



Fakultät für Mathematik

Praktikumsbericht

Betriebspraktikum Jülich

Toni Volkmer

Chemnitz, den 7. August 2008

Betreuer: Dr. G. Sutmann
(Forschungszentrum Jülich)

Betr. Hochschullehrer: Prof. Dr. D. Potts
(TU Chemnitz)

1 Aufgabe

Die Laufzeiten von FMM und NFFT-basierte schneller Summation für die Berechnung von elektrischen Potentialen von geladenen Partikeln und der zugehörigen Kräfte im dreidimensionalen Fall sind zu vergleichen. Hierbei sollen verschiedene Partikelverteilungen betrachtet werden. Die Testläufe sollen auf dem Cluster JUMP des Forschungszentrum Jülich durchgeführt werden.

2 Einleitung

Die Berechnung der elektrischen Potentiale

$$\phi_j = \phi(y_j) := \sum_{\substack{i=1 \\ i \neq j}}^N \frac{q_i}{\|y_j - y_i\|}, \quad j = 1, \dots, N \quad (2.1)$$

und der zugehörigen Kräfte

$$F(y_j) = -q_j \nabla \phi(y_j), \quad j = 1, \dots, N \quad (2.2)$$

bzw. der zugehörigen elektrischen Felder

$$E(y_j) = -\frac{1}{4\pi\epsilon_0} \nabla \phi(y_j), \quad j = 1, \dots, N \quad (2.3)$$

von N geladenen Partikeln an den Orten $y_j \in \mathbb{R}^3$ mit den Ladungen $q_j \in \mathbb{R}$ mittels direkter Summation benötigt $\mathcal{O}(N^2)$. Durch approximative Verfahren kann dieser Rechenaufwand jedoch auf $\mathcal{O}(N \log N)$ oder $\mathcal{O}(N)$ reduziert werden.

Hierbei bezeichnet $\|\cdot\|$ die Euklidische Norm, d.h. für $y_j = \left(y_j^{(1)}, y_j^{(2)}, y_j^{(3)}\right)^T \in \mathbb{R}^3$ gilt

$$\|y_j\| = \|y_j\|_2 := \sqrt{y_j^{(1)2} + y_j^{(2)2} + y_j^{(3)2}}.$$

Im Folgenden werden zwei solcher approximativer Verfahren untereinander und gegenüber der direkten Summation im Bezug auf Laufzeit und Genauigkeit verglichen.

3 Überblick zu FMM und NFFT-basierter schneller Summation

3.1 FMM

Die Fast Multipole Method (FMM) ermöglicht die Berechnung von (2.1) und (2.2) in $\mathcal{O}(N)$. Für eine detaillierte Beschreibung wird auf [1, Abschnitt 2.2.2] verwiesen.

Unter [2] ist die FCS library des Jülich Supercomputing Centre des Forschungszentrum Jülich zu finden. Diese enthält verschiedene Algorithmen zur Berechnung von (2.1) und (2.2), darunter die direkte Summation und FMM. Verfügbar ist diese für die Plattformen IBM p690 sowie LINUX IA-32, der Quellcode steht nicht zur Verfügung.

3.2 NFFT-basierte schnelle Summation

Die NFFT-basierte schnelle Summation ermöglicht die Berechnung von (2.1) und (2.2) bzw. (2.3) in $\mathcal{O}(N \log N)$. Die Non-equidistant Fast Fourier Transform (NFFT) ist in [3, Abschnitt 1.1] beschrieben. Die NFFT-basierte schnelle Summation wird für den eindimensionalen Fall in [3, Abschnitt 2.1, 2.2] sowie die für den mehrdimensionalen Fall in [4] erläutert. Die Berechnung von Coulomb-Potential und -Kräften mittels schneller Summation ist in [5] beschrieben.

In [6] befindet sich die NFFT-Bibliothek Version 3, welche u.a. die NFFT-Routinen und eine Implementierung der schnellen Summation [4] enthält.

Im Folgenden werden die bezüglich [4] und [5] vorgenommenen Modifikationen beschrieben. Dabei wird explizit vom dreidimensionalen Fall ausgegangen und die Übereinstimmung von Quell- und Zielknoten angenommen, d.h. $y_j = x_j$ ($j = 1, \dots, N$).

3.2.1 Erweiterung auf Ellipsoide

Bei der derzeitigen Implementierung der NFFT-basierten schnellen Summation in [6] sind die möglichen Orte der Partikel beschränkt auf die Menge D ,

$$D := \{x \in \mathbb{R}^3; \|x\| < 1/4 - \varepsilon_B/2\},$$

desweiteren ist die NFFT-Transformationslänge $n = (n_1, n_2, n_3)^T$ in allen 3 Raumkomponenten gleich groß, d.h. $n_1 = n_2 = n_3$. Es ist zwar möglich, beliebig im Raum verteilte Partikel in D zu verschieben und zu skalieren, aber dies kann bei bestimmten Verteilungen die numerische Genauigkeit und die Laufzeit negativ beeinflussen.

Es werden nun folgende Änderungen vorgenommen. Die NFFT-Transformationslänge kann in jeder Komponente beliebig sein. Seien

$$\begin{aligned} n_{\min} &:= \min\{n_1, n_2, n_3\}, \\ d_i &:= n_i/n_{\min} \quad (i = 1, 2, 3), \\ x &:= \left(x^{(1)}, x^{(2)}, x^{(3)}\right)^T \in \mathbb{R}^3, \\ \nu(x) &:= \left(\left(x^{(1)}/d_1\right)^2 + \left(x^{(2)}/d_2\right)^2 + \left(x^{(3)}/d_3\right)^2\right)^{1/2}, \end{aligned}$$

dann wird die Menge D wie folgt definiert:

$$D := \{x \in \mathbb{R}^3; \nu(x) < 1/4 - \varepsilon_B/2\}.$$

Wichtig ist hierbei, dass der zulässige Bereich der Orte der Partikel vom Verhältnis der Komponenten der NFFT-Transformationslänge abhängt.

Seien

$$\begin{aligned} r_{\text{outer}}(x) &:= 0.5/\nu(x) \cdot \|x\|, \\ r_{\text{inner}}(x) &:= (0.5 - \varepsilon_B)/\nu(x) \cdot \|x\|, \\ n_{\max} &:= \max\{n_1, n_2, n_3\}. \end{aligned}$$

Dann wird der regularisierte Kern K_R definiert durch

$$K_R = \begin{cases} K_I(x) & \text{falls } \|x\| \in [0, \varepsilon_I], \\ K_B(x) & \text{falls } \nu(x) \in [1/2 - \varepsilon_B, 1/2], \\ 1/\|x\| & \text{falls } \|x\| > \varepsilon_I \text{ and } \nu(x) < 1/2 - \varepsilon_B, \\ 1/(n_{\max}/n_{\min}/2) & \text{sonst.} \end{cases} \quad (3.1)$$

Dabei bleibt die Nahfeldregularisierung durch das Polynom K_I gegenüber [4] unverändert, K_B wird durch 2-Punkt-Taylor-Interpolation ([4, S. 55]) konstruiert,

$$K_B(x) := \frac{1}{2^p} \sum_{j=0}^{p-1} \gamma_j (1 - z^2)^j \left((1 - z)^{p-j} + (1 + z)^{p-j} \right),$$

mit

$$\begin{aligned} \gamma_j &:= \sum_{\ell=0}^j \binom{p-1+\ell}{\ell} \frac{\varepsilon_B^{j-\ell}}{2^\ell (j-\ell)!} a_{j-\ell}(\|x\|), \\ z &:= \frac{\|x\| - 0.5(r_{\text{inner}}(x) + r_{\text{outer}}(x))}{\varepsilon_I}, \\ a_k(r) &:= \begin{cases} \frac{d^k(1/r)}{dr^k} & \text{falls } k > 0, \\ 1/r & \text{sonst.} \end{cases} \end{aligned}$$

Schließlich sind die Fourierkoeffizienten b_ℓ zur Approximation des Kerns K_R gegeben durch

$$b_\ell := \frac{1}{n_1 n_2 n_3} \sum_{(j_1, j_2, j_3) \in I_n^3} K_R \left(\begin{pmatrix} \ell_1 d_1/n_1 \\ \ell_2 d_2/n_2 \\ \ell_3 d_3/n_3 \end{pmatrix} \right) e^{2\pi i \left(\frac{j_1 \ell_1}{n_1 d_1} \times \frac{j_2 \ell_2}{n_2 d_2} \times \frac{j_3 \ell_3}{n_3 d_3} \right)}, \quad \ell \in I_n^3,$$

wobei

$$I_n^3 := \{-n_1/2, \dots, n_1/2 - 1\} \times \{-n_2/2, \dots, n_2/2 - 1\} \times \{-n_3/2, \dots, n_3/2 - 1\}.$$

3.2.2 Explizite Berechnung des Nahfeld-Regularisierungspolynoms

Bei der schnellen Summation, welche mit der NFFT-Bibliothek ausgeliefert wird, liegt das Nahfeld-Regularisierungspolynom $K_I(x)$ nicht explizit vor, sondern für jede benötigte Stelle x wird eine erneute 2-Punkt-Taylor-Interpolation durchgeführt, basierend auf dem Funktionswert und Ableitungen des radial-symmetrischen Kerns K an der Stelle $(\varepsilon_I, 0, 0)^T$.

Im Folgenden wird stattdessen $K_I(x)$ für den Kern $1/\|x\|$ für $p = 3, 4, 5$ und 6 mittels 2-Punkt-Taylor-Interpolation explizit berechnet und im Programm hinterlegt. Beispielsweise gilt für $p = 5$:

$$K_I(x) = \frac{315}{128 \varepsilon_I} + \left(\frac{-105}{32 \varepsilon_I^3} + \left(\frac{189}{64 \varepsilon_I^5} + \left(\frac{-45}{32 \varepsilon_I^7} + \frac{35}{128 \varepsilon_I^9} \|x\|^2 \right) \|x\|^2 \right) \|x\|^2 \right) \|x\|^2.$$

3.2.3 Berechnung der Kräfte

Anstatt die elektrischen Felder bzw. Kräfte explizit wie in [5] zu berechnen, werden diese wie folgt ermittelt. Sei

$$f(y_j) := \phi(y_j) = \sum_{\substack{k=1 \\ k \neq j}}^N \frac{q_k}{\|y_j - y_k\|} \approx f_{\text{NE}}(y_j) + f_{\text{RF}}(y_j), \quad j = 1, \dots, N,$$

wobei $f_{\text{NE}}(y_j)$ der Nahfeldkorrekturterm ist,

$$f_{\text{NE}}(y_j) = \sum_{\substack{i=1 \\ i \neq j \\ \|y_j - y_i\| \leq \varepsilon_I}}^N q_i \left(\frac{1}{\|y_j - y_i\|} - K_I(y_j - y_i) \right),$$

und $f_{\text{RF}}(y_j)$ die Fourier-Approximation von K_R

$$f_{\text{RF}}(y_j) = \sum_{\ell \in \mathbb{I}_n^3} b_\ell \left(\sum_{k=1}^N \alpha_k e^{-2\pi i \ell y_k} \right) e^{2\pi i \ell y_j}$$

mit

$$\mathbb{I}_n^3 := \{-n_1/2, \dots, n_1/2 - 1\} \times \{-n_2/2, \dots, n_2/2 - 1\} \times \{-n_3/2, \dots, n_3/2 - 1\}.$$

Wird nun wie in (2.2) der Gradient von $f(y_j)$ berechnet, so gilt

$$F(y_j) = -q_j \nabla f(y_j) \approx -q_j (\nabla f_{\text{NE}}(y_j) + \nabla f_{\text{RF}}(y_j)).$$

und hieraus folgt

$$\begin{aligned} & \frac{\partial}{\partial y_j^{(r)}} f_{\text{RF}}(y_j) \\ &= \frac{\partial}{\partial y_j^{(r)}} \sum_{\ell \in \mathbb{I}_n^3} b_\ell \left(\sum_{k=1}^N \alpha_k e^{-2\pi i \left(\frac{\ell_1 y_k^{(1)}}{n_1 d_1} \times \frac{\ell_2 y_k^{(2)}}{n_2 d_2} \times \frac{\ell_3 y_k^{(3)}}{n_3 d_3} \right)} \right) e^{2\pi i \left(\frac{\ell_1 y_j^{(1)}}{n_1 d_1} \times \frac{\ell_2 y_j^{(2)}}{n_2 d_2} \times \frac{\ell_3 y_j^{(3)}}{n_3 d_3} \right)} \\ &= \sum_{\ell \in \mathbb{I}_n^3} b_\ell \left(\sum_{k=1}^N \alpha_k e^{-2\pi i \left(\frac{\ell_1 y_k^{(1)}}{n_1 d_1} \times \frac{\ell_2 y_k^{(2)}}{n_2 d_2} \times \frac{\ell_3 y_k^{(3)}}{n_3 d_3} \right)} \right) \frac{\partial}{\partial y_j^{(r)}} e^{2\pi i \left(\frac{\ell_1 y_j^{(1)}}{n_1 d_1} \times \frac{\ell_2 y_j^{(2)}}{n_2 d_2} \times \frac{\ell_3 y_j^{(3)}}{n_3 d_3} \right)}, \\ &= \sum_{\ell \in \mathbb{I}_n^3} b_\ell 2\pi i \frac{\ell_r}{n_r d_r} \left(\sum_{k=1}^N \alpha_k e^{-2\pi i \left(\frac{\ell_1 y_k^{(1)}}{n_1 d_1} \times \frac{\ell_2 y_k^{(2)}}{n_2 d_2} \times \frac{\ell_3 y_k^{(3)}}{n_3 d_3} \right)} \right) e^{2\pi i \left(\frac{\ell_1 y_j^{(1)}}{n_1 d_1} \times \frac{\ell_2 y_j^{(2)}}{n_2 d_2} \times \frac{\ell_3 y_j^{(3)}}{n_3 d_3} \right)}, \end{aligned}$$

wobei $y_j = \left(y_j^{(1)}, y_j^{(2)}, y_j^{(3)}\right)^T$ und $\frac{\partial}{\partial y_j^{(r)}}$ die partielle Ableitung nach der r -ten Komponente ist ($r = 1, 2, 3$).

Desweiteren gilt

$$\nabla f_{\text{NE}}(y_j) = \sum_{\substack{i=1 \\ i \neq j \\ \|y_j - y_i\| \leq \varepsilon_I}}^N q_i \left(-\frac{y_j}{\|y_j - y_i\|^3} - \nabla_{y_j} K_I(y_j - y_i) \right).$$

Da K_I durch die Berechnung nach Punkt 3.2.2 explizit vorliegt, kann nun $\nabla_{y_j} K_I(y_j - y_i)$ analytisch berechnet werden. Für $p = 5$ ergibt sich beispielsweise

$$\begin{aligned} & \nabla_{y_j} K_I(y_j - y_i) \\ = & -y_j \left[\frac{420}{64 \varepsilon_I^3} + \left(\frac{-378}{32 \varepsilon_I^5} + \left(\frac{1080}{128 \varepsilon_I^7} + \frac{-35}{16 \varepsilon_I^9} \|y_j - y_i\|^2 \right) \|y_j - y_i\|^2 \right) \|y_j - y_i\|^2 \right]. \end{aligned}$$

4 Numerische Tests

4.1 Erzeugung von Simulationsdaten

Betrachtet wurden 8 homogene pseudo-zufällige, 3 gitterbasierte sowie eine inhomogene Verteilung mit je 500, 5 000 sowie 50 000 Partikeln; bei 3 von den 8 homogenen pseudozufälligen Verteilungen zusätzlich 500 000 Partikel. Eine Beschreibung für jede dieser 12 Verteilungen befindet sich in Punkt 4.4.

Bei den 8 homogenen pseudo-zufälligen Verteilungen wurden Hammersley- bzw. Halton-Verteilungen ([7]) verwendet, da hierdurch immer ein Mindestabstand zwischen den Partikeln gewährleistet ist. Im Folgenden werden diese beiden Verteilungsarten erläutert.

Sei p eine Primzahl. Dann lässt sich jede nichtnegative ganze Zahl k eindeutig bezüglich p wie folgt darstellen:

$$k = a_0 + a_1p + a_2p^2 + \dots + a_rp^r,$$

mit $a_i \in [0, p - 1]$. Die Funktion $\Phi_p(k)$ ist definiert durch

$$\Phi_p(k) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \dots + \frac{a_r}{p^{r+1}}.$$

Hammersley-Verteilung. Seien d die Raumdimension der zu erzeugenden Daten sowie p_1, p_2, \dots, p_{d-1} Primzahlen mit $p_1 < p_2 < \dots < p_{d-1}$. Desweiteren sei N die Anzahl der zu erzeugenden Punkte. Dann ist der k -te d -dimensionale Punkt der Hammersley-Verteilung definiert durch

$$\left(\frac{k}{N}, \Phi_{p_1}(k), \Phi_{p_2}(k), \dots, \Phi_{p_{d-1}}(k) \right), \quad k = 0, \dots, N - 1.$$

Da die 1. Komponente des k -ten Partikels k/N ist, muss die Anzahl zu erzeugender Partikel beim Generieren des ersten Partikels bereits feststehen.

Halton-Verteilung. Sei nun zusätzlich p_d eine Primzahl mit $p_{d-1} < p_d$. Dann ist der k -te d -dimensionale Punkt der Halton-Verteilung gegeben durch

$$(\Phi_{p_1}(k), \Phi_{p_2}(k), \dots, \Phi_{p_d}(k)), \quad k = 0, \dots, N - 1.$$

Da jede Komponente des k -ten Partikels nicht von N abhängt, kann jederzeit ein zusätzliches Partikel erzeugt werden.

Für beide Verteilungsarten ist eine Implementierung von John Burkardt namens *HAMMERSLEY* ([8]) verfügbar. Diese enthält Routinen zum Erzeugen von Hammersley- und Halton-Verteilungen im Quadrat $[0, 1]^2$ sowie im Würfel $[0, 1]^3$. Desweiteren sind Hilfsroutinen enthalten, welche Punkte aus dem Quadrat $[0, 1]^2$ auf die Oberfläche der dreidimensionalen Einheitskugel sowie aus dem Würfel $[0, 1]^3$ auf Koordinaten innerhalb der dreidimensionalen Einheitskugel abbilden.

4.2 Testparameter, Fehlerberechnung und Zeiten

Die Testläufe der FMM und NFFT-basierten schnellen Summation mit 500 Partikeln wurden für die Zeitmessung 5-mal durchgeführt, mit 5 000, 50 000 sowie 500 000 Partikel 3-mal.

Bei der FMM wurde der maximal erlaubte relative Fehler für die potentielle Energie bzgl. der direkten Summation `pot_en_rel_error(direct, FMM)` auf 0.001 gesetzt,

$$\text{pot_en_rel_error}(\text{direct}, t) := U_{(t)} / U_{(\text{direct})},$$

$$U_{(t)} := \frac{1}{2} \sum_{i=1}^N q_i \phi_{(t)}(x_i),$$

wobei $\phi_{(\text{direct})}(x_i)$ nach (2.1) berechnet wird und $t \in \{\text{direct}, \text{FMM}, \text{NFFT}\}$. Da die NFFT-basierte schnelle Summation keinen automatischen Fehlerschätzer besitzt, wurden hier die Parameter so gewählt, dass diese Vorgabe für den maximalen relativen Fehler von U erreicht wird.

Desweiteren wurde der Fehler ε_2 des Coulomb-Potentials berechnet,

$$\varepsilon_2(\text{direct}, t) = \left(\sum_{i=1}^N (\phi_{(t)}(x_i) - \phi_{(\text{direct})}(x_i))^2 \right)^{1/2} \left(\sum_{i=1}^N (\phi_{(\text{direct})}(x_i))^2 \right)^{-1/2}$$

wobei $t \in \{\text{FMM}, \text{NFFT}\}$. Weiterhin wurde der mittlere relative Kraft-Fehler `ferr` bestimmt,

$$\text{ferr}(\text{direct}, t) = \frac{1}{3} \sum_{\alpha} \varepsilon_{F_{(t)}^{(\alpha)}},$$

$$\varepsilon_{F_{(t)}^{(\alpha)}} = \sum_{i=1}^N \|F_{(t)}^{(\alpha)}(x_i) - F_{(\text{direct})}^{(\alpha)}(x_i)\| \left(\sum_{i=1}^N \|F_{(\text{direct})}^{(\alpha)}(x_i)\| \right)^{-1}$$

wobei $F_{(\text{direct})}(x_i)$ nach (2) berechnet wird, und $F_{(\text{direct})}^{(\alpha)}(x_i)$ die α -te Komponente von $F_{(\text{direct})}(x_i)$ ist, $\alpha = 1, 2, 3$.

t_{direct} , t_{FMM} bzw. t_{NFFT} ist das arithmetische Mittel der jeweiligen Laufzeiten in Sekunden.

Bei der NFFT werden Kaiser-Bessel-Funktionen verwendet, der Oversamplingfaktor wird auf 2 gesetzt. Desweiteren werden bei Aufruf der NFFT (`nfft_init`) die Flags `PRE_PHI_HUT`, `PRE_LIN_PSI`, `MALLOC_X`, `MALLOC_F_HAT`, `MALLOC_F`, `FFTW_INIT`, `FFT_OUT_OF_PLACE`, `FFTW_MEASURE` sowie `FFTW_DESTROY_INPUT` gesetzt.

Die übrigen Parameter werden in Tabelle 4.1 beschrieben.

N	Anzahl erzeugter Partikel
#direct	Anzahl Testläufe um mittlere Laufzeit der direkten Summation zu bestimmen
#fmm	Anzahl Testläufe um mittlere Laufzeit der FMM zu bestimmen
#nfft	Anzahl Testläufe um mittlere Laufzeit der NFFT-basierten schnellen Summation zu bestimmen
n	NFFT-Transformationslänge
m	NFFT, Cut-off-Parameter der Fensterfunktion ([3, S. 15] der NFFT)
p	gibt bei der NFFT-basierten schnellen Summation an, wie oft der regularisierte Kern K_R ist $(p - 1)$ -mal stetig differenzierbar ist
ε_I	NFFT-basierte schnelle Summation, Nahfeldgröße, siehe Gleichung (3.1)
ε_B	NFFT-basierte schnelle Summation, siehe Gleichung (3.1)

Tabelle 4.1: Weitere Testparameter.

4.3 Testumgebung

Alle Testläufe wurden auf dem Cluster JUMP (IBM Regatta p690+, [9]) des Forschungszentrum Jülich durchgeführt. Dieser Cluster besteht aus 41 Rechenknoten, wobei jeder dieser Knoten 32 Prozessoren des Typs Power4+ 1.7 GHz und 128 GB RAM besitzt. Für die Zeitmessungen wurde ein kompletter Rechenknoten (32 Prozessoren) reserviert, wobei nur 1 Prozessor benutzt wurde.

Die FCS library (Stand 16.08.2007) wurde mit `xlf` und `xlc` mit Optimierungsstufe `-O5` übersetzt, FFTW und NFFT mit `xlc` mit der Optimierungsstufe `-O5`. Die Bibliotheken und Programme wurden dabei für den 64-Bit-Modus (Flag `-q64`) erzeugt.

4.4 Verteilungen und Ergebnisse

4.4.1 Verteilung `hammersley_ball`

Die Partikel werden Hammersley-verteilt im Würfel $[0, 1]^3$ erzeugt, $p_1 = 2$, $p_2 = 3$, und anschließend auf Koordinaten innerhalb der Kugel

$$(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 \leq (0.5)^2$$

abgebildet. Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

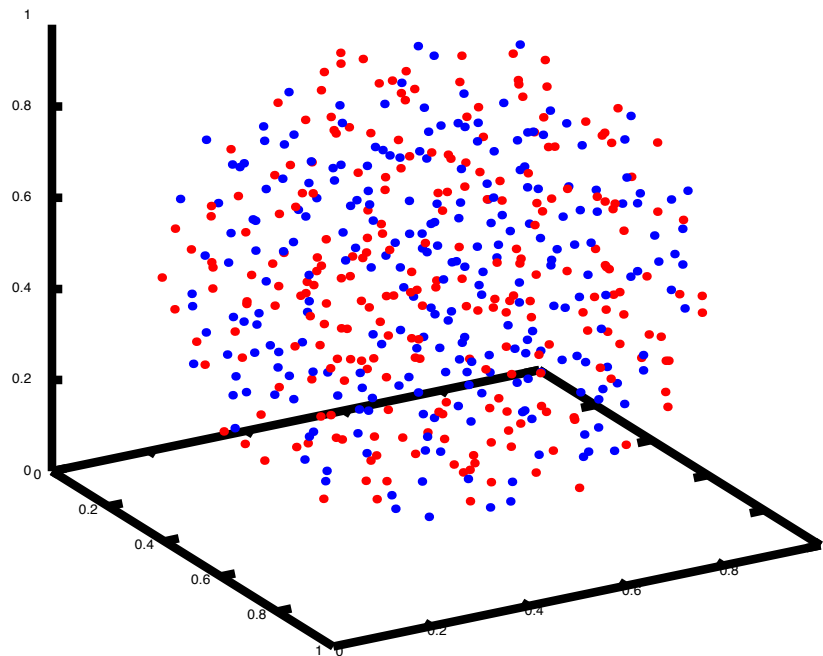


Abbildung 4.1: Verteilung `hammersley_ball` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	3.5/32
5 000	3	3	3	32,32,32	2	5	3.5/32
50 000	3	3	3	64,64,64	2	5	3.5/64
500 000	1	3	3	128,128,128	2	5	3.5/128

Tabelle 4.2: Algorithmen-Parameter für die Verteilung `hammersley_ball`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.01	0.11	9.677e-16	5.584e-05
5 000	0.86	0.24	0.34	1.339e-06	4.903e-05
50 000	87.06	3.47	4.20	1.113e-07	4.087e-04
500 000	8903.23	46.77	59.48	1.075e-09	3.726e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	1.751e-15	4.189e-04	2.328e-15	1.092e-03
5 000	5.471e-06	3.190e-04	1.454e-05	6.404e-04
50 000	2.579e-07	2.507e-04	7.650e-07	7.416e-04
500 000	7.366e-08	2.962e-04	2.195e-07	7.613e-04

Tabelle 4.3: Zeiten und Fehler für die Verteilung `hammersley_ball`.

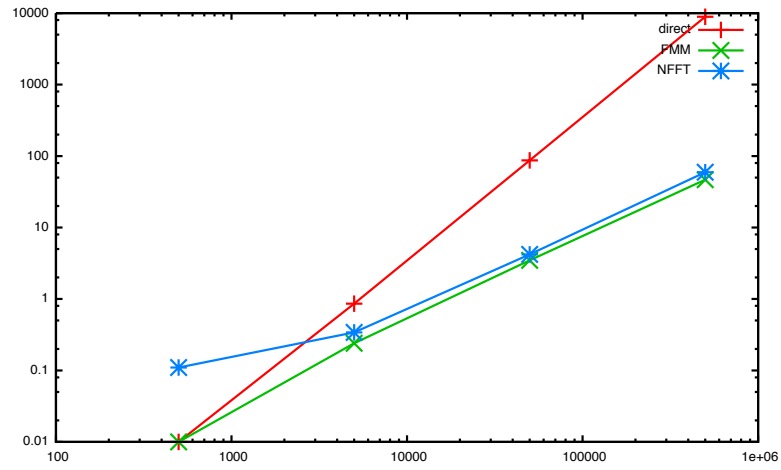


Abbildung 4.2: Anzahl Partikel vs. Zeit für die Verteilung `hammersley_ball`.

4.4.2 Verteilung `hammersley_ball_neg_charge`

Die Partikel werden Hammersley-verteilt im Würfel $[0, 1]^3$ erzeugt, $p_1 = 2$, $p_2 = 3$, und anschließend auf Koordinaten innerhalb der Kugel

$$(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 \leq (0.5)^2$$

abgebildet, für die Ladungen q_i gilt $q_i = -1$.

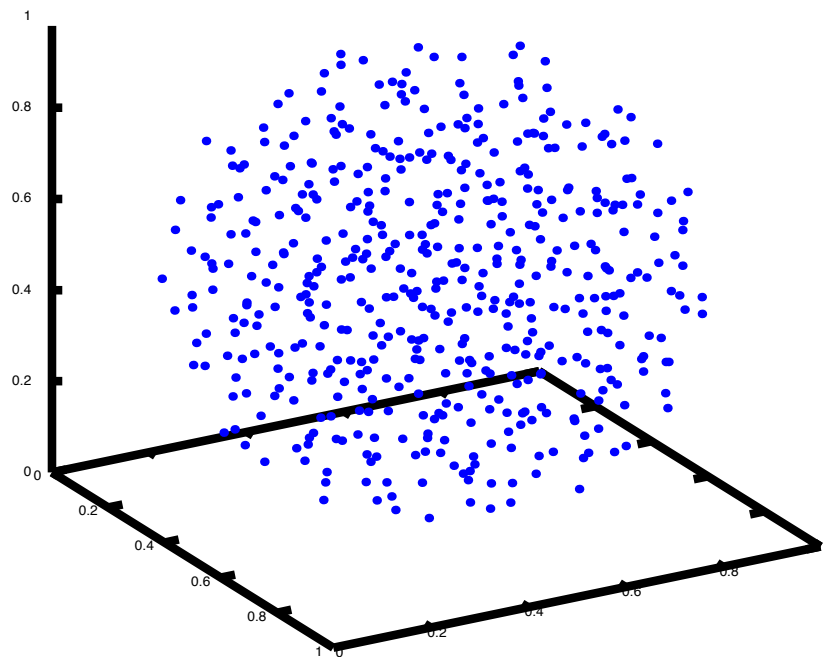


Abbildung 4.3: Verteilung `hammersley_ball_neg_charge` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	2/32
5 000	3	3	3	32,32,32	2	5	2/32
50 000	3	3	3	64,64,64	2	5	1.5/64
500 000	1	3	3	128,128,128	2	5	1/128

Tabelle 4.4: Algorithmen-Parameter für die Verteilung `hammersley_ball_neg_charge`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.01	0.11	3.678e-04	7.904e-05
5 000	0.86	0.09	0.19	2.509e-05	9.031e-05
50 000	87.16	0.90	2.36	7.463e-05	6.227e-05
500 000	8903.23*	9.11	35.59	1.131e-04	1.115e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	8.039e-04	1.436e-04	9.938e-03	3.257e-03
5 000	5.631e-04	1.260e-04	1.142e-02	5.137e-04
50 000	5.764e-04	1.081e-04	1.172e-02	1.391e-03
500 000	5.844e-04	1.414e-04	1.171e-02	3.273e-03

Tabelle 4.5: Zeiten und Fehler für die Verteilung `hammersley_ball_neg_charge`, *Zeit geschätzt.

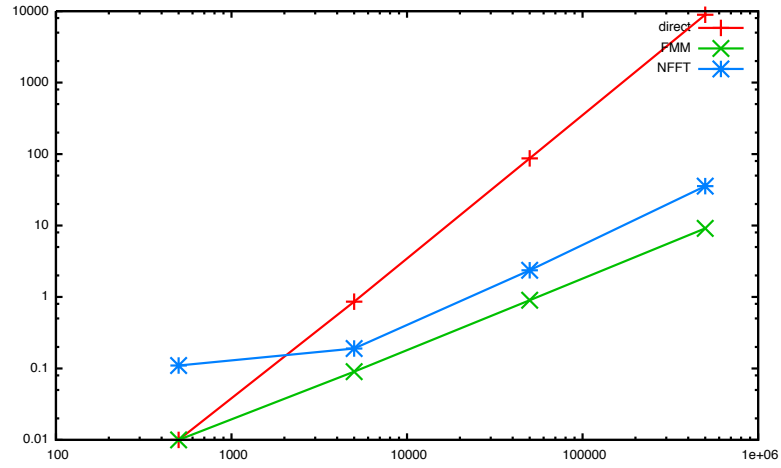


Abbildung 4.4: Anzahl Partikel vs. Zeit für die Verteilung `hammersley_ball_neg_charge`.

4.4.3 Verteilung halton_ellipsoid

Diese Verteilung besitzt die Parameter a, b, c . Die Partikel werden Halton-verteilt im Ellipsoid

$$(x/a)^2 + (y/b)^2 + (z/c)^2 \leq (0.5)^2$$

mit $p_1 = 2, p_2 = 3, p_3 = 5$ erzeugt und anschließend in den Würfel $[0, 1]^3$ verschoben und skaliert unter Beibehaltung der Seitenverhältnisse. Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

Im Folgenden werden die Parameter $a = 3, b = 2, c = 1$ gewählt.

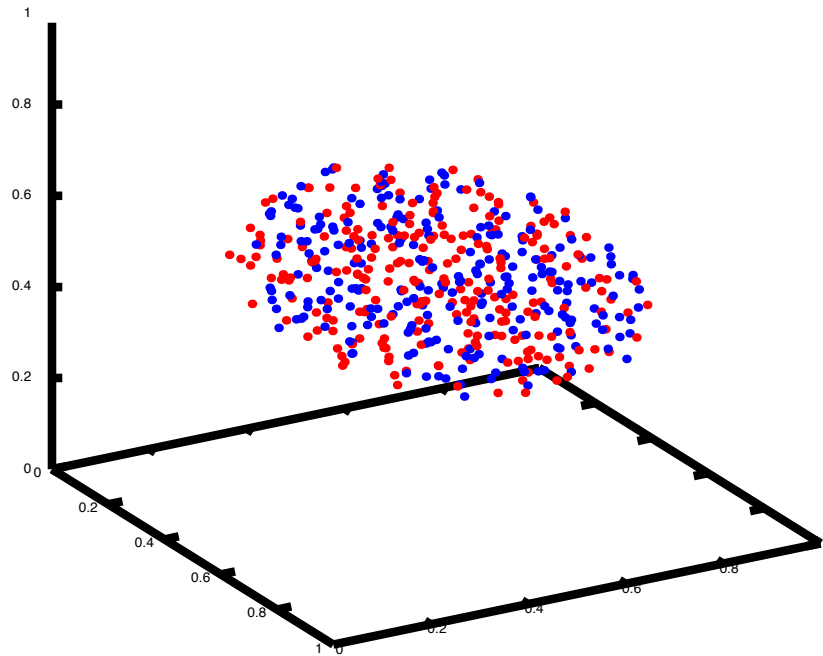


Abbildung 4.5: Verteilung halton_ellipsoid mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	3/32
5 000	3	3	3	32,32,32	2	5	3/32
50 000	3	3	3	64,64,64	2	5	3/64
500 000	1	3	3	128,128,128	2	5	3.5/128

Tabelle 4.6: Algorithmen-Parameter für die Verteilung halton_ellipsoid.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.02	0.11	1.208e-05	1.810e-04
5 000	0.86	0.21	0.46	8.998e-07	2.763e-04
50 000	87.09	3.58	7.01	7.657e-09	3.820e-06
500 000	8903.23*	64.21	135.45	1.099e-08	9.496e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	3.144e-05	5.584e-04	3.413e-05	9.762e-04
5 000	6.439e-06	7.210e-04	1.076e-05	6.324e-04
50 000	1.144e-07	4.323e-04	3.177e-07	9.023e-04
500 000	3.321e-09	2.358e-04	6.890e-09	4.547e-04

Tabelle 4.7: Zeiten und Fehler für die Verteilung halton_ellipsoid, *Zeit geschätzt.

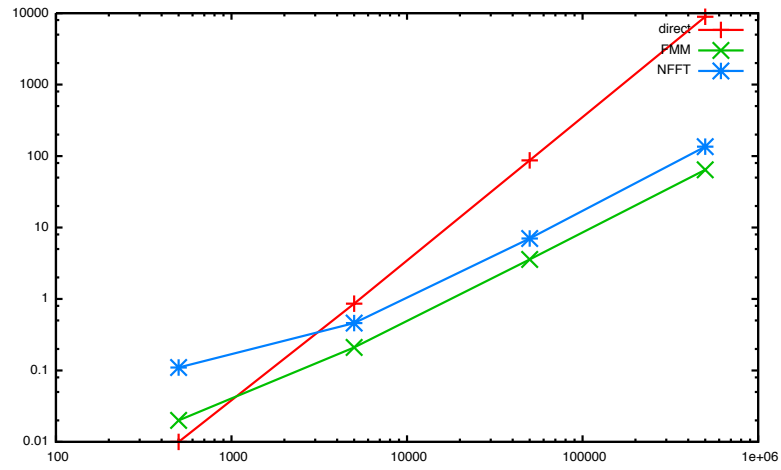


Abbildung 4.6: Anzahl Partikel vs. Zeit für die Verteilung halton_ellipsoid.

4.4.4 Verteilung halton_cylinder

Diese Verteilung besitzt den Parameter `ratio_radius_length`.

Die Partikel werden Halton-verteilt im Zylinder $0 \leq x \leq \text{length}$, $(y - 0.5)^2 + (z - 0.5)^2 \leq (\text{radius})^2$ mit $p_1 = 2$, $p_2 = 3$, $p_3 = 5$ erzeugt, wobei `ratio_radius_length` = `radius/length`. Anschließend werden die Partikel in den Würfel $[0, 1]^3$ verschoben und skaliert unter Beibehaltung der Seitenverhältnisse. Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

Im Folgenden wird der Parameter `ratio_radius_length` = 0.1 gewählt.

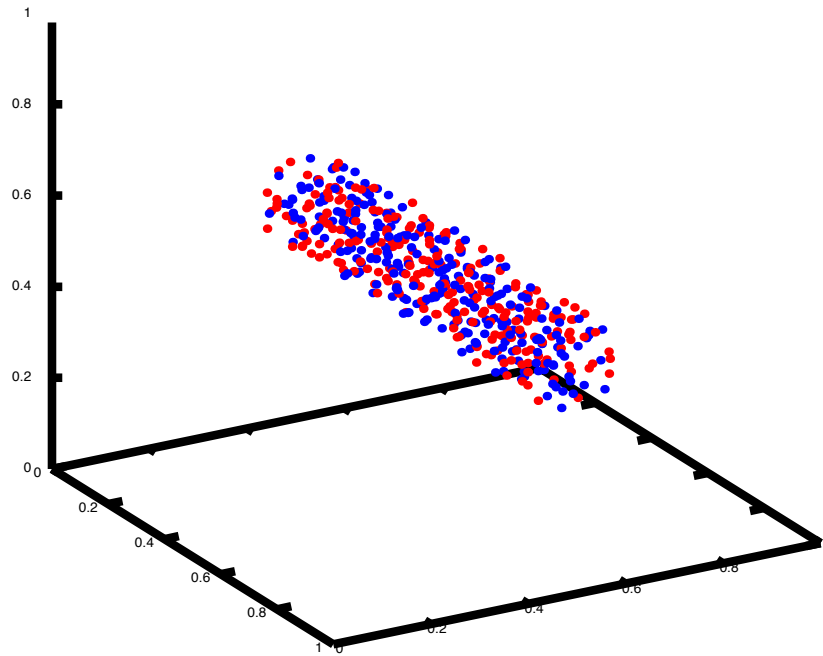


Abbildung 4.7: Verteilung `halton_cylinder` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
N	5	5	5	32,32,32	2	5	3/32
N	3	3	3	32,32,32	2	5	3/32
N	1	3	3	256,64,64	3	5	3.5/64

Tabelle 4.8: Algorithmen-Parameter für die Verteilung `halton_cylinder`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.04	0.12	1.401e-06	1.201e-04
5 000	0.86	0.25	0.60	2.938e-07	2.161e-04
50 000	87.19	3.80	9.94	6.025e-08	1.065e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	1.295e-05	4.374e-04	1.535e-05	3.973e-04
5 000	2.030e-06	3.914e-04	4.497e-06	2.491e-04
50 000	2.663e-08	3.011e-04	7.379e-08	1.302e-03

Tabelle 4.9: Zeiten und Fehler für die Verteilung `halton_cylinder`.

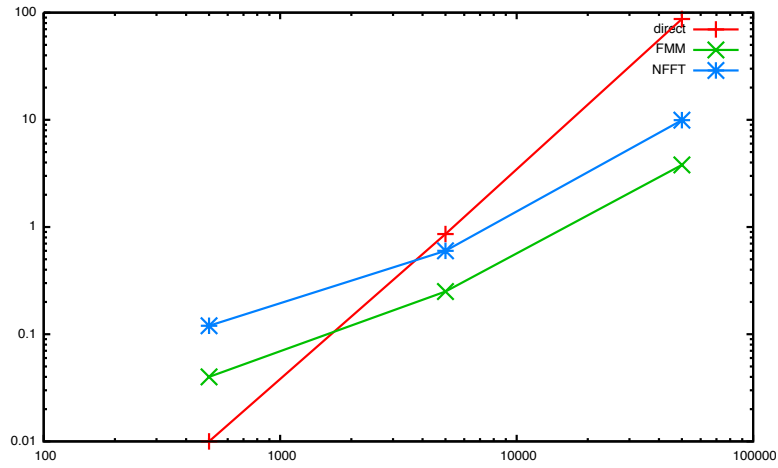


Abbildung 4.8: Anzahl Partikel vs. Zeit für die Verteilung `halton_cylinder`.

4.4.5 Verteilung `hammersley_circle`

Die Partikel werden Hammersley-verteilt im Quadrat $[0, 1]^2$ erzeugt, $p_1 = 2$, und anschließend auf Koordinaten innerhalb des Kreises

$$(x - 0.5)^2 + (y - 0.5)^2 \leq (0.5)^2$$

abgebildet, $z = 0.5$. Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

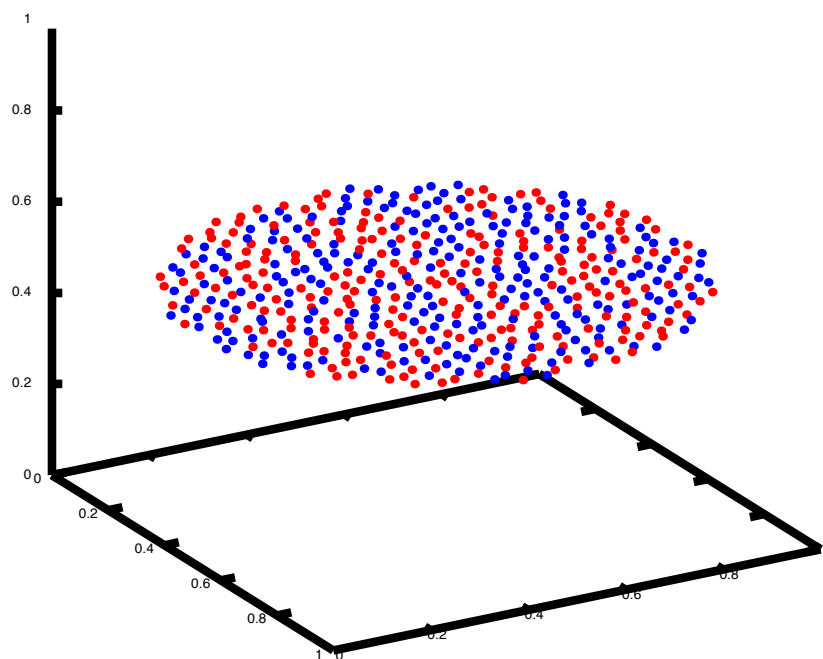


Abbildung 4.9: Verteilung `hammersley_circle` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
N	5	5	5	32,32,16	2	5	3/16
N	3	3	3	32,32,16	2	5	2.4/16
N	1	3	3	64,64,16	3	5	2.4/16

Tabelle 4.10: Algorithmen-Parameter für die Verteilung `hammersley_circle`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.02	0.06	8.092e-07	1.415e-04
5 000	0.86	0.16	0.29	4.596e-07	5.686e-05
50 000	87.04	2.60	6.43	8.304e-08	4.359e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	6.915e-06	5.776e-04	1.048e-05	4.271e-04
5 000	9.023e-07	1.159e-03	8.952e-07	4.522e-04
50 000	2.446e-08	8.683e-04	2.369e-08	2.521e-04

Tabelle 4.11: Zeiten und Fehler für die Verteilung `hammersley_circle`.

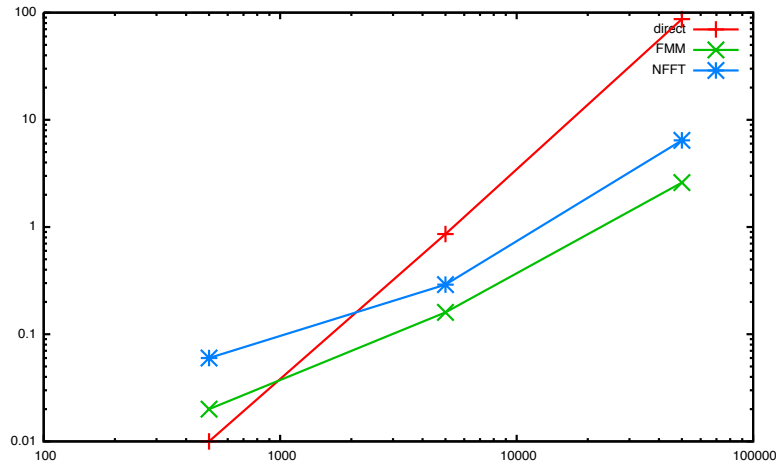


Abbildung 4.10: Anzahl Partikel vs. Zeit für die Verteilung `hammersley_circle`.

4.4.6 Verteilung `hammersley_cube`

Die Partikel sind Hammersley-verteilt im Würfel $[0, 1]^3$, $p_1 = 2$, $p_2 = 3$. Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

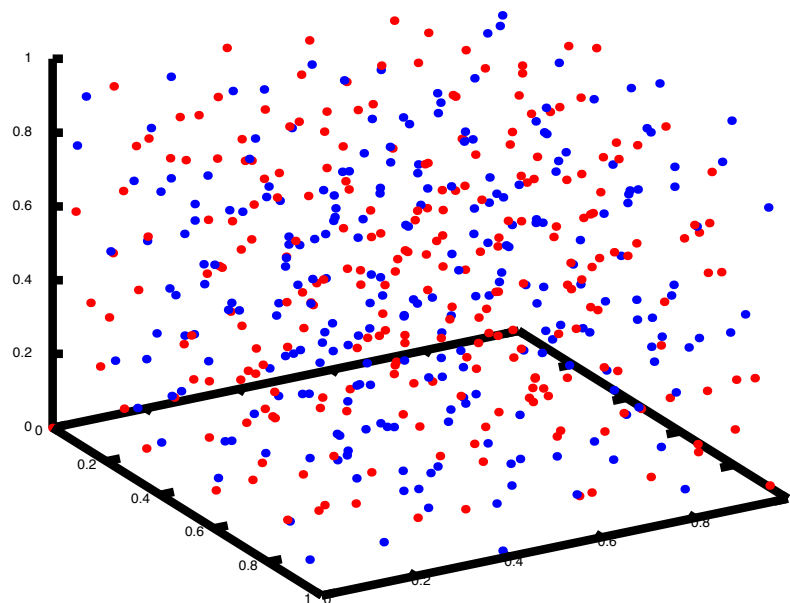


Abbildung 4.11: Verteilung `hammersley_cube` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	3/32
5 000	3	3	3	32,32,32	2	5	3/32
50 000	1	3	3	64,64,64	2	5	3/64

Tabelle 4.12: Algorithmen-Parameter für die Verteilung `hammersley_cube`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.01	0.11	1.544e-15	3.094e-04
5 000	0.86	0.25	0.38	1.668e-07	9.205e-05
50 000	87.09*	3.00	4.63	2.388e-08	3.006e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	1.603e-15	8.086e-04	1.689e-15	1.917e-03
5 000	2.571e-06	5.626e-04	7.270e-06	9.513e-04
50 000	3.222e-07	5.454e-04	8.060e-07	1.240e-03

Tabelle 4.13: Zeiten und Fehler für die Verteilung `hammersley_cube`, *Zeit geschätzt.

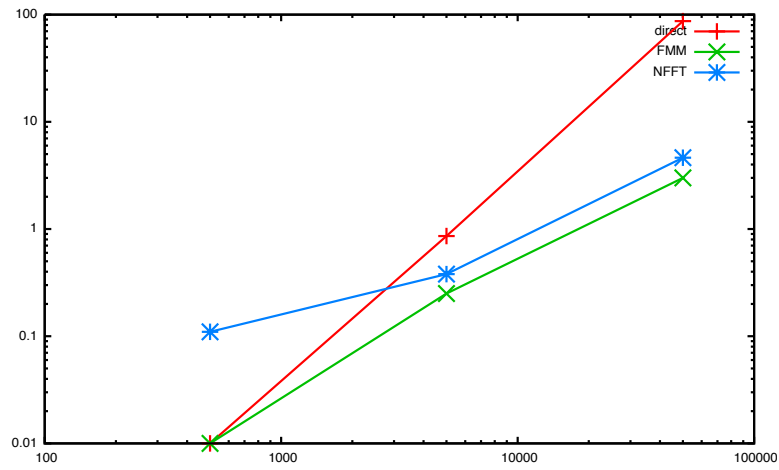


Abbildung 4.12: Anzahl Partikel vs. Zeit für die Verteilung `hammersley_cube`.

4.4.7 Verteilung `hammersley_sphere`

Die Partikel werden Hammersley-verteilt im Quadrat $[0, 1]^2$ erzeugt, $p_1 = 2$, und anschließend auf Koordinaten auf der Kugeloberfläche

$$(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 = (0.5)^2$$

abgebildet. Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

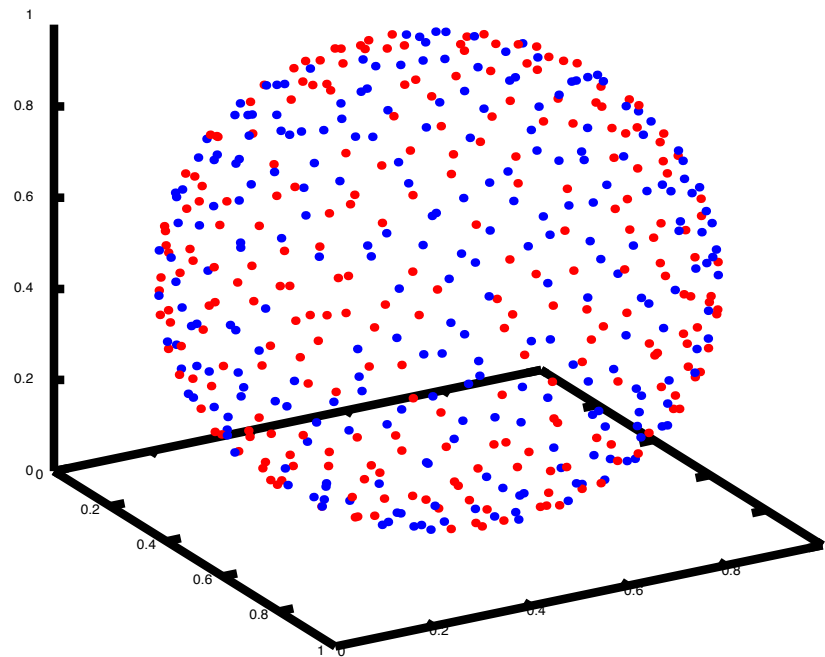


Abbildung 4.13: Verteilung `hammersley_sphere` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	3/32
5 000	3	3	3	32,32,32	2	5	3/32
50 000	1	3	3	64,64,64	2	5	3/64

Tabelle 4.14: Algorithmen-Parameter für die Verteilung `hammersley_sphere`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.01	0.11	0.000e+00	1.338e-04
5 000	0.86	0.17	0.23	9.167e-08	1.957e-04
50 000	87.13	2.58	3.41	2.460e-08	4.866e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	1.740e-15	6.655e-04	2.983e-15	1.697e-03
5 000	1.202e-06	4.101e-04	1.478e-06	3.644e-04
50 000	5.522e-08	2.422e-04	5.200e-08	1.545e-04

Tabelle 4.15: Zeiten und Fehler für die Verteilung `hammersley_sphere`.

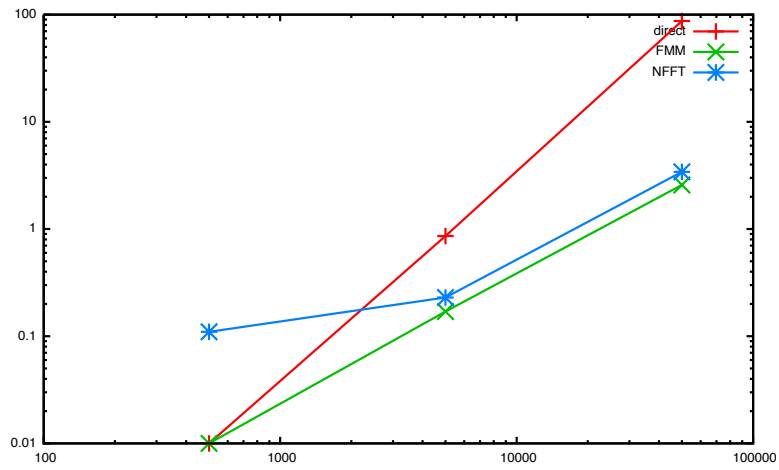


Abbildung 4.14: Anzahl Partikel vs. Zeit für die Verteilung `hammersley_sphere`.

4.4.8 Verteilung `hammersley_two_balls`

Diese Verteilung besitzt die beiden Parameter `n2_rel_to_n1` und `dist_rel_to_r1`.

Sei N die Anzahl zu erzeugender Partikel. Es werden nun $((1 - \text{n2_rel_to_n1}) \cdot N)$ Partikel Hammersley-verteilt im Würfel $[0, 1]^3$ erzeugt, $p_1 = 2$, $p_2 = 3$, und anschließend auf Koordinaten innerhalb der Kugel

$$(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 \leq (0.5)^2$$

abgebildet. Dann wird eine zweite Kugel erzeugt mit $(\text{n2_rel_to_n1} \cdot N)$ Partikeln und derselben Dichte wie die erste Kugel. Dabei wird die Distanz dist_rel_to_r1 zwischen beiden Kugelmittelpunkten in Vielfachen des Radius der ersten Kugel angegeben. Zuletzt werden die Partikel in den Würfel $[0, 1]^3$ verschoben und skaliert. Die Ladungen $q_i \in \{-1; 0; 1\}$ sind für jede der beiden Kugeln zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

Im Folgenden werden die Parameter `n2_rel_to_n1` = 1/64, `dist_rel_to_r1` = 20 gewählt.

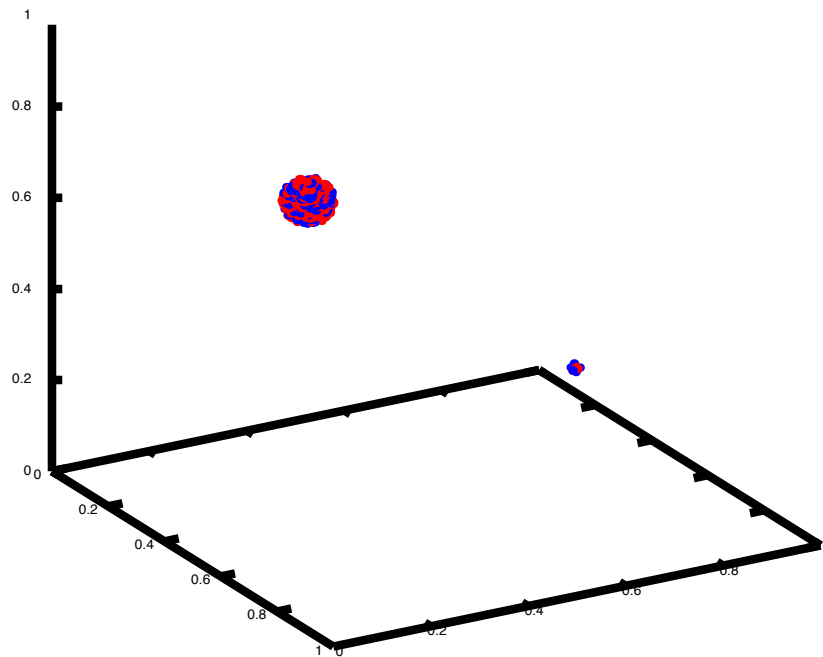


Abbildung 4.15: Verteilung `hammersley_two_balls` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	2/32
5 000	3	3	3	32,32,32	2	5	2/32
50 000	1	3	3	384,48,48	2	5	3.2/48

Tabelle 4.16: Algorithmen-Parameter für die Verteilung `hammersley_two_balls`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.01	0.12	6.111e-15	2.410e-04
5 000	0.86	0.26	1.15	1.188e-07	5.444e-04
50 000	87.21	22.44	17.01	5.916e-08	7.673e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	9.988e-15	4.439e-04	1.794e-14	2.286e-04
5 000	8.945e-07	6.288e-04	8.189e-08	2.322e-04
50 000	1.115e-07	6.624e-04	7.228e-09	1.033e-03

Tabelle 4.17: Zeiten und Fehler für die Verteilung `hammersley_two_balls`.

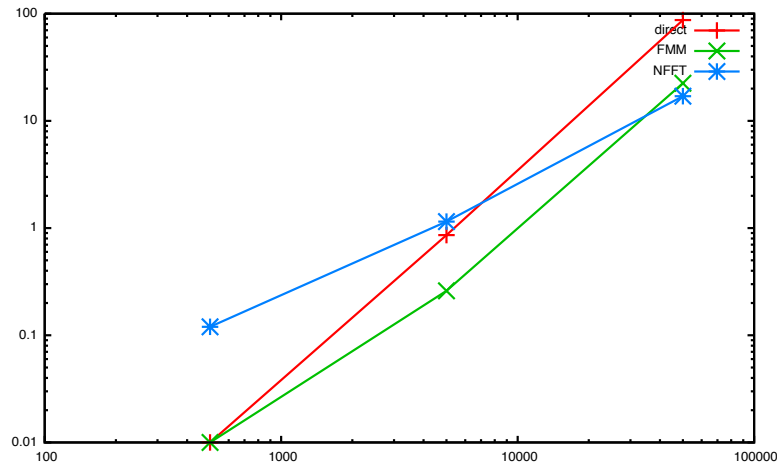


Abbildung 4.16: Anzahl Partikel vs. Zeit für die Verteilung `hammersley_two_balls`.

4.4.9 Verteilung `grid_body_centered_cube`

Sei N die Anzahl zu erzeugender Partikel mit $N = 2 \cdot N_c^3$, $N_c \in \mathbb{N} \setminus \{1\}$. Dann gilt für

$$j = (N_c \cdot u + v) \cdot N_c + w \quad (u = 0, \dots, N_c - 1; v = 0, \dots, N_c - 1; w = 0, \dots, N_c - 1)$$

für den Ort x_{2j+1} bzw. x_{2j+2} des $(2j+1)$ -ten bzw. $(2j+2)$ -ten Partikel:

$$\begin{aligned} x_{2j+1} &= (u, v, w)^T / (N_c - 1), \\ x_{2j+2} &= (u + 0.5, v + 0.5, w + 0.5)^T / (N_c - 1). \end{aligned}$$

Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

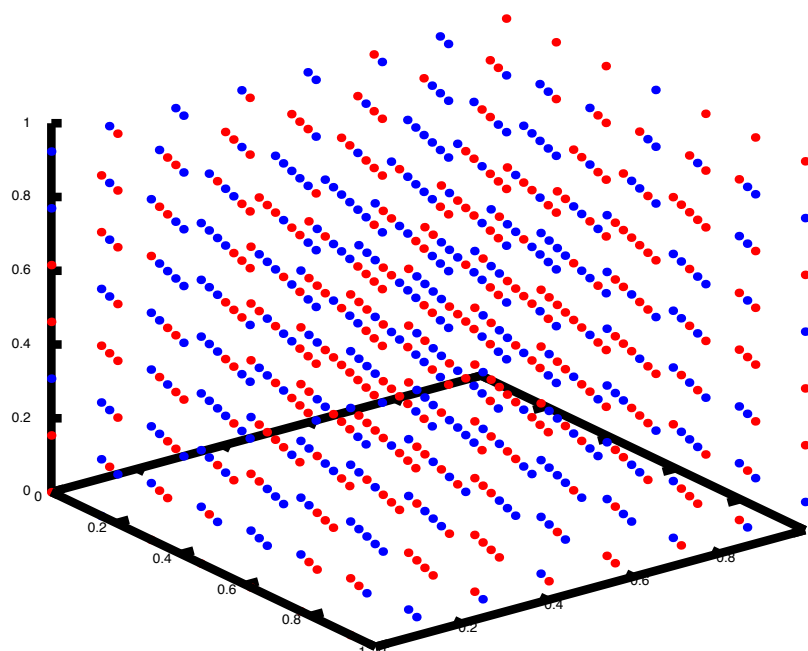


Abbildung 4.17: Verteilung `grid_body_centered_cube` mit 686 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
686	5	5	5	32,32,32	2	5	3.5/32
5488	3	3	3	32,32,32	2	5	6.4/32
54000	1	3	3	64,64,64	2	5	3.5/64

Tabelle 4.18: Algorithmen-Parameter für die Verteilung `grid_body_centered_cube`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
686	0.02	0.01	0.12	6.241e-16	1.693e-04
5 488	1.03	0.46	1.38	3.542e-05	6.803e-04
54 000	106.73*	3.09	6.71	2.798e-08	1.003e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
686	1.476e-15	3.595e-04	1.047e-15	9.086e-04
5 488	1.014e-07	6.216e-05	3.294e-07	1.928e-05
54 000	2.862e-07	2.701e-04	8.882e-07	6.880e-04

Tabelle 4.19: Zeiten und Fehler für die Verteilung `grid_body_centered_cube`, *Zeit geschätzt.

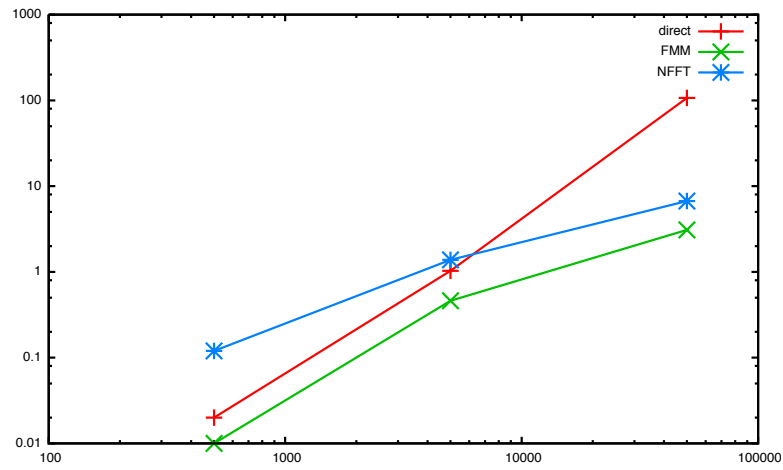


Abbildung 4.18: Anzahl Partikel vs. Zeit für die Verteilung `grid_body_centered_cube`.

4.4.10 Verteilung `grid_face_centered_cube`

Sei N die Anzahl zu erzeugender Partikel mit $N = 4 \cdot N_c^3$, $N_c \in \mathbb{N}$. Dann gilt für

$$j = (N_c \cdot u + v) \cdot N_c + w \quad (u = 0, \dots, N_c - 1; v = 0, \dots, N_c - 1; w = 0, \dots, N_c - 1)$$

für die Orte x_{4j+1} bis x_{4j+4} der $(4j+1)$ -ten bis $(4j+4)$ -ten Partikel:

$$\begin{aligned} x_{4j+1} &= (u, v, w)^T / (N_c - 0.5), \\ x_{4j+2} &= (u + 0.5, v + 0.5, w)^T / (N_c - 0.5), \\ x_{4j+3} &= (u + 0.5, v, w + 0.5)^T / (N_c - 0.5), \\ x_{4j+4} &= (u, v + 0.5, w + 0.5)^T / (N_c - 0.5). \end{aligned}$$

Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

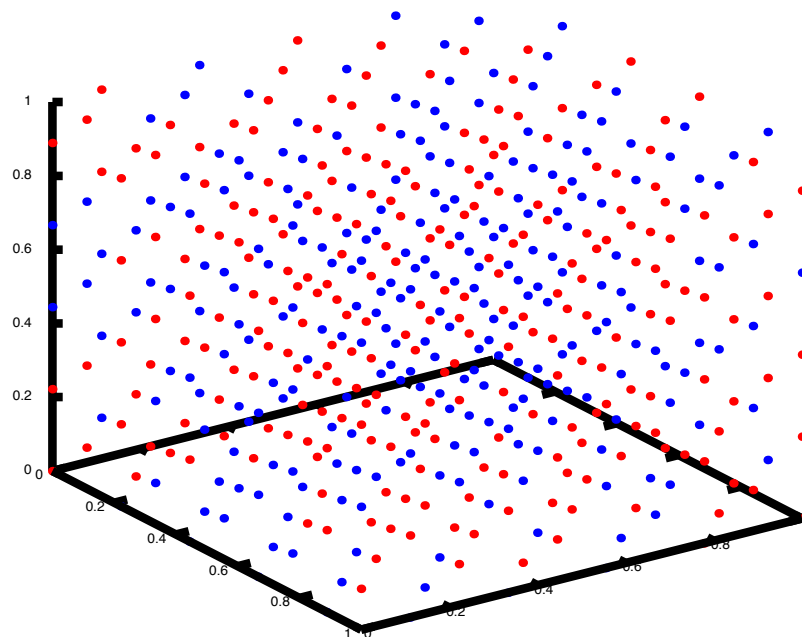


Abbildung 4.19: Verteilung `grid_face_centered_cube` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	3.5/32
5 324	3	3	3	32,32,32	2	5	3.5/32
55 296	1	3	3	64,64,64	3	5	3.5/64

Tabelle 4.20: Algorithmen-Parameter für die Verteilung `grid_face_centered_cube`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.01	0.11	3.498e-16	6.696e-06
5 324	0.98	0.33	0.48	8.840e-07	7.042e-04
55 296	106.73	4.61	7.42	3.561e-08	8.383e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	1.579e-15	4.809e-04	9.748e-16	1.002e-03
5 324	3.136e-07	2.552e-04	1.319e-06	4.664e-04
55 296	2.323e-08	1.801e-04	6.914e-08	6.695e-04

Tabelle 4.21: Zeiten und Fehler für die Verteilung `grid_face_centered_cube`.

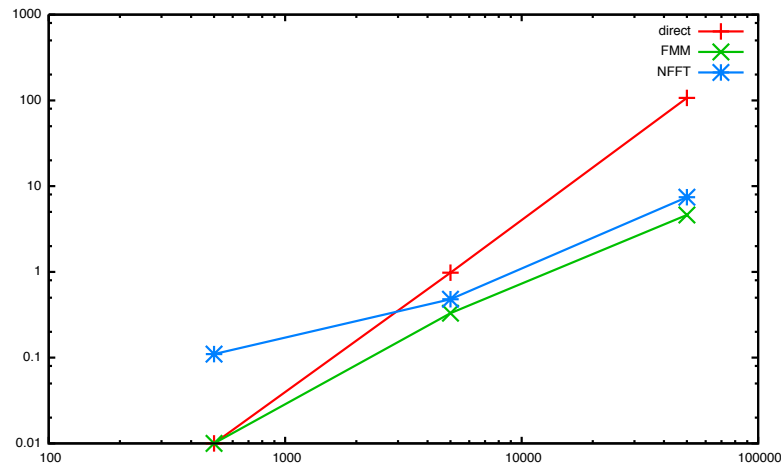


Abbildung 4.20: Anzahl Partikel vs. Zeit für die Verteilung `grid_face_centered_cube`.

4.4.11 Verteilung `grid_nacl_cube`

Sei N die Anzahl zu erzeugender Partikel mit $N = N_c^3$, $N_c \in \mathbb{N} \setminus \{1\}$. Dann gilt für

$$j = (N_c \cdot u + v) \cdot N_c + w \quad (u = 0, \dots, N_c - 1; v = 0, \dots, N_c - 1; w = 0, \dots, N_c - 1)$$

für den Ort x_j des j -ten Partikel:

$$x_j = (u, v, w)^T / (N_c - 1).$$

Für die Ladungen q_j des j -ten Partikel gilt $q_j = (-1)^{u+v+w+1}$.

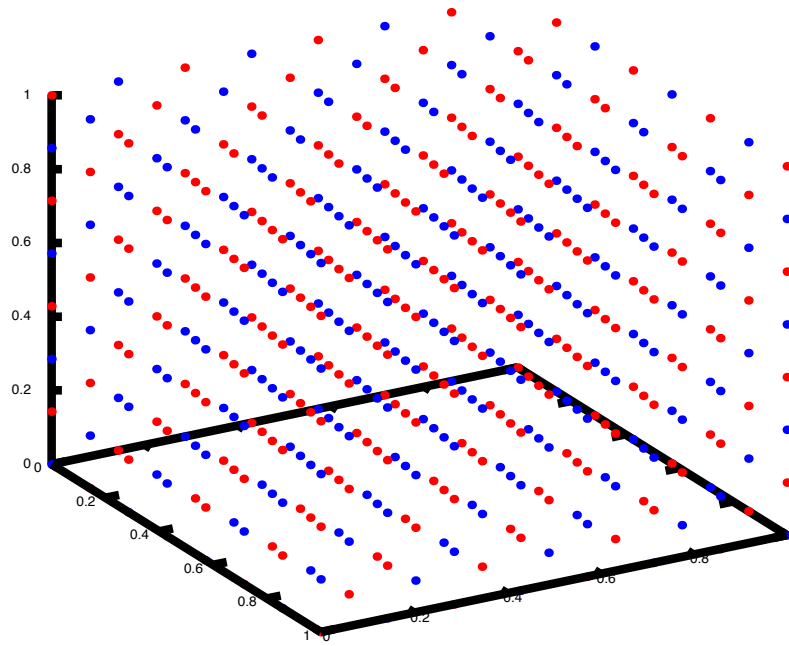


Abbildung 4.21: Verteilung `grid_nacl_cube` mit 512 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
512	5	5	5	32,32,32	2	5	4/32
5 832	3	3	3	32,32,32	2	5	2.5/32
50 653	1	3	3	64,64,64	2	5	2.5/64

Tabelle 4.22: Algorithmen-Parameter für die Verteilung `grid_nacl_cube`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
512	0.01	0.01	0.12	1.648e-15	2.316e-05
5 832	1.17	0.13	0.33	3.132e-07	4.491e-05
50 653	106.73*	2.41	3.21	3.094e-07	5.422e-05

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
512	1.454e-15	2.892e-05	8.933e-15	2.799e-03
5 832	1.406e-05	5.430e-05	5.102e-04	9.325e-04
50 653	1.054e-05	7.071e-05	9.372e-04	1.451e-03

Tabelle 4.23: Zeiten und Fehler für die Verteilung `grid_nacl_cube`, *Zeit geschätzt.

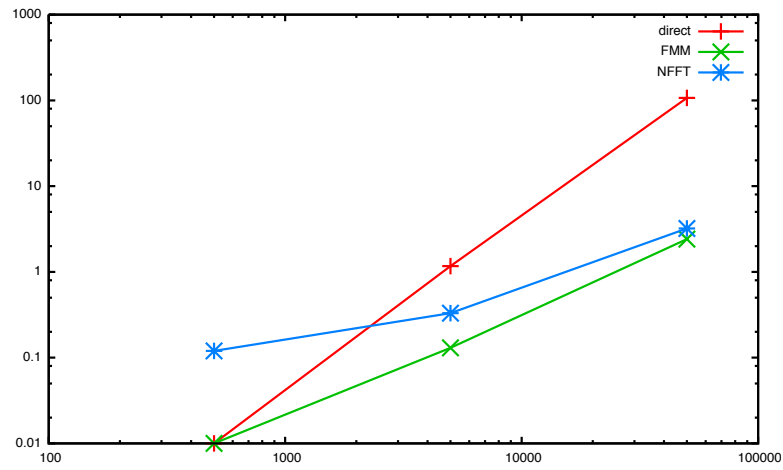


Abbildung 4.22: Anzahl Partikel vs. Zeit für die Verteilung `grid_nacl_cube`.

4.4.12 Verteilung `plummer_ball`

Diese inhomogene Verteilung wird bei N Partikeln wie folgt konstruiert. Für $j = 1, \dots, N$ wird ein Radius r_j bestimmt, sodass

$$\frac{j-1}{N} \leq \int_0^{r_j} 375 t^2 \cdot \left(1 + (t/0.2)^2\right)^{-5/2} dt < \frac{j}{N},$$

sowie zufällig gleichverteilt die Winkel $\theta_j \in [0, \pi]$ und $\varphi_j \in [0, 2\pi]$ erzeugt. Danach werden die Kugelkoordinaten $(r_j, \theta_j, \varphi_j)$ in kartesische Koordinaten transformiert sowie anschließend in den Würfel $[0, 1]^3$ verschoben und skaliert. Die Ladungen $q_i \in \{-1; 1\}$ sind zufällig gewählt mit

$$\sum_{i=1}^N q_i \in \{-1; 0; 1\}.$$

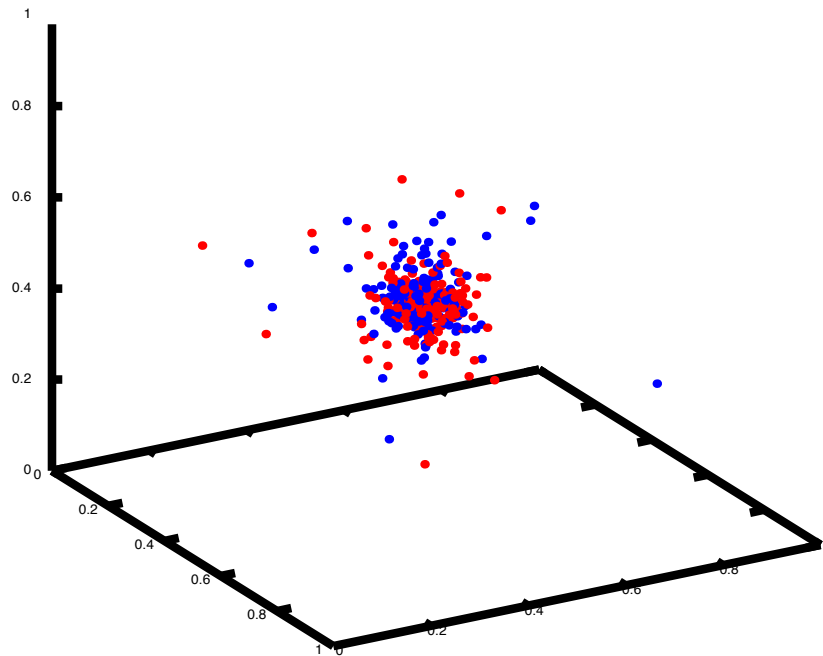


Abbildung 4.23: Verteilung `plummer_ball` mit 500 Partikeln.

N	#direct	#fmm	#nfft	n	m	p	$\varepsilon_I = \varepsilon_B$
500	5	5	5	32,32,32	2	5	2/32
5 000	3	3	3	32,32,32	2	5	1.5/32
50 000	1	3	3	64,64,64	2	5	2/64

Tabelle 4.24: Algorithmen-Parameter für die Verteilung `plummer_ball`.

N	t_{direct}	t_{FMM}	t_{NFFT}	pot_en_rel_error (direct, FMM)	pot_en_rel_error (direct, NFFT)
500	0.01	0.01	0.12	3.400e-14	1.593e-03
5 000	0.86	0.25	1.17	5.341e-14	9.707e-04
50 000	87.20	29.82	106.44	9.846e-12	3.339e-04

N	ε_2 (direct, FMM)	ε_2 (direct, NFFT)	ferr (direct, FMM)	ferr (direct, NFFT)
500	1.060e-14	9.255e-04	2.892e-14	3.314e-04
5 000	1.111e-13	6.478e-04	4.128e-13	8.349e-05
50 000	3.830e-13	2.053e-05	1.867e-12	5.815e-07

Tabelle 4.25: Zeiten und Fehler für die Verteilung `plummer_ball`.

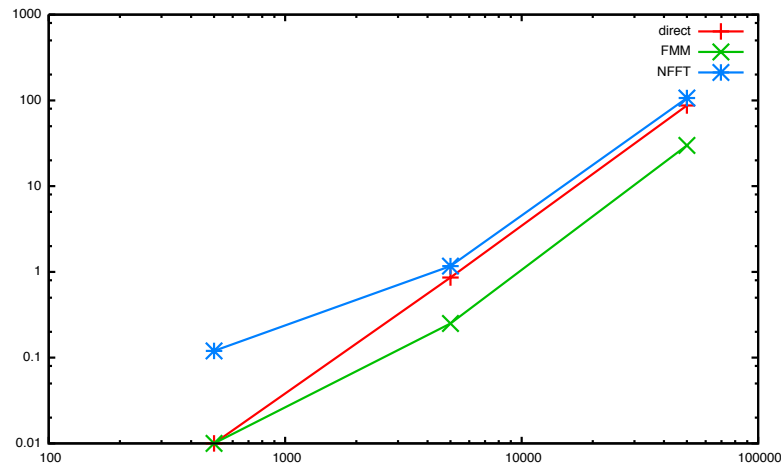


Abbildung 4.24: Anzahl Partikel vs. Zeit für die Verteilung `plummer_ball`.

5 Auswertung

Mit der FMM und der NFFT-basierten schnellen Summation sind zwei leistungsfähige Verfahren verfügbar, um die Berechnung von (2.1) und (2.2) in erheblich kürzerer Zeit durchzuführen als mit der direkten Summation.

Bei den Verteilungen `hammersley_ball`, `hammersley_sphere` und `grid_face_centered_cube` ist die FMM bezogen auf die NFFT-basierte schnelle Summation ab 5 000 Partikeln um den Faktor 1.2 bis 1.6 schneller bei ähnlichen oder geringeren Fehlern, bei den restlichen homogenen und gitterbasierten Verteilungen in der Regel um den Faktor 2 bis 4.

Bei der inhomogenen Verteilung `plummer_ball` ist die FMM bezogen auf die NFFT-basierte schnelle Summation 5 000 Partikeln um etwa den Faktor 4 schneller. Desweiteren ist die NFFT-basierte schnelle Summation bei dieser einen Verteilung langsamer als die direkte Summation.

6 Danksagung

Ich möchte mich bei Dr. Godehard Sutmann für die sehr gute Betreuung im Forschungszentrum bedanken sowie bei Prof. Dr. Daniel Potts für die sehr hilfreichen Ratschläge und Hinweise.

Desweiteren möchte ich mich Dr. Holger Dachsel für die umfassenden Erläuterungen zur FMM und bei Anke Visser für die schnelle Bereitstellung einer aktuellen Version der FCS library danken.

Literaturverzeichnis

- [1] Godehard Sutmann. Molecular Dynamics - Vision and Reality. *Computational Nanoscience: Do It Yourself!*, J. Grotendorst, S. Blügel, D. Marx (Eds.), John von Neumann Institute for Computing, Jülich, NIC Series, 31:159–194, 2006.
- [2] FCS library. <http://www.fz-juelich.de/jsc/fcs/>.
- [3] Daniel Potts. Schnelle Fourier-Transformationen für nichtäquidistante Daten und Anwendungen. Habilitationsschrift, Universität zu Lübeck, März 2003.
- [4] Markus Fenn. Fast Fourier Transform at Nonequispaced Nodes And Applications. PhD thesis, Mannheim University, December 2005.
- [5] D. Potts G. Pöplau and U. van Rienen. Calculation of 3D Space-Charge Fields of Bunches of Charged Particles by Fast Summation. In G. Ali A.M. Anile and G. Mascaly, editors, *Scientific Computing in Electrical Engineering*, pages 241–246. 2006. Springer-Verlag, Berlin.
- [6] NFFT library. <http://www-user.tu-chemnitz.de/~potts/nfft/>.
- [7] Tien-Tsin Wong, Wai-Shing Luk, and Pheng-Ann Heng. Sampling with Hammersley and Halton Points. *Journal of graphics tools*, 2(2):9–24, 1997.
- [8] HAMMERSLEY. The Hammersley Quasirandom Sequence.
http://people.scs.fsu.edu/~burkardt/cpp_src/hammersley/hammersley.html.
- [9] Jump - Juelich Multi Processor. <http://jumpdoc.fz-juelich.de>.