

Parallele nichtäquidistante schnelle Fourier-Transformation und schnelle Summation

Michael Pippig

Fakultät für Mathematik
Technische Universität Chemnitz

08. Mai 2009

mit freundlicher Unterstützung des BMBF

- 1 Diskrete Fourier-Transformationen
- 2 Parallelisierung der schnellen Fourier-Transformation nichtäquidistanter Daten
- 3 Schnelle Summation
- 4 Zusammenfassung

Diskrete Fourier-Transformationen

Trigonometrisches Polynom

$$f : \mathbb{T}^3 \rightarrow \mathbb{C}, \quad f(\mathbf{x}) = \sum_{\mathbf{k} \in I_N} \hat{f}_{\mathbf{k}} e^{-2\pi i \mathbf{k} \mathbf{x}}, \quad \hat{f}_{\mathbf{k}} \in \mathbb{C}$$

$$\mathbf{N} \in 2\mathbb{N}^3, \quad I_N := \mathbf{N}\mathbb{T} \cap \mathbb{Z}^3, \quad \mathbf{l}/\mathbf{N} := \left(\frac{l_0}{N_0}, \frac{l_1}{N_1}, \frac{l_2}{N_2}\right)^T \in \mathbb{T}^3$$

Aufgabenstellung 3D-DFT (Realisierung mit 3D-FFT)

$$f_{\mathbf{l}} = f(\mathbf{l}/\mathbf{N}) = \sum_{\mathbf{k} \in I_N} \hat{f}_{\mathbf{k}} e^{-2\pi i \mathbf{k}(\mathbf{l}/\mathbf{N})}, \quad \mathbf{l} \in I_N$$

$$\Rightarrow \mathcal{O}(|I_N| \log |I_N|) \text{ statt } \mathcal{O}(|I_N|^2)$$

Aufgabenstellung 3D-NDFT (Realisierung mit 3D-NFFT)

$$f(\mathbf{x}_j) = \sum_{\mathbf{k} \in I_N} \hat{f}_{\mathbf{k}} e^{-2\pi i \mathbf{k} \mathbf{x}_j}, \quad j \in \mathcal{M} := \{1, \dots, M\}$$

$$\Rightarrow \mathcal{O}(|I_N| \log(|I_N|) + \log^3(\frac{1}{\varepsilon})M) \text{ statt } \mathcal{O}(|I_N|M)$$

Approximationsidee der NFFT

Ansatz [Dutt, Rohklin 93, Beylkin 95, Steidl 96, ...]

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in I_N} \hat{f}_{\mathbf{k}} e^{-2\pi i \mathbf{k} \mathbf{x}} \approx \check{s}(\mathbf{x}) := \sum_{\mathbf{l} \in I_n} g_{\mathbf{l}} \tilde{\varphi}(\mathbf{x} - \mathbf{l}/n)$$

mit $N \leq n$

Entwicklung in eine Fourier-Reihe

$$\check{s}(\mathbf{x}) = \sum_{\mathbf{r} \in \mathbb{Z}^3} \sum_{\mathbf{k} \in I_n} \hat{g}_{\mathbf{k}} c_{\mathbf{k} + \mathbf{n} \odot \mathbf{r}}(\tilde{\varphi}) e^{-2\pi i (\mathbf{k} + \mathbf{n} \odot \mathbf{r}) \mathbf{x}}$$

mit

$$\hat{g}_{\mathbf{k}} := \sum_{\mathbf{l} \in I_n} g_{\mathbf{l}} e^{+2\pi i \mathbf{k}(\mathbf{l}/n)}, \quad \mathbf{k} \in I_n$$

$$c_{\mathbf{k}}(\tilde{\varphi}) := \int_{\mathbb{T}^3} \tilde{\varphi}(\mathbf{x}) e^{+2\pi i \mathbf{k} \mathbf{x}} d\mathbf{x}, \quad \mathbf{k} \in \mathbb{Z}^3$$

Approximationsidee der NFFT

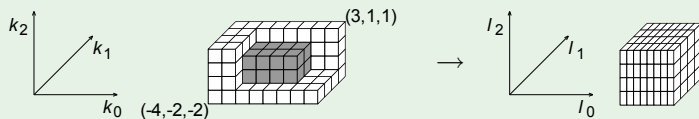
Approximation im Frequenzbereich

$$\hat{g}_{\mathbf{k}} = \begin{cases} \frac{\hat{f}_{\mathbf{k}}}{c_{\mathbf{k}}(\tilde{\varphi})} & : \mathbf{k} \in I_N \\ 0 & : \mathbf{k} \in I_n \setminus I_N \end{cases}$$

3D-FFT der Größe $n_0 \times n_1 \times n_2$

$$g_{\mathbf{l}} = \frac{1}{|I_n|} \sum_{\mathbf{k} \in I_N} \hat{g}_{\mathbf{k}} e^{-2\pi i \mathbf{k}(\mathbf{l}/n)}, \quad \mathbf{l} \in I_n$$

Beispiel für $N = (4, 2, 2)$ und $n = (8, 4, 4)$



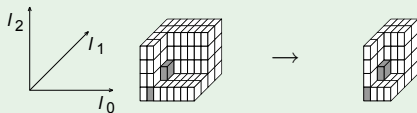
Approximationsidee der NFFT

Approximation im Ortsbereich

$$s(\mathbf{x}) := \sum_{\mathbf{l} \in I_{n,m}(\mathbf{x})} g_{\mathbf{l}} \tilde{\varphi}(\mathbf{x} - \mathbf{l}/\mathbf{n})$$

$$I_{n,m}(\mathbf{x}) := \{\mathbf{l} = \lfloor \mathbf{n} \odot \mathbf{x} \rfloor + \mathbf{r} \in I_n : \mathbf{r} \in [-m, m+2]^3\}$$

Beispiel für $I_{n,m}(\mathbf{x})$ mit $m = 1$



Datenparallelisierung der FFT [FFTW]

Prozessorenetz

$$\mathcal{P} := \{\mathbf{p} \in \mathbb{Z}^3 : \mathbf{0} \leq \mathbf{p} < \mathbf{P}\}, \quad \mathbf{P} \in \mathbb{N}^3$$

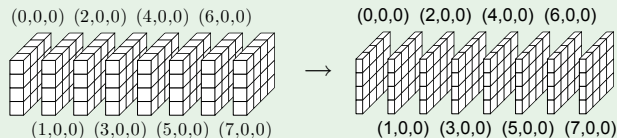
Aufgabenstellung parallele FFT

geg.: $\{I_n^{\mathbf{p}} \subseteq I_n : \mathbf{p} \in \mathcal{P}\}$ Zerlegung von I_n
 $\hat{g}_{\mathbf{k}} \in \mathbb{C}, \mathbf{k} \in I_n^{\mathbf{p}}$

ges.:

$$g_{\mathbf{l}} = \sum_{\mathbf{r} \in \mathcal{P}} \sum_{\mathbf{k} \in I_n^{\mathbf{r}}} \hat{g}_{\mathbf{k}} e^{-2\pi i \mathbf{k}(\mathbf{l}/n)}, \quad \mathbf{l} \in I_n^{\mathbf{p}}$$

Beispiel für $\mathbf{N} = (4, 2, 2)$, $\mathbf{n} = (8, 4, 4)$, $\mathbf{P} = (8, 1, 1)$



Parallelisierung der NFFT

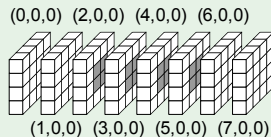
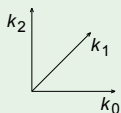
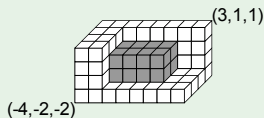
Approximation im Frequenzbereich - seriell

$$\hat{g}_{\mathbf{k}} := \frac{\hat{f}_{\mathbf{k}}}{|I_n| c_{\mathbf{k}}(\tilde{\varphi})}, \quad \mathbf{k} \in I_N$$

Approximation im Frequenzbereich - parallel

$$\hat{g}_{\mathbf{k}} := \frac{\hat{f}_{\mathbf{k}}}{|I_n| c_{\mathbf{k}}(\tilde{\varphi})}, \quad \mathbf{k} \in \mathcal{I}_N^p := I_n^p \cap I_N$$

Beispiel für $N = (4, 2, 2)$, $n = (8, 4, 4)$, $P = (8, 1, 1)$



Parallelisierung der NFFT

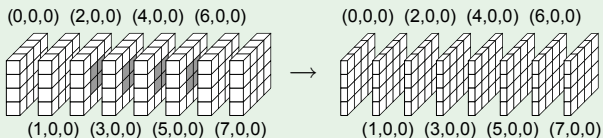
3D-FFT - seriell

$$g_l := \sum_{k \in I_N} \hat{g}_k e^{-2\pi i k(l/n)}, \quad l \in I_n$$

3D-FFT - parallel

$$g_l := \sum_{r \in \mathcal{P}} \sum_{k \in \mathcal{I}_N^r} \hat{g}_k e^{-2\pi i k(l/n)}, \quad l \in I_n^{\mathcal{P}}$$

Beispiel für $N = (4, 2, 2)$, $n = (8, 4, 4)$, $\mathcal{P} = (8, 1, 1)$



Parallelisierung der NFFT

Approximation im Ortsbereich - seriell

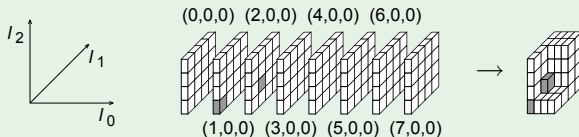
$$s_j := \sum_{\mathbf{l} \in I_{n,m}(\mathbf{x}_j)} g_{\mathbf{l}} \tilde{\varphi}(\mathbf{x}_j - \mathbf{l}/\mathbf{n}), \quad j \in \mathcal{M}$$

Approximation im Ortsbereich - parallel

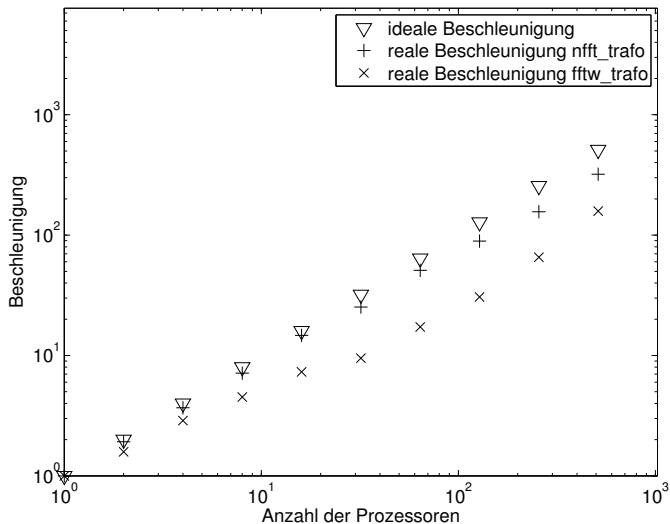
$$s_j := \sum_{\mathbf{l} \in I_{n,m}(\mathbf{x}_j)} g_{\mathbf{l}} \tilde{\varphi}(\mathbf{x}_j - \mathbf{l}/\mathbf{n}), \quad j \in \mathcal{M}^P$$

$$\mathcal{M}^P := \{j \in \mathcal{M} : \lfloor \mathbf{n} \odot \mathbf{x}_j \rfloor - m\mathbf{1} \in I_n^P\}$$

Beispiel für $N = (4, 2, 2)$, $\mathbf{n} = (8, 4, 4)$, $P = (8, 1, 1)$, $m = 1$

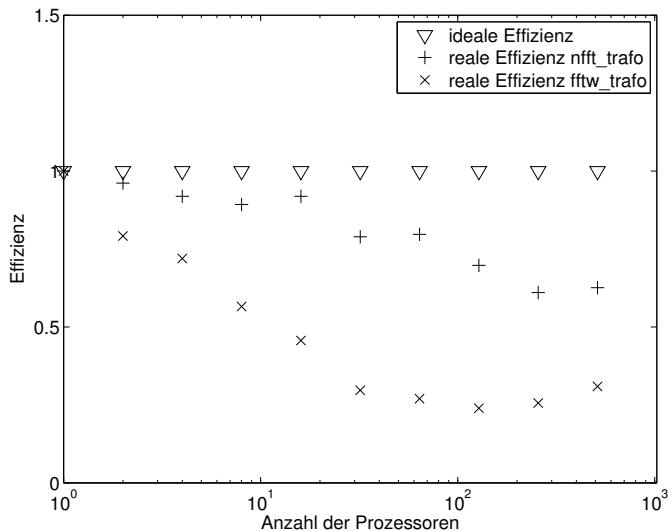


Beschleunigung der parallelen NFFT auf JUGENE



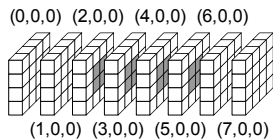
$$\mathbf{N} = (196, 196, 196), \quad \mathbf{n} = (392, 392, 392), \quad M = 196^3, \quad m = 4$$

Effizienz der parallelen NFFT auf JUGENE

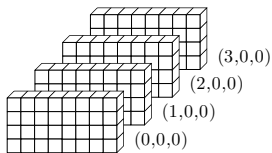


$$\mathbf{N} = (196, 196, 196), \mathbf{n} = (392, 392, 392), M = 196^3, m = 4$$

Datenaufteilung der parallelen FFTW

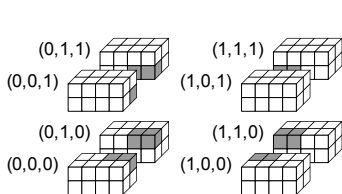


(a) Initialisierung der Fourier-Koeffizienten

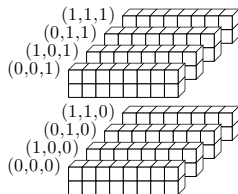


(b) 1D-FFT in erster Dimension

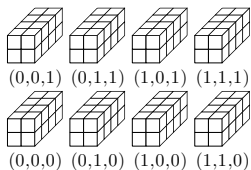
3D-Datenaufteilung [Eleftheriou et al. 03]



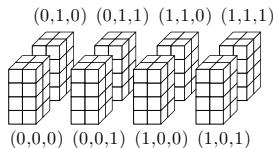
(a) Initialisierung der Fourier-Koeffizienten



(b) 1D-FFT in erster Dimension



(c) 1D-FFT in zweiter Dimension



(d) 1D-FFT in dritter Dimension

Grenzen der Parallelisierung der NFFT

$$N_0 = N_1 = N_2 = N, \quad n_0 = n_1 = n_2 = 2N, \quad M = N^3, \quad m = 4$$

Begrenzung der Prozessorenanzahl mit FFTW

$$P \leq \max \{ \min \{ n_0, n_1 \}, \min \{ n_1, n_2 \}, \min \{ n_2, n_0 \} \} =: P_{\max}$$

RAM in GByte	0,5	1	4
$N_0 = N_1 = N_2$	885	1252	2505
P_{\max}	1770	2504	5010
	JUGENE	CHiC	JUMP

Begrenzung der Prozessorenanzahl mit BlueGeneFFT

$$P \leq \min \{ n_0 n_1, n_1 n_2, n_2 n_0 \}$$

\Rightarrow Ab $n_0 = n_1 = n_2 = 256$ bereits 65536 Prozessoren nutzbar

P	64.000
$N_0 = N_1 = N_2$	4.230
$N_0 \times N_1 \times N_2$	$7,57 \cdot 10^{10}$

Schnelle Summation

Aufgabenstellung

geg.: $\mathbf{x}_l \in \mathbb{R}^3$, $l = 1, \dots, M_1$, $\mathbf{y}_j \in \mathbb{R}^3$, $j = 1, \dots, M_2$

$a_l \in \mathbb{C}$, $l = 1, \dots, M_1$, $K : \mathbb{R}_+ \rightarrow \mathbb{R}$

ges.:

$$h_j = \sum_{l=1}^{M_1} a_l K(\|\mathbf{y}_j - \mathbf{x}_l\|_2), \quad j = 1, \dots, M_2$$

Beispiele für Kernfunktionen $K(\|\mathbf{x}\|_2)$

$$\frac{1}{\|\mathbf{x}\|_2}, \quad \log \|\mathbf{x}\|_2, \quad (\|\mathbf{x}\|_2^2 + c^2)^{\frac{1}{2}}$$

Anwendung in Partikelsimulationen

$$\phi(\mathbf{x}_j) := \sum_{\substack{l=1 \\ l \neq j}}^{M_1} \frac{q_l}{\|\mathbf{x}_j - \mathbf{x}_l\|_2}, \quad j = 1, \dots, M_2$$

Schnelle Summation

Matrix-Vektor-Notation

$$\mathbf{h} = \mathbf{K} \mathbf{a}$$

mit $\mathbf{h} = (h_j)_{j=1}^{M_2}$, $\mathbf{K} = (K(\|\mathbf{y}_j - \mathbf{x}_l\|_2))_{j,l=1}^{M_2, M_1}$, $\mathbf{a} = (a_l)_{l=1}^{M_1} \in \mathbb{C}^{M_1}$

Approximation

$$\mathbf{h} = \mathbf{K} \mathbf{a} \approx \mathbf{A}_2 \mathbf{D} \mathbf{A}_1^H \mathbf{a} + \mathbf{K}_{NF} \mathbf{a}$$

mit

$$\mathbf{A}_2 = (e^{-2\pi i k y_j})_{j=1, k \in I_N}^{M_2} \in \mathbb{C}^{M_2 \times |I_N|}, \quad \mathbf{D} \in \mathbb{R}^{|I_N| \times |I_N|},$$

$$\mathbf{A}_1 = (e^{-2\pi i k x_l})_{l=1, k \in I_N}^{M_1} \in \mathbb{C}^{M_1 \times |I_N|}, \quad \mathbf{K}_{NF} \in \mathbb{R}^{M_2 \times M_1}$$

$$\Rightarrow \mathcal{O}(|I_N| \log |I_N| + (M_1 + M_2)(\log \frac{1}{\varepsilon})^3) \text{ statt } \mathcal{O}(M_1 M_2)$$

Parallelisierung

Zusammenfassung

