HP-FASS: A Hybrid Parallel Fast Acoustic Scattering Solver. CMMSE-09

Jesús A. López Fernández, Miguel López Portugués, José Manuel Taboada Varela, Henry J. Rice, Fernando Obelleiro

► To cite this version:

Jesús A. López Fernández, Miguel López Portugués, José Manuel Taboada Varela, Henry J. Rice, Fernando Obelleiro. HP-FASS: A Hybrid Parallel Fast Acoustic Scattering Solver. CMMSE-09. International Journal of Computer Mathematics, 2011, pp.1. 10.1080/00207160.2010.521239 . hal-00677962

HAL Id: hal-00677962

<https://hal.science/hal-00677962>

Submitted on 11 Mar 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**HP-FASS: A Hybrid Parallel Fast Acoustic Scattering Solver.
CMMSE-09**

Journal:	<i>International Journal of Computer Mathematics</i>
Manuscript ID:	GCOM-2009-0783-B.R2
Manuscript Type:	Original Article
Date Submitted by the Author:	29-Jun-2010
Complete List of Authors:	López Fernández, Jesús; Universidad de Oviedo, Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas López Portugués, Miguel; Universidad de Oviedo, Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas Taboada Varela, José Manuel; Universidad de Extremadura, Departamento de Informática Rice, Henry; Trinity College Dublin, Department of Mechanical and Manufacturing Engineering Obelleiro, Fernando; Universidade de Vigo, Departamento de Teoría do Sinal e Comunicaci3ns
Keywords:	Acoustic scattering, Boundary Elements Method, Fast Multipole Method, MPI, OpenMP

SCHOLARONE™
Manuscripts

RESEARCH ARTICLE

HP-FASS: A Hybrid Parallel Fast Acoustic Scattering Solver. CMMSE-09

J. A. López-Fernández^a, M. L. Portugués^{a*}, J. M. Taboada^b, H. J. Rice^c and F.
 Obelleiro^d

^a*Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y de Sistemas,
 Universidad de Oviedo, Edificio Polivalente de Viesques, 33204 Gijón, Spain;*

^b*Departamento de Tecnología de los Computadores y de las Comunicaciones, Escuela
 Politécnica, Universidad de Extremadura, 10071 Cáceres, Spain;* ^c*Department of
 Mechanical and Manufacturing Engineering, Trinity College Dublin, Ireland;*

^d*Departamento de Teoría de Sinal e Comunicacóns, ETSE Telecomunicación, Campus
 Universitario s/n, 36200 Vigo, Spain*

(Received 00 Month 200x; in final form 00 Month 200x)

Acoustic scattering problems may be accurately posed in terms of the *Boundary Elements Method* (BEM). Nevertheless, the solution of the linear system of equations produced by the BEM usually requires huge computational resources which limits its applicability. The use of efficient solvers in combination with parallel programming may considerably increase the scope of application of the BEM. In this work, a hybrid parallel implementation of the *Fast Multipole Method* (FMM) using a hybrid domain decomposition is presented. In addition, relying on the use of oct-tree structures, a technique to select the group size that minimizes the execution time is suggested. As a result, a highly scalable acoustic scattering solver is produced which may be used, for instance, in aircraft noise control applications.

Keywords: Acoustic scattering; Boundary Elements Method; Fast Multipole Method; MPI; OpenMP

AMS Subject Classification: 45-04; 45J05

1. Introduction

Low-frequency methods, such as the *Boundary Elements Method* (BEM) [16], offer the potential of precise numerical formulations of acoustic scattering problems. The use of the BEM on this type of problems yields a linear system of N equations with N unknowns. The computational cost associated to the direct solution is $\mathcal{O}(N^3)$ in terms of time and $\mathcal{O}(N^2)$ in terms of memory, meanwhile, iterative solvers have an $\mathcal{O}(N^2)$ time cost per iteration. Hence, if the number of iterations to achieve certain residual error is smaller than N , which is usually the case, a reduction in the execution time is achieved. Among the different iterative solvers applied to acoustic scattering, it is worth mentioning the *Generalized Minimum Residual* (GMRES) method [13] which is a very robust solver for this type of problems [8].

*Corresponding author. Email: mlopez@tsc.uniovi.es

The complexity, $\mathcal{O}(N^2)$ per iteration, of iterative solvers is due to the computation of a *Matrix-Vector Product* (MVP). The *Fast Multipole Method* (FMM) [12] and its multilevel version, known as *Multilevel Fast Multipole Algorithm* (MLFMA) [5, 14], avoid matrix explicit calculation yielding a dramatic reduction in the MVP time without significantly affecting to BEM's iterative solution accuracy. The FMM and the MLFMA reduce the cost per iteration to $\mathcal{O}(N^{1.5})$ and to $\mathcal{O}(N \log(N))$, respectively.

For the last decade, the interest in using parallel computing to decrease solution times and to enlarge the scope of application of BEM solvers has been growing. On the one hand, the parallel implementation of a direct solver (based on LU factorisation) presented in [4] achieved quite good *scalability*, although, its application to big size problems was limited by the high computational cost of the solver. On the other hand, the different approaches reported of the parallel MLFMA achieve poor scalability results due to the interdependence of the data required by the different hierarchy levels [15].

In this work, a hybrid parallel scattering solver tool is developed. The GMRES is chosen, as iterative solver, due to its robustness for solving acoustic scattering problems [8]. **No preconditioners have been used in the iterative solution.** In addition, the FMM is employed as accelerator for the MVP that each GMRES iteration requires.

This paper is organised as follows. In section §2, the statement of the acoustic scattering problem is posed. Section §3 is a brief summary of the FMM. Section §4 describes the hybrid parallel implementation of the FMM developed in this work. Section §5 is a benchmarking study of the proposed implementation. Finally, some conclusions are discussed in section §6.

2. Statement of the Physical Problem

The physical problem studied here consists in predicting the acoustic pressure on the space that surrounds a 3-D obstacle on which an incident acoustic wave is impinging. This problem may be posed in terms of the integral form of the Helmholtz equation [16] which is also known as *Conventional Boundary Integral Equation* (CBIE). Thus, the solution of the scattering problem may be carried out into two steps. First, the CBIE applied to the surface S of the obstacle is solved. This yields the distribution of the pressure over S . Second, the solution provided by the first step is used to obtain the scattered pressure by means of the application of the CBIE to the acoustic domain. **This work is focused on the first step for solving the scattering problem**, that is usually the one that demands more computational resources. In order to overcome the *non-uniqueness difficulty* [2] that appears in the CBIE at resonant frequencies, the CBIE is linearly combined with its normal derivative yielding the Burton and Miller equation [2]. By means of the BEM, the Burton and Miller equation may be discretised over S and formulated in terms of a linear system of equations in the form:

$$\mathbf{K} \mathbf{p} = \mathbf{g}, \quad (1)$$

where $\mathbf{K}$ is the coupling or system matrix, $\mathbf{p}$ is the coefficients vector, and $\mathbf{g}$ is the excitation vector which is related to the incident pressure.

In the sequel, the surface of the obstacle is considered to be discretised into N elements which means that the linear system in (1) has N unknowns. In order to produce accurate results, S must be discretised into 6 to 10 elements per linear wavelength (λ). Since the wavelength and the frequency f are in inverse proportion,

N increases, for a given obstacle, with the frequency squared (f^2).

3. Accelerated Iterative Solution

The solution of equation (1) may be achieved by means of an iterative method such as the GMRES [13] which is an extension of *Minimum Residual* (MINRES), designed to solve nonsymmetric linear systems. The GMRES is, among the Krylov solvers, the most efficient for solving acoustic scattering problems [8]. The most time consuming operation in Krylov solvers is the MVP to get the iterate and the residual error (stop condition). The GMRES computes the residual norm without the iterate having to be explicitly formed, which reduces to one the number of MVPs per iteration. Once the residual norm is deemed small enough, the iterate is formed.

The FMM computes a MVP without explicit calculation of the matrix. The algorithm requires grouping the N elements of the problem into N_g groups and then it efficiently calculates *far interactions*, between non-neighboring groups, by means of the *Addition Theorems* and *plane wave decomposition* [1]. *Near interactions*, between elements that pertain to the same group or to neighboring groups, do not satisfy the conditions associated with the transformations described by the Addition Theorems [12]. As a consequence, near (or local) interactions must be computed directly evaluating its corresponding part of the system matrix. The whole procedure may be summarised as follows:

- (1) *Setup*: the geometry is divided into groups of elements. This step is also known as spatial grouping and it pertains to the setup of the algorithm. Nonetheless, it is an important step because the way the geometry division is performed may yield important computational savings. The use of *oct-tree* structures in combination with *hierarchical numbering* [6] produces very efficient spatial ordering calculations needed in the FMM.
- (2) *Near interactions*: the radiation of the elements of each group over elements pertaining to itself and to nearby groups is evaluated directly, see Fig. 1.
- (3) *Far interactions*: the radiation of the elements of each group over elements pertaining to far groups is efficiently computed by means of the following operations:
 - a) *Aggregation*: the radiation produced by the elements of each group is represented by a multipole expansion placed at the group centre, see the schematic drawing in Fig. 1.
 - b) *Translation*: the above-mentioned multipole representation is translated from each source group centre to each observation group centre, see Fig. 1. Source group translated contributions are added together at the centre of each observation group. The translation operation is diagonalised which leads to the major computational saving of the method [12].
 - c) *Disaggregation*: the radiation produced by the observation group centre multipole at every observation element within the group, see Fig. 1, is computed.
- (4) *Near-Far interactions addition*: near and far interactions are added together yielding the MVP without explicit formation of the system matrix.

The number of multipoles (L) that are necessary to represent the field as an expansion at each group centre and the average number of near elements to each one (N_c) are proportional to the wavelength size of the group [12]. The computational cost savings produced when using FMM instead of direct MVP comes from the fact

Figure 1. Illustration of near interactions vs. far interactions computed by FMM.

that instead of having to compute the interaction between every pair of elements, $\mathcal{O}(N^2)$ operations, only the interactions between far groups, $\mathcal{O}(L^2 N_g^2)$ operations, and between close elements, $\mathcal{O}(N_g N_c^2)$ operations, are computed. It is possible to show that the optimum number of groups increases proportionally to $\sqrt{N}$ [12], yielding a cost $\mathcal{O}(N^{1.5})$. In this work, the efficient computation of elements-groups relations by means of oct-tree structures is exploited in order to select a group size that diminishes as possible the execution time while requiring an $\mathcal{O}(N)$ memory cost.

4. Parallel Implementation

The software presented in this work (HP-FASS, *Hybrid Parallel Fast Acoustic Scattering Solver*) has two main computational goals: *low memory footprint* and *high scalability*. To achieve these goals we developed a hybrid parallel implementation using OpenMP [11] threads and the *Message Passing Interface* (MPI) [9, 10].

To achieve a low memory footprint two different actions were taken. To begin with, data structures of size $\mathcal{O}(N^{1.5})$, typical of common FMM implementations, are completely avoided which diminishes the memory cost to $\mathcal{O}(N)$. Nonetheless, this memory saving strategy may increase the constant factor in the time cost. In addition, the parallel implementation has been carried out paying special attention to reduce as much as possible data replication among processes. First, the computation associated to the aggregation and disaggregation steps is distributed using groups, minimising the replication of the geometry data. Furthermore, the computation associated to the translation step is distributed using integration points, avoiding any replication of the translation operator.

High scalability is the second goal of the parallel implementation proposed in here. Since the group size is critical to minimise the execution time and improve the scalability, the following model is used to calculate the time cost of each MVP:

$$T_{MVP}(N, p) = T_{near}(N, p) + T_{aggregation}(N, p) + T_{translation}(N, p) + T_{disaggregation}(N, p) + T_{addition}(N, p), \quad (2)$$

where N is the number of unknowns and p is the number of processors (cores).

In order to obtain each term of the equation (2), a model that includes both computation and communication has been used. The computation time is calculated taking into account the number of operations associated to each step of the algorithm and the correspondent machine-dependent constant factors, the latter estimated using empirical data. The communications time is calculated taking into consideration the number and the size of the messages, together with the estimated network latency (α) and the estimated transfer time per byte (β).

The time related to the near interactions has been estimated by the following equation:

$$T_{near}(N, p) = K_{near} \frac{\sum_{i=1}^{N_g} n_i \cdot n_i^{near}}{p} + \alpha + \beta \frac{N}{p}, \quad (3)$$

where N_g is the number of groups, n_i is the number of elements in the i^{th} group, n_i^{near} is the number of elements in the neighbourhood of the i^{th} group and K_{near} is the near interactions constant factor.

The time related to the far interactions has been approximated by means of the following set of equations:

$$T_{aggregation}(N, p) = K_{agg} k_l \frac{N}{p} + (p-1) \left(\alpha + \beta \left\lceil \frac{k_l N_g}{p} \right\rceil \right), \quad (4a)$$

$$T_{translation}(N, p) = K_{trans} k_l \frac{\sum_{i=1}^{N_g} g_i^{far}}{p} + (p-1) \left(\alpha + \beta \left\lceil \frac{k_l N_g}{p} \right\rceil \right), \quad (4b)$$

$$T_{disaggregation}(N, p) = K_{dis} k_l \frac{N}{p} + \alpha + \beta \frac{N}{p}, \quad (4c)$$

where k_l is the number of integration points (related to the square of the number of multipoles), g_i^{far} is the number of distant groups to the i^{th} group, and $\lceil \cdot \rceil$ is the ceiling function. K_{agg} , K_{trans} , and K_{dis} are the constant factors for the aggregation, translation, and disaggregation steps, respectively.

Finally, the time taken by the addition of near and far interactions follows:

$$T_{addition}(N, p) = K_{add} N, \quad (5)$$

where K_{add} is the addition constant factor.

In the setup stage (at runtime), the algorithm exhaustively tests different group sizes and precisely calculates the time cost of the MVP for the resulting spatial groupings. In this manner, the optimum group size, in the sense of minimizing the time cost, is chosen for an arbitrary problem solved in any number of processors. This technique results in an *adaptive group size* as opposed to the most common fixed group size. It is worth mentioning that taking advantage of the oct-tree theory [6], the time cost of searching for the optimum group size is $\mathcal{O}(N)$.

A methodical procedure [3] has been followed to design and develop this software, taking into account *partitioning*, *load balancing*, and *communications*. In addition, HP-FASS has been developed using the *masteronly* programming model in which only one thread per process (master thread) communicates. Thus, MPI communications take place only outside of OpenMP parallel regions, avoiding possible

saturation of the inter-node network bandwidth. As a result, this programming model is well suited to hybrid systems (clusters of multi-core nodes).

A *domain decomposition* technique taken at two levels is used in this work. In aggregation, disaggregation, and near interactions, both coarse (MPI) and fine (OpenMP) partitions use the group as minimum computing entity. However, in the translation step a new approach is provided: fine partition is taken at group level whereas coarse partition is taken using integration points. This technique is scalable since both groups and integration points scale with $\sqrt{N}$. It is worth mentioning that there is not redundant computation, resulting in scalable codes, and there is minimum data replication, which yields a very low memory footprint.

Load balancing is accomplished at a very low cost thanks to the oct-tree based space subdivision in addition to a hierarchical numbering. As above-mentioned, an optimal geometry subdivision trying different group sizes is performed during the setup of the algorithm. Then, the computational cost associated to each group is computed at runtime taking advantage of the oct-tree structure. Finally, a simple algorithm is used to distribute consecutive groups and integration points among processes, achieving a well-balanced workload at a very low cost. There are also two different levels to deal with load balancing: the MPI coarse level, explained above, and the OpenMP fine level. Coarse level uses a static load-balancing scheme whereas fine level group assignment is done on demand thanks to OpenMP dynamic and guided schedulers. This dynamic scheme easily corrects runtime imbalances among threads inside a process.

The main interchanges of data, those performed at the end of aggregation and translation, are carried out by means of point-to-point *non-blocking* communications that take place between **neighbour processes**. Since the communication partners do not change over time, the processes are organised as a static ring structure, allowing an efficient interchange of the data computed by each process. In addition, the MPI collectives are used as little as possible with the aim of limiting the impact of global communications in the scalability. Global communications that are performed throughout the algorithm have a static tree structure, where one process is the root and the other ones are leafs. MPI_Bcast is used to distribute groups data during the setup step and to provide the residual error (stop condition) at the end of each iteration. MPI_Scatterv is employed to spread the geometry information during the setup just as to distribute, at the beginning of each iteration, the updated pressure vectors. Finally, MPI_Gatherv is used to collect computed data (far interactions, near interactions, and solution pressure vectors once the desired residual error is reached).

5. Results

The results shown in this work have been obtained using the CMI cluster, which is part of the *Clúster de Modelización Científica* (Universidad de Oviedo - Spain). CMI consists of 32 nodes (with Red Hat EL 4 Update 7) linked through a 20 Gbps InfiniBand network. Each node contains 2 AMD Opteron quad-core processors at 2.3 GHz and 32 GB of RAM, resulting in a total of 256 cores and 1 TB of RAM. Furthermore, we have used Intel MKL 10.0.011 for linear algebra operations, HP-MPI 2.3 middleware for communications over InfiniBand, and AMD/GCC 4.2 to compile our source code. It is also worth mentioning that **single precision complex arithmetic is used in our codes**.

The nodes in the CMI cluster have a *cache-coherent Non-Uniform Memory Access* architecture (ccNUMA), where each CPU socket (4 cores) has an integrated memory controller and can access the local memory faster than the remote mem-

ory. Therefore, each node may be viewed as two different nodes linked through a HyperTransport [7] link. We have taken this into consideration and we have carried out the parallel executions using 2 MPI processes per node and 4 OpenMP threads per process, taking into account the CPU binding to maximise memory bandwidth and to minimise latency.

Since the availability of the cluster for benchmarking is limited, we decided to analyse rather small problems. The geometry of the obstacle analysed corresponds to an aircraft model of an Airbus A300 series used in the wind tunnel test. This aircraft has one wing and a type of wing-shaped pylon to stand the model. The geometry has been used to obtain the distribution of the acoustic pressure at two different frequencies: 500 Hz and 1 kHz. As a consequence, two different meshes have been used to model the geometry. One of them consists of 253836 triangles (500 Hz frequency) and the other one consists of 1009392 (1 kHz frequency) triangles. Each triangle corresponds to one unknown in the problem.

Figure 2 shows the acoustic pressure on the wing of the aircraft. The engine noise is modelled by means of two point sources placed over the wing.

Figure 2. Real part of the pressure on the wing of an aircraft model. Frequency: 500 Hz.

As we have shown earlier in the paper, the FMM algorithm consists of two parts: an initial setup and standard iterations. The initial setup is only performed once at the beginning of the problem solution whereas the number of required iterations is very dependent upon the geometry, requiring some tens of iterations in the case of an aircraft like the one shown. As a consequence, the setup usually takes much less time than the iterative solution. Hence, we chose to focus the study of the scalability on the MVP, that is the most computationally demanding operation of each iteration.

Table 1 shows some benchmarking data and the group size (the cube shaped group side measured in wavelengths) when using the adaptive group size algorithm. It is worth mentioning that 80 and 92 GMRES iterations were necessary to achieve a residual error smaller than 10^{-2} , for the problems at 500 Hz and 1 kHz, respectively. In this work, the residual error of the predicted total pressure on the aircraft, used to stop the GMRES algorithm, is defined as follows:

$$\epsilon = \frac{\|\mathbf{K}\mathbf{p}^{(i)} - \mathbf{g}\|_2}{\|\mathbf{g}\|_2}, \quad (6)$$

where $\|\cdot\|_2$ is the Euclidean norm (or 2-norm), $\mathbf{p}^{(i)}$ is the iterate at step (i) , $\mathbf{K}$ is the system matrix, and $\mathbf{g}$ is the excitation vector.

Table 1. MVP time, global solution time, group size, and total memory consumption for HP-FASS

Cores	253836 unknowns				1009392 unknowns			
	MVP	Global Solution	Group Size	Cluster Mem. use	MVP	Global Solution	Group Size	Cluster Mem. use
	[s]	[s]	[λ]	[MB]	[s]	[s]	[λ]	[MB]
1	501.7	41046.6	1.01	1571	3563.2	334226.3	1.40	7286
8	70.2	5767.0	1.01	1603	496.0	46631.7	1.40	7397
16	36.4	3003.3	1.01	1618	253.4	23862.3	1.40	7441
32	19.8	1642.5	1.01	1647	132.8	12544.0	1.40	7529
64	10.9	903.4	1.19	1396	74.1	7034.1	1.40	7705
96	7.7	642.4	1.20	1409	52.5	4988.0	1.49	7367
128	6.0	503.5	1.20	1445	40.9	3883.3	1.60	6714
160	5.0	419.8	1.20	1480	34.0	3226.9	1.70	6226
192	4.2	353.9	1.20	1516	29.4	2795.7	1.70	6322
224	3.8	318.9	1.30	1352	25.9	2468.7	1.70	6418
256	3.3	278.2	1.30	1379	23.5	2243.6	1.70	6515

From Table 1, it is possible to see that, for a given group size, a low increase of the memory use is produced with increasing the processors count. In addition, since the memory used by the FMM implemented in here is related in inverse proportion to the group size, the memory use may diminish when increasing the processors count if the group size also increases. It is interesting to note that the direct solution of these problems -253836 and 1009392 unknowns- would require an approximate memory of 480 GB and 7.4 TB, respectively. Meanwhile, the HP-FASS requires no more than 1.6 GB and 7.5 GB, respectively.

Figure 3.(a) shows the global solution time for the small problem whereas Figure 3.(b) shows the global solution time for the big problem. Results for fixed group size vs. adaptive group size are shown in the plots. It is interesting to note the differences among both implementations of HP-FASS. **Using the whole cluster, the new implementation with the adaptive group size selection reduces time-to-solution from more than 11 hours to less than 5 minutes for the small problem, and from almost 93 hours to less than 38 minutes for the big one.**

Figure 3. Global solution time of adaptive vs. fixed group size. (a) $N = 253836$. (b) $N = 1009392$.

Since the reference time was measured on 1 core, the *relative efficiency* may be

defined as follows:

$$E_p = \frac{T_1}{p T_p} \quad (7)$$

Figure 4 shows the parallel efficiency of the MVP for both problems analysed in this study. For the whole cluster usage and moderate size problems, the tool developed in this work provides an efficiency around 60% (using the adaptive group size implementation). It is also worth mentioning the steep drop in the efficiency from 1 core to 8 cores. For the execution on 1 core, the only thread has all the level 3 cache (2 MB) and the full memory bandwidth (10.7 GB/s) of the CPU. Meanwhile, for the execution on 8 cores, 4 threads have to share the level 3 cache and the memory bandwidth of each CPU. This reduction in cache and memory bandwidth per thread has an impact over the translation step that is memory bounded.

Figure 4. MVP efficiency of adaptive vs. fixed group size. (a) $N = 253836$. (b) $N = 1009392$.

6. Conclusions

A hybrid implementation of the FMM applied to acoustic scattering problems is presented in this work. The developed software tool incorporates some techniques suggested by the authors that diminish as possible the constant factor of the FMM time cost while requiring an $\mathcal{O}(N)$ memory cost. In addition, the memory footprint grows very slowly with increasing processors counts thanks to a low data replication. The scalability of the above-mentioned tool is demonstrated by means of execution times (MVP and global solution times) and efficiency metrics. The combination of the FMM as speedup for the MVP required by an iterative method with the highly scalable implementation shown in this paper results on an interesting engineering tool for noise control applications.

Acknowledgements

This work has been partially supported by the “Ministerio de Ciencia e Innovación” from Spain and by FEDER, under the research project TEC2008-01638/TEC (IN-VENTA), and by infrastructure project EQP06-015 co-financed by the European Union. The Airbus A300 series geometry has been provided by the research project GRD1-2001-40147 financed by the European Union. Many thanks are due to Dr. Dimas Suárez-Rodríguez, Dr. Ángel Gutiérrez-Rodríguez, and Dr. José Ranilla-Pastor for their technical support and for the chance given to the authors of using the Clúster de Modelización Científica (Universidad de Oviedo - Spain) in a dedicated way.

References

- [1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, in *Dover Publications*, New York, 1972.
- [2] A. J. Burton and G. F. Miller, *The Application of Integral Equation Methods to the Numerical Solution of Some Exterior Boundary-Value Problems*, Proc. of the Royal Society of London 323 (1971), pp. 201–210.
- [3] I. Foster, *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*, in *Addison Wesley*, 1995.
- [4] P. Geng, J. T. Oden and R. A. van de Geijn, *Massively Parallel Computation for Acoustical Scattering Problems using Boundary Element Methods*, Journal of Sound and Vibration 191 (1996), pp. 145–165.
- [5] L. Greengard, J. F. Huang, V. Rokhlin, and S. Wandzura, *Accelerating fast multipole methods for the Helmholtz equation at low frequencies*, IEEE Computational Science & Engineering, Volume 5, Issue 3 (1998), pp. 32–38.
- [6] Nail A. Gumerov, Ramani Duraiswami and Eugene A. Borovikov, *Data Structures, Optimal Choice of Parameters, and Complexity Results for Generalized Multilevel Fast Multipole Methods in d Dimensions*, Tech. Rep. UMIACS TR 2003-28, Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland, USA, 2003.
- [7] HyperTransport Consortium, *HyperTransport*, (2010). Available online at: www.hypertransport.org
- [8] S. Marburg and S. Schneider, *Performance of iterative solvers for acoustic problems. Part I. Solvers and effect of diagonal preconditioning*, Engineering Analysis with Boundary Elements 27 (2003), pp. 727–750.
- [9] Message Passing Interface Forum, *Message-Passing Interface Standard, Rel. 1.1*, (1995). Available online at: www.mpi-forum.org
- [10] ———, *MPI-2: Extensions to the Message-Passing Interface*, (1997). Available online at: www.mpi-forum.org
- [11] The OpenMP ARB, *OpenMP*, (2004). Available online at: www.openmp.org
- [12] V. Rokhlin, *Diagonal Forms of Translation Operators for the Helmholtz Equation in Three Dimensions*, Applied and Computational Harmonic Analysis 1 (1993), pp. 82–93.
- [13] Y. Saad and M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Non-symmetric Linear Systems*, SIAM J. of Sci. and Statist. Comput. 7 (1986), pp. 856–869.
- [14] J. Song and W. Chew, *Multilevel Fast-Multipole Algorithm for Solving Combined Field Integral Equations of Electromagnetic Scattering*, Microwave and Optical Technology Letters 10 (1995), pp. 14–19.
- [15] C. Waltz, K. Sertel, M. A. Carr and B. C. Usner and J. L. Volakis, *Massively Parallel Fast Multipole Method Solutions of Large Electromagnetic Scattering Problems*, Transactions on Antennas and Propagation 55 (2007), pp. 1810–1816.
- [16] T. W. Wu, *Boundary Element Acoustics*, WIT Press, 2000.