GMD Research Series

GMD –
Forschungszentrum
Informationstechnik
GmbH

Tanja Füllenbach

Mehrgitterverfahren für die zwei- und dreidimensionale Poissongleichung mit periodischen Randbedingungen und eine Anwendung in der Molekulardynamik

N° 11/2000

GMD – Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
http://www.gmd.de

Anschrift der Verfasserin/Address of the author:
Tanja Füllenbach
Institut für Algorithmen und Wissenschaftliches Rechnen
GMD – Forschungszentrum Informationstechnik GmbH
D-53754 Sankt Augustin
E-Mail: tanja.fuellenbach@gmd.de

Die Deutsche Bibliothek - CIP-Einheitsaufnahme:
Füllenbach, Tanja:
Mehrgitterverfahren für die zwei- und dreidimensionale Poissongleichung mit periodischen Randbedingungen und eine Anwendung in der Molekulardynamik / Tanja Füllenbach.
GMD – Forschungszentrum Informationstechnik GmbH. - Sankt Augustin :
GMD – Forschungszentrum Informationstechnik, 2000
(GMD Research Series ; 2000, No. 11)
Zugl.: Köln, Univ., Diplomarbeit, 1999
ISBN 3-88457-383-7

ISSN 1435-2699
ISBN 3-88457-383-7

Kurzfassung

Bioinformatik spielt heutzutage sowohl in der Forschung als auch in der Industrie eine immer größere Rolle. Ein wichtiges Teilgebiet, das hohe Anforderungen an die Soft- und Hardware stellt, ist die Molekulardynamik und darin besonders die Kraftfeldrechnung. Für die dort auftretenden (bis zu) dreidimensionalen Poissongleichungen mit periodischen Randbedingungen sind in der vorliegenden Diplomarbeit geometrische Mehrgitterverfahren entwickelt worden, die genauso effizient wie die schon bekannten schnellen Mehrgitterverfahren für die Poissongleichung mit anderen Randbedingungen sind. Insbesondere weisen sie eine Komplexität (Rechenaufwand) von O(N) auf und sind parallelisierbar. In der Arbeit sind aber auch besonders die Unterschiede und Besonderheiten in der Auswahl geeigneter Komponenten gegenüber anderen Randbedingungen eingehend untersucht worden. Die entwickelten Algorithmen wurden in ein neues Verfahren von Takumi Washio zur Kraftfeldrechnung eingebracht, welches dank der Effizienz der neuen Mehrgitterverfahren eine Komplexität aufweist, die im Gegensatz zu vielen anderen Verfahren im wesentlichen nur von der Anzahl der Teilchen der Basiszelle abhängt. Da es außerdem Werte mit sehr hoher Genauigkeit liefern kann und für Systeme mit Millionen von Teilchen geeignet ist, stellt es vor allem für die Zukunft eine sehr gute Alternative zu den heutzutage eingesetzten FFT- bzw. Multipole-Verfahren dar.

Schlagworte: Molekulardynamik, Kraftfeldrechnung, Poissongleichung, periodische Randbedingungen, geometrisches Mehrgitter

Abstract

Bioinformatics is playing a significant role in both research and industry. An important part in this area is molecular dynamics which requires powerful soft- and hardware, especially for force field calculations. To solve the occurring three-dimensional Poisson equations with periodic boundary conditions, new geometric multigrid approaches are developed in this diploma thesis. They are as efficient as already known multigrid approaches for the Poisson equation with other boundary conditions. For instance, they reach a complexity (computational work) of O(N) and can be parallelized. Particularly special features and differences to other boundary conditions, regarding the choice of multigrid components, are investigated in this thesis. The developed algorithms are built in a new approach proposed by Takumi Washio for computing force fields. Because of the usage of efficient multigrid approaches its complexity is esentially proportional only to the number of particles in the basic cell in contrast to most of the other known approaches. It yields results with high accuracy and can be applied to systems with millions of particles. Therefore it is a promising alternative to nowadays used FFT- and Multipole approaches especially for the next generation of force field calculation software.

Key words: molecular dynamics, force field calculation, Poisson equation, periodic boundary conditions, geometric multigrid

Inhaltsverzeichnis

Einleitung

Die Simulation von Systemen biologisch wichtiger Moleküle wie Enzyme, Proteine, DNA-Stränge und Membranen besonders in der Gegenwart eines Lösungsmittels stellt nach wie vor eine große Herausforderung an die Molekulardynamik dar. Die entsprechenden Systeme bestehen aus etwa tausend bis hin zu vielen Millionen Teilchen. Heutzutage sind zum Beispiel in der Kraftfeldrechnung aber maximal wenige Millionen Partikel simulierbar.

In den Modellen zur Beschreibung des Verhaltens der Systeme treten eine Reihe von partiellen Differentialgleichungen auf. Beispielsweise stößt man auf Poisson- oder Poisson-ähnliche Gleichungen, wenn Größen wie die elektrostatische Energie und das Kraftfeld betrachtet werden. Oft wird die Annahme gemacht, daß ein periodisches System von Teilchen vorliegt, weshalb man an effizienten Lösungsverfahren für die dreidimensionale Poissongleichung mit periodischen Randbedingungen interessiert ist.

In den heutzutage eingesetzten Softwarepaketen für die Kraftfeldrechnung finden sich meist Algorithmen, die hinsichtlich der Anzahl λ der Teilchen etwa bei FFT-basierten Verfahren eine Komplexität $O(\lambda \log(\lambda))$ und bei hierarchischen Multipole-Verfahren eine Komplexität zwischen $O(\lambda)$ und $O(\lambda \log(\lambda))$ erreichen. Hinsichtlich der Anzahl N der Gitterpunkte erreichen FFT-Verfahren für die Poissongleichung mit periodischen Randbedingungen ebenfalls die Komplexität $O(N \log(N))$. Diese Verfahren weisen aber noch Mängel in Bezug auf die Genauigkeit, die Parallelisierbarkeit (FFT) und den Rechenaufwand auf.

Da bekanntermaßen für die Poissongleichung beispielsweise mit Dirichlet- oder Neumannschen Randbedingungen sehr effiziente, gut parallelisierbare Mehrgitter-Verfahren der Komplexität $O(N)$ existieren, macht es Sinn, auch hier die Einsetzbarkeit dieser Methoden zu untersuchen.

Der erste Teil der vorliegenden Diplomarbeit - Kapitel 1 bis 3 - beschäftigt sich mit der systematischen Entwicklung entsprechender Algorithmen für periodische Randbedingungen. Dazu werden eingehende Untersuchungen der ein-, zwei- und dreidimensionalen Aufgabe vorgenommen.

Im ersten Kapitel finden sich zuerst die Problemstellung und Angaben über Existenz und Eindeutigkeit von Lösungen, insbesondere die „analytischen Kompatibilitätsbedingungen“. Dann wird anhand des entsprechenden eindimensionalen Modellproblems untersucht, wie insbesondere die Randbedingungen geeignet diskretisiert werden können. Drei verschiedene Möglichkeiten und ihre jeweiligen Eigenschaften werden diskutiert und miteinander verglichen. Die durch die Diskretisierung des Gesamtproblems entstehenden Gleichungssysteme sind erwartungsgemäß singulär und liefern für ihre Lösbarkeit diskrete Kompatibilitätsbedingungen, die eine Anpassung der rechten Seiten erforderlich machen. Die Auswirkungen auf die numerische Lösung der Systeme werden mit einem direkten und einem iterativen Lösungsverfahren untersucht.

Schon im ersten Kapitel stellt sich heraus, daß man das gegebene periodische Problem am günstigsten auf einem Torus betrachtet. Auch bei der Entwicklung von Mehrgitter-Verfahren für den zwei- und dreidimensionalen Fall in den nächsten Kapiteln wird diese Struktur konsequent ausgenutzt.

Veränderungen der rechten Seiten, die Wahl der Mehrgitter-Komponenten und eine Anpassung von berechneten Vektoren werden in Kapitel 2 eingehend für das zweidimensionale Modellproblem untersucht, besonders im Zusammenhang mit der Lösbarkeit der Gleichungssysteme auf allen betrachteten Gittern. Hinzu kommen numerische Tests der entwickelten Programme, sowohl für die CS- als auch die FMG-Versionen. Ihre Effizienz im Vergleich untereinander und zu anderen Verfahren wird diskutiert. Das dritte Kapitel stellt dann das Analogon zum zweiten Kapitel für das dreidimensionale Modellproblem dar. Die Ergebnisse zeigen deutlich, daß die entwickelten Algorithmen bei geeigneter Komponentenwahl hinsichtlich der Konvergenzraten und des Rechenaufwandes den schnellen Mehrgitter-Verfahren für die Poissongleichung mit Dirichlet-Randbedingungen entsprechen.

Im zweiten Teil der Diplomarbeit, d.h. in den Kapiteln 4 und 5, wird eine Anwendung der entwickelten effizienten Algorithmen im Rahmen einer neuen Methode zur Berechnung der elektrostatischen Energie und des Kraftfeldes periodischer Teilchensysteme vorgestellt.

In Kapitel 4 werden zuerst die zu betrachtenden Systeme und elektrostatischen Größen definiert und das wichtige Konzept der Ewald-Summation erklärt. Um ein neues Verfahren in den Kontext der bereits zur Simulation solcher Systeme existierenden einordnen zu können, erfolgt ebenfalls noch in Kapitel 4 eine Vorstellung der wichtigsten klassischen und neueren Verfahren, insbesondere der FFT- und hierarchischen Multipole-Ansätze.

Die Herleitung der neuen Methode, die von Takumi Washio entwickelt wurde, findet sich zu Beginn des fünften Kapitels. Dargestellt wird dabei besonders eine Aufteilung der zu berechnenden Summe, die von der Idee her Ähnlichkeiten mit der Ewald-Summation aufweist. Die Diskretisierung des Verfahrens, die Parameterwahl und der Ablauf des Algorithmus werden beschrieben, und anschließend gezeigt, daß es sich um ein $O(\lambda)$-Verfahren handelt. In Untersuchungen zur Genauigkeit der berechneten Werte in Abhängigkeit der Parameter zeigt sich nochmals die Effizienz des eingesetzten Mehrgitter-Verfahrens. Schließlich wird in einem Ausblick kurz auf weitere nötige und mögliche Arbeiten zur Verbesserung des Verfahrens und zu seiner Ausdehnung auf realistischere Systeme eingegangen.

Herzlich bedanken möchte ich mich bei Prof. Dr. Ulrich Trottenberg, der diese Arbeit am Institut SCAI der GMD - Forschungszentrum Informationstechnik GmbH ermöglichte. Mein Dank gilt ebenfalls Horst Schwichtenberg und Priv. Doz. Dr. Kees Oosterlee für die ausgezeichnete Betreuung und auch vielen anderen Mitarbeitern von SCAI für die interessanten Diskussionen und Anregungen.

Bemerkung: Aus Platzgründen sind die erstellten FORTRAN-Programme nicht in dieser Arbeit abgedruckt (siehe aber Abschnitt 5.5, Bemerkung).

Kapitel 1

Problemstellung und Vorüberlegungen

1.1 Problemstellung

1.1.1 Theoretische Betrachtung

Betrachtet wird im Gebiet $\Omega = \mathbb{R}^3$ die dreidimensionale Poissongleichung

$$-u_{xx} - u_{yy} - u_{zz} = f \quad \text{in } \Omega\,. \tag{1.1}$$

Hier wird zu vorgegebener Funktion f möglichst eine Lösung $u \in C_t^2(\mathbb{R}^3, \mathbb{R})$ gesucht, wobei für $n \in \mathbb{N}_0$ und $t \in \mathbb{R}_+$ definiert wird:

$$C_t^n(\mathbb{R}^3, \mathbb{R}) = \{u \in C^n(\mathbb{R}^3, \mathbb{R}) \mid \forall\, x, y, z \in \mathbb{R} \,:\, u(\cdot, y, z),\, u(x, \cdot, z) \text{ und } u(x, y, \cdot)\ t\text{-periodisch}\}\,.$$

Dabei soll eine Funktion t-periodisch heißen, wenn sie eine Periode der Länge t besitzt. Funktionen aus $C_t^n(\mathbb{R}^3, \mathbb{R})$ werden im folgenden ebenfalls t-periodisch genannt. Notwendige Bedingungen für die Existenz von Lösungen gibt der folgende Satz an:

Satz 1 (dreidimensionale Kompatibilitätsbedingungen) . *Falls es Lösungen $u \in C_t^2(\mathbb{R}^3, \mathbb{R})$ des Problems (1.1) gibt, so müssen folgende Bedingungen erfüllt sein:*

$$f \in C_t^0(\mathbb{R}^3, \mathbb{R})\,, \tag{1.2}$$

$$\int_{[0,t]^3} f(r)\, d\lambda^3(r) = 0\,. \tag{1.3}$$

Beweis[1]**:** Wegen $u \in C_t^2(\mathbb{R}^3, \mathbb{R})$ gilt insbesondere $u_{xx}, u_{yy}, u_{zz} \in C_t^0(\mathbb{R}^3, \mathbb{R})$. Daraus erhält man (1.2) als eine notwendige Bedingung für die Existenz von

[1] Die Beweisidee für (1.3) findet sich in [5]. Sie ist im Beweis eines speziellen Theorems von Green für periodische Funktionen enthalten. λ bezeichnet das übliche Längenmaß für $\mathbb{R}$. Im folgenden wird statt $d\lambda^d(r)$ oft einfach dr geschrieben.

Lösungen der gestellten Aufgabe. Sei nun $v \in C_t^2(\mathbb{R}^3, \mathbb{R})$ und $K = [0,t]^3$. Definiere dann folgende Funktion:

$$\begin{aligned} g : K \times \mathbb{R}^3 &\to \mathbb{R} \\ (r, \tilde{r}) &\mapsto v(r + \tilde{r}) \; . \end{aligned}$$

Dann gilt also

$$\forall\, r \in K \; : \; g(r, \cdot) \in C_t^2(\mathbb{R}^3, \mathbb{R}) \, .$$

Da außerdem K kompakt ist, kann man den Satz über die differenzierbare Abhängigkeit eines Integrals von einem Parameter (vergleiche [16]) zweimal auf die Funktion

$$\begin{aligned} I : \mathbb{R}^3 &\to \mathbb{R} \\ \tilde{r} &\mapsto \int_K g(r, \tilde{r}) \, d\lambda^3(r) = \int_K v(r + \tilde{r}) \, d\lambda^3(r) \end{aligned}$$

anwenden und erhält so

$$\forall\, \tilde{r} \in \mathbb{R}^3 \; : \; (I_{xx} + I_{yy} + I_{zz})(\tilde{r}) = \int_K (v_{xx} + v_{yy} + v_{zz})(r + \tilde{r}) \, d\lambda^3(r) \; .$$

Weil aber offensichtlich I konstant ist, ergibt sich

$$I_{xx} + I_{yy} + I_{zz} \equiv 0$$

und somit

$$\forall\, \tilde{r} \in \mathbb{R}^3 \; : \; \int_K (v_{xx} + v_{yy} + v_{zz})(r + \tilde{r}) \, d\lambda^3(r) = 0 \; .$$

Insbesondere für $\tilde{r} = (0,0,0)$ gilt dann

$$\int_K (v_{xx} + v_{yy} + v_{zz})(r) \, d\lambda^3(r) = 0 \; .$$

Wählt man nun $v = -u$, so erhält man die Bedingung (1.3). □

Offensichtlich sind mit einem u auch alle $u + const.$ Lösungen des Problems. Der folgende Satz 2 (vergleiche [6]) gibt nun an, daß dies dann alle Lösungen sein müssen, d.h. die Lösung - falls sie existiert - bis auf eine Konstante eindeutig bestimmt ist. Außerdem liefert er notwendige und hinreichende Bedingungen für die Existenz zumindest „schwacher“ Lösungen (für ihre Definition vergleiche z.B. [6, 3]):

Satz 2 . *Sei (M, g) eine orientierbare kompakte Riemannsche C^∞-Mannigfaltigkeit der Dimension d mit Riemannscher C^∞-Metrik g, die in einer lokalen Karte die Funktionen $g_{ij} = g_{ij}(\xi)$ als Koeffizienten besitzt. Weiterhin ist dann in dieser lokalen Karte*

$$\begin{aligned} (g^{ij}) &:= (g_{ij})^{-1} \; , \\ \sqrt{g} &:= \det (g_{ij}) \; . \end{aligned}$$

Dann bezeichne $H^{2,1}(M)$ den Sobolevraum mit $p = 2$ und $m = 1$ und $C^{n+\alpha}(M)$ den Hölderraum ($0 < \alpha \leq 1$) (siehe zur Definition von Sobolevräumen und Hölderräumen auf Riemannschen Mannigfaltigkeiten [6]). Genau dann existiert eine schwache Lösung $u \in H^{2,1}(M)$ des Laplace-Beltrami-Operators (vergleiche [40])

$$-\frac{1}{\sqrt{g}} \sum_{j=1}^{d} \frac{\partial}{\partial \xi^j} \left(\sqrt{g} \sum_{k=1}^{d} g^{jk} \frac{\partial u}{\partial \xi^k} \right) = f$$

mit $f \in L^2(M)$, wenn $\int f(x) dV = 0$ gilt. Die Lösung u ist dabei bis auf eine Konstante eindeutig bestimmt. Falls zusätzlich $f \in C^{n+\alpha}(M)$ mit $n \in \mathbb{N}_0 \cup \{\infty\}$ ist, so ist $u \in C^{n+2+\alpha}(M)$.

Bemerkung: In einer anderen Karte müssen die g_{ij} die Transformationsformel (vergleiche [40])

$$\tilde{g}_{ij}(x(\xi)) = \sum_{k,l} \frac{\partial \xi^k}{\partial x_i} \frac{\partial \xi^l}{\partial x_j} g_{kl}(\xi) \tag{1.4}$$

für den Kartenwechsel ($\xi \mapsto x(\xi)$) erfüllen. Hat man einen Atlas für M und C^∞-Funktionen g_{ij} für die Karten gegeben, so daß die Bedingung (1.4) erfüllt ist, so definieren die entsprechenden $g_{ij}(\xi)$ dann eine Riemannsche Metrik auf M, wenn die Matrizen $(g_{ij}(\xi))_{i,j}$ symmetrisch positiv definit sind.

Der Satz ist auf Problem (1.1) anwendbar. Die 1-periodischen Funktionen können (kanonisch) als Funktionen auf dem Torus $T^3 := \mathbb{R}^3/\mathbb{Z}^3$ identifiziert werden, weil sie genau diejenigen Funktionen sind, die durch die Operation von $\mathbb{Z}^3$ auf $\mathbb{R}^3$ invariant gelassen werden. Als Quotient aus einer Liegruppe ($\mathbb{R}^3$) und einer abgeschlossenen Untergruppe ($\mathbb{Z}^3$) dieser Liegruppe besitzt T^3 die Struktur einer C^∞-Mannigfaltigkeit (der Dimension d=3) (siehe [61]). Die Topologie auf T^3 ist die Identifizierungstopologie [52], d.h. die feinste Topologie von T^3, so daß die surjektive Abbildung $\Pi : \mathbb{R}^3 \to T^3, \xi \mapsto \xi + \mathbb{Z}^3$ stetig ist.[2] Lokale Karten um $[x_0] \in T^3$ werden in natürlicher Weise gegeben durch

$$\phi_{[x_0]} : U = (0,1)^3 \to T^3, \quad \xi \mapsto [x_0 + \xi] \,. \tag{1.5}$$

Da $[0,1]^3$ kompakt, die Abbildung $\pi := \Pi|_{[0,1]^3}$ stetig und $\pi([0,1]^3) = T^3$ ist, ist auch T^3 kompakt. Definiert man in einer lokalen Karte die Koeffizienten der Riemannschen Metrik durch $g_{ij} = \delta_{ij}$ (mit dem Kronecker-Symbol δ_{ij}), so sieht man leicht (da Kartenwechsel die Gestalt $\phi_{[y_0]}^{-1} \; \phi_{[x_0]} : \xi \mapsto x(\xi) = x_0 - y_0 + \xi$ für Karten der Form (1.5) haben), daß die g_{ij} durch Kartenwechsel ineinander übergehen. Damit hat man also aufgrund obiger Bemerkung eine Riemannsche Metrik definiert. Offenbar ist $\sqrt{g} = 1$, und aus der Gleichung $(g^{ij})_{i,j} = (g_{ij})_{i,j}^{-1}$ erhält man $g^{ij} = \delta_{ij}$. Die Betrachtung der obigen Kartenwechsel ergibt außerdem, daß T^3 orientierbar ist. Insgesamt sieht man

[2] Daß Π die identifizierende Abbildung der Topologie ist, ist äquivalent dazu, daß eine Menge $U \subseteq T^3$ genau dann offen ist, wenn ihr Urbild bezüglich Π offen ist.

nun, daß der Satz 2 also im hier vorliegenden Fall anwendbar ist und die Lösungen u des Laplace-Beltrami-Operators im Satz gerade den Lösungen der dreidimensionalen Poissongleichung mit periodischen Randbedingungen (siehe (1.1)f.) entsprechen.

Bemerkung: Durch die obigen Karten sieht man, daß T^3 lokal isometrisch isomorph zu offenen Teilmengen des $\mathbb{R}^3$ (mit dem euklidischen Skalarprodukt) ist, und deswegen flach. T^3 ist jedoch keine eingebettete Mannigfaltigkeit des $\mathbb{R}^3$, sondern wegen $T^3 \cong S^1 \times S^1 \times S^1$ im $\mathbb{R}^6$ eingebettet.

1.1.2 Zur numerischen Lösung

Offenbar genügt es, eine Lösung u z.B. auf $\overline{\Omega}$ zu kennen, wobei Ω ein Gebiet mit $[0,t]^3 \subseteq \overline{\Omega} \subseteq \mathbb{R}^3$ sei. Da für $u \in C_t^2(\mathbb{R}^3, \mathbb{R})$ bereits alle $\frac{\partial u}{\partial v}$, $\frac{\partial^2 u}{\partial v \partial w}$ mit $v, w \in \{x, y, z\}$ t-periodisch sind, muß $u|_{\overline{\Omega}}$ bei der Wahl von $t = 1$ insbesondere eine Lösung der folgenden Aufgabe sein:

$$-u_{xx} - u_{yy} - u_{zz} = f \quad \text{in } \Omega \tag{1.6}$$

mit periodischen Randbedingungen in $\partial[0,1]^3$:

$$\begin{aligned} u(0,y,z) &= u(1,y,z)\,, \\ u(x,0,z) &= u(x,1,z)\,, \\ u(x,y,0) &= u(x,y,1) \end{aligned} \tag{1.7}$$

und

$$\begin{aligned} u_x(0,y,z) &= u_x(1,y,z)\,, \\ u_y(x,0,z) &= u_y(x,1,z)\,, \\ u_z(x,y,0) &= u_z(x,y,1)\,. \end{aligned} \tag{1.8}$$

Um dieses Problem numerisch mit Hilfe eines Mehrgitter-Verfahrens zu lösen, muß es diskretisiert werden. Dazu werden die den Größen u und f entsprechenden diskreten Funktionen u_h und f_h (siehe auch Abschnitte 1.2.3, 2.1.3 und 3.1.1) auf einem unendlichen Gitter

$$G_h = \{(jh, kh, lh) \mid j, k, l \in \mathbb{Z}\}$$

der Maschenweite h betrachtet. Dabei ist $h = \frac{1}{N}$ mit $N = 2^p$, $p \in \mathbb{N}$. Die übliche 7-Punkte-Diskretisierung zweiter Ordnung[3] von (1.6) lautet dann in Sternschreibweise:

$$\frac{1}{h^2} \left[\begin{array}{c|c|c} & & -1 \\ \hline -1 & -1 \quad 6 \quad -1 & -1 \\ \hline & & -1 \end{array}\right]_h u_h = f_h \quad \text{in } \Omega_h = G_h \cap \Omega\,. \tag{1.9}$$

Die Diskretisierung von (1.7) liegt auf der Hand, aber für die Bedingung (1.8) bieten sich erst einmal mehrere Möglichkeiten an.

[3] falls $u \in C^4(\overline{\Omega})$, also insbesondere $f \in C^2(\overline{\Omega})$ ist.

Da (1.9) einen lokalen Diskretisierungsfehler zweiter Ordnung aufweist, lautet ein Ziel, dies auch für (1.8) zu erreichen, um *insgesamt* ein Verfahren zweiter Ordnung zu erhalten. Andererseits soll dabei aber der Arbeitsaufwand möglichst klein gehalten und der Periodizität der gesuchten Lösung Rechnung getragen werden.

Im folgenden werden nun drei naheliegende Diskretisierungsmöglichkeiten (im eindimensionalen Fall) untersucht und miteinander verglichen.

1.2 Eindimensionales Modellproblem

1.2.1 Exakte Lösbarkeit

Dazu genügt es, das zum obigen Problem (1.6) - (1.8) korrespondierende eindimensionale Modellproblem zu betrachten, nämlich die eindimensionale Poissongleichung mit periodischen „Rand"bedingungen in $\overline{\Omega}$, wobei Ω ein offenes Intervall mit $]0,1[\subseteq \Omega \subseteq \mathbb{R}$ sei:

$$-u''(x) = f(x) \quad \text{in } \Omega\,, \tag{1.10}$$

$$u(0) = u(1)\,, \tag{1.11}$$

$$u'(0) = u'(1)\,. \tag{1.12}$$

Lösungen von (1.10)-(1.12) mit $\Omega =]0,1[$ und $f \in C([0,1])$ existieren genau dann, wenn die (eindimensionale) Kompatibilitätsbedingung

$$\int_0^1 f(s)\,ds = 0 \tag{1.13}$$

erfüllt ist. Sie haben dann die folgende Form:

$$u(x) = -\int_0^x \int_0^t f(s)\,ds\,dt + x\int_0^1 \int_0^t f(s)\,ds\,dt + c \text{ mit } c \in \mathbb{R}\,.$$

Um u zu einer Funktion aus $C_1^2(\mathbb{R},\mathbb{R})$ fortsetzen zu können, ist die (zusätzliche) Bedingung $f(0) = f(1)$ notwendig und hinreichend, d.h. genauer: Die Fortsetzung von f muß in $C_1^0(\mathbb{R},\mathbb{R})$ liegen. Die eindimensionalen Analoga zu (1.2) und (1.3) sind also notwendig und sogar hinreichend, um Lösungen u von $-u'' = f$ aus $C_1^2(\mathbb{R},\mathbb{R})$ zu erhalten.

1.2.2 Diskretisierungsmöglichkeiten

Die zu (1.9) analoge Diskretisierung zweiter Ordnung von (1.10) lautet:

$$\frac{1}{h^2}[-1 \quad 2 \quad -1]_h u_h = f_h \quad \text{in } \Omega_h\,. \tag{1.14}$$

Alle Bezeichnungen seien dabei analog dem dreidimensionalen Fall. Außerdem sei im folgenden immer $x_j = jh = \frac{j}{N}$.

Wie man durch Taylorreihenentwicklung sehen kann, gilt für eine Funktion $v \in C^2(I)$, wobei $I \subseteq \mathbb{R}$ ein (offenes) Intervall mit $x, x \pm t \in I$ und $t \neq 0$ sei:

$$v'(x) = \frac{1}{t}[v(x+t) - v(x)] + O(t) \,, \tag{1.15}$$

$$v'(x) = \frac{1}{t}[v(x) - v(x-t)] + O(t) \,, \tag{1.16}$$

$$v'(x) = \frac{1}{2t}[v(x+t) - v(x-t)] + O(t^2) \,. \tag{1.17}$$

Benutzt man also für $u'(0)$ und $u'(1)$ (1.15), so erhält man für (1.12) folgende Bedingung mit einem lokalen Diskretisierungsfehler erster Ordnung:

$$\begin{aligned} &\frac{1}{h}[u_h(h) - u_h(0)] = \frac{1}{h}[u_h(1+h) - u_h(1)] \\ \Leftrightarrow\ &-u_h(h) + u_h(1+h) = 0 \quad \text{wegen (1.11).} \end{aligned}$$

$u'(0)$ und $u'(1)$ werden also in „dieselbe Richtung“ diskretisiert. Hierbei muß $\overline{\Omega}_h$ auch den Punkt $x_{N+1} = 1+h$ enthalten. Man kann statt der Bedingung (1.12) auch gleich für die zu berechnende Näherungslösung u_h die geforderte Periodizität von u berücksichtigen (d.h. auf einem eindimensionalen Torus rechnen) und erhält auf diese Weise ebenfalls die Bedingung

$$-u_h(h) + u_h(1+h) = 0 \,. \tag{1.18}$$

Diese Vorgehensweise ändert dann an der Konsistenzordnung des gesamten Verfahrens nichts: Es bleibt zweiter Ordnung.

Wenn man aber für $u'(0)$ (1.15) und für $u'(1)$ (1.16) benutzt, so erhält man für (1.12) folgende Bedingung mit einem lokalen Diskretisierungsfehler erster Ordnung:

$$\begin{aligned} &\frac{1}{h}(u_h(h) - u_h(0)) = \frac{1}{h}(u_h(1) - u_h(1-h)) \\ \Leftrightarrow\ &-u_h(h) - u_h(1-h) + 2u_h(1) = 0 \,. \end{aligned} \tag{1.19}$$

Hier wird an beiden Seiten ins Innere des Intervalls diskretisiert.

Benutzt man schließlich für $u'(0)$ und $u'(1)$ (1.17), so erhält man für (1.12) folgende Bedingung mit einem lokalen Diskretisierungsfehler zweiter Ordnung:

$$\begin{aligned} &\frac{1}{2h}(u_h(h) - u_h(-h)) = \frac{1}{2h}(u_h(1+h) - u_h(1-h)) \\ \Leftrightarrow\ &-u_h(-h) + u_h(h) + u_h(1-h) - u_h(1+h) = 0 \,. \end{aligned} \tag{1.20}$$

Hier muß $x_{-1}, x_{N+1} \in \overline{\Omega}_h$ gelten. In diesem Fall wird also zentral über den jeweiligen Randpunkt diskretisiert. Diese Bedingung kann man übrigens wie schon (1.18) aus der verlangten Periodizität von u erhalten.

Man kann also erwarten, daß sich Version (1.18) als die günstigste und Version (1.19) als die ungünstigste erweist. Zur genauen Klärung müssen nun

alle drei Möglichkeiten weiter untersucht werden.

In den weiteren Abschnitten wird die Lösungsfunktion u_h des jeweiligen diskreten Problems mit dem Vektor $(u_h(x_j))_{j\in J}^T$ mit $J = \{k \,|\, x_k \in \overline{\Omega}_h\}$ identifiziert.

1.2.3 Zur ersten Diskretisierungsmöglichkeit

Hier liefert die Diskretisierung ein lineares Gleichungssystem, das aus (1.18), (1.11) und (1.14) für die Stellen $u_h(x_1), \ldots, u_h(x_N)$ besteht:

$$\underbrace{\left[\begin{array}{c|cccccc|c} 1 & & & & & & -1 & \\ \hline -1 & 2 & -1 & & & & & \\ & -1 & 2 & -1 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & -1 & 2 & -1 & & \\ & & & & -1 & 2 & -1 & \\ \hline & -1 & & & & & 1 & \end{array}\right]}_{A_h} \underbrace{\left[\begin{array}{c} u_h(x_0) \\ \hline u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-1}) \\ u_h(x_N) \\ \hline u_h(x_{N+1}) \end{array}\right]}_{u_h} = \underbrace{\left[\begin{array}{c} 0 \\ \hline h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) \\ \hline 0 \end{array}\right]}_{r_h}$$

Die Matrix A_h ist singulär, da die Addition der ersten $N+1$ Zeilen das Negative der letzten Zeile ergibt. Daraus ergibt sich eine diskrete Kompatibilitätsbedingung, die für die Lösbarkeit des Gleichungssystems erfüllt sein muß:

$$h^2 \sum_{j=1}^{N} f(x_j) = 0 \,. \tag{1.21}$$

Falls $f \in C^2([0,1])$ ist, gilt wegen $f(x_0) = f(x_N)$ und der Trapezformel:

$$\int_0^1 f(s)\, ds = h \sum_{j=1}^{N} f(x_j) + O(h^2) \,.$$

Da f aber die Kompatibilitätsbedingung (1.13) erfüllt, erhält man

$$h^2 \sum_{j=1}^{N} f(x_j) = O(h^3) \,.$$

Also führt (1.13) zu einer Näherung dritter Ordnung für (1.21), weswegen im Gegensatz zur kontinuierlichen Kompatibilitätsbedingung (1.13) eine vorgegebene Funktion f die diskrete Kompatibilitätsbedingung im allgemeinen nicht exakt erfüllt. Daher könnte es bei der Lösung des Gleichungssystems (in dieser oder einer äquivalenten Form) zu Problemen kommen. Für diesen Fall werden die Funktionswerte $f(x_j)$ durch Werte $f_h(x_j)$ ersetzt, und zwar in folgender Weise:

$$f_h(x_j) := f(x_j) - \bar{f} \,, \tag{1.22}$$

wobei $\bar{f}$ den Durchschnitt der Werte $f(x_i)$ bezeichne:

$$\bar{f} = \frac{1}{N}\sum_{i=1}^{N} f(x_i) = h\sum_{i=1}^{N} f(x_i)\ .$$

Damit erfüllt die neue Funktion f_h also die diskrete Kompatibilitätsbedingung (1.21) und stellt eine Näherung zweiter Ordnung für $f|_{\Omega_h}$ ($\Omega_h = \{x_1, \ldots, x_N\}$) dar:

$$f(x_j) - f_h(x_j) = h\sum_{i=1}^{N} f(x_i) = \int_0^1 f(s)\,ds + O(h^2) = O(h^2)\ .$$

Zur Auswirkung auf die numerische Lösung des Gleichungssystems vergleiche man für den eindimensionalen Fall Abschnitt 1.2.6 und für den zwei- bzw. dreidimensionalen Fall die Kapitel 2 bzw. 3.

Zur Vereinfachung des Gleichungssystems werden nun die Randpunkte $u_h(x_0)$ und $u_h(x_{N+1})$ eliminiert:

$$A_h u_h = r_h$$

$$\Leftrightarrow \left[\begin{array}{c|ccccc|c} 1 & & & & & -1 & \\ \hline & 2 & -1 & & & -1 & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & -1 & & & -1 & 2 & \\ \hline & -1 & & & & & 1 \end{array}\right] \left[\begin{array}{c} u_h(x_0) \\ \hline u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-1}) \\ u_h(x_N) \\ \hline u_h(x_{N+1}) \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) \\ \hline 0 \end{array}\right]$$

Das innere Gleichungssystem

$$\widetilde{A}_h \begin{bmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-1}) \\ u_h(x_N) \end{bmatrix} = \begin{bmatrix} h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) \end{bmatrix} \tag{1.23}$$

mit

$$\widetilde{A}_h = \begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix}$$

ist also unabhängig von den Randpunkten.

Um ein auf diese Diskretisierung des Problems angewandtes Mehrgitter-Verfahren theoretisch mit Hilfe der (rigorosen) Fourier-Analyse (vergleiche etwa [58]) untersuchen zu können und zusätzlich $Kern(\widetilde{A}_h)$ zu bestimmen,

ist es nötig, die Eigenfunktionen und zugehörigen Eigenwerte von $\widetilde{A}_h$ zu kennen. Darum werden sie nun bestimmt. Es gilt $\forall\, j,k \in \mathbb{Z}$:

$$\begin{aligned}&-\exp(i2k\pi x_{j-1}) + 2\exp(i2k\pi x_j) - \exp(i2k\pi x_{j+1})\\ =&(2-\exp(-i2k\pi h) - \exp(i2k\pi h))\exp(i2k\pi x_j)\\ =&(2-2\cos(2k\pi h))\exp(i2k\pi x_j)\ .\end{aligned}$$

Weil die Exponentialfunktion $2\pi i$-periodisch ist, gilt außerdem:

$$\exp(i2k\pi x_j) = \exp(i2k\pi x_{j+N})\ .$$

Deswegen ist

$$(\exp(i2k\pi x_j))_{j=0,1,\ldots,N-1}$$

Eigenvektor bzw. ϕ_k mit $\phi_k(x) = \exp(i2k\pi x)$ Eigenfunktion von $\widetilde{A}_h$ zum Eigenwert $\lambda_k = 2-2\cos(2k\pi h)$. Falls N gerade ist, erhält man so für z.B. $-\frac{N}{2} < k \leq \frac{N}{2}$ N *linear unabhängige* Eigenfunktionen ϕ_k zu den Eigenwerten λ_k. Somit sind also *alle* Eigenwerte von $\widetilde{A}_h$ gefunden, und insgesamt gilt

$$Eig(\widetilde{A}_h, 2-2\cos(2k\pi h)) = span\left\{\begin{pmatrix}\exp(i2k\pi x_0)\\ \vdots\\ \exp(i2k\pi x_{N-1})\end{pmatrix}, \begin{pmatrix}\exp(-i2k\pi x_0)\\ \vdots\\ \exp(-i2k\pi x_{N-1})\end{pmatrix}\right\}$$

Dabei sind die Eigenräume zu $\lambda_0 = 0$ und $\lambda_{\frac{N}{2}} = 4$ eindimensional und alle anderen zweidimensional, weil *cos* y-achsensymmetrisch ist und daher $\lambda_k = \lambda_{-k}$ gilt. Insgesamt zeigen die obigen Ausführungen also die Gültigkeit von

Lemma 1 . *Das Gleichungssystem $A_h u_h = r_h$ ist genau dann lösbar, wenn die Kompatibilitätsbedingung* (1.21) *erfüllt ist. Dabei gilt für Lösungen u_h und $\tilde{u}_h$:*

$$u_h - \tilde{u}_h \in Kern(A_h) = span\left\{(1,1,\ldots,1)^T\right\}\ .$$

Offensichtlich sind die λ_k alle nicht negativ. Damit ist insgesamt $\widetilde{A}_h$ symmetrisch, positiv semidefinit und (schwach) diagonaldominant, außerdem eine Tridiagonalmatrix mit zwei zusätzlichen Einträgen, insbesondere also dünnbesetzt.

Für die zweidimensionale Poissongleichung

$$-u_{xx} - u_{yy} = f \quad \text{in } \Omega \supseteq]0,a_1[\times]0,a_2[\tag{1.24}$$

mit periodischen Randbedingungen in $\partial([0,a_1]\times[0,a_2])$:

$$\begin{aligned}u(0,y) &= u(a_1,y)\ ,\\ u(x,0) &= u(x,a_2)\ ,\end{aligned} \tag{1.25}$$

$$\begin{aligned}u_x(0,y) &= u_x(a_1,y)\ ,\\ u_y(x,0) &= u_y(x,a_2)\ ,\end{aligned} \tag{1.26}$$

kann man ganz analog vorgehen und erhält für entsprechendes $\widetilde{A}_h$,

$$\begin{aligned} &N = (N_1, N_2) \quad \text{mit geradem } N_j, \\ &h = (h_1, h_2) \quad \text{mit} \quad h_j = \frac{a_j}{N_j}, \\ &k = (k_1, k_2) \quad \text{mit} \quad \frac{-N_j}{2} < k_j \leq \frac{N_j}{2} \end{aligned}$$

die Eigenfunktionen

$$\phi_k(x, y) = \exp\,(i2k_1\pi\frac{x}{a_1})\exp\,(i2k_2\pi\frac{y}{a_2})$$

zu den Eigenwerten

$$\lambda_k = 4 - 2\cos\,(2k_1\pi\frac{1}{N_1}) - 2\cos\,(2k_2\pi\frac{1}{N_2})\ .$$

Für das dreidimensionale Problem

$$-u_{xx} - u_{yy} - u_{zz} = f \quad \text{in } \Omega \supseteq =]0, a_1[\times]0, a_2[\times]0, a_3[\tag{1.27}$$

mit periodischen Randbedingungen in $\partial([0, a_1] \times [0, a_2] \times [0, a_3])$:

$$\begin{aligned} u(0, y, z) &= u(a_1, y, z)\ , \\ u(x, 0, z) &= u(x, a_2, z)\ , \\ u(x, y, 0) &= u(x, y, a_3)\ , \end{aligned} \tag{1.28}$$

$$\begin{aligned} u_x(0, y, z) &= u_x(a_1, y, z)\ , \\ u_y(x, 0, z) &= u_y(x, a_2, z)\ , \\ u_z(x, y, 0) &= u_z(x, y, a_3)\ , \end{aligned} \tag{1.29}$$

erhält man für entsprechendes $\widetilde{A}_h$,

$$\begin{aligned} &N = (N_1, N_2, N_3) \quad \text{mit geradem } N_j, \\ &h = (h_1, h_2, h_3) \quad \text{mit} \quad h_j = \frac{a_j}{N_j}, \\ &k = (k_1, k_2, k_3) \quad \text{mit} \quad \frac{-N_j}{2} < k_j \leq \frac{N_j}{2} \end{aligned}$$

die Eigenfunktionen

$$\phi_k(x, y, z) = \exp\,(i2k_1\pi\frac{x}{a_1})\exp\,(i2k_2\pi\frac{y}{a_2})\exp\,(i2k_3\pi\frac{z}{a_3})$$

zu den Eigenwerten

$$\lambda_k = 6 - 2\cos\,(2k_1\pi\frac{1}{N_1}) - 2\cos\,(2k_2\pi\frac{1}{N_2}) - 2\cos\,(2k_3\pi\frac{1}{N_3})\ .$$

Wie schon im eindimensionalen Fall sind die $\widetilde{A}_h$ für den zwei- und dreidimensionalen Fall symmetrisch, positiv semidefinit, (schwach) diagonaldominant und dünnbesetzt. Solche Eigenschaften wirken sich in vielen numerischen Verfahren (besonders den in dieser Arbeit betrachteten) günstig aus.

1.2.4 Zur zweiten Diskretisierungsmöglichkeit

Hier liefert die Diskretisierung ein lineares Gleichungssystem, das aus (1.19), (1.11) und (1.14) für die Stellen $u_h(x_1), \ldots, u_h(x_{N-1})$ besteht:

$$\underbrace{\left[\begin{array}{c|ccccc|c} 1 & & & & & & -1 \\ \hline -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ \hline & -1 & & & & -1 & 2 \end{array}\right]}_{B_h} \underbrace{\left[\begin{array}{c} u_h(x_0) \\ \hline u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-2}) \\ u_h(x_{N-1}) \\ \hline u_h(x_N) \end{array}\right]}_{u_h} = \underbrace{\left[\begin{array}{c} 0 \\ \hline h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-2}) \\ h^2 f(x_{N-1}) \\ \hline 0 \end{array}\right]}_{r_h}$$

Die Matrix B_h ist singulär, da die Addition der ersten N Zeilen das Negative der letzten Zeile ergibt. Analog zum letzten Abschnitt ergibt sich: Wenn das Gleichungssystem lösbar ist, muß gelten:

$$h^2 \sum_{j=1}^{N-1} f(x_j) = 0 \,. \tag{1.30}$$

Anders als bei der ersten Diskretisierung führt hier (1.13) nicht zu einer Näherung für (1.30), weswegen man also nicht erwarten kann, daß die vorgegebene Funktion f die diskrete Kompatibilitätsbedingung (1.30) auch nur annähernd erfüllt. Damit das Gleichungssystem aber lösbar ist, muß $f|_{\Omega_h}$ durch eine Funktion f_h ersetzt werden, die der Kompatibilitätsbedingung (1.30) genügt. Wie im letzten Abschnitt geschieht dies durch Abzug des Mittelwertes $\bar{f}$ von jedem $f(x_i)$. Hier ist aber

$$\bar{f} = \frac{1}{N-1} \sum_{i=1}^{N-1} f(x_i) \,.$$

Für numerische Auswirkungen vergleiche man Abschnitt 1.2.6. Das Gleichungssystem $B_h u_h = r_h$ wird nun durch Eliminierung von $u_h(x_0)$ in folgendes umgeformt:

$$\left[\begin{array}{c|ccccc} 1 & & & & & -1 \\ \hline & 2 & -1 & & & -1 \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & -1 & & & -1 & 2 \end{array}\right] \left[\begin{array}{c} u_h(x_0) \\ \hline u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-1}) \\ u_h(x_N) \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-1}) \\ 0 \end{array}\right]$$

Betrachtet man das sich ergebende Gleichungssystem für $u_h(x_1), \ldots, u_h(x_N)$, so erkennt man die Matrix $\tilde{A}_h$ wieder. Die rechten Seiten unterscheiden sich jedoch in der letzten Komponente: Hier steht 0, in (1.2.1) $h^2 f(1)$. Die Eigenwerte und -vektoren der Matrix $\tilde{A}_h$ sind schon im Abschnitt 1.2.3 bestimmt worden. Daraus ergibt sich, daß die $N \times N$-Matrix $\tilde{A}_h$ den Rang $N-1$ und

somit die $(N+1)\times(N+1)$-Matrix B_h den Rang N besitzt. Außerdem ist offensichtlich $(1,\ldots,1)^T$ Eigenvektor zum Eigenwert 0 (von B_h). Dies liefert insgesamt als Analogon zu Lemma 1:

Lemma 2 . *Das Gleichungssystem $B_h u_h = r_h$ ist genau dann lösbar, wenn die Kompatibilitätsbedingung* (1.30) *erfüllt ist. Dabei gilt für Lösungen u_h und $\tilde{u}_h$:*

$$u_h - \tilde{u}_h \in Kern(B_h) = span\left\{(1,1,\ldots,1)^T\right\} .$$

Durch die Eliminierung des anderen Randpunktes $u_h(x_N)$ (und Weglassen der Gleichungen für $u_h(x_0)$ und $u_h(x_N)$) erhält man:

$$\begin{bmatrix} \frac{3}{2} & -1 & & & \frac{-1}{2} \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ \frac{-1}{2} & & & -1 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-2}) \\ u_h(x_{N-1}) \end{bmatrix} = \begin{bmatrix} h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-2}) \\ h^2 f(x_{N-1}) \end{bmatrix} .$$

Die linksstehende Matrix sei mit $\widetilde{B}_h$ bezeichnet. Die durchgeführten Umformungen haben den Vorteil, daß $\widetilde{B}_h$ symmetrisch und schwach diagonaldominant, außerdem eine Tridiagonalmatrix mit zwei zusätzlichen Einträgen ist (insbesondere also eine dünnbesetzte Matrix). Allerdings scheinen weder $\widetilde{B}_h$ noch dazu äquivalente symmetrische Matrizen einfach zu bestimmende Eigenfunktionen zu haben.

1.2.5 Zur dritten Diskretisierungsmöglichkeit

Hier liefert die Diskretisierung ein lineares Gleichungssystem, das aus (1.20), (1.11) und (1.14) für die Stellen $u_h(x_0),\ldots,u_h(x_N)$ besteht:

$$\underbrace{\left[\begin{array}{c|ccccccc|c} & -1 & & & & & 1 & & \\ \hline -1 & 2 & -1 & & & & & & \\ & -1 & 2 & -1 & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & -1 & 2 & -1 & & \\ & & & & & -1 & 2 & -1 & \\ \hline -1 & & 1 & & & & 1 & & -1 \end{array}\right]}_{C_h} \underbrace{\left[\begin{array}{c} u_h(x_{-1}) \\ \hline u_h(x_0) \\ u_h(x_1) \\ \vdots \\ \vdots \\ u_h(x_{N-1}) \\ u_h(x_N) \\ \hline u_h(x_{N+1}) \end{array}\right]}_{u_h} = \underbrace{\left[\begin{array}{c} 0 \\ \hline h^2 f(x_0) \\ h^2 f(x_1) \\ \vdots \\ \vdots \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) \\ \hline 0 \end{array}\right]}_{r_h}$$

Die Matrix C_h ist singulär: Wenn man nämlich die zweite und vorletzte Zeile und je das Doppelte der mittleren N-1 Zeilen addiert, ergibt sich die letzte Zeile. Hieraus erhält man folgende notwendige Bedingung für die Lösbarkeit des Gleichungssystems:

$$h^2\left(f(x_0) + 2\sum_{j=1}^{N-1} f(x_j) + f(x_N)\right) = 0 . \tag{1.31}$$

Wie bei der ersten Diskretisierung führt (1.13) aufgrund der Trapezformel zu einer Näherung dritter Ordnung für (1.31). Man verfahre nun analog Abschnitt 1.2.3.

Eliminierung der Punkte $u_h(x_N)$ und $u_h(x_{N+1})$ führt durch geeignete Umformungen und Beachtung von $f(x_0) = f(x_N)$ und $f(x_{-1}) = f(x_{N-1})$ auf

$$\begin{bmatrix} 2 & -1 & & & -1 & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & -1 & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_h(x_{-1}) \\ u_h(x_0) \\ \vdots \\ \vdots \\ u_h(x_{N-2}) \\ u_h(x_{N-1}) \end{bmatrix} = \begin{bmatrix} h^2 f(x_{-1}) \\ h^2 f(x_0) \\ \vdots \\ \vdots \\ h^2 f(x_{N-2}) \\ h^2 f(x_{N-1}) \end{bmatrix} .$$

Die linksstehende Matrix sei mit $\widetilde{C}_h$ bezeichnet. Durch analoge Rechnung wie in 1.2.3 sieht man, daß sie folgende Eigenwerte und -vektoren besitzt:

für $-\frac{N}{2} < k \leq \frac{N}{2}$

$$Eigenvektor \quad (\exp(i2k\pi x_j))_{j=0,\ldots,N}$$

$$zum\ \ Eigenwert\ \ \lambda_k = 2 - 2cos(2k\pi h) .$$

Man erhält hier also dieselben Eigenwerte[4] und „dieselben“ Eigenvektoren wie für $\widetilde{A}_h$. Wie im Anhang A.1 bewiesen wird, gilt

$$dim(Kern(\widetilde{C}_h)) = 1 . \tag{1.32}$$

Insgesamt erhält man folgendes

Lemma 3 . *Das Gleichungssystem $C_h u_h = r_h$ ist genau dann lösbar, wenn die Kompatibilitätsbedingung* (1.31) *erfüllt ist. Dabei gilt für Lösungen u_h und $\tilde{u}_h$:*

$$u_h - \tilde{u}_h \in Kern(C_h) = span\left\{(1, 1, \ldots, 1)^T\right\} .$$

Faßt man die Ergebnisse der letzten drei Abschnitte zusammen, zeichnet sich ab, daß die erste Diskretisierungmöglichkeit die günstigste ist: Zwar liefern sowohl die erste als auch die dritte Diskretisierung diskrete Kompatibilitätsbedingungen (1.21) bzw. (1.31), die - im Gegensatz zur zweiten Diskretisierung - Näherungen zweiter Ordnung für die Kompatibilitätsbedingung (1.13) darstellen, jedoch ist die $N \times N$-Matrix $\widetilde{A}_h$ im Gegensatz zur $(N+1) \times (N+1)$-Matrix $\widetilde{C}_h$ symmetrisch und kleiner.

1.2.6 Numerische Ergebnisse

Um festzustellen, ob die Diskretisierungsordnung der vorgestellten drei Verfahren zur Konvergenz gleicher Ordnung führt, werden die jeweiligen Gleichungssysteme erst mit einem direkten und zusätzlich mit einem iterativen Verfahren gelöst. Dann wird jeweils der Fehler der berechneten Näherungslösung zur exakten (kontinuierlichen) Lösung der Aufgabe (1.10)-(1.12) berechnet und seine Entwicklung mit wachsendem N beobachtet.

[4] Allerdings hat hier z.B. für N=4, N=8 der Eigenwert 2 algebraische Vielfachheit 3, aber geometrische Vielfachheit 2.

Als direktes Lösungsverfahren soll der Gauß-Algorithmus verwendet werden. Dazu muß man aber vorher die Gleichungssysteme durch eine weitere Bedingung regulär machen. Es gilt offenbar:

$$A_h u_h = r_h$$

$$(1.33) \quad \Leftrightarrow \begin{bmatrix} 1 & & & & -1 & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} u_h(x_0) \\ u_h(x_1) \\ \vdots \\ \vdots \\ u_h(x_N) \\ u_h(x_{N+1}) \end{bmatrix} = \begin{bmatrix} 0 \\ h^2 f(x_1) \\ \vdots \\ \vdots \\ h^2 f(x_N) \\ a \end{bmatrix}$$

mit $a = h^2 \sum_{j=1}^{N} f(x_j)$.

$$B_h u_h = r_h$$

$$(1.34) \quad \Leftrightarrow \begin{bmatrix} 1 & & & & & -1 \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} u_h(x_0) \\ u_h(x_1) \\ \vdots \\ \vdots \\ u_h(x_{N-1}) \\ u_h(x_N) \end{bmatrix} = \begin{bmatrix} 0 \\ h^2 f(x_1) \\ \vdots \\ \vdots \\ h^2 f(x_{N-1}) \\ b \end{bmatrix}$$

mit $b = h^2 \sum_{j=1}^{N-1} f(x_j)$.

$$C_h u_h = r_h$$

$$(1.35) \quad \Leftrightarrow \begin{bmatrix} & -1 & & & 1 & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} u_h(x_{-1}) \\ u_h(x_0) \\ \vdots \\ \vdots \\ u_h(x_N) \\ u_h(x_{N+1}) \end{bmatrix} = \begin{bmatrix} 0 \\ h^2 f(x_0) \\ \vdots \\ \vdots \\ h^2 f(x_N) \\ c \end{bmatrix}$$

mit $c = h^2[f(x_0) + 2\sum_{j=1}^{N-1} f(x_j) + f(x_N)] = 2a$.

Nach Lemmata 1, 2 und 3 sind diese Gleichungssysteme jeweils genau dann lösbar, wenn a=0, b=0 bzw. c=0 gilt. Wie in den Abschnitten 1.2.3-1.2.5 bereits beschrieben wurde, ist dies durch eine Ersetzung der jeweiligen $f(x_i)$ durch $f(x_i) - \bar{f}$ erreichbar. Hier sollen aber auch die numerischen Konsequenzen getestet werden, wenn diese Art der Anpassung von f zur Erfüllung der diskreten Kompatibilitätsbedingungen nicht vorgenommen wird.

Setzt man in (1.33), (1.34) bzw. (1.35) für die letzte Variable jeweils einen Wert k fest, streicht ansonsten aber die letzte Zeile des jeweiligen

Gleichungssystems, sind die so entstehenden Gleichungssysteme eindeutig lösbar, weil die obigen Matrizen eindimensionale Kerne besitzen. Das Gauß-Verfahren kann dann eingesetzt werden.

Als Testbeispiel wurde die Funktion $f(t) = 4 \cdot 10^{10}\pi^2 \sin(2\pi(t+r))$ mit $r = 0.1$ benutzt[5]. Die exakte Lösung von (1.10)-(1.12) lautet dann:

$$u(t) = 10^{10} \sin(2\pi(t+r)) + s \quad \text{mit } s \in \mathbb{R}\,.$$

Da die exakte Lösung und die diskrete Näherungslösung (vor der Wahl von k) nur bis auf eine additive Konstante s eindeutig bestimmt sind, kann der Fehler F_N zwischen der exakten und der jeweils berechneten Lösung bestimmt werden, wenn man von u_h bzw. $u|_{\Omega_h}$ jeweils zum Beispiel den Durchschnitt $\overline{u_h} := \frac{1}{\sharp J}\sum_{i\in J} u_h(x_i)$ bzw. $\overline{u} := \frac{1}{\sharp J}\sum_{i\in J} u(x_i)$ abzieht und von den so erhaltenen Funktionen die Differenz zum Beispiel in der Maximumsnorm betrachtet:

$$F_N = \max_{i\in J}\{|(u_h(x_i) - \overline{u_h}) - (u(x_i) - \overline{u})|\}\,.$$

Für (1.33) ist $J = \{0,\ldots,N\}$, für (1.34) ist $J = \{0,\ldots,N-1\}$, und für (1.35)) ist $J = \{-1,\ldots,N\}$. Die Rate der Fehlerverkleinerung von $N = 2^{m-1}$ zu $N = 2^m$ ist dann definiert als

$$R_{2^m} := \frac{F_{2^m}}{F_{2^{m-1}}}\,.$$

Die sich daraus ergebenden Werte finden sich in den Tabellen 1.1, 1.2 und 1.3. Es ist jeweils angegeben, wie sich die Fehler ohne bzw. mit Ersetzung der $f(x_i)$ durch $f_h(x_i) = f(x_i) - \bar{f}$ verhalten. Dabei zeigt sich, daß diese Maßnahme für (1.33) bzw. (1.35) *keine Rolle* spielt: Die Fehler F_N unterscheiden sich jeweils nicht - auch wenn die Kompatibilitätsbedingungen (1.21) bzw. (1.31) (d.h. $a = 0$ bzw. $c = 0$) nicht erfüllt sind. Die für (1.33) bzw. (1.35) berechneten Raten R_N zeigen, daß sich die Fehler F_N (asymptotisch) um den Faktor 0.25 verkleinern. Folglich verhalten sich die erste und die dritte Diskretisierungsmöglichkeit auch hinsichtlich der Konvergenz wie Verfahren zweiter Ordnung.

Anders liegt der Fall bei der zweiten Diskretisierungsmöglichkeit. Hier ist natürlich die Kompatibilitätsbedingung ($b = 0$) nicht annähernd erfüllt (aber für kleiner werdendes h immer „besser"), weswegen sich nach Ersetzung der $f(x_i)$ durch $f(x_i) - \bar{f}$ größere Fehler und eine Fehlerverkleinerungsrate von (asymptotisch) 0.5 ergeben. Deutlich wirkt sich hier also aus, daß die analytische Bedingung $\int_0^1 f(x)\,dx = 0$ eben nicht zu einer Näherung für die diskrete Kompatibilitätsbedingung (1.30) führt. Die zweite Diskretisierungsmöglichkeit verhält sich dann entsprechend der Taylorentwicklung auch hinsichtlich der Konvergenz nur wie ein Verfahren erster Ordnung.

Ohne die $f(x_i)$ zu ersetzen, zeigen sich aber Fehler derselben Größenordnung wie bei der ersten und dritten Diskretisierungsmöglichkeit und ebenfalls eine Fehlerverkleinerungsrate von 0.25. Dies erklärt sich dadurch, daß

[5] Andere Testbeispiele liefern aber ein analoges Bild.

sich die Gleichungssysteme (1.33)-(1.35) abgesehen von der Kompatibilitätsbedingung im Prinzip nicht unterscheiden: (1.34) ist (außer der Kompatibilitätsbedingung) in (1.33) und (1.35) enthalten. Hat man also (1.34) *nach dem Streichen der letzten Zeile* gelöst, kann man die zusätzlichen Werte $u_h(x_{-1})$ bzw. $u_h(x_{N+1})$ in (1.33) und (1.35) daraus berechnen. In diesem Fall hat man aber die Gleichung, die durch die Diskretisierung erster Ordnung der Bedingung $u'(0) = u'(1)$ entsteht, sowie ihre Konsequenzen aus dem System (1.34) völlig entfernt.

Zur Lösung der drei Gleichungssysteme wird nun noch das iterative Gauß-Seidel-Verfahren herangezogen. Hier werden aber die folgenden singulären Systeme benutzt (vergleiche dazu Abschnitte 1.2.3 bis 1.2.5)[6]:

$$\widetilde{A}_h \begin{bmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-1}) \\ u_h(x_N) \end{bmatrix} = \begin{bmatrix} h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) \end{bmatrix}, \tag{1.36}$$

$$\widetilde{B}_h \begin{bmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-2}) \\ u_h(x_{N-1}) \end{bmatrix} = \begin{bmatrix} h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-2}) \\ h^2 f(x_{N-1}) \end{bmatrix}, \tag{1.37}$$

$$\widetilde{C}_h \begin{bmatrix} u_h(x_{-1}) \\ u_h(x_0) \\ \vdots \\ u_h(x_{N-2}) \\ u_h(x_{N-1}) \end{bmatrix} = \begin{bmatrix} h^2 f(x_{-1}) \\ h^2 f(x_0) \\ \vdots \\ h^2 f(x_{N-2}) \\ h^2 f(x_{N-1}) \end{bmatrix}. \tag{1.38}$$

Die Tabellen 1.4 und 1.5 zeigen die Ergebnisse. Die erste und dritte Diskretisierung liefern dabei dasselbe Bild wie schon bei der Lösung mit dem Gauß-Verfahren: Die asymptotische Fehlerverkleinerungsrate beträgt 0.25, und eine Ersetzung der $f(x_i)$ spielt keine Rolle. Auch wenn bei anderen Testbeispielen a und c noch größer sind und anders als bei der Lösung mit dem oben beschriebenen Gauß-Verfahren die Gleichungssysteme (1.36) und (1.38) ohne Ersetzung der $f(x_i)$ eigentlich keine Lösung besitzen, da nicht „streng“ $a = 0$ bzw. $c = 0$ gilt, ist eine „strenge Null“ nicht erforderlich.

Bei (1.37) dagegen zeigt sich sowohl mit als auch ohne Ersetzung der $f(x_i)$ deutlich das Verfahren erster Ordnung: Die Fehler F_N reduzieren sich nur um (etwa) den Faktor 0.5.

Diese Untersuchungen zeigen also numerisch eine Übereinstimmung der Konsistenz- und Konvergenzordnungen bei jedem der drei Verfahren. Ins-

[6] Zur Verwendung iterativer Verfahren zur Lösung singulärer Gleichungssysteme vergleiche [38].

gesamt ergibt sich, daß das erste Verfahren in der Form (1.36) den anderen vorzuziehen ist, da es einen lokalen und globalen Diskretisierungsfehler zweiter Ordnung aufweist und eine für numerische Zwecke günstige Matrix liefert (siehe auch Abschnitt 1.2.5). Durch Verwendung der Gleichungen $u_h(x_0) = u_h(x_N)$ und $u_h(x_1) = u_h(x_{N+1})$ wird letztlich auf einem eindimensionalen diskreten Torus gerechnet und damit der Periodizität von u besonders Rechnung getragen.

N	F_N	R_N	a
4	$2.165E{+}9$		$1.7E{-}5$
8	$5.584E{+}8$	0.2579	$3.3E{-}6$
16	$1.324E{+}8$	0.2371	$3.6E{-}7$
32	$3.274E{+}7$	0.2473	$2.7E{-}7$
64	$8.102E{+}6$	0.2475	$8.2E{-}8$
128	$2.017E{+}6$	0.2490	$1.3E{-}7$
256	$5.031E{+}5$	0.2494	$4.0E{-}8$
512	$1.256E{+}5$	0.2497	$2.9E{-}7$

Tabelle 1.1: Ergebnisse für (1.33) (Gauß-Verfahren). Die Ersetzung der $f(x_i)$ durch $f_h(x_i)$ liefert die gleichen Werte.

N	F_N	R_N	F_N	R_N	b
4	$1.891E{+}9$		$4.669E{+}9$		$-1.5E{+}10$
8	$5.238E{+}8$	0.2770	$2.408E{+}9$	0.5156	$-3.6E{+}9$
16	$1.279E{+}8$	0.2442	$1.208E{+}9$	0.5017	$-9.1E{+}8$
32	$3.216E{+}7$	0.2515	$6.043E{+}8$	0.5002	$-2.3E{+}8$
64	$8.030E{+}6$	0.2496	$3.021E{+}8$	0.5000	$-5.7E{+}7$
128	$2.008E{+}6$	0.2501	$1.511E{+}8$	0.5000	$-1.4E{+}7$
256	$5.020E{+}5$	0.2500	$7.554E{+}7$	0.5000	$-3.5E{+}6$
512	$1.255E{+}5$	0.2500	$3.777E{+}7$	0.5000	$-8.9E{+}5$

Tabelle 1.2: Ergebnisse für (1.34) (Gauß-Verfahren). Das (F_N, R_N)-Paar auf der linken Seite gibt die Werte ohne, das auf der rechten Seite mit Ersetzung der $f(x_i)$ durch $f_h(x_i)$ an.

N	F_N	R_N	c
4	$1.977E{+}9$		$1.5E{-}5$
8	$5.466E{+}8$	0.2765	$-1.7E{-}6$
16	$1.338E{+}8$	0.2448	$3.6E{-}7$
32	$3.312E{+}7$	0.2475	$6.0E{-}7$
64	$8.163E{+}6$	0.2465	$5.4E{-}7$
128	$2.026E{+}6$	0.2482	$-3.8E{-}7$
256	$5.042E{+}5$	0.2489	$-1.7E{-}8$
512	$1.258E{+}5$	0.2495	$5.2E{-}7$

Tabelle 1.3: Ergebnisse für (1.35) (Gauß-Verfahren). Die Ersetzung der $f(x_i)$ durch $f_h(x_i)$ liefert die gleichen Werte.

N	F_N	R_N	F_N	R_N
4	1.891E+9		2.269E+9	
8	5.238E+8	0.2770	5.330E+8	0.2349
16	1.279E+8	0.2442	1.297E+8	0.2433
32	3.216E+7	0.2514	3.257E+7	0.2511
64	8.030E+6	0.2497	8.092E+6	0.2484
128	2.008E+6	0.2501	2.017E+6	0.2493
256	5.020E+5	0.2500	5.031E+5	0.2494
512	1.227E+5	0.2444	1.228E+5	0.2441

Tabelle 1.4: Ergebnisse für (1.36) (links) und (1.38) (rechts) mit dem Gauß-Seidel-Verfahren. Die Ersetzung der $f(x_i)$ durch $f_h(x_i)$ liefert jeweils die gleichen Werte.

N	F_N	R_N	F_N	R_N
4	4.575E+9		2.239E+9	
8	1.747E+9	0.3819	1.333E+9	0.5954
16	9.538E+8	0.5460	8.812E+8	0.6611
32	5.300E+8	0.5557	5.159E+8	0.5855
64	2.822E+8	0.5325	2.793E+8	0.5414
128	1.459E+8	0.5170	1.453E+8	0.5202
256	7.423E+7	0.5088	7.407E+7	0.5098
512	3.744E+7	0.5044	3.740E+7	0.5049

Tabelle 1.5: Ergebnisse für (1.37) mit dem Gauß-Seidel-Verfahren. Das (F_N, R_N)-Paar auf der linken Seite gibt die Werte ohne, das auf der rechten Seite mit Ersetzung der $f(x_i)$ durch $f_h(x_i)$ an.

Kapitel 2

Zweidimensionales Modellproblem

2.1 Verschiedene Mehrgitter-Verfahren

2.1.1 Diskretisierung

Das Ergebnis des letzten Abschnitts wird nun auf das zweidimensionale Modellproblem (1.24), (1.25), (1.26) mit $a_1 = a_2 = 1$ übertragen. Es wird also **analog dem obigen 1. Verfahren** diskretisiert, wobei sich die folgenden Gleichungen ergeben:

$$\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h u_h = h^2 f_h \quad \text{in } \Omega_h \tag{2.1}$$

und in $\partial\Omega_h$

$$\begin{aligned} u_h(x_0, y_j) = u_h(x_N, y_j) \quad &, \quad u_h(x_1, y_j) = u_h(x_{N+1}, y_j)\,, \\ u_h(x_j, y_0) = u_h(x_j, y_N) \quad &, \quad u_h(x_j, y_1) = u_h(x_j, y_{N+1})\,. \end{aligned} \tag{2.2}$$

Hierbei bezeichnen

$$\begin{aligned} G_h &= \{(x_j, y_k) \mid j, k \in \mathbb{Z}\}\,, \\ \Omega_h &= G_h \cap]0, 1+h[^2 = G_h \cap [h, 1]^2\,, \\ \partial\Omega_h &= G_h \cap \partial([0, 1+h]^2)\,, \end{aligned}$$

$$x_j = jh,\ y_j = jh \text{ und } h = \frac{1}{N} \text{ mit } N = 2^p,\ p \in \mathbb{N}\,.$$

Das Gleichungssystem (2.1)-(2.2) ist entsprechend den Überlegungen im Abschnitt 1.2.3 (vergleiche Lemma 1) genau dann bis auf eine Konstante eindeutig lösbar, wenn die diskrete Kompatibilitätsbedingung erfüllt ist:

$$h^2 \sum_{j,k=1}^{N} f_h(x_j, y_k) = 0\,. \tag{2.3}$$

Dieser Bedingung soll ein gegebenes f_h hier genügen (siehe auch Abschnitt 2.1.3).

Mit Hilfe von (2.2) werden nun sämtliche Randpunkte, d.h. Punkte aus $\partial\Omega_h$, aus (2.1) eliminiert. Das bedeutet konkret: Immer wenn eine Gleichung in (2.1) auf einen Punkt aus $\partial\Omega_h$ zugreift, wird stattdessen gemäß (2.2) der entsprechende Punkt aus Ω_h verwendet. In Operatorschreibweise lautet das neue System dann

$$L_h u_h = h^2 f_h \quad \text{in } \Omega_h \,, \tag{2.4}$$

wobei in $\{(x_j, y_k) \mid j,k = 2, ..., N-1\}$ gilt:

$$L_h = \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h .$$

Für die restlichen Punkte aus Ω_h muß L_h entsprechend obigem abgeändert werden. Damit entspricht das Gebiet, auf dem man rechnet, einem diskreten Torus.

2.1.2 Mehrgitter-Komponenten

Sei nun N fest vorgegeben. (2.4) soll mit Hilfe eines Mehrgitter-Verfahrens gelöst werden. Die Prinzipien und Funktionsweisen solcher Methoden werden hier nicht dargestellt. Man vergleiche dazu etwa [60, 58, 30]. Die im folgenden auftretenden Bezeichnungen werden ebenfalls dort erklärt.

Für das vorliegende lineare Problem werden Correction-Scheme- (CS) bzw. Full-Multigrid-Methoden (FMG) benutzt. Dazu wird eine Folge von Gittern $\Omega_1, \Omega_2, ..., \Omega_m$ mit $m \leq p$ betrachtet, die hier die folgende Gestalt haben:

$$\begin{aligned} G_j &= \{(kh_j, lh_j) \mid k,l \in \mathbb{Z}\} \,, \\ \Omega_j &= \Omega_{h_j} = G_j \cap [h_j, 1]^2 \,, \\ \partial\Omega_j &= \partial\Omega_{h_j} \,, \\ N_1 = N, h_1 = h,\ h_j = 2h_{j-1},\ N_j &= \frac{1}{h_j} \text{ für } j \in \{2, ..., m\} \,. \end{aligned}$$

Ω_1 ist hier das feinste und Ω_m das gröbste Gitter. Auch auf den gröberen Gittern sollen die Operatoren L_{h_j} (siehe (2.4)) verwendet werden. Statt L_{h_j}, f_{h_j}, u_{h_j} wird meistens kürzer L_j, f_j, u_j geschrieben. Gitterfunktionen u_j auf dem jeweiligen Gitter Ω_j haben die folgende Gestalt:

$$u_j : \Omega_j \mapsto \mathbb{R} \,.$$

Für den Raum der Gitterfunktionen auf dem Gitter Ω_j wird dann das folgende (skalierte Standard-)Skalarprodukt eingeführt (um z.B. Adjungierte berechnen zu können):

$$< v_j, w_j > := h^2 \sum_{p \in \Omega_j} v_j(p)\, w_j(p) \,. \tag{2.5}$$

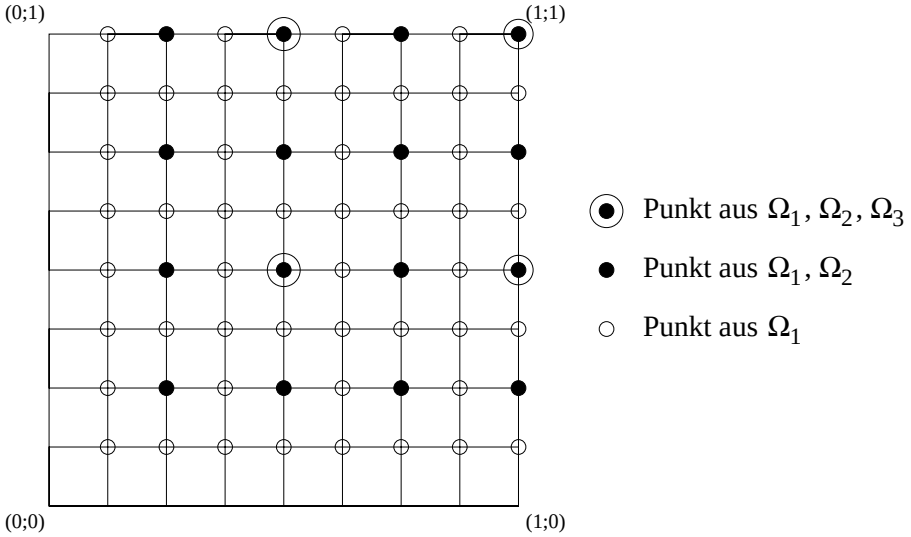


Abbildung 2.1: Das Einheitsquadrat, überzogen mit einer Folge dreier Gitter: Ω_1 mit $h = \frac{1}{8}$ und die zwei Vergröberungen Ω_2, Ω_3.

Folgende Komponenten charakterisieren hier die *CS-Methoden*:

- *Art des Mehrgitter-Cycles*

  Hier wird der V-Cycle ($\gamma = 1$) oder der W-Cycle ($\gamma = 2$) benutzt (siehe auch Abbildung 2.2).

- *Relaxation*

  Es werden ν_1 Relaxationsschritte vor und ν_2 Relaxationsschritte nach der Grobgitterkorrektur durchgeführt, wobei hier als Relaxationsverfahren das Gauß-Seidel-Verfahren mit schachbrettartiger (red-black) (GS-RB) oder lexikographischer (GS-LEX) Numerierung und ein überrelaxiertes GS-RB-Verfahren mit dem (Über-)Relaxationsparameter ω benutzt werden. Für das im Sinne der Glättungseigenschaften optimale ω_{opt} gelten die folgenden Aussagen (siehe [65]), falls nur ein Relaxationsschritt zwischen aufeinanderfolgenden Grobgitterkorrektur-Schritten stattfindet:

  $$1 \leq \omega_{opt} < 2 \,,$$
  $$\omega_{opt} < \omega_{ub} = \frac{2}{1 + \sqrt{1 - C_{max}^2}} \,,$$

  wobei $C_{max} = 1 - c_{min}$ mit $c_{min} = \frac{1}{2}$ für den hier vorliegenden zweidimensionalen Laplace-Operator ist. Ein etwas kleinerer Wert als die obere Grenze $\omega_{ub} = 1.0718$ erweist sich allerdings bereits als sehr gute Arbeitsgröße (auch für mehrere Relaxationsschritte), so daß hier

  $$\omega = 1.07$$

  verwendet wird.

Zur Definition von Glättungsfaktoren und zur Analyse der Glättungseigenschaften des Gauß-Seidel-Verfahrens vergleiche man [58][1].

Der Relaxationsoperator auf dem Gitter Ω_j wird mit S_j bezeichnet.

- *Restriktion*

  Als Restriktion zum nächstgröberen Gitter wird hier Injection, Half Weighting (HW) oder Full Weighting (FW) benutzt. Für ein Verfahren, das GS-RB als Glättungsverfahren verwendet, erweist sich Injection als nicht geeignet (vergleiche z.B. [30]). Daher werden in diesem Fall Full Weighting oder Half Weighting eingesetzt. Letzteres bedeutet hier Half Injection, da der Restriktion dann (mindestens) ein GS-RB-Schritt vorausgeht. Der HW-Operator liefert nämlich einen gewichteten Mittelwert der Defekte eines (roten) Grobgitterpunktes p_{red} und seiner vier direkten Nachbarn, die alle schwarze Punkte im Sinne der Red-Black-Numerierung darstellen. Nach einem GS-RB-Schritt verschwinden aber die Defekte in den schwarzen Punkten, so daß die Anwendung von HW auf p_{red} den Wert $\frac{1}{2}p_{red}$ ergibt (vergleiche [58]).

  Für GS-LEX kommen Injection oder Full Weighting zum Einsatz, für ω-GS-RB nur Full Weighting.

  Der Operator für die Restriktion von Daten vom Gitter Ω_j zum Gitter Ω_{j+1} sei jeweils mit R_j^{j+1} bezeichnet.

  Das auf dem Gitter Ω_{j+1} $(1 \leq j \leq m-1)$ zu lösende Gleichungssystem lautet dann:

  $$L_{j+1} u_{j+1} = f_{j+1} \tag{2.6}$$

  mit

  $$f_{j+1} := 4R_j^{j+1}(f_j - L_j S_j^{\nu_1} w_j)\ , \tag{2.7}$$

  wobei w_j eine „Start"approximation[2] für u_j und f_{j+1} den auf das gröbere Gitter $j+1$ restringierten Defekt darstellt.

  Die Lösbarkeit des Gleichungssystems garantiert der noch folgende Satz 3, falls Full Weighting als Restriktion verwendet wird (siehe zu Half Weighting und Injection ebenfalls Abschnitt 2.1.3).

- *Exakte Lösung auf dem gröbsten Gitter*

  Hier wird auf dem Gitter Ω_m exakt gelöst. Dabei muß man beachten, daß A_m singulär und $Kern(A_m)$ eindimensional ist (vergleiche Abschnitt 1.2.3), wobei A_m die Matrix des entsprechenden Gleichungssystems bezeichne. Die letzte Zeile ist eine Linearkombination der übrigen Zeilen[3], so daß man für die letzte Variable einen Wert festsetzt

[1] Für Aussagen über das hier betrachtete Modellproblem mit periodischen Randbedingungen vergleiche dort insbesondere Kap. 9 und die Bemerkungen in Abschnitt 7.6.

[2] Das kann im W-Cycle auch eine (neue) Näherung für u_j sein, die durch ν_2 Relaxationsschritte nach einer Grobgitterkorrektur erhalten wurde.

[3] Die Spaltensumme (und die Zeilensumme) ist immer Null.

(bzw. den bereits vorhandenen Näherungswert beibehält) und dann mit dem Gauß-Verfahren nur die ersten $N^2 - 1$ Zeilen behandelt.

In den numerischen Tests (siehe Abschnitt 2.2) werden durchgängig alle zur Verfügung stehenden gröberen Gitter verwendet, d.h. es wird $m = p$ festgesetzt.

Bei FW als Restriktion ist die Lösbarkeit des singulären Gleichungssystems auf dem gröbsten Gitter garantiert, wie Satz 3 zeigt. Setzt man allerdings, wie oben beschrieben, für die letzte Variable einen Wert fest, ist das *übrige* Gleichungssystem immer lösbar - auch wenn die letzte Gleichung, die ja nach Umformung die Gestalt $0 \cdot u_h(x_N, y_N) = c$ hat, nicht lösbar ist (für $c \neq 0$).

- *Prolongation*

  Hier wird die auf einem groben Gitter Ω_{j+1} errechnete Korrektur durch bilineare Interpolation auf das nächstfeinere Gitter j transferiert und zur dortigen Näherung addiert. Der Operator für diese Interpolation von Daten vom Gitter Ω_{j+1} zum Gitter Ω_j wird mit P_{j+1}^j bezeichnet.

  Allgemein gilt: Sei m_d die Diskretisierungsordnung des Operators L_j, m_p die Ordnung des Prolongationsoperators[4] P_{j+1}^j und m_r die Ordnung des Restriktionsoperators[5] R_j^{j+1}. Dann sollten die Ordnungen m_p und m_r der Transferoperatoren die folgende Ungleichung („Faustregel“) erfüllen, die durch lokale Fourier-Analyse erhalten werden kann (siehe [30, 34]):

  $$m_r + m_p > m_d \,. \tag{2.8}$$

  L_j besitzt hier offenbar die Diskretisierungsordnung $m_d = 2$ und der bilineare Interpolationsoperator P_{j+1}^j die Ordnung $m_p = 2$. Da dieser Operator aber die Adjungierte des FW-Operators darstellt:

  $$(R_j^{j+1})^* = P_{j+1}^j \,,$$

  wenn man das Skalarprodukt (2.5) verwendet, gilt für FW $m_r = 2$. Die Kombination von FW mit bilinearer Interpolation erfüllt also die Faustregel (2.8). Sowohl für Injection als auch für Half Weighting gilt jedoch $m_r = 0$, so daß (2.8) gerade nicht erfüllt ist. Trotzdem werden Injection (bei GS-LEX) und Half Weighting (bei GS-RB) getestet, weil sie in Kombination mit bilinearer Interpolation sowohl bei Dirichlet- als auch Neumannschen Randbedingungen erfolgreich eingesetzt werden.

- *Behandlung der „Randpunkte“*

  Die Vorgehensweise, die auch schon zu (2.4) geführt hat, wird konsequent angewandt: Immer wenn ein L_j, S_j, R_j^{j+1} bzw. P_j^{j-1} auf einen

[4] Die Ordnung der Interpolation ist $k+1$ mit maximalem k, so daß P_{j+1}^j alle Polynome des Grades $\leq k$ invariant läßt.

[5] Die Ordnung der Restriktion ist $k+1$ mit maximalem k, so daß $s(R_j^{j+1})^*$ mit geeignetem Faktor s alle Polynome des Grades $\leq k$ invariant läßt.

Punkt aus $\partial\Omega_j$ zugreift, wird die (geforderte) Periodizität der Lösung ausgenutzt und der entsprechende Punkt aus Ω_j verwendet. Also rechnet man letztlich immer auf einem Torus.

Die Tabelle (2.1) zeigt die getesteten Kombinationsmöglichkeiten.

Verfahren	ILEX	HRED	FLEX	FRED	ω-FRED
Glättungsverfahren	GS-LEX	GS-RB	GS-LEX	GS-RB	ω-GS-RB
Restriktion	Injection	HW	FW	FW	FW

Tabelle 2.1: Getestete Kombinationen. Es sind nur die Komponenten aufgeführt, in denen sich die Programme unterscheiden.

Bemerkung: Offensichtlich können die hier vorgestellten Verfahren dank ihrer Struktur prinzipiell wie entsprechende Algorithmen für Dirichlet- oder Neumannsche Randbedingungen parallelisiert werden (vergleiche z.B. [60]).

Folgendermaßen sind hier die *FMG-Methoden* gekennzeichnet:

Auf dem Gitter Ω_j werden, ausgehend von Ω_m bis hin zu Ω_1, r Mehrgitter-Iterationen mit einem der oben beschriebenen CS-Verfahren durchgeführt, wofür die Gitter $\Omega_j, \ldots, \Omega_m$ benutzt werden. Für den Datentransport vom Gitter Ω_j zum nächstfeineren Gitter Ω_{j-1} sollte eine Interpolation verwendet werden, die eine höhere Ordnung κ_{FMG} als die Diskretisierungsordnung m_d von L_h aufweist (vergleiche [58]). Für $m_d = 2$ (wie im vorliegenden Fall) wird oft die kubische Interpolation eingesetzt. Für das vorliegende Modellproblem kann allerdings auch eine spezielle, billigere Interpolation 4. Ordnung benutzt werden (vergleiche [58]). Bezeichne $u_{j-1}^{(0)}(x,y)$ den zu berechnenden Startwert auf dem Gitter Ω_{j-1} und $\tilde{u}_j(x,y)$ die berechnete Approximation für $u_j(x,y)$ auf dem Gitter Ω_j. Die Interpolation läuft dann in drei Schritten ab:

1. Für Punkte $(x,y) \in \Omega_{j-1} \cap \Omega_j$ definiert man

$$u_{j-1}^{(0)}(x,y) = \tilde{u}_j(x,y)\,. \tag{2.9}$$

2. Für Punkte $(x,y) \in \Omega_{j-1} \setminus \Omega_j$ mit $x = l_1 h_{j-1}$, $y = l_2 h_{j-1}$ und geradem $l_1 + l_2$ wird dann definiert:

$$\frac{1}{2h_{j-1}^2} \begin{bmatrix} -1 & & -1 \\ & 4 & \\ -1 & & -1 \end{bmatrix} u_{j-1}^{(0)}(x,y) = f_{j-1}(x,y)\,. \tag{2.10}$$

3. Für Punkte $(x,y) \in \Omega_{j-1}$ mit $x = l_1 h_{j-1}$, $y = l_2 h_{j-1}$ und ungeradem $l_1 + l_2$ definiert man nun

$$\frac{1}{h_{j-1}^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} u_{j-1}^{(0)}(x,y) = f_{j-1}(x,y)\,. \tag{2.11}$$

Dies kann offenbar als ein Halbschritt von GS-RB interpretiert werden.

Die Struktur eines FMG-Verfahrens ist in Abbildung 2.2 dargestellt. Weitere Eigenschaften, die ein FMG-Verfahren besitzen soll, finden sich in Abschnitt 2.2.2.

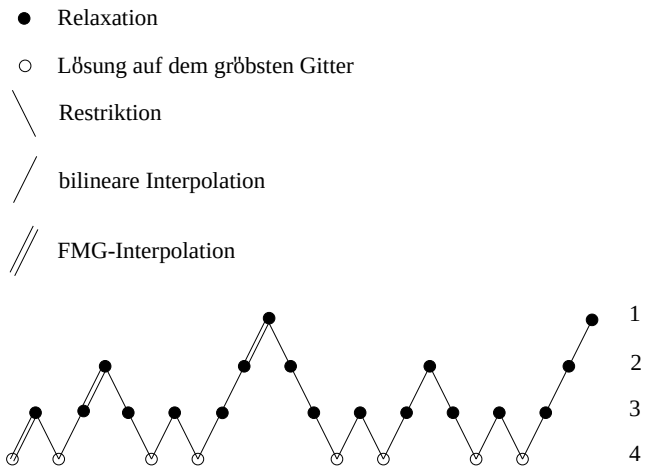


Abbildung 2.2: FMG mit $m = 4$, $r = 1$ und $\gamma = 2$.

Weitere Bezeichnungen

NAME-X(ν_1,ν_2) bezeichnet im weiteren das Verfahren NAME in der CS-Version im X-Cycle mit ν_1 Relaxationsschritten vor und ν_2 Relaxationsschritten nach der Grobgitterkorrektur, z.B. HRED-W(2,1).

FMG(NAME-X(ν_1,ν_2),r) oder kürzer FMG(X(ν_1,ν_2),r) steht dann für die FMG-Version des Verfahrens NAME-X(ν_1,ν_2) mit dem oben beschriebenen r.

2.1.3 Lösbarkeit der Gleichungssysteme

Um die Lösbarkeit von (2.4) zu gewährleisten, muß (2.3) erfüllt sein. Analog dem eindimensionalen Modellproblem gilt jedoch aufgrund der (zweidimensionalen) Kompatibilitätsbedingungen (vergleiche Satz 1) und der Trapezformel (falls zusätzlich $f \in C^2([0,1]^2)$ ist):

$$h^2 \sum_{i,j=1}^{N} f(x_i, y_j) = \int_{[0,1]^2} f(r)\, dr + O(h^2) = O(h^2)\ .$$

Verwendet man also $f|_{\Omega_h}$ als f_h, führt $\int_{[0,1]^2} f(r)\, dr = 0$ zu einer Näherung *zweiter* Ordnung für (2.3) - im Eindimensionalen ergab sich aus (1.13) eine Näherung *dritter* Ordnung für (1.21).

Wie in (1.22) wird nun definiert:

$$f_h(x_k, y_l) := f(x_k, y_l) - \bar{f}\ , \tag{2.12}$$

wobei $\bar{f}$ den Durchschnitt der Werte $f(x_i, y_j)$ angibt:

$$\bar{f} := \frac{1}{N^2} \sum_{i,j=1}^{N} f(x_i, y_j) = h^2 \sum_{i,j=1}^{N} f(x_i, y_j) \; .$$

Somit erfüllt die neue Funktion f_h die diskrete Kompatibilitätsbedingung (2.3) und stellt eine Näherung zweiter Ordnung für $f|_{\Omega_h}$ dar (wie im eindimensionalen Fall). Also liegt nach wie vor ein Verfahren zweiter Ordnung bezüglich der Konsistenz vor.

Die Gültigkeit von (2.3) reicht übrigens bereits aus, um die Lösbarkeit der (singulären) Gleichungssysteme auf allen betrachteten Gittern zu folgern, falls FW als Restriktion verwendet wird. Dies sieht man mit Hilfe des folgenden Satzes:

Satz 3 . *Sei $d_1 := f_1$ die rechte Seite von (2.4), v_j für $j \in \{1, ..., m\}$ je* **irgendeine** *Funktion auf Ω_j, R_j^{j+1} der FW-Operator und $< \cdot, \cdot >$ ein skaliertes Standardskalarprodukt (wie z.B. (2.5)). Definiert man nun für $j \in \{2, ..., m\}$*

$$d_j := 4R_{j-1}^{j}(d_{j-1} - L_{j-1}v_{j-1})$$

und für $j \in \{1, ..., m\}$

$$\mathbf{1}_j \equiv 1 \quad \textit{auf} \quad \Omega_j \; ,$$

dann gilt folgendes:

Falls $< d_1, \mathbf{1}_1 >= 0$ ist, gilt sogar

$$\forall \, j \in \{1, ..., m\} \; < d_j, \mathbf{1}_j >= 0 \; . \tag{2.13}$$

Beweis[6] durch vollständige Induktion über j:

Für $j = 1$ ist nichts zu zeigen. Sei daher nun die Behauptung für ein $j \geq 1$ gültig. Dann ist

$$\begin{aligned} < d_j - L_j v_j, \mathbf{1}_j > &=< d_j, \mathbf{1}_j > - < L_j v_j, \mathbf{1}_j > \\ &= 0 - < v_j, L_j^* \mathbf{1}_j > \; . \end{aligned}$$

Dabei bezeichne L_j^* die Adjungierte von L_j. Offenbar ist

$$L_j^* \mathbf{1}_j = L_j \mathbf{1}_j = 0 \; .$$

Dann gilt also

$$< d_j - L_j v_j, \mathbf{1}_j >= 0 \; .$$

Außerdem ist

$$\begin{aligned} < d_{j+1}, \mathbf{1}_{j+1} > &= 4 < R_j^{j+1}(d_j - L_j v_j), \mathbf{1}_{j+1} > \\ &= 4 < d_j - L_j v_j, (R_j^{j+1})^* \mathbf{1}_{j+1} > \; . \end{aligned}$$

[6] Die Beweisidee findet sich in [63].

Durch Nachrechnen sieht man, daß für Full Weighting

$$(R_j^{j+1})^* \mathbf{1}_{j+1} = \sigma P_{j+1}^j \mathbf{1}_{j+1} = \sigma \mathbf{1}_j \tag{2.14}$$

mit $\sigma = 1$ gilt, wenn man das Skalarprodukt (2.5) verwendet. Nun folgt insgesamt

$$\begin{aligned} < d_{j+1}, \mathbf{1}_{j+1} > &= 4\sigma < d_j - L_j v_j, \mathbf{1}_j > \\ &= 0 \, , \end{aligned}$$

womit die Behauptung also bewiesen ist. □

Also sind - bei Verwendung von FW als Restriktion - mit $L_1 u_1 = f_1$ auch alle $L_j u_j = f_j$ lösbar, da wegen (2.13) jeweils die Kompatibilitätsbedingung erfüllt ist und v_j frei gewählt werden kann (etwa $v_j = S_j^{\nu_1} w_j$, siehe auch Abschnitt 2.1.2, Restriktion). Man sieht, daß für die *Lösbarkeit* der Gleichungssysteme die Art der Relaxationsschritte ebensowenig eine Rolle spielen wie die Wahl des Cycles (d.h. die Wahl von γ).

Bei der Verwendung von Half Weighting oder Injection ergeben sich Probleme, weil in diesen Fällen (2.14) nicht erfüllt ist: weder Half Weighting noch Injection stellen Restriktionsoperatoren mindestens erster Ordung ($m_r \geq 1$) dar, da für sie $m_r = 0$ gilt (vergleiche Abschnitt 2.1.2, Prolongation). Und tatsächlich ergeben numerische Tests, daß die Gleichungssysteme auf den gröberen Gittern dann oft formal nicht lösbar sind (siehe Abschnitte 2.2 und 3.2).

Eine Möglichkeit wäre natürlich, dies zu erzwingen - wie schon auf dem feinsten Gitter durch die Definition von $f_h (= f_1)$ - und statt des entsprechenden f_j folgendes $\tilde{f}_j$ zu verwenden:

$$\tilde{f}_j(x_k, y_l) := f_j(x_k, y_l) - \bar{f}_j \tag{2.15}$$

mit

$$\bar{f}_j := \frac{1}{N^2} \sum_{k,l=1}^{N} f_j(x_k, y_l) \, .$$

Aber anders als auf dem feinsten Gitter, wo f_h eine Näherung zweiter Ordnung für $f|_{\Omega_h}$ darstellt, ist auf den gröberen Gittern i.a. $\tilde{f}_j$ keine Näherung für f_j, so daß diese Maßnahme das zu lösende System stark verändert. Für numerische Ergebnisse vergleiche man Abschnitt 2.2.

2.1.4 Anpassung

Bei Problemen, die nur bis auf eine additive Konstante eindeutig bestimmt sind[7], kann es passieren, daß durch die Relaxationsschritte oder durch die direkte Lösung auf dem gröbsten Gitter Vektoren mit betraglich sehr großen Einträgen entstehen.

Numerische Fehler haben dann eventuell einen großen Einfluß auf die Berechnung z.B. der Residuen, die dann möglicherweise auf den gröberen

[7] z.B. auch die Poissongleichung mit Neumannschen Randbedingungen, vergleiche [9].

Gittern nicht mehr vernünftig zu sehen wären. Und selbst bei Full Weighting könnte es dann passieren, daß $< d_j, \mathbf{1}_j >= 0$ (d.h. die Lösbarkeit der Gleichungssysteme, vergleiche Satz 3) nicht mehr erfüllt wäre.

Um dies zu verhindern, kann es nützlich sein, das Verfahren um eine optionale „Anpassung“ zu erweitern. Die **Anpassung** $anp(v)$ **eines Vektors** $v = (v_i)_{i=1,\ldots,N_j}$ sei wie folgt definiert:

$$anp(v) = (v_i - \bar{v})_{i=1,\ldots,N_j} , \tag{2.16}$$

wobei $\bar{v}$ der Mittelwert der Komponenten von v sei, also:

$$\bar{v} = \frac{\sum_{i=1}^{N_j} v_i}{N_j} .$$

Auf diese Weise liegen die Komponenten von $anp(v)$ um Null. Mit dieser Maßnahme ändert man an der Lösung des jeweiligen Gleichungssystems nichts, da mit v auch $anp(v)$ in der Lösungsmenge

$$\{(v_i + c)_{i=1,\ldots,N_j} \mid c \in \mathbb{R}\}$$

liegt. Formal wird v durch $anp(v)$ genauso ersetzt wie schon $f|_{\Omega_h}$ durch f_h (siehe (2.12)) oder f_j durch $\tilde{f}_j$ (siehe (2.15)) (falls man diese Maßnahme einsetzt). Der Unterschied liegt allerdings darin, daß die Anpassung einen anderen Repräsentanten der jeweiligen Lösungsmenge auswählt, (2.12) und (2.15) dagegen die Gleichungssysteme (mehr oder weniger stark) verändern.

Zwei Varianten dieser Anpassung werden hier verwendet:

- die *komplette Anpassung*:

  In den Verfahren wird jeweils nach der Relaxation (vor und nach der Grobgitterkorrektur) und nach der Lösung auf dem gröbsten Gitter der erhaltene Vektor v nach obigem Schema angepaßt.

- die *grobe Anpassung*:

  Hier wird lediglich auf dem gröbsten Gitter die berechnete exakte Lösung angepaßt.

Wie im Abschnitt 2.2.3 noch gezeigt wird, ist eine (komplette) Anpassung in vielen Fällen unnötig, vor allem, wenn FMG verwendet wird. Bei sehr großen Startresiduen und für die Berechnung von asymptotischen Konvergenzraten ist sie allerdings (für CS) unbedingt erforderlich!

2.2 Numerische Ergebnisse

Die fünf FORTRAN-Programme wurden anhand mehrerer Beispiele getestet. Dabei machte es sich beim Verfahren HRED nicht bemerkbar, daß die Lösbarkeit der Gleichungssysteme auf den gröberen Gittern nicht gesichert ist: Die diskrete Kompatibilitätsbedingung (2.3) ist hier auch auf den gröberen Gittern trotzdem erfüllt, so daß keine Schwierigkeiten auftreten.

Bei ILEX liegt der Fall anders, da die Bedingung (2.3) auf den gröberen Gittern nicht annähernd erfüllt ist, wenn auch die Werte der Summe in (2.3) von Cycle zu Cycle kleiner werden.

Es erweist sich bei nicht erfüllter Kompatibilitätsbedingung auf den **gröberen** Gittern als **nicht hilfreich**, f_j durch $\tilde{f}_j$ (siehe (2.15)) zu ersetzen: Die Raten verbessern sich in keinem Fall. Diese Änderung sorgt zwar für eine Lösbarkeit des entsprechenden neuen Gleichungssystems, verändert die Ausgangsaufgabe aber so stark, daß sie nicht mehr „erkennbar" ist.

Außerdem ist kein Unterschied zu bemerken, wenn man statt f_h (siehe (2.12)) doch $f|_{\Omega_h}$ verwendet: Die „exakte" Null in der diskreten Kompatibilitätsbedingung ist nicht erforderlich. Die Programme vertragen auch Größenordnungen von etwa 10^{-7} (teilweise auch größer), ohne daß sich die Raten ändern. Sicherheitshalber sollte trotzdem f_h verwendet werden.

Die Ergebnisse der noch folgenden Abschnitte zeigen dann eindeutig, daß die Red-Black-Verfahren FRED und HRED und im W-Cycle auch ω-FRED den lexikographischen Verfahren FLEX und ILEX vorzuziehen sind. Im V-Cycle jedoch erweist sich ω-FRED sogar als langsamer als die anderen mit Ausnahme von ILEX.

Da für einen Mehrgitter-Cycle bei HRED der Rechenaufwand dank Half Injection geringer als bei FRED ist, das wiederum schneller als ω-FRED einen Cycle durchläuft, ergibt sich letztlich, daß im X(2,1)-Cycle HRED und im X(1,1)-Cycle FRED vorzuziehen ist. $\omega-$FRED lohnt sich für die *zweidimensionale* Poissongleichung nicht.

Die erhaltenen Werte wurden mit den Konvergenzraten analoger Verfahren für Dirichlet-Randbedingungen bzw. Werten aus der rigorosen und lokalen Fourier-Analyse verglichen, HRED-W(2,1) zudem mit einem Galerkin-Verfahren von Hackbusch [29], das ebenfalls für periodische Randbedingungen gedacht ist. Es ergab sich hierbei folgendes:

- Die aus der rigorosen Fourier-Analyse stammenden Werte stimmen mit den entsprechenden asymptotischen Konvergenzraten für FRED bzw. HRED überein. Der ω-FRED-Wert ist etwas schlechter.

- Werte für FLEX entsprechen den aus der lokalen Fourier-Analyse stammenden ρ_{loc}. Bei ILEX zeigen sich große Abweichungen, wie es dank der Probleme auf den gröberen Gittern auch zu erwarten ist.

- HRED und das Galerkin-Verfahren zeigen sehr ähnliche Konvergenzraten von 0.033 bzw. 0.032.

- Die entsprechenden Verfahren für Dirichlet-Randbedingungen liefern demgegenüber etwa dieselben oder aber etwas schlechtere Konvergenzraten.

Insgesamt bestätigt dies auch für (ω-)FRED, HRED und FLEX eine Beobachtung von Hackbusch in [30], daß *die Wahl anderer* (d.h. hier periodischer) *Randbedingungen die Konvergenz der Multigrid-Verfahren nicht verschlechtert*[8].

[8] In [30] wurde dies für das Galerkin-Verfahren [29] beobachtet.

Desweiteren wurden die FMG-Verfahren getestet sowie Untersuchungen zur Wirkung der Anpassung durchgeführt. Die einzelnen Ergebnisse finden sich in den nächsten Abschnitten.

2.2.1 Konvergenzordnung und -raten

Ergebnisse werden hier für zwei ausgewählte Beispiele angegeben. Für andere Funktionen[9] zeigt sich aber das gleiche Bild.

Beispiel (1)

$$f(x,y) = 8\pi^2 \sin(2\pi x)\sin(2\pi y)\,. \tag{2.17}$$

Dieses Problem besitzt als exakte Lösung die Funktionenschar

$$u(x,y) = \sin(2\pi x)\sin(2\pi y) + c \quad \text{mit } c \in \mathbb{R}\,. \tag{2.18}$$

Für dieses Beispiel wird $N = 64$ gewählt. Als Startwerte werden $u_h^{(0)}(x_i, y_j) = i + j$ verwendet.

Beispiel (2)

$$f(x,y) = 16\pi^2 \sin(2\pi x + 1) + 60\pi^2 \cos(2\pi(2x + y))\,. \tag{2.19}$$

Die exakte Lösung dieses Problems ist die Funktionenschar

$$u(x,y) = 4\sin(2\pi x + 1) + 3\cos(2\pi(2x + y)) + c \quad \text{mit } c \in \mathbb{R}\,. \tag{2.20}$$

Für dieses Beispiel werden, wenn nicht anders angegeben, $N = 256$ sowie als Startwerte $u_h^{(0)}(x_i, y_j) = i + j$ benutzt.

Definitionen

Im folgenden sei $u_h^{(j)}$ die nach j Mehrgitter-Iterationen berechnete Näherung für die diskrete Lösung u_h. Mit $\tilde{u}_h$ ist dann der Grenzwert dieses Iterationsverfahrens gemeint. Mit r_j wird die Maximumsnorm des Residuums nach der j-ten Mehrgitter-Iteration bezeichnet:

$$r_j = \left\| f_h - \frac{1}{h^2} L_1 u_h^{(j)} \right\|_\infty .$$

Mit „Anfangsresiduum“ ist r_0 gemeint, mit „Endresiduum“ das für das jeweilige Verfahren und Beispiel kleinstmögliche numerisch erhältliche r_j (abhängig vom verwendeten Rechner).

Weiter wird für $j = 1, 2, \ldots$ der Faktor R_j der Residuenverkleinerung nach der j-ten Mehrgitter-Iteration definiert:

$$R_j = \frac{r_j}{r_{j-1}}\,,$$

[9] $u \in C^4(\overline{\Omega})$, vergleiche Abschnitt 1.1.2, Fußnote.

Die „durchschnittliche Konvergenzrate ρ“ (Average Reduction Factor, ARF) ist dann

$$\rho = \left(\frac{r_e}{r_a}\right)^{\frac{1}{e-a}} = \left(\prod_{j=a+1}^{e} R_j\right)^{\frac{1}{e-a}} ,$$

wobei $e > a$ geeignet gewählt sind. Weiterhin wird noch die asymptotische Konvergenzrate ρ^∞ der R_j für $j \to \infty$ definiert[10], falls der Grenzwert existiert:

$$\rho^\infty := \lim_{j\to\infty} R_j .$$

Zum Vergleich für die durchschnittlichen Konvergenzraten wurden die folgenden Verfahren verwendet:

- MG00, ein Mehrgitter-Verfahren aus [25] (auch abgedruckt in [58]), daß zur Lösung der entsprechenden Aufgabe mit Dirichlet-Randbedingungen gedacht ist. Es entspricht in der Wahl der Komponenten dem Verfahren HRED.
- ein Galerkin-Verfahren [29], welches bilineare Interpolation, Full Weighting und zebra-line Gauß-Seidel benutzt.

Konvergenzordnung:

Die vorgestellten CS-Verfahren weisen alle die Konvergenzordnung 2 auf, wie man anhand der Tabelle 2.9 erkennen kann: [11]

$$||u - \tilde{u}_h||_\infty = O(h^2) .$$

Somit besitzen die Verfahren sowohl Konsistenz- als auch Konvergenzordnung 2.

(Un-)Abhängigkeit der Konvergenzraten

Berechnet man die (asymptotischen) Konvergenzraten ρ (wie z.B. in Tabelle 2.2 für Beispiel (2) und $N = 256$ geschehen), so zeigt sich, daß sie prinzipiell unabhängig von der Wahl von h, f (siehe Beginn von Abschnitt 2.2.1) und dem Startwert $u_h^{(0)}$ sind. Die asymptotischen Konvergenzraten lassen sich allerdings nur mit $f \equiv 0$ und mit kompletter Anpassung gut berechnen (aufgrund von Rundungsfehlern, siehe Abschnitt 2.2.3). Lediglich bei ILEX und ω-FRED-V(ν_1,ν_2) zeigen sich Unterschiede (siehe Tabelle 2.4), die bei ILEX allerdings ihre Ursache darin haben sollten, daß die diskrete Kompatibilitätsbedingung auf den gröberen Gittern unterschiedlich schlecht bei verschiedenen Beispielen erfüllt ist, was wiederum die Konvergenzraten unterschiedlich verschlechtert.

[10] Bei praktischen Berechnungen werden hier aber höchstens 500 Iterationen durchgeführt.

[11] Hier ist $||u - \tilde{u}_h||_\infty = ||u - u_h||_\infty$, siehe daher Spalte $||u - u_h||_\infty$.

Die Auswertung der Tabelle 2.2 ergibt (ohne Betrachtung von MG00):

- Im W-Cycle weist ω-FRED je die besten durchschnittlichen Konvergenzraten ρ auf, gefolgt von FRED, HRED, FLEX und mit großem Abstand ILEX.

- HRED-W(2,1) und HRED-V(2,1) sind lediglich etwas schlechter als die entsprechenden FRED-Verfahren. HRED-W(1,1) und HRED-V(1,1) konvergieren allerdings nur etwa halb so schnell. Für X(2,1) ist dank des kleineren Rechenaufwandes also HRED, für X(1,1) dagegen FRED vorzuziehen.

- ρ(FRED) ist zwar gegenüber $\rho(\omega$-FRED) im W-Cycle etwa anderthalbmal so groß, aber dafür muß bei ω-FRED für die Berechnung jeder Komponente der neuen Näherung pro Relaxationsschritt eine Multiplikation und eine Addition mehr durchgeführt werden, weswegen sich *effektiv* FRED und ω-FRED nicht besonders unterscheiden. Da sich im V-Cycle ω-FRED sogar deutlich schlechter als FLEX verhält, ist dieses Verfahren im zweidimensionalen Fall keine gute Alternative zu FRED oder HRED.

- Weil vom Rechenaufwand her FLEX dem Verfahren FRED entspricht, aber ρ(FLEX) jeweils deutlich größer als ρ(FRED) ist, kann FLEX mit FRED nicht konkurrieren.

- Für FRED, HRED, FLEX (und MG00) gilt:

  $$\rho(W(2,1)) < \rho(V(2,1)) < \rho(W(1,1)) < \rho(V(1,1))\,.$$

  Für ω-FRED und ILEX gilt dagegen:

  $$\rho(W(2,1)) < \rho(W(1,1)) < \rho(V(2,1)) < \rho(V(1,1))\,.$$

  $\rho(FRED)$ bleibt immer unterhalb von 0.1, MG00 nur für V(1,1) nicht, HRED und FLEX für W(1,1) und V(1,1) nicht und ω-FRED für V(2,1) und V(1,1) nicht.

- Da die Raten für ILEX gegenüber den anderen schlecht sind und die Kompatibilitätsbedingung (2.3) auf den gröberen Gittern nicht erfüllt (s.o.) ist, ist dieses Verfahren für die gestellte Aufgabe nicht geeignet.

- Die durchschnittlichen Konvergenzraten stimmen für HRED gut mit den asymptotischen (siehe Tabelle 2.3) überein. Für (ω-)FRED und ILEX sind die Grenzwerte etwas schlechter, da nach einigen Iterationsschritten die Raten R_j etwas größer werden. Für FLEX konvergieren die R_j in keinem der getesteten Fälle (was seine Ursache letztlich darin hat, daß der ganze Iterationsoperator für das Mehrgitter-Verfahren in diesem Fall nicht symmetrisch ist und daher auch komplexe Eigenwerte haben kann).

Gegenüberstellung der (asymptotischen) Konvergenzraten $\rho(^{\infty})$ mit vergleichbaren Werten:

- Für FRED und HRED im W(ν_1,1)-Cycle ergeben sich Grenzwerte ρ^{∞}, die den Zwei-Gitter-Konvergenzraten ρ^* aus der rigorosen Fourieranalyse (siehe [60]) für die entsprechenden Verfahren mit Dirichlet-Randbedingungen sehr gleichen (siehe Tabelle 2.6). Die ρ^{∞}-Werte für HRED und MG00 sind zudem mit Ausnahme von V(1,1) fast identisch (siehe Tabelle 2.3).

  Der Wert für ω-FRED ist etwas schlechter als das entsprechende ρ^*. Dies liegt daran, daß nach etwa 30 Iterationen Rundungsfehler zu einem Sprung der R_j führen (um etwa 0.01) und sich danach die R_j auf einem etwas schlechteren Niveau von etwa 0.058 bis 0.061 bewegen.

- Die in [60] durchgeführte lokale Fourieranalyse für GS-LEX und Full Weighting bzw. Injection liefert lokale Raten ρ_{loc}, die recht nahe an den ρ-Werten für FLEX liegen (siehe Tabelle 2.7). Die ρ^{∞}(ILEX) dagegen weichen stark von den entsprechenden ρ_{loc} ab. Die Ursache dafür ist, daß auf den gröberen Gittern die Gleichungssysteme nicht mehr lösbar sind (s.o.).

- Der Vergleich von HRED mit MG00 zeigt, daß sie im W(2,1)-Cycle etwa gleichschnell sind, ansonsten allerdings MG00 vor allem für X(1,1) deutlich schneller ist (für die asymptotische Betrachtung stimmt dies nicht, dort unterscheiden sie sich kaum, s.o.).

- Vergleicht man nun HRED-W(2,1) zusätzlich mit dem Galerkin-Verfahren für periodische Randbedingungen, so ergeben sich für Beispiel (1)[12] etwa dieselben Raten, nämlich 0.033 (HRED) bzw. 0.032 (Galerkin). Allerdings ist der Rechenaufwand für das Galerkin-Verfahren höher - zumindest für quadratische Gebiete -, so daß HRED-W(2,1) hier effektiver ist.

  MG00-W(2,1) und das Galerkin-Verfahren für Dirichlet-Randbedingungen ergeben ähnliche Raten: MG00 konvergiert (je nach Beispiel) geringfügig schneller oder gleichschnell als HRED-W(2,1), das Galerkin-Verfahren dagegen etwas langsamer (siehe Tabelle 2.5).

Von den vorgestellten CS-Verfahren erweisen sich insgesamt FRED-W(1,1) und HRED-W(2,1) als die effektivsten: Sie weisen das richtige Verhältnis von kleiner Konvergenzrate ρ (bzw. ρ^{∞}) und Rechenaufwand pro Cycle vor und sind damit etwa gleich schnell. Da sowohl der Rechenaufwand (bis auf „Randeffekte“) als auch die asymptotischen Konvergenzraten für diese Verfahren gegenüber entsprechenden für Dirichlet-Randbedingungen etwa gleich sind, kann man dieses Ergebnis der Effektivitätsanalyse in [58] (Kapitel 8.2) entnehmen.

Damit ist also auch FRED-W(1,1) effektiver als das Galerkin-Verfahren für periodische Randbedingungen (für quadratische Gebiete).

[12] Für dieses Beispiel fanden sich Angaben von Konvergenzraten für das Galerkin-Verfahren.

Allerdings kann die Verwendung von Half Weighting zu Schwierigkeiten auf den gröberen Gittern führen (siehe Abschnitt 2.1.3). Als effektives *und* sicheres Verfahren bleibt damit FRED-W(1,1) übrig.

	Anpassung	ω-FRED	FRED	HRED	FLEX	ILEX	MG00
W(2,1)	nein	0.014	0.025	0.033	0.081	0.278	0.032
W(2,1)	ja			0.032	0.084		—
W(1,1)	nein	0.030	0.059	0.117	0.144	0.439	0.058
W(1,1)	ja	0.029		0.118	0.147		—
V(2,1)	nein	0.162	0.040	0.051	0.097	0.514	0.045
V(2,1)	ja		0.042	0.052	0.100		—
V(1,1)	nein	0.326	0.083	0.191	0.147	0.611	0.124
V(1,1)	ja		0.085				—

Tabelle 2.2: Durchschnittliche Konvergenzraten ρ für Beispiel (2). Wenn nicht anders angegeben, liefert die komplette Anpassung den gleichen Wert. Die grobe Anpassung hat hier keine Wirkung.

	ω-FRED	FRED	HRED	FLEX	ILEX	MG00
W(2,1)	0.035	0.052	0.034	k.K.	0.414	0.034
W(1,1)	0.061	0.074	0.125	k.K.	0.493	0.124
V(2,1)	0.276	0.080	0.066	k.K.	0.675	0.063
V(1,1)	0.497	0.115	0.201	k.K.	0.727	0.184

Tabelle 2.3: Asymptotische Konvergenzraten ρ^∞. Die Werte wurden für $f \equiv 0$, $u_h^{(0)}(x_i, y_j) = 10^{70}(i+j)$, $N = 256$ und mit kompletter Anpassung berechnet (k.K.=keine Konvergenz).

2.2.2 Test der FMG-Verfahren

Ein FMG-Verfahren soll folgende Eigenschaften besitzen (vergleiche [58]):

- Für die vom Verfahren berechnete Näherungslösung u_h^{FMG} der diskreten Lösung u_h gilt (in einer geeigneten Norm):

$$||u - u_h^{FMG}|| \leq 2||u - u_h|| \,. \tag{2.21}$$

- Die Anzahl der benötigten arithmetischen Operationen ist proportional der Anzahl der Gitterpunkte.

Da der Rechenaufwand für die hier beschriebenen Verfahren und der für entsprechende mit Dirichlet-Randbedingungen gleich ist (bis auf vernachlässigbar kleine Unterschiede durch die Behandlung der Randpunkte), ergibt sich die Gültigkeit der zweiten Bedingung aus den Überlegungen in [60].

Die Ergebnisse des Tests der Gültigkeit von (2.21) für die verschiedenen Verfahren (im W-Cycle) sind exemplarisch für Beispiel (2) in Tabelle 2.8 dargestellt. Für andere Beispiele ergibt sich ein analoges Bild.

	ω-FRED		ILEX	
N	V(2,1)	V(1,1)	W(2,1)	V(1,1)
32	0.131	0.319	0.369	0.578
64	0.187	0.387	0.397	0.648
128	0.233	0.446	0.410	0.694
256	0.276	0.497	0.414	0.727

Tabelle 2.4: Asymptotische Konvergenzraten ρ^{∞}. Die Werte wurden für $f \equiv 0$, $u_h^{(0)}(x_i, y_j) = 10^7 0(i+j)$, $N = 256$ und mit kompletter Anpassung berechnet.

	Randbedingungen	
Verfahren	periodisch	Dirichlet
HRED-W(2,1)	0.033	—
Galerkin	0.032	0.038
MG00-W(2,1)	—	0.031

Tabelle 2.5: Vergleich der durchschnittlichen Konvergenzraten verschiedener Verfahren für Beispiel (1).

Führt man die Berechnungen für $r = 1$ (siehe Abschnitt 2.1.2, FMG-Methoden) durch, so ist aus der Tabelle ersichtlich, daß (2.21) bei FRED, HRED und FLEX für FMG(X(ν_1,ν_2),1) immer, bei ω-FRED aber nur für FMG(W(ν_1,ν_2),1) erfüllt ist. ILEX erreicht aufgrund der schlechten Konvergenzraten (2.21) nicht.

Beginnt man mit einem ungünstigen Startwert für $u_m(x_{N_m}, y_{N_m})$ [13] , so zeigt sich, daß die (grobe) Anpassung unbedingt erforderlich ist, um (2.21) zu erreichen, wie man in Tabelle 2.9 im Vergleich zu Tabelle 2.10 klar sehen kann: Mit (grober) Anpassung kann man sogar $u_h^{(0)}(x_i, y_j) = 10^7 0(i+j)$ wählen und erhält trotzdem Ergebnisse, die denen aus Tabelle 2.8 entsprechen. Ohne Anpassung dagegen erhält man nicht einmal bei der Wahl von $u_h^{(0)}(x_j, y_k) = 10^7(i+j)$ und FMG(W(2,1),1) Werte, die (2.21) erfüllen. Zur Anpassung siehe auch den nächsten Abschnitt.

Tabelle 2.9 zeigt deutlich, daß es sich bei FMG(X(ν_1,ν_2),1) für FRED, HRED und FLEX um ein Verfahren zweiter Ordnung bzgl. der Konvergenz handelt: $||u - u_h^{FMG}||_{\infty} = O(h^2)$. ω-FRED erfüllt auch dies nur für W(ν_1,1) [14] und ILEX lediglich ansatzweise für W(2,1).

[13] Dieser Wert wird bei der Lösung des singulären Gleichungssystems auf dem gröbsten Gitter beibehalten. Hat man also z.B. als Startwert $u_h^{(0)}(x_i, y_j) = 10^7(i+j)$ gewählt, so ist $u_m(x_{N_m}, y_{N_m}) = u_h^{(0)}(x_N, y_N) = 10^7(N+N)$, und da hier immer $m = p$ verwendet wird: $u_p(x_2, y_2) = 10^7 2N$, $h_p = 1/2$. Für das konkrete Beispiel (2) erweist sich diese Wahl als ungünstig.

[14] wobei für W(1,1) die Werte bei diesem Beispiel „zu gut“ sind, bei einer Wahl von $r \geq 2$ allerdings wieder im Rahmen.

Verfahren	ρ^*	ρ^∞
ω-FRED-W(1,1)	0.052	0.061
FRED-W(1,1)	0.074	0.074
FRED-W(2,1)	0.053	0.052
HRED-W(1,1)	0.125	0.125
HRED-W(2,1)	0.033	0.034

Tabelle 2.6: Vergleich der aus der rigorosen Fourier-Analyse (RFA) erhaltenen Werte ρ^* mit den berechneten ρ^∞.

Verfahren	ρ_{loc}	ρ bzw. ρ^∞
FLEX-W(1,1)	0.193	0.144
FLEX-W(2,1)	0.119	0.081
ILEX-W(1,1)	0.200	0.493
ILEX-W(2,1)	0.089	0.414

Tabelle 2.7: Vergleich der aus der lokalen Fourier-Analyse (LFA) erhaltenen Werte ρ_{loc} mit den berechneten ρ(FLEX) bzw. ρ^∞(ILEX).

2.2.3 Anpassung

Verschiedene Tests mit den Programmen (ω-)FRED, FLEX, HRED und ILEX zeigen, daß bei Benutzung der *kompletten Anpassung* die Residuen im CS-Modus bis auf einen Wert c verkleinert werden, wobei c zwar abhängig von der gewählten Maschenweite h und der Beispielfunktion f ist, aber dann *unabhängig* vom Verfahren und vom Startwert (siehe Tabelle 2.11). Auch das Verfahren MG00 liefert bei der Maschenweite h als Endresiduum den Wert c. Daraus kann man schließen, daß c nur abhängig von der Maschinengenauigkeit ist.

Mit *grober Anpassung* oder völlig ohne Anpassung werden dagegen die Residuen *insgesamt* um etwa den Faktor 10^{15} (abhängig vom Rechner) verkleinert, falls nicht vorher schon ein Residuum der Größenordnung c erreicht ist. Das minimale Residuum *hängt* also in diesem Fall vom Startwert ab (siehe Tabelle 2.11).

Für Anfangsresiduen r_0 moderater Größe (d.h. z.B. maximal 10^{11}, falls man ein Endresiduum der Größe 10^{-4} erreichen möchte) kann man dann tatsächlich beobachten, daß die CS-Verfahren Näherungslösungen im Bereich des Diskretisierungsfehlers schon berechnet haben, *bevor* die komplette Anpassung signifikante Auswirkungen zeigt. Dies zeigt Tabelle 2.12.

In den Tabellen 2.2 und 2.8 sieht man für diesen Fall außerdem, daß die (komplette oder grobe) Anpassung im wesentlichen auch keine Auswirkungen auf die Konvergenzraten bzw. die Ergebnisse des FMG-Tests hat. Daß die Konvergenzraten für die Verfahren mit kompletter Anpassung meist geringfügig schlechter sind als ohne Anpassung, liegt daran, daß für erstere ein paar Iterationen mehr zur Berechnung von ρ benutzt werden können und zum Ende hin (d.h. in der Nähe von c) die Raten R_j sich meist geringfügig

Z	Anpassung	ω-FRED	FRED	HRED	FLEX	ILEX
V(1,1)	ja	4.3E-1	1.0E-3	6.5E-4	1.3E-3	1.8E-1
	nein	4.5E-1				
V(2,1)	ja	4.7E-2	9.0E-4	6.6E-4	1.0E-3	4.0E-2
	nein					
W(1,1)	ja	1.9E-4	7.1E-4	7.1E-4	7.1E-4	3.3E-2
	nein	2.0E-4				
W(2,1)		7.0E-4	7.1E-4	7.1E-4	7.1E-4	3.4E-3

Tabelle 2.8: FMG-Test für Beispiel (2). Angegeben sind jeweils $||u_h^{FMG} - u||_\infty$ nach FMG(Z,1). Mit Anpassung ist hier die komplette Anpassung gemeint. Wenn nicht anders angegeben, liefert sie den gleichen Wert. Werte, die man mit grober Anpassung erhält, liegen zwischen denen mit kompletter bzw. ohne Anpassung. Hier ist $||u - u_h||_\infty = 7.1 \cdot 10^{-4}$.

verschlechtern.

Anders liegt der Fall bei der Berechnung der asymptotischen Konvergenzraten ρ^∞. Hier ist unbedingt die komplette Anpassung erforderlich, weil sonst nicht genug Iterationsschritte betrachtet werden können: Eine Residuenverkleinerung um 10^{15} reicht hier nicht aus. Nur mit der kompletten Anpassung kann man für $f \equiv 0$ die Residuen auf 0 verkleinern.

Die FMG-Verfahren kommen nicht ohne eine Anpassung aus, wenn die berechnete Lösung auf dem gröbsten Gitter zu groß ist, d.h. wenn

$$\sum_{j,k=1}^{N_m} u_m(x_j, y_k) \gg 0$$

gilt. Allerdings genügt in einem solchen Falle üblicherweise die grobe Anpassung, um numerische Schwierigkeiten durch betraglich große Werte auszuschalten, da der Transfer dieser „kleinen“ exakten Lösung auf das nächstfeinere Gitter eine Startnäherung $u_{m-1}^{(0)}$ ergibt, die ein kleines Anfangsresiduum r_0 liefern sollte. Ergebnisse dieser (erfolgreichen) Vorgehensweise finden sich in Tabelle 2.9.

Zusammenfassend kann man also sagen, daß nur bei sehr großen Anfangsresiduen Vorsicht geboten ist. Da die komplette Anpassung den Rechenaufwand erhöht, sollte also zu Beginn eines *CS-Verfahrens* durch Berechnung von r_0 getestet werden, ob sie nötig ist. Weil aber die einmalige Anpassung auf dem gröbsten Gitter weniger aufwendig als die Berechnung von r_0 ist, sollte bei FMG direkt die grobe Anpassung verwendet werden.

2.2.4 Fazit

Von den getesteten fünf Verfahren erweisen sich HRED-W(2,1) und FRED-W(1,1) sowohl in der FMG- als auch in der CS-Version als die günstigsten, weil sie die kleinsten *effektiven* Konvergenzraten (unter Einbeziehung des Rechenaufwandes) besitzen und die in 2.2.2 beschriebenen Kriterien für FMG-Verfahren erfüllen. Allerdings kann es bei Verwendung von Half

Z	N	$\|\|u-u_h\|\|_\infty$	ω-FRED	FRED	HRED	FLEX	ILEX
V(1,1)	32	4.6E-2	2.3E-1	6.0E-2	4.1E-2	7.6E-2	5.9E-1
	64	1.1E-2	3.2E-1	1.6E-2	1.1E-2	2.0E-2	4.2E-1
	128	2.9E-3	3.8E-1	4.1E-3	2.6E-3	5.3E-3	3.1E-1
	256	7.1E-4	4.3E-1	1.0E-3	6.8E-4	1.3E-3	2.4E-1
V(2,1)	32	4.6E-2	2.3E-2	5.5E-2	4.3E-2	6.0E-2	1.9E-1
	64	1.1E-2	2.4E-2	1.4E-2	1.1E-2	1.6E-2	1.1E-1
	128	2.9E-3	3.9E-2	3.6E-3	2.7E-3	4.0E-3	6.6E-2
	256	7.1E-4	4.7E-2	9.0E-4	6.6E-4	1.0E-3	4.1E-2
W(1,1)	32	4.6E-2	4.3E-2	5.0E-2	4.5E-2	5.5E-2	4.2E-1
	64	1.1E-2	7.1E-3	1.2E-2	1.1E-2	1.2E-2	2.1E-1
	128	2.9E-3	9.4E-4	2.9E-3	2.8E-3	2.9E-3	9.9E-2
	256	7.1E-4	1.9E-4	7.1E-4	7.1E-4	7.1E-4	4.2E-2
W(2,1)	32	4.6E-2	4.8E-2	4.9E-2	4.5E-2	5.2E-2	1.7E-1
	64	1.1E-2	1.1E-2	1.2E-2	1.1E-1	1.2E-2	5.5E-2
	128	2.9E-3	2.8E-3	2.9E-3	2.8E-3	2.9E-3	1.5E-2
	256	7.1E-4	7.0E-4	7.1E-4	7.1E-4	7.1E-4	3.5E-3

Tabelle 2.9: FMG-Test für Beispiel (2) und Startwert $u_h^{(0)}(x_i,y_j) = 10^{70}(i+j)$. Angegeben ist jeweils $||u_h^{FMG}-u||_\infty$ nach FMG(Z,1). Die Werte wurden mit grober Anpassung berechnet.

Z	N	$\|\|u-u_h\|\|_\infty$	ω-FRED	FRED	HRED	FLEX	ILEX
W(2,1)	256	7.1E-4	1.7E-2	7.2E-2	4.2E-2	6.0E-2	4.6E-2

Tabelle 2.10: FMG-Test für Beispiel (2) und Startwert $u_h^{(0)}(x_i,y_j) = 10^{7}(i+j)$. Angegeben sind jeweils $||u_h^{FMG}-u||_\infty$ nach FMG(Z;1). Die Werte wurden ohne Anpassung berechnet.

Weighting zu Schwierigkeiten auf den gröberen Gittern kommen, da die Lösbarkeit der Gleichungssysteme nicht gesichert ist, auch wenn in allen getesteten Beispielen dies nicht in sichtbar war. Eine erzwungene Kompatibilität auf den groben Gittern (durch Änderung der f_j, siehe (2.15)) für Verfahren, die Injection oder Half Weighting (siehe dazu auch Kapitel 3) verwenden, verbessert die Werte nicht. Damit bleibt also FRED-W(1,1) als effektives *und* sicheres Verfahren.

Eine Anpassung ist nur bei sehr großen Anfangsresiduen oder bei Berechnung von asymptotischen Konvergenzraten notwendig. Bei FMG reicht die grobe Anpassung, die den Rechenaufwand nur unwesentlich heraufsetzt.

Weil außerdem (zumindest für die hier betrachteten quadratischen Gebiete) FRED-W(1,1) und HRED-W(2,1) effektiver als das Galerkin-Verfahren [29] sind, ergibt sich insgesamt:

Zur Lösung der zweidimensionalen Poissongleichung auf einem quadratischen Gebiet mit periodischen Randbedingungen können mit Erfolg CS-

f(x,y)	N	$u_h^{(0)}(x_i, y_j)$	AR	ER	GRF	c
(2.17)	64	0	7.9E+01	3.6E-12	2.2E+13	3.6E-12
		i+j	5.2E+05	4.7E-10	1.1E+15	4.1E-12
		i+1000j	2.6E+08	1.3E-07	2.0E+15	3.2E-12
	256	0	7.9E+01	6.5E-11	1.2E+12	6.5E-11
		i+j	3.4E+07	3.0E-08	1.1E+15	7.3E-11
		i+1000j	1.7E+10	1.0E-05	1.7E+15	6.5E-11
(2.19)	256	0	7.5E+02	5.8E-10	1.3E+12	5.8E-10
		i+j	3.4E+07	3.0E-08	1.1E+15	5.8E-10
		i+1000j	1.7E+10	1.0E-05	1.7E+15	5.8E-10

Tabelle 2.11: Zur Anpassung. Es ist AR das Anfangsresiduum r_0, ER das Endresiduum (mit grober oder ohne Anpassung) und GRF=AR/ER. Zu c siehe Text.

und FMG-Verfahren eingesetzt werden, die denen für die Poissongleichung mit Dirichlet-Randbedingungen in der Wahl der Komponenten sowie hinsichtlich des Rechenaufwandes und der Parallelisierbarkeit entsprechen. Es ergeben sich bei geeigneter Komponentenwahl sehr ähnliche durchschnittliche bzw. asymptotische Konvergenzraten.

Verfahren		j	i_{anp}
ω-FRED	W(2,1)	4	8
	W(1,1)	5	9
	V(2,1)	11	17
	V(1,1)	24	34
FRED	W(2,1)	5	10
	W(1,1)	6	11
	V(2,1)	6	11
	V(1,1)	10	11
HRED	W(2,1)	5	10
	W(1,1)	8	16
	V(2,1)	8	11
	V(1,1)	13	18
FLEX	W(2,1)	6	12
	W(1,1)	8	16
	V(2,1)	9	14
	V(1,1)	12	17
ILEX	W(2,1)	10	23
	W(1,1)	19	> 30
	V(2,1)	33	> 50
	V(1,1)	46	> 60

Tabelle 2.12: Zur Anpassung. Werte wurden für Beispiel (2) berechnet. j gibt die Anzahl der Iterationen an, die zum Erreichen von $||u_h^{(j)} - u||_\infty \leq 7.1E-4$ nötig ist (ohne Anpassung). Erst nach i_{anp} Iterationen unterscheiden sich die R_k der Verfahren mit kompletter bzw. ohne Anpassung.

Kapitel 3

Dreidimensionales Modellproblem

3.1 Diskretisierung und Mehrgitter-Verfahren

3.1.1 Diskretisierung

Mit Hilfe der bisherigen Ergebnisse soll nun das dreidimensionale Ausgangsproblem in der Form (1.6) - (1.8) (bzw. (1.27) - (1.29) mit $a_1 = a_2 = a_3 = 1$)

$$-u_{xx} - u_{yy} - u_{zz} = f \quad \text{in } \Omega \tag{3.1}$$

mit periodischen Randbedingungen in $\partial[0,1]^3$:

$$\begin{aligned} u(0,y,z) &= u(1,y,z)\,, \\ u(x,0,z) &= u(x,1,z)\,, \\ u(x,y,0) &= u(x,y,1) \end{aligned} \tag{3.2}$$

und

$$\begin{aligned} u_x(0,y,z) &= u_x(1,y,z)\,, \\ u_y(x,0,z) &= u_y(x,1,z)\,, \\ u_z(x,y,0) &= u_z(x,y,1) \end{aligned} \tag{3.3}$$

mit einem Mehrgitter-Verfahren gelöst werden. Im folgenden bezeichnen

$$\begin{aligned} G_h &= \{(x_j, y_k, z_l) \mid j,k,l \in \mathbb{Z}\}\,, \\ \Omega_h &= G_h \cap [h,1]^3\,, \\ \partial\Omega_h &= G_h \cap \partial([0,1+h]^3)\,, \end{aligned}$$

$$x_j = jh,\ y_j = jh,\ z_j = jh \text{ und } h = \frac{1}{N} \text{ mit } N = 2^p,\ p \in \mathbb{N}\,.$$

Dann führt eine zu Kapitel 2 analoge Vorgehensweise zu folgender Diskretisierung zweiter Ordnung ($m_d = 2$):

$$L_h u_h = h^2 f_h \quad \text{in } \Omega_h\,, \tag{3.4}$$

wobei in $\{(x_j, y_k, z_l) \mid j,k,l = 2, ..., N-1\}$

$$L_h = \left[\begin{array}{c|ccc|c} & & -1 & & \\ -1 & -1 & 6 & -1 & -1 \\ & & -1 & & \end{array}\right]_h$$

gilt und L_h für die restlichen Punkte aus Ω_h durch Ausnutzung der geforderten Periodizität von u_h abgeändert werden muß. Dazu werden also die folgenden Gleichungen verwendet:

$$\begin{array}{rll} & u_h(x_0, y_k, z_l) = u_h(x_N, y_k, z_l) & , \quad u_h(x_1, y_k, z_l) = u_h(x_{N+1}, y_k, z_l) \, , \\ (3.5) & u_h(x_j, y_0, z_l) = u_h(x_j, y_N, z_l) & , \quad u_h(x_j, y_1, z_l) = u_h(x_j, y_{N+1}, z_l) \, , \\ & u_h(x_j, y_k, z_0) = u_h(x_j, y_k, z_N) & , \quad u_h(x_j, y_k, z_1) = u_h(x_j, y_k, z_{N+1}) \, . \end{array}$$

Somit greift L_h **nur** auf Punkte in Ω_h zu. Die Eigenfunktionen und Eigenwerte des Operators L_h finden sich in Abschnitt 1.2.3, wobei in (1.27),(1.28) und (1.29) dann $a_1 = a_2 = a_3 = 1$ zu setzen ist.

Das Problem (3.4) ist entsprechend dem ein- und zweidimensionalen Fall (vergleiche Lemma 1 und (2.3)) genau dann bis auf eine Konstante eindeutig lösbar, wenn die diskrete Kompatibilitätsbedingung

$$h^2 \sum_{j,k,l=1}^{N} f_h(x_j, y_k, z_l) = 0 \tag{3.6}$$

erfüllt ist. Diese Bedingung soll ein gegebenes f_h hier erfüllen. Dazu geht man wie in den Abschnitten 1.2.3 und 2.1.3 vor:

Aufgrund der Trapezformel (für $f \in C^2([0,1]^3)$) und den dreidimensionalen Kompatibilitätsbedingungen (siehe Satz 1) gilt:

$$h^3 \sum_{j,k,l=1}^{N} f(x_j, y_k, z_l) = \int_{[0,1]^3} f(r)\, dr + O(h^2) = O(h^2) \, .$$

Daraus folgt:

$$h^2 \sum_{j,k,l=1}^{N} f(x_j, y_k, z_l) = O(h) \, .$$

Hier führt (1.3) lediglich zu einer Näherung erster Ordnung für (3.6), wenn man direkt $f|_{\Omega_h}$ als f_h einsetzt. Da eine vorgegebene Funktion f also die diskrete Kompatibilitätsbedingung im allgemeinen nicht exakt erfüllt, wird wie in (1.22) und (2.12) definiert:

$$f_h(x_j, y_k, z_l) := f(x_j, y_k, z_l) - \bar{f} \, , \tag{3.7}$$

wobei $\bar{f}$ den Durchschnitt der Werte $f(x_j, y_k, z_l)$ angibt:

$$\bar{f} := \frac{1}{N^3} \sum_{j,k,l=1}^{N} f(x_j, y_k, z_l) = h^3 \sum_{j,k,l=1}^{N} f(x_j, y_k, z_l) \, .$$

Somit erfüllt die neue Funktion f_h die diskrete Kompatibilitätsbedingung (3.6) und stellt wegen

$$\begin{aligned} f(x_j, y_k, z_l) - f_h(x_j, y_k, z_l) &= \bar{f} \\ &= \int_{[0,1]^3} f(r)\, dr + O(h^2) \\ &= O(h^2) \end{aligned}$$

eine Näherung zweiter Ordnung für $f|_{\Omega_h}$ dar. Zusammen mit den Ergebnissen für den ein- und zweidimensionalen Fall (siehe 1.2.3 und 2.1.3) ergibt sich also:

Die Kompatibilitätsbedingung $\int_{[0,1]^d} f(r)\, dr = 0$ *für die d-dimensionale Poissongleichung* ($d \in \{1,2,3\}$) *mit periodischen Randbedingungen führt zu einer Näherung* $(4-d)$*-ter Ordnung für die diskrete Kompatibilitätsbedingung*

$$h^2 \sum_{j_1,\ldots,j_d=1}^{N} f(x_{j_1}, \ldots, x_{j_d}) = 0\,.$$

Die Funktion $f_h := f - \bar{f}$ *stellt aber immer eine Näherung zweiter Ordnung für* $f|_{\Omega_h}$ *dar, so daß sich mit* (3.4) *insgesamt ein Verfahren zweiter Ordnung bzgl. der Konsistenz zur Bestimmung einer Näherungslösung für u ergibt.*

3.1.2 Mehrgitter-Verfahren

Wie die Ergebnisse des vorherigen Kapitels gezeigt haben, sind die Red-Black-Verfahren (ω-)FRED und HRED den lexikografischen Verfahren ILEX und FLEX vorzuziehen. Ein ähnliches Bild erhält man für entsprechende Verfahren für die dreidimensionale Poissongleichung mit Dirichlet-Randbedingungen (vergleiche [7]). Daher werden hier nur Mehrgitter-Verfahren betrachtet, die (ω-)Red-Black-Gauß-Seidel als Glätter verwenden. Im einzelnen sind sie durch die folgenden Komponenten charakterisiert:

- *Mehrgitter-Cycle*

  Es wird $\gamma = 1$ (V-Cycle) oder $\gamma = 2$ (W-Cycle) verwendet.

- *Relaxationsverfahren*

  Es werden $\nu_1 \in \{1,2\}$ Relaxationsschritte vor und $\nu_2 = 1$ Relaxationsschritte nach der Grobgitterkorrektur durchgeführt, und zwar mit ω-GS-RB oder GS-RB.

  In Abschnitt 2.1.2 (Relaxation) wurden bereits obere und untere Grenzen für ω_{opt} sowie eine sehr gute Arbeitsgröße (ein etwas kleinerer Wert als ω_{ub}) angegeben. Im vorliegenden dreidimensionalen Fall berechnet sich ω_{ub} zu:

$$\omega_{ub} = \frac{2}{1 + \sqrt{1 - C_{max}^2}} = 1.1459$$

mit $C_{max} = 1 - c_{min} = 1 - \frac{1}{3}$ für den dreidimensionalen Laplace-Operator. Hier werden die auch in der Literatur (siehe z.B. [65, 64]) verwendeten Werte

$$\omega_1 := 1.1\,,$$
$$\omega_2 := 1.15$$

getestet.

- *Restriktion*

  Als Restriktion zum nächstgröberen Gitter kommen Half oder Full Weighting zum Einsatz. Im Zusammenspiel mit GS-RB ist Half Weighting wie im zweidimensionalen Fall dasselbe wie Half Injection.

  Da Satz 3 genauso für den dreidimensionalen Fall gilt, reicht die Gültigkeit von (3.6) wie im zweidimensionalen bereits aus, um die Lösbarkeit der (singulären) Gleichungssysteme auf allen betrachteten Gittern zu folgern, falls FW als Restriktion verwendet wird.

- *Exakte Lösung auf dem gröbsten Gitter*

  Dies wird analog dem zweidimensionalen Fall durchgeführt. In den numerischen Tests wird grundsätzlich $m = p$ festgesetzt, d.h. alle zur Verfügung stehenden gröberen Gitter werden verwendet. Zur Lösung auf dem gröbsten Gitter wird dabei $u_{h_p}(x_1, y_2, z_2) = 0$ festgelegt.

- *Prolongation*

  Trilineare Interpolation wird verwendet.

- *Behandlung der „Randpunkte" auf allen Leveln*

  Analog dem zweidimensionalen Fall wird konsequent (3.5) (d.h. die Torusstruktur) ausgenutzt.

- *FMG-Interpolation*

  Hier wird die (tri)kubische Interpolation benutzt. Da für ihre Interpolationsordnung $\kappa_{FMG} = 4$ gilt, ist also $\kappa_{FMG} > m_d$ erfüllt (hier ist $m_d = 2$, vergleiche Abschnitt 2.1.2).

In der Tabelle 3.1 sind die getesteten Kombinationen aufgeführt.

Verfahrensname	ω_1-FR3D	ω_2-FR3D	FR3D	HR3D
Glättungsverfahren	ω_1-GS-RB	ω_2-GS-RB	GS-RB	GS-RB
Restriktion	FW	FW	FW	HW

Tabelle 3.1: Die getesteten Mehrgitter-Verfahren für das dreidimensionale Modellproblem. Es sind nur die Komponenten aufgeführt, in denen sich die Verfahren unterscheiden.

Bemerkung: Die vorgestellten Verfahren können wie im zweidimensionalen Fall analog entsprechenden Algorithmen z.B. für Dirichlet- oder Neumannsche Randbedingungen parallelisiert werden.

3.2 Numerische Ergebnisse

Konkrete Werte werden im folgenden meist für die Beispielfunktion (1), die auch in [7] verwendet wurde, oder für die Beispielfunktion (2) angegeben. Für andere Beispiele[1] zeigt sich aber das gleiche Bild. Weitere Beispiele (mit $f \notin C^2(\overline{\Omega})$) finden sich in Kapitel 5.

Beispiel (1)

$$f(x,y,z) = 12\pi sin(2\pi x)sin(2\pi y)sin(2\pi z)\,.$$

Die exakte Lösung dieses Problems ist die Funktionenschar

$$u(x,y,z) = sin(2\pi x)sin(2\pi y)sin(2\pi z) + c \quad \text{mit } c \in \mathbb{R}\,.$$

Beispiel (2)

$$f(x,y,z) = 4\pi sin(2\pi x + 1) + cos(2\pi y + 2) + sin(2\pi z + 3)\,.$$

Die exakte Lösung dieses Problems ist die Funktionenschar

$$u(x,y,z) = sin(2\pi x + 1) + cos(2\pi y + 2) + sin(2\pi z + 3) + c \quad \text{mit } c \in \mathbb{R}\,.$$

Wenn nicht anders angegeben, werden jeweils $N = 128$ und $u_h^{(0)}(x_j, y_k, z_l) = j + k + l$ verwendet.

Für Beispiel (1) mit $N = 64$ fanden sich asymptotische Konvergenzraten für das entsprechende Dirichlet-Problem in [7]: B-FR bzw. B-HR meinen dabei Mehrgitter-Verfahren, die in der Wahl der Komponenten FR3D bzw. HR3D entsprechen.

3.2.1 Konvergenzordnung und -raten

Den dreidimensionalen Fall kann man prinzipiell mit dem zweidimensionalen vergleichen. Er weist aber doch einige Besonderheiten auf. Wie im zweidimensionalen sind fast alle CS-Verfahren auch hinsichtlich ihrer Konvergenz zweiter Ordnung:

$$||u - \tilde{u}_h||_\infty = O(h^2)\,.$$

Lediglich HR3D-V(1,1) konvergiert nicht bzw. nur extrem schlecht: Die Rate beträgt etwa 0.99. Dies ist ein erster Hinweis darauf, daß es bei HR3D anders als im zweidimensionalen Fall zu Schwierigkeiten kommt. Der Grund

[1] mit $u \in C^4(\overline{\Omega})$, siehe Abschnitt 1.1.2, Fußnote.

dafür ist, daß die Gleichungssysteme auf den gröberen Gittern bei allen getesteten Beispielen tatsächlich nicht lösbar sind (siehe Abschnitt 2.1.3), was Auswirkungen auf die vom Glätter bzw. auf dem gröbsten Gitter berechneten (Näherungs-)Lösungen hat und letztlich die Konvergenzraten stark verschlechtert.

Dieses Problem tritt bei HR3D-X(ν_1,ν_2) für die getesteten $\nu_1 \in \{1,2\}$, $\nu_2 = 1$ auf, allerdings sind die Auswirkungen nicht in jedem Fall so dramatisch wie bei V(1,1). Es zeigt sich im direkten Vergleich, daß sowohl $\gamma = 2$ gegenüber $\gamma = 1$ als auch mehr Relaxationsschritte dazu führen, daß die diskrete Kompatibilitätsbedingung auf den gröberen Gittern jeweils immer besser erfüllt ist. Die Relaxationsverfahren reduzieren auch bei nicht erfüllter Kompatibilitätsbedingung die Residuen (meistens), allerdings dies umso besser, je kleiner der Wert der Summe in der Kompatibilitätsbedingung ist. So kommt es, daß die ρ^∞(HR3D) für W(2,1), W(1,1) und V(2,1) nicht zu stark von entsprechenden Werten für B-HR abweichen (siehe Tabelle 3.3). Bei V(1,1) dagegen summieren sich die Fehler.

Es nützt wie im zweidimensionalen Fall allerdings nichts, auf den gröberen Gittern die Kompatibilität der rechten Seite zu erzwingen (vergleiche Abschnitte 2.1.3 und 2.2).

(Un-)Abhängigkeit der Konvergenzraten

Wie auch die Tabellen 3.3, 3.4 und 3.5 zeigen, sind die Konvergenzraten ρ und die asymptotischen Raten ρ^∞ für HR3D, ω_1- und ω_2-FR3D im W-Cycle, für FR3D aber durchgängig prinzipiell unabhängig von h, f (siehe Beginn von Abschnitt 3.2) und der Startnäherung. Die ρ^∞ lassen sich wie im zweidimensionalen Fall nur mit $f \equiv 0$ und kompletter Anpassung berechnen.

Beim V-Cycle macht sich bei HR3D wieder die nicht erfüllte Kompatibilitätsbedingung auf den gröberen Gittern bemerkbar: Bei V(2,1) werden die Werte mit wachsendem N nur etwas größer (um 0.01 bis 0.02), bei V(1,1) dagegen um mehr als 0.1 für verdoppeltes N (siehe Tabelle 3.5).

ω_1- und ω_2-FR3D zeigen im V-Cycle wie ω-FRED eine h-Abhängigkeit der (asymptotischen) Konvergenzraten.

Abhängigkeit von ω

Beim Vergleich der Verfahren ω_1-FR3D, ω_2-FR3D und FR3D stellt sich heraus, daß ω_2 **für den W-Cycle** die beste Wahl ist. Dies ergibt sich aus den folgenden Beobachtungen:

Die Tabellen 3.2, 3.3 und 3.4 zeigen, daß die (asymptotischen) Raten für ω_2-FR3D kleiner als die für ω_1-FR3D und diese wiederum kleiner als die für FR3D sind. Wählt man ein *etwas* kleineres ω als 1.15, ändern sich die Raten nicht merklich. Eine Wahl von $\omega_1 = 1.1$ erhöht allerdings bereits die Werte, so daß sich ω_2 hier als die richtige Wahl erweist.

Den Aussagen in [65] entsprechend verbessert das optimale ω die Glättungseigenschaften für die dreidimensionale Poissongleichung so sehr, daß sich der Einsatz dieses Überrelaxationsparameters lohnt. Dies kann man hier daran beobachten, daß $\rho(\omega_2$-FR3D) deutlich weniger als halb so groß

wie ρ(FR3D) und ρ^∞(ω_2-FR3D) nur knapp mehr als halb so groß wie ρ^∞(FR3D), der Rechenaufwand aber klar weniger als doppelt so groß ist. Also ist anders als im zweidimensionalen Fall ω_2-FR3D im W-Cycle effektiver als FR3D. Da außerdem ρ(ω_2-FR3D-W(1,1)) deutlich mehr als doppelt so groß wie ρ(ω_2-FR3D-W(2,1)) ist (bei ρ^∞ ist es fast das Doppelte), aber der Rechenaufwand bei W(2,1) nicht doppelt so groß wie bei W(1,1) ist, erweist sich letztlich ω_2-FR3D-W(2,1) als das günstigste Verfahren.

Im V-Cycle liegt der Fall anders. Hier liefert FR3D mit Abstand die besten Raten, und das Verhalten kehrt sich im ganzen um, denn nun ist auch ω_1- besser als ω_2-FR3D.

Vergleich mit entsprechenden Werten für Dirichlet-Randbedingungen

Die asymptotischen Konvergenzraten ρ^∞ (siehe Tabelle 3.3) unterscheiden sich für FR3D und B-FR kaum, die Werte für FR3D sind nur geringfügig schlechter (dritte Nachkommastelle). Bei HR3D liegt der Fall anders. Von W(2,1) bis V(1,1) werden die Unterschiede zu B-HR immer größer: 0.009, 0.034, 0.082, 0.424. Daß die Werte für HR3D immer schlechter gegenüber denen für B-HR sind, liegt an den nicht erfüllten Kompatibilitäsbedingungen auf den gröberen Gittern (s.o.).

Vergleicht man die aus der RFA für das entsprechende Dirichlet-Problem stammenden ρ^*(W(1,1)) mit den berechneten ρ^∞(W(1,1)) für FR3D und ω_i-FR3D (siehe Tabelle 3.6), so stellt sich heraus, daß sie für $\omega = 1$ (d.h. FR3D) sehr gut übereinstimmen, für $\omega = 1.1$ die ρ^∞ nur geringfügig schlechter sind, aber für $\omega = 1.15$ deutlich abweichen (um etwa 0.025).

Der Grund hierfür ist, daß bei der Berechnung der ρ^∞ für ω_1- bzw. ω_2-FR3D nach etwa 100 bzw. 50 Schritten innerhalb von einigen wenigen Schritten ein etwas größerer Wertesprung der R_j zu beobachten ist. Vor dem Sprung liegen die R_j etwa auf dem Niveau der ρ^* (bei etwa 0.095 bzw. 0.078), danach allerdings bei 0.10 bzw. etwas darunter, und gehen dann langsam auf die in der Tabelle angegebenen Werte zu. Der Wertesprung sollte seine Ursache in Rundungsfehlern haben, die (im dreidimensionalen Fall) auch durch die komplette Anpassung nicht mehr abgefangen werden können.

Wie sehr die ρ^∞ von den ρ^* abweichen, ist also im dreidimensionalen Fall davon abhängig, wieviele Iterationen man maximal zur Bestimmung des „Endwertes“ zuläßt.

3.2.2 Test der FMG-Verfahren

Anders als im zweidimensionalen Fall genügt hier in der Mehrzahl der Fälle nicht $r = 1$, um die Bedingung

$$||u_h^{FMG} - u||_\infty < 2||u_h - u||_\infty \tag{3.8}$$

(siehe (2.21)) zu erfüllen. FR3D und HR3D kommen sowohl bei W(ν_1,ν_2), als auch bei V(2,1) mit $r = 1$ aus, die anderen schaffen dies nur für W(2,1)

	Anpassung	ω_1-FR3D	ω_2-FR3D	FR3D	HR3D
W(2,1)	nein	0.038	0.024	0.084	0.127
W(2,1)	ja				0.125
W(1,1)	nein	0.093	0.070	0.184	0.275
W(1,1)	ja			0.185	
V(2,1)	nein	0.278	0.323	0.118	0.229
V(2,1)	ja			0.120	
V(1,1)		0.432	0.502	0.224	0.993

Tabelle 3.2: Durchschnittliche Konvergenzraten ρ (für N=128 und Beispiel (1)). Wenn nicht anders angegeben, liefert die komplette Anpassung den gleichen Wert. Die grobe Anpassung hat hier keine Wirkung.

	ω_1-FR3D	ω_2-FR3D	FR3D	HR3D	B-FR	B-HR
W(2,1)	0.070	0.056	0.099	0.132	0.096	0.123
W(1,1)	0.104	0.102	0.197	0.296	0.191	0.262
V(2,1)	0.332	0.375	0.144	0.231	0.150	0.149
V(1,1)	0.541	0.617	0.242	0.876	0.222	0.452

Tabelle 3.3: Werte für ρ^∞. Sie wurden für $f \equiv 0$, $u_h^{(0)} = 10^{70}(i+j+k)$, N=64 und mit kompletter Anpassung berechnet. Die Werte für B-FR und B-HR stammen aus [7].

(siehe Tabelle 3.7).[2] Im W(1,1)-Cycle reicht dann allerdings immer $r = 2$.

Bei den Verfahren ω_i-FR3D macht es sich im V-Cycle und bei HR3D im V(1,1)-Cycle deutlich bemerkbar, daß die Konvergenzraten schlecht sind: selbst $r = 2$ hilft noch nicht. Lediglich bei FR3D-V(ν_1,ν_2) und HR3D-V(2,1) genügt diese Wahl: Diese Werte erfüllen (3.8), allerdings nicht mehr (wie vorher meistens) $||u_h^{FMG} - u||_\infty < ||u_h - u||_\infty$.

In den Fällen, wo ein Verfahren dann für passendes r im entsprechenden Cycle sowohl für $N = 64$, als auch für $N = 128$ der Bedingung (3.8) genügt, erweist es sich als Verfahren zweiter Ordnung bzgl. der Konvergenz:

$$||u - u_h^{FMG}||_\infty = O(h^2) \; .$$

3.2.3 Anpassung

Bei Untersuchungen zur Wirkung der groben bzw. der kompletten Anpassung kommt man zu den gleichen Ergebnissen wie im zweidimensionalen Fall (siehe Abschnitt 2.2.3). Mit grober oder völlig ohne Anpassung werden bei den CS-Verfahren die Residuen um etwa den Faktor 10^{15} verkleinert, falls nicht vorher schon ein Residuum der Größenordnung c erreicht ist. Mit

[2] Bei ω_i-FR3D und HR3D kann es je nach Beispiel passieren, daß für $r = 1$ manche Werte „zu gut“ sind. Dies ist Zufall, wie dann die Werte für $r = 2$ (oder größer) zeigen, die wieder denen für $||u_{1/64} - u||_\infty$ bzw. $||u_{1/128} - u||_\infty$ entsprechen.

	ω_1-FR3D	ω_2-FR3D	FR3D	HR3D
W(2,1)	0.070	0.059	0.100	0.132
W(1,1)	0.109	0.107	0.197	0.296
V(2,1)	0.386	0.445	0.148	0.243
V(1,1)	0.604	0.691	0.244	0.991

Tabelle 3.4: Werte für ρ^∞. Sie wurden für $f \equiv 0$, $u_h^{(0)} = 10^{70}(i+j+k)$, N=128 und mit kompletter Anpassung berechnet.

N	32	64	128
W(2,1)	0.132	0.132	0.132
W(1,1)	0.296	0.296	0.296
V(2,1)	0.216	0.231	0.243
V(1,1)	0.754	0.876	0.991

Tabelle 3.5: Werte für ρ^∞(HR3D). Sie wurden für $f \equiv 0$, $u_h^{(0)} = 10^{70}(i+j+k)$ und mit kompletter Anpassung berechnet.

kompletter Anpassung werden die Werte eben auf diesen vom Verfahren und Startwert unabhängigen Wert c verkleinert.

Bei den FMG-Verfahren genügt es, die grobe Anpassung - d.h. Anpassung nur auf dem gröbsten Gitter - durchzuführen. Diese ist aber nur dann nötig (und hat nur dann Wirkung), wenn die Festlegung von $u_{h_p}(x_1, y_2, z_2) = 0$ zu folgendem führt:

$$\sum_{j,k,l=1}^{2} u_{h_p}(x_j, y_k, z_l) \gg 0 \ .$$

3.2.4 Fazit

Anders als beim zweidimensionalen Modellproblem lohnt es sich hier im W-Cycle, ein überrelaxiertes Verfahren einzusetzen. $\omega_2 = 1.15$ erweist sich als optimaler Überrelaxationsparameter und letztendlich ω_2-FR3D-W(2,1) als das effektivste CS-Verfahren, wenn man ρ bzw. ρ^∞ *und* den Rechenaufwand berücksichtigt.

Im dreidimensionalen Fall macht es sich zudem bemerkbar, daß für Half Weighting die Lösbarkeit der Gleichungssysteme auf den gröberen Gittern nicht gesichert ist, was dann auch tatsächlich zu einer Verschlechterung der Konvergenzraten führt, die dann je nach Beispiel aber unterschiedlich stark ausfallen kann. Das Erzwingen der Kompatibilität auf den gröberen Gittern hilft nicht, dieses Problem zu beseitigen, da sich die Raten nicht verbessern. HR3D ist also kein sicheres Verfahren, und selbst für W(2,1) ist schon Vorsicht geboten.

Bei den FMG-Verfahren genügt es, FR3D zu verwenden, da es nie schlechtere Ergebnisse als ω_i-FR3D liefert und diese zudem schneller berechnet.

	$\omega = 1$		$\omega = 1.1$		$\omega = 1.15$	
N	ρ^*	ρ^∞	ρ^*	ρ^∞	ρ^*	ρ^∞
32	0.194	0.198	0.097	0.103	0.077	0.103
64	0.197	0.197	0.098	0.104	0.077	0.102

Tabelle 3.6: Vergleich der aus der RFA stammenden Werte ρ^* [64] mit den berechneten ρ^∞. Alle Werte sind für den W(1,1)-Cycle angegeben.

	r	N	ω_1-FR3D	ω_2-FR3D	FR3D	HR3D
W(2,1)	1	64	2.3E-3	2.2E-3	2.4E-3	1.9E-3
		128	6.2E-4	5.7E-4	6.0E-4	4.7E-4
	2	64	2.4E-3	2.4E-3	2.4E-3	2.4E-3
		128	6.0E-4	6.0E-4	6.0E-4	6.1E-4
W(1,1)	1	64	1.6E-3	2.9E-3	2.4E-3	2.6E-3
		128	1.4E-3	2.6E-3	6.0E-4	6.6E-4
	2	64	2.4E-3	2.4E-3	2.4E-3	2.5E-3
		128	6.0E-4	6.1E-4	6.0E-4	6.4E-4
V(2,1)	1	64	5.4E-2	6.7E-2	3.6E-3	1.8E-3
		128			9.3E-4	5.0E-4
	2	64	2.0E-2	3.5E-2	2.5E-3	2.5E-3
		128			6.2E-4	6.4E-4
V(1,1)	2	64	9.3E-2	1.8E-1	2.7E-3	5.9E-2
		128			6.9E-4	

Tabelle 3.7: FMG-Test für Beispiel (2). Angegeben ist jeweils $||u_h^{FMG} - u||_\infty$ nach FMG(Z,1). Es sind $||u_{1/64} - u||_\infty = 2.4E-3$ und $||u_{1/128} - u||_\infty = 6.0E-4$. Die Anpassung spielt (in diesem Beispiel) keine Rolle.

Eine komplette Anpassung ist in den CS-Verfahren nur nötig, falls eine Residuenreduktion um mehr als 15 Zehnerpotenzen gewünscht wird oder nötig ist (bei Berechnung der ρ^∞ etwa). Den FMG-Verfahren reicht die grobe Anpassung, die den Rechenaufwand vernachlässigbar wenig heraufsetzt.

Abschließend ist also zu sagen, daß zur Lösung der zwei- oder dreidimensionalen Poissongleichung auf einem quadratischen Gebiet mit periodischen Randbedingungen mit Erfolg CS- und FMG-Methoden eingesetzt werden können, die bei geeigneter Komponentenwahl hinsichtlich der Konvergenzraten, des Rechenaufwandes und der Paralelisierbarkeit Verfahren für Dirichlet-Randbedingungen entsprechen. Dabei muß aber beachtet werden, daß anders als bei einem Problem mit reinen Dirichlet-Randbedingungen die Periodizität zu singulären Systemen auf allen Gittern führt, deren Lösbarkeit jeweils erfüllt sein muß. Dies schränkt die Auswahl an Restriktionsverfahren ein, so daß sich etwa Injection oder Half Weighting als nicht verwendbar bzw. zumindest nicht sicher erweisen.

Kapitel 4

Berechnung elektrostatischer Größen

4.1 Einleitung

Die Kenntnis der elektrostatischen Energie und des Kraftfeldes eines makroskopischen, periodischen Systems von Teilchen spielt in der Molekulardynamik besonders für Biomoleküle eine große Rolle. Klassische Modelle zerlegen die elektrostatische Energie E meist in eine Summe aus fünf Bestandteilen: die Beiträge der Bindungslängen, der Valenz- und Torsionswinkel, der effektiven Ladungen (Coulomb-Wechselwirkungen) und der van-der-Waalsschen Wechselwirkungen. Durch Bestimmung des negativen Gradienten von E erhält man das Kraftfeld F_i für jedes Teilchen i, womit wiederum in zeitabhängigen Simulationen aus den klassischen Bewegungsgleichungen die (neue) Position jedes Teilchen sowie seine Geschwindigkeit berechnet werden.

Die letzten beiden Summanden in der Beschreibung von E, d.h. die Beiträge der Coulomb-Wechselwirkungen und der van-der-Waalsschen Wechselwirkungen, sind für den größten Teil, nämlich etwa 90%, der totalen Rechenzeit verantwortlich. Der Beitrag der Coulomb-Wechselwirkungen soll in den folgenden Betrachtungen im Vordergrund stehen. Insbesondere wird er im folgenden allein als elektrostatische Energie des periodischen Systems bezeichnet.

Die Periodizität eines Systems, das man in der Praxis antrifft, bedeutet dabei, daß man es mit einer Grundstruktur zu tun hat, die sich in den drei Raumrichtungen in jeweils gleichen Abständen mehrfach wiederholt, bis sie an den Rand des Systems stößt. Dieser weist jedoch im allgemeinen eine unregelmäßige (bzw. „abgebrochene“) Struktur auf. Die bekanntesten Beispiele für solche Systeme sind Ionenkristallgitter.

Da man die elektrostatische Energie und das Kraftfeld als „innere“ Materialeigenschaften ansehen kann, werden oft in Simulationen die Einflüsse des Randes erst einmal nicht betrachtet, sondern lediglich die innere Struktur des Systems. Aus diesem Grunde untersucht man dann *unendliche* periodische Systeme. Weiterhin nimmt man oft an, daß die Partikel im System als „Punktladungen“ gesehen werden können; d.h. die Masse oder das Volumen

eines Teilchens gehen in die Betrachtungen (von E und F_i) nicht ein, weil sonst die Modellierung zu kompliziert wird.

Um allerdings zu realistischeren Ergebnissen zu kommen, müssen natürlich auch Systeme ohne unendlich periodische Fortsetzung, d.h. mit Rand, betrachtet sowie außerdem die anderen vier Beiträge zur elektrostatischen Energie und die Zeitabhängigkeit (s.o.) miteinbezogen werden. Dies soll allerdings hier, wie schon erwähnt, nicht geschehen.

In den folgenden Ausführungen und Untersuchungen wird davon ausgegangen, daß ein unendlich periodisches System mit einen würfelförmigen Aufbau vorliegt. Betrachtet wird also folgende periodische Verteilung von λ Punktladungen im $\mathbb{R}^3$:

$$Q := \{(p_i + n,\, q_i) |\, \forall\, i = 1, \ldots, \lambda \,:\, p_i \in]0, d]^3,\, q_i \in \mathbb{R}_+,\, n \in \mathcal{N}(d)\}$$

mit einer Würfelkantenlänge $d \in \mathbb{R}_+$ und

$$\mathcal{N}(d) = \{(n_1 d, n_2 d, n_3 d) \mid n_1, n_2, n_3 \in \mathbb{Z}\}\,.$$

Hierbei meint q_i die Größe der Ladung, die im Punkt $p_i + n$ lokalisiert ist. Verwendet man charakteristische Funktionen[1] χ_p, so kann man die Verteilung der Punktladungen im dreidimensionalen Raum auch durch folgende Funktion q ausdrücken:

$$\begin{aligned} q : \mathbb{R}^3 &\to \mathbb{R}_+ \\ x &\mapsto \sum_{n \in \mathcal{N}(d)} \sum_{i=1}^{\lambda} q_i \chi_{p_i + n}(x)\,. \end{aligned}$$

Als Einheitszelle von Q wird dann die folgende Menge Q_1 bezeichnet:

$$Q_1 := \{(p_i,\, q_i) |\, i = 1, \ldots, \lambda\}\,.$$

Wie viele Teilchen Q_1 formal enthält, hängt im allgemeinen davon ab, wie die fiktiven Würfel der Kantenlänge d um die Partikel des Systems geschnitten sind. Die Abbildungen 4.1 (a)-(c) zeigen verschiedene Möglichkeiten, ein periodisches System (2D) in Würfel (bzw. hier Quadrate) zu zerlegen. Für die verschiedenen Q_1 gilt:

(a) $Q_1 = \{(\frac{1}{2}, \frac{1}{2}; -1), (0, 0; +\frac{1}{4}), (0, 1; +\frac{1}{4}), (1, 0; +\frac{1}{4}), (1, 1; +\frac{1}{4})\}$ mit $d = 1$,

(b) $Q_1 = \{(\frac{3}{4}, \frac{1}{2}; -1), (\frac{1}{4}, \frac{3}{4}; +1)\}$ mit $d = 1$,

(c) $Q_1 = \{(\frac{1}{4}\sqrt{2}, \frac{1}{4}\sqrt{2}; -1), (\frac{3}{4}\sqrt{2}, \frac{3}{4}\sqrt{2}; -1), (\frac{1}{4}\sqrt{2}, \frac{3}{4}\sqrt{2}; +1), (\frac{3}{4}\sqrt{2}, \frac{1}{4}\sqrt{2}; +1)\}$ mit $d = \sqrt{2}$.

Um den Rechenaufwand möglichst klein zu halten, sollten zum Beispiel Ladungsteilungen wie in (a) möglichst weitgehend vermieden werden.

Das System Q soll elektrisch neutral sein. Dies ist aufgrund der Periodizität gleichbedeutend mit der elektrischen Neutralität seiner Einheitszelle:

$$\sum_{i=1}^{\lambda} q_i = 0\,.$$

[1] definiert durch $\chi_p(x) = 0$ für $p \neq x$, $\chi_p(p) = 1$.

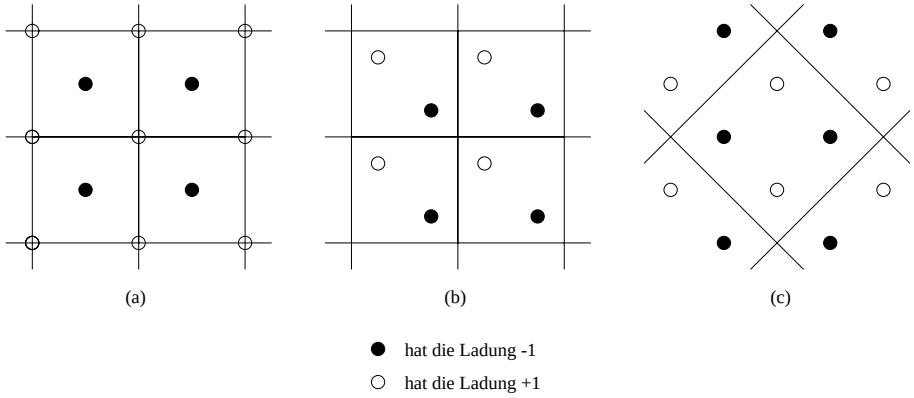


Abbildung 4.1: Verschiedene Q für ein periodisches System von Partikeln. Erläuterungen finden sich im Text.

Das elektrostatische Potential P des Systems Q ist für Orte $x \notin Q$ gegeben durch

$$P(x) = \sum_{n \in \mathcal{N}(d)} \sum_{j=1}^{\lambda} \frac{q_j}{|x - p_j + n|} , \tag{4.1}$$

wobei $|\cdot|$ immer die euklidische Norm bezeichne. Dieses Potential gibt an, welche Arbeit notwendig ist, um eine positive Einheitsladung aus dem Unendlichen an den Ort $x \notin Q$ zu bringen (siehe [22]). Definiert man noch für $i = 1, \ldots, \lambda$

$$\tilde{P}(p_i) = \left(\sum_{j \neq i : j=1}^{\lambda} \frac{q_j}{|p_i - p_j|} + \sum_{n \in \mathcal{N}(d) \setminus \{0\}} \sum_{j=1}^{\lambda} \frac{q_j}{|p_i - p_j + n|} \right) ,$$

kann man (letztlich mit Hilfe von P) nun die im ganzen System Q gespeicherte elektrostatische Energie E angeben:

$$\begin{aligned} E(p_1, \ldots, p_\lambda) &= \frac{1}{2} \sum_{i=1}^{\lambda} q_i \tilde{P}(p_i) \\ &= \frac{1}{2} \left(\sum_{i \neq j : i,j=1}^{\lambda} \frac{q_i q_j}{|p_i - p_j|} + \sum_{n \in \mathcal{N}(d) \setminus \{0\}} \sum_{i,j=1}^{\lambda} \frac{q_i q_j}{|p_i - p_j + n|} \right) . \end{aligned} \tag{4.2}$$

Das auf das Teilchen i wirkende Kraftfeld ist dann

$$F_i = -\nabla_{p_i} E(p_1, \ldots, p_\lambda) . \tag{4.3}$$

Die erste Summe in (4.2) beschreibt dabei die Wechselwirkungen innerhalb der Einheitszelle, die zweite dagegen die Wechselwirkungen zwischen der Einheitszelle und ihren „periodischen Kopien“. Problematisch an dieser „Definition“ ist, daß obiger Summenwert von der Reihenfolge der Summation abhängt, die Reihe somit nicht (absolut) konvergiert.

Wollte man diesem Problem aus dem Wege gehen, indem man doch ein System aus endlich vielen Zellen und einem Randbereich betrachten würde, hätte man es mit einer sehr komplizierten Formel zu tun, da man das Verhältnis jeder Zelle (bzw. jedes Teilchens) zum Rand betrachten müßte. Wie oben aber bereits erwähnt wurde, sollen Randeffekte hier gerade nicht betrachtet werden.

Um zu einer eindeutigen und physikalisch sinnvollen Definition der elektrostatischen Energie zu gelangen, muß man also die Reihenfolge der Summation in (4.2) festlegen. Per Konvention geschieht dies dadurch, daß man das unendliche periodische System in (ungefähr) kugelförmigen Schichten um die Einheitszelle aufbaut.

Bei der *direkten* numerischen Berechnung der Reihe in (4.2) tritt das Problem auf, daß diese Reihe nur sehr langsam konvergiert. Um eine gute Approximation von E zu erhalten, ist der Rechenaufwand entsprechend groß. Würde man also hier bereits ein Cut-Off-Schema oder die Minimum Image Convention einsetzen (analog den in Abschnitt 4.3.2 beschriebenen Vorgehensweisen, aber für die Berechnung der komplette Summe), wäre die Rechenzeit zwar kürzer, aber dafür der Approximationfehler zu groß: Schon bei einem fast „statischen“ System führt dies zu einem künstlichen Verhalten, und erst recht dann bei Langzeit-Simulationen, wo sich die Fehler dieser Nichtbeachtung der weitreichenden Wechselwirkungen stark summieren können.

Im noch folgenden Kapitel 5 wird ein neuer Algorithmus zur Berechnung der elektrostatischen Energie und des Kraftfeldes vorgestellt und getestet. Er verwendet eines der in Kapitel 3 entwickelten Mehrgitter-Verfahren für die dreidimensionale Poissongleichung mit periodischen Randbedingungen. Diesen neuen Algorithmus kann man in eine Klasse von Verfahren einordnen, die (verbesserte) Varianten der Ewald-Summation durchführen. Daher wird zuerst ein Überblick über das Konzept der Ewald-Summation gegeben, um anschließend kurz einige Standard-Methoden und neuere Verfahren, die Fast-Fourier-Transformationen durchführen, vorzustellen. Als direkte Konkurrenz zu diesen Algorithmen erweisen sich hierarchische Verfahren, die Multipole-Entwicklungen verwenden. Diese Klasse soll hier ebenfalls vorgestellt werden.

4.2 Die Ewald-Summation (ES)

4.2.1 Ewalds Idee

Die Ewald-Summation (ES) [22] wurde bereits 1921 einführt. Ewalds Idee war es, die nur langsam konvergierende Reihe (4.2) in eine Summe aus zwei schnell konvergierenden Reihen und einem konstanten Term zu zerlegen:

$$E(p_1, \ldots, p_\lambda) = E_r + E_m + E_o \,, \tag{4.4}$$

wobei die Reihe für den „direkten Raum“ E_r (im folgenden „direkte Summe“ genannt), die reziproke (oder Fourier-)Reihe E_m und der konstante Term

(self term) E_o folgende Gestalt besitzen:

$$E_r = \frac{1}{2} \sum_{j \neq k : j,k=1}^{\lambda} \sum_{n} q_j q_k \frac{erfc(\alpha |p_j - p_k + n|)}{|p_j - p_k + n|} ,$$

$$E_m = \frac{1}{2\pi V} \sum_{j,k=1}^{\lambda} q_j q_k \sum_{m \neq 0} Re\left(\frac{\exp\left(-(\pi m/\alpha)^2 + 2\pi i < m, p_j - p_k >\right)}{m^2} \right) ,$$

$$E_o = -\frac{\alpha}{\sqrt{\pi}} \sum_{j=1}^{\lambda} q_j^2 .$$

Hierbei bezeichnen das nicht als Index auftretende i die imaginäre Einheit, Re den Realteil einer komplexen Zahl, $< \cdot, \cdot >$ das Standardskalarprodukt im $\mathbb{R}^3$, V das Volumen der Simulationsbox (hier $V = d^3$), $m = (m_1, m_2, m_3)$ die zu den $n \in \mathcal{N}(d)$ reziproken[2] Vektoren und *erfc* die komplementäre Fehlerfunktion:

$$erfc(t) := 1 - erf(t) = 1 - \frac{2}{\sqrt{\pi}} \int_0^t \exp\left(-u^2\right) du .$$

Da die Basis des hier betrachteten Koordinatensystems die Standardeinheitsbasis ist und diese reziprok zu sich selbst bzgl. $< \cdot, \cdot >$ ist - sie bildet nämlich eine Orthonormalbasis - , durchlaufen die m ebenfalls $\mathcal{N}(d)$ (siehe [22]).

Der konstante Term E_o ist ein Korrekturterm, der die Wechselwirkungen der künstlich eingeführten „Gegenladungen“ (siehe Abschnitt 4.2.2) jeweils mit sich selbst wieder auslöscht.

In der Herleitung [22] dieser streng gültigen Formel erweist sich der positive Parameter α als die Trennungsstelle einer Integration: Rechts und links von α werden dann unterschiedliche, äquivalente Darstellungen benutzt, um zur obigen Formel zu gelangen, die günstiger als die ursprüngliche (4.2) ist. Der Wert der Summe ist aber unabhängig von α.

Die Ewald-Formel für das Kraftfeld kann aus obiger Formel (4.4) durch Differentiation der direkten und der reziproken Summe erhalten werden. E_o fällt hierbei weg, da es konstant ist.

Zur Theorie der Ewald-Summation vergleiche auch [39].

4.2.2 Eine physikalische Interpretation

Die Spaltung von E in der Ewald-Summation läßt sich folgendermaßen veranschaulichen: Jede Punktladung denkt man sich umgeben von einer Gaußschen Ladungsverteilung derselben Größe und gegensätzlichem Vorzeichen (siehe Abbildung 4.2), wobei die folgende Ladungsdichtefunktion [2] verwendet wird:

$$\rho_j(r) = q_j \left(\frac{\alpha}{\sqrt{\pi}}\right)^3 \exp\left(-\alpha^2 r^2\right) .$$

[2]Sind a_1, a_2, a_3 drei Vektoren im $\mathbb{R}^3$, so sind die zu ihnen reziproken Vektoren b_1, b_2, b_3 durch die folgenden Gleichungen bestimmt: $< a_i, b_j >= \delta_{ij}$ für $i, j = 1, 2, 3$. Dabei bezeichne δ_{ij} das Kronecker-Symbol.

Dabei gibt α die Weite der Verteilung und r die relative Position zum Zentrum p_i der Verteilung an. Diese eingeführte Ladungsverteilung schirmt die Wechselwirkungen zwischen den Punktladungen soweit ab, daß sie auf einen kleinen Bereich beschränkt werden. Dies hat die schnelle Konvergenz von E_r zur Folge. Um die künstlich eingeführten ρ_j zu kompensieren, wird für jeden Punkt p_i eine zweite Gaußsche Verteilung derselben Größe und mit demselben Vorzeichen wie q_i hinzugefügt. Die daraus entstehende Summe wird im reziproken Raum berechnet, wobei Fouriertransformationen zur Lösung der sich ergebenden Poissongleichung eingesetzt werden (siehe [22]). Statt der Gaußschen Ladungsverteilung können auch andere Funktionen verwendet werden (siehe z.B. Abschnitt 4.4.1 oder [35]).

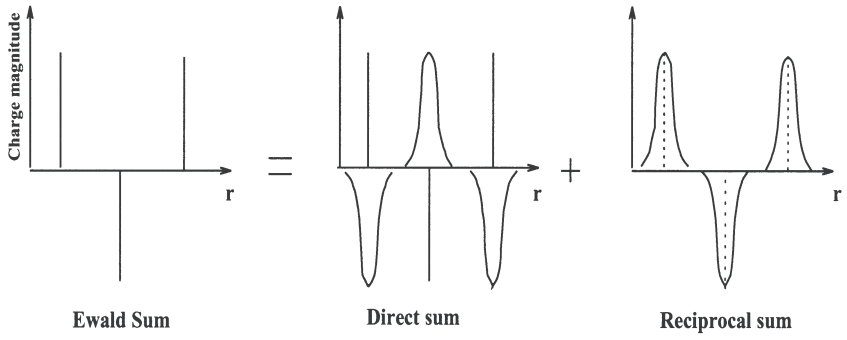


Abbildung 4.2: Zur physikalischen Interpretation. Die Abbildung entstammt [59].

4.2.3 Numerische Berechnung

Möchte man nun die Ewald-Summation (4.4) numerisch durchführen, muß man für eine Reihe von Parametern Werte festsetzen und so aufeinander abstimmen, daß die Summation schnell abläuft und doch der Fehler möglichst klein bleibt. Drei Parameter bestimmen die Effizienz der ES:

- n_{max} gibt die maximale Anzahl der für die Summation verwendeten Vektoren n an und bestimmt so den Aufwand für die direkte Summe E_r.
- Analog dazu gibt m_{max} die maximale Anzahl der Vektoren m in E_m an.
- Der Ewald-Konvergenzfaktor α bestimmt die relative Konvergenzrate zwischen der direkten und der reziproken Summe: Ein großes α bewirkt eine enge Gaußverteilung und somit eine schnelle Konvergenz der direkten Summe ($\lim_{\alpha\to\infty} erfc(\alpha|t|) = 0$). Dann genügt also ein kleines n_{max}. Andererseits sorgt ein kleines α für die schnelle Konvergenz der reziproken Summe ($\lim_{\alpha\to 0} \exp\left(-(t/\alpha)^2\right) = 0$). Dann ist ein kleines m_{max} ausreichend.

Wie man z.B. in den Abschnitten 4.3.1 und 4.4 sieht, kann die reziproke Summe so umgeformt werden, daß ihre Auswertung deutlich effizienter als

die der direkten Summe ist. Daher wird in der Praxis α so groß gewählt, daß der Aufwand für die direkte Summe minimiert wird (siehe auch Abschnitt 4.3.2), und somit die „Hauptlast“ (d.h. ein großes m_{max}) auf der reziproken Summe liegt. Eine Vielzahl von Arbeiten haben sich damit beschäftigt, durch eine geeignete Wahl der Parameter einen Kompromiß zwischen Genauigkeit und Geschwindigkeit des Verfahrens zu finden. Dabei entstanden eine Reihe von Abschätzungen für die „richtige“ Größe von α abhängig von der Systemgröße und der gewünschten Genauigkeit (siehe z.B. [41, 45, 49]).

4.3 Standard-ES-Methoden

4.3.1 Verbesserte ES

Durch einige Umformungen können die Reihen in (4.4) effizienter berechnet werden (vergleiche [50]). Beispielsweise kann man die Reihe für E_m folgendermaßen umformen:

$$E_m = \frac{1}{2\pi V} \sum_{m \neq 0} \frac{\exp\left(-(\pi m/\alpha)^2\right)}{m^2} \sum_{j,k=1}^{\lambda} q_j q_k Re\Big(\exp\left(2\pi i < m, p_j - p_k >\right)\Big) .$$

Nun ist aber

$$\begin{aligned}
&\sum_{j,k=1}^{\lambda} q_j q_k Re\Big(\exp\left(2\pi i < m, p_j - p_k >\right)\Big) \\
&= \sum_{j,k=1}^{\lambda} q_j q_k \cos\left(2\pi < m, p_j - p_k >\right) \\
&= \sum_{j,k=1}^{\lambda} q_j q_k \Big(\cos\left(2\pi < m, p_j >\right) \cos\left(-2\pi < m, p_k >\right) \\
&\qquad - \sin\left(2\pi < m, p_j >\right) \sin\left(-2\pi < m, p_k >\right)\Big) \\
&= \Big(\sum_{j=1}^{\lambda} q_j \cos\left(2\pi < m, p_j >\right)\Big)^2 + \Big(\sum_{j=1}^{\lambda} q_j \sin\left(2\pi < m, p_j >\right)\Big)^2 \\
&= |S(m)|^2
\end{aligned}$$

mit dem „Strukturfaktor“

$$S(m) := \sum_{k=1}^{\lambda} q_k \exp\left(2\pi i < m, p_k >\right) .$$

Folglich läßt sich E_m schreiben als

$$E_m = \frac{1}{2\pi V} \sum_{m \neq 0} \frac{\exp\left(-(\pi m/\alpha)^2\right)}{m^2} |S(m)|^2 . \tag{4.5}$$

Auf diese Weise wird der Summationsaufwand von $O(\lambda^2)$ (bzw. $O(\lambda^2 m_{max})$) auf $O(\lambda)$ (bzw. $O(\lambda m_{max})$) gesenkt.

4.3.2 Truncation

Um den Aufwand zur Berechnung der direkten Summe der ES (d.h. den Parameter n_{max}) möglichst klein zu halten, kann man ein großes α wählen (siehe Abschnitt 4.2.3). Dies hat nämlich den Vorteil, daß man nun einen sphärischen Cut-Off-Radius R_{cutoff} oder die Minimum Image Convention (MIC) einsetzen kann. Im Gegensatz zum Einsatz solcher Methoden bei der Berechnung (der ganzen Summe) von (4.2) führt dies nämlich dank der schnellen Konvergenz der direkten Summe E_r (für großes α) **nicht** zu großen Fehlern.

Bei der Minimum Image Convention werden für jedes Teilchen i in der Summe nur die Wechselwirkungen herangezogen, die es mit seinen genau $\lambda - 1$ nächsten Nachbarn eingeht. Diese sind gerade in einem Würfel mit Kantenlänge d und Zentrum p_i enthalten.

Benutzt man einen Radius R_{cutoff}, der üblicherweise zwischen 8 und 12 Å liegt, betrachtet man für E_r nur die Wechselwirkungen des Teilchens i ($i = 1, \ldots, \lambda$) mit den Teilchen, die maximal R_{cutoff} von i entfernt sind:

$$E_r = \frac{1}{2} \sum_{i,j,n:|p_i - p_j + n| \leq R_{cutoff}} q_i q_j \frac{erfc(\alpha |p_i - p_j + n|)}{|p_i - p_j + n|} .$$

Für $R_{cutoff} \leq \frac{d}{2}$ werden also für jedes Teilchen i höchstens $\lambda - 1$ Wechselwirkungen betrachtet, da eine Sphäre um p_i mit dem Radius R_{cutoff} dann in einem Würfel mit Zentrum p_i der Kantenlänge d enthalten ist (siehe auch Abbildung 4.3). Somit ist in diesem Fall die Gesamtzahl der Interaktionen beim Cut-Off kleiner gleich der Gesamtzahl bei der MIC. Letztere beträgt $\frac{1}{2}\lambda(\lambda - 1)$. Für große Systeme, $\lambda > 10^4$, ist die MIC also immer noch sehr aufwendig, weswegen in der Praxis dann ein $R_{cutoff} < \frac{d}{2}$ und ein großes α verwendet werden: Durch die Wahl eines (eventuell sehr) kleinen R_{cutoff} kann man die Komplexität der direkten Summe auf $O(\lambda)$ senken.

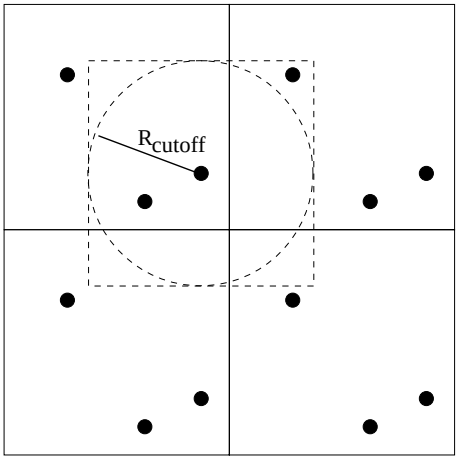


Abbildung 4.3: MIC und $R_{cutoff} = d/2$ im Vergleich. Bei der MIC werden hier für das ausgewählte Teilchen Wechselwirkungen mit zwei weiteren, für den Cut-Off aber nur mit einem weiteren Partikel berechnet.

4.3.3 Vernachlässigung des reziproken Raumes

In [49] haben Rycerz und Jacobs für ganz spezielle Beispiele (z.B. für bestimmte Bi_2O_3- und $NaCl$-Systeme) beobachtet, daß man bei geeigneter Wahl der Parameter komplett auf die Berechnung der reziproken Summe verzichten kann, da der Anteil von E_m an E dort sehr klein ist (z.B. nur etwa $\frac{1}{1500}$). Diese systemabhängige Methode sollte allerdings nicht allgemein eingesetzt werden, da nicht klar ist, ob man einen allgemeineren Anwendungsbereich überhaupt angeben kann. Für kleine Systeme etwa erweist sich dieses Verfahren oft als nicht geeignet (vergleiche [49]).

4.3.4 Tabulationsmethoden

Um die Geschwindigkeit eines Verfahrens zur Berechnung von E und F_i zu beschleunigen, kann es sinnvoll sein, Werte für E bzw. F_i zu tabulieren und daraus durch Interpolation weitere Werte zu gewinnen. Üblicherweise werden dazu die elektrostatische Energie bzw. das Kraftfeld von Systemen mit $Q_1 = \{(p_1, q_1), (p_2, q_2)\}$ berechnet, wobei $p_1 = (0,0,0)$ ist und die $p_2 = (p_{21}, p_{22}, p_{23})$ ein $[0, \frac{d}{2}]^3$ durchziehendes Gitter der Maschenweite a beschreiben. Es kommt nur auf den Abstand(svektor) der beiden Punkte bzw. ihrer zueinander am nächsten liegenden Kopien und nicht auf ihre genaue Lage an. Die so gewonnenen Werte werden dann (z.B.) als Funktion von $r_{12} = p_1 - p_2$ tabuliert.

Es genügt, für die p_2 nur die Punkte des Gitters zu nehmen, die die folgende Bedingung erfüllen (siehe [50]):

$$\frac{d}{2} \geq p_{21} \geq p_{22} \geq p_{23} \,.$$

Die Gültigkeit dieser Ungleichung kann durch eine Transformation (Verschiebung, Drehung bzw. Spiegelung der Einheitszelle) eines (periodischen) Systems mit zwei Punktladungen in Q_1 erreicht werden (vergleiche auch Abbildung 4.1). Zur Erhöhung der Genauigkeit werden bei Sangester [50] für die Tabulation allerdings die Summanden, die die Wechselwirkungen zwischen den beiden zueinander am nächsten liegenden Teilchen angeben, ausgelassen. Sie werden erst während der Simulation des jeweils zu untersuchenden Systems berechnet. Diese Methode reduziert den Speicherplatzbedarf, erhöht aber gleichzeitig dank der Transformationen den Rechenaufwand.

Die Maschenweite a und die Ordnung der Interpolation[3] bestimmen ebenfalls die Genauigkeit der interpolierten Werte, gleichzeitig aber auch die Effizienz des Verfahrens: Hier muß ein Kompromiß zwischen der Genauigkeit, der Geschwindigkeit der Berechnung und dem Speicherplatzbedarf gefunden werden.

Verglichen mit der Standard-Ewald-Summation ist der Rechenaufwand eines solchen Verfahrens immer noch $O(\lambda^2)$. Man kann alternativ auch nur Werte für die direkte Summe tabulieren, falls die reziproke Summe effizienter berechnet werden kann. Kann man nämlich die Ewald-Parameter geeignet

[3] Sangester [50] z.B. verwendet die trilineare Interpolation. Hier kann natürlich auch eine andere benutzt werden.

wählen, beträgt der Aufwand für die direkte Summe $O(\lambda)$ (siehe Abschnitt 4.3.2). Belhadj et al. [10] haben so in ihren Tests eine vierfache Beschleunigung festgestellt.

Es können auch Werte für $exp(\cdot)$ oder $erfc(\cdot)$ tabuliert werden, falls ihre Berechnung auf speziellen Computern zu lange dauert. Weiterhin liefert eine Spline-Interpolation [23] gute Näherungen für $erfc(x)$ und ihre Ableitung $-\frac{2}{\sqrt{\pi}} \exp(-x^2)$.

Fazit: Tabulationsmethoden können Verfahren beschleunigen, falls eine mittlere Genauigkeit der Werte ausreicht. Ansonsten werden sie zu teuer bzw. übersteigen den vorhandenen Speicherplatz.

4.3.5 Polynom-Approximations-Methoden

Statt Tabulation zu verwenden, ist es auch möglich, ein Polynom $P(x)$ einzusetzen, das E (in der Form (4.4)) annähert, aber billiger auszuwerten ist. Zur Berechnung des Kraftfeldes wird es dann analytisch abgeleitet.

Verschiedene Polynom-Approximationen sind bereits vorgeschlagen worden: von einer einfachen von Brush et al. [15] bis zu einer genaueren mit kubisch harmonischen Funktionen von Von der Lage und Bethe [42].

Man kann $P(x)$ auch aufteilen in einen isotropischen und einen kubisch symmetrischen Teil und dann exakte kubisch harmonische Funktionen verwenden (anisotropische Approximation von Hansen [31] für Monte-Carlo-Simulationen). Der hierbei beobachtete Fehler für E lag unter $0,1\%$. Slattery et al. [56] haben in einem ähnlichen Verfahren Poissongleichungen mit kubischer Symmetrie verwendet.

In [1] haben Adams und Dubey verschiedene Approximationen für Ladung-Ladung-, Ladung-Dipol- bzw. Dipol-Dipol-Systeme getestet. Beispielsweise wurde $P(x)$ in eine Reihe entwickelt, die zur Berechnung der Koeffizienten mit kubisch symmetrischen Funktionen angenähert wurde. Dies lieferte

$$P_l(x) = \frac{1}{|x|} + S + A_2|x|^2 + \sum_{n=4,6}^{l} \left(A_n K h_n(x) + B_n K h_n'(x)\right) ,$$

wobei $4 \leq l \leq 20$ und l gerade ist, B_n für $n \leq 10$ verschwindet und S den „self term “ bezeichnet. Die A_n und B_n sind in [1] tabuliert. Für $l = 6$ entstand ein relativer[4] Fehler von $4 \cdot 10^{-3}$ und für $l = 14$ ein Fehler von $2,5 \cdot 10^{-5}$. Allerdings haben Toukmaji und Board [59] durch Tests gezeigt, daß dieses Verfahren teuer wird, wenn man eine mittlere bis hohe Genauigkeit verlangt.

Falls Tabulation zu teuer ist, kann man auch für $erfc(\cdot)$ ein schnell auswertbares Polynom verwenden.

Zusammenfassend ergibt sich wie schon bei der Tabulation: Approximation ist nur effizient, falls keine hohe Genauigkeit verlangt wird.

[4] Der relative Fehler für Werte $w_1, \ldots, w_n$ ist definiert als $\sqrt{\sum_{i=1}^{n}(w_i - \tilde{w}_i)^2 / \sum_{i=1}^{n} w_i^2}$, wobei die w_i die exakten und die $\tilde{w}_i$ die approximativ berechneten Werte darstellen.

4.3.6 Ein $O(\lambda^{\frac{3}{2}})$-Algorithmus

Ohne Approximationen zu verwenden, haben Perram et al. [45] in ihrem Verfahren die Komplexität der ES verringert, und zwar durch die „linked-cell spatial decomposition"-Technik von Hockney und Eastwood [36]. Durch eine geeignete Spaltung der Potential-Auswertung in einen Teil für die Wechselwirkungen mit kurzer Reichweite (Komplexität $O(\lambda^2)$) und einen Teil für die weitreichenden Wechselwirkungen (Komplexität $O(\lambda)$) kann die Komplexität der Summation auf $O(\lambda^{\frac{3}{2}})$ gesenkt werden. Ein weiterer Beweis für dieses Ergebnis findet sich in [24].

4.3.7 Zusammenfassung

Von den bisher vorgestellten, klassischen Verfahren stellt der $O(\lambda^{\frac{3}{2}})$-Algorithmus von Perram wohl die beste Kombination aus Geschwindigkeit und Genauigkeit dar (vergleiche [59]). Eine Vielzahl von Molekulardynamik-Codes verwenden allerdings Approximationen. Für große Systeme ist aber auch bei Perrams Algorithmus der Rechenaufwand noch zu groß. Aus diesem Grund sollen in den folgenden Abschnitten Verfahren diskutiert werden, die die Komplexität der Berechnung nochmals verringern.

4.4 Auf FFT basierende ES-Methoden

In diesem Abschnitt werden einige Methoden vorgestellt, die die Komplexität der ES auf $O(\lambda log(\lambda))$ senken, indem sie die klassische E_m oder eine ähnliche reziproke Summe effizienter durch dreidimensionale Fast-Fourier-Transformationstechniken (FFT) berechnen.

4.4.1 Particle-Particle Particle-Mesh (P^3M)

Diese Klasse von Verfahren wurde von Hockney und Eastwood [36] entwickelt und von Luty et al. [44] und Rajagopal und Needs [47] erweitert. Das elektrostatische Potential, hervorgerufen durch weitreichende Wechselwirkungen zwischen den Teilchen, wird ähnlich der ES in eine Summe aus zwei Komponenten gespalten: Der erste Teil (Short-Range-Potential) verschwindet außerhalb eines Cut-Off-Radius R_{cutoff} und repräsentiert die Kräfte mit kurzer Reichweite; der zweite Teil, die „Referenzkraft" (Long-Range-Potential), steht für die Kräfte mit einem großen Wirkungsradius, ist glatt und läßt sich daher auf einem Gitter approximieren.

Die Analogie zur direkten bzw. reziproken Summe der klassischen ES ist gut zu erkennen. Aber anders als in der ES wird in [44] nicht die Gauß-Verteilung, sondern die P^3M-Standard-Ladungsdistribution (S2-Funktion) verwendet, eine Sphäre mit „einheitlich fallender Dichte". Diese S2-Funktion s ist gegeben durch:

$$s(x) = \begin{cases} \frac{48}{\pi a^4}(\frac{a}{2} - |x|) & \text{für } |x| < \frac{a}{2}, \\ 0 & \text{sonst,} \end{cases}$$

wobei a die Breite der Verteilung bestimmt. Das Short-Range-Potential ψ_{short} zweier Teilchen i und j kann dann mit einem Cut-Off-Radius $R_{cutoff} \approx 0.7a$ folgendermaßen berechnet werden:

$$\psi_{short}(\zeta_{ij}) = \frac{1}{4\pi\epsilon_0}\left(\frac{1}{|p_i - p_j|} - \frac{1}{70a}\sum_{n=-1}^{7} C_n \zeta_{ij}^n\right) \text{ für } 0 \leq \zeta_{ij} < 2$$

mit $\zeta_{ij} = \frac{2|p_i - p_j|}{a}$ und

$$(C_{-1}, \ldots, C_7) = \begin{cases} (0, 208, 0, -112, 0, 56, -14, -8, 3) & \text{für } 0 \leq \zeta_{ij} \leq 1, \\ (12, 128, 224, -448, 280, -56, -14, 8, -1) & \text{für } 1 < \zeta_{ij} < 2. \end{cases}$$

Zur Berechnung des Long-Range-Potentials ψ_{long} geht man so vor:

1. Die Gesamtladungsverteilungsfunktion V_{ges} für ein dreidimensionales Gitter, das die Einheitszelle ausfüllt, muß bestimmt werden. Dazu wird zuerst jeder Ladungspunkt auf die umliegenden Gitterpunkte mit einem Wichtungsverfahren - der Inversen einer Interpolation - verteilt, wofür viele Möglichkeiten existieren, z.B. die „triangle-shaped cloud“ (TSC) (siehe [36]). Möglichst wenige Gitterpunkte sollen dabei miteinbezogen werden, außerdem soll die Verteilung eine möglichst glatte Funktion der Position des Ladungspunktes sein. Dies bedeutet, daß die Maschenweite des Gitters passend gewählt sein muß, was wiederum den Rechenaufwand bestimmt. Jede Ladung auf einem Gitterpunkt wird dann mit der (Einzel-)Ladungsverteilungsfunktion s auf umliegende Gitterpunkte verteilt.

2. Aus V_{ges} erhält man $\hat{V}_{ges}$ mit Hilfe der Fast-Fourier-Transformation. Im Fourier-Raum wird dann die Transformierte von ψ_{long} berechnet:

   $$\hat{\psi}_{long}(\mathbf{k}) = \hat{V}_{ges}(\mathbf{k})\hat{G}(\mathbf{k}) \, .$$

   Die „Einflußfunktion“ $\hat{G}$ ist üblicherweise durch $\hat{G}(\mathbf{k}) = \frac{\hat{s}(\mathbf{k})}{\epsilon_0 |k|^2}$ bestimmt, kann aber je nach Systemgröße und Art der Ladungsverteilung noch optimiert werden. Die optimierte Funktion ist dann aber nicht generell verwendbar, muß also für ein anderes System neu bestimmt werden. Durch die inverse Fourier-Transformation erhält man letztlich Werte für ψ_{long} in den Gitterpunkten.

3. Eine numerische Differentiation des Potentials liefert das Kraftfeld für Gitterpunkte.

4. Werte für das Potential bzw. das Kraftfeld ursprünglicher Ladungspunkte werden durch dieselbe Interpolation berechnet, die umgekehrt schon im ersten Schritt benutzt wurde.

Dank der FFT ergibt diese Vorgehensweise einen $O(\lambda log(\lambda))$-Algorithmus. Um die Genauigkeit der Berechnungen zu erhöhen, muß man z.B. das Gitter verfeinern oder ein besseres Wichtungs- bzw. Interpolationsschema bzw. ein Differentiationsverfahren höherer Ordnung einsetzen. Auch hier muß also (gewöhnlich experimentell) ein guter Kompromiß zwischen Genauigkeit, Laufzeitkosten und Speicherplatzbedarf gefunden werden.

4.4.2 Particle-Mesh Ewald (PME)

Auch dieses Verfahren [17, 20] wurde durch die P^{3}M-Methode von Hockney und Eastwood [36] angeregt. Doch anders als dort verwendet PME die direkte und reziproke Summe von Ewald und auch die Gaußverteilung der Ewald-Summation.

Zur Auswertung der direkten Summe wird der Ewald-Parameter α so groß gewählt, daß ein Cut-Off-Radius eingesetzt werden kann und die Komplexität der direkten Summe von $O(\lambda^2)$ auf $O(\lambda)$ reduziert wird (siehe Abschnitt 4.3.2).

Zur Berechnung der reziproken Summe verteilt man mit einem Wichtungsverfahren (siehe Abschnitt 5.3.1) die Punktladungen auf ein Gitter der Dimensionen $K_1 \times K_2 \times K_3$, das die Einheitszelle ausfüllt, und erhält so eine Ladungsmatrix M: $M(k_1, k_2, k_3)$ gibt dann die Ladung des Gitterpunktes $(\frac{k_1}{K_1}, \frac{k_2}{K_2}, \frac{k_2}{K_2})$ an $(k_i = 1, \ldots, K_i)$. Mit $\mathcal{F}(M)$ sei die diskrete Fast-Fourier-Transformation von M bezeichnet. Schreibt man nun die Gleichung (4.5) in der Form[5]

$$E_m = \frac{1}{2\pi V} \sum_{m \neq 0} \frac{\exp\left(-(\pi m/\alpha)^2\right)}{m^2} S(m) S(-m)$$

und approximiert den Strukturfaktor S

$$S(m) = \sum_{k_1,k_2,k_3} M(k_1, k_2, k_3) \exp\left(2\pi i \left(m_1 \frac{k_1}{K_1} + m_2 \frac{k_2}{K_2} + m_3 \frac{k_3}{K_3}\right)\right)$$

durch

$$S(m) \approx \tilde{S}(m) = \mathcal{F}(M)(m)\,,$$

so erhält man folgende Approximation für die reziproke Summe E_m:

$$\tilde{E}_m = \frac{1}{2\pi V} \sum_{m \neq 0} \frac{\exp\left(-(\pi m/\alpha)^2\right)}{m^2} \mathcal{F}(M)(m) \mathcal{F}(M)(-m)\,.$$

Nach einigen Umformungen gelangt man dann zu

$$\tilde{E}_m = \frac{1}{2} \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} M(m_1, m_2, m_3)(\psi_{rec} * M)(m_1, m_2, m_3)\,,$$

wobei ψ_{rec} das reziproke Ladungspaar-Potential und * eine Konvolution angibt (siehe z.B. [20, 36]). Um also die reziproke Summe E_m auszuwerten, wird zuerst die Matrix M berechnet und danach durch (inverse) 3D-FFT transformiert (um u.a. die Strukturfaktoren S zu erhalten). Mit der letzten Gleichung wird dann $\tilde{E}_m$ bestimmt, wobei 3D-FFT zur Berechnung der Konvolution $\psi_{rec} * M$ eingesetzt wird.

Die reziproke Summe wird also mit einer dreidimensionalen FFT-Technik berechnet, deren Rechenaufwand proportional zu $\lambda log(\lambda)$ ist. Insgesamt ist PME somit ein $O(\lambda log(\lambda))$-Verfahren.

[5] Für den Strukturfaktor S (siehe Abschnitt 4.3.1) gilt nämlich $|S(m)| = S(m)S(-m)$.

Ursprünglich wurde als Ladungsinterpolationsfunktion im PME-Verfahren die Lagrange-Interpolation [17] verwendet. Ein weiterentwickeltes PME [20] benutzt die B-Spline-Interpolationsfunktion. Sie hat einerseits den Vorteil, auf einfache Weise die Interpolationsordnung und damit die Genauigkeit zu erhöhen, andererseits erlaubt sie durch ihre Glattheit eine analytische Differentiation der direkten und reziproken Summe. Auf diese Weise kann der Ausdruck für das Kraftfeld mit hoher Genauigkeit analytisch ausgewertet und Speicherplatzverbrauch deutlich gesenkt werden.

Desweiteren kann PME so ohne großen weiteren Rechenaufwand eine hohe Genauigkeit, d.h. einen relativen Fehler von etwa 10^{-6} bis 10^{-8}, erzielen. Dies ist auch der Vorteil gegenüber P^3M, welches zwar effizient ist, aber bisher nicht so leicht eine höhere Genauigkeit erreichen kann.

Tests zeigen eine hohe Effizienz dieses Verfahrens: Für hinreichend kleine Systeme ist zwar die konventionelle Ewald-Summation effizienter, aber bereits ab einer Systemgröße von etwa 600 bis 900 Teilchen (je nach Rechner) ist PME [20] schneller. Beispielsweise liegt für etwa 20000 Teilchen der Verschnellerungsfaktor bei ungefähr 60, 176 bzw. 189 (auf einer SGI R4400) für geringe, mittlere bzw. hohe Genauigkeit (siehe [20]).

4.4.3 Fast Fourier Poisson (FFP)

Die Fast-Fourier-Poisson-Methode [66] formt die ES in einer anderen Weise als das PME-Verfahren um, benutzt aber ebenfalls die FFT und besitzt auch die Komplexität $O(\lambda log(\lambda))$. Eine hohe Genauigkeit wird hier ohne Interpolation [17] oder Multipole-Entwicklung [51] (siehe nächsten Abschnitt) erreicht.

Wie in der klassischen Ewald-Summation (siehe Abschnitt 4.2) wird folgende (Gesamt-)Ladungsdichtefunktion ρ_s verwendet, die um jede Punktladung eine sphärische Gaußsche Funktion mit derselben Größe und gegensätzlichem Vorzeichen legt:

$$\rho_s(x) = -\sum_{j=1}^{\lambda} q_j (\frac{\alpha}{\sqrt{\pi}})^3 \exp\left(-\alpha^2 |x - p_j|^2\right) = \sum_{j=1}^{\lambda} \rho_j(|x - p_j|) \; .$$

Die sich daraus ergebende reziproke Summe E_m und das reziproke Potential ϕ_{recip} lauten dann (vergleiche Abschnitt 4.2):

$$\phi_{recip}(p_k) = \frac{1}{\pi V} \sum_{j=1}^{\lambda} q_j \sum_{m \neq 0} \frac{\exp\left(-(\pi m/\alpha)^2\right)}{m^2} \exp\left(2\pi i < m, p_k - p_j >\right)$$

und

$$E_m = \frac{1}{2} \sum_{i=1}^{\lambda} q_i \phi_{recip}(p_i) \; . \tag{4.6}$$

Die Funktionen ρ_s und ϕ_{recip} sind dabei über eine Poissongleichung miteinander verbunden:

$$\Delta \phi_{recip}(x) = 4\pi \rho_s(x) \; . \tag{4.7}$$

Das reziproke Potential wird auf einem Gitter mit Hilfe der FFT ausgewertet. Dadurch erhält man gute Werte für ϕ_{recip} (und seinem Gradienten) in *Gitterpunkten*. Ebenso könnte man natürlich (4.7) mit einem der in Kapitel 3 beschriebenen Mehrgitter-Verfahren lösen.

Da man zur Berechnung von E_m in der Form (4.6) die Werte von ϕ_{recip} in den Ladungspunkten p_i benötigen würde, wird die reziproke Summe folgendermaßen aufgespalten:

$$\begin{aligned} E_m &= \frac{1}{2}\sum_{i=1}^{\lambda} q_i \phi_{recip}(p_i) \\ &= \frac{1}{2}\int \rho(\tilde{p})\phi_{recip}(\tilde{p})\, d\tilde{p} \\ &= \frac{1}{2}\int [\rho(\tilde{p}) + \rho_s(\tilde{p})]\phi_{recip}(\tilde{p})\, d\tilde{p} - \frac{1}{2}\int \rho_s(\tilde{p})\phi_{recip}(\tilde{p})\, d\tilde{p} \end{aligned}$$

mit der Punktladungsfunktion der Einheitszelle

$$\rho(x) = \sum_{i=1}^{\lambda} q_i \chi_{p_i}(x)\ .$$

Dahinter steckt wieder die Idee, die Interaktion von ϕ_{recip} mit der Punktladung (p_i, q_i) durch eine Interaktion mit einer Gaußschen Funktion, die dieselbe Gesamtladung q_i hat und deren Zentrum gerade auf p_i liegt, plus einem Korrekturterm zu ersetzen. Letztendlich kann man dann mit den folgenden Gleichungen arbeiten:

$$E = \frac{1}{2}\sum_{|p_i - p_j| < r_c;\, i,j=1}^{\lambda} q_i q_j \frac{erfc(\alpha|p_i - p_j|/\sqrt{2})}{|p_i - p_j|} - \frac{\alpha}{\sqrt{\pi}}\sum_{i}^{\lambda} q_i^2 - \frac{1}{2}\int \rho_s(\tilde{p})\phi_{recip}(\tilde{p})\, d\tilde{p}\ ,$$

$$F_i = -\, q_i \sum_{|p_i - p_j| < r_c;\, j=1}^{\lambda} q_j \left(\alpha\sqrt{\frac{2}{\pi}}\exp\left(-(\alpha|p_i - p_j|)^2/2\right) - \frac{erfc(\alpha|p_i - p_j|/\sqrt{2})}{|p_i - p_j|}\right)\frac{p_i - p_j}{|p_i - p_j|^2}$$
$$+ \int \rho_s^i(\tilde{p})\nabla\phi_{recip}(\tilde{p})\, d\tilde{p}\ .$$

Für die etwas modifizierte direkte Summe wird hier ein Cut-Off-Radius r_c eingesetzt. Die auftretenden Integrale müssen numerisch integriert werden.

Der Vorteil der FFP-Methode liegt darin, daß sie die Energie und ihre Gradienten als stetige Funktionen der Ladungsposition liefert. Allerdings zeigen die in [66] präsentierten Rechenzeiten, daß für eine mittlere Genauigkeit (10^{-4} als relativem Fehler für das Kraftfeld) und eine Systemgröße von $\lambda = 5768$ Teilchen die Laufzeitkosten etwa dreimal höher als bei einer konventionellen Abbruchmethode mit $R_{cutoff} = 9$Å sind ([59]). Die Implementierung dieser Methode muß also noch verbessert werden.

4.4.4 Zusammenfassung

Die vorgestellten auf FFT basierenden Methoden besitzen alle eine Komplexität von $O(\lambda log(\lambda))$. Sowohl die PME- als auch Lutys P^3M-Methode [44] sind sehr effizient. Es ist aber nicht klar, ob P^3M so leicht wie PME eine hohe Genauigkeit erreichen kann. FFP kann zwar hinsichtlich der Genauigkeit, nicht jedoch der Geschwindigkeit mit PME konkurrieren (siehe [20, 59]).

4.5 Auf Multipole basierende Methoden

4.5.1 Fast-Multipole-Algorithmus (FMA)

Der FMA von Greengard und Rokhlin [28, 27, 26] für einzelne (nicht periodische) Zellen sowie vor allem Erweiterungen für periodische Systeme (s.u.) sind in vielen Anwendungen erfolgreich zur Simulation eingesetzt worden. Die entscheidende Eigenschaft dieser Algorithmen ist, daß sie in vielen Fällen $O(\lambda)$-, schlechtestenfalls aber $O(\lambda log(\lambda))$-Verfahren darstellen. Dank rigoros abgeleiteter Fehlergrenzen wird eine bekannte Genauigkeit erreicht.

Verwendet wird wie auch schon in vielen anderen Verfahren folgendes: Das Potential, das auf ein Teilchen wirkt, kann in zwei Komponenten aufgespalten werden. Diejenige, die auf benachbarte Teilchen zurückzuführen ist, wird auch in der FMA direkt berechnet. Diejenige aber (P_{far}), die durch entfernte Teilchen zustande kommt, wird bei Greengard und Rokhlin durch Multipole-Entwicklungen approximiert:

Befinden sich Punkte p_i, $i = 1, \ldots, k$ in einer Kugel mit dem Radius a und ist r der Abstand zwischen dem Kugelzentrum und dem (entfernten) Punkt p (siehe Abbildung 4.4) mit $r > a$, so ist das Potential $P_{far}(r)$, das auf p wirkt und durch die Punktladungen (p_i, q_i) zustande kommt, gegeben durch eine unendliche Multipole-Entwicklung (siehe [28, 37]). Stattdessen sind aber auch andere Entwicklungen zur Approximation von P_{far} denkbar, z.B. Andersons „Ring“- bzw. „Kugel“-Approximation [4].

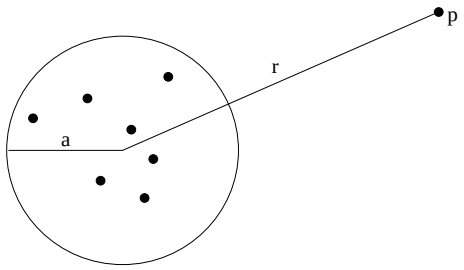
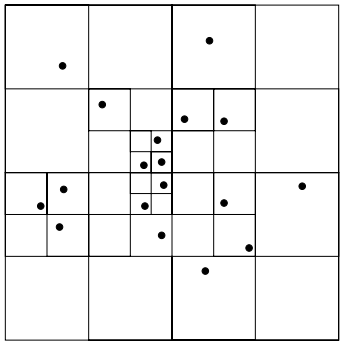


Abbildung 4.4: Zur Multipole-Entwicklung.

Die grundlegende Idee der FMA ist es aber, eine baumstrukturierte hierarchische Unterteilung der Ausgangszelle (Würfel, Level 0) vorzunehmen (Barnes und Hut [8], bereits ein $= O(\lambda log(\lambda))$-Verfahren): Der Ausgangswürfel wird in (üblicherweise) acht gleichgroße Würfel („Kinderzellen“, Level 1) geteilt, und jede dieser Kinderzellen, die mehr als ein Partikel enthält, wird wieder in acht Würfel geteilt (Level 2). Dies wird fortgeführt, bis ein bestimmtes feinstes Level m erreicht ist (siehe Abbildung 4.5).

Auf dem feinsten Level m werden für jedes Teilchen die Wechselwirkungen mit den anderen Teilchen derselben Zelle sowie mit den Teilchen der maximal 26 Nachbarzellen direkt berechnet. Auf dem Level m wird außerdem für jede Zelle eine nur endlich viele Summanden berücksichtigende Multipole-Entwicklung berechnet, die die Wechselwirkungen aller Teilchen in dieser Zelle relativ zum Zentrum der Zelle mit entfernten Teilchen ausdrückt. Diese Entwicklungen werden dann entsprechend der Hierarchie so miteinander kombiniert, daß sie die Effekte von immer größeren Gruppen



(a)

F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
I	I	I	I	I	I	F	F
I	I	I	I	I	I	F	F
I	I	N	N	N	I	F	F
I	I	N	A	N	I	F	F
I	I	N	N	N	I	F	F
I	I	I	I	I	I	F	F

(b)

Abbildung 4.5: (a) Die beschriebene hierarchische Unterteilung des Ausgangswürfels (zur Vereinfachung nur zweidimensional dargestellt). $m = 4$ führt hier bereits dazu, daß jede Zelle auf dem Level m maximal ein Teilchen enthält. (b) Die (maximal 26) direkten Nachbarzellen (gekennzeichnet mit N) einer Zelle A des Levels 3, die Zellen I ihrer Interaktionsliste sowie die von A entfernten Zellen I und F. Eine Box B gehört genau dann zur Interaktionsliste von A, wenn A und B nicht benachbart sind und die Elternboxen von A und B benachbart sind. Zwei Zellen A und B liegen entfernt voneinander, wenn sie keine Nachbarn sind.

von Teilchen repräsentieren („upward pass“). Dabei verschieben jeweils die Kinderzellen ihre Multipole-Entwicklungen zum Zentrum ihrer Elternzelle („multipole to multipole translation“).

Mit Hilfe der „multipole to local translation“ wird ausgehend vom Level 1 für eine Zelle A eine lokale Taylor-Entwicklung berechnet, die das durch alle von A entfernten Teilchen hervorgerufene Potential P_{far} beschreibt. Dazu werden die Multipole-Entwicklungen der Zellen in der Interaktionsliste von A in lokale Entwicklungen konvertiert und diese addiert. Im „downward pass“ werden die lokalen Entwicklungen benutzt, um diejenigen für die jeweiligen Kinderzellen zu berechnen („local to local translation“). Letztendlich werden dann für jedes Teilchen die direkt berechneten Wechselwirkungen mit benachbarten Teilchen und die durch Multipole- und Taylor-Entwicklungen approximierten Wechselwirkungen mit entfernten Teilchen addiert. Genaue Beschreibungen des Algorithmus finden sich in [28, 27]. Dreidimensionale parallele Implementierungen wurden z.B. in [12, 13, 53] getestet.

Die Fast-Multipole-Methode ist erweitert worden, um auch Systeme mit periodischen Randbedingungen behandeln zu können. Um Multipole-Entwicklungen für die periodischen Kopien der Einheitszelle zu erhalten, wird dabei von der Tatsache Gebrauch gemacht, daß die Einheitszelle und ihre Kopien „dieselbe“ Multipole-Entwicklung besitzen: Diejenige der Einheitszelle muß nur auf das Zentrum der jeweiligen Kopie verschoben werden. Dies wurde beispielsweise in [53] für periodische Systeme verwendet.

Schmidt und Lee [51] haben eine Methode entwickelt, die neutrale Punktladungssysteme mit periodischen Randbedingungen in drei Dimensionen simuliert und dazu sowohl FMA- als auch Ewald-Techniken ebenfalls unter Beachtung der obigen Tatsache benutzt. In ihren Geschwindigkeits- und Ge-

nauigkeitstests für ihre FMA-Implementierung und die Ewald-Summation ergaben sich je nach Parameterwahl in beiden Fällen relative Fehler der Größenordnungen 10^{-3} bis 10^{-5}. Da allerdings die Verfahren nicht optimiert wurden, ist nicht klar, bei welcher Systemgröße λ beide Verfahren gerade noch gleich schnell sind, um die gleiche Fehlergröße zu erreichen. Daher kann über die Effizienz ihres Algorithmus lediglich festgestellt werden, daß die Behandlung eines periodischen Systems gegenüber der Behandlung einer einzigen Einheitszelle etwa 2% aufwendiger ist.

4.5.2 Reduced-Cell-Multipole-Methode (RCMM)

Die RCMM [19] versucht, die Kosten der Ewald-Summation ebenfalls durch Verwendung eines hierarchischen Verfahrens [27, 18] zu reduzieren. Ähnlich dem FMA werden Wechselwirkungen zwischen der Einheitszelle und den 26 nächstgelegenen Kopien der Einheitszelle (Nachbarzellen) mit der Zell-Multipole-Methode [18] berechnet, während anders als im FMA die Wechselwirkungen mit den entfernten Kopien der Einheitszelle (Distanzzellen) durch die Ewald-Summation ausgewertet werden.

Um die Distanzwechselwirkungen effizient zu berechnen, wird jede Distanzzelle durch eine reduzierte Zelle ersetzt. Diese bestehen jeweils aus 35 zufällig verteilten geladenen Teilchen, deren q_i so berechnet werden, daß die ersten fünf Glieder einer Multipole-Entwicklung für die reduzierte Zelle mit denen der Einheitszelle übereinstimmen.

In [19] wird berichtet, daß dieses Verfahren sehr genau sei und sein Rechenaufwand proportional mit der Anzahl λ der Teilchen in der Einheitszelle steige. Hierbei muß aber betont werden, daß die reduzierten Zellen mit 35 Teilchen die Multipole-Entwicklung eines Teilchensystems nur mit fünfter Ordnung approximieren. Vorsicht ist also geboten, weil dies in manchen Simulationen nur zu durchschnittlich genauen Werten etwa für das Potential führen kann.

4.5.3 Particle^3-Mesh/Multipole-Expansion (P^{3}M/MPE)

Die P^{3}M/MPE-Methode von Shimada et al. [54, 55] stellt eine Erweiterung der P^{3}M-Methode von Hockney und Eastwood [36] (vergleiche Abschnitt 4.4.1) mit Multipole-Entwicklungen dar.

Zuerst wird die Simulationsbox in $M_1 \times M_2 \times M_3$ Zellen geteilt. Im Zentrum jeder Zelle befindet sich ein Gitterpunkt, für den das Potential, die Multipole-Entwicklung usw. bestimmt werden. Das elektrostatische Potential bzw. das Kraftfeld, das auf ein Teilchen i wirkt, wird in Partikel-Partikel- (PP-) und Partikel-Gitter- (PM-) Wechselwirkungen gespalten.

Die nur auf kurze Entfernung wirkenden PP-Wechselwirkungen zwischen zwei Partikeln in derselben oder in benachbarten Zellen werden direkt berechnet. Die weitreichenden PM-Wechselwirkungen eines Teilchens i mit den restlichen Zellen, die deutlich getrennt von p_i liegen (Distanzzellen), werden für i's Zellenzentrum berechnet, indem man das durch alle entfernten Zellen hervorgerufene Potential durch eine Multipole-Entwicklung ausdrückt.

Dann werden die PM-Methoden aus [36] verwendet, denen FFT-Verfahren (und nicht hierarchische Schemata wie in FMA oder RCMM) zugrunde liegen. Da die PM-Potentiale (bzw. -Kraftfelder) wie glatte Funktionen behandelt werden können, können die Resultate für die tatsächlichen Teilchenpunkte interpoliert werden.

Die Leistung der P^3M/PME-Methode für die Simulation eines zeitabhängigen Systems wird verbessert, wenn die Behandlung der beiden Bereiche (PP bzw. PM) folgendermaßen abläuft: Die PP-Wechselwirkungen werden in jedem Zeitschritt berechnet, wobei jeweils die aktuellsten Informationen über die Teilchenpositionen benutzt werden. Die PM-Wechselwirkungen werden dagegen nur nach jedem 10. bis 20. Zeitschritt neu berechnet. Diese Verfahrensweise hat in Tests die Berechnungszeit um durchschnittlich ein Drittel verkürzt.

In [54] wird behauptet, daß P^3M-Methoden allein „keine extrem genauen Resultate erzielen“ und daher nur mit Vorsicht für präzise Simulationen langer Zeiträume eingesetzt werden sollten. Dort ([54]) findet sich auch ein Vergleich der Methode mit Hockney und Eastwoods Verfahren. Hinsichtlich der Gesamtleistung sollen beide vergleichbar sein.

4.5.4 Macroscopic-Multipole-Methode (MMM)

Die MM-Methode von Lambert et al. [43], ein $O(\lambda)$-Algorithmus, verwendet Fast-Multipole-Verfahren, um die elektrostatischen Kräfte eines Systems zu berechnen, das aus endlich vielen periodischen Einheitszellen besteht. Die Teilchen werden dabei über ein Gitter aus $3^k \times 3^k$ (2D) bzw. $3^k \times 3^k \times 3^k$ (3D) Zellen verteilt. Die Methode basiert wieder auf der Beobachtung, daß die periodischen Kopien der Einheitszelle dieselben Multipole-Koeffizienten wie diese besitzen, eine einmalige Berechnung dieser Koeffizienten also genügt (siehe auch Abschnitt 4.5.1).

Die Simulationszelle wird wie im FMA rekursiv in kleinere Unterzellen geteilt. Die Multipole-Entwicklung jeder Unterzelle wird auf dem feinsten Level berechnet, und Unterzellen werden je zu größeren Strukturen bis hin zur Einheitszelle zusammengefaßt (upward pass). Nun werden makroskopische Multipole-Prozeduren aufgerufen, um die Multipole-Entwicklung M_i für die Zelle S_i $(i = 1, \ldots, k-2)$ zu berechnen.

Der nun folgende Schritt startet auf dem höchsten Level $(i = k)$ und konvertiert M_i in eine lokale Entwicklung um die zentrale Einheitszelle, falls das makroskopische Gebiet auf diesem Level „weit genug“ von der zentralen Einheitszelle getrennt ist. Sonst wird das Gebiet in 9 (2D) bzw. 27 (3D) Zellen geteilt, und obiges für jede dieser Zellen durchgeführt usw. Auf dem untersten Level dieser Rekursion werden dann Kräfte zwischen Zellen bzw. Teilchen, die nicht weit genug voneinander entfernt sind, direkt berechnet. Sind auf diese Weise alle makroskopischen Zellen behandelt worden, beginnt der Downward Pass der FMA.

In Simulationen, die in der Praxis ablaufen, sorgen $k = 3, \ldots, 6$ und $p = 8, 16$ (p sei die Anzahl der Multipole-Terme) für durchschnittliche bis sehr genaue Resultate. Der vorgestellte Algorithmus ist außerdem sehr effizient und lediglich 25-30% teurer als ein Fast-Multipole-Algorithmus für eine

einzelne Einheitszelle. Die Ergebnisse der MMM mit $p = 8$ und $k = 4$ und Ergebnisse der Ewald-Summation zeigten eine Übereinstimmung in 3 bis 4 signifikanten Stellen. Eine höhere Genauigkeit wird durch Vergrößerung von p erreicht, was allerdings die Rechenzeit verlängert. MMM ist auch in der Lage, effizient nicht-kubische Systeme, d.h. $3^i \times 3^j \times 3^k$-Gitter mit $i \neq j \neq k$, zu behandeln. Dies erlaubt letztlich die Untersuchung z.B. von Oberflächen, also Systemen, die endlich in einer der drei Dimensionen sind.

4.5.5 Zusammenfassung

Zusammenfassend ist festzustellen, daß die Komplexität aller vorgestellten Multipole-Methoden zwischen $O(\lambda)$ und $O(\lambda log(\lambda))$ liegt. Die Macroscopic-Multipole-Methode zeigt dabei die beste Kombination aus Genauigkeit und Effizienz und ist zudem in der Lage, Systeme mit „irgendeiner“ geometrischen Konfiguration zu simulieren.

4.6 Weitere Verfahren

Berman und Greengard [11] haben eine neue generelle Methode zur schnellen Auswertung von Summen bestimmter Potentialfunktionen unendlicher periodischer Systeme vorgestellt. Sie wurde für zwei- und dreidimensionale Systeme entwickelt. Für elektrostatische Punktladungssysteme verwendet das Verfahren Multipole- und Taylor-Entwicklungen, um zu einer rekursiven, unendlichen Summe zu gelangen. Diese benötigt die Auswertung einer Funktion, die den Gitterpunktkoordinaten bestimmte endliche Summen zuordnet.

In [48, 32] wurde die Ewald-Summation modifiziert, um Systeme mit weitreichenden Wechselwirkungen zu simulieren, die lediglich in zwei der drei Dimensionen „unendlich“ sind. Beispiele hierfür sind biologische Membranen und polare Flüssigkeiten. Solche quasi-zweidimensionalen Systeme lassen sich nämlich nicht direkt mit dreidimensionalen Implementierungen der Ewald-Summation behandeln, weil dies eventuell zu unrealistischen Wechselwirkungen zwischen den Schichten der endlichen Dimension führt und das Verfahren außerdem ineffizient wäre. Die modifizierte Methode soll dagegen eine vernünftige Geschwindigkeit und Genauigkeit erreichen, benötigt dafür allerdings viel Speicherplatz.

Die hier bereits erwähnten Verfahren und Ansätze sind die momentan wohl am häufigsten eingesetzten. Es gibt aber noch eine ganze Reihe anderer, die teilweise auch Mehrgitter-Verfahren verwenden. Hier sind z.B. Arbeiten von Brandt und Lubrecht [14] zu nennen, die ein schnelles Mehrgitter-Verfahren für die Multi-Integration (und diskrete Analoga) entwickelt haben.

4.7 Zusammenfassung

Algorithmen aller drei Klassen (Standard-, Fourier- bzw. Multipole-basierte Ewald-Summationen) sind heutzutage in Simulationspaketen vertreten, wenn

auch eine konventionelle Ewald-Summation selten verwendet wird, da sie teuer ist. Die Standard-Methoden sind am einfachsten zu programmieren und die Fourier- etwas leichter als die Multipole-basierten.

Eine vergleichende Zusammenstellung von Berechnungszeiten und relativen Fehlern für spezielle Tests, die in der Literatur für einige der oben beschriebenen FFT- bzw. Multipole-Verfahren aufgezeichnet sind, findet sich z.B. in Tafel 1 in [59]. Auch wenn die Werte schwer vergleichbar sind, da verschiedene Rechner und Teilchensysteme verwendet wurden und ein Vergleich in der Literatur oft nur zu Standard-Ewald-Methoden gezogen wird, spiegelt die Tabelle doch folgenden allgemein beobachteten Trend wieder:

Die Multipole-Methoden erweisen sich als starke Konkurrenten zu den Standard- und Fourier-Verfahren. Kleine Systeme, $\lambda \leq 1000$, können effizient (hinsichtlich Rechenzeit und Genauigkeit) mit Standard-ES-Verfahren simuliert werden, für Systeme der Größenordnung $10^3 - 10^4$ sind Fourier-Methoden üblicherweise effizienter, und für noch größere Systeme eventuell spezielle Multipole-Methoden.

Allerdings hängen solche Vergleiche stark davon ab, wie der jeweilige Algorithmus auf dem jeweiligen Computer implementiert wird. So wird in der Literatur von Systemgrößen $\lambda = 300$ [19] bis $\lambda = 30000$ [57] berichtet, bei denen Fourier- gerade noch so schnell wie Multipole-Verfahren sein sollen. In [21] wird sogar berichtet, daß FMA und eine direkte Implementierung der ES sich erst bei $\lambda = 100000$ trennen sollen. Ausführliche Tests von Pollock und Glasli [46] für P^3M, FMA und ES, die auch parallele Implementierungen von P^3M und der Ewald-Summation vergleichen, kommen zu dem Ergebnis, daß selbst bei sehr großen Teilchensystemen die P^3M der FMA und der ES (auch klar im Parallelen) vorzuziehen sei. Eine Entscheidung zwischen Fourier-basierten und Multipole-Methoden ist also längst noch nicht gefallen.

Für Teilchensysteme, die wie Kristalle eine „vollständige“ Periodizität aufweisen, scheint aber noch immer die „echte“ Ewald-Summation bzw. ihre Durchführung mit Hilfe der FFT die beste Vorgehensweise zu sein.

Kapitel 5

Ein neues Verfahren

5.1 Einleitung

Die hier vorgestellte neue Methode, beschrieben in den Abschnitten 5.1 und 5.2, wurde im wesentlichen von Takumi Washio [62] entwickelt. Sie basiert auf der folgenden Definition der elektrostatischen Energie E, die den Vorteil hat, „mathematisch besser faßbar“ als die Definition (4.2) zu sein:

$$E(p_1,\ldots,p_\lambda) := \lim_{S\to\infty} E(S : p_1,\ldots,p_\lambda) \tag{5.1}$$

mit[1]

$$E(S : p_1,\ldots,p_\lambda) := \frac{1}{2}\left(\sum_{i\neq j} q_i q_j \frac{\phi(|p_i-p_j|/S)}{|p_i-p_j|} + \sum_{n\neq 0}\sum_{i,j} q_i q_j \frac{\phi(|p_i-p_j+n|/S)}{|p_i-p_j+n|}\right)$$

für $S \in \mathbb{R}_+$. Dabei sei ϕ eine nichtnegative $C^\infty(\mathbb{R})$-Funktion mit den folgenden Eigenschaften:

$$\begin{aligned} \phi(0) &= 1\ , \\ C(\phi) := \int_0^\infty r\phi''(r)dr &< \infty\ . \end{aligned} \tag{5.2}$$

Das Kraftfeld für die Punktladung (p_i, q_i) ist nun wieder definiert durch

$$F_i := -\nabla_{p_i} E(p_1,\ldots,p_\lambda)\ . \tag{5.3}$$

Definiert man noch für $x, y \in \mathbb{R}^3$, $x \notin y+\mathcal{N}(d)$ und $S \in \mathbb{R}_+$ die Funktionen $G(S : \cdot,\cdot)$ durch

$$G(S : x, y) := \left\{ \frac{\phi(|x-y|/S)}{|x-y|} + \sum_{n\neq 0}\left(\frac{\phi(|x-y+n|/S)}{|x-y+n|} - \frac{\phi(|n|/S)}{|n|}\right)\right\}$$

[1] Die Indexgrenzen 1 bzw. λ für Indizes, die sich auf die p_i bzw. q_i beziehen, werden im folgenden oft weggelassen. Die entsprechenden Summen sind natürlich trotzdem endlich.

und beachtet, daß wegen $\sum_{i=1}^{\lambda} q_i = 0$

$$\sum_i q_i^2 = -\sum_{i \neq j} q_i q_j$$

gilt, so erhält man

$$E(S : p_1, \ldots, p_\lambda) = \frac{1}{2} \sum_{i \neq j} q_i q_j G(S : p_i, p_j) \; .$$

Sei nun ϕ so gewählt, daß die Funktionen $G(S : \cdot, \cdot)$ für $S \to \infty$ gleichmäßig gegen eine Funktion $G(\cdot, \cdot)$ konvergieren. Dann existiert offenbar auch der Grenzwert in (5.1).

Wie im Anhang A.2 gezeigt wird, ist die Funktion G d-periodisch und durch folgende Poissongleichung charakterisiert:

$$\begin{aligned} &\forall\, x, y \in \mathbb{R}^3 \; : \; -\Delta_x G(x,y) = 4\pi \sum_{n \in \mathcal{N}(d)} \chi_{y+n}(x) - \frac{4\pi}{d^3} C(\phi) \; , \\ &\lim_{x \to y} \left(G(x,y) - \frac{1}{|x-y|} \right) = 0 \; . \end{aligned} \tag{5.4}$$

Die zweite Gleichung sorgt dafür, daß die Lösung für G eindeutig bestimmt ist. G läßt sich also mit einer Greenschen Funktion auf dem dreidimensionalen Torus $T^3(d) := \mathbb{R}^3 / \mathcal{N}(d)$ identifizieren. Hätte man nun eine d-periodische Lösung Φ der Poissongleichung

$$\begin{aligned} -\Delta \Phi(x) &= \sum_{i=1}^{\lambda} q_i \left(4\pi \sum_{n \in \mathcal{N}(d)} \chi_{p_i+n}(x) - \frac{4\pi}{d^3} C(\phi) \right) \\ &= \sum_{i=1}^{\lambda} q_i 4\pi \sum_{n \in \mathcal{N}(d)} \chi_{p_i+n}(x) \; , \end{aligned}$$

so würde man wegen

$$\Phi(x) = \sum_{i=1}^{\lambda} q_i G(x, p_i) + const.$$

(Begründung analog Abschnitt 5.2.3) die elektrostatische Energie und das Kraftfeld (formal) aus

$$\begin{aligned} E(p_1, \ldots, p_\lambda) &= \frac{1}{2} \sum_{i=1}^{\lambda} q_i (\Phi(x) - q_i G(x, p_i))|_{x=p_i} \; , \\ F_i &= -q_i \nabla_x \left(\Phi(x) - q_i G(x, p_i) \right)|_{x=p_i} \end{aligned}$$

erhalten. Hierbei ergeben sich nun die folgenden Schwierigkeiten für die Berechnung von E und F_i aus einer (diskreten) Lösung für Φ:

- Der Term $\lim_{x \to p_i} G(x, p_i)$, der (in $\mathbb{R}$) nicht konvergiert, fließt in die obige Formulierung mit ein. Er würde dem „Einfluß jeder Punktladung auf sich selbst“ entsprechen.
- Die rechte Seite der obigen Poissongleichung weist Singularitäten auf.

5.2 Aufspaltung der Greenschen Funktion

5.2.1 Definition von ρ und Aufspaltung von G

Um die elektrostatische Energie E und das Kraftfeld F_i letztendlich effizient berechnen zu können und dabei die oben geschilderten Probleme zu umgehen, wird zuerst (ähnlich der Ewald-Summation und Varianten) eine Aufspaltung der Greenschen Funktion G in zwei Funktionen U und V vorgenommen. Dazu wird eine auf $[0,\infty[$ definierte, stetige Funktion ρ mit den folgenden Eigenschaften verwendet:

$$
\begin{aligned}
&\rho \in C^1]0,\infty[\cap C^3]0,1[\,, \\
&\lim_{r \gtrdot 0} \rho'(r) = 0\,, \\
&\sup_{r\in]0,1[} |\rho'(r)/r| < \infty\,, \\
&\forall\, j \in \{0,1,2,3\} \sup_{r\in]0,1[} |\rho^{(j)}| < \infty\,, \\
&\forall\, r > 1 \,:\, \rho(r) = 0\,, \\
&\int_0^1 \rho(r) 4\pi r^2\, dr = 1\,.
\end{aligned}
\tag{5.5}
$$

Ein Beispiel für eine solche Funktion findet sich im Abschnitt 5.3.1. Die ersten vier Bedingungen werden vor allem im Abschnitt 5.3.3 für Fehlerabschätzungen verwendet. Definiert man nun für $R < d/2$ die Funktionen U und V als d-periodische Lösungen der folgenden Poissongleichungen mit zusätzlichen Bedingungen zum Erhalt jeweils einer eindeutigen Lösung:

$$
\forall x \in \mathbb{R}^3 \,:\, -\Delta U(R:x) = 4\pi \sum_{n\in\mathcal{N}(d)} \chi_n(x) - \frac{4\pi}{R^3}\rho\left(\frac{|x|_d}{R}\right)\,, \tag{5.6}
$$

$$
\lim_{x\to 0}\left(U(R:x) - \frac{1}{|x|}\right) = 0\,, \tag{5.7}
$$

$$
\forall x \in \mathbb{R}^3 \,:\, -\Delta V(R:x) = \frac{4\pi}{R^3}\rho\left(\frac{|x|_d}{R}\right) - \frac{4\pi}{d^3}C(\phi)\,, \tag{5.8}
$$

$$
V(R:0) = 0\,, \tag{5.9}
$$

wobei

$$
|x|_d := \min_{n\in\mathcal{N}(d)} |x+n|
$$

sei, so ist

$$
G(x,y) = U(R:x-y) + V(R:x-y)\,,
$$

und die elektrostatische Energie läßt sich aus

$$
E(p_1,\ldots,p_\lambda) = \frac{1}{2}\left(\sum_{i\neq j} q_i q_j U(R:p_i-p_j) + \sum_{i,j} q_i q_j V(R:p_i-p_j)\right) \tag{5.10}
$$

berechnen. Daß in der zweiten Summe auch die Summanden mit $i = j$ auftreten können, kommt daher, daß nach Definition $V(R:0) = 0$ gilt.

Die Bedeutung von ρ besteht darin, eine stetige und *stückweise* (mindestens) dreimal stetig differenzierbare, sphärisch symmetrische Verteilung

$$q_i \rho\left(\frac{|x - p_i|_d}{R}\right)$$

der Ladung q_i um den Punkt p_i zu bestimmen. Die Weite der Verteilung ist durch den Radius R gegeben. Dies entspricht zusammen mit der Spaltung von G der Grundidee der Ewald-Summation und verwandter Verfahren (vergleiche besonders die Abschnitte 4.2.2, 4.4.2 und 4.4.3). Im folgenden werden nun Gleichungen hergeleitet, die eine effiziente Berechung von U und V und somit E und F_i gestatten.

5.2.2 Berechnung von U

Für $x \in [-d/2, d/2]^3$ ist $|x|_d = |x|$, und daher

$$-\Delta U(R:x) = 4\pi\chi_0(x) - \frac{4\pi}{R^3}\rho\left(\frac{|x|}{R}\right).$$

Insbesondere für $|x| \leq R$ ist wegen $R < d/2$ also $|x|_d = |x|$. Unter der Annahme, daß U sphärisch symmetrisch, d.h. $U(R:x) = X(R:|x|_d)$ mit passender Funktion X ist, liefert das Divergenz-Theorem von Gauß (siehe [33]) aufgrund obigem, daß X durch das Lösen der folgenden gewöhnlichen Differentialgleichung erhalten werden kann:

$$\begin{aligned} \forall\, 0 < r \leq d/2 : \quad -4\pi r^2 \frac{dX}{dr}(R:r) &= 4\pi - \frac{4\pi}{R^3}\int_{|x|\leq r} \rho\left(\frac{|x|}{R}\right) dx \\ &= 4\pi - \frac{4\pi}{R^3}\int_0^r \rho(t/R)4\pi t^2\, dt \\ &= 4\pi - 4\pi\int_0^{r/R} \rho(s)4\pi s^2\, ds \end{aligned} \tag{5.11}$$

mit der zusätzlichen Bedingung

$$\lim_{r\to 0}\left(X(R:r) - \frac{1}{r}\right) = 0\,. \tag{5.12}$$

Definiert man

$$\begin{aligned} \Theta(r) &:= 4\pi\int_0^r \rho(s)s^2\, ds\,, \\ \Gamma(r) &:= 4\pi\int_0^r \rho(s)s\, ds\,, \end{aligned}$$

so erhält man folgende Schar von Lösungen der Differentialgleichung ($a \neq 0$, $c \in \mathbb{R}$):

$$\begin{aligned} X(R:r) + c &= \frac{1}{r} + \int_a^r \frac{\Theta(t/R)}{t^2}\,dt \\ &= \frac{1}{r} + \left[-\frac{\Theta(t/R)}{t}\right]_a^r + \int_a^r \frac{\Theta'(t/R)}{tR}\,dt \\ &= \frac{1}{r} - \frac{\Theta(r/R)}{r} + \frac{\Theta(a/R)}{a} + \frac{4\pi}{R^3}\int_a^r \rho(t/R)t\,dt \\ &= \frac{1}{r} - \frac{\Theta(r/R)}{r} + \frac{\Theta(a/R)}{a} + \frac{4\pi}{R}\int_{a/R}^{r/R} \rho(s)s\,ds \\ &= \frac{1}{r} - \frac{\Theta(r/R)}{r} + \frac{\Theta(a/R)}{a} + \frac{\Gamma(r/R)}{R} - \frac{\Gamma(a/R)}{R}\,, \end{aligned}$$

wobei sich aus (5.12) und (5.5)

$$\frac{\Theta(a/R)}{a} - \frac{\Gamma(a/R)}{R} - c = 0$$

ergibt. Dies liefert für $0 < |x|_d \leq d/2$ also:

$$\begin{aligned} U(R:x) &= X(R:|x|_d) \\ &= \frac{1}{|x|_d} - \frac{\Theta(|x|_d/R)}{|x|_d} + \frac{\Gamma(|x|_d/R)}{R}\,. \end{aligned} \tag{5.13}$$

Für $r \geq 1$ ist wegen (5.5) offenbar

$$\begin{aligned} \Theta(r) &= 1\,, \\ \Gamma(r) &= \Gamma(1)\,. \end{aligned} \tag{5.14}$$

Ebenfalls wegen (5.5) ergibt sich für $|x|_d > R$

$$-\Delta U(R:x) = 0\,. \tag{5.15}$$

$U(R:x)$ ist also für $|x|_d > R$ konstant. Weil aber $R < d/2$ ist und für $|x|_d \in [R, d/2]$ aus den obigen Ergebnissen (5.13) und (5.14)

$$\begin{aligned} U(R:x) &= \frac{1}{|x|_d} - \frac{1}{|x|_d} + \frac{\Gamma(1)}{R} \\ &= \frac{\Gamma(1)}{R} \end{aligned}$$

folgt, ist auch

$$\forall\, |x_d| \geq R\ :\quad U(R:x) = \frac{\Gamma(1)}{R}\,. \tag{5.16}$$

Die oben gemachte Annahme, daß $U(R:x) = X(R:|x|_d)$ gilt, führt also zu einer eindeutigen Lösung U der Gleichungen (5.11) und (5.12). Andererseits erfüllt die erhaltene Funktion auch die Gleichungen (5.6) und (5.7),

wie man leicht nachrechnen kann, und ist damit die gesuchte Lösung dieser Gleichungen. Aus (5.16) und den Gleichungen

$$\sum_{i\neq j,|p_i-p_j|_d\geq R} q_iq_j = \sum_i q_i\Bigg(\sum_{j(\neq i),|p_i-p_j|_d\geq R} q_j\Bigg),$$
$$\sum_{j(\neq i):|p_i-p_j|_d\geq R} q_j + \sum_{j(\neq i):|p_i-p_j|_d<R} q_j + q_i = \sum_j q_j = 0$$

erhält man nun

(5.17)

$$\begin{aligned}
&\frac{1}{2}\sum_{i\neq j} q_iq_jU(R:p_i-p_j)\\
=&\frac{1}{2}\sum_{i\neq j,|p_i-p_j|_d<R} q_iq_jU(R:p_i-p_j)+\frac{\Gamma(1)}{2R}\sum_{i\neq j,|p_i-p_j|_d\geq R} q_iq_j\\
=&\frac{1}{2}\sum_{i\neq j,|p_i-p_j|_d<R} q_iq_jU(R:p_i-p_j)+\frac{\Gamma(1)}{2R}\sum_i q_i(-q_i-\sum_{j(\neq i):|p_i-p_j|_d<R} q_j)\\
=&\frac{1}{2}\sum_{i\neq j,|p_i-p_j|_d<R} q_iq_j\left(U(R:p_i-p_j)-\frac{\Gamma(1)}{R}\right)-\frac{\Gamma(1)}{2R}\sum_i q_i^2\\
=&\frac{1}{2}\sum_{i\neq j,|p_i-p_j|_d<R} q_iq_j\left(\frac{1-\Theta(|p_i-p_j|_d/R)}{|p_i-p_j|_d}+\frac{\Gamma(|p_i-p_j|_d/R)-\Gamma(1)}{R}\right)\\
&-\frac{\Gamma(1)}{2R}\sum_i q_i^2\,.
\end{aligned}$$

Dies zeigt, daß man für jedes p_i den Cut-Off außerhalb des Kreises um p_i mit Radius R in der Berechnung des ersten Terms der rechten Seite von (5.17) ohne jeden Fehler anwenden kann.

5.2.3 Berechnung von V

Nachdem nun also eine Formel zur schnellen Berechnung von U entwickelt wurde, soll dies auch für V geschehen. Sei nun $\psi(R:\cdot)$ eine d-periodische Lösung der Gleichung

(5.18)

$$\begin{aligned}
-\Delta\psi(R:x) &= \sum_{j=1}^{\lambda} q_j\left(\frac{4\pi}{R^3}\rho\left(\frac{|x-p_j|_d}{R}\right)-\frac{4\pi}{d^3}C(\phi)\right)\\
&= \frac{4\pi}{R^3}\sum_{j=1}^{\lambda} q_j\rho\left(\frac{|x-p_j|_d}{R}\right).
\end{aligned}$$

Da dies eine (dreidimensionale) Poissongleichung darstellt, deren rechte Seite wegen $R<d/2$ stetig (und stückweise (dreimal) stetig differenzierbar) und d-periodisch ist und die Kompatibilitätsbedingung (1.3) erfüllt, gilt mit der Definition von V offensichtlich

$$\psi(R:x) = \sum_{j=1}^{\lambda} q_jV(R:x-p_j)+const.\,,$$

und somit wegen $\sum_{i=1}^{\lambda} q_i = 0$

$$\frac{1}{2}\sum_{i,j} q_i q_j V(r : p_i - p_j) = \frac{1}{2}\sum_i q_i \psi(R : p_i) \,. \tag{5.19}$$

Damit hat man nun die Berechnung der rechten Summe in (5.10) auf die Bestimmung von $\psi(R : \cdot)$ zurückgeführt. Dies hat den Vorteil, daß zur Berechnung einer Näherungslösung $\psi_h(R : \cdot)$ von (5.18) eines der Mehrgitter-Verfahren aus Kapitel 3 eingesetzt werden kann. Hierbei muß allerdings beachtet werden, daß im allgemeinen nicht $f \notin C^2([0,d]^3)$ gilt, wobei f die rechte Seite von (5.18) darstellt (vergleiche Abschnitte 5.2.1 und 5.4).

5.2.4 Berechnung von E und F_i

Unter Ausnutzung von (5.17) und (5.19) für (5.10) ergibt sich zusammenfassend also folgende Gleichung zur Berechnung von E:

$$\begin{aligned}
&E(p_1,\ldots,p_\lambda)\\
=&\frac{1}{2}\Bigg(\sum_{i\neq j} q_i q_j U(R : p_i - p_j) + \sum_{i,j} q_i q_j V(R : p_i - p_j)\Bigg)\\
=&\frac{1}{2}\sum_{i\neq j,\,|p_i-p_j|_d<R} q_i q_j \left(\frac{1-\Theta(|p_i-p_j|_d/R)}{|p_i-p_j|_d} + \frac{\Gamma(|p_i-p_j|_d/R)-\Gamma(1)}{R}\right)\\
&-\frac{\Gamma(1)}{2R}\sum_i q_i^2 + \frac{1}{2}\sum_i q_i\psi(R : p_i) \,.
\end{aligned}$$

Zur Herleitung der entsprechenden Formel für die F_i muß E gemäß (5.3) noch abgeleitet werden. Dazu werden die folgenden Gleichungen benötigt:

$$\begin{aligned}
\nabla_x\left(\frac{1}{|x|}\right) &= -\frac{x}{|x|^3} \,,\\
\nabla_x\left(\frac{\Theta(|x|/R)}{|x|}\right) &= \frac{4\pi}{R^3} x\,\rho\left(\frac{|x|}{R}\right) - \frac{x}{|x|^3}\Theta\left(\frac{|x|}{R}\right) \,,\\
\nabla_x\left(\frac{\Gamma(|x|/R)}{R}\right) &= \frac{4\pi}{R^3} x\,\rho\left(\frac{|x|}{R}\right) \,.
\end{aligned}$$

Mit $x_d = a \in x + \mathcal{N}(d)$, so daß $|a| = |x|_d$ gilt, ist dann

$$\nabla_x U(R : x) = -\frac{x_d}{|x|_d^3} - \frac{4\pi}{R^3} x_d \rho\left(\frac{|x|_d}{R}\right) + \frac{x_d}{|x|_d^3}\Theta\left(\frac{|x|_d}{R}\right) + \frac{4\pi}{R^3} x_d \rho\left(\frac{|x|_d}{R}\right)$$

und somit

$$\nabla_{p_i} U(R : p_i - p_j) = -\frac{(p_i-p_j)_d}{|p_i-p_j|_d^3}\left(1 - \Theta\left(\frac{|p_i-p_j|_d}{R}\right)\right) \,.$$

Außerdem gilt (da V wie U entsprechend Abschnitt 5.2.2 sphärisch symmetrisch ist):

$$\begin{aligned}\nabla_{p_i}\frac{1}{2}\sum_j q_j\psi(R:p_j) &= \frac{1}{2}\nabla_{p_i}\sum_{j,k} q_jq_kV(R:p_j-p_k)\\ &= \frac{1}{2}\nabla_{p_i}\sum_j 2q_iq_jV(R:p_i-p_j)\\ &= q_i\nabla_{p_i}\psi(R:p_i)\ .\end{aligned}$$

Dann kann das auf die i-te Punktladung wirkende Kraftfeld insgesamt folgendermaßen berechnet werden:

$$\begin{aligned}F_i &= -q_i\sum_{j(\neq i):|p_i-p_j|_d<R} q_j\nabla_{p_i}U(R:p_i-p_j) - q_i\nabla_{p_i}\psi(R:p_i)\\ &= q_i\sum_{j(\neq i):|p_i-p_j|_d<R} q_j\frac{(p_i-p_j)_d}{|p_i-p_j|_d^3}\left(1-\Theta\left(\frac{|p_i-p_j|_d}{R}\right)\right) - q_i\nabla_{p_i}\psi(R:p_i)\ .\end{aligned}$$

5.3 Diskrete Lösung

5.3.1 Ein Algorithmus zur Berechnung von E und F_i

Im folgenden bezeichnen E_h und $F_{h,i}$ die Approximationen für E und F_i, die mit Hilfe einer Näherungslösung $\psi_h(R:\cdot)$ für $\psi(R:\cdot)$ auf einem Gitter Ω_h (siehe dazu Kapitel 3) berechnet werden sollen. Die Zellgröße ist hierbei (o.B.d.A.) durch $d=1$ gegeben.

Zur Bestimmung von E_h und $F_{h,i}$ wurde das FORTRAN-Programm CHEM erstellt, welches in sechs Schritten die diskreten Lösungen bestimmt:

1. *ρ, R und h geeignet wählen; p_i und q_i einlesen.*
2. *Die (diskrete) rechte Seite von* (5.18) *berechnen.*
3. *Eine diskrete Lösung für $\psi(R:\cdot)$ bestimmen.*
4. *Mit obigem ρ die Funktionen Θ und Γ durch (exakte) Integration berechnen.*
5. *Die Näherung E_h berechnen.*
6. *Die Näherung $F_{h,i}$ für alle $i=1,\ldots,\lambda$ berechnen.*

Zum ersten Schritt:

Beispielsweise erfüllt die folgende Funktion ρ alle Bedingungen in (5.5):

$$\rho(r) := \begin{cases}\frac{3\pi}{4\pi^2-24}(1+\cos(\pi r)) & \text{für } r\le 1,\\ 0 & \text{sonst.}\end{cases}$$

Durch diese Wahl von ρ können die Funktionen Θ und Γ leicht berechnet werden (siehe vierter Schritt). Sie wird im folgenden immer verwendet. Als weitere Parameter müssen der Radius R und die Maschenweite $h = 2^p$ mit $p \in \mathbb{N}$ des Gitters Ω_h (siehe auch Kapitel 3) festgelegt werden. Siehe dazu auch Abschnitt 5.3.2.

Aus einer vorher erzeugten Datei werden λ und die Ladungspunkte (p_i, q_i) $(i = 1, \ldots, \lambda)$ nacheinander eingelesen. Da die Werte $|p_i - p_j|_d$ später bei der Berechnung sowohl von E_h als auch F_h gebraucht werden, kann man sie an dieser Stelle einmal berechnen und abspeichern (wenn genug Speicher vorhanden ist). Da sich alle p_i in $]0,1]^3$ befinden, genügt zur Berechnung von $|p_i - p_j|_d$ die Betrachtung der $|p_i - p_j + n|$ mit $n \in \{(n_1, n_2, n_3) \mid n_i \in \{-1, 0, 1\},\ i = 1, 2, 3\}$. Ist für den Abstand ein Wert ≤ 0.5 erreicht (falls dies möglich ist), hat man $|p_i - p_j|_d$ bereits gefunden und kann die Suche abbrechen. Ansonsten wird das Minimum $(\leq 0.5\sqrt{3})$ bestimmt. Zusätzlich sollte auch der Vektor $(p_i - p_j)_d$ abgespeichert werden (für F_h).

Zum zweiten Schritt:

Bevor man (5.18) numerisch lösen kann, muß die rechte Seite

$$f(x) = \frac{4\pi}{R^3} \sum_i q_i \rho\left(\frac{|x - p_i|_d}{R}\right) \tag{5.20}$$

auf dem Gitter Ω_h diskretisiert werden. Dies geschieht in mehreren Schritten:

Schritt (A): Zuerst wird eine Hilfsfunktion ρ_{hilf} berechnet. Sie stellt die Diskretisierung der Ladungsverteilungsfunktion $\rho(|x|/R)$ mit Zentrum $(0,0,0)$ dar. Der Faktor $4\pi/R^3$ in (5.20) wird dabei bereits berücksichtigt:

$$\forall\, x \in G_h \cap [-R, R]^3 : \quad \rho_{hilf}(x) := \begin{cases} \frac{4\pi\rho(|x|/R)}{R^3} & \text{für } |x| \leq R, \\ 0 & \text{sonst.} \end{cases}$$

Im Algorithmus wird ρ_{hilf} natürlich nur in $G_h \cap [0, R]^3$ berechnet, da die Funktion sphärisch symmetrisch ist.

Schritt (B): Für jeden Ladungspunkt $p_i = (p_{i1}, p_{i2}, p_{i3})$ $(i = 1, \ldots \lambda)$ wird festgestellt, ob er ein Gitterpunkt ist, d.h. $p_i \in \Omega_h$ gilt. Falls dies nicht der Fall ist, wird eine trilinear gewichtete Verteilung seiner Ladung auf die umliegenden 8 Gitterpunkte vorgenommen, wobei die Torusstruktur gegebenenfalls ausgenutzt wird. Sind die Wichtungsfaktoren s_1, s_2 und s_3 durch

$$\begin{aligned} s_1 &= 1 - (p_{i1}/h - int(p_{i1}/h)) , \\ s_2 &= 1 - (p_{i2}/h - int(p_{i2}/h)) , \\ s_3 &= 1 - (p_{i3}/h - int(p_{i3}/h)) \end{aligned}$$

gegeben, wobei $int(z)$ den ganzzahligen Anteil von z bezeichne, so erhält beispielsweise der Gitterpunkt „rechts vorn oben“ [2] von p_i ,

$$(int(p_{i1}/h) * h + h, int(p_{i2}/h) * h, int(p_{i3}/h) * h + h) ,$$

[2] falls er in Ω_h liegt, sonst die entsprechende periodische Kopie.

den Ladungsanteil

$$(1-s_1)s_2(1-s_3)q_i\,.$$

Analog berechnen sich die Anteile für die anderen 7 Punkte.

Schritt (C): Nun wird die diskrete Entsprechung f_h zu f berechnet. Dazu wird die Hilfsfunktion ρ_{hilf} (d.h. ihr Zentrum) auf p_i bzw. die 8 Gitterpunkte um p_i jeweils verschoben (für alle $i=1,\ldots,\lambda$). Dabei wird die Torusstruktur explizit beachtet, d.h. für jedes p_i wird vorher berechnet, wann ρ_{hilf} die Grenzen von Ω_h erreicht. Dann werden die entsprechenden Teile von $\{\rho_{hilf}(x)\,|\,x\in G_h\cap[0,R]^3\}$ gezielt verschoben. Auf diese Weise erspart man sich (wie auch schon in den Mehrgitter-Verfahren der Kapitel 2 und 3) die Anwendung der Modulo-Funktion.

Hinzu kommen gemäß (5.20) die Ladungen q_i bzw. die durch die s_j gewichteten Anteile als Faktoren. Die sich so ergebenden Werte werden zu bereits vorhandenen für die jeweiligen Gitterpunkte addiert. Letztlich erhält man so die Gitterfunktion f_h.

Schritt (B) und Schritt (C) werden im Algorithmus miteinander verbunden. Dabei stellt Schritt (B) die äußere und Schritt (C) die innere Schleife dar.

Zum dritten Schritt:

Die trilinear gewichtete Verteilung der Ladungen bewirkt offensichtlich, daß f_h bereits die diskrete Kompatibilitätsbedingung (3.6) erfüllt. Daher ist die Lösung der diskreten Poissongleichung

$$-\Delta_h\psi_h(R:\cdot)=f_h$$

existent und bis auf eine Konstante eindeutig bestimmt. Sie wird nun mit Hilfe eines der Mehrgitter-Verfahren (ω_2-FR3D oder FR3D) aus Kapitel 3 berechnet.

Zum vierten Schritt:

Um nur an möglichst wenigen Stellen im ganzen Prozeß mit Näherungslösungen rechnen zu müssen, ist es von Vorteil, die Funktionen Γ und Θ explizit angeben zu können. Andererseits ist es wichtig, beide Funktionen schnell auszuwerten, um den Rechenaufwand möglichst gering zu halten. Insbesondere soll eine zeitraubende numerische Integration vermieden werden.

Verwendet man das oben angegebene ρ, so erhält man durch partielle

Integration:

$$\begin{aligned}
\Theta(r) &= 4\pi \int_0^r \rho(s) s^2 \, ds \\
&= \frac{3\pi^2}{\pi^2 - 6} \left(\frac{1}{3} r^3 + \frac{1}{\pi} r^2 \sin(\pi r) + \frac{2}{\pi^2} r \cos(\pi r) - \frac{2}{\pi^3} \sin(\pi r) \right), \\
\Gamma(r) &= 4\pi \int_0^r \rho(s) s \, ds \\
&= \frac{3\pi^2}{\pi^2 - 6} \left(\frac{1}{2} r^2 + \frac{1}{\pi} r \sin(\pi r) + \frac{1}{\pi^2} \cos(\pi r) - \frac{1}{\pi^2} \right), \\
\Gamma(1) &= \frac{3\pi^2 - 12}{2\pi^2 - 12} .
\end{aligned}$$

In diesem Falle sind beide Funktionen also explizit bestimmbar und schnell auswertbar.

Zum fünften Schritt:

Mit Hilfe der bereits bestimmten Funktion $\psi_h(R : \cdot)$ kann man nun eine Näherungslösung für E angeben:

(5.21)

$$\begin{aligned}
E_h(p_1, \ldots, p_\lambda) =& \frac{1}{2} \sum_{i \neq j, |p_i - p_j|_d < R} q_i q_j \left(\frac{1 - \Theta(|p_i - p_j|_d / R)}{|p_i - p_j|_d} + \frac{\Gamma(|p_i - p_j|_d / R) - \Gamma(1)}{R} \right) \\
& - \frac{\Gamma(1)}{2R} \sum_i q_i^2 + \frac{1}{2} \sum_i q_i \psi_h(R : p_i) \\
=& \sum_{i=1}^{\lambda} \sum_{j \neq i, |p_i - p_j|_d < R; j=1}^{i-1} q_i q_j \left(\frac{1 - \Theta(|p_i - p_j|_d / R)}{|p_i - p_j|_d} \right. \\
& \left. + \frac{\Gamma(|p_i - p_j|_d / R) - \Gamma(1)}{R} \right) - \frac{\Gamma(1)}{2R} \sum_i q_i^2 + \frac{1}{2} \sum_i q_i \psi_h(R : p_i) .
\end{aligned}$$

Die Umformung hat den Vorteil, daß die Gleichheit von je zwei Summanden der Doppelsumme ausgenutzt wird: Da sie jeweils nur einmal berechnet werden, halbiert sich der Aufwand zur Berechnung der Doppelsumme.

Falls ein p_i kein Gitterpunkt ist, existiert kein Wert für $\psi_h(R : p_i)$. In diesem Falle wird mit Hilfe der umliegenden 8 Gitterpunkte durch trilineare Interpolation (vergleiche den ersten Schritt) ein Wert gewonnen.

Zum sechsten Schritt:

Ebenso kann mit Hilfe der $\psi_h(R : p_j)$ schließlich eine Approximation $F_{h,i}$ bestimmt werden:

(5.22)

$$\begin{aligned}
F_{h,i} =& q_i \sum_{j(\neq i): |p_i - p_j|_d < R} q_j \frac{(p_i - p_j)_d}{|p_i - p_j|_d^3} \left(1 - \Theta\left(\frac{|p_i - p_j|_d}{R} \right) \right) \\
& - q_i \nabla_{h, p_i} \psi_h(R : p_i) .
\end{aligned}$$

Hierbei wird die Approximation $\nabla_{h,p_i}\,\psi_h(R:p_i)$ für den Gradienten wie folgt berechnet:

$$\nabla_{h,p_i}\,\psi_h(R:p_i) = \begin{pmatrix} \frac{1}{h}\Big(\psi_h(R:(p_{i1}+0.5h,p_{i2},p_{i3})) - \psi_h(R:(p_{i1}-0.5h,p_{i2},p_{i3}))\Big) \\ \frac{1}{h}\Big(\psi_h(R:(p_{i1},p_{i2}+0.5h,p_{i3})) - \psi_h(R:(p_{i1},p_{i2}-0.5h,p_{i3}))\Big) \\ \frac{1}{h}\Big(\psi_h(R:(p_{i1},p_{i2},p_{i3}+0.5h)) - \psi_h(R:(p_{i1},p_{i2},p_{i3}-0.5h))\Big) \end{pmatrix}$$

Dieses Verfahren zur näherungsweisen Berechnung des Gradienten ist dank der hier verwendeten symmetrischen Sterne zweiter Ordnung (siehe auch (1.17)). Die Werte für $\psi_h(R:x)$ an den entsprechenden Stellen x erhält man jeweils wieder durch trilineare Interpolation aus den umliegenden 8 Gitterpunkten.

5.3.2 Rechenaufwand

Der Rechenaufwand $W(\lambda,R,h)$ hängt ab von der Anzahl λ der Teilchen in der Einheitszelle Q_1, der Größe des Radius R und der Maschenweite h. Er setzt sich aus den vier Anteilen W_i $(i\in\{2,3,5,6\})$ für den zweiten, dritten, fünften und sechsten Schritt zusammen, die *im wesentlichen* folgende Gestalt haben:

$$\begin{aligned} W_2 &= \omega_{2,1}(R/h)^3 + \omega_{2,2}\lambda(R/h)^3\,,\\ W_3 &= \omega_3(1/h)^3\,,\\ W_5 &= \omega_{5,1}\lambda_{den}(R,\lambda)\lambda + \omega_{5,2}\lambda\,,\\ W_6 &= \omega_{6,1}\lambda_{den}(R,\lambda)\lambda + \omega_{6,2}\lambda\,. \end{aligned}$$

Hierbei bezeichnen die ω_j von λ, R und h unabhängige Proportionalitätskonstanten. Ist $\lambda_{den,i}(R,\lambda)$ $(i=1,\ldots,\lambda)$ die Anzahl der Ladungspunkte p_j, die höchstens den Abstand R (gemessen mittels $|\cdot|_d$) von p_i $(i\neq j)$ haben, so ist $\lambda_{den}(R,\lambda)$ der Durchschnitt dieser Werte. R ist nun für ein System aus λ Teilchen immer so wählbar, daß $\lambda_{den}(R,\lambda)$ unterhalb eines fest vorgegebenen Wertes bleibt, der bei großem λ deutlich kleiner als λ ist.

Der erste Summand von W_2 gibt die Arbeit für Schritt A, der zweite Summand die Arbeit für die Kombination von Schritt B und C an. Letzterer ist vor allem dank der Abhängigkeit von λ und der trilinear gewichteten Verteilung auf je 8 Punkte deutlich aufwendiger, so daß man für W_2

$$W_2 = \omega_{2,2}\lambda(R/h)^3$$

schreiben kann. Für W_3 wird angenommen, daß ψ_h mit einem der Mehrgitter-Verfahren aus Kapitel 3 in der FMG-Version berechnet wird. W_5 ergibt sich direkt aus der Gleichung zur Berechnung von E_h (siehe fünften Schritt), wobei in $\omega_{5,2}\lambda$ der Aufwand für die letzten beiden Summanden steckt. W_6 spiegelt schließlich den Aufwand für die Berechnung aller $F_{h,i}$ $(i=1,\ldots,\lambda)$ wider.

Liegt eine gleichmäßige Verteilung der p_i im Würfel $[0,1]^3$ vor, so ist $\lambda_{den}(R,\lambda)$ (im Idealfall) proportional zu $(\lambda^{1/3}R)^3 = \lambda R^3$. Dann gilt mit

$\lambda_{den}(R,\lambda) = \omega_7 \lambda R^3$:

$$W_5 = \omega_{5,1}\omega_7 \lambda R^3 \lambda + \omega_{5,2}\lambda\,,$$
$$W_6 = \omega_{6,1}\omega_7 \lambda R^3 \lambda + \omega_{6,2}\lambda\,.$$

Bei der Wahl der Parameter R und h muß beachtet werden, daß ein kleineres R bei festem h und **gegebenem Teilchensystem, d.h. festem** λ, zwar den Rechenaufwand deutlich herabsetzt - W_2 und jeweils der erste Term von W_5 und W_6 sind proportional zu R^3 -, allerdings dazu führt, daß $\psi_h(R:\cdot)$ eine schlechtere Näherung für $\psi(R:\cdot)$ darstellt, weil die rechte Seite f von (5.18) im Grenzfall $R \to \infty$ Singularitäten in den p_i aufweist. Bei kleinerem R muß also normalerweise auch h kleiner gewählt werden, um etwa die gleiche Genauigkeit zu erreichen.

Verkleinert man h sogar im gleichen Maß wie R, so daß also R/h etwa konstant bleibt, ist auch W_2 konstant, und man hat einerseits ein proportional zu $(1/h)^3$ (d.h. der Anzahl der Gitterpunkte) wachsendes W_3, andererseits gegen $\omega_{5,2}\lambda$ bzw. $\omega_{6,2}\lambda$ schrumpfende W_5 bzw. W_6. Ein zu kleines R ist bei festem λ also nicht vorteilhaft, da der Verlust an Genauigkeit nicht mehr durch einen kleineren Rechenaufwand gerechtfertigt wird. Ähnliche Überlegungen gelten offensichtlich umgekehrt für zu große h.

Der Gesamtrechenaufwand **in Abhängigkeit der Systemgröße** λ soll nun im Vordergrund stehen. W_5 und W_6 sind offensichtlich im wesentlichen proportional zu λ, wenn R jeweils so klein gewählt wird, daß $\lambda_{den}(R,\lambda)$ etwa konstant bleibt. Bei einer gleichmäßigen Teilchenverteilung bedeutet das:

$$\begin{aligned} & \lambda R^3 = const. \\ \Leftrightarrow \quad & R \quad = const.^{1/3} \lambda^{-1/3}\,. \end{aligned}$$

Bei festem h überwiegen dann für genügend großes λ die Terme W_5 und W_6 die Terme W_2 und W_3 deutlich, so daß hier ein $O(\lambda)$-Verfahren vorliegt.

Aber auch in dem Fall, daß aufgrund obiger Überlegungen zusätzlich h kleiner gewählt wird, und zwar sogar so klein, daß R/h (etwa) konstant bleibt, ist W_2 im wesentlichen proportional zu λ. Außerdem ist dann h proportional zu $\lambda^{-1/3}$, und daher auch W_3 im wesentlichen proportional zu λ. Auf diese Weise ergibt sich ebenfalls ein $O(\lambda)$-Algorithmus zur Berechnung von E und F_i.

5.3.3 Fehlerabschätzungen

Wie man aus (5.21) und (5.22) ersehen kann, hängt die Genauigkeit der berechneten Näherungslösungen E_h bzw. $F_{h,i}$ direkt von den globalen Diskretisierungsfehlern $||\psi(R:\cdot) - \psi_h(R:\cdot)||_\infty$ bzw. $||\frac{\partial\psi(R:\cdot)}{x_j} - (\nabla_{h,x}\psi_h(R:\cdot))_j||_\infty$ (mit $x = (x_1, x_2, x_3)$) ab. Abschätzungen für diese Größen gibt der folgende Satz an:

Satz 4 . *Wenn ρ den Bedingungen* (5.5) *genügt (wie etwa die Funktion aus Abschnitt 5.3.1) und $\forall\, x \in \Omega_h$: $f_h(x) = f(x)$ angenommen wird, gilt für*

jedes $a \in]0, 1/h[$ *und* $j = 1, 2, 3$

$$||\psi(R:\cdot) - \psi_h(R:\cdot)||_\infty \leq c_1 h^2 (a - \ln(ah)) \sum_{k=1}^{3} \max_{\Omega_h} |\nabla_{h,k}^2 \psi_h| + c_2 \frac{h^2}{R^3} \left(\sup_{]0,1[} |\rho| + \sup_{]0,1[} |\rho'| + \sup_{]0,1[} |\rho''| \right), \tag{5.23}$$

$$\left|\left| \frac{\partial \psi(R:\cdot)}{x_j} - (\nabla_{h,x} \psi_h(R:\cdot))_j \right|\right|_\infty \leq c_3 h^2 (a - \ln(ah)) \sum_{k=1}^{3} \max_{\Omega_h} |\nabla_{h,j} \nabla_{h,k}^2 \psi_h| + c_4 \frac{h^2}{R^4} \left(\sup_{]0,1[} |\rho| + \sup_{r \in]0,1[} |\rho'(r)/r| + \sup_{]0,1[} |\rho''| + \sup_{]0,1[} |\rho'''| \right). \tag{5.24}$$

Hierbei sind die c_i *Konstanten und*

$$\nabla_{h,j} \psi_h := (\nabla_{h,x} \psi_h(R:\cdot))_j ,$$

wobei Werte außerhalb Ω_h *(bzw.* G_h*, siehe Kapitel 3) wie im sechsten Schritt trilinear interpoliert werden. Falls die entsprechenden Ableitungen existieren, gilt außerdem:*

$$\left| \frac{\partial^2 \psi}{\partial x_k^2}(R:x) \right| \leq c_5 \frac{(1 + |\ln R|) \sup |\rho| + \sup |\rho'|}{R^3} , \tag{5.25}$$

$$\left| \frac{\partial^3 \psi}{\partial x_j x_k^2}(R:x) \right| \leq c_6 \frac{(1 + |\ln R|) \sup |\rho'| + \sup_{r \in]0,1[} |\rho'(r)/r| + \sup |\rho''|}{R^4} . \tag{5.26}$$

Der Beweis dieses Satzes findet sich in [62]. Wie man leicht nachrechnen kann, befindet sich bei festem $0 < h < 1$ das absolute Minimum der Funktion $a - \ln(ah)$ (für $0 < a$) über $a = 1$. Es beträgt $1 - \ln(h)$.

Die folgende Diskussion der Konvergenzordnung wird zuerst für konstantes R geführt. Wenn jeweils der erste Term in den obigen Abschätzungen überwiegt, handelt es sich bei dem oben vorgestellten Algorithmus etwa um ein $O(h^2(1 - \ln(h)))$-Verfahren hinsichtlich der Konvergenz. Sind beide Terme ungefähr gleichgewichtig, verbessert sich dies auf $O(h^2(-\ln(h)))$, und bei einem Überwiegen des zweiten Termes auf $O(h^2)$ (für $h < 1/e$ ist $h^2(-\ln(h))$ größer als h^2). Da $h^2(1 - \ln(h)) < h$ für die betrachteten h ist, konvergiert das Verfahren also schneller als linear $(O(h))$.

Ist nun h konstant und wird das Verhalten obiger Abschätzungen für variables R betrachtet, muß man beachten, daß jeweils der zweite Term der rechten Seite von (5.23) bzw. (5.24) zwar proportional zu R^{-3} bzw. R^{-4} ist, die Abhängigkeit des jeweiligen ersten Terms von R aber nicht so deutlich zu erkennen ist. Die Gleichungen (5.25) und (5.26) geben lediglich einen Hinweis auf das Verhalten von ψ_h.

Außerdem fehlt aufgrund der Annahme $\forall x \in \Omega_h : f_h(x) = f(x)$ noch die Berücksichtigung der Verteilung der Ladungen auf die jeweils umliegenden

acht Gitterpunkte (siehe Abschnitt 5.3.1, zweiter Schritt). Dies hat Auswirkungen auf die Abhängigkeit der Fehler von h und R.

Die Auswirkungen der Wahl verschiedener R und h auf die Ergebnisse werden nun im nächsten Abschnitt anhand einiger Beispiele untersucht.

5.4 Numerische Ergebnisse

Um dem grundlegenden numerischen Verhalten des Verfahrens auf die Spur zu kommen, werden Systeme mit nur zwei bzw. drei Partikeln untersucht. Für R werden Werte aus dem maximal möglichen Intervall $]0, d/2[=]0, 1/2[$ und für h die Werte $\frac{1}{32}$, $\frac{1}{64}$ und $\frac{1}{128}$ gewählt. Die getesteten Beispielsysteme finden sich am Ende des Abschnitts.

5.4.1 Das Verhalten des verwendeten Mehrgitter-Verfahrens

Zur Berechnung der Funktion ψ_h werden entsprechend der Ergebnisse in Kapitel 3 die Programme ω_2-FR3D oder FR3D verwendet. Bei allen getesteten Beispielen ergeben sich durchschnittliche Konvergenzraten, die vor allem für kleine R etwas schlechter als die in Kapitel 3 angegebenen sind. Hier macht es sich bemerkbar, daß $f \notin C^2([0,1]^3)$ ist und im Grenzfall $R \to \infty$ die Funktion f, also die rechte Seite der zu lösenden Poissongleichung (5.18), Singularitäten in den p_i aufweist. Die Raten liegen aber in jedem Fall *unter* den asymptotischen Konvergenzraten ρ^∞ und sind beispielsweise bei der Verwendung von ω_2-FR3D-W(2,1) kleiner als 0.05.

Die entsprechenden FMG-Versionen der beiden Programme verhalten sich wie in Abschnitt 3.2.2 beschrieben. Insbesondere ist für ω_2-FR3D-W(2,1) und FR3D im W-Cycle $r = 1$ ausreichend, um die verlangte Genauigkeit zu erreichen (siehe Bedingung (3.8)). Die im Abschnitt 5.3.2 gemachte Annahme über den Rechenaufwand W_3 trifft also zu.

5.4.2 Das Fehlerverhalten von E_h und F_h

Um Referenzwerte für E und die F_i zu erhalten, wird (5.1) mit großen S und mittels eines Cut-Off-Schemas (vergleiche 4.3.2) ausgewertet. Die jeweiligen minimalen Fehler $|E - E_h|$ liegen dann bei den Beispielen zwischen $2 \cdot 10^{-5}$ und $7 \cdot 10^{-4}$, die $\max_i ||F_i - F_{h,i}||_\infty$ zwischen $4 \cdot 10^{-5}$ und $2 \cdot 10^{-3}$, wobei hinsichtlich der Anzahl λ der Teilchen kein (negativer) Trend zu erkennen ist.

Grundsätzlich ist zu beobachten, daß bei gegebenem System und festem h die Auftragungen von

$$\ln |E - E_h| \text{ gegen } \ln (R) \tag{5.27}$$

bzw. von

$$\ln \max_i ||F_i - F_{h,i}||_\infty \text{ gegen } \ln (R)\,, \tag{5.28}$$

die zur Ermittlung der Konvergenzordnungen bezüglich R dienen, Knicke oder zumindest deutliche Steigungsänderungen aufweisen (siehe Abbildungen 5.1, 5.2 und 5.3), deren Anzahl gleich der Anzahl der verschiedenen

$|p_i - p_j|_d$ $(i \neq j)$ ist. Unter der Annahme, daß nicht zwei verschiedene Punktepaare den gleichen Abstand aufweisen, tritt die maximal mögliche Zahl von Knicken auf, nämlich

$$\left(\begin{array}{c} \lambda \\ 2 \end{array} \right) = \frac{\lambda(\lambda - 1)}{2} .$$

Der Grund für das Auftreten dieser Knicke ist darin zu sehen, daß die Funktion f für ein $R > |p_i - p_j|_d$ gegenüber einem $R < |p_i - p_j|_d$ eine auch qualitativ andere Gestalt hat. Je mehr Abstände $|p_i - p_j|_d$ der Radius R übersteigt, desto mehr Überschneidungen der Ladungsverteilungsfunktionen $\rho(|x - p_i|_d/R)$ $(j = 1, \ldots, \lambda)$ weist f auf. Dies schlägt sich letztendlich in den Abschätzungen nieder, wie im Beweis des Satzes 4 in [62] zu erkennen ist.

Das Fehlerverhalten von E_h in Abhängigkeit von R

Die Auftragungen (5.27) für 2-Teilchen-Systeme ergeben Kurven, deren Steigungen für $R < |p_1 - p_2|_d$ etwa bei -3 liegen, für $R > |p_1 - p_2|_d$ dagegen etwa bei -4 (siehe Abbildung 5.1). Das bedeutet, daß sich die Fehler bei konstantem h wie R^{-3} bzw. R^{-4} verhalten. Bei diesen und den folgenden Aussagen muß beachtet werden, daß für sehr kleine R, d.h. für $R \leq 0.1$, Abweichungen auftreten können.

Bei 3-Teilchen-Systemen verwischen die Knicke etwas, wenn sich die Werte $|p_i - p_j|_d$ über den möglichen Bereich verteilen. Die Systeme zeigen aber erwartungsgemäß nur einen Knick bei $|p_{i_0} - p_{j_0}|_d$, wenn dieser Wert etwas weiter von 0.5 entfernt ist und die zwei anderen $|p_i - p_j|_d$ nahe 0.5 liegen $(R < d/2 = 0.5)$. Auch bei 3-Teilchen-Systemen besitzen die Kurven Steigungen zwischen -3 und -4, in einigen Fällen in der Nähe von 0.5 auch noch etwas kleinere (-4.5).

Das Fehlerverhalten von F_h in Abhängigkeit von R

Die Kurven (5.28) zeigen dagegen kein einheitliches Verhalten hinsichtlich der Steigungen. Die Systeme lassen sich in Bezug auf die Gestalt ihrer Graphen in zwei Gruppen teilen. Die erste Gruppe besteht aus Systemen, deren p_i Gitterpunkte darstellen („Gitterpunktsysteme“). Hier entfällt die Verteilung einer Ladung q_i auf die umliegenden acht Gitterpunkte. Die zweite Gruppe besteht aus den übrigen Systemen. Hier findet mindestens eine Ladungsverteilung statt.

Bei allen getesteten 2-Teilchen-Systemen existiert erwartungsgemäß ein Knick bzw. ein Sprung der Kurven bei $|p_1 - p_2|_d$. Falls es sich um ein Gitterpunktsystem handelt, liegt für $R < |p_1 - p_2|_d$ die Steigung der Kurve (5.28) zwischen -1 und 0, bei $R = |p_1 - p_2|_d$ macht die Kurve einen Sprung nach unten, und für $R < |p_1 - p_2|_d$ zeigen sich negative Steigungen etwa in der Größenordnung -2 (vergleiche Abbildung 5.2).

Liegt kein Gitterpunktsystem vor (vergleiche Abbildung 5.3), besitzt die Kurve (5.28) einen Knick bei $R = |p_1 - p_2|_d$. Links davon ergeben sich negative Steigungen, die für kleine R zwischen -4 und -5 liegen. Dementsprechend verhalten sich die Fehler wie R^{-4} bzw. R^{-5}. Auf der rechten Seite

der Knickstelle steigt die Kurve erst kurz bis zu einem Maximum, um dann wieder mit ähnlichen Steigungen zu fallen. Dank des lokalen Minimums bei $|p_1 - p_2|_d$, ist dieses R eine gute Wahl, falls es nicht zu nah an 0.5 liegt. Denn auch wenn für große R der Fehler wieder etwas kleiner wird, ist doch bei $R = |p_1 - p_2|_d$ der Rechenaufwand dann letztlich geringer.

3-Teilchen-Systeme weisen bis zu drei Knicke bzw. Sprünge auf. Für die Umgebung jeder dieser Stellen gilt das für 2-Teilchen-Systeme Gesagte.

Diese Ergebnisse zeigen, daß die Abschätzungen (5.23) und (5.24) im Satz 4 mit Vorsicht betrachtet werden müssen. Für den ersten Term der jeweiligen rechten Seite darf man nicht leichtfertig die Abschätzungen (5.25) bzw. (5.26) verwenden und erwarten, daß sich die Fehler überall wie R^{-3} bzw. R^{-4} verhalten. Außerdem wurde in Satz 4 die Ladungsverteilung noch nicht berücksichtigt (s.o.).

Tatsächlich muß bei E_h für „große“ R eher mit einer Proportionalität zu R^{-4} und bei F_h für nicht zu große R (bei Nicht-Gitterpunktsystemen) mit einer Proportionalität zu R^{-5} gerechnet werden. Gerade die letzte Beobachtung ist wichtig, da besonders die Fehlerentwicklung bei kleinen R interessant ist, um den Nutzen einer Absenkung des Rechenaufwandes bei der Wahl eines kleineren R zu beurteilen (siehe Abschnitt 5.3.2).

Das Fehlerverhalten von E_h und F_h in Abhängigkeit von h

Deutlich ist bei Betrachtung der Fehlerreduktionsraten in Abhängigkeit von h das Verhalten von Gitterpunkt- gegenüber Nicht-Gitterpunktsystemen zu erkennen. Die Gitterpunktsysteme weisen sowohl für E_h als auch F_h bei Halbierung der Maschenweite h (von $\frac{1}{32}$ auf $\frac{1}{64}$, dann auf $\frac{1}{128}$) eine Fehlerreduktionsrate von 0.25 auf (siehe Tabelle 5.1). Dieses Ergebnis ist zu erwarten, da keine Ladungsverteilung stattfindet, die Näherung ψ_h mit einem Mehrgitter-Verfahren und die Näherung für den Gradienten von ψ mit einem Verfahren ebenfalls zweiter Ordnung (unter geeigneten Voraussetzungen) berechnet wird.

Bei Nicht-Gitterpunktsystemen dagegen ergeben sich teilweise deutlich von 0.25 abweichende Raten. Die getesteten Systeme zeigen für E_h Raten etwa zwischen 0.27 und 0.51 beim Übergang von $\frac{1}{32}$ zu $\frac{1}{64}$ und Raten etwa zwischen 0.17 und 0.27 beim Übergang von $\frac{1}{64}$ zu $\frac{1}{128}$ (siehe Tabellen 5.2 und 5.4). Dabei verändern sich die Raten für ein gegebenes System in Abhängigkeit von R kaum. Die in Abschnitt 5.3.3 angegebenen und diskutierten Abschätzungen lassen bezüglich h auch bei Berücksichtigung der logarithmischen Terme Raten zwischen 0.25 und 0.30 (bei den hier betrachteten h) erwarten. Hier treten also mit 0.51 deutliche Abweichungen auf. Allerdings verhalten sich die Systeme mindestens linear, und für kleinere h scheinen die Raten Werte um 0.25 zu erreichen.

Für F_h sieht das Verhalten wieder anders aus. Betrachtet man die Abhängigkeit der Raten von R bei Nicht-Gitterpunktsystemen, so zeigen sich Sprünge der entsprechenden Kurven bei den $|p_i - p_j|_d$. Die Raten für $(\frac{1}{32}, \frac{1}{64})$ und $(\frac{1}{64}, \frac{1}{128})$ bei festem R liegen je nach System zwischen 0.12 und 0.35 (siehe Tabellen 5.3 und 5.4). Dabei steigen sie mit fallendem R, ausgehend von

etwa 0.20 bis 0.30, geringfügig bis zum ersten Sprung und liegen dann oft auf niedrigerem Niveau. Dies wiederholt sich für die Umgebungen der anderen $|p_i - p_j|_d$. Für kleine R (bis zu einer gewissen Untergrenze etwa bei 0.1) ergeben sich schließlich Raten der Größenordnung 0.15. Im ganzen bleiben die auftretenden Raten (bis 0.35) im Rahmen der erwarteten (s.o.).

5.4.3 Der Einfluß der Ladungsverteilung auf die Genauigkeit

Beobachtet man das Verhalten des Verfahrens, wenn ein Gitterpunktsystem etwas verschoben wird, so daß es zum Nicht-Gitterpunktsystem wird und somit Ladungsverteilungen auf Gitterpunkte nötig sind, stellt man fest, daß die minimalen Fehler größer werden (bei gegebenem h und jeweils (!) günstigstem R).

Die Fehler bei festem h *und* R werden bei E_h durchgängig größer: Diese Fehler-Vergrößerungsrate ist beinahe unabhängig von R, wird aber bei kleinerem h größer. Für $h = \frac{1}{32}$ liegt sie zwischen 1 und 2, für $h = \frac{1}{64}$ knapp unter 3.5 und für $h = \frac{1}{128}$ knapp über 3.5 (siehe Tabelle 5.5).

Anders ist das Verhalten bei F_h. Wie schon bei den vorherigen Betrachtungen existieren Knicke oder Sprünge. An diesen Stellen zeigen sich in den getesteten Beispielen sehr kleine Raten zwischen 0.5 und 1, d.h. trotz nötiger Ladungsverteilung sind die Fehler etwas kleiner als beim entsprechenden Gitterpunktsystem bei gleichem R und h (siehe Abbildung 5.4). Die anderen Raten sind wesentlich größer (siehe Tabelle 5.5). Das Verhalten ist durch die schon oben beschriebene unterschiedliche Gestalt der Kurven (5.28) für Gitterpunkt- bzw. Nicht-Gitterpunktsysteme zu erklären, wie auch in Abbildung 5.4 deutlich zu erkennen ist. Ein kleines (aber nicht zu kleines) $|p_i - p_j|_d$ als R zu verwenden, erweist sich bei festem h wie schon oben als gute Wahl.

Der Unterschied der Fehler $|E - E_h|$ für das Gitterpunktsystem und ein verschobenes System (analog für $\max_i ||F_i - F_{h,i}||_\infty$) jeweils für festes h und R ist ein Maß für den durch die Ladungsverteilung induzierten Fehler. Die Reduzierung dieses Wertes (für festes R) bei Halbierung der Maschenweite h gestaltet sich aber nicht einheitlich. Für E_h ist beim Übergang von $h = \frac{1}{32}$ zu $\frac{1}{64}$ die Rate fast 1, beim Übergang von $h = \frac{1}{64}$ zu $\frac{1}{128}$ dagegen zwischen 0.25 und 0.30 (siehe Tabelle 5.6). Für F_h sind diese Raten wieder R-abhängig, und wie oben treten auch hier Sprünge auf. Der Grund dafür ist wieder die verschiedene Gestalt der Kurven (5.28) für Gitterpunktsysteme gegenüber Nicht-Gitterpunktsystemen (s.o.). Beim Übergang von $h = \frac{1}{32}$ zu $\frac{1}{64}$ liegen die Raten für R rechts von der Sprungstelle (falls nur eine existiert) bei etwa 0.35, beim Übergang von $\frac{1}{64}$ zu $\frac{1}{128}$ aber schon bei 0.25. Für R links von der Sprungstelle sind die Raten meist (deutlich) kleiner, dafür aber die durch die Ladungsverteilung induzierten Fehler größer.

Das Verhalten der trilinear gewichteten Ladungsverteilung nähert sich für kleinere h also dem eines Verfahrens zweiter Ordnung (auch wenn f nur stückweise zweimal stetig differenzierbar ist).

5.4.4 Beispielsysteme

Im folgenden sind die verwendeten Beispielsysteme angegeben. Hier gilt überall $|p_i - p_j|_d = |p_i - p_j|$.

Beispiel (1)

$$\begin{aligned} p_1 &= (0.25, 0.25, 0.25), \quad q_1 = -1 \\ p_2 &= (0.50, 0.50, 0.50), \quad q_2 = +1 \\ |p_1 - p_2| &= 0.433013 \\ E &= -0.209831 \\ F_1 &= (+0.187695, +0.187695, +0.187695) \\ F_2 &= (-0.187695, -0.187695, -0.187695) \end{aligned}$$

Dies ist ein Beispiel für ein einfaches, „symmetrisches“ Gitterpunktsystem.

Beispiel (2)

$$\begin{aligned} p_1 &= (0.4, 0.5, 0.3), \quad q_1 = -1 \\ p_2 &= (0.1, 0.4, 0.1), \quad q_2 = +1 \\ |p_1 - p_2| &= 0.374166 \\ E &= -0.234778 \\ F_1 &= (-0.352445, -0.134972, -0.258911) \\ F_2 &= (+0.352445, +0.134972, +0.258911) \end{aligned}$$

Bei diesem und dem folgenden Beispiel muß dagegen die Ladungsverteilung durchgeführt werden.

Beispiel (3)

$$\begin{aligned} p_1 &= (0.63, 0.72, 0.81), \quad q_1 = -3 \\ p_2 &= (0.54, 0.45, 0.68), \quad q_2 = +3 \\ |p_1 - p_2| &= 0.312890 \\ E &= -2.435278 \\ F_1 &= (-1.932359, -5.420910, -2.771251) \\ F_2 &= (+1.932359, +5.420910, +2.771251) \end{aligned}$$

Beispiel (4)

$$\begin{aligned} p_1 &= (0.5000, 0.6250, 0.6875), \quad q_1 = -1 \\ p_2 &= (0.1250, 0.5625, 0.8125), \quad q_2 = -1 \\ p_3 &= (0.4375, 0.8750, 0.7500), \quad q_2 = +2 \\ |p_1 - p_2| &= 0.400195 \\ |p_1 - p_3| &= 0.265165 \\ |p_2 - p_3| &= 0.446339 \\ E &= -0.813150 \\ F_1 &= (-0.204811, +2.013896, +0.362801) \\ F_2 &= (+0.089675, +0.317030, +0.042171) \\ F_3 &= (+0.115136, -2.330927, -0.404972) \end{aligned}$$

Hierbei handelt es sich um ein 3-Teilchen-Gitterpunktsystem.

Beispiel (5)

$$
\begin{aligned}
p_1 &= (0.5000, 0.6250, 0.6875), \quad q_1 = -1 \\
p_2 &= (0.4375, 0.8750, 0.7500), \quad q_2 = +1 \\
|p_1 - p_2| &= 0.265165 \\
E &= -0.312426 \\
F_1 &= (-0.251986, +0.971045, +0.251986) \\
F_2 &= (+0.251986, -0.971045, -0.251986)
\end{aligned}
$$

Dieses Beispiel stellt ein Teilsystem von (4) dar, wobei das Punktepaar mit dem kleinsten Abstand ausgewählt wurde.

Beispiel (6)

$$
\begin{aligned}
p_1 &= (0.5050, 0.6300, 0.6925), \quad q_1 = -1 \\
p_2 &= (0.4425, 0.8800, 0.7550), \quad q_2 = +1
\end{aligned}
$$

Die restlichen Werte sind mit denen in Beispiel (5) identisch, weil das System lediglich verschoben wurde.

Beispiel (7)

$$
\begin{aligned}
p_1 &= (0.4950, 0.6200, 0.6825), \quad q_1 = -1 \\
p_2 &= (0.1200, 0.5575, 0.8075), \quad q_2 = -1 \\
p_3 &= (0.4325, 0.8700, 0.7450), \quad q_2 = +2
\end{aligned}
$$

Die restlichen Werte sind mit denen in Beispiel (4) identisch, weil das System lediglich verschoben wurde.

Beispiel (8)

$$
\begin{aligned}
p_1 &= (0.4, 0.70, 0.6), \quad q_1 = -1 \\
p_2 &= (0.4, 0.80, 0.7), \quad q_2 = -1 \\
p_3 &= (0.2, 0.65, 0.4), \quad q_2 = +2 \\
|p_1 - p_2| &= 0.141421 \\
|p_1 - p_3| &= 0.287228 \\
|p_2 - p_3| &= 0.390513 \\
E &= -0.468426 \\
F_1 &= (-1.220009, -3.093772, -4.000676) \\
F_2 &= (-0.448825, +2.434477, +2.175354) \\
F_3 &= (+1.668834, +0.659295, +1.825322)
\end{aligned}
$$

Dieses Beispiel stellt ein 3-Teilchen-System dar, bei dem die Ladungsverteilung durchgeführt werden muß.

5.4.5 Zusammenfassung

Die in [62] vertretene Meinung, die Konvergenzordnungen des Verfahrens seien im wesentlichen $O(h^2R^{-3})$ für E_h und $O(h^2R^{-4})$ für F_h, stellt lediglich eine grobe Tendenz dar. Tatsächlich geben beide Ausdrücke vor allem nicht die Knicke bzw. Sprünge der Fehlerkurven für F_h sowie die stellenweise deutlichen Abweichungen von h^2 bzw. R^{-3} und R^{-4} wieder. Für kleine h

($\leq \frac{1}{128}$) scheint sich eine Proportionalität zu h^2 oder zu einem nur etwas schlechteren Wert (Auswirkung logarithmischer Terme) einzustellen, bzgl. R bleiben aber wohl auch bei kleineren h Steigungsänderungen und Abweichungen bei E_h (R^{-4}) oder bei F_h (R^{-5}) je nach Größe von R bestehen.

5.5 Fazit und Ausblick

Die neue Methode stellt eine Alternative zu den bisherigen Verfahren dar, muß allerdings noch optimiert werden. Hinsichtlich des Rechenaufwandes ist der vorgestellte Algorithmus zwar ein $O(\lambda)$-Verfahren, jedoch ist die Proportionalitätskonstante noch zu groß. Auch die Genauigkeit sollte noch etwas verbessert werden, um (bei gesenktem Rechenaufwand) in Konkurrenz zu den in Kapitel 4 vorgestellten FFT- und Multipole-Verfahren treten zu können. Einige Möglichkeiten zur Verbesserung des Algorithmus werden im folgenden genannt.

Der Vorteil der Verwendung eines Mehrgitter-Verfahrens wie (ω_2-)FR3D anstatt etwa einer dreidimensionalen Fast-Fourier-Transformation besteht vor allem darin, daß es sehr gut parallelisiert werden kann (siehe [60]). Dies trifft ebenfalls auf den aufwendigen zweiten Schritt der Berechnung (siehe 5.3.1) zu, wenn man das System nicht jeweils von einem p_i, sondern von einem Gitterpunkt aus betrachtet, nacheinander die Abstände (gemessen in $|\cdot|_d$) zu den p_i bzw. den acht um p_i liegenden Gitterpunkten berechnet und die entsprechenden ρ_{hilf}, jeweils gewichtet mit dem Ladungsanteil, addiert. Die Parallelisierung sollte in einem optimierten Algorithmus zur Verkürzung der Rechenzeit genutzt werden.

Um möglichst genaue Werte bei kleinem Rechenaufwand zu erhalten, sollten vor allem bei größeren Systemen noch Untersuchungen zur Wahl von R (siehe Abschnitt 5.4) und ρ durchgeführt werden.

Weiterhin ist es möglich, die Genauigkeit der Werte E_h und F_h zu erhöhen, indem zur Berechnung von ψ beispielsweise ein Mehrgitter-Verfahren mit einem kompakten 19-Punkte-Differenzenschema vierter Ordnung (3D-Mehrstellendiskretisierung (fourth-order scheme, FOS), siehe [60, 67]), Full Weighting und Red-Black-Gauß-Seidel verwendet wird. Bei der Verwendung von Diskretisierungen höherer Ordnung kann auch an ein Defekt-Korrektur-Verfahren (siehe [60]) gedacht werden.

Außerdem sollte zusätzlich versucht werden, ein anderes Verfahren zur Ladungsverteilung, d.h. letztlich zur Berechnung von f_h, zu finden, das zwar genauer, aber nicht wesentlich aufwendiger als die Verteilung auf die jeweils umliegenden acht Gitterpunkte mittels trilinearer Wichtungsfaktoren s_i ist. Vielleicht erweisen sich auch in den Umgebungen der p_i lokal verfeinerte Gitter als einsetzbar.

Zur Ausdehnung des Verfahrens auf realistischere Systeme müssen in der Berechnung zusätzlich noch beispielsweise die Beiträge der Bindungslängen, der Valenz- und Torsionswinkel und der van der Waals- und Dipol-Wechselwirkungen zu E und F_i berücksichtigt werden, außerdem die Bewegung der Moleküle im Raum. Die Möglichkeit, sehr genaue Werte der elektrostatischen Größen für jeden Zeitpunkt erhalten zu können, ist entscheidend wichtig bei

der Betrachtung realistischer Systeme, die oft eine starke Zeitabhängigkeit zeigen.

Insbesondere für sehr große Teilchensysteme ist dann ein intensiver Vergleich mit dem Verhalten effizienter aktueller, eventuell parallelisierter Verfahren nötig, um zu entscheiden, ob die neue, optimierte Methode eine Konkurrenz hinsichtlich Genauigkeit *und* Rechenaufwand darstellt.

Bemerkung: Die FORTRAN-Programme aus den Kapiteln 2 und 3 sowie das Programm CHEM (siehe Abschnitt 5.3.1) sollen im Internet unter

http://www.gmd.de/SCAI/

zur Verfügung gestellt werden.

5.6 Tabellen und Abbildungen

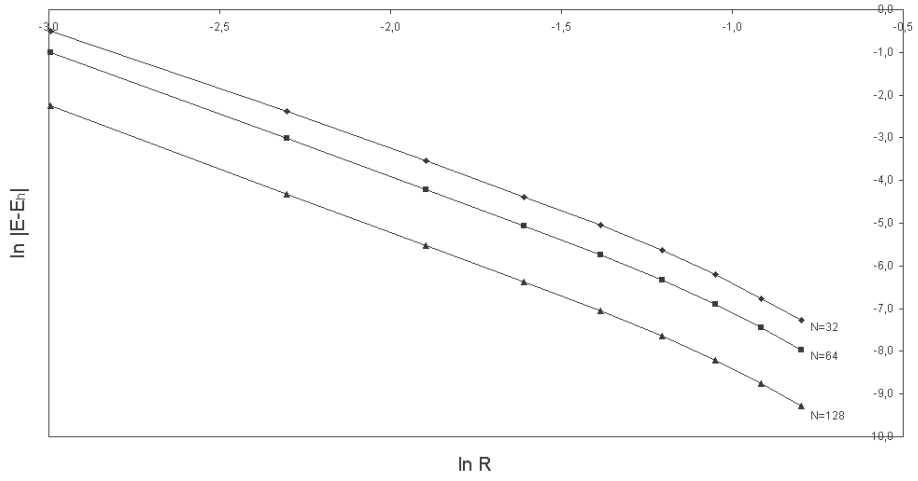


Abbildung 5.1: Die Auftragung (5.27) für Beispiel (6). Die Graphen für die anderen Beispiele sehen ähnlich aus.

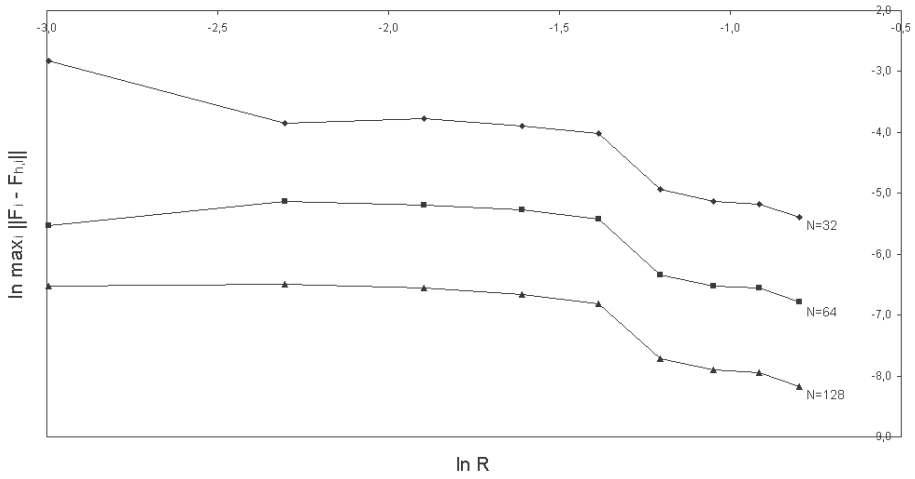


Abbildung 5.2: Die Auftragung (5.28) für Beispiel (4). Die Graphen für die Beispiele (1) und (5) sehen ähnlich aus.

R	$R_{E,1}$	$R_{E,2}$	$R_{F,1}$	$R_{F,2}$
0,45	0,2494	0,2498	0,2504	0,2504
0,40	0,2494	0,2499	0,2499	0,2500
0,35	0,2492	0,2498	0,2496	0,2501
0,30	0,2491	0,2497	0,2478	0,2501
0,25	0,2489	0,2497	0,2429	0,2513
0,20	0,2481	0,2494	0,2528	0,2495
0,15	0,2452	0,2500	0,2422	0,2560
0,10	0,2425	0,2484	0,2793	0,2575
0,05	0,2392	0,2430	0,0672	0,3701

Tabelle 5.1: Raten für Beispiel (4). Dabei bedeuten $R_{E,1} := \frac{|E-E_{1/64}|}{|E-E_{1/32}|}$, $R_{E,2} := \frac{|E-E_{1/128}|}{|E-E_{1/64}|}$, $R_{F,1} := \frac{\max_i ||F_i-F_{1/64,i}||_\infty}{\max_i ||F_i-F_{1/32,i}||_\infty}$ $R_{F,2} := \frac{\max_i ||F_i-F_{1/128,i}||_\infty}{\max_i ||F_i-F_{1/64,i}||_\infty}$ (auch in den Tabellen 5.2, 5.3 und 5.4).

	Bsp. 2		Bsp. 3		Bsp. 6		Bsp. 7	
R	$R_{E,1}$	$R_{E,2}$	$R_{E,1}$	$R_{E,2}$	$R_{E,1}$	$R_{E,2}$	$R_{E,1}$	$R_{E,2}$
0,45	0,3694	0,1731	0,3148	0,2501	0,5014	0,2696	0,5155	0,2702
0,40	0,3653	0,1736	0,3177	0,2456	0,4992	0,2696	0,5115	0,2700
0,35	0,3615	0,1741	0,3218	0,2401	0,4979	0,2696	0,5079	0,2700
0,30	0,3599	0,1751	0,3288	0,2316	0,4974	0,2697	0,5045	0,2700
0,25	0,3598	0,1760	0,3374	0,2239	0,4949	0,2698	0,5014	0,2700
0,20	0,3599	0,1769	0,3442	0,2192	0,4990	0,2703	0,5030	0,2704
0,15	0,3642	0,1776	0,3530	0,2166	0,5082	0,2710	0,5107	0,2710
0,10	0,3741	0,1797	0,3684	0,2173	0,5278	0,2738	0,5284	0,2738
0,05	0,4198	0,1894	0,4365	0,2279	0,6042	0,2880	0,6030	0,2880

Tabelle 5.2: Raten für die Beispiele (2), (3), (6) und (7).

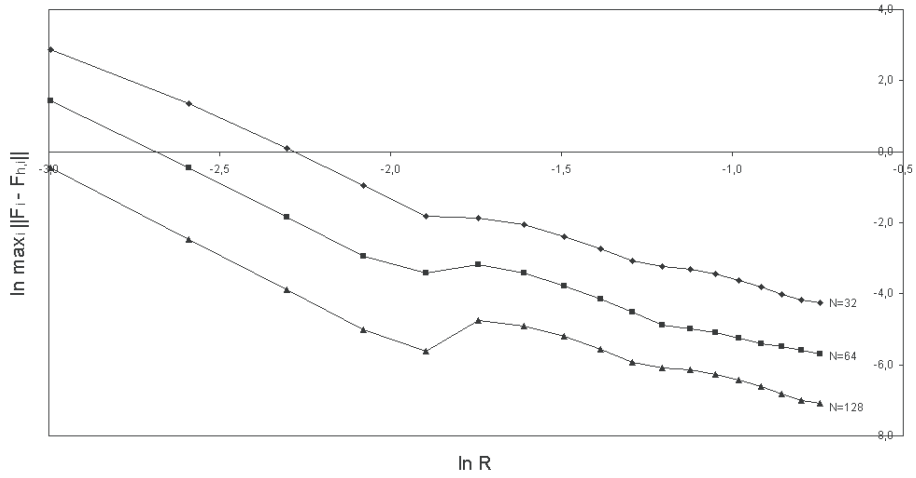


Abbildung 5.3: Die Auftragung (5.28) für Beispiel (8). Die Graphen für die Beispiele (2), (3), (6) und (7) sehen ähnlich aus.

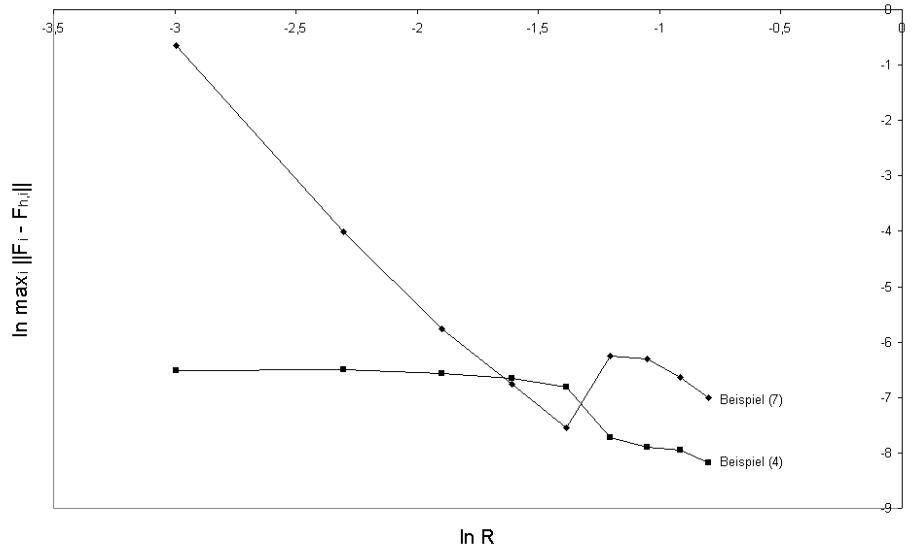


Abbildung 5.4: Die Auftragung (5.28) für Beispiele (4) und (7) für $h = \frac{1}{128}$.

	Bsp. 2		Bsp. 3		Bsp. 6		Bsp. 7	
R	$R_{F,1}$	$R_{F,2}$	$R_{F,1}$	$R_{F,2}$	$R_{F,1}$	$R_{F,2}$	$R_{F,1}$	$R_{F,2}$
0,45	0,2490	0,2496	0,2515	0,2396	0,3142	0,2500	0,3112	0,2478
0,40	0,2496	0,2495	0,2571	0,2346	0,3185	0,2504	0,3151	0,2499
0,35	0,2497	0,2500	0,2729	0,2215	0,3251	0,2506	0,3258	0,2555
0,30	0,2482	0,2504	0,2719	0,2914	0,3460	0,2504	0,3545	0,2626
0,25	0,2529	0,2505	0,2265	0,2808	0,1431	0,2460	0,1356	0,2158
0,20	0,2492	0,2513	0,1685	0,2096	0,1694	0,1988	0,1543	0,1727
0,15	0,2668	0,2425	0,1426	0,1518	0,1547	0,1520	0,1470	0,1357
0,10	0,2244	0,2414	0,1438	0,1300	0,1528	0,1230	0,1508	0,1183
0,05	0,0711	0,1730	0,1977	0,1394	0,2209	0,1289	0,2221	0,1282

Tabelle 5.3: Raten für die Beispiele (2), (3), (6) und (7).

R	$R_{E,1}$	$R_{E,2}$	$R_{F,1}$	$R_{F,2}$
0,475	0,2776	0,2245	0,2366	0,2495
0,450	0,2782	0,2242	0,2425	0,2406
0,425	0,2791	0,2238	0,2285	0,2668
0,400	0,2801	0,2233	0,2023	0,3008
0,375	0,2812	0,2224	0,1997	0,3039
0,350	0,2824	0,2217	0,1957	0,3088
0,325	0,2835	0,2211	0,1891	0,3174
0,300	0,2844	0,2208	0,1939	0,3015
0,275	0,2850	0,2205	0,2381	0,2448
0,250	0,2863	0,2201	0,2454	0,2429
0,225	0,2879	0,2195	0,2495	0,2378
0,200	0,2897	0,2188	0,2574	0,2283
0,175	0,2919	0,2184	0,2740	0,2081
0,150	0,2956	0,2185	0,2023	0,1093
0,125	0,2996	0,2219	0,1374	0,1268
0,100	0,3025	0,2243	0,1445	0,1300
0,075	0,3098	0,2291	0,1631	0,1350
0,050	0,3485	0,2372	0,2423	0,1487

Tabelle 5.4: Raten für Beispiel (8).

R	$R_{E,1/32}$	$R_{E,1/64}$	$R_{E,1/128}$	$R_{F,1/32}$	$R_{F,1/64}$	$R_{F,1/128}$
0,45	1,4134	2,9215	3,1590	2,6000	3,2316	3,1976
0,40	1,4674	3,0093	3,2517	2,9279	3,6917	3,6895
0,35	1,5277	3,1136	3,3647	3,7339	4,8734	4,9785
0,30	1,5934	3,2273	3,4892	2,9148	4,1705	4,3803
0,25	1,6586	3,3404	3,6124	0,9907	0,5529	0,4747
0,20	1,6605	3,3670	3,6508	2,1660	1,3221	0,9153
0,15	1,6071	3,3467	3,6284	6,8692	4,1699	2,2108
0,10	1,4934	3,2543	3,5870	48,2189	26,0376	11,9673
0,05	1,1676	2,9436	3,4888	312,7386	1033,8066	358,1134

Tabelle 5.5: Beispiele (4) und (7) im Vergleich. $R_{E,h}$ und $R_{F,h}$ bezeichnen die (Vergrößerungs-)Raten des Fehlers beim Übergang vom Gitterpunktsystem (4) zum Nichtgitterpunktsystem (7).

R	$R_{E,1}$	$R_{E,2}$	$R_{F,1}$	$R_{F,2}$
0,45	1,1592	0,2807	0,3492	0,2466
0,40	1,0724	0,2800	0,3489	0,2498
0,35	0,9983	0,2795	0,3537	0,2569
0,30	0,9349	0,2791	0,4103	0,2666
0,25	0,8846	0,2787	11,6698	0,2953
0,20	0,8889	0,2793	0,0698	0,0657
0,15	0,9479	0,2800	0,1308	0,0978
0,10	1,1079	0,2851	0,1481	0,1128
0,05	2,7734	0,3112	0,2226	0,1280

Tabelle 5.6: Beispiele (4) und (7) im Vergleich. Die $R_{E,i}$ und $R_{F,i}$ bezeichnen die (Verkleinerungs-)Raten des Unterschiedes zwischen dem Fehler des Wertes E_h bzw. F_h für das Gitterpunktsystem (4) und dem entsprechenden Wert für das Nichtgitterpunktsystem (7) (beim Übergang von 1/32 zu 1/64 ($i = 1$) bzw. von 1/64 zu 1/128 ($i = 2$)).

Anhang A

Ergänzungen

A.1 Beweis der Behauptung (1.32)

In diesem Abschnitt soll die Gültigkeit der Behauptung (1.32) aus Abschnitt 1.2.5 nachgewiesen werden:

$$dim(Kern(\widetilde{C}_h)) = 1\,.$$

Es ist bereits bekannt (siehe 1.2.5), daß die $(N+1) \times (N+1)$-Matrix $\widetilde{C}_h$ singulär ist, daß also

$$Rang(\widetilde{C}_h) \leq N \tag{A.1}$$

gilt. Zeigt man nun noch, daß auch

$$Rang(\widetilde{C}_h) \geq N \tag{A.2}$$

gilt, erhält man insgesamt $Rang(\widetilde{C}_h) = N$ und somit die Gültigkeit der Gleichung (1.32). Zum Beweis von (A.2) werden folgende Lemmata benötigt:

Lemma 4 . *Gegeben sei für $N \in \mathbb{N}_{\geq 5}$ die $(N-3) \times (N-2)$-Matrix*

$$\widetilde{M}_N = \left[\begin{array}{ccccccc} -1 & 2 & -1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ \hline -1 & & & & 0 & 0 & \end{array}\right]\,.$$

Diese Matrix läßt sich durch geeignete Zeilenumformungen auf folgende Gestalt bringen:

$$\widehat{M}_N = \left[\begin{array}{ccccccc} -1 & 2 & -1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ \hline 0 & & & & -(N-3) & N-4 & \end{array}\right]\,.$$

Beweis durch vollständige Induktion über N :

Für $N = 5$ ergibt sich die Behauptung durch einfaches Ausrechnen. Gelte nun die Behauptung für $N \geq 5$. $\widetilde{M}_{N+1}$ hat folgende Gestalt:

$$\widetilde{M}_{N+1} = \left[\begin{array}{ccccccc|c} -1 & 2 & -1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & -1 & 2 & -1 & & \\ \hline & & & & -1 & 2 & -1 & \\ \hline -1 & & & & 0 & 0 & 0 & \end{array}\right] .$$

Offenbar ist $\widetilde{M}_N$ in $\widetilde{M}_{N+1}$ enthalten (nämlich in „zwei Teilen“: links oben und links unten), und zwar so, daß sich $\widetilde{M}_{N+1}$ nach Induktionsvoraussetzung also in folgende Matrix umformen läßt:

$$M_{N+1} = \left[\begin{array}{ccccccc|c} -1 & 2 & -1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & -1 & 2 & & -1 & \\ \hline & & & & -1 & 2 & & -1 \\ \hline 0 & & & & -(N-3) & N-4 & & 0 \end{array}\right] .$$

Eine weitere Umformung der letzten Zeile mit Hilfe der vorletzten liefert dann sofort $\widehat{M}_{N+1}$. □

Lemma 5 . *Gegeben sei nun für $N \in \mathbb{N}_{\geq 5}$ und $h = \frac{1}{N}$ die $(N+1)\times(N+1)$-Matrix*

$$\widetilde{C}_h = \left[\begin{array}{cc|ccccccc|cc} 2 & -1 & & & & & & & & -1 & \\ -1 & 2 & -1 & & & & & & & & \\ \hline & -1 & 2 & -1 & & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & & \\ & & & \ddots & \ddots & \ddots & & & & & \\ & & & & -1 & 2 & -1 & & & & \\ \hline & & & & & -1 & 2 & -1 & & & \\ & & & & & & -1 & 2 & -1 & & \\ \hline & -1 & & & & & 0 & 0 & & -1 & 2 \end{array}\right] .$$

Diese Matrix läßt sich durch geeignete Zeilenumformungen auf folgende Ge-

stalt bringen:

$$\widehat{C}_h = \left[\begin{array}{cc|ccccccc|cc} 2 & -1 & & & & & & & & -1 & \\ -1 & 2 & -1 & & & & & & & & \\ \hline & -1 & 2 & -1 & & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & & \\ & & & \ddots & \ddots & \ddots & & & & & \\ & & & & -1 & 2 & -1 & & & & \\ \hline & & & & & -1 & 2 & -1 & & & \\ & & & & & & -1 & 2 & -1 & & \\ \hline & & 0 & & & & 0 & 0 & & -N & N \end{array}\right] .$$

Beweis: Offenbar ist die Matrix $\widetilde{M}_N$ in $\widetilde{C}_h$ „enthalten“. Daher kann man also $\widetilde{C}_h$ nach obigem in folgende Matrix umformen:

$$\check{C}_h = \left[\begin{array}{cc|ccccccc|cc} 2 & -1 & & & & & & & & -1 & \\ -1 & 2 & -1 & & & & & & & & \\ \hline & -1 & 2 & -1 & & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & & \\ & & & \ddots & \ddots & \ddots & & & & & \\ & & & & -1 & 2 & & -1 & & & \\ \hline & & & & & & -1 & 2 & -1 & & \\ & & & & & & & -1 & 2 & -1 & \\ \hline & & 0 & & & & -(N-3) & N-4 & & -1 & 2 \end{array}\right] .$$

Eliminiert man nun in der letzten Zeile mit Hilfe der drittletzten und dann der vorletzten Zeile den viertletzten und dann den drittletzten Eintrag, so ergibt sich sofort die Matrix $\widehat{C}_h$. □

Aus dem letzten Lemma folgt sofort die Gültigkeit von (A.2), womit also auch (1.32) bewiesen ist. □

A.2 Beweis der Gleichungen (5.4)

Offensichtlich gilt für $x, y \in \mathbb{R}^3$, $x \neq y$ und $n \in \mathcal{N}(d)$:

$$\begin{aligned} G(S:x,y) &= G(S:x+n,y) && \text{(Periodizität)}, \\ G(S:x,y) &= G(S:y,x) && \text{(Symmetrie)}, \\ G(S:x-y,0) &= G(S:x,y) && \text{(Translationsinvarianz)}. \end{aligned} \tag{A.3}$$

Die Funktion G als Grenzwert der $G(S:\cdot,\cdot)$ besitzt offenbar auch diese Eigenschaften, ist also insbesondere d-periodisch und kann daher mit einer Funktion auf dem Torus $T^3(d)$ identifiziert werden. Die Charakterisierung von G durch (5.4) wird nun im folgenden Satz bewiesen:

Satz 5 . *Es gelten folgende Gleichungen:*

$$-\Delta_x G(x,y) = 4\pi \sum_{n\in\mathcal{N}(d)} \chi_{y+n}(x) - \frac{4\pi}{d^3} C(\phi) ,$$
$$\lim_{x\to y}\left(G(x,y) - \frac{1}{|x-y|}\right) = 0 .$$

Beweis: Aus

$$\nabla\frac{1}{|x|} = -\frac{1}{|x|^3}x ,$$
$$\Delta\frac{1}{|x|} = -4\pi\chi_0(x) ,$$
$$\nabla\phi(|x|/S) = \frac{\phi'(|x|/S)}{S|x|}x ,$$
$$\Delta\phi(|x|/S) = \frac{2\phi'(|x|/S)}{S|x|} + \frac{\phi''(|x|/S)}{S^2}$$

erhält man

$$\begin{aligned}\Delta\left(\frac{\phi(|x|/S)}{|x|}\right) &= (\Delta\phi(|x|/S))\frac{1}{|x|} + 2\nabla\phi(|x|/S)\cdot\nabla\frac{1}{|x|} + \phi(|x|/S)\Delta\frac{1}{|x|}\\ &= \frac{\phi''(|x|/S)}{S^2|x|} - \phi(|x|/S)4\pi\chi_0(x)\\ &= \frac{\phi''(|x|/S)}{S^2|x|} - 4\pi\chi_0(x)\end{aligned}$$

wegen $\phi(0) = 1$. Daraus folgt nun

$$\begin{aligned}-\Delta_x G(S:x,y) &= -\sum_{n\in\mathcal{N}(d)}\left(\frac{\phi''(|x-y+n|/S)}{S^2|x-y+n|} - 4\pi\chi_{y-n}(x)\right)\\ &= 4\pi\sum_{n\in\mathcal{N}(d)}\chi_{y-n}(x) - \frac{1}{S^2}\sum_{n\in\mathcal{N}(d)}\frac{\phi''(|x-y+n|/S)}{|x-y+n|} .\end{aligned}$$

Die Berechnung des Grenzwertes des zweiten Termes der rechten Seite ergibt unter Verwendung von (5.2) und der schalenweisen Integration:

$$\begin{aligned}\lim_{S\to\infty}\frac{1}{S^2}\sum_{n\in\mathcal{N}}\frac{\phi''(|z+n|/S)}{|z+n|} &= \lim_{S\to\infty}\frac{1}{d^3}\sum_{\tilde{n}\in(z+\mathcal{N}(d))/S}\frac{\phi''(|\tilde{n}|)}{|\tilde{n}|}\frac{d^3}{S^3}\\ &= \frac{1}{d^3}\int_{\mathbb{R}^3}\frac{\phi''(|x|)}{|x|}dx\\ &= \frac{1}{d^3}4\pi\int_0^\infty r\phi''(r)dr\\ &= \frac{4\pi}{d^3}C(\phi) ,\end{aligned}$$

und insgesamt erhält man

$$\lim_{S\to\infty} -\Delta_x G(S:x,y) = 4\pi \sum_{n\in\mathcal{N}(d)} \chi_{y+n}(x) - \frac{4\pi}{d^3} C(\phi) \,,$$

und damit den ersten Teil der Behauptung. Der zweite Teil folgt aus der Tatsache, daß auch die $G(S:x,y)$ diese Eigenschaft haben. □

Literaturverzeichnis

[1] Adams, D.J. und G.S. Dubey: *Taming the Ewald sum in the computer simulation of charged systems.* J. Comp. Phys. 72, Seiten 156–176, 1987.

[2] Allen, M. und D. Tildesley: *Computer Simulation of Liquids.* Oxford Science, London, 1990.

[3] Alt, H.W.: *Lineare Funktionalanalysis.* Springer, Berlin, 1992.

[4] Anderson, C.R.: *An implementation of the fast multipole method without multipoles.* SIAM J. Sci. Stat. Comput. 13, Seiten 923–947, 1992.

[5] Ashcroft, N.W. und N.D. Mermin: *Solid State Physics.* Saunders College HRW, Philadelphia, USA, 1976.

[6] Aubin, T.: *Some Nonlinear Problems in Riemannian Geometry.* Springer, Berlin, 1998.

[7] Bannasch, F.: *Mehrgitterverfahren für die dreidimensionale Poissongleichung.* Diplomarbeit, Universität zu Köln, Deutschland, Mai 1983.

[8] Barnes, J. und P. Hut: *A hierarchical $O(N \log(N))$ force-calculation algorithm.* Nature 324, Seiten 446–449, 1986.

[9] Becker, K.: *Mehrgitterverfahren zur Lösung der Helmholtz-Gleichung im Rechteck mit Neumannschen Randbedingungen.* Diplomarbeit, Universität Bonn, Deutschland, August 1981.

[10] Belhadj, M., H.E. Alper und R. Levy: *Molecular dynamics simulations of water with Ewald summation for the long range electrostatic interactions.* Chem. Phys. Lett. 179, Seiten 13–20, 1991.

[11] Berman, C.L. und L. Greengard: *A renormalization method for the evaluation of lattice sums.* J. Math. Phys. 35, Seiten 6036–6048, 1994.

[12] Board, J.A., R. Batchelor und J.F. Leathrum. *Proc. AIAA/ASME Thermophysics and Heat Transfer Conf.*, 1990.

[13] Board, J.A., J.W. Causey, J.F. Leathrum, A. Windemuth und K. Schulten: *Accelerated molecular dynamics simulation with the parallel fast multipole algorithm.* Chem. Phys. Lett. 198, Seiten 89–94, 1992.

[14] Brandt, A. und A.A. Lubrecht: *Multilevel matrix multiplication and fast solution of integral equations.* J. Comp. Phys. 90, Seiten 348–370, 1990.

[15] Brush, S.G., H.L. Sahlin und E. Teller: *Monte Carlo study of a one-component plasma. I.* J. Chem. Phys. 45, Seiten 2101–2118, 1966.

[16] Courant, R.: *Vorlesungen über Differential- und Integralrechnung, Band II.* Springer, Berlin, 1963.

[17] Darden, T.A., D. York und L.G. Pedersen: *Particle mesh Ewald: An $N \cdot log(N)$ method for Ewald sums in large systems.* J. Chem. Phys. 98, Seiten 10089–10092, 1993.

[18] Ding, H.-Q., N. Karasawa und W.A. Goddard III: *Atomic level simulations on a million particles: The cell multipole method for Coulomb and London nonbond interactions.* J. Chem. Phys. 97, Seiten 4309–4315, 1992.

[19] Ding, H.-Q., N. Karasawa und W.A. Goddard III: *The reduced cell multipole method for Coulomb interactions in periodic systems with million-atom unit cells.* Chem. Phys. Lett. 196, Seiten 6–10, 1992.

[20] Essmann, U., L. Perera, M.L. Berkowitz, T.A. Darden, H. Lee und L.G. Pedersen: *A smooth particle mesh Ewald method.* J. Chem. Phys. 103, Seiten 8577–8593, 1995.

[21] Esselink, K.: *A comparison of algorithms for longe-range interactions.* Comput. Phys. Commun. 87, Seiten 375–395, 1995.

[22] Ewald, P.P.: *Die Berechnung optischer und elektrostatischer Gitterpotentiale.* Ann. Phys. 64, Seiten 253–287, 1921.

[23] Farouki, R.T. und S. Hamaguchi: *Spline approximation of 'effective' potentials under periodic boundary conditions.* J. Comp. Phys. 115, Seiten 276–287, 1994.

[24] Fincham, D. *Mol. Simulation 13*, Seite 1ff., 1994.

[25] Foerster, H. und K. Witsch: *Multigrid software for the solution of elliptic problems on rectangular domains: MG00 (Release 1).* In: Hackbusch, W. und U. Trottenberg (Herausgeber): *Multigrid Methods*, Band 960 der Reihe *Lecture Notes in Mathematics*, Seiten 427–460. Springer, Berlin, 1982.

[26] Greengard, L.: *The numerical solution of the n-body problem.* Computers in Physics, März/April 1990, Seiten 142–152.

[27] Greengard, L.: *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, MA., 1988.

[28] Greengard, L. und V. Rokhlin: *A fast algorithm for particle simulations.* J. Comp. Phys. 73, Seiten 325–348, 1987.

[29] HACKBUSCH, W.: *A multi-grid method applied to a boundary value problem with variable coefficients in a rectangle.* Report 77-17, Universität zu Köln, Deutschland, 1977.

[30] HACKBUSCH, W.: *Multigrid Methods and Applications.* Springer, Berlin, 1985.

[31] HANSEN, J.P.: *Statistical mechanics of dense ionized matter. I. Equilibrium properties of the classical one-component plasma.* Phys. Rev. A 8, Seiten 3096–3109, 1973.

[32] HAUTMAN, J. und M. KLEIN. *Mol. Sim. 75*, Seite 379ff., 1992.

[33] HELLWIG, G.: *Partial Differential Equations: An Introduction.* B. G. Teubner, Stuttgart, 1977.

[34] HEMKER, P.W.: *On the order of prolongations and restrictions in multigrid procedures.* J. Comp. Appl. Math. 32, Seiten 423–429, 1990.

[35] HEYES, D.M.: *Electrostatic potentials and fields in infinite point charge lattices.* J. Chem. Phys. 74, Seiten 1924–1929, 1981.

[36] HOCKNEY, R.W. und J.W. EASTWOOD: *Computer Simulation using Particles.* McGraw-Hill, New York, 1981.

[37] JACKSON, J.D.: *Classical Electrodynamics.* John Wiley and Sons, Chichester, 1975.

[38] KELLER, H.B.: *On the solution of singular and semidefinite linear systems by iteration.* J. SIAM Numer. Anal. Ser. B, Vol. 2, No. 2, Seiten 281–290, 1965.

[39] KITTEL, C.: *Introduction to Solid State Physics.* John Wiley and Sons, Chichester, 1971.

[40] KLINGENBERG, W.: *Eine Vorlesung über Differentialgeometrie.* Springer, Berlin, 1973.

[41] KOLAFA, J. und J.W. PERRAM. *Mol. Simulation 9*, Seite 351ff., 1992.

[42] LAGE, F.C. VON DER und H.A. BETHE: *A method for obtaining electronic eigenfunctions and eigenvalues in solids with an application to sodium.* Phys. Rev. 71, Seiten 612–622, 1947.

[43] LAMBERT, C., J.A. BOARD und T.A. DARDEN: *A multipole-based algorithm for efficient calculation of forces and potential in macroscopic periodic assemblies of particles.* Technical Report 95-001a, Department of Electrical Engineering, P.O. Box 90291, Durham, NC 27708-0291, 1995.

[44] LUTY, B., I. TIRONI und W. VAN GUNSTEREN: *Lattice-sum methods for calculating electrostatic interactions in molecular simulations.* J. Chem. Phys. 103, Seiten 3014–3021, 1995.

[45] Perram, J.W., H.G. Petersen und S.W. De Leeuw: *An algorithm for the simulation of condensed matter which grows as the $\frac{3}{2}$ power of the number of particles.* Mol. Phys. 65, Seiten 875–893, 1988.

[46] Pollock, E.L. und J. Glasli: *Comments on P^3M, FMM, and the Ewald method for large periodic Coulombic systems.* Comp. Phys. Commun. 95, Seiten 93–110, 1996.

[47] Rajagopal, G. und R.J. Needs: *An optimized Ewald method for long-ranged potentials.* J. Comp. Phys. 115, Seiten 399–405, 1994.

[48] Rhee, Y.-J., J.W. Halley, J. Hautman und A. Rahman: *Ewald methods in molecular dynamics for systems of finite extent in one of three dimensions.* Phys. Rev. B 40, Seiten 36–42, 1989.

[49] Rycerz, Z. und P. Jacobs. *Mol. Simulation 8*, Seite 197ff., 1992.

[50] Sangester, M. und M. Dixon: *Interionic potentials in alkalihalides and their use in simulations of the molten salts.* Adv. Phys. 25, Seiten 247–342, 1976.

[51] Schmidt, K.E. und M.A. Lee: *Implementing the fast multipole method in three dimensions.* J. Stat. Phys. 63, Seiten 1223–1235, 1991.

[52] Schubert, H.: *Topologie.* B.G. Teubner, Stuttgart, 1975.

[53] Schwichtenberg, H., G. Winter und H. Wallmeier: *Acceleration of molecular mechanic simulation by parallelization and fast multipole techniques.* Parallel Computing 25, Seiten 535–546, 1999.

[54] Shimada, J., H. Kaneko und T. Takada: *Efficient calculations of Coulombic interactions in biomolecular simulations with periodic boundary conditions.* J. Comp. Chem. 14, Seiten 867–878, 1993.

[55] Shimada, J., H. Kaneko und T. Takada: *Performance of fast multipole methods for calculating electrostatic interactions in biomacromolecular simulations.* J. Comp. Chem. 15, Seiten 28–43, 1994.

[56] Slattery, W.L., G.D. Doolen und H.E. DeWitt: *Improved equation of state for the classical one-component plasma.* Phys. Rev. A 21, Seiten 2087–2095, 1980.

[57] Solvason, D., J. Kolafa, H.G. Petersen und J.W. Perram: *A rigorous comparison of the Ewald method and the fast multipole method in two dimensions.* Comput. Phys. Commun. 87, Seiten 307–318, 1995.

[58] Stüben, K. und U. Trottenberg: *Multigrid methods: fundamental algorithms, model problem analysis and applications.* In: Hackbusch, W. und U. Trottenberg (Herausgeber): *Multigrid Methods*, Band 960 der Reihe *Lecture Notes in Mathematics*, Seiten 1–176. Springer, Berlin, 1982.

[59] Toukmaji, A.Y. und J.A. Board: *Ewald summation techniques in perspective: A survey.* Comput. Phys. Commun. 95, Seiten 73–92, 1996.

[60] Trottenberg, U., C.W. Oosterlee und A. Schüller: *Multigrid.* Academic Press, London, erscheint 1999/2000.

[61] Warner, F.W.: *Foundations of Differentiable Manifolds and Lie Groups.* Springer, Berlin, 1983.

[62] Washio, T.: *Calculation and its error analysis of the electrostatic potential and the force fields in macroscopic periodic assemblies of particles.* Noch nicht veröffentlicht.

[63] Wesseling, P.: *An Introduction to Multigrid Methods.* John Wiley and Sons, Chichester, 1992.

[64] Wienands, R., C.W. Oosterlee und T. Washio: *Fourier analysis of GMRES(m) preconditioned by multigrid.* SIAM J. Sci. Comput., erscheint 2000.

[65] Yavneh, I.: *On red-black SOR smoothing in multigrid.* SIAM J. Sci. Comput. 17, Seiten 180–192, 1996.

[66] York, D. und W. Yang: *The fast fourier Poisson method for calculating Ewald sums.* J. Chem. Phys. 101, Seiten 3298–3300, 1994.

[67] Zhang, J.: *Fast and high accuracy multigrid solution of the three dimensional Poisson equation.* J. Comp. Phys. 143, Seiten 449–461, 1998.