

Performance-Optimierung und Grenzen eines Multi-Level Fast Multipole Algorithmus für akustische Berechnungen

Ralf Burgschweiger¹, Martin Ochmann¹, Ingo Schäfer², Bodo Nolte²

¹ Beuth Hochschule für Technik Berlin, 13353 Berlin, E-Mail:burgi@beuth-hochschule.de

² Forschungsbereich für Wasserschall und Geophysik (FWG) der WTD71, 24148 Kiel

Einleitung

Die Multi-Level Fast Multipole Methode (MLFMM) ermöglicht die numerische Berechnung akustischer Problemstellungen auf Basis der Randelementemethode (BEM), bei denen die diskretisierten Modelle aus sehr großen Anzahlen von Elementen bestehen.

Lösungszeit und Speicherbedarf liegen im Vergleich mit konventionellen Methoden in der Regel deutlich niedriger, da ein potentialbasierendes Clustering-Verfahren zur approximativen Berechnung der für iterative Löser benötigten Matrix-Vektor-Produkte verwendet wird.

Ein zuvor entwickelter Code, basierend auf einer Multi-Level/Single-Order-Variante des Algorithmus, wurde auf eine Multi-Level/Adaptive-Order-Version erweitert und hinsichtlich Lösungsqualität, Parallelisierbarkeit sowie Performance untersucht und optimiert.

Fast Multipole Methode (MLFMM)

Die Multi-Level Fast-Multipol-Methode stellt ein Verfahren zur schnellen Bildung eines Matrix-Vektor-Produktes (MVP) dar, ohne dabei jemals die Matrix vollständig bilden zu müssen. Sie eignet sich daher zum Einsatz bei „großen“ Gleichungssystemen, bei denen die Wechselwirkungen zwischen vielen Quell- (N_x) und Zielpunkten (N_y) berücksichtigt werden müssen. Abb. 1 verdeutlicht die Abnahme der Anzahl der so zu berücksichtigenden Interaktionen.

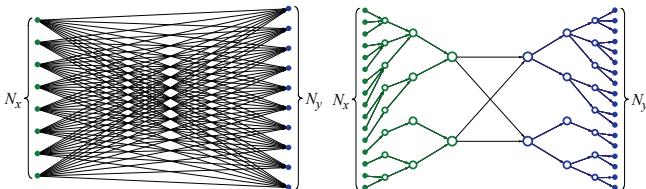


Abbildung 1: Interaktionen bei konventioneller Berechnung (links) und cluster-basiert bei der MLFMM (rechts)

Die Wirkungen einzelner Quellen x_i werden innerhalb eines „Quell“-Clusters C_x mit dem Radius R zu einer Multipol-Quelle am Punkt z_x zusammengefasst (Abb. 2).

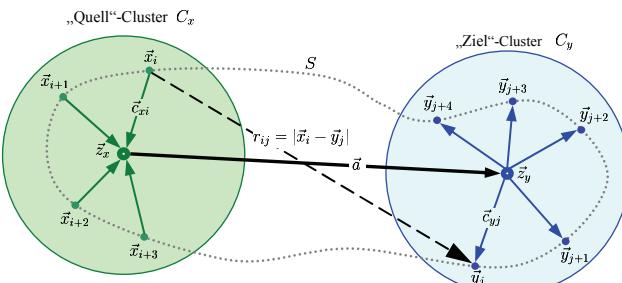


Abbildung 2: Zerlegung des Weges zwischen Quell- und Zielpunkten

Deren Potential wird zu einem „Ziel“-Cluster C_y mit dem Mittelpunkt z_y transformiert und dort auf die Zielpunkte y_j verteilt. Für jeden Teil des Weges zwischen dem Quell- und Zielpunkt wird eine entsprechende Übertragungsfunktion (h , μ^M und f) verwendet (Abb. 3), die vollständigen Details hierzu sind z. B. bei [4] zu finden.

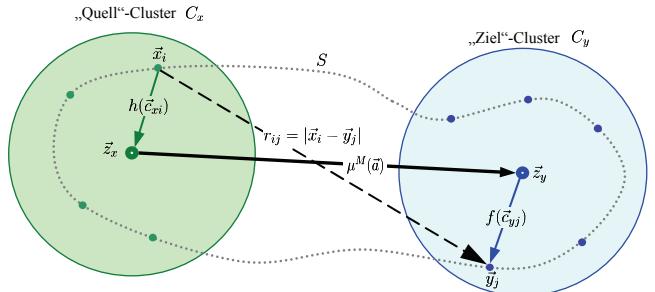


Abbildung 3: FMM-Übertragungsfunktionen h , μ^M und f

Eine der kritischsten Funktionen hierbei ist der Translations-Operator μ^M , der das Multipol-Potential (die sog. Signaturen) zwischen den Clusterzentren überträgt. Er kann als Reihenentwicklung mit einer maximalen Entwicklungsordnung O_{mp} („Multipol-Ordnung“) dargestellt werden:

$$\mu^M(\vec{a}, \hat{s}) = \frac{1}{4\pi} \sum_{l=1}^{O_{mp}} (2l+1) i^l h_l(k|\vec{a}|) P_l(\hat{s} \cdot \hat{a}) \quad (1)$$

mit \vec{a} Abstandsvektor zwischen den Cluster-Zentren
 \hat{s} Menge der Vektoren auf der Einheitskugel
 k Wellenzahl
 $h_l(k|\vec{a}|)$ Hankel-Funktion
 $P_l(\hat{s} \cdot \hat{a})$ Legendre-Polynom

Viele Artikel beschäftigen sich mit den Details zur Bestimmung dieses Operators [5, 6], speziell zur Hankel-Funktion und den Legendre-Polynomen, dies im Detail zu erläutern überschreitet jedoch den Rahmen dieses Beitrags.

Ergebnisse

In Vorprojekten wurde zunächst eine Version des MLFMM-Algorithmus unter Verwendung einer fest vorgegebenen Ordnung [1, 2, 4] implementiert (Vs. 1.x), um die Grenzen hinsichtlich der Elementanzahlen und Frequenzobergrenze ermitteln zu können.

Dieser Algorithmus wurde zu einer adaptiven Version (Vs. 2.x) auf Grundlage von [6, 7] ausgebaut, in dem die Multipol-Ordnung als Funktion von Cluster-Abstand und Wellenzahl definiert und zur Laufzeit berechnet wird.

Zusätzlich wurden weitere Optimierungen zur Verkürzung der Lösungszeit vorgenommen, vor allem bei der Vorausberechnung und der Parallelisierung einzelner Schritte.

Beispiel I: schallharte Kugel

Für dieses Beispiel wurde eine schallharte Kugel ($r = 0,5 \text{ m}$) in Wasser bei einer Frequenz $f = 1 \text{ kHz}$ verwendet.

Tabelle 1: Ergebnisse zur schallharten Kugel mit MLFMM (Vs. 1.030) bzw. adaptiver MLFMM (Vs. 2.030)

N_{elem}	feste Ordnung, $O_{mp} = 6$		adaptive Ordnung $O_{mp} = 6 \dots 10$			
	Δt_{solve}	N_{iter}	Δt_{solve}	N_{iter}	Δt_{MVP1}	Δt_{MVP}
1 k	0,64 s	13	0,29 s	13	0,07 s	0,02 s
2,5 k	0,62 s	12	0,42 s	13	0,12 s	0,03 s
5 k	2,06 s	13	0,91 s	12	0,32 s	0,05 s
10 k	2,59 s	12	1,33 s	12	0,56 s	0,07 s
20 k	7,63 s	13	2,57 s	12	1,12 s	0,13 s
50 k	15,13 s	12	7,38 s	12	4,04 s	0,29 s
100 k	31,87 s	12	12,79 s	12	6,56 s	0,54 s
200 k	57,16 s	12	25,88 s	12	13,63 s	1,08 s
500 k	138,26 s	12	63,74 s	12	33,05 s	2,66 s
1 M	373,61 s	12	134,61 s	12	71,36 s	5,50 s
2 M	561,82 s	12	257,12 s	12	140,23 s	10,10 s
5 M	1.922,26 s	13	856,73 s	12	532,74 s	25,99 s

Die Reduzierung der Lösungszeit gegenüber der "alten" Version (Details hierzu siehe [1]) ist deutlich zu erkennen.

Beispiel II: Ellipsoid

Dieses Beispiel zeigt die Abhängigkeit der Lösungsqualität von der gewählten Multipol-Ordnung (Abb. 4).

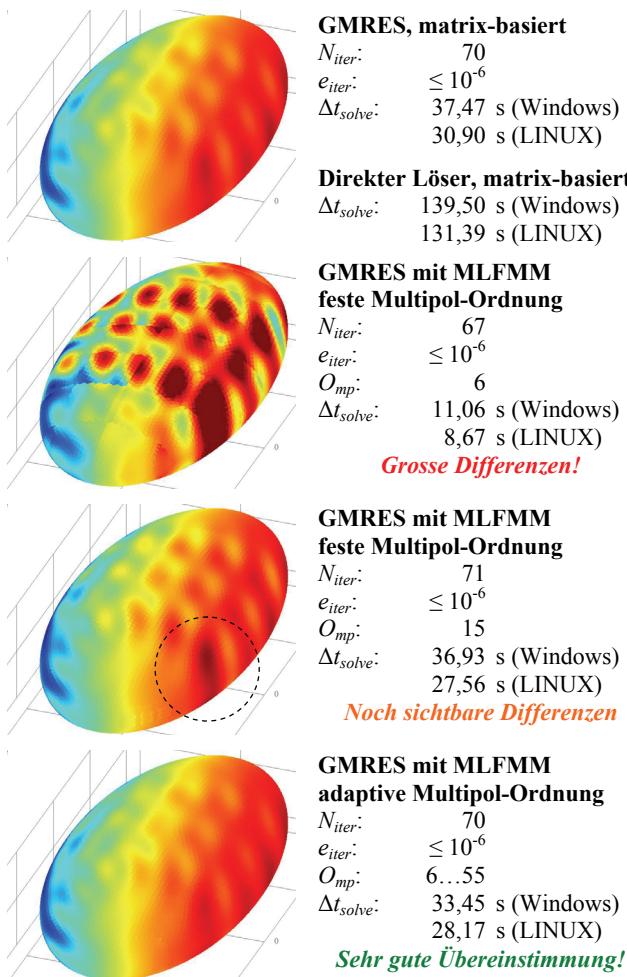


Abbildung 4: Ellipsoid ($4 \times 2 \times 1 \text{ m}$), $Re(p_{surf})$ für verschiedene Methoden und Multipol-Ordnungen, $f = 2,5 \text{ kHz}$

Es ist deutlich zu erkennen, dass bei unzureichender Ordnung trotz des Erreichens eines vorgegebenen Fehlers das Resultat aufgrund der Fehler in der approximativen Matrix-Vektor-Produktbildung nicht mit der erwarteten Lösung übereinstimmt. Erst die adaptive Version des MLFMM-Algorithmus liefert eine gute Übereinstimmung.

Zusammenfassung und Ausblick

Die vorgestellten Ergebnisse zeigen, dass die adaptive und performance-optimierte Version des MLFMM-Algorithmus eine bessere Qualität und schnellere Lösungen liefert.

Bei höheren Frequenzen zeigt die MLFMM erhebliche qualitative Unterschiede, da sich der Fehler in der methodenbasierten Matrix-Vektor-Produktbildung verstärkt auswirkt. Hier sind weitere Untersuchungen zur Optimierung des Codes für höhere Multipol-Ordnungen notwendig.

Es sollen geeignete Vorkonditionierungsverfahren getestet verwendet, um eine bessere Konvergenz des iterativen Verfahrens zu erreichen. Erste Ergebnisse wurden in [3] veröffentlicht, aber diese waren nicht wirklich erfolgversprechend, so dass hier weitere Arbeiten geplant sind.

Erste aktuelle Ergebnisse der Kombination des adaptiven MLFMM-Algorithmus mit der Burton-Miller-Methode zeigen aufgrund der besseren Kondition der Systemmatrix eine deutliche Reduktion der Anzahl an Iterationen, sodass hier ein weiterer Schwerpunkt zu setzen ist.

Literatur

- [1] R. Burgschweiger, I. Schäfer und M. Ochmann: „Ein Multilevel-Fast-Multipole-Algorithmus (MLFMM) zur Berechnung der Schallstreuung am Beispiel von Objekten unter Wasser“, DAGA 2010, Düsseldorf
- [2] R. Burgschweiger und M. Ochmann: “A Multi-Level Fast Multipole BEM-Method for computing the sound field in rooms”, Proceedings of the Forum Acusticum, 2011, Aalborg, Dänemark
- [3] M. Ochmann, R. Burgschweiger and C. Steuck: “Numerical experiments for testing the convergence of the acoustical Fast Multipole Method”, Proceedings of the 1st EAA Congress on Sound & Vibration (EuroRegio), 2010, Ljubljana
- [4] T. Sakuma, S. Schneider and Y. Yasuda: “Fast Solution Methods”, chapter 12, in ”Computational Acoustics of Noise Propagation in Fluids”, 2008, Springer Verlag
- [5] T. Sakuma and Y. Yasuda: “Fast Multipole Boundary Element Method for Large-Scale Steady-State Sound Field Analysis, Part I: Setup and Validation”, Acta Acustica united with Acustica, 2002, Vol. 88
- [6] Y. Yasuda and T. Sakuma: “Fast Multipole Boundary Element Method for Large-Scale Steady-State Sound Field Analysis, Part II: Examination of Numerical Items”, Acta Acustica, 2003, Vol. 89
- [7] H. Cheng et.al.: “A wideband fast multipole method for the Helmholtz equation in three dimensions”, Journal of Computational Physics, Vol. 216, 2006