

Ein Multilevel-Fast-Multipole-Algorithmus (MLFMM) zur Berechnung der Schallstreuung am Beispiel von Objekten unter Wasser

R. Burgschweiger¹, I. Schäfer², M. Ochmann¹

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

² Forschungsbereich der Bundeswehr für Wasserschall und Geophysik, 24148 Kiel

Einleitung

Im Rahmen des Forschungsprojektes „Numerische Methoden zur Detektion von Objekten“ wurde zur Beschleunigung der Matrix-Vektor-Produktbildung für BEM-basierende Berechnungen ein auf der Multi-Level-Fast Multipole-Method (MLFMM) basierender Algorithmus entwickelt.

Die implementierte Version des von Greengard und Rokhlin vorgestellten Algorithmus ermöglicht die Berechnung hochkomplexer ungekoppelter Probleme in Verbindung mit iterativen Lösungsverfahren (hier: GMRES).

Grundlagen der Multi-Level-Fast-Multipole Method (MLFMM)

Das Verfahren der sog. Fast-Multipole-Method zur Lösung von Randelementproblemen (BEM) basiert auf der 1987 von Greengard und Rokhlin vorgestellten Methode eines schnellen Algorithmus zur Partikelsimulation ([1]). Die 1999 durch Cheng, Greengard und Rokhlin vorgestellte adaptive Variante des Algorithmus für dreidimensionale Problemstellungen ([2]) bildet die Grundlage des implementierten Codes.

Das Verfahren basiert auf der Summierung der Wirkungen benachbarter Quellpunkte x_i innerhalb eines Clusters in einem Punkt z_x (Multipol), die Translation dessen Potentials an einen entfernten Punkt z_y und die Verteilung der dortigen Wirkung auf die cluster-eigenen Zielpunkte y_j .

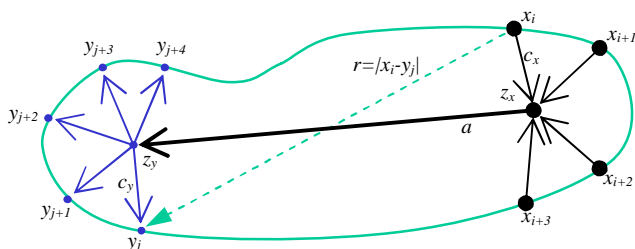


Abb. 1: Aufteilung des Pfades $y_j - x_i$ zwischen Quell- und Zielpunkten (Quelle: [4])

Das Ziel besteht darin, die Anzahl der zur Auswertung der Interaktionen zwischen den Quellpunkten x_i und den Zielpunkten y_j benötigten Rechenschritte soweit wie möglich zu verringern, die bei direkter Lösung N^2 betragen würde.

Die Zerlegung des Weges $y_j - x_i$ in 3 Abschnitte mit

$$y_j - x_i = (y_j - z_y) + (z_y - z_x) + (z_x - x_i)$$

und eine damit verbundene Bildung von sog. Clustern oder Boxen in Abhängigkeit von den betrachteten Strukturen bildet die geometrische Grundlage für die nachfolgenden Berechnungen.

Ablauf des Verfahrens

- **Partitioning** (Boxen-Generierung)
Berechnung der (Cluster-) Größen, Dimensionierung und Aufbau des 3D-Octree und Erstellung der Interaktionslisten (einmalig)
- **Berechnung der direkten Koeffizienten**
gemäß den Interaktionslisten (einmalig)
- **Source to Multipole Expansion**
Bildung der Multipole in den Mittelpunkten aller kinderlosen Boxen durch Berechnung der Fernfeldsignaturen aus den zugehörigen Quellen
- **Multipole to Multipole Translation**
Generierung der Multipole aller Elter-Boxen durch Translation aus den Fernfeldsignaturen ihrer jeweiligen Kinder-Boxen (rekursiv mit fallendem Level)
- **Multipole to Local Translation**
Erzeugung der Nahfeldsignaturen der Boxen unter Berücksichtigung der jeweiligen Interaktionsliste L2 (rekursiv mit steigendem Level, s. Abb. 2)
- **Local to Local Translation**
Verteilung der Nahfeldsignatur einer Box auf ihre Kinder (rekursiv mit steigendem Level)
- **Final Summation**
Bildung der Werte in den Zielpunkten durch Addition der Anteile des Nahfelds (L1), der Box-eigenen Nahfeldsignatur und aus der Interaktionsliste L3 (s. Abb. 2)

Partitioning (Aufbau der adaptiven Boxenstruktur)

Das Ziel der sog. Partitionierung, auch als Boxing bezeichnet, besteht in der Zuordnung der Quellpunkte zu den einzelnen Boxen, der Definition der Elter-Kind-Beziehungen und der Aufstellung der Nachbarschaftsbeziehungen in Form sog. Interaktionslisten (L1 bis L4) mit.

F	L_1	L_2	L_2	L_2	L_2
		L_1	L_1	L_2	L_2
F	L_2	L_1	B	L_1	L_4
	L_2	L_1	L_1	L_1	
			L_3	L_3	L_3
F	L_2	L_2	L_2	L_2	L_4
	L_2	L_2	L_2	L_2	
F	F	F	F	F	

Abb. 2: Interaktionsumfeld einer Box bei einer 2D-Partitionierung, Quelle: [2], S. 483, Abb. 4

- L1:** Liste aller unmittelbaren Nachbarn von **B**
- L2:** Liste aller Kinder der Nachbarn des eigenen Elters, die nicht unmittelbare an **B** grenzen (gleiche Größe wie **B**)
- L3:** Liste aller Kinder von **B**'s Nachbarn, die nicht an **B** angrenzen, aber deren Elter an **B** angrenzt
- L4:** enthält alle Boxen, in deren L3-Liste die Box **B** selbst enthalten ist (diese sind kinderlos und größer als **B**)
- F:** Fernfeld

Für den dreidimensionalen Fall ist der Aufbau komplexer, da hier die Boxen durch Würfel repräsentiert werden, wobei eine solche bis zu 27 unmittelbare Nachbarn und mit bis zu 189 Boxen interagieren kann. Der Preprozessor ermöglicht eine Visualisierung zu Kontrollzwecken:

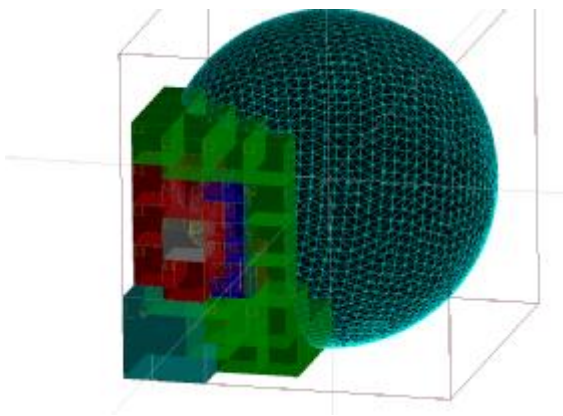


Abb. 3: Screenshot: Interaktionsumfeld einer Box in 3D

Ablauf im Calculator

Innerhalb des Calculators erfolgen die weitestgehend parallelisierten Berechnungsschritte, deren Umfang und Dauer sich im Konsolenmodus verfolgen lassen:

```

Quellpunkte ermitteln und normieren...
Ermittelte Quellpunkte (100.316) sortieren...
Boxen erstellen...
Größe der Zuweisungslisten für 10.000 Boxen berechnen...
Zuweisungslisten bereitzustellen...
Zuweisungsorten in die Boxen übertragen...
Zuweisungslisten in die Boxen übertragen...
Zuweisungslisten für L4 erstellen...

Generierung der Boxen und der Zuweisungslisten abgeschlossen.
Zeiten:
Box-IDs bestimmen (100.316 Punkte): 0,047 s
Zuweisungsliste sortieren: 0,015 s
Baumstruktur erstellen (MaxPoints = 20, MaxLevel = 6, BoxCount = 10000): 0,000 s

Boxenspezifische Listen erzeugen (N1+N2+N3+N4+NP = 1099052): 0,967 s
Insgesamt: 1,029 s
Using 8 threads for Matrix-Setup and/or MUP calculation

calculating Hankel-Values by distance      0,374 s, t = 104761
direct precalculation of coeffs           2,527 s, t = 13153619
1/6: UP1, S2M Multipole Expansion         0,468 s, t = 14876
2/6: UP2, M2M Multipole to Multipole      0,016 s, t = 3924
3/6: DP1, AddLPotentials                  0,031 s, t = 18888
4/6: DP2, AddSRForI2                      0,733 s, t = 18880
5/6: DP3, AddPotentialToChilds            0,141 s, t = 3932
6/6: BuildPotentials                       0,951 s, t = 14876
Expansions: 199599
Translations: 1214674
Hankel-Array: 104761

```

Abb. 4: Screenshot: MLFMM-spezifische Parameter während der Bildung eines Matrix-Vektor-Produkts (Kugel mit ca. 100.000 Elementen)

Gute Java-Beispiele zur Visualisierung des Ablaufs der verschiedenen Varianten des FMM-Algorithmus finden sich in der Master-Thesis von Y. Wang (siehe [3]).

Ergebnisse (schallharte Kugel)

Die Tabelle zeigt die Rechenzeiten für eine schallharte Kugel in Wasser bei unterschiedlicher Diskretisierung unter Verwendung eines iterativen GMRES-Lösers ohne Vorkonditionierung ($r=0,5$ m, Fehler $e_{max} = 10^{-10}$, max. 20 Punkte/Box, Ordnung Multipol-Expansion = 6, Ordnung Einheitskugel-Integration = 7)

Anz. Elemente	RAM-Bedarf Matrix	Direkter Löser (Intel MKL)	Iterativ (GMRES)	Anz. Iter.	Iterativ mit MLFMM	Anz. Iter.
1.000	16 MB	0,359 s	0,33 s	11	0,86 s	12
2.500	90 MB	1,578 s	0,99 s	10	1,11 s	11
5.000	0,6 GB	8,578 s	3,95 s	10	3,31 s	11
10.000	1,5 GB	50,204 s	15,06 s	10	3,86 s	11
20.000	6,1 GB	341,297 s	72,69 s	10	10,39 s	12
50.000	38 GB	4.689,406 s	459,80 s	10	20,64 s	11
100.000	153 GB	n.v.	13.346,31 s	10	43,91 s	11
200.000	612 GB	n.v.	n.v.	n.v.	79,30 s	11
500.000	3.858 GB	n.v.	n.v.	n.v.	195,90 s	11
1.000.000	15.438 GB	n.v.	n.v.	n.v.	512,08 s	11
2.000.000	61.157 GB	n.v.	n.v.	n.v.	1.003,67 s	11

Ergebnisse / Optimierung

Die erzielten Ergebnisse sind zwar beeindruckend, Untersuchungen zeigen jedoch, dass die Iterationsanzahl mit steigender Frequenz stark zunimmt bzw. sich die Konvergenz verschlechtert. Dies tritt insbesondere bei den sog. kritischen Frequenzen, d. h. den Resonanzfrequenzen des äquivalenten Innenraumproblems auf.

Bei iterativen Lösungsverfahren ist diese Problematik grundsätzlicher Natur, da hier die Konvergenz des Löser erheblich von der Konditionierung der Matrix abhängt. Durch die Berücksichtigung von CHIEF-Punkten konnten die erzielten Ergebnisse sowohl mit dem direkten als auch dem iterativen Löser deutlich verbessert werden.

Zusammenfassung

Es wurde schnelles parallelisiertes BEM-Verfahren zur Bildung des Matrix-Vektor-Produkts für ungekoppelte Rechnungen auf Basis der MLFMM implementiert und mit iterativen Lösern getestet. Die Ergebnisse zeigen, dass sich sehr gute Ergebnisse bei feiner Diskretisierung bzw. großer Anzahl an Elementen erzielen lassen, wenn die Lösung des Problems gut konvergiert. Bei höheren Frequenzen und komplexen Strukturen (z.B. mit Kanten) steigt die Anzahl der Iterationen deutlich an, so dass hier zusätzliche Maßnahmen ergriffen werden sollten.

Ausblick / weitere Tätigkeiten

- Integration geeigneter Vorkonditionierungsverfahren
- Erweiterung auf beliebige Randbedingungen
- Erweiterung für gekoppelte Rechnungen

Literaturverweise

[1] Greengard, L., Rokhlin, V.: "A fast algorithm for particle simulations", JCP, 1987, Vol., 73, S. 325 – 348

[2] Cheng, H., Greengard, L., and Rokhlin, V.,: "A Fast Adaptive Multipole Algorithm in Three Dimensions", JCP, 1999, Vol., 155, S. 468 – 498

[3] Wang, Y., „The Fast Multipole Method for 2D Coulombic Problems“, M.Sc. Thesis, Univ. of Maryland, 2005, S. 6-35, URL: <http://www.umiacs.umd.edu/~wpwy/fmm>

[4] Sakuma, T., Schneider, S. und Yasuda, Y., in „Computational Acoustics of Noise Propagation in Fluids“, herausgegeben von S. Marburg und B. Nolte, Kap. 12, „Fast Solution Methods“, 2008, Springer Verlag Berlin, S. 333ff