

Schnelle Verfahren für Vielteilchensysteme

Seminar „Vielteilchensysteme“

Steffen Börm

University of Kiel, Germany

1. November 2022

Aufgabenstellung

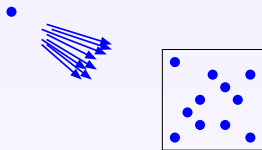
Ziel: Auswertung der Coulomb-Kräfte, die auf geladene Teilchen wirken.



$$f_{ij} = q_i q_j g(x_i, y_j), \quad g(x, y) = c \frac{y_j - x_i}{\|y_j - x_i\|^3},$$

Aufgabenstellung

Ziel: Auswertung der Coulomb-Kräfte, die auf geladene Teilchen wirken.

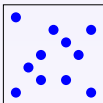
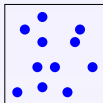


$$f_{ij} = q_i q_j g(x_i, y_j),$$

$$g(x, y) = c \frac{y_j - x_i}{\|y_j - x_i\|^3},$$

$$f_i = q_i \sum_{j=1}^n q_j g(x_i, y_j)$$

Aufgabenstellung



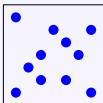
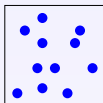
Ziel: Auswertung der Coulomb-Kräfte, die auf geladene Teilchen wirken.

$$f_{ij} = q_i q_j g(x_i, y_j), \quad g(x, y) = c \frac{y_j - x_i}{\|y_j - x_i\|^3},$$

$$f_i = q_i \sum_{j=1}^n q_j g(x_i, y_j) \quad \text{für alle } i \in [1 : n].$$

Problem: Aufwand wächst quadratisch mit der Anzahl der Teilchen.

Aufgabenstellung



Ziel: Auswertung der Coulomb-Kräfte, die auf geladene Teilchen wirken.

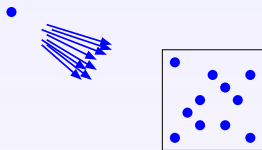
$$f_{ij} = q_i q_j g(x_i, y_j), \quad g(x, y) = c \frac{y_j - x_i}{\|y_j - x_i\|^3},$$
$$f_i = q_i \sum_{j=1}^n q_j g(x_i, y_j) \quad \text{für alle } i \in [1 : n].$$

Problem: Aufwand wächst quadratisch mit der Anzahl der Teilchen.

Einfache Lösungsansätze:

- Vektorisierung (beispielsweise vier Kräfte simultan mit AVX),
- Parallelisierung (beispielsweise auf Mehrkernprozessoren),
- verteiltes Rechnen (beispielsweise ein Computer pro Teilgebiet).

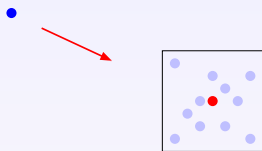
Ersatzteilchen



Idee: Weit entfernte Ansammlungen von Teilchen können durch ein einziges Teilchen ersetzt werden, das ungefähr dasselbe Kraftfeld hervorruft.

Dadurch lässt sich der Rechenaufwand spürbar reduzieren.

Ersatzteilchen



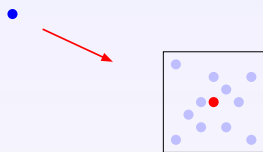
Idee: Weit entfernte Ansammlungen von Teilchen können durch ein einziges Teilchen ersetzt werden, das ungefähr dasselbe Kraftfeld hervorruft.

Dadurch lässt sich der Rechenaufwand spürbar reduzieren.

Berechnung vereinfacht sich zu

$$\sum_{j=1}^n q_j g(x_i, y_j) \approx \underbrace{\sum_{j=1}^n q_j}_{=:\hat{q}} g(x_i, \hat{y}) = \hat{q} g(x_i, \hat{y}).$$

Ersatzteilchen



Idee: Weit entfernte Ansammlungen von Teilchen können durch ein einziges Teilchen ersetzt werden, das ungefähr dasselbe Kraftfeld hervorruft.

Dadurch lässt sich der Rechenaufwand spürbar reduzieren.

Berechnung vereinfacht sich zu

$$\sum_{j=1}^n q_j g(x_i, y_j) \approx \underbrace{\sum_{j=1}^n q_j}_{=: \hat{q}} g(x_i, \hat{y}) = \hat{q} g(x_i, \hat{y}).$$

Genauigkeit lässt sich abschätzen durch

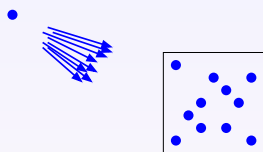
$$\|g(x_i, y_j) - g(x_i, \hat{y})\| \lesssim \frac{\text{diam}(s)}{\text{dist}(x_i, s)^3}, \quad s \text{ ist eine Kugel um } \hat{y} \text{ mit } \{y_1, \dots, y_n\} \subseteq s.$$

Interpolation

Ziel: Höhere Genauigkeit ohne deutlich höheren Rechenaufwand.

Ansatz: Mehrere Ersatzteilchen anstelle eines einzigen.

Konstruktion: Interpolation der Funktion $y \mapsto g(x_i, y)$ in s .
Interpolationspunkte $(\xi_{s,\nu})_{\nu \in M}$, Lagrange-Polynomen $(\ell_{s,\nu})_{\nu \in M}$.



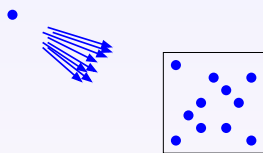
$$g(x, y) \approx \sum_{\nu \in M} \ell_{s,\nu}(y) g(x, \xi_{s,\nu}).$$

Interpolation

Ziel: Höhere Genauigkeit ohne deutlich höheren Rechenaufwand.

Ansatz: Mehrere Ersatzteilchen anstelle eines einzigen.

Konstruktion: Interpolation der Funktion $y \mapsto g(x_i, y)$ in s .
Interpolationspunkte $(\xi_{s,\nu})_{\nu \in M}$, Lagrange-Polynomen $(\ell_{s,\nu})_{\nu \in M}$.



$$g(x, y) \approx \sum_{\nu \in M} \ell_{s,\nu}(y) g(x, \xi_{s,\nu}).$$

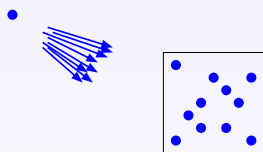
$$f_i = q_i \sum_{j=1}^n q_j g(x_i, y_j)$$

Interpolation

Ziel: Höhere Genauigkeit ohne deutlich höheren Rechenaufwand.

Ansatz: Mehrere Ersatzteilchen anstelle eines einzigen.

Konstruktion: Interpolation der Funktion $y \mapsto g(x_i, y)$ in s .
Interpolationspunkte $(\xi_{s,\nu})_{\nu \in M}$, Lagrange-Polynomen $(\ell_{s,\nu})_{\nu \in M}$.



$$g(x, y) \approx \sum_{\nu \in M} \ell_{s,\nu}(y) g(x, \xi_{s,\nu}).$$

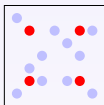
$$f_i = q_i \sum_{j=1}^n q_j g(x_i, y_j) \approx q_i \sum_{j=1}^n q_j \sum_{\nu \in M} \ell_{s,\nu}(y_j) g(x_i, \xi_{s,\nu})$$

Interpolation

Ziel: Höhere Genauigkeit ohne deutlich höheren Rechenaufwand.

Ansatz: Mehrere Ersatzteilchen anstelle eines einzigen.

Konstruktion: Interpolation der Funktion $y \mapsto g(x_i, y)$ in s .
Interpolationspunkte $(\xi_{s,\nu})_{\nu \in M}$, Lagrange-Polynomen $(\ell_{s,\nu})_{\nu \in M}$.



$$g(x, y) \approx \sum_{\nu \in M} \ell_{s,\nu}(y) g(x, \xi_{s,\nu}).$$

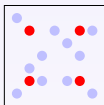
$$\begin{aligned} f_i &= q_i \sum_{j=1}^n q_j g(x_i, y_j) \approx q_i \sum_{j=1}^n q_j \sum_{\nu \in M} \ell_{s,\nu}(y_j) g(x_i, \xi_{s,\nu}) \\ &= q_i \sum_{\nu \in M} \left(\sum_{j=1}^n q_j \ell_{s,\nu}(y_j) \right) g(x_i, \xi_{s,\nu}) \end{aligned}$$

Interpolation

Ziel: Höhere Genauigkeit ohne deutlich höheren Rechenaufwand.

Ansatz: Mehrere Ersatzteilchen anstelle eines einzigen.

Konstruktion: Interpolation der Funktion $y \mapsto g(x_i, y)$ in s .
Interpolationspunkte $(\xi_{s,\nu})_{\nu \in M}$, Lagrange-Polynomen $(\ell_{s,\nu})_{\nu \in M}$.



$$g(x, y) \approx \sum_{\nu \in M} \ell_{s,\nu}(y) g(x, \xi_{s,\nu}).$$

$$\begin{aligned} f_i &= q_i \sum_{j=1}^n q_j g(x_i, y_j) \approx q_i \sum_{j=1}^n q_j \sum_{\nu \in M} \ell_{s,\nu}(y_j) g(x_i, \xi_{s,\nu}) \\ &= q_i \sum_{\nu \in M} \left(\sum_{j=1}^n q_j \ell_{s,\nu}(y_j) \right) g(x_i, \xi_{s,\nu}) = q_i \sum_{\nu \in M} \hat{q}_{s,\nu} g(x_i, \xi_{s,\nu}). \end{aligned}$$

Zulässigkeit

Problem: Die Genauigkeit der Interpolation hängt von

$$\eta := \frac{\text{diam}(s)}{\text{dist}(x_i, s)}$$

ab, dem Verhältnis zwischen Durchmesser und Abstand.

Zulässigkeit

Problem: Die Genauigkeit der Interpolation hängt von

$$\eta := \frac{\text{diam}(S)}{\text{dist}(x_i, S)}$$

ab, dem Verhältnis zwischen Durchmesser und Abstand.
Für den Interpolationsfehler m -ten Grades gilt die Abschätzung

$$\|f_i - \tilde{f}_i\| \lesssim \frac{1}{\text{dist}(x_i, S)^2} \left(\frac{\eta}{\eta + 1} \right)^m.$$

Zulässigkeit

Problem: Die Genauigkeit der Interpolation hängt von

$$\eta := \frac{\text{diam}(S)}{\text{dist}(x_i, S)}$$

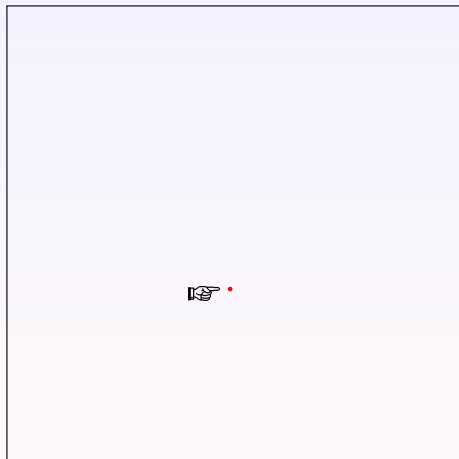
ab, dem Verhältnis zwischen Durchmesser und Abstand.
Für den Interpolationsfehler m -ten Grades gilt die Abschätzung

$$\|f_i - \tilde{f}_i\| \lesssim \frac{1}{\text{dist}(x_i, S)^2} \left(\frac{\eta}{\eta + 1} \right)^m.$$

Konsequenz: Wir können die Kräfte nicht global approximieren, sondern nur in einem gewissen Abstand von x_i .

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

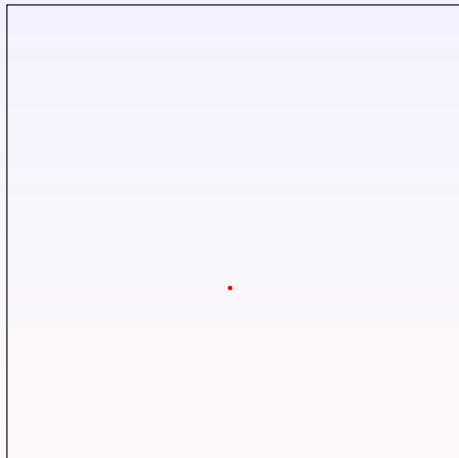


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

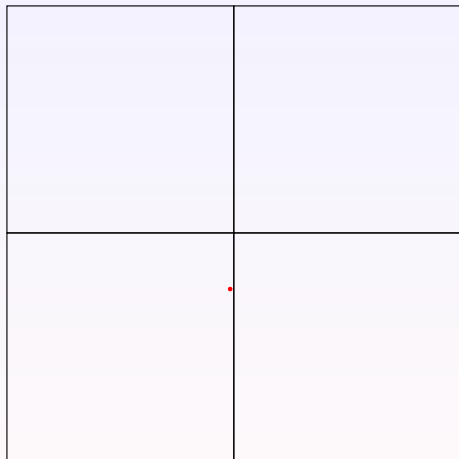


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

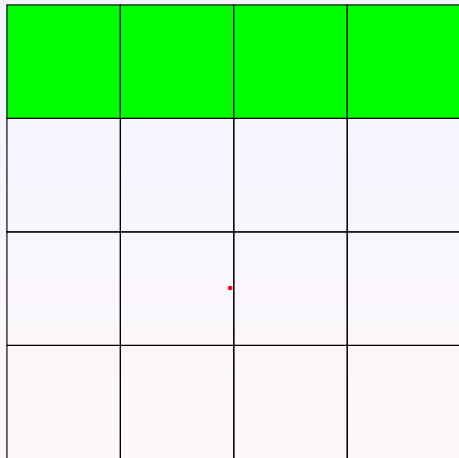


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

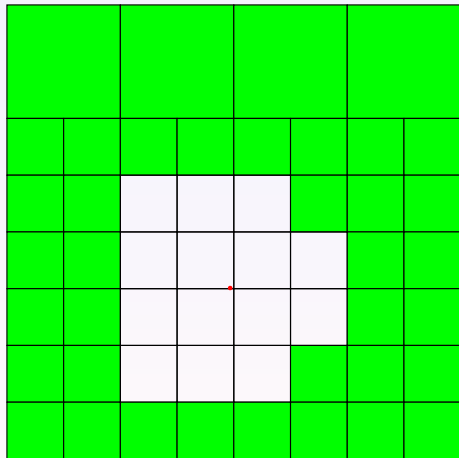


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

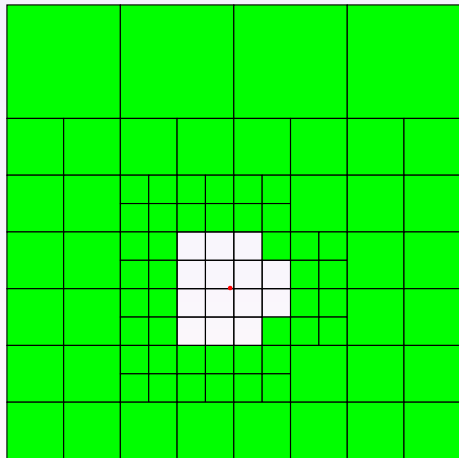


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

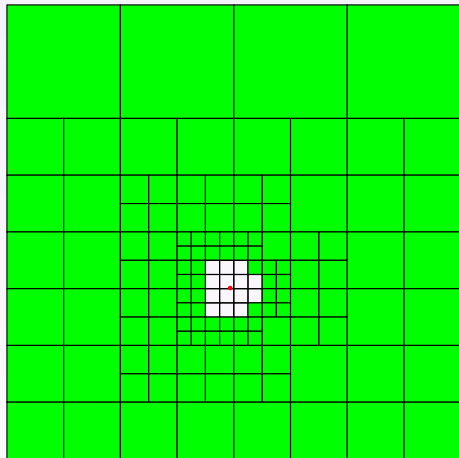


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

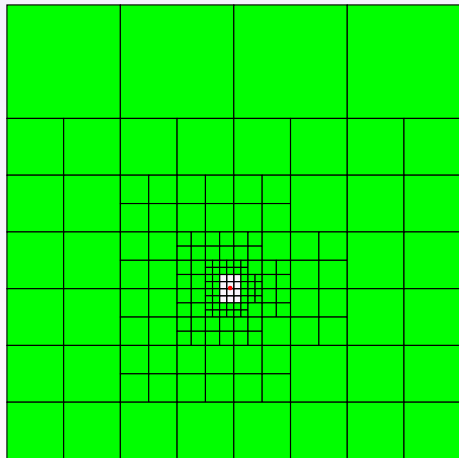


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

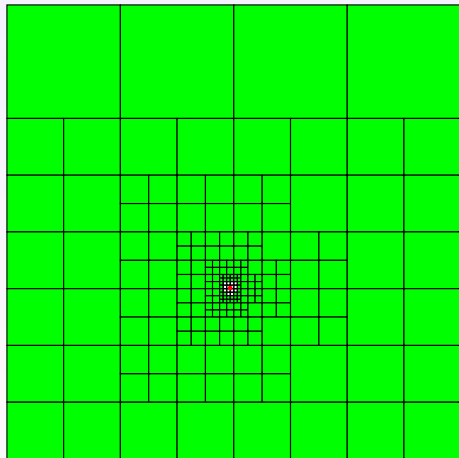


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.

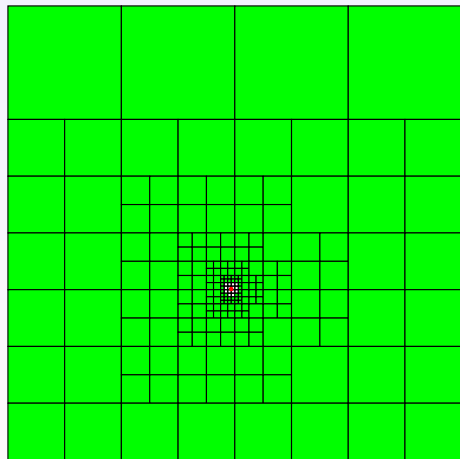


Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.
- Hinreichend kleine Teilgebiete können wir direkt behandeln.

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.



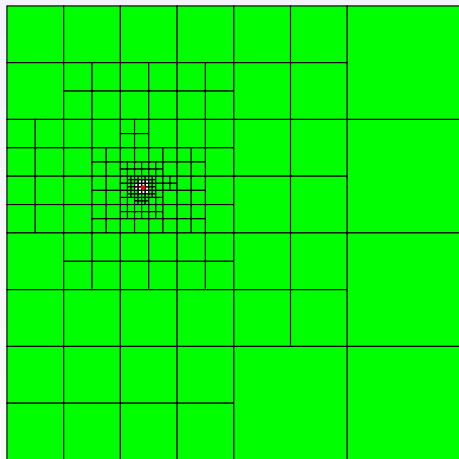
Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.
- Hinreichend kleine Teilgebiete können wir direkt behandeln.

Komplexität: Aufwand $\mathcal{O}(m^2 \log(\frac{H}{h}))$ für die Auswertung in einem Punkt x_i .

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.



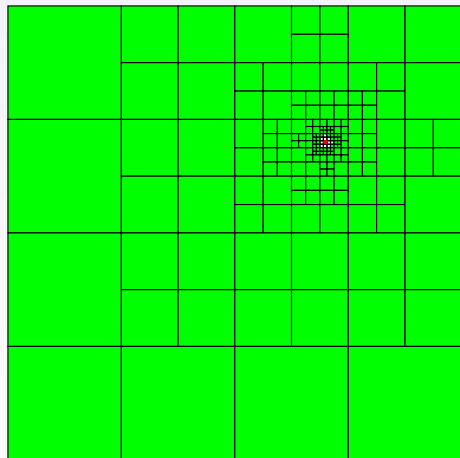
Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.
- Hinreichend kleine Teilgebiete können wir direkt behandeln.

Komplexität: Aufwand $\mathcal{O}(m^2 \log(\frac{H}{h}))$ für die Auswertung in einem Punkt x_i .

Partition

Idee: Wir zerlegen den Raum in Teilgebiete und prüfen deren Zulässigkeit.



Rekursion:

- Falls ein Teilgebiet zulässig ist, benutzen wir Interpolation.
- Ansonsten wird es weiter zerlegt.
- Hinreichend kleine Teilgebiete können wir direkt behandeln.

Komplexität: Aufwand $\mathcal{O}(m^2 \log(\frac{H}{h}))$ für die Auswertung in einem Punkt x_i .

Vorwärtstransformation

Problem: Um das Kraftfeld in unterschiedlichen Punkten auszuwerten, benötigen wir Interpolationskoeffizienten auf unterschiedlichen Gebieten.

Idee: Wir berechnen alle Koeffizienten für eine Hierarchie von Teilgebieten im Voraus.

Vorwärtstransformation

Problem: Um das Kraftfeld in unterschiedlichen Punkten auszuwerten, benötigen wir Interpolationskoeffizienten auf unterschiedlichen Gebieten.

Idee: Wir berechnen alle Koeffizienten für eine Hierarchie von Teilgebieten im Voraus.

Rekursion: Falls wir für alle Gebiete denselben Polynomgrad verwenden, gilt

$$\ell_{s,\nu} = \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \ell_{s',\nu'}.$$

Falls s in disjunkte Teilgebiete s' zerlegt wird, folgt

$$\hat{q}_{s,\nu} = \sum_{y_j \in s} q_j \ell_{s,\nu}(y_j)$$

Vorwärtstransformation

Problem: Um das Kraftfeld in unterschiedlichen Punkten auszuwerten, benötigen wir Interpolationskoeffizienten auf unterschiedlichen Gebieten.

Idee: Wir berechnen alle Koeffizienten für eine Hierarchie von Teilgebieten im Voraus.

Rekursion: Falls wir für alle Gebiete denselben Polynomgrad verwenden, gilt

$$\ell_{s,\nu} = \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \ell_{s',\nu'}.$$

Falls s in disjunkte Teilgebiete s' zerlegt wird, folgt

$$\hat{q}_{s,\nu} = \sum_{y_j \in s} q_j \ell_{s,\nu}(y_j) = \sum_{s' \subseteq s} \sum_{y_j \in s'} q_j \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \ell_{s',\nu'}(y_j)$$

Vorwärtstransformation

Problem: Um das Kraftfeld in unterschiedlichen Punkten auszuwerten, benötigen wir Interpolationskoeffizienten auf unterschiedlichen Gebieten.

Idee: Wir berechnen alle Koeffizienten für eine Hierarchie von Teilgebieten im Voraus.

Rekursion: Falls wir für alle Gebiete denselben Polynomgrad verwenden, gilt

$$\ell_{s,\nu} = \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \ell_{s',\nu'}.$$

Falls s in disjunkte Teilgebiete s' zerlegt wird, folgt

$$\hat{q}_{s,\nu} = \sum_{y_j \in s} q_j \ell_{s,\nu}(y_j) = \sum_{s' \subseteq s} \sum_{y_j \in s'} q_j \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \ell_{s',\nu'}(y_j) = \sum_{s' \subseteq s} \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \hat{q}_{s',\nu'}.$$

Vorwärtstransformation

Problem: Um das Kraftfeld in unterschiedlichen Punkten auszuwerten, benötigen wir Interpolationskoeffizienten auf unterschiedlichen Gebieten.

Idee: Wir berechnen alle Koeffizienten für eine Hierarchie von Teilgebieten im Voraus.

Rekursion: Falls wir für alle Gebiete denselben Polynomgrad verwenden, gilt

$$\ell_{s,\nu} = \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \ell_{s',\nu'}.$$

Falls s in disjunkte Teilgebiete s' zerlegt wird, folgt

$$\hat{q}_{s,\nu} = \sum_{y_j \in s} q_j \ell_{s,\nu}(y_j) = \sum_{s' \subseteq s} \sum_{y_j \in s'} q_j \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \ell_{s',\nu'}(y_j) = \sum_{s' \subseteq s} \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \hat{q}_{s',\nu'}.$$

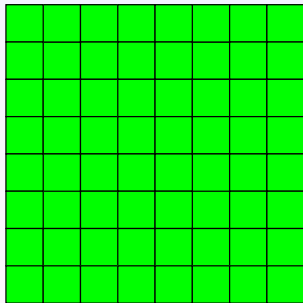
Wir können also die Koeffizienten für s aus denen für die Teilgebiete s' berechnen.

Vorbereitungsphase

Aufgabe: Alle Koeffizienten $\hat{q}_{s,\nu}$ für alle Teilgebiete sind zu berechnen.

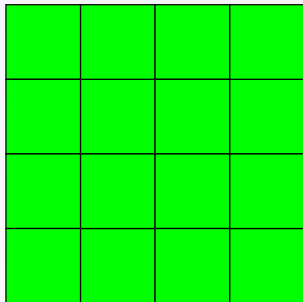
Kleinste Gebiete: Wir berechnen direkt

$$\hat{q}_{s,\nu} = \sum_{j \in s} \ell_{s,\nu}(y_j) q_j.$$



Vorbereitungsphase

Aufgabe: Alle Koeffizienten $\hat{q}_{s,\nu}$ für alle Teilgebiete sind zu berechnen.



Kleinste Gebiete: Wir berechnen direkt

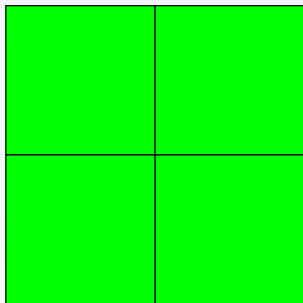
$$\hat{q}_{s,\nu} = \sum_{j \in s} \ell_{s,\nu}(y_j) q_j.$$

Größere Gebiete: Wir berechnen rekursiv

$$\hat{q}_{s,\nu} = \sum_{s' \subseteq s} \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \hat{q}_{s',\nu'}.$$

Vorbereitungsphase

Aufgabe: Alle Koeffizienten $\hat{q}_{s,\nu}$ für alle Teilgebiete sind zu berechnen.



Kleinste Gebiete: Wir berechnen direkt

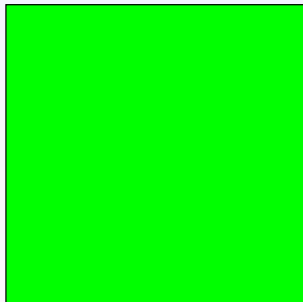
$$\hat{q}_{s,\nu} = \sum_{j \in s} \ell_{s,\nu}(y_j) q_j.$$

Größere Gebiete: Wir berechnen rekursiv

$$\hat{q}_{s,\nu} = \sum_{s' \subseteq s} \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \hat{q}_{s',\nu'}.$$

Vorbereitungsphase

Aufgabe: Alle Koeffizienten $\hat{q}_{s,\nu}$ für alle Teilgebiete sind zu berechnen.



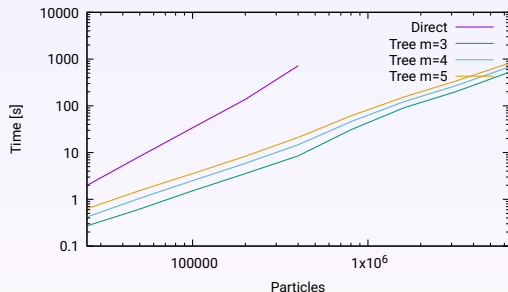
Kleinste Gebiete: Wir berechnen direkt

$$\hat{q}_{s,\nu} = \sum_{j \in s} \ell_{s,\nu}(y_j) q_j.$$

Größere Gebiete: Wir berechnen rekursiv

$$\hat{q}_{s,\nu} = \sum_{s' \subseteq s} \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \hat{q}_{s',\nu'}.$$

Experiment: Baumalgorithmus

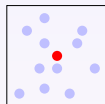


Teilchen	Direkt	Vorw.	Kopplung	Fehler
25 000	1.99	0.05	0.37	7.71 ₋₆
50 000	8.29	0.11	0.93	6.52 ₋₆
100 000	33.99	0.23	2.26	5.18 ₋₆
200 000	137.32	0.46	5.40	4.50 ₋₆
400 000	720.17	0.93	13.70	4.18 ₋₆
800 000		1.87	43.86	
1 600 000		3.83	118.10	
3 200 000		7.77	264.24	
6 400 000		15.61	661.20	

Ersatzziele

Bisher: Die Auswertung für einen Punkt benötigt $\sim m^2 \log(\frac{H}{h})$ Operationen.

Ziel: Aufwand weiter reduzieren, um größere Systeme rechnen zu können.



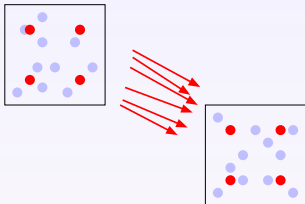
Ansatz: Falls ein zweiter Punkt x_k nahe an x_i liegt, werden auf ihn ungefähr dieselben Kräfte wirken wie auf x_i .

→ Es sollte genügen, diese Kräfte nur einmal zu berechnen.

- **Realisierung:** Wir ersetzen auch die Zielpunkte x_i in Teilgebieten t durch „Ersatzziele“, aus denen wir die wirkenden Kräfte näherungsweise rekonstruieren können.

Symmetrische Interpolation

Ziel: Höhere Genauigkeit der Approximation.



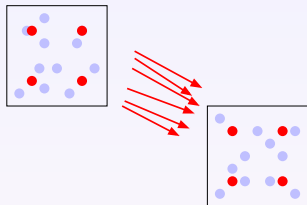
Ansatz: Interpolation in beiden Variablen.

$$g(x, y) \approx \sum_{\nu \in M} \sum_{\mu \in M} \ell_{t,\nu}(x) g(\xi_{t,\nu}, \xi_{s,\mu}) \ell_{s,\mu}(y)$$

gültig für Quellen in s und Ziele in t .

Symmetrische Interpolation

Ziel: Höhere Genauigkeit der Approximation.



Ansatz: Interpolation in beiden Variablen.

$$g(x, y) \approx \sum_{\nu \in M} \sum_{\mu \in M} \ell_{t,\nu}(x) g(\xi_{t,\nu}, \xi_{s,\mu}) \ell_{s,\mu}(y)$$

gültig für Quellen in s und Ziele in t .

Zulässigkeitsbedingung muss verschärft werden zu

$$\max\{\text{diam}(t), \text{diam}(s)\} \leq \eta \text{dist}(t, s),$$

um schnelle Konvergenz garantieren zu können.

Rückwärtstransformation

Erinnerung: Bei der Vorwärtstransformation haben wir die Koeffizienten auf einem größeren Gebiet s aus den Koeffizienten auf Teilgebieten $s' \subseteq s$ zusammengesetzt:

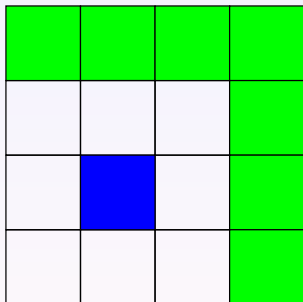
$$\hat{q}_{s,\nu} = \sum_{s' \subseteq s} \sum_{\nu' \in M} \ell_{s,\nu}(\xi_{s',\nu'}) \hat{q}_{s',\nu'}.$$

Kräfte können in die entgegengesetzte Richtung weitergegeben werden. Falls wir die Kräfte $\hat{f}_{t,\mu}$ für ein größeres Gebiet t kennen, erhalten wir für ein Teilgebiet $t' \subseteq t$

$$\hat{f}_{t',\mu'} = \sum_{\mu \in M} \ell_{t,\mu}(\xi_{t',\mu'}) \hat{f}_{t,\mu}.$$

Blockpartition

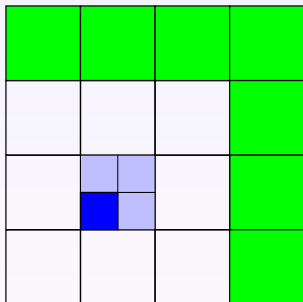
Idee: Wir verwenden hierarchisch unterteilte Teilgebiete sowohl für die Quellen als auch für die Ziele.



Kräfte werden für Interpolationspunkte in zulässigen Paare von Quell- und Zielgebieten berechnet.

Blockpartition

Idee: Wir verwenden hierarchisch unterteilte Teilgebiete sowohl für die Quellen als auch für die Ziele.

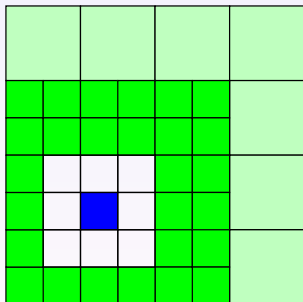


Kräfte werden für Interpolationspunkte in zulässigen Paare von Quell- und Zielgebieten berechnet.

Transfer: Auf größeren Gebieten berechnete Kräfte können auf kleineren interpoliert und so „vererbt“ werden.

Blockpartition

Idee: Wir verwenden hierarchisch unterteilte Teilgebiete sowohl für die Quellen als auch für die Ziele.



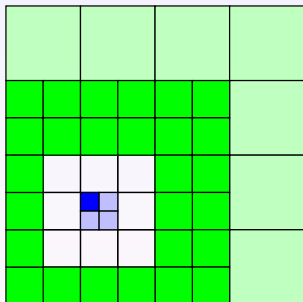
Kräfte werden für Interpolationspunkte in zulässigen Paare von Quell- und Zielgebieten berechnet.

Transfer: Auf größeren Gebieten berechnete Kräfte können auf kleineren interpoliert und so „vererbt“ werden.

Akkumulation: Kräfte für kleinere Teilgebiete werden den „geerbten“ Kräften hinzugefügt.

Blockpartition

Idee: Wir verwenden hierarchisch unterteilte Teilgebiete sowohl für die Quellen als auch für die Ziele.



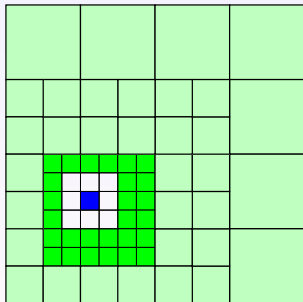
Kräfte werden für Interpolationspunkte in zulässigen Paare von Quell- und Zielgebieten berechnet.

Transfer: Auf größeren Gebieten berechnete Kräfte können auf kleineren interpoliert und so „vererbt“ werden.

Akkumulation: Kräfte für kleinere Teilgebiete werden den „geerbten“ Kräften hinzugefügt.

Blockpartition

Idee: Wir verwenden hierarchisch unterteilte Teilgebiete sowohl für die Quellen als auch für die Ziele.



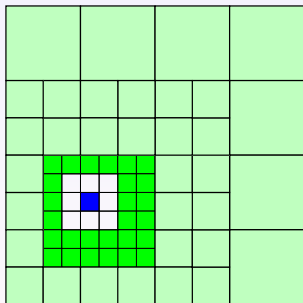
Kräfte werden für Interpolationspunkte in zulässigen Paare von Quell- und Zielgebieten berechnet.

Transfer: Auf größeren Gebieten berechnete Kräfte können auf kleineren interpoliert und so „vererbt“ werden.

Akkumulation: Kräfte für kleinere Teilgebiete werden den „geerbten“ Kräften hinzugefügt.

Blockpartition

Idee: Wir verwenden hierarchisch unterteilte Teilgebiete sowohl für die Quellen als auch für die Ziele.



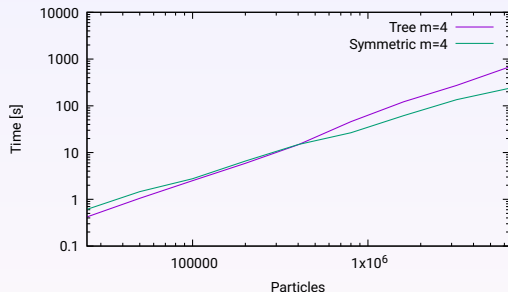
Kräfte werden für Interpolationspunkte in zulässigen Paare von Quell- und Zielgebieten berechnet.

Transfer: Auf größeren Gebieten berechnete Kräfte können auf kleineren interpoliert und so „vererbt“ werden.

Akkumulation: Kräfte für kleinere Teilgebiete werden den „geerbten“ Kräften hinzugefügt.

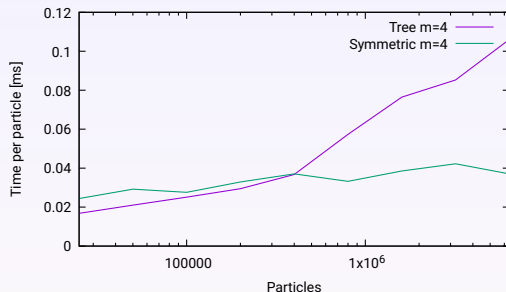
Ergebnis: Bei geeigneter Wahl der Rekursionstiefe genügen $\mathcal{O}(m^2 n)$ Operationen.

Experiment: Symmetrischer Algorithmus



Teilchen	Baum	Fehler	Symm	Fehler
25 000	0.42	7.71 ₋₆	0.61	5.64 ₋₆
50 000	1.04	6.52 ₋₆	1.45	5.84 ₋₆
100 000	2.49	5.18 ₋₆	2.74	6.67 ₋₆
200 000	5.86	4.50 ₋₆	6.55	6.40 ₋₆
400 000	14.63	4.18 ₋₆	14.76	6.34 ₋₆
800 000	45.73		26.43	
1 600 000	121.93		61.32	
3 200 000	272.01		134.28	
6 400 000	676.81		235.21	

Experiment: Symmetrischer Algorithmus



Teilchen	Baum	Fehler	Symm	Fehler
25 000	0.42	7.71 ₋₆	0.61	5.64 ₋₆
50 000	1.04	6.52 ₋₆	1.45	5.84 ₋₆
100 000	2.49	5.18 ₋₆	2.74	6.67 ₋₆
200 000	5.86	4.50 ₋₆	6.55	6.40 ₋₆
400 000	14.63	4.18 ₋₆	14.76	6.34 ₋₆
800 000	45.73		26.43	
1 600 000	121.93		61.32	
3 200 000	272.01		134.28	
6 400 000	676.81		235.21	

Verteiltes Rechnen

Problem: Falls wir sehr viele Teilchen berechnen wollen, ist der Rechenaufwand auch mit dem symmetrischen Verfahren noch recht hoch.

Ansatz: Wir beschäftigen mehrere Computer, von denen jeder für ein einziges oder mehrere Teilgebiete zuständig ist.

Algorithmus:

- 1 Vorwärtstransformation auf jedem Computer für sein Teilgebiet.
- 2
- 3 Berechnung der Kräfte in den Interpolationspunkten.
- 4 Rückwärtstransformation auf jedem Computer für sein Teilgebiet.

Verteiltes Rechnen

Problem: Falls wir sehr viele Teilchen berechnen wollen, ist der Rechenaufwand auch mit dem symmetrischen Verfahren noch recht hoch.

Ansatz: Wir beschäftigen mehrere Computer, von denen jeder für ein einziges oder mehrere Teilgebiete zuständig ist.

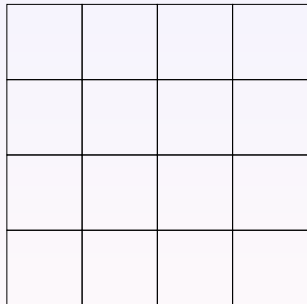
Algorithmus:

- 1 Vorwärtstransformation auf jedem Computer für sein Teilgebiet.
- 2 Austausch der Koeffizienten zwischen den Computern.
- 3 Berechnung der Kräfte in den Interpolationspunkten.
- 4 Rückwärtstransformation auf jedem Computer für sein Teilgebiet.

Kommunikationsphase

Aufgabe: Jeder Computer muss wissen,

- welche Daten er von anderen Computern empfangen wird und
- welche Daten er ihnen senden muss.

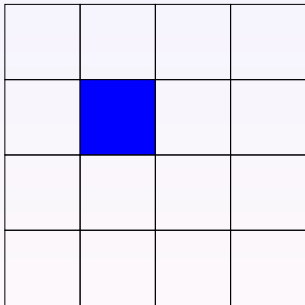


Ausgangspunkt: Zerlegung des gesamten Raums in Teilgebiete.

Kommunikationsphase

Aufgabe: Jeder Computer muss wissen,

- welche Daten er von anderen Computern empfangen wird und
- welche Daten er ihnen senden muss.



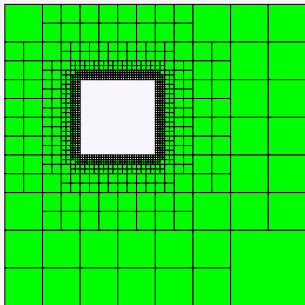
Ausgangspunkt: Zerlegung des gesamten Raums in Teilgebiete.

Verteiltes Rechnen: Jeder Computer ist für ein Teilgebiet zuständig.

Kommunikationsphase

Aufgabe: Jeder Computer muss wissen,

- welche Daten er von anderen Computern empfangen wird und
- welche Daten er ihnen senden muss.



Ausgangspunkt: Zerlegung des gesamten Raums in Teilgebiete.

Verteiltes Rechnen: Jeder Computer ist für ein Teilgebiet zuständig.

Sende- und Empfangsstruktur: Mit der Zulässigkeitsbedingung kann jeder Computer unabhängig von den anderen berechnen, mit welchen Teilgebieten er interagieren wird.

Kommunikationsphase

Aufgabe: Jeder Computer muss wissen,

- welche Daten er von anderen Computern empfangen wird und
- welche Daten er ihnen senden muss.

Ausgangspunkt: Zerlegung des gesamten Raums in Teilgebiete.

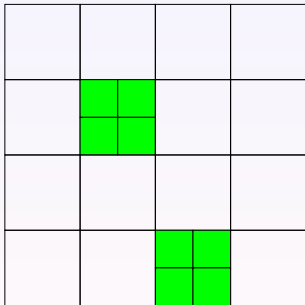
Verteiltes Rechnen: Jeder Computer ist für ein Teilgebiet zuständig.

Sende- und Empfangsstruktur: Mit der Zulässigkeitsbedingung kann jeder Computer unabhängig von den anderen berechnen, mit welchen Teilgebieten er interagieren wird.

Kommunikationsphase

Aufgabe: Jeder Computer muss wissen,

- welche Daten er von anderen Computern empfangen wird und
- welche Daten er ihnen senden muss.



Ausgangspunkt: Zerlegung des gesamten Raums in Teilgebiete.

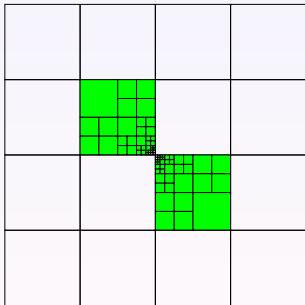
Verteiltes Rechnen: Jeder Computer ist für ein Teilgebiet zuständig.

Sende- und Empfangsstruktur: Mit der Zulässigkeitsbedingung kann jeder Computer unabhängig von den anderen berechnen, mit welchen Teilgebieten er interagieren wird.

Kommunikationsphase

Aufgabe: Jeder Computer muss wissen,

- welche Daten er von anderen Computern empfangen wird und
- welche Daten er ihnen senden muss.



Ausgangspunkt: Zerlegung des gesamten Raums in Teilgebiete.

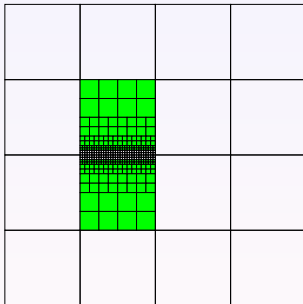
Verteiltes Rechnen: Jeder Computer ist für ein Teilgebiet zuständig.

Sende- und Empfangsstruktur: Mit der Zulässigkeitsbedingung kann jeder Computer unabhängig von den anderen berechnen, mit welchen Teilgebieten er interagieren wird.

Kommunikationsphase

Aufgabe: Jeder Computer muss wissen,

- welche Daten er von anderen Computern empfangen wird und
- welche Daten er ihnen senden muss.



Ausgangspunkt: Zerlegung des gesamten Raums in Teilgebiete.

Verteiltes Rechnen: Jeder Computer ist für ein Teilgebiet zuständig.

Sende- und Empfangsstruktur: Mit der Zulässigkeitsbedingung kann jeder Computer unabhängig von den anderen berechnen, mit welchen Teilgebieten er interagieren wird.

Zusammenfassung

Interpolation: Wir ersetzen die Quellen des Felds durch „Ersatzquellen“ in Interpolationspunkten, die ungefähr dasselbe Feld erzeugen.

$$\sum_{y_j \in S} q_j g(x, y_j) \approx \sum_{\nu \in M} \hat{q}_{s,\nu} g(x, \xi_{s,\nu}).$$

Zulässigkeitskriterium $\text{diam}(s) \leq \eta \text{ dist}(x_i, s)$ garantiert die Genauigkeit und führt zu einer Zerlegung des Raums in Teilgebiete.

Symmetrische Interpolation führt zu linearer, also optimaler Komplexität.

Verteiltes Rechnen ermöglicht die Behandlung auch sehr großer Aufgaben.

Zusammenfassung

Interpolation: Wir ersetzen die Quellen des Felds durch „Ersatzquellen“ in Interpolationspunkten, die ungefähr dasselbe Feld erzeugen.

$$\sum_{y_j \in S} q_j g(x, y_j) \approx \sum_{\nu \in M} \hat{q}_{s,\nu} g(x, \xi_{s,\nu}).$$

Zulässigkeitskriterium $\text{diam}(s) \leq \eta \text{dist}(x_i, s)$ garantiert die Genauigkeit und führt zu einer Zerlegung des Raums in Teilgebiete.

Symmetrische Interpolation führt zu linearer, also optimaler Komplexität.

Verteiltes Rechnen ermöglicht die Behandlung auch sehr großer Aufgaben.

Verfeinerungen: Multipolentwicklung oder Quadraturformeln statt Interpolation, Interpolation variabler Ordnung, algebraische Rekompensation.

Literatur

Ursprünge:

- V. Rokhlin: Rapid solution of integral equations of classical potential theory
J. Comp. Phys. 60:187–207 (1985)
- W. Hackbusch, Z. P. Nowak: On the fast matrix multiplication in the boundary element method by panel clustering
Numer. Math. 54(4):463–491 (1989)

Verfeinerungen:

- L. Greengard, V. Rokhlin: A fast algorithm for particle simulations
J. Comp. Phys. 73:325–348 (1987)
- C. R. Anderson: An implementation of the fast multipole method without multipoles
SIAM J. Sci. Stat. Comp. 13:923–947 (1992)
- S. Börm, M. Löhndorf, J. M. Melenk: Approximation of integral operators by variable-order interpolation
Numer. Math. 99(4):605–643 (2005)

Literatur

Hybride Verfahren:

- S. Börm: Approximation of integral operators by $\mathcal{H}^2$ -matrices with adaptive bases
Computing 74:249–271 (2005)
- S. Börm, C. Börst: Hybrid matrix compression for high-frequency problems
SIAM J. Mat. Anal. Appl. 41(4):1704–1725 (2020)

Hierarchische Matrizen:

- W. Hackbusch: Hierarchical Matrices: Algorithms and Analysis
Springer (2015)
- S. Börm: Efficient Numerical Methods for Non-local Operators: $\mathcal{H}^2$ -Matrix
Compression, Algorithms and Analysis
EMS Tracts in Mathematics 14 (2010)
- M. Faustmann, J. M. Melenk, D. Praetorius: $\mathcal{H}$ -matrix approximability of the inverses
of FEM matrices
Numer. Math. 131(4):615–642 (2015)