Proceedings of 20th International Congress on Acoustics, ICA 2010

23-27 August 2010, Sydney, Australia

A Multi-Level Fast Multipole Algorithm (MLFMM) for calculating the Sound scattered from Objects within Fluids

Ralf Burgschweiger (1), Ingo Schäfer (2) and Martin Ochmann (1)

(1) Beuth Hochschule für Technik Berlin (BHT), University of Applied Sciences,
Luxemburger Str. 10, D-13353 Berlin, Germany
(2) Wehrtechnische Dienststelle für Schiffe und Marinewaffen, Maritime Technologie und Forschung (WTD71/FWG) ,
Klausdorfer Weg 2-24, D-24128 Kiel, Germany

PACS: 43.20.Fn, 43.30.Gv

ABSTRACT

The Multi-Level Fast Multipole Method (MLFMM) allows the calculation of the sound scattering from objects with a very high level of discretization. The required calculation time is much less when compared with conventional boundary element methods because the algorithm uses a level-based composition of the potentials from different point sources to acoustic multipoles, which highly accelerates the computation of the matrix-vector-products required.
The basic theory and the functionality of the implementation of this algorithm in a parallelized version will be described.
The results for direct and iterative BEM-based solution methods with and without use of the MLFMM algorithm will be shown using a scattering rigid body with different levels of discretization (consisting of more than one million elements).
The CPU time of the different methods will be compared and the current limits of the algorithm will be discussed.

INTRODUCTION

An algorithm to accelerate the matrix-vector-product generation for BEM-based calculations using the Multi-Level Fast Multipole Method (MLFMM) was developed in the context of the research project „Numerical methods for the detection of objects in the frequency domain“.

Our implementation of this algorithm that was first presented by Greengard and Rohklin allows at this time the calculation of complex uncoupled problems in conjunction with iterative solvers (here: GMRES). The intention was to build a high performance solver for large problems running parallelized on one multi-core system.

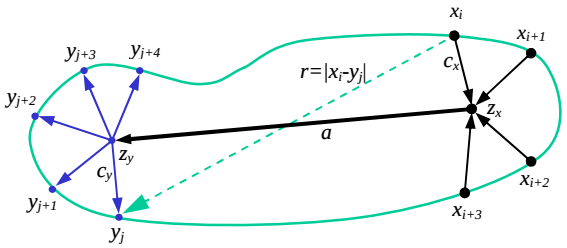
The basics of the method and the current results will be shown and discussed.

BASICS OF THE MULTI-LEVEL FAST MULTIPOLE METHOD (MLFMM)

The so called Fast Multipole Method for solving boundary element problems (BEM) is based on the article of Greengard and Rokhlin [1], which was presented in 1987.

The method was extended during the following years and the adaptive version of the algorithm for three-dimensional problems [2], presentend by Cheng, Greengard und Rokhlin in 1999, provided the basis of the code implemented.

The method uses a summation of the potentials of adjacent source points x_i within a cluster centered at a point z_x (multipole), the translation of its potential to a far point z_y and the distribution to the destination points y_j which are adjacent within the far cluster.



(source: [4], chap. 12, „Fast Solution Methods“, p. 333ff)

Fig. 1: The partitioning of the path between source (x_i) and destination points (y_j)

To guarantee the far field condition, it is required that the distance $a=|z_x-z_y| > |c_{x,max}| + |c_{y,max}|$ ($c_{x,max}$ and $c_{y,max}$ are the maximum distances between the cluster center and the source resp. destination point).

The intent is to minimize the required calculation steps in the evaluation of the interactions between the source points x_i and the destination (field) points y_j which would be of the order $O(N^2)$ using a direct solver.

The partitioning of all paths y_j-x_i into three parts with

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

by setting up clusters (or boxes) depending on the structure(s) of the problem is required to build the geometric base for the following calculations.

The main idea of the Fast Multipole Method consists in approximating a kernel function $k(x-y)$, which depends on the distance $r = |x_i - y_j|$, for all far points using three auxiliary functions g, μ und h (the complete details and formulations may be found in [4] and will not be shown here due to paper limits):

$$k(x-y) \approx g(y-z_y)\,\mu(z_y-z_x)\,h(z_x-x)$$

$$\approx g(c_y)\,\mu(a)\,h(c_x)$$

The number and the positions of source and destination points are equal for acoustic problems using the boundary element method, but that may be different for others kinds of physical problems.

MAIN STEPS OF THE ADAPTIVE ALGORITHM

- **Partitioning (generation of boxes)**

  Calculation of the (cluster) sizes depending on the structure(s), generation of the cluster tree and of all interaction lists (one-time)

- **Calculation of direct coefficients**

  Calculation of all near field coefficients as defined by the interaction lists using conventional collocation BEM, including the allocation of required memory (one-time)

- **Source to Multipole Expansion (S2M, S|M, Upward Pass Step 1**)

  Generation of multipoles in the center points of all childless boxes by calculating the far-field signatures of the associated source points.

- **Multipole to Multipole Translation (M2M, S|S, Upward Pass Step 2)**

  Generation of multipoles in the center points of all parent boxes by translating the far-field signatures from their children (recursive with descending level)

- **Multipole to Local Translation (M2L, S|R, Downward Pass Step 1)**

  Generation of the near-field signatures of all boxes with respect to their interaction list L2 (recursive with ascending level)

- **Local to Local Translation (L2L, R|R, Downward Pass Step 2)**

  Distribution of the near-field signature of all parent boxes to their children (recursive with ascending level in the center points)

- **Final Summation**

  Formation of the results at the destination points by adding the portions from the local neighbour source points, the box-specific near-field signature and the boxes which are within the interaction list L3.

PARTITIONING (GENERATION OF THE MULTI-LEVEL BOX-TREE)

The intention of the so-called partitioning (or boxing) is the assignment of source points to a (cluster) box, the definition of the parent-child relationships and the assembly of the neighbour relations by building the interaction lists L1 to L4 for each box.

Definition of the interaction lists of a box **B**:

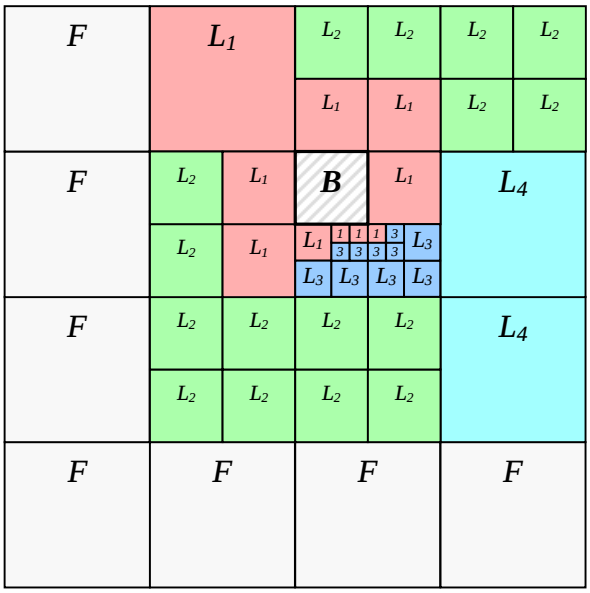
L1: contains all immediate neighbours of **B**

L2: contains all children of **B**'s parent's neighbours, which are not immedate neighbours of **B** (all boxes in L2 have the same size as **B**)

L3: contains all children of **B**'s neighbours, which are not immediate neighbours of **B**, but whose parent is a neighbour of **B**

L4: contains all boxes, which have **B** in their own L3 list (they are always childless and bigger than **B**)

F: far field (not in interaction list)



(source: [2], Cheng, H.,, Greengard, L., and Rokhlin, V.,: A Fast Adaptive Multipole Algorithm in Three Dimensions", p. 483, fig. 4)

Fig. 2: Interaction lists of one box, using a two-dimensional partitioning

The setup for a three-dimensional problem is much more complex due to the use of cubic boxes. One box may have a maximum of 27 immediate neighbours ("colleagues") and can contain a maximum of 189 entries in the interaction lists.

A binary-based transformation for all points is used to accelerate the setup time of the 3D octtree and to figure out the neighbourship relations. The preprocessor can visualize the resulting interaction field of each box for control reasons (as shown in fig. 3).

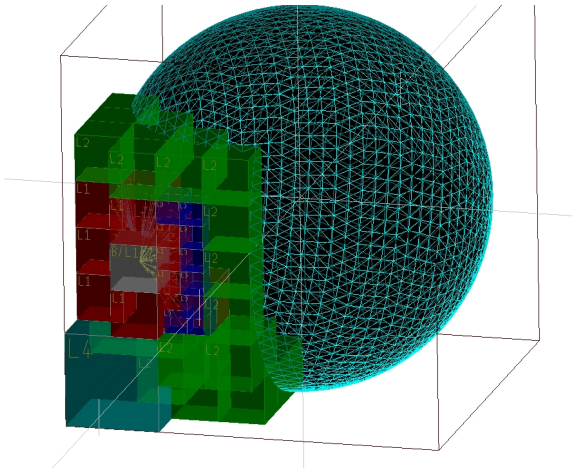


Fig. 3: Screenshot: interaction field of a 3D box

ACTIVITY OF THE CALCULATOR MODULE

The calculator module is a standalone application running on multi-core hardware. It implements different kinds of direct, iterative and approximative solving methods resp. solvers.

Here the module is responsible for the current calculation steps during the matrix-vector-product generation which are parallelized whenever possible.

Using a console mode, one can follow the sequence of steps, their runtime and the current iteration. The screenshot (fig. 4) shows an example for a sphere made up from about 100,000 elements.

```
Quellpunkte ermitteln und normieren...
Ermittelte Quellpunkte (100.316) sortieren...
Boxen erstellen...
Größe der Zuweisungslisten für 18.809 Boxen berechnen...
Zuweisungslisten bereitstellen...
Quellpunkt-Zuweisungen 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 = 18809): 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 MVP 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, AddL4Potentials                            0,031 s, t = 18808
4/6: DP2, AddSRForL2                                 0,733 s, t = 18800
5/6: DP3, AddPotentialToChilds                       0,141 s, t = 3932
6/6: BuildPotentials                                 0,951 s, t = 14876
Expansions:     199599
Translations:  1214674
Hankel-Array:   104761
```

Fig.4: Screenshot: MLFMM step specific values during the generation of the first matrix vector product

For further information one should take a look at the master thesis of Wang, Y. [3], which contains very descriptive samples to visualize three different variations of the Fast Multipole Method (non-adaptive and adaptive) and may be run in every browser with Java support.

RESULTS FOR A RIGID SPHERE

The following tables shows the required solving times for a rigid sphere in water using different discretizations using a direct solver (Intel Math Kernel Library, if possible) and a non-preconditioned GMRES-solver without and with the use of the MLFMM method.

Parameters

Radius	r	0.5 m
Density (water)	ρ	1500 kg/m³
Sound speed (water)	c	1000 m/s
Frequency	f	1 kHz
Residual	e_{res}	1×10^{-10}
Max. source points per box:	N_{box}	20
Order of multipole expansion	O_{mp}	6
Order of unit sphere integration	O_{us}	7

Table 1: Results of Vs. 1.020 (10/2009)

Ele-ments	Required RAM (matrix) [GB]	Direct solver (IMKL) [s]	Iterativ (GMRES) [s]	Iter.	Iterative with MLFMM [s]	Iter.
1 k	0.01	0.36	0.33	11	0.86	12
2.5 k	0.09	1.58	0.99	10	1.11	11
5 k	0.6	8.58	3.95	10	3.31	11
10 k	1,5	50.2	15.06	10	3.86	11
20 k	6,1	341.30	72.69	10	10.39	12
50 k	38	4,689.41	459.80	10	20.64	11
100 k	153	n.v.	*13,346.31	10	43.91	11
200 k	612	n.v.	n.v.		79.30	11
500 k	3,858	n.v.	n.v.		195.90	11
1 M	15,438	n.v.	n.v.		512.08	11
2 M	61,157	n.v.	n.v.		1.003.67	11

System: Dual Intel XEON E5430, 2.66 GHz, 8 cores,
64 GB DDR-2 (used memory: ≈ 13.5 GB for 2 M elements)
*: complete recalculation of the matrix for each iteration

After some optimizations (parallelization of additional steps, memory allocation) and using new hardware we achieved some better results and raised the maximum element number.

Table 2: Results of Vs. 1.030 (05/2010)

Ele-ments	Required RAM (matrix) [GB]	Direct solver (IMKL) [s]	Iterativ (GMRES) [s]	Iter.	Iterative with MLFMM [s]	Iter.
1 k	0.01	0.20	0.16	11	0.63	12
2.5 k	0.09	1.23	0.73	10	0.62	11
5 k	0.6	7.44	3.03	10	2.06	12
10 k	1,5	43,99	11.92	10	2.59	11
20 k	6,1	310.75	52.56	10	7.63	12
50 k	38	4,352.43	329.36	10	15.13	11
100 k	153	n.v.	*11,719,95	10	31.87	11
200 k	612	n.v.	*46,635.42	10	57.16	11
500 k	3,858	n.v.	n.v.		138.26	11
1 M	15,438	n.v.	n.v.		373.61	11
2 M	61,157	n.v.	n.v.		561.82	11
5 M	382,771	n.v.	n.v.		1,922.26	12

System: Dual Intel XEON E5550, 2.66 GHz, 8 cores,
96 GB DDR-3 (used memory: ≈ 32.7 GB for 5 M elements)
*: complete recalculation of the matrix for each iteration

It seems that the solving time using the MLFMM will raise approx. linear with respect to the number of source points (elements), but this is limited to this special case with a very low number of iterations.

ADDITIONAL OPTIMIZATIONS

The results shown above are encouraging but additional tests (frequency sweeps) have shown that the number of iterations is raising significantly when the frequency increases or the convergence of the iterative solver decreases.

This is a well known problem when using iterative solvers because the quality and the convergence of the solution depend on the condition of the matrix.

The quality of the solution using the MLFMM also gets worse if a frequency near a resonance frequency of the equivalent inner space problem is chosen.

An option for generating and using CHIEF points [5] was added, tested and compared with analytical solutions. The improved quality of the results using this option is shown in fig. 5, including the MLFMM.

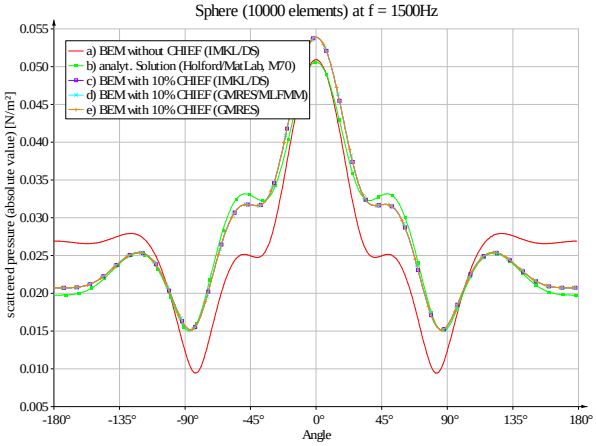


Fig.5: results for a rigid sphere in water at f = 1500 Hz (inner space resonance frequency)

CONCLUSION

A fast parallelized method for building the matrix-vector-product of uncoupled calculations bases on the MLFMM was implemented and tested for calculating the acoustic field scattered from a rigid sphere.

It was demonstrated that the performance and the quality of the results achieved for a high discretization is very good if the iterative solver used converges well.

The choice of higher frequencies or more complex structures (i.e. with sharp borders) leads to high numbers of iterations, so that additional techniques should be used.

OUTLOOK / FUTURE WORK

- Integration of adequate preconditioning methods into the implemented iterative solvers
- Extension of the MLFMM algorithm to variable boundary conditions
- Extension of the MLFMM algorithm to coupled problems

REFERENCES / LINKS

[1] Greengard, L., Rokhlin, V., “A fast algorithm for particle simulations”, *Journal of Computational Physics*, 1987, Vol. 73, pp. 325 – 348

[2] Cheng, H., Greengard, L., and Rokhlin, V., “A Fast Adaptive Multipole Algorithm in Three Dimensions”, *Journal of Computational Physics*, 1999, Vol. 155, pp. 468 – 498

[3] Wang, Y., ”The Fast Multipole Method for 2D Coulombic Problems: Analysis, Implementation and Visualization”, *Master of Science Thesis*, University of Maryland, 2005, pp. 6-35
(download at http://www.umiacs.umd.edu/~wpwy/fmm)

[4] Sakuma, T., Schneider, S. und Yasuda, Y., ”Computational Acoustics of Noise Propagation in Fluids”, chap. 12 in *Fast Solution Methods*, (2008, Springer Verlag, ISBN 978-3-540-77447-1) pp. 333ff

[5] Schenck, H. A., ”Improved Integral Formulation for Acoustic Radiation Problems”, *Journal of the Acoustical Society of America*, 1968, Vol. 44, No. 1, pp. 41ff