit räumlich variierender Dielektrizitätskonstanten

Die Herleitung für den dynamischen Teil des Algorithmus wurde vollständig durchgeführt. Sie verlief in weiten Abschnitten analog zur Herleitung mit konstantem dielektrischen Hintergrund, mit einigen wenigen Anpassungen in der Zustandssumme des Systems.

Als Herausforderung erwiesen sich die statische Initiallösung des Systems, da es bisher noch keine implementierten Methoden gab, dieses Problem zu lösen. Und die Korrektur der Selbstenergie-Wechselwirkung, die über keine der bekannten Methoden (Gitter-Greens-Funktion oder Yukawa-Potentiale) für lokal variierende Dielektrika angewandt

werden kann. Für beide Probleme mussten Theorien und Herleitungen entwickelt werden. Eine Publikation mit allen Herleitungen befindet sich in Arbeit.

Implementierung und Validierung der Methode

Die Methode wurde zuerst im Programmpaket ESPRESSO implementiert, da die SCACOS-Bibliothek zu diesem Zeitpunkt noch nicht funktionsfähig war, und um die Validierung zu vereinfachen. Anschließend wurde die Methode modularisiert und in die Bibliothek übertragen.

Zur Validierung wurden für die Initiaillösung und den dynamischen Teil des Algorithmus jeweils zwei analytisch lösbare Probleme und eine Vergleichsrechnung mit dem Randelement Verfahren verwendet. Als Beispiel: Ein geladenes Teilchen zwischen zwei dielektrischen Wänden wurde simuliert und das Ergebnis mit der theoretischen Vorhersage verglichen. Die Permittivität um das Teilchen betrug hierbei 80 (Wasser), außerhalb der Wände wurde sie auf 1 (Vakuum) gesetzt. Das Ergebnis und der theoretische Verlauf sind in Abbildung 2.5 gezeigt und zeigen sehr gute Übereinstimmung.

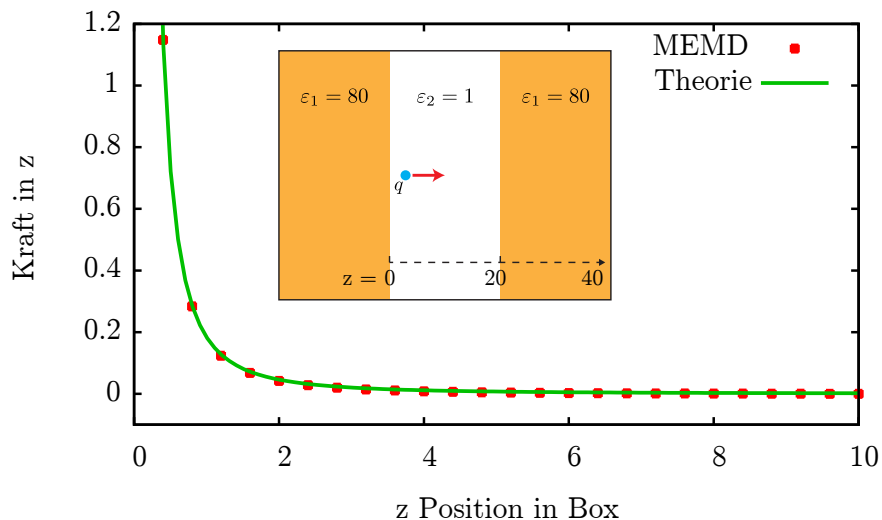


Abbildung 2.5: Der Graph zeigt die Kraft auf das Teilchen an den Positionen $0 < z \leq 10$ zwischen zwei dielektrischen Wänden an den Positionen $z = 0$ und $z = 20$. Die theoretische Vorhersage als Vergleich wurde mit der Methode der Spiegelladungen berechnet.

Optimierung der freien Parameter und Formalismus für Fehlerabschätzung

Für den Algorithmus gibt es 2 frei wählbare Parameter: Den Vorfaktor $1/c^2$, der in der vollen Elektrodynamik durch die Lichtgeschwindigkeit c festgelegt ist, und die Größe des interpolierenden Gitters. Das Stabilitätskriterium für den Algorithmus, dass die Ausbreitungsgeschwindigkeit der magnetischen Felder und damit der genannte Vorfaktor die Geschwindigkeit der Teilchen signifikant übersteigen müssen, liefert einen Zusammenhang zwischen diesen beiden Parametern.

Für höhere Lichtgeschwindigkeiten wird der algorithmische Fehler kleiner, aber es ist ein größeres Gitter notwendig um die magnetischen Felder mit dieser Geschwindigkeit zu propagieren. Damit wird der Fehler der linearen Interpolation größer. Die Fehler sind gegenläufig und es existiert folglich ein Minimum. In unserer Herleitung zeigten wir, dass das Minimum für ein typisches System darin liegt, die Gitterkonstante so zu wählen, dass der kleinste Teilchenabstand im System gerade der Grobheit des Gitters entspricht.

Diesem Ansatz folgend wurde ein auf der Anfangskonfiguration basierender Schätzalgorithmus implementiert, der die optimalen Parameter wählt. Bei genauerem Wissen über das zu simulierende System und dessen Dynamik können jedoch durch manuelle Parameterwahl noch exaktere oder schnellere Ergebnisse erzielt werden.

Parallelisierung und Loadbalancing

Die Parallelisierung eines lokalen Algorithmus ist eindeutig. Das System wird in Teilgebiete zerlegt (domain decomposition) und diese werden auf unterschiedlichen Prozessoren berechnet. Für die Kommunikation wurde ein neuer MPI-Datenaustauschtyp erstellt, der die gesamte Berührungsfläche der beiden Regionen enthält. So wird die Latenzzeit der Kommunikation minimiert, da jeden Zeitschritt nur ein großes Paket ausgetauscht werden muss.

Die Kommunikation findet asynchron statt. Während des Informationsaustausches wird bereits die Feldpropagation im Inneren jedes Teilgebietes durchgeführt.

Loadbalancing ist für diesen Algorithmus automatisch gegeben, da eine gleichmäßige Gebietsaufteilung des Systems beinhaltet, dass jeder Prozessor gleich viele Zellen berechnet. Durch die einfache lineare Interpolation und Rückinterpolation findet der Großteil der Berechnungen als Propagationen auf dem Gitter statt und ist damit weitgehend unabhängig von der Anzahl der Teilchen im Teilgebiet. Dies bestätigt sich in Laufzeitversuchen.

Vergleich der Effizienz des Algorithmus mit anderen Methoden

Durch die unumgängliche lineare Interpolation auf ein Gitter ist mit diesem Algorithmus nur die Kraftberechnung mit einem mittleren relativen Fehler von 10^{-3} möglich. In diesem Bereich schneidet der Algorithmus bei weitgehend homogenen Ladungsverteilungen

verhältnismäßig gut ab und zeigt sich sogar effizienter als die meisten anderen Methoden, mit Ausnahme der Schnellen Multipolmethode. Bei inhomogen verteilten Systemen leidet die Effizienz darunter, dass die Propagation der lokalen Felder auch durch vollständig leere Bereiche sehr viel Rechenzeit kostet, und die Geschwindigkeit des Algorithmus erreicht die Werte der anderen Methoden nicht mehr.

Zudem mussten bei Skalierungstests festgestellt werden, dass die Implementation bereits bei geringeren Prozessorzahlen als die Methoden der Partner in der Skalierung einknickt. Dies scheint jedoch ein Problem der Implementation und nicht des Algorithmus selber zu sein und steht damit zukünftigen Optimierungen offen.

Zusammenfassend ist die Methode für homogene Systeme durchaus konkurrenzfähig in der Effizienz, wenn der Benutzer sich mit dem minimal zu erreichenden Fehler zufrieden geben kann. Für inhomogene Systeme bleibt das Alleinstellungsmerkmal der lokal variierenden Dielektrika bestehen, auch wenn es für konstante dielektrische Hintergründe deutlich schnellere Methoden gibt.

WP 2-6: Kurzreichweitige Wechselwirkungen

Vergleich bestehender Implementierungen

Der kritische Teil bei der Evaluierung der paarweisen kurzreichweitigen Wechselwirkungen ist die Bestimmung der gegenseitigen Abstände zwischen Teilchenpaaren. Hierzu existieren verschiedene Implementierungen von Listentechniken. Die Linked-Cell Methode ist dabei eine Basistechnik, da sie sowohl $\mathcal{O}(N)$ Speicherbedarf als auch $\mathcal{O}(N)$ Rechenbedarf zur Erstellung der Listen hat. Zusätzlich hierzu kann ein Verlet-Listen Verfahren implementiert werden, das einen erheblich größeren Speicherbedarf hat, der abhängig von dem gewählten Einfluß-Radius kubisch steigt. Um Speicherkonflikte in der Bibliothek zu vermeiden, wurde daher die Linked-Cell Methode bevorzugt. Hierbei wurden verschiedene Varianten miteinander verglichen und implementiert, wobei die Adressierung der Nachbarzellen und das Auslesen dieser Zellen mithilfe eines binären Schlüssels die schnellste Variante darstellte.

Definition und Implementierung einer allgemeinen Schnittstelle

Im Zusammenhang mit kurzreichweitigen Wechselwirkungen wurde eine allgemeine Schnittstelle entwickelt, da einige Methoden einen Löser für kurzreichweitige Wechselwirkungen benötigen. Entsprechende Methoden sind beispielsweise P3M und P2NFFT, da sie Korrekturen ihrer Lösung durchführen müssen, die auf dem Nahfeld der Ladungen basieren. Für diese Methoden wurde ein Nahfeldlöser implementiert, der in der Lage ist diese Korrekturen durchzuführen. Als Alternative wird es dem Benutzer der Bibliothek ermöglicht, diese Nahfeldkorrektur an sein Programm zu delegieren, damit er die Korrektur von seinem (MD-)Programm berechnen lassen kann.

Der Nahfeldlöser ist vollständig parallel, verwendet die parallelen Sortiermethoden der TU Chemnitz, basiert auf dem Linked-Cell Algorithmus, der dafür sorgt, dass die Komplexität der Suche nach den nächsten Nachbarn der Ladungen nicht mehr quadratisch zur Ladungsanzahl ist, sondern sich linear zu dieser verhält. Bei der Implementierung dieses Algorithmus wurden kleinere Optimierungen durchgeführt, die im folgenden Absatz beschrieben werden. Der Nahfeldlöser berechnet Systeme mit übergebenen Potentialen, für die jeweils ein Cut-Off-Radius mitangegeben werden muss.

Parallelisierung und Loadbalancing

Die Parallelisierung des Nahfeldlöser wurde durch die Verwendung der parallelen Sortiermethoden erzielt. Mit Hilfe der Sortiermethoden werden die Ladungen auf die Prozesse des verwendeten MPI-Kommunikators verteilt und dann dort in Linked Cells einsortiert. Da der Nahfeldlöser somit auf einer Gebietszerlegung basiert, wurden auch Untersuchungen zum verbesserten Ausgleich von Arbeit zwischen den Prozessen angestellt. Im Rahmen dieser Untersuchungen wurde ein Verfahren entwickelt, dass Loadbalancing durch das Anpassen von lokalen Gebietsgrenzen ermöglicht[27, 28]. Das Loadbalancingverfahren wurde in dieser Form allerdings nicht in den Löser übernommen, da für das Verfahren es garantiert sein muss, dass die Gebietszerlegung des Verfahrens weiter verwendet wird. Da allerdings die Ladungen in jedem Simulationsschritt aus dem Programm des Benutzers neu übertragen werden und dort eventuell einem eigenen Loadbalancing-Verfahren unterliegen, das die Teilchen umsortiert, ist die Verwendung eines lokalen Loadbalancingverfahrens ungeeignet.

Optimierte Linked-Cell-Methode

Um die Speichernutzung des verwendeten Linked-Cell Verfahrens zu optimieren wurde darauf verzichtet sämtliche Zellen eines Prozessgebietes auf einmal zu allokalieren. Durch die dynamische Allokation von verwendeten Zellen wird nicht nur der Speicher- aufwand reduziert, auch der Aufwand die Zellen zu verwalten wird verkleinert. Außerdem entfällt die Suche nach Nachbarpartikeln in leeren Nachbarzellen, da diese Zellen gar nicht angelegt wurden.

Implementierung verschiedener Potentiale

Da sich die Korrekturen der beiden Methoden P3M und P2NFFT unterscheiden, wurde der Nahfeldlöser so angelegt, dass es möglich ist, ihm ein externes Potential, das er berechnen soll, mitzugeben. Auf diese Weise ist es auch möglich in Zukunft weitere Löser zur Bibliothek hinzuzufügen, die ebenfalls Nahfeldkorrekturen benötigen. Theoretisch wäre es auch möglich, durch Übergabe von Parametern eines kurzreichweitigen Potentials,

z.B. Lennard-Jones, die Nahfeldberechnungen eines MD-Programmes zu ersetzen, wobei ein externer Integrator hinzugefügt werden müsste. Diese Möglichkeit ist allerdings noch nicht weiter verfolgt oder getestet worden.

2.1.4 WP 3 – Benchmark und Evaluierung

Benchmarks

Die in der SCAFACOS Bibliothek vereinten Methoden lassen sich in vier Klassen unterteilen – (1) Gittermethoden, (2) FFT-basierte Methoden, (3) Multipolmethoden und (4) lokale Gittermethoden. Historisch bedingt stammen die Methoden aus unterschiedlichen Forschungsbereichen, so dass ein direkter Vergleich nicht möglich war. Die zugrundeliegenden Algorithmen weisen außerdem eine unterschiedliche Laufzeit- und Speicherkomplexität auf, so dass ein rein theoretischer Vergleich der Algorithmen unabhängig von der Implementierung wenig Rückschlüsse auf die tatsächliche Verwendbarkeit im Forschungsalltag ziehen lässt. Daher ist ein fairer Vergleich der Implementierungen bezüglich Rechenzeit und Genauigkeitsanforderungen unverzichtbar. Die SCAFACOS Bibliothek ermöglicht erstmals diesen Vergleich, da durch die Bereitstellung eines gemeinsamen Interfaces, die Benutzung stark vereinfacht wurde und so außerdem einheitliche Voraussetzungen (Datenstrukturen, Parameterübergabe) für alle Methoden geschaffen worden sind.

Für die hier dargestellten Benchmarks wurden exemplarisch zwei Teilchensysteme ausgewählt, die durch ihre Eigenschaften, wie z.B. ein dominanter Coulomb-Anteil für solche Vergleiche zu bevorzugen sind. Beide Systeme sind ladungsneutral und erfordern 3d-periodische Randbedingungen mit einer kubischen Einheitsbox. System A beschreibt geladene Wände aus Partikeln mit umgebenden Ladungswolken (siehe Abbildung 2.6). Die Systemgröße wurde von 8.100 über 9.830.400 bis 102.900.000 Partikel variiert (Set A1, A2, A3). System B zeigt die Partikelverteilung einer Silica-Schmelze (siehe Abbildung 2.6), Die Systemgröße wurde hier von 12.960 über 9.447.840 bis 103.680.000 Partikel variiert (Set B1, B2, B3). Die Referenzdaten wurden mit hochgenauen Ewald und FMM Rechnungen erzeugt. Neben der Systemgröße wurde auch die Genauigkeitsanforderung mit in die Benchmarks einbezogen. Als Genauigkeitsmaß wurde der mittlere quadratische Fehler der Potentiale $\Delta\Phi_{\text{rms}}$ gewählt. Die in Abbildung 2.7 und 2.8 dargestellten Kurven zeigen in der jeweils linken Spalte die Ergebnisse der Berechnung mit $\Delta\Phi_{\text{rms}} = 10^{-3}$ und in der jeweils rechten Spalte die Ergebnisse mit $\Delta\Phi_{\text{rms}} = 10^{-5}$.

Aus den Eingabedaten (Koordinaten, Ladungen) wurden die zugehörigen Potentiale und elektrischen Felder berechnet. Die Zeitmessung erfolgte für jeweils einen MD-Schritt. Die methodenabhängigen Parameter wurden so eingestellt, dass die jeweils geforderte Fehlerschranke $\Delta\Phi_{\text{rms}} = 10^{-3}$ und $\Delta\Phi_{\text{rms}} = 10^{-5}$ eingehalten wurde. Die Skalierungstests wurden auf Blue Gene System JUGENE in Jülich durchgeführt.

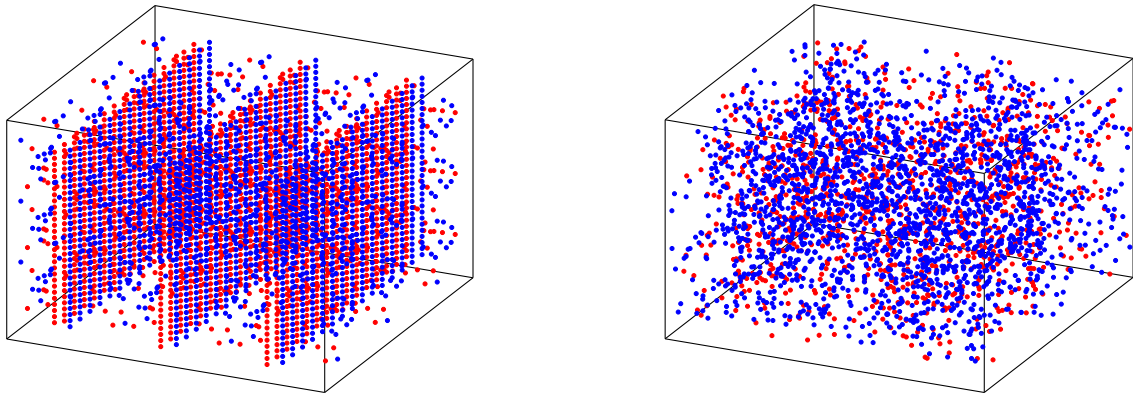


Abbildung 2.6: Für die Benchmarks wurde zwei unterschiedliche Systeme mit 3d periodischen Randbedingungen ausgewählt. System A (links) erlaubt eine gute Genauigkeitskontrolle durch den hohen langreichweitigen Anteil. System B (rechts) enthält Simulationsdaten aus einer Silica-Schmelze, welche auch zur Stabilitätsprüfung der Methoden verwendet wurden.

Die MEMD Implementierung ist aufgrund algorithmischer Beschränkungen nur für geringe Genauigkeiten geeignet. Allerdings erzielt die Methode im Gegenzug eine ähnliche Performance wie die Implementierungen P3M oder P2NFFT. Weiter bleibt zu beachten, dass die hier gezeigten MEMD Laufzeiten sich auf den dynamischen Anteil des Algorithmus beziehen. Für die Initialisierung und auch bei größeren Änderungen im System muss eine anfängliche numerische Relaxation durchgeführt werden, die nicht nur schlechter skaliert, sondern auch sehr viel zeitaufwendiger ist.

Die beiden Multigrid Löser PP3MG und VMG sind in ihrem algorithmischen Aufbau und der eigentlichen Implementierung sehr ähnlich. Aus diesem Grund ist teilweise nur ein Multigrid Löser im Benchmark vertreten. Für die beiden FFT-basierten Löser P3M und P2NFFT gilt dies analog. Da beide die FFT als Kernalgorithmus nutzen, ergibt sich ein vergleichbares Skalierungsverhalten bei hohen Prozessorzahlen. Aus diesem Grund wurde bei entsprechend großer Prozessoranzahl nur noch die P2NFFT im Benchmark aufgenommen.

Anbindung an die Simulationssoftware IMD als Testumgebung

Die SCAFACOS-Bibliothek wurde in einem frühen Stadium in den Molekulardynamik-Code IMD [104] eingebunden, um einerseits die Zweckmäßigkeit des Interface anhand der Erfordernisse eines realen MD-Codes zu überprüfen, und andererseits, um die verschiedenen Methoden über das einheitliche Interface miteinander vergleichen und deren Implementation auf Korrektheit und Vollständigkeit überprüfen zu können.

2.1 VERWENDUNG DER ZUWENDUNG UND ERZIELTE ERGEBNISSE

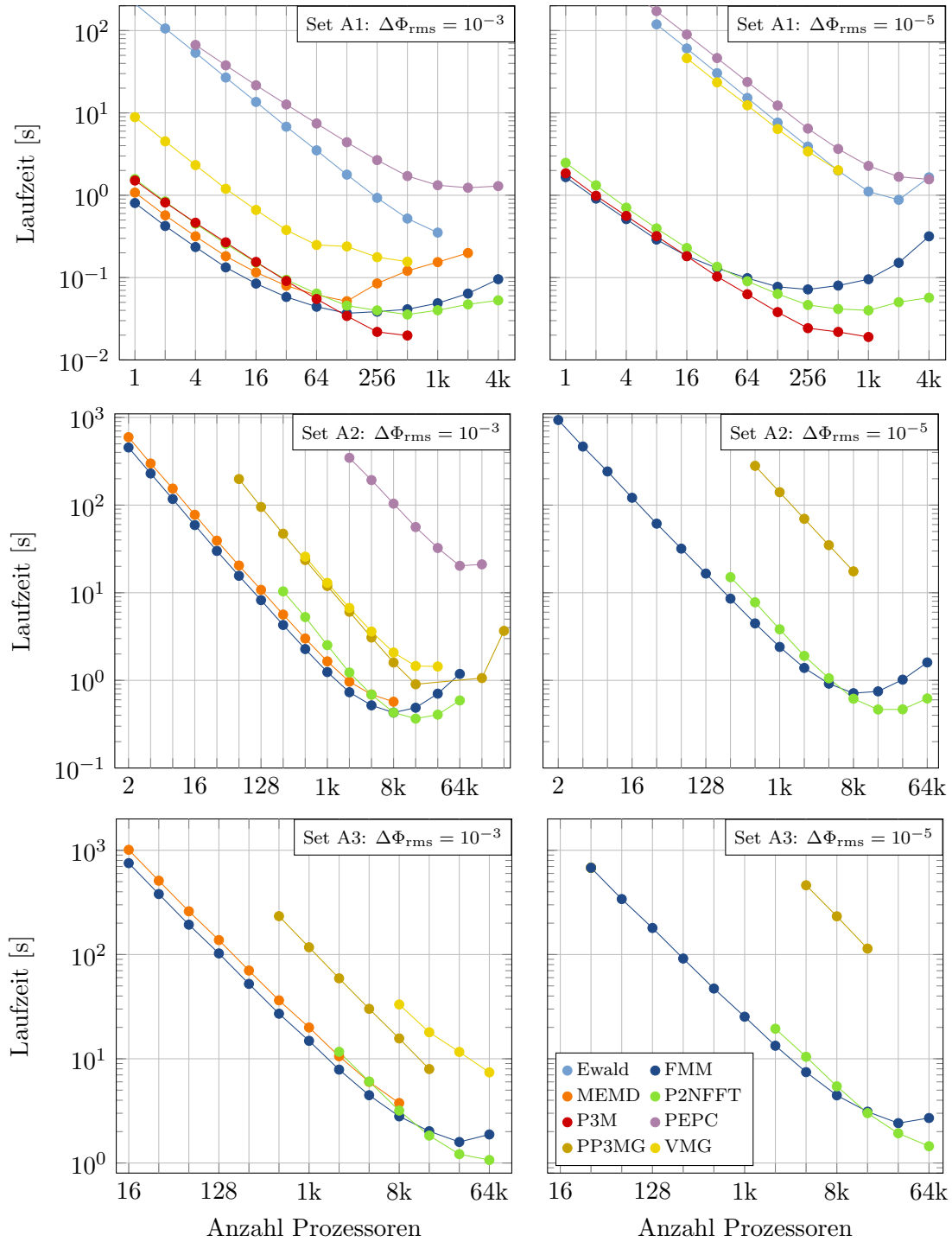


Abbildung 2.7: Benchmark für einen MD Zeitschritt im System A mit 8100, 9830400 und 102900000 Teilchen. Als obere Fehlerschranke wurde $\Delta\Phi_{\text{rms}} = 10^{-3}$ (links) und $\Delta\Phi_{\text{rms}} = 10^{-5}$ (rechts) gewählt.

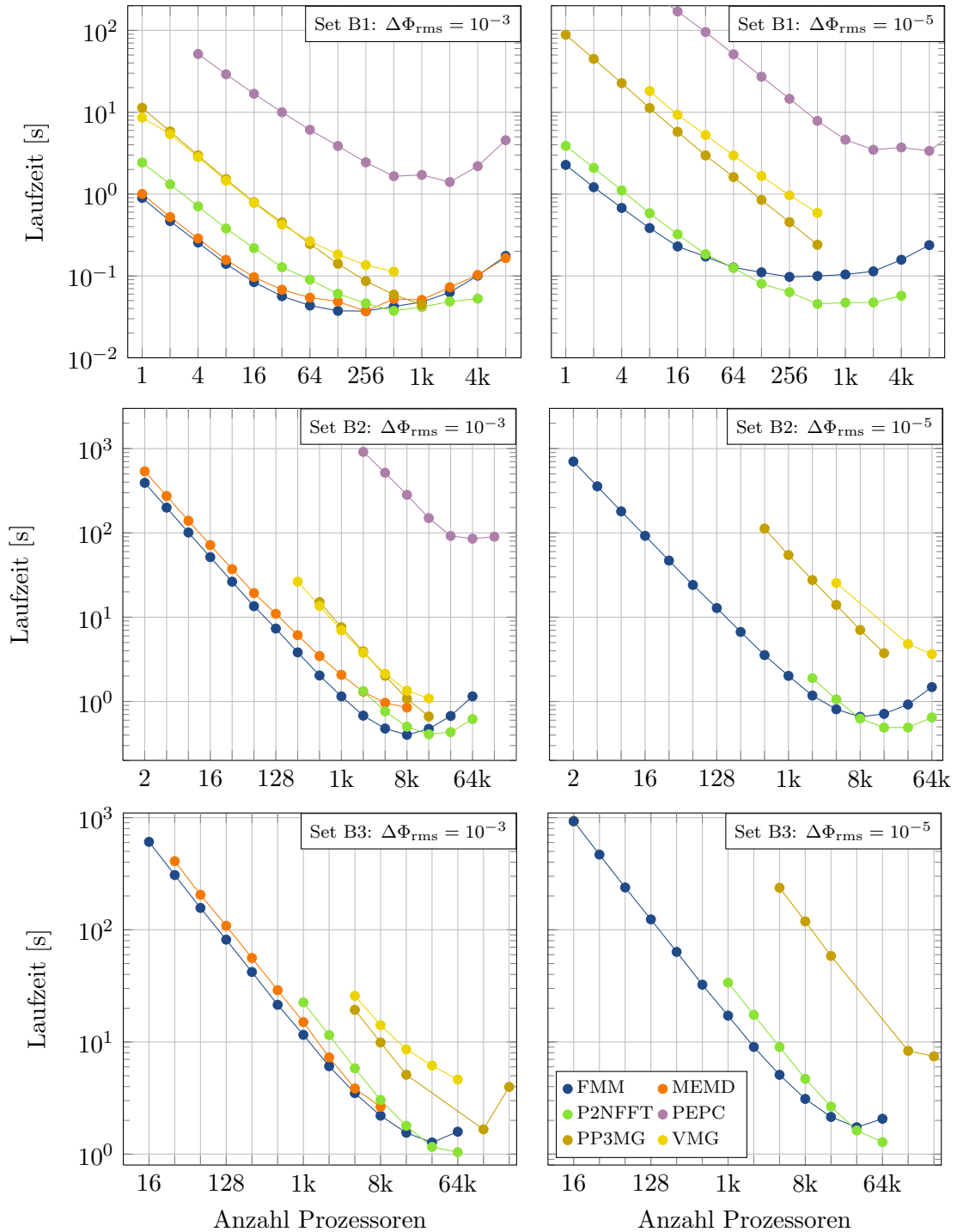


Abbildung 2.8: Benchmark für einen MD Zeitschritt im System B mit 12960, 9447840 und 103680000 Teilchen. Als obere Fehlerschranke wurde ebenfalls $\Delta\Phi_{\text{rms}} = 10^{-3}$ (links) und $\Delta\Phi_{\text{rms}} = 10^{-5}$ (rechts) gewählt.

Der SCAFACOS-Methodenaufruf war im Falle von IMD mit wenigen Zeilen Code zu implementieren, und dies dürfte bei den meisten MD-Codes nicht anders sein. Positionen und Ladungen der Teilchen müssen auf jedem Prozessor in einer flachen Liste an die Bibliothek übergeben werden. Da IMD andere Datenstrukturen aufweist, müssen diese Listen erst zusammenkopiert, und die berechneten Kräfte anschließend entsprechend wieder verteilt werden. Solche Interface-Routinen zu schreiben ist jedoch sehr einfach, und die Rechenzeit dafür ist vernachlässigbar gegenüber der Kraftberechnung. Der Vorteil ist, dass die Teilchen in einer beliebigen Verteilung an die Bibliothek übergeben werden können, und die Kräfte und Potenziale in derselben Verteilung und derselben Reihenfolge der Teilchen wieder an den aufrufenden Code zurückgegeben werden. Da die Teilchen in einer beliebigen Verteilung an die Bibliothek übergeben werden können, kümmert sich die Bibliothek gegebenenfalls um eine passende Umverteilung auf die Prozessoren, je nach Anforderungen der aufgerufenen Methode.

Es ist im Prinzip möglich, über Parameter Hinweise über die Teilchenverteilung an die Bibliothek zu übergeben, welche die aufgerufene Methode ausnutzen, und deshalb eventuell auf eine Umsortierung der Teilchen verzichten kann. Aktuell wird dies von den Methoden noch nicht implementiert, es kann (und soll) aber noch nachgerüstet werden. Es gibt jedoch zwei Methoden (P3M und P2NFFT), welche die Berechnung des Nahfeldes an den MD-Code delegieren können, falls der MD-Code dies unterstützt (bei IMD ist dies der Fall). Der Vorteil davon ist, dass das Nahfeld dann zusammen mit weiteren kurzreichweitigen Wechselwirkungen berechnet werden kann, und somit nur ein einziger Durchlauf über die Teilchenpaare notwendig ist. IMD zum Beispiel berechnet alle kurzreichweitigen Kräfte gemeinsam, mit einem einzigen tabellierten Potenzial, weshalb das SCAFACOS-Nahfeld ohne zusätzlichen Aufwand mitberechnet wird. Dadurch kann eine Zeitersparnis von rund 30% erzielt werden.

Das gemeinsame Interface erwies sich als sehr hilfreich, um die verschiedenen Methoden bezüglich der Ergebnisse und der Performance miteinander zu vergleichen. Auf diese Weise konnten Fehler in einem frühen Stadium eliminiert werden.

Numerische Stabilität

Die Langzeitstabilität einer MD-Simulation ist ein guter Test für die Korrektheit der berechneten Kräfte. Zu diesem Zwecke wurden MD-Simulationen einer Silica-Schmelze von 12.960 Atomen bei etwa 2.950° C über 100.000 Zeitschritte von je 0.2 Femtosekunden durchgeführt. Dies ist ein sehr konservativer, kleiner Zeitschritt, der Diskretisierungsfehler in der Zeitintegration ausschließen lässt. Zudem verwendet IMD einen symplektischen Integrator vom leap-frog Typ, welcher eine hervorragende Langzeitstabilität aufweist.

Die Erhaltung der Gesamtenergie ist in Abbildung 2.9 ersichtlich, jeweils bei einer relativen Genauigkeit der Energieberechnung von 10^{-5} (oben) bzw. 10^{-3} (unten). Es zeigt

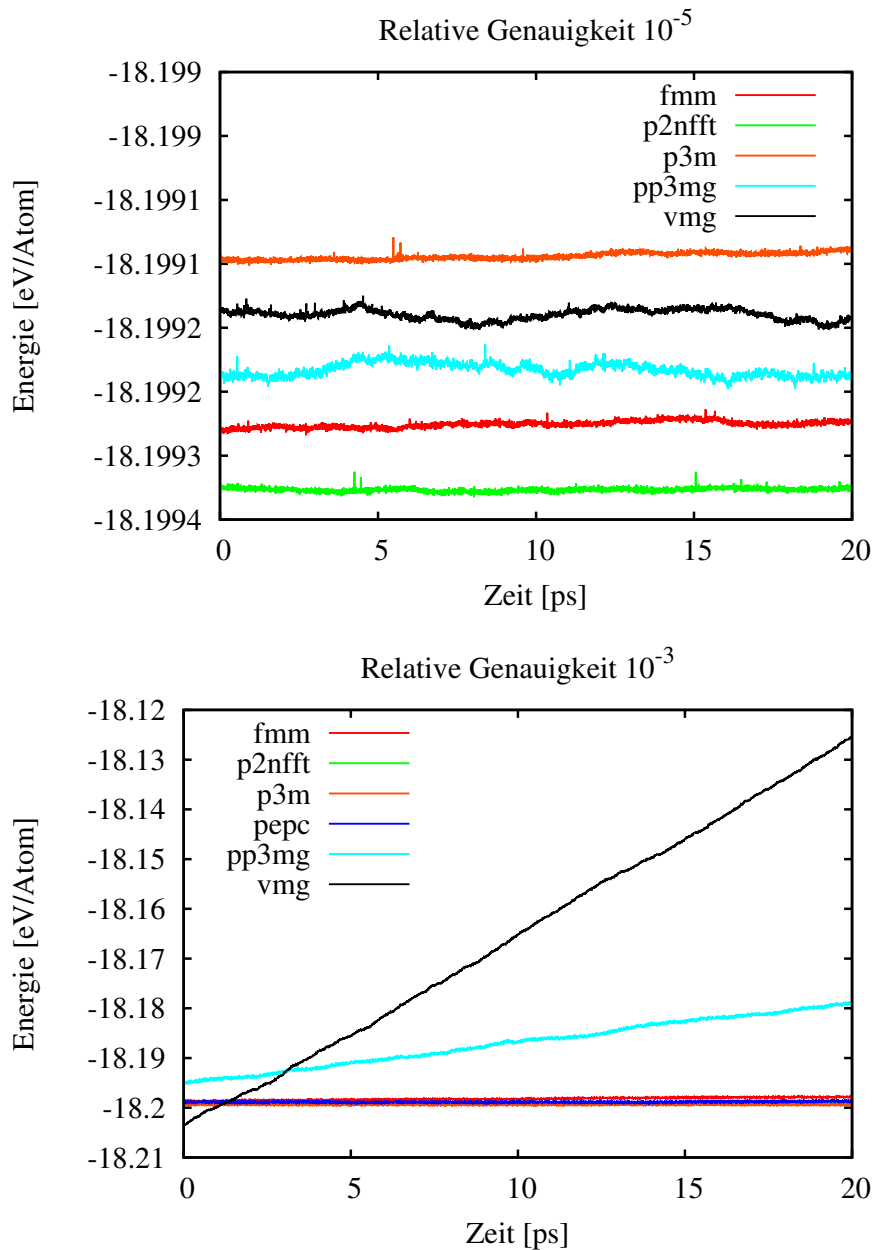


Abbildung 2.9: Gesamtenergieerhaltung der verschiedenen Methoden bei einer relativen Genauigkeit von 10^{-5} (oben) bzw. 10^{-3} (unten). Bei der niedrigen Genauigkeit zeigt sich bei den Mehrgitter-Verfahren PP3MG und VMG ein deutlicher Drift. Die Kurven wurden leicht verschoben, um sie besser auseinander halten zu können. Die Methode PEPC wurde nur bei der kleinen Genauigkeit verwendet, da sie bei höherer Genauigkeit zu langsam wird.

sich, dass bei der größeren Genauigkeit alle Methoden eine hervorragende Energieerhaltung aufweisen, was der Genauigkeit der Kräfte ein sehr gutes Zeugnis ausstellt. Bei einer relativen Genauigkeit von 10^{-3} muss man dagegen differenzieren. Während die meisten Methoden gut damit zurecht kommen, tritt bei den Mehrgitter-Verfahren (PP3MG und VMG) ein deutlicher bis starker Energiedrift auf, was auf systematische Ungenauigkeiten in den Kräften hinweist. Diese Methoden erfordern deshalb eine höhere Genauigkeitsvorgabe, um für eine MD-Simulation eingesetzt werden zu können, was natürlich mit einer erhöhten Rechenzeit verbunden ist.

Ebenfalls geprüft wurde die Impulserhaltung während der Simulation. Um diese zu illustrieren, zeigen wir in Abbildung 2.10 die kinetische Energie der Schwerpunktsbewegung pro Atom, wiederum für eine relative Genauigkeit von 10^{-5} (oben) bzw. 10^{-3} (unten). Es zeigt sich, dass die Methoden FMM, P3M und VMG die Schwerpunktsbewegung perfekt erhalten, wohl aufgrund von Symmetrien im Algorithmus. Die übrigen Methoden weisen Fluktuationen in der Schwerpunktsbewegung auf, wobei diese aber, verglichen mit der totalen kinetischen Energie, in einem akzeptablen Rahmen bleiben. Einzig bei PP3MG und Genauigkeit 10^{-3} kommt man in einen kritischen Bereich, wobei diese Methode bei niedriger Genauigkeitsvorgabe aber schon aufgrund der unzureichenden Energieerhaltung ausfällt.

Anbindung an die Simulationssoftware ESPRESO

Nach der Integration der verschiedenen Algorithmen in die SCAFACOS-Bibliothek wurde die Bibliothek wiederum an die ESPRESSO-Software angebunden[51], welche weltweit zur Simulation von vergrößerten Modellen weicher Materie verwendet wird [93, 63].

Durch diese Anbindung wurden sämtliche Lösermethoden in ESPRESSO verfügbar. ESPRESSO hat einen Schwerpunkt auf elektrostatischen Algorithmen, und verfügt daher über eine Reihe von Testsystemen mit bekannten Ergebnissen. Diese Testsysteme wurden erfolgreich mit den verschiedenen Lösern gerechnet. Damit konnten die Methoden erneut validiert werden. Die Ergebnisse sind konsistent mit den Ergebnissen der Validierung durch IMD und sollen daher hier nicht weiter vertieft werden.

Durch die Anbindung der Bibliothek an ESPRESSO werden zahlreiche neue Methoden zur Berechnung elektrostatischer Wechselwirkungen in der Software verfügbar, und die Möglichkeiten der Software steigen.

Die verschiedenen Löser sind Forschungsgegenstand beim Projektpartner Stuttgart. Neuentwicklungen und Verbesserungen wurden bislang in die Simulationssoftware ESPRESSO eingebaut, getestet und validiert. Die Integration der Algorithmen in die Bibliothek hat daher den Vorteil, dass zukünftige Entwicklungen direkt in der SCAFACOS-Bibliothek stattfinden werden. Dadurch kommen diese Verbesserungen unmittelbar anderen Nutzern der Bibliothek zu Gute.

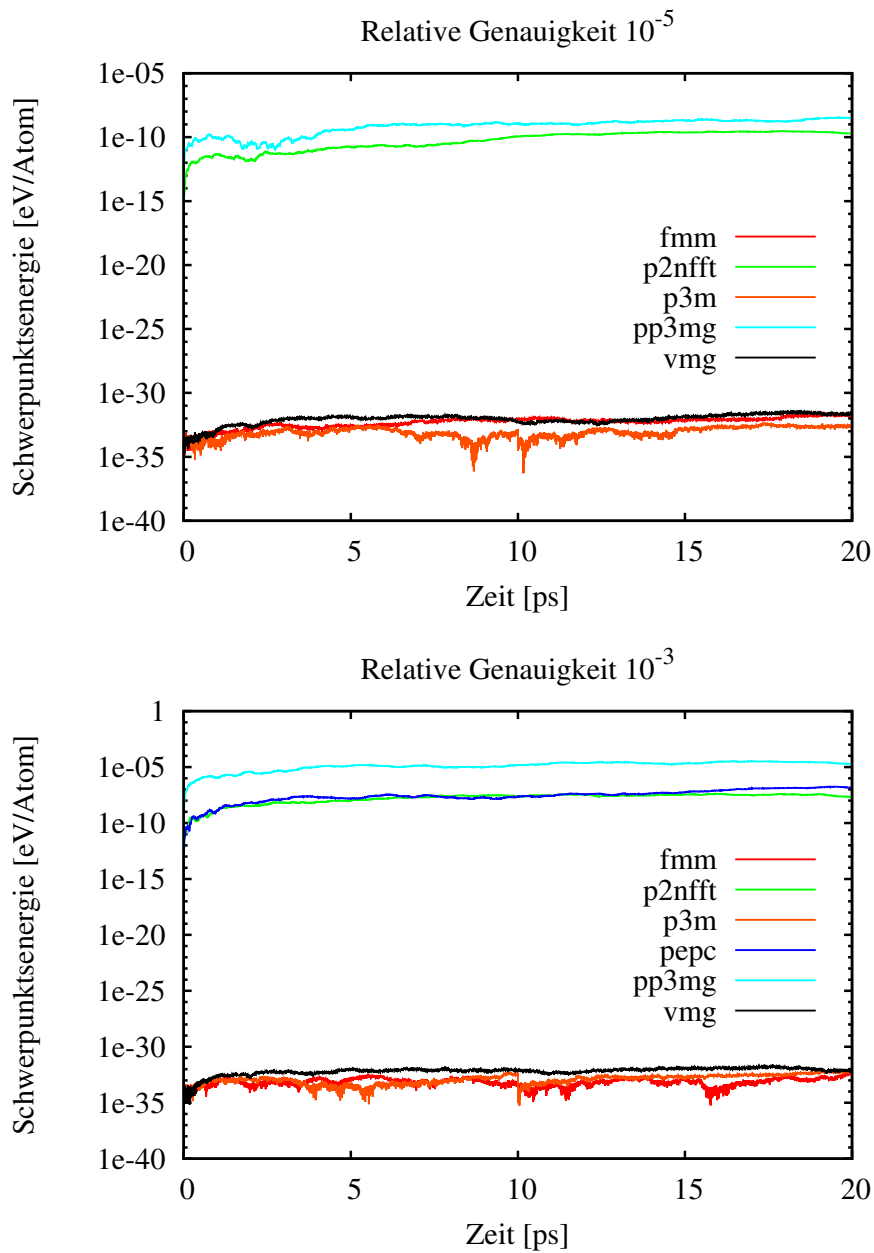


Abbildung 2.10: Kinetische Energie der Schwerpunktsbewegung pro Atom, bei einer relativen Genauigkeit von 10^{-5} (oben) bzw. 10^{-3} (unten). Die Methoden FMM, P3M und VMG erhalten die Schwerpunktsbewegung perfekt, wogegen die anderen Methoden leichte Fluktuationen aufweisen.

Anbindung an die Simulationssoftware MP2C

MP2C, das am JSC entwickelt wird [83], ist ein massiv paralleles Programm, das auf $> 10^5$ Rechenkerne skaliert [105]. Es koppelt ein mesoskopisches Lösungsmittelmodell an eine Molekulardynamik-Simulation und wird für Simulationen im Bereich der weichen Materie angewendet. Das Lösungsmittel wird durch das *Multi Particle Collision Dynamics* (MPC) Verfahren beschrieben, das ein stochastisches, rein lokales und Partikelbasiertes Verfahren darstellt. Die Teilchen, die über Kraftfelder miteinander wechselwirken, werden im Rahmen der Molekulardynamik beschrieben und an das MPC Modell gekoppelt.

Durch Anbindung der SCAFACOS-Bibliothek konnte der funktionelle Umfang von MP2C auf die Klasse von elektrostatischen Wechselwirkungen erweitert werden, wodurch die Beschreibung von Elektrolytlösungen ermöglicht wurde. Als Testsysteme wurden Polymerketten mit Partialladungen gerechnet, was auch für große Systeme noch sehr gute Skalierungseigenschaften zeigte.

Anwendung: Tabakmosaikvirus

In der Biophysik kommt durch das Lösungsmittel Wasser statischen Dipolkräften und damit der langreichweitigen Coulomb-Wechselwirkung eine große Bedeutung zu.

Obwohl Methoden existieren, um das Lösungsmittel nicht explizit in die Moleküldynamiksimulation aufnehmen zu müssen, gibt es immer noch viele Fälle, wo eine explizite Betrachtung unabdingbar ist. Um die Komplexität der Berechnungen dennoch reduzieren zu können, werden Periodizitäten der Strukturen ausgenutzt.

1D-periodische Strukturen existieren in der Biophysik beispielsweise bei dem Tabakmosaikvirus (TMV). Zur Modellierung ist hierbei die explizite Modellierung des Wassers essentiell, da Diffusionsprozesse von hydratisierten Ionen im Innern der TMV-Röhre, aber auch auf der Oberfläche des TMV untersucht werden, siehe [1]. Beispiele für 2D-periodische Strukturen sind Biomembranen. Sie bestehen typischerweise aus einer wenige Nanometer dicken Doppelschicht von Lipiden, in die verschiedene Proteine eingebettet sind.

Nach der Einbindung der SCAFACOS-Bibliothek an die Bonner Moleküldynamiksoftware konnten nun TMVs unter Einbeziehung des langreichweitigen Anteils der Coulomb-Wechselwirkung effizient berechnet werden.

Im Folgenden wurde untersucht, inwieweit sich in der Software bereits vorhandene, interne Coulomb-Methoden wie SPME gegen solche aus der Bibliothek in Bezug auf die Skalierung verhalten. In Abbildung 2.11(a) finden sich die Laufzeitverhalten der einzelnen Methoden über einen Zeitraum von zehn Zeitschritten. Hierbei wurde ein periodisches Segment des Tabakmosaikvirus in Wasser gelöst gerechnet. Das Rechengebiet maß 22,6 nm, beinhaltete ca. 900.000 Atome und wurde über 10 Zeitschritte simuliert.

Das System wurde auf 64 Prozesse unterteilt auf einem Cluster für all jene Löser der Bibliothek gerechnet, welche wie die interne SPME-Methode ebenfalls nach dem Splitting-Verfahren arbeiten. Der nullte Zeitschritt gibt jeweils die Zeit für das Finden optimaler Löser-spezifischer Parameter an.

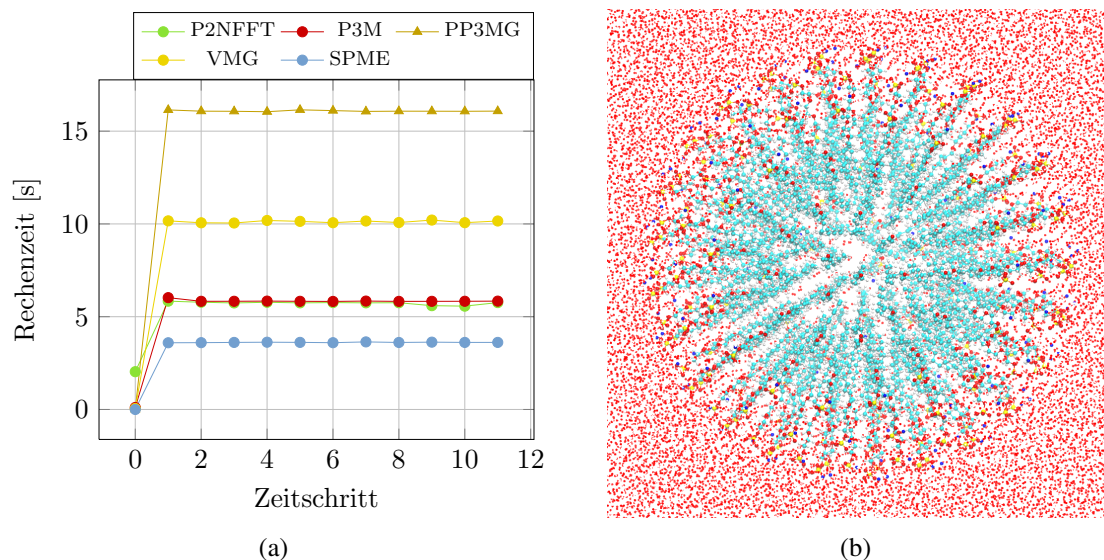


Abbildung 2.11: (a) Laufzeitverhalten pro Zeitschritt der internen Methode SPME gegen gitterbasierte Methoden aus der SCAFACOS-Bibliothek. Schritt Null entspricht der Initialisierung. (b) Mizelle aus 200 SLES-Molekülen in Wasser gelöst.

Dabei ist zu erkennen, dass die von der Bibliothek zur Verfügung gestellten Löser nur wenig gegenüber der internen und sehr stark optimierten SPME-Methode verlieren. Die Mehrarbeit, welche durch das zusätzliche Kopieren, bedingt durch das Interface, entsteht, fällt also nicht entscheidend ins Gewicht. Demgegenüber steht die Ersparnis in der Implementation, welche besonders gewichtig wird, wenn auf Systemen mit sehr vielen Kernen noch effizient gerechnet werden soll. Dabei hätte die Arbeit für dieses kleinere Tabakmosaikvirussystem schon auf bis zu 4096 Kernen unterteilt werden können.

Anwendung: Tenside

Tenside spielen eine wichtige Rolle in den Geschäftsfeldern des Industrie-Partners Cognis (jetzt BASF). Sie werden in zahlreichen Produkten eingesetzt. Die Strukturen und die Dynamik der Tenside in wässriger Lösung und an Phasengrenzen zu Luft bestimmen maßgeblich die praxisrelevanten Eigenschaften dieser Produkte, daher sind numerische Simulationen zum besseren Verständnis dieser Systeme relevant. Es ist bekannt,

dass Tenside über einer bestimmten Konzentration sogenannte Mizellen bilden, die sich allerdings in Form, Größe und Austauschdynamik abhängig vom Tensidtyp, der Konzentration und, bei ionischen Tensiden, von der Salzmenge unterscheiden.

Im Rahmen dieses Arbeitspakets wurde zunächst ein geeignetes kurzreichweitiges Potential gesucht, welches das Tensid-Molekül des Projektpartners beschreiben kann. Die Wahl fiel auf eine Parametrisierung mittels der CHARMM-Potentiale. In einer Reihe von Treffen mit Dr. Matthias Hloucha (Cognis) wurden diese Simulationsergebnisse besprochen. Zunächst wurde untersucht, ob sich in der Simulation aus einer wahllosen Anordnung von Tensid-Molekülen eine stabile Mizellen-Konfiguration von selbst bildet. Die mit der Moleküldynamik möglichen Zeitskalen auf mittleren HPC-Systemen im Bereich von einigen Nanosekunden sind jedoch nicht ausreichend. Deswegen wurden anschließend Programme geschrieben, um solche Mizellen-Strukturen direkt zu erzeugen. Diese konstruierte Struktur wurden dann auf ihre Stabilität in der Simulation über einen gewissen Zeitraum geprüft. Beispielhaft sei in Abbildung 2.11(b) die Visualisierung einer Konfiguration bestehend aus 200 Sodium Laureth Sulphate (SLES)-Molekülen in Wasser gezeigt. Mit den verwendeten CHARMM-Potentiale zerfiel die Mizellen-Struktur jedoch nach sehr kurzer Zeit, unabhängig von der Anwesenheit des Lösungsmittels Wasser. Grundsätzliches Problem scheint eine auf diesen Anwendungsfall nicht übertragbare Parametrisierung der van der Waals-Wechselwirkungen zwischen den langen Kohlenwasserstoffketten der Tensiden zu sein. Eine sinnvolle Rechnung lässt sich hier nur mit einem wirksamen Potential durchführen. Dies ist unabhängig von der langreichweitigen Wechselwirkung, deren Berechnung mit der SCAFACOS-Bibliothek sich für diesen Fall als genauso effizient erwies wie für den TMV.

2.1.5 Erzielte Fortschritte in der entwickelten Löser-Bibliothek

Fortschritt Interface

- Entwicklung eines einheitlichen Interfaces für alle Methoden
- Bereitstellung des Interfaces in C und Fortran 2003
- Möglichkeit der Verwendung von Standardparametern für alle Methoden, die eine einfache Nutzung erlauben
- Interner Nahfeldlöser zur Berechnung von Nahfeldanteilen (benötigt von einigen Methoden)
- Berechnung einer gegebenenfalls nötigen Dipolkorrektur möglich
- Einfache Nutzung durch Reduktion der benötigten Befehle auf ein Minimum (6 notwendige Befehle zur Verwendung der Bibliothek bei Benutzung der Standardparameter)

Fortschritt Parallele Sortierung

- Anforderungsanalyse von Coulomb-Solvern und Partikelanwendungen
- Neuer Partitionierungsalgorithmus für paralleles Sortieren basierend auf Partitionieren
- Neuer In-place-Algorithmus für `MPI_Alltoallv`
- Neue MPI-Operation zur feingranularen Datenumverteilung in Partikelcodes
- Parallele Partikelsortierung in FMM, P3M, P2NFFT, PEPC, MEMD und Nahfeldlöser
- Adaptives Tuning für Sortierbibliothek und FMM

Fortschritt P2NFFT

- Neue Implementation einer hochskalierenden, flexiblen, parallelen FFT-Softwarebibliothek (PFFT) [14]
- Erste In-place-Implementation einer hochskalierenden parallelen FFT
- Erste öffentlich verfügbare FFT-Softwarebibliothek zur Berechnung vier- und mehrdimensionaler FFTs mit zwei-dimensionaler Datenaufteilung
- Erstmalige Implementation einer hochskalierenden parallelen NFFT-Softwarebibliothek (PNFFT) [15]
- Erstmalige parallele Implementation der schnellen Summation (P2NFFT) für nicht-periodische Randbedingungen
- Erweiterung der schnellen Summation auf periodische Randbedingungen und nicht-kubische Boxen
- Erstmalige modularisierte Implementation einer parallelen Gradienten-NFFT liefert optimierte Berechnung der Kräfte in der schnellen Summation
- Verbessertes Loadbalancing in der parallelen NFFT und der schnellen Summation durch Anpassung der parallelen FFT
- Neuartige Regularisierung liefert verbesserte Genauigkeit der nicht-periodischen schnellen Summation
- Einheitliche Schnittstelle innerhalb der SCAFACOS-Bibliothek

Fortschritt P3M

- Die Implementation des Algorithmus in der Simulationssoftware ESPRESSO wurde modularisiert und als Bibliotheksfunktion aus der Software herausgelöst.
- Die Implementation wurde in die Bibliothek eingefügt.
- Die Implementation wurde zum besseren Lesbarkeit und Modularität refaktoriert.
- Die Parallelisierung wurde überarbeitet und an die Bibliothek angepasst.
- Die Implementation wurde auf nicht-kubische Geometrien erweitert.

- Die Methode der Wechseltechnik wurde wiederentdeckt, die den Algorithmus beschleunigt.
- Die verschiedenen Komponenten des Algorithmus (Funktion zur Ladungsverstärkung, Influenzfunktion, Wechseltechnik, Typ der Kraftberechnung) wurden systematisch im Hinblick auf Genauigkeit und Geschwindigkeit untersucht. Die optimale Kombination ist ein Algorithmus mit P3M-Influenzfunktion, Wechseltechnik und analytischer Ableitung (P3M-ad-i).
- Die analytischen approximativen Fehlerformeln für das neue Verfahren P3M-ad-i wurden hergeleitet.
- Die P3M-Implementation in der Bibliothek wurde auf das neue Verfahren P3M-ad-i umgestellt.
- Jegliche Weiterentwicklung des Algorithmus und der Implementation findet in Zukunft direkt in der Bibliothek statt.
- In Zusammenarbeit mit dem Projektpartner in Chemnitz wurde festgestellt, dass der P3M-Algorithmus und die vollperiodische Ewald-Summation mit Hilfe der nichtäquidistanten schnellen Fourier-Transformation mathematisch äquivalent sind [2].

Fortschritt Dipolarer P3M

- Die Implementation des Algorithmus in der Simulationssoftware ESPRESSO wurde modularisiert und als Bibliotheksfunktion aus der Software herausgelöst.
- Die Implementation wurde zum besseren Lesbarkeit und Modularität refaktoriert.
- Die bisher sequentielle Implementation wurde nach dem Vorbild des P3M für Monopole parallelisiert.
- Da die Methode ein anderes Interface erfordert, als alle anderen Methoden, wird sie in einer eigenständigen Bibliothek veröffentlicht, deren Interface jedoch an die SCAFACOS-Bibliothek angelehnt ist.

Fortschritt MMM1D und MMM2D

- Die Implementierungen des Algorithmus in der Simulationssoftware ESPRESSO wurden modularisiert und als Bibliotheksfunktionen aus der Software herausgelöst.
- Die Implementierungen wurde zum besseren Lesbarkeit und Modularität refaktoriert.
- Die Implementierungen wurden in die Bibliothek eingefügt.
- Die Parallelisierungen wurde überarbeitet und an die Bibliothek angepasst.

Fortschritt FMM

- Vollständige einseitige, nicht-blockierende Parallelisierung der FMM

- Lastbalancierung mittels raumfüllender Kurven zur Partitionierung
- Erweiterung der FMM auf 1d- und 2d-periodische Randbedingungen
- Erweiterung des Fehlerschätzers auf 1d- und 2d-periodische Randbedingungen
- Reduzierung des Speicherbedarfs auf bis zu 43 Bytes pro Teilchen bei 300k Prozessoren
- Implementierung eines zusätzlichen analytischen Fehlerschätzers
- Erweiterung des analytischen Fehlerschätzers auf 1d- und 2d-periodische Systeme
- Skalierung auf bis zu 300k BG/P Prozessoren mit bis zu 3 Billionen Teilchen
- Einbindung der FMM in eine einheitliche Schnittstelle innerhalb der SCAFACOS Bibliothek

Fortschritt PEPC

- Implementierung einer skalierbare Teilchensortierung mit dynamischer Lastverteilung
- Erweiterung der Gebietszerlegung auf verschiedenen raumfüllenden Kurven mit verbesserter Lokalitätseigenschaften
- Erweiterung des Barnes-Hut Algorithmus auf nicht kubische, periodische Randbedingungen
- Reduzierung des Speicherbedarfs mit Hilfe von Zusammenlegung von Endknoten bei der Gebietszerlegung
- Minimierung von Branchknoten mit Hilfe eines Filteralgorithmus
- Implementierung eines hybriden, voll asynchronen MPI-POSIX threads Baumdurchlaufs
- Lastbalancierte Skalierung bis 300k BG/P Kernen
- Vollständige Umstrukturierung des PEPC-Frameworks, der nun aus dem Bibliothekskernel, unabhängigen Anwendungen und wechselwirkungs-spezifischen Multipolstrukturen (Coulomb, Gravitation, Fluid-Vortex) besteht
- Anwendungen in den Bereichen Plasmaphysik und Astrophysik

Fortschritt VMG

- Neue Implementation einer hochskalierenden, flexiblen, parallelen geometrischen Mehrgitter-Bibliothek (VMG)
- Objekt-orientierte und stark modularisierte Implementation der Mehrgitter-Bibliothek (VMG):
 - Vollständige Abstraktion des Parallelisierungsmodells, des Gebietszerlegungsmodells und des Mehrgitterzyklus
 - Speicherverwaltung/Garbage Collection mittels Registry und Singleton-Pattern
 - Verschiedene Grobgitterlöser: LAPACK und alternativ Givens-Rotation

– Verschiedene Glätter: Gauss-Seidel, Gauss-Seidel Red/Black, Jacobi

- Visualisierung von rechter Seite, Zwischenschritten und finaler Lösung mittels VTK
- Optimierung der Routinen mittels Loop-Unrolling und Perfect Forwarding
- Verwendung von asynchroner Kommunikation, u. a. im Gauss-Seidel Red/Black-Glätter
- Testsuite zur Überprüfung einzelner Komponenten
- Implementation von hochskalierenden Verfahren zur Behandlung von offenen und quasi-periodischen Systemen mit Finite-Volumen-Diskretisierung mittels Full Approximation Scheme (FAS)
- Anbindung an ab-initio-Verfahren zur Lösung des langreichweitigen Teils der elektronischen Schrödingergleichung

Fortschritt PP3MG

- Das vorhandene Verfahren wurde komplett nach C portiert.
- Das Verfahren für periodische Randbedingungen wurde zu Bibliotheksreife entwickelt und in die SCAFACOS-Bibliothek eingebunden.
- Ein Lastverteilungsalgorithmus wurde im Rahmen einer Masterarbeit entwickelt [57].
- Erweiterungen für 1D- und 2D-periodische Systeme wurden gemeinsam mit der Universität Bonn entworfen und diskutiert, dies diente als Grundlage für die Arbeiten an VMG.
- Im Rahmen einer Masterarbeit [57] wurden Diskretisierungen mittels finiter Elemente und die dazugehörigen Fehlerschranken untersucht.
- Zwei verschiedene Parallelisierungsansätze für den Löser für offene Randbedingungen wurden in einer Masterarbeit [58] konzeptuell untersucht und der erfolgversprechendere Ansatz wurde im Rahmen dieser Arbeit implementiert.

Fortschritt MEMD

- Das Randelementverfahren durch Berechnung von dielektrischen Einschlüssen wurde im Programmpaket ESPRESSO implementiert und verifiziert.
- Der dynamische Teil des MEMD Algorithmus wurde auf lokal variierende dielektrische Konstanten erweitert.
- Die statische Initiallösung des Algorithmus wurde durch eine Methode der numerischen Relaxation ersetzt, um lokale Permittivitäten einzubinden.
- Eine Methode der direkten Subtraktion wurde entwickelt, um die Selbstenergie-Wechselwirkung im Algorithmus zu korrigieren.

- Der Algorithmus wurde im Programmpaket ESPRESSO vollständig neu implementiert.
- Der Algorithmus wurde mit einer gleichmäßigen Gebietszerlegung parallelisiert.
- Mit Hilfe des Randelementverfahrens und einiger theoretischer Beispiele wurde die Korrektheit des statischen Initiallösers und des dynamischen Teiles verifiziert.
- Es wurde eine Fehlerabschätzung für den Algorithmus entwickelt und auf ihr aufbauend eine Tuningroutine geschrieben.
- Der Code wurde modularisiert und auf das inzwischen fertige SCAFACOS-Interface portiert.

2.2 Wichtigste Positionen des zahlenmäßigen Nachweises

2.2.1 Nachweis Universität Bonn

Unterprojektpartner

Die Bergische Universität Wuppertal war als Unterauftragnehmer der Rheinischen Friedrich-Wilhelms-Universität Bonn tätig.

Projektbearbeiter

1. Ralf Wildenhues: 20 Monate x 100% = 20 FTE (Build-System der SCAFACOS-Bibliothek, Optimierung von PP3MG und Einbau in Bibliothek, Entwicklung Mehrgitterverfahren für quasiperiodische Systeme)
2. Frederik Heber: 8 Monate x 100% = 8 FTE (Anbindung SCAFACOS an Bonner Moleküldynamiksoftware, Implementation zur Mizellengeometrieerstellung, Simulationen mit Tensiden und Tabakmosaikvirus)
3. Daniel Wissel: 3 Monate x 100% = 3 FTE (Tensidpotenziale und erste Tests)
4. Michael Ankele: 2 Monate x 50% = 1 FTE (Implementation zur Mizellengeometrieerstellung)
5. Julian Iseringhausen: HiWi-Stelle (Build-System und Implementation VMG, Implementation und Optimierung Mehrgitterverfahren für quasiperiodische Systeme)
6. Thomas Müller: HiWi-Stelle (Interlaced Grid Method, Optimierung der Datenstrukturen)
7. Gregor Bollerhey: HiWi-Stelle (Anbindung SCAFACOS an Bonner Moleküldynamiksoftware)
8. Daniel Alm: HiWi-Stelle (Test der Greensfunktion bei quasiperiodischen Systemen in MatLab)
9. Nadine Pastuzka: HiWi-Stelle (Durchführung von Tensidsimulationen)

Reisemittel

- Projektbesprechungen in Jülich, Stuttgart, Chemnitz, Bonn (Ralf Wildenhues, Daniel Wissel, Frederik Heber, Julian Iseringhausen)
- BMBF-Statustagung, Berlin, 2010 (Ralf Wildenhues)
- BMBF-Statustagung, Darmstadt, 2011 (Frederik Heber)

2.2.2 Nachweis Universität Chemnitz

Projektbearbeiter

1. Michael Pippig: 35 Monate x 100% + 2 Monate x 50 % = 36 FTE (parallele FFT, parallele NFFT, parallele schnelle Summation für nicht-periodische Daten)
2. Michael Hofmann: 16 Monate x 100 % = 16 FTE (parallele Sortier- und Datenumverteilungsoperationen, Sortierbibliothekskonzept)
3. Jens Lang: 25 Monate x 75% = 18,75 FTE (Buildsystem der SCAFACOS-Bibliothek, generisches Testprogramm, Nahfeldlöser, adaptives Tuning für Sortierbibliothek und FMM)
4. Lutz Kämmerer: 9 Monate x 75% = 6,75 FTE (Buildsystem der PFFT und PNFFT-Bibliothek)
5. Toni Volkmer: 1 Monat x 75% = 0,75 FTE (Buildsystem der PFFT und PNFFT-Bibliothek)
6. Sebastian Banert: HiWi-Stelle (Evaluation der parallelen FFT, Parallele schnelle Summation für 3d-periodische Daten)
7. Alexander Köwitsch: HiWi-Stelle (Erweiterung der P2NFFT auf nicht-kubische Geometrien)
8. Franziska Nestler: HiWi-Stelle (Approximationsverfahren zur schnellen Energieberechnung in Partikelsystemen)

Reisemittel

- HPC-Statuskonferenz der Gauß-Allianz e.V., Schwetzingen, 2010, Vortrag (Michael Pippig)
- WE Heraeus Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 2010, Vorträge (Michael Pippig, Daniel Potts)
- BMBF-Statuskonferenz, Berlin, 2010, Zuarbeiten durch Michael Hofmann und Michael Pippig zum Projektvortrag (gehalten von Godehard Sutmann)
- Arbeitstreffen, Stuttgart, 2011, Vortrag (Michael Pippig)
- SIAM Conference on Computational Science and Engineering, Reno, Nevada, 2011, Vortrag (Michael Pippig)
- CECAM Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 2011, Vorträge (Michael Hofmann, Michael Pippig, Daniel Potts)
- DMV Jahrestagung 2011, Köln, 2011, Vortrag (Michael Pippig)
- HPC-Statuskonferenz, Darmstadt, 2011, Zuarbeiten durch Michael Hofmann und Michael Pippig zum Projektvortrag (gehalten von Godehard Sutmann)
- 22. Rhein-Ruhr-Workshop, Bestwig, 2012, Vortrag (Michael Pippig)
- 83. Jahrestagung der Gesellschaft für Angewandte Mathematik und Mechanik (GAMM 2012), Darmstadt, 2012, Vortrag (Michael Pippig)

- 2012 SIAM Conference on Applied Linear Algebra, Valencia, Spanien, 2012, Vortrag (Michael Pippig)
- 7th International Workshop on Parallel Matrix Algorithms and Applications (PMAA 2012), London, UK, 2012, Vortrag (Michael Pippig)
- Arbeitstreffen in Stuttgart, 2012 (Michael Pippig)
- SCAFACOS-Meetings (Michael Hofmann, Michael Pippig, Daniel Potts, Gudula Rünger):
 - Kickoff-Meeting, Jülich, Februar 2009
 - Projektbesprechung, Jülich, Juni 2009
 - Projektbesprechung, Bonn, Januar 2009
 - Projektbesprechung, Stuttgart, Februar 2011
 - Projektbesprechung, Jülich, Mai 2011
 - Projektbesprechung, Stuttgart, Juli 2011
 - Scaling-Workshop, Jülich, Februar 2012
 - Projektbesprechung, Jülich, Juni 2012

2.2.3 Nachweis Jülich Supercomputing Centre

Projektbearbeiter

1. Rene Halver: 28×100 % FTE (Interface, Nahfeldlöser, Benchmarks und Anwendungen, Integration in MD-Codes)
2. Ivo Kabadshow: 25×100 % FTE (Koordination, FMM)
3. Lidia Westphal: 16×100 % FTE (Interface, Baum-Algorithmus, Nahfeldlöser, Dokumentation)
4. Lukas Arnold: 5×100 % FTE (Baum-Algorithmus)
5. Robert Speck: eigene Doktorandenstelle (Baum-Algorithmus, Vortex-Anwendung)
6. Mathias Winkel: eigene Doktorandenstelle (hybrider Baum-Algorithmus)
7. Andreas Breslau: HiWi-Stelle (Gravitation PEPC)
8. Helge Hübner: HiWi-Stelle (Raumfüllende Kurven im Baum-Algorithmus)

Reisemittel

- SIAM Conference on Computational Science and Engineering, Reno, 2011, Vortrag (Ivo Kabadshow, Holger Dachsel)
- ScicomP 17, Paris, 2011, Vortrag (Ivo Kabadshow)
- CECAM Workshop, Lausanne, 2011, Vortrag (Ivo Kabadshow)
- SIAM Conference on Computational Science and Engineering, Miami, 2009, Vortrag (Ivo Kabadshow)

- Conference on Computational Physics (CCP), Trondheim, 2010, Poster (Rene Halver)
- Arbeitstreffen in Chemnitz, 2011 (Rene Halver)
- SCAFACOS-Meetings (Holger Dachsels, Lidia Westphal, Rene Halver, Ivo Kabadshow, Godehard Sutmann)
 - Projektbesprechung, Bonn, 2009
 - Projektbesprechung, Chemnitz, 2010
 - Projektbesprechung, Stuttgart, 2010
 - Projektbesprechung, Chemnitz, 2011
 - Projektbesprechung, Stuttgart, 2011
- BMBF-Statustagung, Berlin, 2010 (Godehard Sutmann)
- BMBF-Statustagung, Darmstadt, 2011 (Godehard Sutmann, Rene Halver)

2.2.4 Nachweis Universität Stuttgart

Projektbearbeiter

1. Alexey Neelov: 7 Monate x 100% = 7 FTE (Entwicklung der Wechsellmethode)
2. Florian Fahrenberger (geb. Rühle): 35 Monate x 75% = 26,25 FTE (Entwicklung, Implementation und Parallelisierung der lokalen Gittermethode MEMD)
3. Owen Hickey: 10 Monate x 10% = 1 FTE (Programmierunterstützung für Alexey Neelov)
4. Olaf Lenz: 17 Monate x 100% = 17 FTE (Modularisierung, Portierung und Parallelisierung der Teilchen-Gitter-Methode P3M)
5. Florian Dommert: 7 Monate x 100% = 7 FTE (Implementation der Randelement-Methode ICC*)
6. Pedro Sanchez: 4 Monate x 75% = 3 FTE (Modularisierung und Portierung der zusätzlichen Methoden MMM*D aus ESPRESSO)
7. Philipp Beck: 6 Monate x 100% = 6 FTE (Anbindung der SCAFACOS-Bibliothek an die Simulationstool IMD und ESPRESSO)
8. Marcello Sega: 1 Monat x 100% = 1 FTE (Verifizierung der portierten Elektrostatik-Algorithmen aus ESPRESSO)
9. Florian Häußermann: HiWi-Stelle (Drucktensor für P3M implementiert)
10. Marco Klinkigt: HiWi-Stelle (Theorie zu Dipolen in P3M entwickelt)
11. Tetyana Kromer-Vovk: HiWi-Stelle (Vergleichsrechnungen der Bibliothek mit anderen Implementationen)
12. Dominic Röhm: HiWi-Stelle (Optimierung der Kommunikation von lokalen Gittermethoden)
13. Konrad Breitsprecher: HiWi-Stelle (Implementationsdetails überprüfen)

14. Tobias Eipperle: HiWi-Stelle (Vergleichsrechnungen mit ESPRESSO)
15. Wojciech Müller: HiWi-Stelle (Optimierung und Korrektur des ICC* Algorithmus)
16. Georg Rempfer: HiWi-Stelle (Theorie zu lokalen Gittermethoden)
17. Florian Weik: HiWi-Stelle (Tuningroutine für P3M optimiert)
18. Alexander Schlaich: HiWi-Stelle (Vergleich der Methoden mit theoretischen Voraussagen)

Reisemittel

- MolSim, Amsterdam, 2010, Poster (Florian Fahrenberger)
- SIAM Conference on Computational Science and Engineering, Reno, 2011, Vortrag (Florian Fahrenberger)
- Summer School for Long-range Interactions, Jülich, 2010-2011, Vorträge (Axel Arnold, Olaf Lenz, Florian Fahrenberger, Florian Weik, Nadya Grivoba, Alexander Schlaich, Christian Holm)
- SCAFACOS-Projekttreffen, verschiedene Städte, 2009-2012 (Axel Arnold, Olaf Lenz, Florian Fahrenberger, Alexey Neelov, Franz Gähler, Andreas Michel, Pedro Sanchez, Philipp Beck)
- Vortrag und Hilfe mit Implementation der Algorithmen, Stuttgart, 2010 (Vincent Ballenegger)
- Nanopores Conference, Arrefice/Lanzarote, 2012 (Stefan Kesselheim)
- Vortrag „Electostatics in systems with dielectric contrasts“, Wien 2011 (Marcello Sega)
- SimBioMa, Konstanz 2011 (Christian Holm)
- DPG Frühjahrstagung, Dresden 2011 (Alexander Schlaich, Marco Klinkigt, Axel Arnold)
- DUNE-Kurs, Heidelberg 2011 (Alexander Schlaich)
- Molecular Modelling of Thermophysical Properties, Dortmund 2011 (Christian Holm)
- Meeting on Scalable Coulomb Solvers, Stuttgart 2011 (Vincent Ballenegger)
- HPC Challenges in Computational Sciences, Catania 2010 (Dominic Röhm)
- BMBF Statustagung "HPC-Software für skalierbare Parallelrechner", Berlin 2010 (Christian Holm, Olaf Lenz)

2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Partnerübergreifend musste für die Bibliothek ein gemeinsames Interface entwickelt werden. Hierfür fand ausführliche Kommunikation zwischen den einzelnen Projektpartnern statt und es wurde sich auf eine Struktur geeinigt. Hierfür war im Projektantrag keine Arbeitszeit vorgesehen, das Unterfangen erwies sich jedoch als weitreichender als erwartet. Nicht nur musste das Interface für alle Methoden und verschiedene Programmiersprachen entwickelt werden, was federführend von den Partnern **Jülich** (Interface), **Bonn** (Build system), **Chemnitz** (gemeinsame Funktionen) und **Stuttgart** (Testsystem) vorangetrieben wurde. Alle einzelnen Methoden mussten zudem ihre Schnittstelle, ihre Datenstrukturen und ihre Parallelisierung an das Interface anpassen. Das erzeugte für alle Projektpartner zusätzliche Arbeit und ist einer der Gründe für die beantragte kostenneutrale Verlängerung des Projektes.

Die Arbeit der **Universität Bonn** wurde zunächst von einem Doktoranden im Bezug auf die Umwandlung des vorhandenen Mehrgitter-Lösers und die Entwicklung der Behandlung von quasi-periodischen Randbedingungen durchgeführt und in Teilen von Hilfskräften ausgeführt. Durch den nicht vorhersehbaren Weggang des Doktoranden in die freie Wirtschaft wurde eine Lücke geschlagen, die im Anschluss überbrückt werden musste. Eine Hilfskraft war weiterhin mit der Implementation des neuen Mehrgitter-Lösers betraut. Der nachfolgende Doktorand kümmerte sich vorallem um die Fertigstellung dieses neuen Mehrgitterlösers und die damit verbundene Koordination der Hilfskräfte. Da ein Teil des Arbeitspakets auch die Simulation in industrienahen Problemfelder war, wurde vorallem auch die Anbindung an die Bonner Moleküldynamiksoftware weiter vorangetrieben und getestet. Bei diesen großen Problemstellungen ergaben sich jedoch besondere Schwierigkeiten, stabile Ausgangsgeometrien aufzusetzen, womit eine wissenschaftliche Hilfskraft beauftragt wurde.

An der **TU Chemnitz** hat ein Doktorand durchgängig an der Entwicklung und Implementierung paralleler hochskalierbarer Algorithmen für die schnelle Fourier-Transformation, die nichtäquidistante schnelle Fourier-Transformation und die schnelle Summation gearbeitet. Neben der Entwicklung der parallelen Software-Module wurden zudem wichtige theoretische Fortschritte, z.B. verbesserte Approximationsgüte und Erweiterung der schnellen Summation auf periodische Randbedingungen, erzielt. Die beiden parallelen Software-Module für schnelle Fourier-Transformationen wurden zusätzlich als selbstständige Bibliotheken veröffentlicht. Für die Entwicklung und Verbesserung des Bibliothekskonzeptes wurden zeitweilig zwei weitere Doktoranden beschäftigt. Der große Implementierungsaufwand konnte nur durch die Auslagerung kleiner Arbeitspakete an drei zeitweilig beschäftigte studentische Hilfskräfte geleistet werden. Die Arbeiten zur parallelen Sortierung innerhalb der SCAFACOS-Bibliothek und ihrer Löser wurde von zwei

weiteren Doktoranden durchgeführt. Eine thematische und arbeitsteilige Gliederung ergab sich dabei wie folgt: Zum einen wurde die Entwicklung und Umsetzung von neuen Algorithmen zur parallelen Sortierung und Datenumverteilung sowie deren Integration in ein weiterentwickeltes Sortierbibliothekskonzept gebündelt. Zum anderen wurden die in enger Zusammenarbeit mit den anderen Projektpartnern durchgeführten Arbeiten (Buildsystem der SCAFACOS-Bibliothek, generisches Testprogramm, Nahfeldlöser) sowie alle Arbeiten hinsichtlich des adaptiven Tunings von Implementierungen zur Performancesteigerung zusammengefasst.

Zur Parallelisierung und Erweiterung der **FMM** haben zwei Mitarbeiter des JSC und ein Master-Student beigetragen.

An der Entwicklung des **Jülicher** Baumalgorithmus PEPC haben während der Projektlaufzeit insgesamt drei Doktoranden, zwei Master-Studenten und zwei JSC-Mitarbeiter beigetragen, finanziert überwiegend aus anderen Drittmittelquellen. Hierbei konnte die starke Synergie zwischen diesen anwendungsorientierten Projekten und dem SCAFACOS-Projekt optimal genutzt werden.

An der **Universität Stuttgart** wurden einerseits die Implementationen der Methoden P3M, MMM1D, MMM2D und ELC aus der Simulationssoftware ESPRESSO modularisiert, herausgelöst, refaktoriert und an die neue Bibliothek angepasst. Neben der Anpassung der Methoden an das gemeinsame Interface erforderte dies zusätzlich auch die Anpassung der Parallelisierungsmethode. Schließlich mussten Benchmarktests für die verschiedenen Methoden durchgeführt werden.

Damit die Simulationssoftware ESPRESSO selbst auch die verschiedenen Verfahren der SCAFACOS-Bibliothek verwenden konnte, wurde sie an die Software angebunden. Mit Hilfe der verschiedenen Testsysteme wurde die Effizienz und Genauigkeit der Methoden überprüft.

Darüberhinaus wurde insbesondere die P3M-Methode algorithmisch weiterentwickelt. Beim *Wechselverfahren* handelt es sich um eine neue Idee zur Beschleunigung des Algorithmus, welches in einer internationalen Fachzeitschrift veröffentlicht wurde. Die Implementation des Algorithmus im Rahmen der Bibliothek ist die erste Implementation, die dieses Verfahren verwendet. Desweiteren wurden die einzelnen Komponenten des Algorithmus systematisch im Hinblick auf die Genauigkeit und Geschwindigkeit untersucht, und damit die beste Kombination der Komponenten bestimmt (P3M-ad-i). Dieser neue, optimierte Algorithmus wurde in die Bibliothek aufgenommen. Neben der eigentlichen Implementation des Algorithmus war es dazu auch notwendig, die dazugehörigen approximative analytischen Fehlerformeln zu entwickeln und das automatische Tuning anzupassen.

Obwohl die Implementation des P3M-Algorithmus aus der Simulationssoftware ESPRESSO zur Berechnung von Wechselwirkungen zwischen punktförmigen Dipolen nicht in die Bibliothek mit aufgenommen wurde, so wurden dennoch sämtliche dafür notwendigen Arbeiten durchgeführt: Modularisierung, Herauslösung und Refaktorisierung. Des-

weiteren wurde die Implementation parallelisiert. Die Implementation eignet sich daher für eine eigenständige Bibliothek, und wird auch als solche veröffentlicht.

Schließlich wurde ein lokales Gitterverfahren entwickelt. Die besonderen Vorteile von lokalen Gitterverfahren und insbesondere der „Maxwell Equations Molecular Dynamics“ (MEMD) sind zum einen die gute Parallelisierungsmöglichkeit, zum anderen die Option, lokal variierende dielektrische Eigenschaften festzulegen. Da uns zum damaligen und auch jetzigen Zeitpunkt aber keine implementierte Methode zur Verfügung stand, die Vergleichsrechnungen zu solchen Systemen anstellen konnte, wurde das iterative Randelementverfahren im Programmpaket ESPRESSO implementiert. Für wissenschaftliche Arbeit ist die Überprüfung neuer Algorithmen und damit verknüpft die Möglichkeit, Vergleichsrechnungen durchzuführen, unabdingbar und trägt maßgeblich zum Vertrauen in den Algorithmus und damit zu seiner Nutzbarkeit bei.

Gleichsam sind eine Fehlerabschätzung und eine Tuning-Methode bei nicht besonder weit verbreiteten Verfahren sehr hilfreich, um die Anwendbarkeit zu erleichtern und voranzutreiben. Dementsprechend wurde beides für den Algorithmus MEMD entwickelt.

Die Entwicklung des Algorithmus selber gestaltete sich schließlich schwieriger als erwartet, da sowohl die Initiallösung als auch die Selbstenergie-Wechselwirkung der ursprünglichen Methode für variierende Permittivitäten nicht mehr zu verwenden waren und neu entwickelt werden mussten. Dies war aber für einen Erfolg des Algorithmus notwendig, um den wissenschaftlichen Vorsprung und eine breitestmögliche Anwendbarkeit zu gewährleisten. Die statische Initiallösung und eine Korrektur des Selbstwechselwirkung wurden dementsprechend entwickelt und implementiert.

2.4 Voraussichtlicher Nutzen, insbesondere Verwertbarkeit der Ergebnisse im Sinne des fortgeschriebenen Verwertungsplans

Neue Rechnerarchitekturen werden in Zukunft eine stetig wachsende Zahl von Rechenkernen aufweisen. Eine effiziente Nutzung dieser Architekturen wird daher nur durch angepasste Programme und numerische Bibliotheken möglich sein. Das Projekt hat in dieser Richtung einen wichtigen Meilenstein gesetzt, insofern es im Bereich der Teilchensimulationen, die einen sehr großen Anteil an Rechenkapazitäten verbrauchen, die langreichweitigen Wechselwirkungen als eine wesentliche Komponente für realistische, atomistische Simulationsszenarien identifiziert hat und eine hochskalierbare Bibliothek mit verschiedenen Methoden der Benutzer-Community zur Verfügung stellt. Die nachhaltige (Weiter-)Entwicklung dieser Bibliothek wird dabei durch eine Open-Source Distribution gewährleistet, die unter der LGPL-Lizenz (Teile davon unter einer GPL-Lizenz) vertrieben wird. Dies sichert z.B. auch Industrie-Partnern zu, die Bibliothek in kommerzielle Software-Pakete zu integrieren.

Der Nutzen der Bibliothek für Anwender der HPC Architekturen in **Jülich** liegt in ihrem breiten Einsatzspektrum in Bereichen der klassischen Molekulardynamik und atomistischen Simulationen, in denen elektrostatische Wechselwirkungen zwischen Punktzentren ausgewertet werden müssen. Durch Verwendung der Bibliothek können hier lange Implementierungszeiten deutlich verringert bzw. vermieden werden.

Durch die ausgewählten Implementierungen sind sowohl Basis-Methoden für die parallele Sortierung und nichtäquidistante schnelle Fourier-Transformationen als auch Methoden zur Behandlung langreichweitiger Wechselwirkungen implementiert worden. Die Verbreitung wird zunächst über einen Jülicher Server erfolgen. Zielgruppen sind dabei vor allem die Nutzer von hochskalierenden Architekturen, wie sie z.B. in den Tier-0 Zentren installiert sind, aber auch Gruppen, die auf kleineren Architekturen und lokalen Clustern arbeiten. Erste Installationen sind bereits in verschiedenen Gruppen in Europa vorgenommen worden bzw. sind in Planung. So wurde SCAFACOS in das OCTOPUS DFT Paket integriert. Anfragen und erste Anpassungen gibt es für das in Daresbury entwickelte Molekulardynamik-Programm DL_POLY und das weit verbreitete Programm GROMACS.

Das bestehende Konsortium aus dem BMBF-Projekt wird auch in Zukunft in engem Kontakt stehen und weiter an der Entwicklung und dem Ausbau der Bibliothek arbeiten. In Zukunft sollen dabei auch weitere Methoden aufgenommen werden, die andere Simulationsszenarien abdecken und es sollen die bestehenden Methoden auf allgemeinere Geometrien erweitert werden.

Die Verbreitung und Anwendung der Bibliothek wird durch gezielte Kurse sowie Dokumentationen gewährleistet. Während der Projektphase wurden bereits zwei interna-

tionale Schulen im Rahmen einer Heraeus-Sommerschule sowie eines CECAM Tutorials durchgeführt. Im September 2013 wird ein weiteres CECAM Tutorial durchgeführt. Weiterhin wird die Bibliothek im Rahmen der Ausbildungsveranstaltungen des JSC (Gaststudentenprogramm, Winterschulen, Workshops, Einführungsveranstaltungen) eingesetzt, wobei in die Arbeitsweise der verwendeten Algorithmen und die Nutzung der Bibliothek eingeführt sowie einzelne Module weiterentwickelt und auf aktuelle moderne Rechnerarchitekturen angepasst werden.

Die generischen Verfahren der **TU Chemnitz** für das parallele Sortieren und die schnellen Fourier-Transformationen finden Einsatz im gesamten wissenschaftlich technischen Umfeld. Durch die Bereitstellung als eigenständige und quelloffene Softwarekomponenten sind alle entwickelten Verfahren auch über Projektende hinaus einem weit gefassten Nutzerkreis zugänglich. Das modulare Konzept der entwickelten Softwarekomponenten ermöglicht die flexible Kombination mit anderen Softwarekomponenten und damit einen einfachen Transfer der geleisteten Entwicklungsarbeit auf andere Produkte. Die entwickelten Module bieten somit eine indirekte wirtschaftliche Verwertbarkeit durch die effizientere Parallelisierung sowohl bereits vorhandener als auch neuer Software. Die technischen Erfolgsaussichten dieser Lösungen werden sowohl durch ihre generische Realisierung als auch durch die Nutzung von bzw. Orientierung an etablierten Standards (wie z.B. MPI) unterstützt.

An der **Universität Stuttgart** wurde einerseits einer der im Umfeld von Molekulardynamiksimulationen gebräuchlichsten Standardalgorithmen zur Berechnung elektrostatischer Wechselwirkungen in die Bibliothek eingefügt (P3M). Da dieser Algorithmus in vielen Anwendergruppen bekannt sein dürfte, ist er für die Bibliothek von zentraler Bedeutung zur Schaffung von Vertrauen, und zur Vergleichbarkeit.

Obwohl die Methoden MMM1D, MMM2D und die reine Ewald-Summation über eine hohe Komplexität verfügen und damit für große Systeme ungeeignet sind, so können sie bei kleinen Systemen zur Berechnung von hochgenauen Referenzkräften dienen.

Außerdem wurde das lokale Gitterverfahren MEMD entwickelt und implementiert. Die große Stärke dieses Verfahrens liegt in seiner Lokalität. Dadurch ist nicht nur eine sehr effiziente Parallelisierung möglich, sondern es eröffnet sich auch die Möglichkeit, lokal variierende dielektrische Eigenschaften in ein System zu integrieren. Diese Möglichkeit ist auf dem Gebiet der Simulation mit elektrostatischen Wechselwirkungen kaum verbreitet und spielt doch in vielen Anwendungen eine entscheidende Rolle. Die Integration dieses lokalen Gitteralgorithmus in die Bibliothek ist ein Alleinstellungsmerkmal.

Desweiteren wurde die Löserbibliothek an das Molekulardynamikprogramm ESPRESSO angebunden, welches weltweit zur Simulation von vergrößerten Modellen weicher Materie verwendet wird. Da ESPRESSO einen Schwerpunkt auf elektrostatischen Algorithmen hat, konnten verschiedene Testsysteme mit bekannten Ergebnissen gerechnet werden. Dies diente zur Validierung der verschiedenen Löser. Die Ergebnisse sind konsistent mit den Benchmarktests und den Ergebnissen der Validierung mit IMD.

Durch die Anbindung der Bibliothek an ESPRESSO werden zahlreiche neue Methoden zur Berechnung elektrostatischer Wechselwirkungen in der Software verfügbar, und die Möglichkeiten der Software steigen.

Die verschiedenen Löser sind Forschungsgegenstand beim Projektpartner Stuttgart. Neuentwicklungen und Verbesserungen wurden bislang in die Simulationssoftware ESPRESSO eingebaut, getestet und validiert. Die Integration der Algorithmen in die Bibliothek hat daher den Vorteil, dass zukünftige Entwicklungen direkt in der SCAFACOS-Bibliothek stattfinden werden. Dadurch kommen diese Verbesserungen unmittelbar anderen Nutzern der Bibliothek zu Gute.

An der **Universität Bonn** konnte die SCAFACOS-Bibliothek erfolgreich in die Bonner Moleküldynamiksoftware integriert werden. Weiterhin steht die parallele Mehrgitter-Bibliothek *VMG* als eigenständiges Open-Source-Modul zur Verfügung und wurde bereits erfolgreich als langreichweitiger Poisson-Löser in einem Code zur approximativen ab-initio Berechnung der Elektronenstruktur eingesetzt.

2.5 Während der Durchführung des Vorhabens bekannt gewordene Fortschritte bei anderen Stellen

Arbeiten an anderen Stellen, die die Entwicklung einer skalierbaren Bibliothek für langreichweitige Wechselwirkungen in atomistischen Systemen zum Ziel haben, sind den Konsortial-Partnern nicht bekannt. Das hier vorgestellte Konzept, die Zielrichtung und die Implementierung der Bibliothek können daher als Alleinstellungsmerkmal betrachtet werden. Lediglich für einzelne Komponenten, die in die Bibliothek integriert worden sind, hat es während der Projektlaufzeit an anderen Stellen Weiterentwicklungen gegeben.

Neben der an der TU Chemnitz entwickelten parallelen Softwarebibliothek für die schnelle Fourier-Transformation sind weitere Bibliotheken bereits seit Beginn des Projekts verfügbar [80, 99, 100] oder wurden während der Projektlaufzeit publiziert [92]. Obwohl die grundlegende Datenaufteilung der verwendeten parallelen Algorithmen gleich ist, besitzt die parallele FFT-Bibliothek der TU Chemnitz entscheidende Alleinstellungsmerkmale. So eignet sich keine der anderen Bibliotheken für effiziente Parallelisierung der nichtäquidistanten schnellen Fourier-Transformation und der schnellen Summation. Auch die Verwendung innerhalb der Simulationen für [69] war nur durch die bisher einzigartige Anpassung der parallelen FFT-Algorithmen an höher-dimensionale Datenaufteilungen möglich.

Durch die Verfügbarkeit größerer Computersysteme mit mehr Prozessoren rückt die Parallelisierung bestehender Algorithmen mehr und mehr in den Mittelpunkt. Aus diesem Grund ist es auch nicht verwunderlich, dass eine ganze Reihe anderer FMM Implementierungen [72, 91, 111] zur Lösung des Coulomb Problems verfügbar sind. Allerdings beschränken sich diese Implementierungen meist nur auf einen kleinen Genauigkeitsbereich, eine bestimmte Rechner-Architektur oder eine bestimmte Zielgruppe. Die Möglichkeit die Fehlerschranke und die Laufzeit a-priori mit einem einzigen Parameter einzustellen, ermöglicht es, die FMM einem größeren Anwenderfeld zur Verfügung zu stellen. Durch das gemeinsame, einheitliche Interface ist es somit auch neuen Nutzergruppen leicht möglich die FMM ohne aufwendige Parametersuche zu verwenden.

Lokale Gittermethoden sind weiterhin eine eher exotische Art, Elektrostatik zu berechnen. Dementsprechend gab es hier wenig Veröffentlichungen von dritter Seite, die für das Projekt von Bedeutung waren. Zu erwähnen sind die Arbeiten von Pasquali und Maggs [98] sowie Rottler und Krayenhoff [102]. Beide beschäftigen sich auch mit Algorithmen, die auf der Idee von Maggs [94] basieren, bewegen sich aber in andere Forschungsrichtungen.

2.6 Erfolgte oder geplante Veröffentlichungen der Ergebnisse

In diesem Abschnitt sind alle im Rahmen des SCAFACOS Projekts entstandenen Veröffentlichungen und Vorträge, einzeln nach Projektpartner aufgelistet.

2.6.1 Veröffentlichungen Bonn

Artikel

- [1] A. Bittner, F. Heber und J. Hamaekers: *Biomolecules as soft matter surfaces*. Surf. Sci., 603:10–12, 2009.

Vorträge

- M. Griebel, *Numerical Simulation in Molecular Dynamics Numerics, Algorithms, Parallelization, Applications*, Molecular Modelling Course, Ludwigshafen, 2009
- M. Griebel, *On the ANOVA decomposition of high-dimensional functions with applications to chemistry*, 27th GAMM-Seminar Leipzig on Approximation of Multiparametric functions, Leipzig, 2011
- M. Griebel, *Generalized sparse grid approximations*, Kolloquium anlässlich des 60. Geburtstag von Peter Oswald, Bremen, 2011
- M. Griebel, *On generalized sparse grids*, 28th GAMM-Seminar Leipzig on Analysis and Numerical Methods in Higher Dimensions, Leipzig, 2012
- M. Griebel, *Greedy and Randomized Versions of the Multiplicative Schwarz Method*, ESF-JSPS Frontier Science Conference for Young Researchers: Mathematics for Innovation: Large and Complex Systems, Tokyo, Japan, 2012
- M. Griebel, F. Heber, *From Electronic Structure Calculations to Molecular Mechanic Potentials via Automated Bond Order Dissection*, 6th Multiscale Materials Modeling, Singapur, 2012

2.6.2 Veröffentlichungen Chemnitz

Artikel

- [2] A. Arnold und M. Pippig: *Equivalence of P3M and NFFT-Based Fast Summation Methods*. 2012. (in preparation).

- [3] H. Dachsel, M. Hofmann, J. Lang und G. Runger: *Automatic Tuning of the Fast Multipole Method Based on Integrated Performance Prediction*. In: *14th IEEE International Conference on High Performance Computing and Communications (HPCC 2012)*, 2012.
- [4] P. Garca-Risueno, J. Alberdi-Rodrguez, M. J. T. Oliveira, X. Andrade, M. Pippig, J. Muguerza, A. Arruabarrena und A. R. Secades: *A Survey of the Performance of Classical Potential Solvers for Charge Distributions*. 2012. (in preparation).
- [5] P. Gibbon, M. Hofmann, G. Runger und R. Speck: *Parallel Sorting Algorithms for Optimizing Particle Simulations*. In: *2010 IEEE International Conference on Cluster Computing, Workshops and Posters (CLUSTER WORKSHOPS)*, S. 1–8. IEEE, 2010.
- [6] M. Hofmann: *Effiziente parallele Sortier- und Datenumverteilungsverfahren fur Partikelsimulationen auf Parallelrechnern mit verteiltem Speicher*. Dissertation, TU Chemnitz, Fakultat fur Informatik, 2012.
- [7] M. Hofmann und G. Runger: *Fine-grained Data Distribution Operations for Particle Codes*. In: *Recent Advances in Parallel Virtual Machine and Message Passing Interface, 16th European PVM/MPI Users Group Meeting*, Bd. 5759 d. Reihe LNCS, S. 54–63. Springer, 2009.
- [8] M. Hofmann und G. Runger: *An In-place Algorithm for Irregular All-to-All Communication with Limited Memory*. In: *Recent Advances in the Message Passing Interface: 17th European MPI User’s Group Meeting*, Bd. 6305 d. Reihe LNCS, S. 113–121. Springer, 2010.
- [9] M. Hofmann und G. Runger: *A Partitioning Algorithm for Parallel Sorting on Distributed Memory Systems*. In: *13th IEEE International Conference on High Performance Computing and Computations (HPCC 2011)*, S. 402–411. IEEE, 2011.
- [10] F. Nestler: *Approximationsverfahren zur schnellen Energieberechnung in Partikelsystemen*. Diplomarbeit, Fakultat fur Mathematik, Technische Universitat Chemnitz, 2012.
- [11] M. Pippig: *Parallele nichtaquidistante schnelle Fourier-Transformation und schnelle Summation*. Diplomarbeit, Fakultat fur Mathematik, Technische Universitat Chemnitz, 2009.
- [12] M. Pippig: *Scaling Parallel Fast Fourier Transform on BlueGene/P*. In: B. Mohr und W. Frings (Hrsg.): *Julich BlueGene/P Scaling Workshop 2010*, S. 27 – 30, Julich, Mai 2010. Forschungszentrum Julich. Technical Report.

- [13] M. Pippig: *An Efficient and Flexible Parallel FFT Implementation Based on FFTW*. In: *Proc. of the CiHPC: Competence in High Performance Computing, HPC Status Konferenz der Gauß-Allianz e.V., Schwetzingen, Germany*. Springer, 2011.
- [14] M. Pippig: *PFFT, Parallel FFT Subroutine Library*, 2011. <http://www.tu-chemnitz.de/~mpip/software>.
- [15] M. Pippig: *PNFFT, Parallel Nonequispaced FFT Subroutine Library*, 2011. <http://www.tu-chemnitz.de/~mpip/software>.
- [16] M. Pippig: *PFFT - An Extension of FFTW to Massively Parallel Architectures*. Preprint TU Chemnitz, Preprint 6, 2012.
- [17] M. Pippig und D. Potts: *Particle Simulation Based on Nonequispaced Fast Fourier Transforms*. S. 1 – 29, Jülich, 2010. Forschungszentrum Jülich. Technical Report.
- [18] M. Pippig und D. Potts: *Parallel Three-Dimensional Nonequispaced Fast Fourier Transforms and Their Application to Particle Simulation*. Preprint TU Chemnitz, Preprint 8, 2012.

Vorträge

- M. Pippig, *Parallele nichtäquidistante schnelle Fourier-Transformation und schnelle Summation*, 15. Südostdeutsches Kolloquium zur Numerischen Mathematik, Chemnitz, 8.5.2009.
- M. Hofmann, *Fine-grained Data Distribution Operations for Particle Codes*, EuroPVM/MPI 2009, Espoo, Finnland, 7.-10.9.2009.
- M. Pippig, *An Efficient and Flexible Parallel FFT Implementation Based on FFTW*, HPC-Statuskonferenz der Gauß-Allianz e.V., Schwetzingen, 22.-24.6.2012.
- M. Pippig, *Introduction to Practical FFT and NFFT*, WE Heraeus Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 06.-10.9.2010.
- D. Potts, *NFFT and Fast Summation*, WE Heraeus Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 06.-10.9.2010.
- M. Hofmann, *An In-place Algorithm for Irregular All-to-All Communication with Limited Memory*, EuroMPI 2010, Stuttgart, 12.-15.9.2010.
- M. Hofmann, *Parallel Sorting Algorithms for Optimizing Particle Simulations*, IEEE International Conference on Cluster Computing, Workshop on High-Performance Computing on Complex Environments (HPCCE), Heraklion, Griechenland, 20.-24.09.2010.

- M. Hofmann, M. Pippig, *Zuarbeiten zum Projektvortrag von G. Sutmann*, BMBF-Statuskonferenz, Berlin, 23.11.2010.
- M. Pippig, *Parallel Fast Ewald Summation Based on Nonequispaced Fourier Transforms*, SIAM Conference on Computational Science and Engineering, Reno, Nevada, 28.2.-4.3.2011.
- G. Runger, *A Partitioning Algorithm for Parallel Sorting on Distributed Memory Systems*, IEEE International Conference on High Performance Computing and Communications (HPCC-2011), Banff, Kanada, 2.-4.9.2011.
- M. Hofmann, *Parallel Sorting in Particle Codes*, CECAM Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Julich, 12.-16.9.2011.
- M. Pippig, *Introduction to Practical FFT and NFFT*, CECAM Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Julich, 12.-16.9.2011.
- D. Potts, *NFFT and Fast Summation*, CECAM Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Julich, 12.-16.9.2011.
- M. Pippig, *Parallel Fast Fourier Transforms and Their Application to Particle Simulation*, DMV Jahrestagung 2011, Koln, 19.-22.09.2011.
- M. Hofmann, M. Pippig, *Zuarbeiten zum Projektvortrag von G. Sutmann*, HPC-Statuskonferenz, Darmstadt, 5.-6.12.2011.
- M. Pippig, *On the Relation of Ewald Summation and NFFT-Based Fast Summation*, 22. Rhein-Ruhr-Workshop, Bestwig, 3.-4.2.2012.
- M. Hofmann, *Effiziente parallele Sortier- und Datenumverteilungsverfahren fur Partikelsimulationen auf Parallelrechnern mit verteiltem Speicher*, Dissertationsverteidigung, Chemnitz, 9.3.2012.
- M. Pippig, *Parallel Fast Computation of Coulomb Interactions Based on Nonequispaced Fourier Methods*, 83. Jahrestagung der Gesellschaft fur Angewandte Mathematik und Mechanik (GAMM 2012), Darmstadt, 26.-30.3.2012.
- M. Pippig, *Parallel Fast Coulomb Solvers*, 2012 SIAM Conference on Applied Linear Algebra, Valencia, Spanien, 18.-22.6.2012.
- J. Lang, *Automatic Tuning of the Fast Multipole Method Based on Integrated Performance Prediction*, IEEE International Conference on High Performance Computing and Communications (HPCC-2012), Liverpool, UK, 25.-27.6.2012.

- M. Pippig, *Massively parallel computation of nonequispaced fast Fourier transforms*, 7th International Workshop on Parallel Matrix Algorithms and Applications (PMAA 2012), London, UK, 28.-30.6.2012.

2.6.3 Veröffentlichungen Jülich

Artikel

- [19] B. Berberich: *Entwicklung und Anwendung eines Tree-Codes in Simulationsszenarios der Plasma-Wand-Wechselwirkung*. Dissertation, Heinrich-Heine-Universität Düsseldorf, September 2011.
- [20] B. Berberich, P. Gibbon und D. Reiter: *Tree-code based meshfree simulation of a gas-puff*. In: *Verhandlungen der Deutschen Physikalischen Gesellschaft*, Bd. 173 d. Reihe 06, S. 45, 2010.
- [21] A. Breslau: *Highly-parallel Smoothed Particle Hydrodynamics modelling of proto-planetary discs*. Diplomarbeit, Universität Köln, August 2011.
- [22] P. Gibbon, D. Reiter, F. Jenko und G. Ackland: *G8-Exascale Project Nu-FuSE to tackle Reactor Design Challenge*. inSiDE, 10(1):90–93, 2012.
- [23] P. Gibbon, R. Speck, L. Arnold, M. Winkel und H. Hübner: *Parallel Tree Codes*. In: G. Sutmann, P. Gibbon und T. Lippert (Hrsg.): *Fast Methods for Long-Range Interactions in Complex Systems, Summer School, 6-10 September 2010*, Bd. 6 d. Reihe IAS Series, S. 65 – 84. Schriften des Forschungszentrums Jülich, 2011, ISBN 978-3-89336-714-6.
- [24] P. Gibbon, R. Speck, B. Berberich, A. Karmakar, L. Arnold und M. Masek: *Plasma Simulation with Parallel Kinetic Particle Codes*. In: G. Münster, D. Wolf und M. Kremer (Hrsg.): *NIC Symposium 2010*, Bd. 3 d. Reihe IAS Series, S. 383–390, Jülich, Feb. 2010. Forschungszentrum Jülich, John von Neumann Institute for Computing, ISBN 978-3-89336-606-4.
- [25] P. Gibbon, R. Speck, A. Karmakar, L. Arnold, W. Frings, B. Berberich, D. Reiter und M. Masek: *Progress in Mesh-Free Plasma Simulation With Parallel Tree Codes*. IEEE Transactions on Plasma Science, 38(9):2367–2376, sep. 2010, ISSN 0093-3813.
- [26] J. Grotendorst, G. Sutmann, G. Gompper und D. Marx (Hrsg.): *Lecture Notes on Hierarchical Methods for Dynamics in Complex Molecular Systems*, Bd. 10, Jülich, 2012. Institute for Advanced Simulation.

- [27] R. Halver: *Adaptives Lastbalance-Verfahren für Gebietszerlegung in der Molekulardynamik*. Masterarbeit, Fachhochschule, Jülich, 2010.
- [28] R. Halver, G. Sutmann und M. Reißel: *Diffusion Based Adaptive Load-Balancing for Domain Decomposition in Particle Simulations*. Poster, June 2010. präsentiert auf CCP 2010, Trondheim Norwegen (24.06.2010).
- [29] H. Hübner: *A priori minimisation of algorithmic bottlenecks in the parallel tree code PEPC*. Masterarbeit, Fachhochschule Aachen, Jülich, July 2011.
- [30] T. Lippert, P. Gibbon und G. Sutmann: *Preface*. In: G. Sutmann, T. Lippert und P. Gibbon (Hrsg.): *Lecture Notes on Fast Methods for Long-Range Interactions in Complex Systems*, Bd. 6, S. 1–37, Jülich, 2011. Institute for Advanced Simulation.
- [31] M. Masek und P. Gibbon: *Mesh-Free Magnetoinductive Plasma Model*. IEEE Transactions on Plasma Science, 38(9):2377–2382, sep. 2010, ISSN 0093-3813.
- [32] H. Peschke: *Hierarchical Tree Construction in PEPC*. In: *Proceedings 2011, JSC Guest Student Programme on Scientific Computing*, S. 21–34. Forschungszentrum Jülich GmbH, 2011.
- [33] R. Speck: *Generalized Algebraic Kernels and Multipole Expansions for massively parallel Vortex Particle Methods*. Dissertation, Bergische Universität Wuppertal, July 2011.
- [34] R. Speck, L. Arnold und P. Gibbon: *Towards a petascale tree code: Scaling and efficiency of the PEPC library*. Journal of Computational Science, 2:137–142, 2011, ISSN 1877-7503.
- [35] R. Speck, P. Gibbon und M. Hofmann: *Efficiency and scalability of the parallel Barnes-Hut tree code PEPC*. In: B. Chapman, F. Desprez, G. R. Joubert, A. Lichnewsky, F. J. Peters und T. Priol (Hrsg.): *Parallel Computing: From Multicores and GPU's to Petascale*, Bd. 19 d. Reihe *Advances in Parallel Computing*. IOS Press, Apr. 2010, ISBN 978-1-60750-529-7.
- [36] R. Speck, P. Gibbon und R. Krause: *Parallel remeshing in tree codes for vortex particle methods*. In: *ParCo'11 Proceedings*, 2011.
- [37] R. Speck, D. Ruprecht, R. Krause, M. Emmett, M. Minion, M. Winkel und P. Gibbon: *Integrating an N-body problem with SDC and PFASST*. In: *21st International Conference on Domain Decomposition Methods (submitted)*, Rennes, France, 2012.

- [38] R. Speck, D. Ruprecht, R. Krause, M. Emmett, M. Minion, M. Winkel und P. Gibbon: *A massively space-time parallel N-body solver*. Proceedings of "The International Conference for High-Performance Computing, Networking, Storage and Analysis 2012 (SC '12)", 2012.
- [39] G. Sutmann: *Molecular Dynamics - Extending the Scale from Microscopic to Mesoscopic*. In: J. Grotendorst, N. Attig, S. Blügel und D. Marx (Hrsg.): *Lecture Notes on Multiscale Simulation Methods in Molecular Sciences*, Bd. 42, S. 1–50, Jülich, 2009. John von Neumann Institute for Computing.
- [40] G. Sutmann: *ScaFaCoS - when long range goes parallel*. Inside, 1:58–59, 2009.
- [41] G. Sutmann: *Summer School on Fast Methods for Long-Range Interactions in Complex Systems*. Inside, 8:64, 2010.
- [42] G. Sutmann: *Classical Particle Simulations*. In: G. Sutmann, T. Lippert und P. Gibbon (Hrsg.): *Lecture Notes on Fast Methods for Long-Range Interactions in Complex Systems*, Bd. 6, S. 1–37, Jülich, 2011. Institute for Advanced Simulation.
- [43] G. Sutmann: *WE-Heraeus Sommer-Schule on Fast Methods for Long-Range Interactions in Complex Systems*. Physik Journal, 10:62, 2011.
- [44] G. Sutmann, T. Lippert und P. Gibbon (Hrsg.): *Lecture Notes on Fast Methods for Long-Range Interactions in Complex Systems*, Bd. 6, Jülich, 2011. Institute for Advanced Simulation.
- [45] G. Sutmann, L. Westphal und M. Bolten: *Particle Based Simulations of Complex Systems with MP2C: Hydrodynamics and Electrostatics*. In: *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics 2010 (ICNAAM)*. AIP, 2010.
- [46] M. Winkel, R. Speck, H. Hübner, L. Arnold, R. Krause und P. Gibbon: *A massively parallel, multi-disciplinary Barnes-Hut tree code for extreme-scale N-body simulations*. Computer Physics Communications, 183(4), Apr. 2012.

Vorträge

- G. Sutmann, *Fast Methods for Long-Range Interactions in Many-Particle Systems*, Max-Planck- Institute for Mathematics in the Sciences, Leipzig, 28.5.2009.
- G. Sutmann, *Efficient Methods for Micro- and Mesoscopic Time- and Length Scales in Classical Particle Simulations*, Workshop on Solvation of Bioactive Compounds: Bridging Experiment, Computation and Experiment, Max-Planck Institute for Mathematics in the Sciences, Leipzig, 7.1.-9.1.2010.

- G. Sutmann, *Particle Simulations and Long Range Interactions*, Simulation Laboratory Workshop, Jülich, 9.-11.6.2010.
- G. Sutmann, *Particle Based Hydrodynamics and Electrostatics coupled to MD on Massively Parallel Computers*, CiHPC - Competence in High Performance Computing, HPC Status Conference of Gauß Alliance, Schwetzingen, 22.-24.6.2010.
- G. Sutmann, *Simulating Complex Systems with MP2C: Hydrodynamics and Electrostatics*, International Conference on Numerical Analysis and Applied Mathematics 2010 (ICNAAM), Rhodes, Greece, 19.-25.9.2010.
- G. Sutmann, *ScaFaCoS - A Scalable Library for Fast Coulomb Solvers*, BMBF Status Report Meeting, Berlin, 23.-24.11.2010.
- G. Sutmann, *Fast Methods for Electrostatic Problems in Particle Simulations*, STFC Workshop on Algorithms and Architectures for Molecular Simulations, Daresbury, 2.12.2010.
- G. Sutmann, *Parallel Particle Simulation Methods for Micro- and Mesoscales*, JICS / GRS workshop on HPC, Aachen, 9.3.2011.
- G. Sutmann, *Fast Methods for the Coulomb Problem in Complex Systems*, AICES seminar, Aachen, 7.7.2011.
- G. Sutmann, *Classical Particle Simulations*, Jülich-CECAM Summerschool on Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 12.-16.9.2011.
- G. Sutmann, *Skalierbare schnelle Löser für langreichweitige Wechselwirkungen (ScaFaCoS) - Scalable fast solvers for long-range interactions (ScaFaCoS)*, Gauß-Alliance Status Meeting, Darmstadt, 5.-6.12.2011.
- R. Halver, *ScaFaCoS Library*, CECAM Sommerschule 2011, Jülich, 13.9.2011
- P. Gibbon, *Progress in mesh free plasma simulation with parallel tree codes*, Talk at 21st International Conference on Numerical Simulation of Plasmas, Lisbon 6.-9.10.2009
- L. Arnold, R. Speck, B. Berberich, M. Winkel, P. Gibbon, *The Parallel Tree Code Framework PEPC: Petascaling and Applications*, Poster Contribution to the NIC-Symposium 25.02.2010
- P. Gibbon, *Plasma Simulation with Parallel Kinetic Particle Codes*, NIC Symposium 2010, Jülich: 25.02.2010

- P. Gibbon, *Progress in Mesh-Free Plasma Simulation with Parallel Tree Codes*, Invited talk at the Institute of Physics, Czech Technical University, Prague, 9.4.2010
- P. Gibbon, *Parallel Particle Methods in Plasma Physics*, 1st Simulation Laboratory Porting Workshop, JSC, 9.-11.6.2010, June 2010
- P. Gibbon, *The Plasma Simulation Laboratory at JSC: Algorithms, Modelling and some Physics*, Invited talk, Scientific Meeting of Helmholtz Institute Jena, 8.7.2010
- P. Gibbon, *Parallel Tree Methods*, in: *Fast Methods for Long-Range Interactions in Complex Systems*, WE-Heraeus Summer School 2010, Jülich: 06.09.2010 - 10.09.2010
- B. Berberich, P. Gibbon, D. Reiter, *Tree-code based meshfree simulation of a gas-puff*, DPG-Frühjahrstagung 2010, Hannover: 08.03.2010 - 12.03.2010
- M. Winkel, R. Speck, H. Hübner, L. Arnold, P. Gibbon, *PEPC - The Pretty Efficient Parallel Coulomb Solver: Applications to Strongly Coupled Plasma*, Poster Contribution: The Structure and Dynamics of Hadrons, International Workshop XXXIX on Gross Properties of Nuclei and Nuclear Excitations Hirschegg, Österreich: 16.01.2011 - 22.01.2011
- M. Winkel, P. Gibbon, L. Arnold, R. Speck, H. Hübner *PEPC - The Pretty Efficient Parallel Coulomb Solver: Applications to Strongly Coupled Plasma*, Poster Contribution to the Helmholtz Alliance EMMI Midterm Evaluation, GSI-Darmstadt, 18.03.2011
- M. Winkel, L. Arnold, P. Gibbon, *Practical PEPC*, Hands-On Session at the CECAM Summer School on Long Range Interactions 12.09.2011 - 16.09.2011
- R. Speck, R. Krause, P. Gibbon, *A Vortex Tree Code for Extreme Scale Fluid Simulations*, 15th SIAM Conference on Parallel Processing for Scientific Computing (SIAMPP12), February 15-17, Savannah GA, USA, 2011
- A. Breslau, M. Winkel, L. Arnold, P. Gibbon, S. Pfalzner, *Combining fluid dynamics with the tree-code PEPC*, Poster contribution to the NIC Symposium 07.-08.02.2012
- P. Gibbon, *Mesh-Free Plasma Simulation with Parallel Tree Codes*, Invited talk at the 2nd Swiss HP2C Workshop, Institute of Computational Science, Lugano, March 2012
- P. Gibbon, *Mesh-Free Plasma Simulation with Parallel Tree Codes*, Invited talk at the Institute of Physics Plasma Physics Annual Meeting, St Hugh's College, Oxford, April 2012

- L. Arnold, M. Winkel, P. Gibbon, *Highly scalable asynchronous parallelization of the Tree Code PEPC*, ScicomP 2012 Conference, May 14th - 18th, Toronto, Canada, May 2012
- P. Gibbon, M. Winkel, *Particle Simulations with Parallel Tree Codes*, JARA Workshop Öctree-based methods in computational physics", AIA, RWTH Aachen, 2012-09-18
- M. Winkel, P. Gibbon, *Electronic correlations in nanoclusters - Molecular dynamics simulations*, 14th International Conference on the Physics of Non-Ideal Plasmas (PNP14), Rostock, September 9-14, 2012
- P. Gibbon, *Mesh-Free Simulation Long-Range N-Body Systems with Parallel Tree Codes*, Invited talk at the IMPRS Seminar, Max-Planck Institute for Complex Systems, Dresden, 2012

2.6.4 Veröffentlichungen Stuttgart

Artikel

- [47] A. Arnold: *Fourier Transformed-Based Methods for Long-Range Interactions: Ewald, P³M and More*. In: G. Sutmann, P. Gibbon und T. Lippert (Hrsg.): *Fast Methods for Long-Range Interactions in Complex Systems*, Bd. 6 d. Reihe IAS Series, S. 59–109. FZ Jülich, 2011.
- [48] A. Arnold, O. Lenz und C. Holm: *Simulating Charged Systems with ESPRESSO*. In: G. Sutmann, P. Gibbon und T. Lippert (Hrsg.): *Fast Methods for Long-Range Interactions in Complex Systems*, Bd. 6 d. Reihe IAS Series, S. 159–166. FZ Jülich, 2011.
- [49] V. Ballenegger, J. J. Cerdà und C. Holm: *Removal of Spurious Self-Interactions in Particle-Mesh Methods*. *Comp. Phys. Comm.*, 182(9, SI):1919–1923, sep 2011.
- [50] J. J. Cerda, V. Ballenegger und C. Holm: *Particle-Particle Particle-Mesh Method for Dipolar Interactions: On Error Estimates and Efficiency of Schemes with Analytical Differentiation and Mesh Interlacing*. *J. Chem. Phys.*, 135(18), nov 2011.
- [51] A. Michel: *Eine Analyse elektrostatischer Algorithmen in der Scafacos-Bibliothek*. Diplomarbeit, Universität Stuttgart, 2012. Eingereicht.
- [52] A. Neelov und C. Holm: *Interlaced P3M Algorithm with Analytical and ik-Differentiation*. *J. Chem. Phys.*, 132(23), jun 2010.

- [53] C. Tyagi, M. Süzen, M. Sega, M. Barbosa, S. Kantorovich und C. Holm: *An iterative, fast, linear-scaling method for computing induced charges on arbitrary dielectric boundaries*. J. Chem. Phys., 132, 2010.
- [54] F. Weik: *Implementation of an improved P3M algorithm*. Diplomarbeit, Universität Stuttgart, 2011.

Vorträge

- A. Arnold, *Fourier-Transform Based Methods*, WE Heraeus Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 06.-10.9.2010.
- A. Arnold, *Optimal Methods: P3M*, WE Heraeus Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 06.-10.9.2010.
- O. Lenz, *Introduction to ESPRESSO*, WE Heraeus Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 06.-10.9.2010.
- A. Arnold, *Long-range interactions: PPPM, MMMxD, ELC and MEMD*, SimTech-CECAM-Tutorial Workshop „Simulating Soft Matter with ESPRESSO“, Stuttgart, 11.-15.10.2010.
- O. Lenz, *Hands-on session: Simulating charged systems with ESPRESSO*, SimTech-CECAM-Tutorial Workshop „Simulating Soft Matter with ESPRESSO“, Stuttgart, 11.-15.10.2010.
- O. Lenz, *Long-range Interactions in Particle Simulations: ScaFaCoS*, European-U.S. Summer School, Lake Tahoe, 2011.
- A. Arnold, *Fourier-Transform Based Methods and Optimal Methods P3M*, CECAM Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 12.-16-9.2011.
- O. Lenz, F. Fahrenberger, *Practical Session: Fourier-Transform Based Methods, Optimal Method P3M and MEMD*, CECAM Sommerschule: Fast Methods for Long-Range Interactions in Complex Systems, Jülich, 12.-16-9.2011.
- A. Arnold, *Long-range interactions: PPPM, MMMxD, ELC and MEMD*, SimTech-CECAM-Tutorial Workshop „Coarse-grained Simulation of Biological Soft Matter Systems using ESPRESSO“, Stuttgart, 10.-14.10.2011.
- O. Lenz, *Hands-on session: Simulating charged systems with ESPRESSO*, SimTech-CECAM-Tutorial Workshop „Coarse-grained Simulation of Biological Soft Matter Systems using ESPRESSO“, Stuttgart, 10.-14.10.2011.

- A. Arnold, *Electrostatic interactions in ESPRESSO*, SimTech-CECAM Summer School „Simulating Soft Matter with ESPRESSO, ESPRESSO++ and VOTCA“, Stuttgart, 8.-12.10.2012.

2.6.5 Veröffentlichungen Wuppertal

Artikel

- [55] M. Bolten: *Highly Scalable Multigrid Algorithm for Particle Simulation*. Proc. in Appl. Math. and Mech., 10:643–644, 2010.
- [56] M. Bolten: *Multigrid methods for long-range interactions*. Bd. 6 d. Reihe *Fast Methods for Long-Range Interactions in Complex Systems*, S. 115–130. Forschungszentrum Jülich, 2011.
- [57] S. Friedhoff: *Load Balancing and Error Estimation for a Parallel, Mesh-Based Particle Simulation Method*. Masterarbeit, Bergische Universität Wuppertal, 2010.
- [58] S. von Helden-Sarnowski: *Parallelization of a Multigrid Method Using Hierarchical Grid Coarsening*. Masterarbeit, Bergische Universität Wuppertal, 2011.
- [59] G. Sutmann, L. Westphal und M. Bolten: *Particle Based Simulations of Complex Systems with MP2C: Hydrodynamics and Electrostatics*. AIP Cong. Proc., 1281:1768–1772, 2010.

Vorträge

- M. Bolten, G. Sutmann, *Parallel Molecular Dynamics Simulation Method Using Multigrid*, SIAM Conference on Computational Science and Engineering 2009, Miami, FL, USA, 2.3.2009.
- M. Bolten, *Highly scalable multigrid algorithm for particle simulation*, International Workshop on Computational Engineering, Hersching, 13.10.2009.
- M. Bolten, *Multigrid methods in particle simulation*, PennState University, State College, PA, USA, 5.3.2010.
- M. Bolten, *Highly scalable multigrid algorithm for particle simulation*, 81. Jahrestagung der Gesellschaft für Angewandte Mathematik und Mechanik, Karlsruhe, 25.3.2010.
- M. Bolten, *Multigrid Methods for Long-Range Interactions*, WE-Heraeus Summer School 2010, Forschungszentrum Jülich, Jülich, 7. und 8.9.2010.

- M. Bolten, *A highly scalable method for Coulomb interactions based on multigrid*, SIAM Conference on Computational Science and Engineering 2011, Reno, NV, USA, 1.3.2011.
- M. Bolten, *Parallelization of a multigrid method for PDEs with open boundary conditions*, 15th Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, USA, 30.3.2011.
- M. Bolten, *Multigrid Methods for Long-Range Interactions*, CECAM Summer School 2011, Forschungszentrum Jülich, Jülich, 12.9.2011.

Literaturverzeichnis

- [60] X. Andrade, J. Alberdi-Rodriguez, D. A. Strubbe, M. J. T. Oliveira, F. Nogueira, A. Castro, J. Muguerza, A. Arruabarrena, S. G. Louie, A. Aspuru-Guzik, A. Rubio und M. A. L. Marques: *Time-dependent Density-functional Theory in Massively Parallel Computer Architectures: The Octopus Project*. J. Phys.: Condens. Matter, 24(23):233202, 2012.
- [61] A. Arnold und C. Holm: *MMM2D: A Fast and Accurate Summation Method for Electrostatic Interactions in 2D Slab Geometries*. Comput. Phys. Commun., 148(3):327–348, Jan. 2002.
- [62] A. Arnold und C. Holm: *MMM1D: A Method for Calculating Electrostatic Interactions in 1D Periodic Geometries*. J. Chem. Phys., 123(12):144103, Sep. 2005.
- [63] A. Arnold, O. Lenz, S. Kesselheim, R. Weeber, F. Fahrenberger, D. Roehm, P. Kosovan und C. Holm: *ESPResSo 3.1 – Molecular Dynamics Software for Coarse-Grained Models*. 2012.
- [64] V. Ballenegger, J. J. Cerdà, O. Lenz und C. Holm: *The Optimal P3M Algorithm for Computing Electrostatic Energies in Periodic Systems*. J. Chem. Phys., 128(3):034109, 2008.
- [65] M. Bolten: *Hierarchical Grid Coarsening for the Solution of the Poisson Equation in Free Space*. Elec. Trans. on Num. Anal., 29:70–80, 2008.
- [66] M. Bolten: *Multigrid Methods for Structured Grids and Their Application in Particle Simulation*. Dissertation, Bergische Universität Wuppertal, 2008.
- [67] P. Brommer, P. Beck, A. Chatzopoulos, F. Gähler, J. Roth und H. R. Trebin: *Direct Wolf Summation of a Polarizable Force Field for Silica*. J. Chem. Phys., 132:194109, 2010.
- [68] R. Car und M. Parrinello: *Unified Approach For Molecular Dynamics and Density Functional Theory*. Phys. Rev. Lett., 55:2471, 1985.
- [69] N. Cardoso, P. J. Silva, P. Bicudo und O. Oliveira: *Landau Gauge Fixing on GPUs*. Comput. Phys. Commun., 2012. (in press).

- [70] J. J. Cerdà, V. Ballenegger, O. Lenz und C. Holm: *P3M Algorithm for Dipolar Interactions*. J. Chem. Phys., 129:234104, 2008.
- [71] D. S. Cerutti, R. E. Duke, T. A. Darden. und T. P. Lybrand: *Staggered Mesh Ewald: An Extension of the Smooth Particle-Mesh Ewald Method Adding Great Versatility*. J. Chem. Theo. and Comp., 5(9):2322–2338, 2009.
- [72] A. Chandramowliswaran, K. Madduri und R. Vuduc: *Diagnosis, Tuning, and Redesign for Multicore Performance: A Case Study of the Fast Multipole Method*. In: *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC '10*, S. 1–12, Washington, DC, USA, 2010. IEEE Computer Society, ISBN 978-1-4244-7559-9.
- [73] H. Dachsel: “*An Error-Controlled Fast Multipole Method*”. J. Chem. Phys., 132(11):119901, 2010.
- [74] H. Dachsel, M. Hofmann und G. Runger: *Library Support for Parallel Sorting in Scientific Computations*. In: *Proc. of the 13th International Euro-Par Conference*, Bd. 4641 d. Reihe LNCS, S. 695–704. Springer, 2007, ISBN 978-3-540-74465-8.
- [75] T. Darden, D. York und L. Pedersen: *Particle Mesh Ewald: An $N \log(N)$ method for Ewald sums in large systems*. J. Chem. Phys., 98:10089–10092, 1993.
- [76] M. Deserno und C. Holm: *How to Mesh up Ewald Sums. I. A Theoretical and Numerical Comparison of Various Particle Mesh Routines*. J. Chem. Phys., 109:7678, 1998.
- [77] M. Deserno und C. Holm: *How to Mesh up Ewald Sums. II. An Accurate Error Estimate for the Particle-Particle-Particle-Mesh Algorithm*. J. Chem. Phys., 109:7694, 1998.
- [78] A. Duncan, R. D. Sedgewick und R. D. Coalson: *Improved Local Lattice Approach for Coulombic Simulations*. Phys. Rev. E, 71(4):046702, April 2005.
- [79] A. Edelman, P. McCorquodale und S. Toledo: *The Future Fast Fourier Transform?* SIAM J. Sc. Comp., 20(3):1094–1114, 1999.
- [80] M. Eleftheriou, J. E. Moreira, B. G. Fitch und R. S. Germain: *Parallel FFT subroutine library*. <http://www.alphaworks.ibm.com/tech/bg13dfft>.
- [81] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee und L. Pedersen: *A Smooth Particle Mesh Ewald Method*. J. Chem. Phys., 103:8577, 1995.

- [82] P. P. Ewald: *Die Berechnung optischer und elektrostatischer Gitterpotentiale*. Ann. Phys., 369(3):253–287, 1921.
- [83] J. Freche, W. Frings und G. Sutmann: *High Throughput Parallel-I/O using SIONlib for Mesoscopic Particles Dynamics Simulations on Massively Parallel Computers*. In: *Proc. of Intern. Conf. ParCo 2009*, S. 423, Amsterdam, 2010. IOS Press.
- [84] M. Frigo und S. G. Johnson: *FFTW, C Subroutine Library*. <http://www.fftw.org>.
- [85] J. R. Hammond, J. Dinan, P. Balaji, I. Kabadshow, S. Potluri und V. Tipparaju: *OS-PRI: An Optimized One-Sided Communication Runtime for Leadership-Class Machines*. In: *The 6th Conference on Partitioned Global Address Space Programming Models (PGAS)*, Okt. 2012.
- [86] F. Hedman und A. Laaksonen: *Ewald Summation Based on Nonuniform Fast Fourier Transform*. Chem. Phys. Lett., 425:142 – 147, 2006.
- [87] R. W. Hockney und J. W. Eastwood: *Computer Simulation Using Particles*. McGraw-Hill, New York, 1981.
- [88] I. Kabadshow, H. Dachsels und J. Hammond: *Poster: Passing the Three Trillion Particle Limit with an Error-Controlled Fast Multipole Method*. In: *Proceedings of the 2011 companion on High Performance Computing Networking, Storage and Analysis Companion, SC '11 Companion*, S. 73–74, New York, NY, USA, 2011. ACM, ISBN 978-1-4503-1030-7.
- [89] J. Keiner, S. Kunis und D. Potts: *NFFT 3.0, C Subroutine Library*. <http://www.tu-chemnitz.de/~potts/nfft>.
- [90] J. Keiner, S. Kunis und D. Potts: *Using NFFT3 - a Software Library for Various Nonequispaced Fast Fourier Transforms*. ACM Trans. Math. Software, 36:Article 19, 1 – 30, 2009.
- [91] I. Lashuk, A. Chandramowlishwaran, H. Langston, T. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin und G. Biros: *A massively parallel adaptive fast multipole method on heterogeneous architectures*. Commun. ACM, 55(5):101–109, Mai 2012, ISSN 0001-0782.
- [92] N. Li: *2DECOMP&FFT, Parallel FFT Subroutine Library*. <http://www.2decomp.org>.

- [93] H. J. Limbach, A. Arnold, B. A. Mann und C. Holm: *ESPResSo – An Extensible Simulation Package for Research on Soft Matter Systems*. *Comput. Phys. Commun.*, 174(9):704–727, Mai 2006.
- [94] A. C. Maggs und V. Rossetto: *Local Simulation Algorithms for Coulomb Interactions*. *Phys. Rev. Lett.*, 88(19):196402, May 2002.
- [95] A. Moroz: *Quasi-Periodic Green’s Functions of the Helmholtz and Laplace Equations*. *J. Phys A*, 39:11247–11282, 2006.
- [96] J. Nieplocha, V. Tipparaju, M. Krishnan und D. K. Panda: *High Performance Remote Memory Access Communication: The Armci Approach*. *IJHPCA*, 20(2):233–253, 2006.
- [97] I. Pasichnyk und B. Dunweg: *Coulomb Interactions via Local Dynamics: A Molecular-Dynamics Algorithm*. *J. Phys.: Condens. Matter*, 16(38):S3999–S4020, September 2004.
- [98] S. Pasquali und A. C. Maggs: *Numerical Studies of Lifshitz Interactions Between Dielectrics*. *Phys. Rev. A: At., Mol., Opt. Phys.*, 79, 2009.
- [99] D. Pekurovsky: *P3DFFT, Parallel FFT Subroutine Library*. <http://code.google.com/p/p3dfft>.
- [100] S. Plimpton: *Parallel FFT Subroutine Library*. <http://www.sandia.gov/~sjplimp/docs/fft/README.html>.
- [101] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2. Aufl., 1992.
- [102] J. Rottler und B. Krayenhoff: *Numerical Studies of Nonlocal Electrostatic Effects on the Sub-nanoscale*. *J. Phys.: Condens. Matter*, 21, 2009.
- [103] J. Rottler und A. C. Maggs: *Long-ranged Electrostatics from Local Algorithms*. *Soft Matter*, 7:3260–3267, 2011.
- [104] J. Stadler, R. Mikulla und H. R. Trebin: *IMD: a Software Package for Molecular Dynamics Studies on Parallel Computers*. *Int. J. Mod. Phys. C.*, 8(5):1131–1140, 1997.
- [105] G. Sutmann und W. Frings: *Extending Scalability of MP²C to more than 250k Compute Cores*. *Techn. Ber. FZJ-JSC-IB-2010-01*, Jülich Supercomputing Centre, 2010.

- [106] U. Trottenberg, C. W. Oosterlee und A. Schuller: *Multigrid*. 2000.
- [107] T. Washio und C. W. Oosterlee: *Error Analysis for a Potential Problem on Locally Refined Grid*. Numer. Math., 86:539–563, 2000.
- [108] C. A. White und M. Head-Gordon: *Rotating around the Quartic Angular Momentum Barrier in Fast Multipole Method Calculations*. J. Chem. Phys., 105(12):5061–5067, 1996.
- [109] A. H. Widmann und D. B. Adolf: *A Comparison of Ewald Summation Techniques for Planar Surfaces*. Comput. Phys. Commun., 107:167–186, 1997.
- [110] D. Wolf, P. Keblinski, S. R. Phillpot und J. Eggebrecht: *Exact Method for the Simulation of Coulombic Systems by Spherically Truncated, Pairwise $1/r$ Summation*. J. Chem. Phys., 110(17):8254–8282, 1999.
- [111] R. Yokota, J. Bardhan, M. Knepley, L. Barba und T. Hamada: *Biomolecular electrostatics using a fast multipole BEM on up to 512 gpus and a billion unknowns*. Comp. Phys. Comm., 182(6):1272 – 1283, 2011, ISSN 0010-4655.