Transactions on the Built Environment vol 69, © 2003 WIT Press, www.witpress.com, ISSN 1743-3509

Multilevel algorithms for the fast evaluation of integral transform in acoustics

I. Hernández-Ramírez & C. H. Venner
Faculty of Engineering and Technology,
University of Twente, The Netherlands

Abstract

The evaluation of integral transforms with oscillatory kernels is a common task in acoustic radiation problems. Integral transforms are generally computationally expensive. Depending on the properties of the kernel in the transform various methods exist to obtain a much faster evaluation. The aim of the present research is to investigate if the algorithm "Multilevel Multi-Integration" developed by Brandt [1], and successfully used in contact mechanics and multi-body interaction problems can also be used for acoustic problems. First the version of the algorithm is described aimed at the case of oscillatory kernels. Next, by means of a model problem, it is shown that at the expense of a small and controllable error that is independent of the wavenumber the algorithm facilitates fast and efficient computation even for high frequencies. Even though the model problem considered is only one dimensional and thereby really very simple the results obtained are encouraging for further research.

1 Introduction

Numerical methods such as the Boundary Element Method have shown to be a powerful tool to evaluate and predict behaviour in real engineering problems. However, for medium and high frequencies acoustic problems can be difficult to deal with accurately and efficiently (small computing times). In a numerical simulation the domain is discretized with discrete elements and for high frequencies the number of elements that must be used in the evaluation to obtain the required accuracy results is large which then leads to enormous computing times because of the quadratic dependence of computing time on the number of elements inherent in the integral formulations.

Several methods have been developed for the fast(er) evaluation of integral transforms, i.e. Fast Fourier Techniques (FFT), Fast Multipole (FM) expansion, and Multilevel Multi-Integration (MLMI), and have been applied to the integral transforms appearing in different fields in science. Each algorithm except for FFT uses some properties (form or otherwise) of the kernel to reduce the computing time at the expense of some error. FFT and FM have already been applied in acoustics; see [2, 3]. However, MLMI, successfully used in contact mechanics, and many-body long-range interactions [4] has not been applied in acoustics yet. The basic idea of MLMI is to use a set of auxiliary coarser grids in such a way that in each part of the domain the contribution to the summation is computed representing the kernel on a grid so coarse that the resulting error remains within a prescribed bound. The algorithm exploits the smoothness properties of the kernel. An oscillatory kernel is obviously not smooth but in that case the algorithm is modified and the problem is rewritten into a combination of problems with asymptotically smooth kernels, see Brandt [1]. The aim of the present research is to investigate if the methodology proposed in [1] leads to an efficient fast evaluation algorithm for integral transforms in acoustics. In this paper the first steps are reported.

2 Theory

In acoustic radiation problems integral transforms of the following form appear:

$$p(x) = \int_{\Omega} G(x,y)\, \mathrm{v}(y)\, dy, \qquad x \in \Omega \subseteq \Re^d, \tag{1}$$

where v(y) is a known function, $p(x)$ is the target function and $G(x,y)$ is the kernel function which is the Green's functions. For two dimensional (2D) and three dimensional (3D) it is given by:

$$\text{2D: } G(x,y) = -\frac{i}{4} H_0^{(2)}\left(k|x-y|\right), \text{ 3D: } G(x,y) = -\frac{e^{ik|x-y|}}{4\pi|x-y|}, \tag{2}$$

see [5], where $H_0^{(2)}$ is the zero-order Hankel function of the second kind [6]. This Green's function (the kernel) represents the free-field pressure in point x due to acoustic point source y. Here it is singular when at $|x-y|=0$ between field and it is frequency dependent by the wavenumber k. This type of integral in most cases cannot be evaluated analytically and a numerical approach is needed. For simplicity assume eqn (1) is discretized on the entire domain Ω with meshsize h and $n=O(h^{-d})$. In that case the discrete problem becomes to evaluate a multi-summation or discrete integral transform of the type:

$$p^h(x_i) = h^d \sum_j \overline{G}(x_i, y_j)\, \mathrm{v}^h(y_j), \tag{3}$$

where each evaluation of $p(x_i)$ costs $O(n)$ operations and the complete evaluation in all points of the domain costs around of $O(n^2)$ operations. In many real problems a large number of points n may be essential for accuracy and as a consequence the amount of work needed to evaluate the integral transform may be huge. The aim Multigrid algorithms is to remove such computational bottlenecks and in this case thus to reduce the complexity of this multi-integration in order to avoid excessive computing times at the expense of a minimal and controllable loss of accuracy.

3 Fast evaluation

In this section the algorithm for the fast evaluation eqn (3) is described. For simplicity assume a one-dimensional case. First the case of a "smooth kernel" is described which illustrates the basic principle. Next, the modifications for an "asymptotically smooth" kernel will be explained as well as the changes needed for the case of an oscillatory kernel. First some details of notation. A MLMI algorithm uses a set of auxiliary coarser grids to evaluate the multi-summation eqn (3). Each grid is coarser than the previous one usually by a factor 2. For simplicity in the description given here only two grids will be assumed. A fine grid with mesh size h on which the transform is to be evaluated and a coarse grid with mesh size H=2h. The case of multiple grids simply follows from the two grid case by recursion. Also for simplicity one may assume the grids to be uniform at first. Small case letters will indicate variables and running index on the grid h. Capital letters will be used for variables and running index on the coarser grid H. So $\mathrm{v}_j^h = \mathrm{v}^h(y_j)$ stands for the value of v in gridpoint j on the fine grid h, which has location $y_j = y_0 + jh$. In the same way $\mathrm{V}_J^H = \mathrm{V}^H(Y_J)$ is the value of a coarse grid representation of v at a coarse grid point J with location $Y_J^H = Y_0 + JH$. Y_0^H and y_0^h are the same for the fine and coarse grids. In the same way the notation will apply to the points of a fine and a coarse grid of the evaluation grid (x_i^h and X_I^H) in which the integral transform is needed.

3.1 Smooth kernel

If $G(x,y)$ is smooth on the scale H as a function of y then its value at any point y_j can be obtained from interpolation of the value at points Y_J of a coarse grid, with mesh size H.

$$\overline{G}(x_i, y_j) = \sum_J \omega_{jJ} \overline{G}(x_i, Y_J) + O(\varepsilon), \tag{4}$$

where ω_{jJ} are interpolation weights and ε is the interpolation error, $\varepsilon = (\gamma H)^P \left|\overline{G}^{(P)}\right|$. Here $\left|\overline{G}^{(P)}\right|$ is the maximum of a *pth* derivative of $\overline{G}$. The

summation over J in eqn (4) usually extends over only p points around of the point y_j, so it is a local summation. Substitution of eqn (4) in eqn (3) gives:

$$p^h(x_i) = \sum_j \sum_J \omega_{jJ} \overline{G}(x_i, Y_J) \mathrm{v}_j^h + O(\varepsilon) . \tag{5}$$

Changing the order of summation yields:

$$p^h(x_i) = \sum_J \overline{G}(x_i, Y_J) \sum_j \omega_{jJ} \mathrm{v}_j^h + O(\varepsilon) = \sum_J \overline{G}(x_i, Y_J) \mathrm{V}_J^H + O(\varepsilon) , \tag{6}$$

where

$$\mathrm{V}_J^H \equiv \sum_J \omega_{jJ} \mathrm{v}_j^h . \tag{7}$$

By definition the summation over j (for each J) in eqn (6) is also a local summation over $O(2p)$. As it represents the transpose of interpolation eqn (7) is referred to an "anterpolation". The change of summation going from eqn (5) to eqn (6) is the first crucial step in the algorithm. Its effect is that a "fine grid" summation over the index j is replaced by a "coarse grid" summation over the index J. Summarizing, if $\overline{G}(x,y)$ is smooth as a function of y the computation of $p(x_i)$ by summation over all y_j at the expense of an $O(\varepsilon)$ error can be replaced by a coarse grid summation eqn (6) over all Y_J with the "collected charges" V_J^H defined by anterpolation of the charge v_j^h to coarse grid according to eqn (7). It is important to note that although the anterpolation appears to be the transfer of a function from a fine grid to a coarse grid with the risk of loosing high frequency information, it is not! No assumptions are made regarding the fine grid function v. It may be highly oscillatory. The only assumption made in the algorithm is regarding the smoothness of the kernel. The anterpolation arises as a result of the change of the order of summation going from eqn (5) to eqn (6).

Next assume $\overline{G}(x, y)$ is also smooth as a function of x. In that case for any x up to ε error $\overline{G}(x, y)$ can be interpolated from its values on a coarse grid X_I with mesh size H.

$$\overline{G}(x, y) = \sum_I \overline{\omega}_{iI} \overline{G}(X_I, y) + O(\overline{\varepsilon}) . \tag{8}$$

For a $\overline{p}$ order interpolation the error is defined by $\overline{\varepsilon} = (\gamma H)^{\overline{p}} \left| \overline{G}^{(\overline{p})} \right|$, often $\overline{p} = p$ can be used. Again the summation over I is a local summation involving only $\overline{p}$ terms. This is the second crucial step in the algorithm. The second step in fact implies that it is not necessary to compute the transform in all points of

the fine grid. Admitting a certain error that can be controlled by the interpolation it is sufficient to only compute the summations in "coarse grid" points, and then obtain its value in all points of a finer grid by means of interpolation of a sufficiently high order.

The result of both steps now is that up to $O(\varepsilon)$ error the computation of $p(x_i)$ in all evaluation grid points x_i involving a summation over all integration grid points y_j can be replaced by:

- Anterpolation eqn (7): $$\mathrm{V}_J^H = \sum_J \omega_{jJ} \mathrm{v}_j^h \,.$$

- Coarse grid summation: $$P^H(X_I) = \sum_J \overline{G}(X_I, Y_J)\, \mathrm{V}_J^H \,. \tag{9}$$

- Interpolation: $$p^h(x_i) = \sum_I \omega_{iI} P^H(X_I) \,. \tag{10}$$

If the kernel is sufficiently smooth the coarsening step from grid y_j to Y_J and x_i to X_I can be such that the evaluation of the coarse grid summation requires $O(n)$ work. As the anterpolation and interpolation also require at most $O(n)$ work, the total work needed to obtain the direct transform has been reduced from $O(n^2)$ to $O(n)$ work at the expense of an error $O(\varepsilon)$. This error can be made small by the choice of the order of interpolation. Generally $p = O(\log(1/\varepsilon))$ is needed [7]. However, in practice it is only needed to ensure that the additional error introduced by the fast evaluation is small compared to the discretization error that is made anyway.

3.2 Asymptotical smooth kernel

In many practical cases the kernel is not smooth everywhere, e.g. it has a singularity at $x=y$. In that case locally in a region $|i-j| \le mh$ the error in the interpolation result will not be small, and the fast evaluation result must be connected for this error. This can be done quite straightforwardly. It involves computations of correction terms of the form:

$$\delta_i = \sum_{|i-j|\le mh} (\overline{G}(x_i, y_j) - \widetilde{G}(x_i, y_j))\, \mathrm{v}_j^h \,, \tag{11}$$

where $\widetilde{G}$ is the interpolated kernel value. For details the reader is referred to Brandt and Lubrecht [4]. For a logarithmic kernel ($1/r$ type singularity) it has been shown that $m = O(p) = O(\log(1/\varepsilon))$ is sufficient to ensure that the evaluation error is small compared to the discretization error. The total amount of work carried out to obtain the discrete transform is the $O(n \log(n))$ [8].

3.3 Oscillatory kernel algorithm (1d)

Finally consider the task of the evaluation of a discrete transform eqn (3), in which the kernel is an implicit oscillatory function defined as $G_o(k|x-y|)$ (as can be the Hankel function), where k is the wavenumber. To apply the fast evaluation algorithm explained now it is necessary to obtain a formulation in which the oscillatory term is explicit. This can be achieved by rewriting the problem. First define:

$$\overline{G}(x_i, y_j) = \overline{G}_o\left(k|x_i - y_j|\right) e^{ik(x_i - y_j)}. \tag{12}$$

Now the integral transform of the eqn (3) can be represented by the multi-summation (The i preceding k will always denote $\sqrt{-1}$, unrelated to i appearing elsewhere):

$$p^h(x_i) = h^d \sum_j \overline{G}_o(x_i, y_j)\, e^{ik(x_i - y_j)}\, e^{-ik(x_i - y_j)}\, \mathrm{v}^{\mathrm{h}}(y_j), \tag{13}$$

where the multiplication of $\overline{G}_o(x_i, y_j)$ by $e^{ik(x_i - y_j)}$ and $e^{-ik(x_i - y_j)}$ is introduced to keep the same meaning of the original equation, but with the advantage that now we have inside the summation asymptotically smooth functions ($\overline{G}_o[k(x_i - y_j)]\, e^{ik(x_i - y_j)}$ and $\overline{G}_o[k(y_j - x_i)]\, e^{-ik(x_i - y_j)}$ in the positive an negative direction respectively from the listener point of view) multiplied with an exponential ($e^{ik(x_i - y_j)}$ and $e^{-ik(x_i - y_j)}$ on positive and negative direction respectively). At this point, we have achieved a form to which the MLMI algorithm for the asymptotically smooth kernel can be applied. Eqn (13) is rewritten as:

$$p^h(x_i) = e^{-ikx_i}\, p_+^h(x_i) + e^{ikx_i}\, p_-^h(x_i), \tag{14}$$

where $p_+^h(x_i)$ and $p_-^h(x_i)$ are the sub-transforms of eqn (13), each representing the summation in one direction which take the form:

$$p_+^h(x_i) = \sum_{j \geq i} \overline{G}(x_i, y_j) e^{iky_j}\, \mathrm{v}_+^{\mathrm{h}}(y_j), \quad p_-^h(x_i) = \sum_{j<i} \overline{G}(x_i, y_j)\, e^{-iky_j}\, \mathrm{v}_-^{\mathrm{h}}(y_j). \tag{15}$$

Here the functions $\mathrm{v}_+^{\mathrm{h}}(y_j)$ and $\mathrm{v}_-^{\mathrm{h}}(y_j)$ are:

$$\mathrm{v}_+^{\mathrm{h}}(y_j) = e^{iky_j}\, \mathrm{v}^{\mathrm{h}}(y_j), \quad \mathrm{v}_-^{\mathrm{h}}(y_j) = e^{-iky_j}\, \mathrm{v}^{\mathrm{h}}(y_j). \tag{16}$$

And the kernel for the sub-transform given by:

$$G_+(x,y)=\begin{cases}\overline{G}(x_i,y_j), & y\geq x,\\ 0, & y<x,\end{cases}\qquad G_-(x,y)=\begin{cases}0, & y\geq x,\\ \overline{G}(x,y), & y<x.\end{cases}\tag{17}$$

Then, we can rewrite eqn (15) in the form,

$$p_+^h(x_i)=\sum_j G_+(x_i,y_j)\,\mathrm{v}_+^h(y_j),\qquad p_-^h(x_i)=\sum_j G_-(x_i,y_j)\,\mathrm{v}_-^h(y_j).\tag{18}$$

Now if $\overline{G}(x,y)$ is asymptotically smooth, so are also $G_+(x,y)$ and $G_-(x,y)$. Hence eqn (18) can now be evaluated by using (twice) the algorithm of the sections (2.1) and (2.2) above. Summarizing, for the one-dimensional problem the algorithm for the fast evaluation of the discrete integral transform with oscillatory kernel is simply twice the application of the algorithm for the smooth or asymptotically smooth kernel. So, to evaluate eqn (13) means first to calculate eqn (16), then evaluate eqn (18) by the algorithm of sections (2.1) and (2.2), and finally compute eqn (14). The total amount of work is then at most twice as large as for the singular smooth kernel, so it is still $O(n\log(n))$. Note that the main step to obtain a fast evaluation algorithm for the oscillatory kernel problem is to incorporate the oscillatory factors into v and p. For the one-dimensional problem this is achieved easily by separately treating the (only) two directions $x-y>0$ and $x-y<0$. Note that in the more general two-dimensional case the approach is more complex. In that case several directions have to be identified.

4 Test example

As a one dimensional test problem is taken the evaluation of:

$$p(x)=\int_{-1}^{1} H_0^{(2)}\left(k|x-y|\right)\sin(ky)\,dy\,,\tag{19}$$

where the Hankel function of the order zero and second kind appears as the kernel. The problem can be interpreted as the computation of the pressure at the surface of a harmonically oscillating infinitely wide strip that is on both sides held in a baffle. The problem was discretized on a uniform grid and approximated by:

$$p(x_i)=\sum_j H_0^2\left(k|x_i-y_j|\right)e^{ik(x_i-y_j)}\;e^{-ik(x_i-y_j)}\;\sin(ky_j),\tag{20}$$

due to the appearance of the Hankel function there is no way to analytically compute the kernel influence factors $\overline{G}(x,y)$. This was done numerically using a double-adaptive algorithm developed by Oliver [9].

Table 1. Real part error of multi-integration eqn (21) evaluated by direct summation and MLMI algorithm (medium frequencies k=50).

l	k=l	K=l-1	k=l-2	k=l-3	k=l-4	k=l-5	k=l-6
3	1.04E-01	1.04E-01	1.04E-01*				
4	1.04E-01	1.04E-01	1.04E-01	1.04E-01			
5	1.50E-02	1.50E-02	1.50E-02	1.50E-02*	1.50E-02		
6	5.78E-03	5.78E-03	5.78E-03	5.78E-03	5.78E-03	5.78E-03	
7	2.07E-03	2.07E-03	2.07E-03	2.07E-03	2.07E-03*	2.07E-03	2.07E-03
8	7.82E-04	7.82E-04	7.82E-04	7.82E-04	7.82E-04	7.82E-04	7.82E-04
9	3.17E-04	3.17E-04	3.17E-04	3.17E-04	3.17E-04	3.17E-04*	3.17E-04
10	1.38E-04	1.38E-04	1.38E-04	1.38E-04	1.38E-04	1.38E-04	1.38E-04
11	6.33E-05	6.33E-05	6.33E-05	6.33E-05	6.33E-05	6.33E-05	6.33E-05*

Table 2. Imaginary part error multi-integration eqn (21) evaluated by direct summation and MLMI algorithm (medium frequencies k=50).

l	k=l	k=l-1	k=l-2	k=l-3	K=l-4	k=l-5	k=l-6
3	9.94E-02	9.94E-02	9.94E-02*				
4	1.02E-01	1.02E-01	1.02E-01	1.02E-01			
5	1.40E-02	1.40E-02	1.40E-02	1.40E-02*	1.40E-02		
6	4.40E-03	4.40E-03	4.40E-03	4.40E-03	4.40E-03	4.40E-03	
7	1.62E-03	1.62E-03	1.62E-03	1.62E-03	1.62E-03*	1.62E-03	1.62E-03
8	7.52E-04	7.52E-04	7.52E-04	7.52E-04	7.52E-04	7.52E-04	7.52E-04
9	4.73E-04	4.73E-04	4.73E-04	4.73E-04	4.73E-04	4.73E-04*	4.73E-04
10	3.17E-04	3.17E-04	3.17E-04	3.17E-04	3.17E-04	3.17E-04	3.17E-04
11	2.33E-04	2.33E-04	2.33E-04	2.33E-04	2.33E-04	2.33E-04	2.33E-04*

In the Tables 1 and 2 are given the error obtained by direct summation and versus MLMI with different set of coarse grids. The results were obtained for the value of wave number k=50. The results for a high frequency k is 300 are shown in the Tables 3 and 4. The MLMI algorithm was used with a sixth order interpolation and anterpolation. In each table the first column represents the target grid identified by the level l. The second column identified with k=l contains the discretization error, or the error in the direct summation result on this target grid. The third column contains the error when the MLMI algorithm is used with one coarse grid, with H=2h, i.e. if the target grid is level l then actual summation is carried out on level k=l-1. The next column the error in the result when two coarser grids are used so in that case the actual summation is carried out on a grid with H=4h (k=l-2). The number of points for each grid is determined by $2^{l+2}+1$, employing a coarsest auxiliary grid with $2^{k+2}+1$ points. So each row represents a given target error in the k=l column and the error in the MLMI result as a function of the gridlevel k on which the actual summation is carried out in the other columns. At each row, the results marked with an asterisk (*) indicate the case when the actual summation gridlevel k contains ($\sqrt{n}$) points if n is the number of nodes on the target grdilevel l. So coarsening from the target grid down to this coarse grid implies that the total work of the algorithm is reaches the minimum of $O(n \log(n))$ operations. From the tables it becomes clear that with the fast evaluation algorithm one can easily coarsen all the way to a grid with $\sqrt{n}$ points at the expense at an error that is at most comparable to the discretization error.

Table 3. Real part error of multi-integration eqn (21) evaluated by direct summation and MLMI algorithm (medium frequencies k=300).

l	k=l	k=l-1	k=l-2	k=l-3	K=l-4	k=l-5	k=l-6
7	3.81E-03	3.81E-03	3.81E-03	3.81E-03	3.81E-03*	3.81E-03	3.81E-03
8	2.03E-03	2.03E-03	2.03E-03	2.03E-03	2.03E-03	2.03E-03	2.03E-03
9	1.31E-03	1.31E-03	1.31E-03	1.31E-03	1.31E-03	1.31E-03*	1.31E-03
10	6.55E-04	6.55E-04	6.55E-04	6.55E-04	6.55E-04	6.55E-04	6.55E-04
11	3.16E-04	3.16E-04	3.16E-04	3.16E-04	3.16E-04	3.16E-04	3.16E-04*
12	1.58E-04	1.58E-04	1.58E-04	1.58E-04	1.58E-04	1.58E-04	1.58E-04
13	8.08E-05	8.08E-05	8.08E-05	8.08E-05	8.08E-05	8.08E-05	8.08E-05

Table 4. Imaginary part error of multi-integration eqn (21) evaluated by direct summation and MLMI algorithm (medium frequencies k=300).

l	k=l	k=l-1	k=l-2	k=l-3	k=l-4	k=l-5	k=l-6
7	6.79E-03	6.79E-03	6.79E-03	6.79E-03	6.79E-03*	6.79E-03	6.79E-03
8	2.16E-03	2.16E-03	2.16E-03	2.16E-03	2.16E-03	2.16E-03	2.16E-03
9	1.22E-03	1.22E-03	1.22E-03	1.22E-03	1.22E-03	1.22E-03*	1.22E-03
10	6.48E-04	6.48E-04	6.48E-04	6.48E-04	6.48E-04	6.48E-04	6.48E-04
11	3.65E-04	3.65E-04	3.65E-04	3.65E-04	3.65E-04	3.65E-04	3.65E-04*
12	2.16E-04	2.16E-04	2.16E-04	2.16E-04	2.16E-04	2.16E-04	2.16E-04
13	1.33E-04	1.33E-04	1.33E-04	1.33E-04	1.33E-04	1.33E-04	1.33E-04

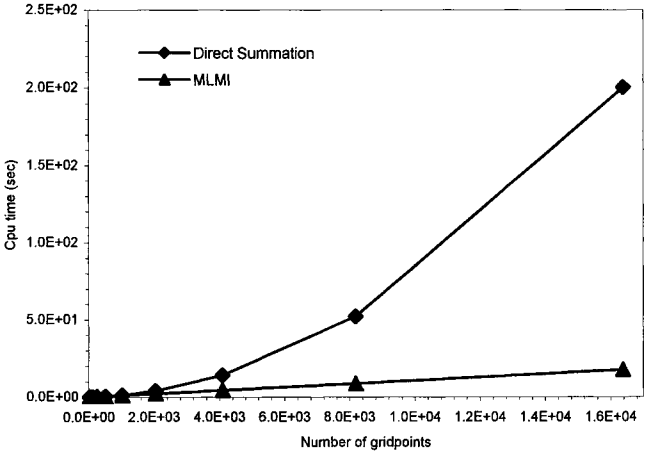

Figure 1. Computing time used for evaluate eqn (21) using Direct Summation and MLMI algorithm.

Finally, the real gain now becomes clear from looking at the computing time that is invested to obtain the result, see Figure 1. This figure shows the invested cpu time for direct summation versus the fast evaluation using Multigrid algorithm. Clearly the computing time for the MLMI algorithm is linear in the number of nodes whereas for the direct summation it is quadratic in the number of nodes, so, at negligible loss of accuracy large computing time reductions are obtained.

5 Conclusions

The Multilevel Algorithm for the fast evaluation of integral transforms with implicit oscillatory kernels was successfully applied on a one-dimensional problem in which the Hankel function appears as a kernel. It was shown that applying the Multilevel Multi-Integration algorithm the required computing time needed to evaluate the discrete transform could be reduced from $O(n^2)$ operations for direct summation to $O(n \log n)$ operations leading to large computing time reductions for large n. The gain in computing time is obtained at the expense of a controllable error that was shown to be small compared to the error that is made anyway because of solving a discrete problem rather than the continuous problem. Naturally this one dimensional problem is still much too simple for practical applications. However, the results are certainly encouraging and stimulating for further development to the two dimensional case.

Acknowledgement

This research is supported by the Dutch Technology Fundation (STW).

References

[1] Brandt, A., Multilevel computations of integral transform and particle interaction with oscillatory kernels, *Computer Physics Communications*, 65, pp. 24-38, 1991.

[2] Tournour, M.A & Atalla, N., Efficient evaluation of the acoustic radiation using multipole expansion, *Int. J. for Numerical Method in Engineering*, 46, pp. 825-837, 1999.

[3] Tournour, M.A. & Atalla, N., A novel acceleration method for the variational boundary element approach based on multipole expansion, *Int. J. for Numerical Methods in Engineering*, 42, pp. 1199-1214, 1998.

[4] Brandt, A. & Lubrecht, A.A., Multilevel matrix multiplication and fast solution of integral equations, J. of Comp. Physics, 90(2), pp. 348-370, 1990.

[5] Blackstock, D.T., *Fundamentals of Physical Acoustics*, John Wiley & Sons, 2000.

[6] Abramowitz, M. & Stegun, I.A., *Handbook of Mathematical Functions: with formulas, graphs and mathematical tables*, Dover publications, 1964.

[7] Brandt, A. & Venner, C.H., Multilevel evaluation of integral transforms with asymptotically smooth kernels, *SIAM J. of Scientific Computation*, 19(2), pp. 468-492, 1998.

[8] Venner, C.H. & Lubrecht, A.A., *Multilevel methods in lubrication*, Elsevier, Tribology Series, 37, 2000.

[9] Oliver, J., A doubly-adaptive Clenshaw-Curtis quadrature method, *The Computer Journal*, 15(2), pp. 141-147, 1972.