Developments of multi-level boundary element methods for steady heat diffusion problems

M.M. Grigoriev*, G.F. Dargush

Department of Civil Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA

Abstract

We have recently developed a novel multi-level boundary element method (MLBEM) for steady heat diffusion in irregular two-dimensional domains. This paper extends the MLBEM methodology to dramatically improve the performance of the original multi-level formulation. First, we perform analyses of numerical error and computational complexity for the multi-level boundary element algorithm and show that the optimal complexity of the algorithm is $O(N \log N)$. Next, we consider a model problem of line multi-integral evaluation and investigate the performance of the MLBEM formulation using a single-patch approach. Then we study the performance of the multi-level boundary element formulation on an example Neumann problem of steady heat diffusion leading to a boundary integral equation of the second kind. Here, we solve a problem involving four million degrees of freedom in less than one hour on a desktop workstation. Finally, we consider a model problem in a unit square with mixed boundary conditions and study the performance for the new MLBEM formulation.

Keywords: Steady heat diffusion; Multi-level boundary element method; Error analysis

1. Introduction

A standard boundary element method (BEM) technique requires two major steps to obtain a numerical solution. During the first step, a global boundary element matrix is formed, which necessitates an integration of the kernel functions over each of the boundary elements. The global matrix solution constitutes the second step to obtain an unknown vector of boundary temperatures and heat fluxes. While the first step requires N^2 operations, the complexity of the second step is of the order N^3 when using Gauss elimination. For relatively small numbers of degrees of freedom (e.g. $N < 10^4$), the regular BEMs are fast and practical even on a single-processor computer, since the integration over the boundary elements dominates over the matrix solution. However, most problems of practical importance, especially in three dimensions, require fine boundary element meshes leading to very large numbers of degrees of freedom, i.e. $N > 10^5$. For these boundary element discretizations, the conventional BEM algorithms become prohibitively expensive even on state-of-the-art supercomputers due to both memory and run-time requirements.

In the past two decades, several fast algorithms have been proposed in order to extend the applicability of regular boundary element methods to larger numbers of degrees of freedom. These fast methods include Barnes–Hut [1,2], wavelets [3,4], fast multipole [5–9], and multi-level multi-integration [10–12] algorithms. While the fast multipole and wavelet-based approaches have been developed extensively, multi-level multi-integration (MLMI) methods have not enjoyed a wide acceptance in computational practice, despite the great potential of the methods. The MLMI method was proposed by Brandt and Lubrecht [11] for fast evaluation of multi-integrals involving both smooth and singular-smooth kernels. Following this pioneering work [11], Lubrecht and Ioannides [13], Polonsky and Keer [14], and Venner and Lubrecht [12] extended the MLMI methods to the solution of elastohydrodynamic lubrication and rough surface-contact problems.

Despite the recent progress in developments of the multi-level boundary element methods (MLBEM) for linear problems [10,15–17], there still exist several very important issues in the MLBEM algorithm that need to be addressed. First and foremost, the thorough analysis

* Corresponding author. Tel.: +1 716 645 2593, ext. 2249; Fax: +1 716 645 3733; E-mail: mmg3@eng.buffalo.edu

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

of errors arising in the numerical formulation is mandatory to set the application borderlines. Although the error analysis was presented by Brandt and Lubrecht in their original paper [11], we believe that more detailed analysis is required to accommodate a level-by-level transfer of the multi-integration for a set of boundary element meshes. The other very important issue is the analysis of the multi-grid algorithm utilized to accelerate the convergence of the iterative solvers. In this paper, we focus on the former issue. We re-visit the error analysis for the MLMI and present the reader with the relations that permit the estimation of the truncation error with the numbers of transfer levels and correction points and the order of kernel interpolation/anterpolation. We show that the theoretical results agree well with the numerical calculations. Moreover, we find an optimal combination between the interpolation/anterpolation order and the number of transfer levels that minimizes the work necessary to obtain an accurate solution. We demonstrate that the asymptotic computational complexity of the presented method is indeed of the order $O\ (N \ln N)$.

Finally in this paper, we rethink and modify the MLBEM formulation [10] to dramatically improve the performance of the numerical approach. First, we pre-evaluate the singular corrections for both matrix-vector and matrix-transpose-vector multiplication. Next, the centered correction stencils are also pre-evaluated for an internal source point and then applied to any other source point, except for the patch-end points. These two modifications in the original algorithm lead to tremendous reductions in run-time and memory requirements. In order to demonstrate the performance of the proposed method, we consider three example problems with exact solutions.

2. Multi-level boundary element methods

2.1. Governing equation and integral formulation

The steady-state diffusion of heat is governed by the following dimensionless equation:

$$\frac{\partial^2 T}{\partial x_i \partial x_i} = 0 \tag{1}$$

In the Laplace equation, Eq. (1), x_i is the Eulerian coordinate and T is the temperature. Both temperatures $\bar{T}(x)$ on $x \in \Gamma_T$ and normal fluxes $\bar{Q}_n(x)$ on $x \in \Gamma_Q$ may be specified as Dirichlet and Neumann boundary conditions, respectively. Note that $\Gamma_T \cap \Gamma_Q = 0$ and $\Gamma_T \cup \Gamma_Q = \Gamma$, where the surface Γ bounds the computational domain Ω. The corresponding well-known integral form of the boundary value problem is given by:

$$c(\xi)T(\xi) + \int_\Gamma Q(x)g(x-\xi)d\Gamma(x) = \int_\Gamma T(x)f(x-\xi)d\Gamma(x) \tag{2}$$

In Eq. (2),

$$g(x-\xi) = \frac{1}{2\pi}\ln\frac{1}{r} \qquad \text{and} \qquad f(x-\xi) = \frac{(x_i-\xi_i)n_i}{2\pi r^2}$$

are the potential kernels for two dimensional problems, with r as the radial distance between x and ξ, $Q(x) = -\frac{\partial T(x)}{\partial n}$ is the normal heat flux, $n_i(x)$ is the unit outward normal to the surface $\Gamma(x)$, and ξ stands for the collocation point $\xi \in \Omega$. The geometric function $c(\xi) = 0.5$ when ξ is on the smooth boundary and $c(\xi) = 1$ when ξ lies inside the domain Ω.

2.2. Boundary element method discretization

Let us introduce P smooth non-overlapping boundary patches Γ_p for $p = 1, 2, \ldots, P$, and assume that the variation of boundary temperature $T(x)$ and normal heat flux $Q(x)$ over every patch Γ_p is smooth. Then, Eq. (2) may be written as follows:

$$c(\xi)T(\xi) + \sum_{p=1}^{P}\int_\Gamma Q(x)g(x-\xi)d\Gamma(x) = \sum_{p=1}^{P}\int_{\Gamma_p} T(x)f(x-\xi)d\Gamma(x) \tag{3}$$

Introducing linear boundary elements over all boundary patches, we discretize integral Eq. (3) to the form:

$$c_m T_m + \sum_{p=1}^{P}\sum_{n=1}^{N_p} Q_n^{(\alpha)} G_{mn}^{(\alpha)} = \sum_{p=1}^{P}\sum_{n=1}^{N_p} T_n^{(\alpha)} G_{mn}^{(\alpha)} \tag{4}$$

The algorithm for an accurate evaluation of discrete coefficients $G_{mn}^{(\alpha)}$ and $F_{mn}^{(\alpha)}$ is detailed in Grigoriev and Dargush [10]. Here, to facilitate the MLBEM algorithm, we apply boundary conditions and recast the discrete integral, Eq. (4), into the following matrix form:

$$\mathbf{Au} = \mathbf{b} \tag{5}$$

where $\mathbf{u}$ is the generalized vector of unknown temperatures and heat fluxes and $\mathbf{b}$ is the known force vector. Note that the global matrix $\mathbf{A}$ of size $N \times N$ is now dense.

Similar to our earlier work [10], we utilize a bi-conjugate gradient method [18] to solve the matrix equation, Eq. (5). Since this iterative approach requires both matrix-vector ($\mathbf{Au}$) and matrix transpose-vector ($\mathbf{A}^T \times \mathbf{u}$) multiplications at any iteration, we use fast MLMI [10,11] for these operations. In the following two

subsections, we briefly outline the MLMI algorithm that was detailed in Grigoriev and Dargush [1] and discuss modifications introduced in the fast matrix-vector operations that allow tremendous savings in computational resources.

2.3. Multi-level multi-integration for matrix-vector multiplication

For patch Γ_p, we introduce a sequence of L boundary element meshes M^0, M^1, …, M^L, where M^0 is the finest-level mesh and M^L is the coarsest boundary element mesh. On the finest-level mesh, the block-matrix-vector multiplication can be represented as the following multi-summation over every patch:

$$w_i^{(0)} = h_0 \sum_{j=0}^{N_0} H_{i,j}^{(0)} u_j^{(0)} - R_i \qquad \text{for} \qquad i = 0, 1, 2, \ldots, N_0 \tag{6}$$

In Eq. (6), the multi-summation is performed over index j representing the set of boundary elements on M^0, h_0 is the scaled boundary element mesh size on the finest-level mesh, $u_j^{(0)}$ stands for the generalized vector of unknown temperatures or heat fluxes at any iteration on the finest level mesh, and $H_{i,j}^{(0)}$ is the generalized form of assembled coefficients.

Following the approach presented earlier by the current authors [10], we introduce a coarse-to-fine-level interpolation of the generalized kernels that permits a fine-to-coarse-level transfer of the multi-summation equation, Eq. (6). Furthermore, owing to the asymptotic smoothness of the kernels, we restrict correction of the interpolated kernels only to m nodes at the vicinity of the source point. In doing so, all corrections outside the singularity zone are neglected, since these are smaller than the error due to a boundary element discretization. Therefore, the multi-integrals on the coarsest level M^L can be written in the following form:

$$w_i^{(L)} \cong h_L \sum_{j=0}^{N_L} H_{i,j}^{(L)} u_j^{(L)} + C_i^{(L)} \tag{7}$$

Here, the total correction $C_i^{(L)}$ includes both the singular and the patch-end corrections [10]. The multi-integrals on any finer-level meshes $k < L$ are evaluated recursively from the coarsest to the finest levels using the following [10]:

$$w_i^{(k)} \cong \left[\hat{\mathcal{I}}_{k+1}^{k} w^{(k+1)}\right]_i + h_k \sum_{|i-j| \le m}^{N_k} \left(H_{i,j}^{(k)} - \hat{H}_{i,j}^{(k)}\right) u_j^{(k)} + C_i^{(k)} - \hat{C}_i^{(k)} \tag{8}$$

We should note that the evaluation of the correction $\hat{C}_i^{(k)}$ requires a sequential evaluation of coefficients $C_i^{(k)}$ for stencil nodes adjacent to the current source node i. Obviously, the computational load as well as storage requirements are reduced significantly provided that the corrections $C_i^{(k)}$ are pre-evaluated. Moreover, note that these pre-evaluations are not needed to be performed for all levels. The computer requirements are reduced even to a more dramatic extent if the centered correction stencils are pre-evaluated for an internal source point and then applied to any other source point for which the correction stencil is not affected by the patch ends. Indeed, several source points close to the patch ends require re-evaluation of the individual correction stencils. However, as the boundary mesh gets refined, the additional computations become insignificant.

2.3.1. Multi-level multi-integration errors

The G-kernel interpolation error for the fine-to-coarse-level transfer can be given by

$$G(n) - \tilde{G}(n) = \frac{(-1)^{p+1/2}(p!!)^2 h_0}{2\pi(p+1)n^{p+1}} \tag{9}$$

where $n = |i - j|$ is the distance between the source and interpolation points and p is the kernel interpolation order. Note that the interpolation error decays rapidly as n increases. Thus, the truncation error for the multi-integral equation, Eq. (6), when only m correction points are retained within the singularity zone will be as follows:

$$E_T^{(0)} \le \frac{(p!!)^2 h_0}{2\pi p(p+1)m^p} \|u(\zeta)\|_\infty \tag{10}$$

In Eq. (10), $\|u(\zeta)\|_\infty$ is the temperature maximum at some intermediate point ζ. Since the inter-level transfer error $E_T^{(k)}$ remains the same for all levels $k = 0, 1, \ldots, L - 1$, then we get the total truncation error as follows:

$$E_T \le |E_T^{(0)}| + |E_T^{(1)}| \cdots |E_T^{(L-1)}| \le L|E_T^{(0)}| \tag{11}$$

However the total truncation error given by Eq. (11) must be less than the linear boundary element discretization error, a lower bound of which can be represented approximately by the following

$$E_d \cong \frac{h_0^2(1 + \ln 2)}{16\pi} \|u''(\eta)\|_c \tag{12}$$

where $\|u''(\eta)\|_c$ is some norm for the second derivative of the temperature at point η. Therefore, we express the number of correction points m within the singularity zone necessary to retain the accuracy of the multi-level transfers as follows

$$m \geq \left[\frac{8L(p!!)^2}{(1+\ln 2)p(p+1)h_0} \frac{\|u(\zeta)\|_\infty}{\|u''(\eta)\|_c} \right]^{\frac{1}{p}} \tag{13}$$

We note that the truncation error for the F-kernel will be dramatically smaller for the same values of m, since the kernel decays much faster than the G-kernel when moving away from the source point. Therefore, m due to G-kernel interpolation controls the total truncation error.

2.3.2. Computational complexity

The anterpolation of the solution vector requires

$$K_a = 2(p+2)N_0\left(1 - \frac{1}{2^L}\right) \tag{14}$$

work units, where N_0 represents the number of finest-level points and each work unit comprises one multiplication and one addition. Meanwhile, evaluation of multi-integrals using Eq. (7) needs

$$K_w^{(L)} = \frac{N_0^2}{4^L} + \frac{mLN_0}{2^L} \tag{15}$$

work units. The complexity of using Eq. (8) to evaluate $w_i^{(k)}$ on level $k = L - k'$ requires

$$K_w^{(k)} = \frac{N_0}{2^{L-k'+1}}(2(p+1) + m + m(L-k')) \tag{16}$$

units. Thus, the total complexity of the algorithm can be given by

$$K_T = \frac{N_0}{2^{L-k'+1}}(2(p+1) + m + m(L-k')) \tag{17}$$

Notice that for $p = \text{const}$, the complexity of the multi-level algorithm appears of the order $O(N^{1+1/p})$. However, when the order of interpolation increases with the mesh refinement, we can show that the optimal complexity of the algorithm is $O(N \ln N)$.

Since the expression in Eq. [11]

$$(p!!)^2 \cong 0.32(p+1)[0.35(p+1)]^{p+1} \tag{18}$$

holds approximately for $p > 3$, we substitute Eq. (18) into Eqs (13) and (16) to minimize K_T with respect to p. This procedure leads to

$$0.7(C\,N_0)^{1/p}\left[1 - \frac{1}{p}\ln(C\,N_0)\right] = -4 \tag{19}$$

Substituting $\alpha = (CN_0)^{1/p}$ into Eq. (19) and solving it yields $\alpha \cong 6.5$. Therefore, $p \sim \text{In } N_0$ and $m \sim \ln N_0$. Finally, the optimal complexity will be of the order $O(N_0 \ln N_0)$.

2.4. Multi-level multi-integration for matrix-transpose-vector multiplication

A block-matrix-transpose-vector multiplication on the finest-level mesh M^0 can be given by the following multi-summation over each patch:

$$w_j^{(0)} = h_0 \sum_{i=0}^{N_0} H_{i,j}^{(0)} u_i^{(0)} - R_j \qquad \text{for} \qquad j = 0, 1, 2, \ldots, N_0 \tag{20}$$

Notice that the summation in Eq. (20) is now performed over index i representing the collocation point ξ_i. Again, the multi-integral in Eq. (20) can be transferred to the coarsest level as follows:

$$w_j^{(L)} \cong h_L \sum_{i=0}^{N_L} H_{i,j}^{(L)} u_i^{(L)} + C_j^{(L)} \tag{21}$$

The multi-integrals $w_j^{(k)}$ on the finer-level mesh points can be represented as follows:

$$w_j^{(k)} \cong \left[\tilde{\mathcal{I}}_{k+1}^k w^{(k+1)}\right]_j + h_k \sum_{|i-j| \leq m}^{N_k} \left(H_{i,j}^{(k)} - \tilde{H}_{i,j}^{(k)}\right) u_i^{(k)} + C_j^{(k)} - \tilde{C}_j^{(k)} \tag{22}$$

Again, we pre-evaluate $\tilde{C}_j^{(k)}$ and the centered correction stencils to dramatically reduce the complexity of the computations similar to the fast matrix-vector operation described above.

2.5. C-cycle multi-grid

Since the bi-conjugate gradient method [18] used in this work requires $O(N^{1/2})$ iterations on the boundary element mesh M^0 for mixed boundary value problems, we utilize the multi-grid technique proposed by Brandt [19] to accelerate the convergence. Although several multi-grid algorithms are readily available, we restrict our consideration to the C-cycle multi-grid [19], as it provides good convergence acceleration for the problems considered in this presentation.

3. Numerical results

3.1. Introduction

The MLBEM code developed earlier for the fast solution of heat diffusion [10] has been extended to accommodate the modifications outlined above. All runs have been performed on a Sun Ultra-Enterprise workstation with quadruple UltraSparc-II 336-MHz processors. However, no parallelization is utilized. In

Table 1
CPU requirements and the number of correction points m inside the singularity zone: Problem 1, $\lambda = 10^4\pi$, $\sim$ implies estimated values

	Direct		MLBEM		m	
N_0	$\|w\|$	CPU time(s)	$\|w\|$	CPU time(s)	Eq. (13)	MLBEM
512	6.00e–4	4.33	6.00e–4	0.10	4	2
1024	6.68e–4	17.3	6.69e–4	0.13	4	2
2048	5.57e–5	69.3	5.58e–5	0.18	4	4
4096	6.44e–5	275	6.45e–5	0.26	4	4
8192	3.47e–6	1.10e3	3.79e–6	0.41	4	4
16 384	3.12e–6	4.42e3	3.38e–6	0.68	4	4
32 768	1.12e–6	1.77e4	1.12e–6	1.61	6	6
65 536	3.04e–7	7.07e4	3.04e–7	3.01	6	6
131 072	–	~2.8e5	8.09e–8	6.88	8	8
262 144	–	~1.1e6	2.11e–8	13.2	8	8
524 288	–	~4.5e6	5.35e–9	29.9	10	10
1 048 578	–	~1.8e7	1.28e–9	65.6	12	12
2 097 152	–	~7.2e7	3.33e–10	130	12	12

this paper, we consider three example problems that possess analytical solutions. We first consider an example problem of line multi-integration involving the log-singular kernel. The next example is formulated for a unit square with homogeneous boundary conditions leading to a Neumann problem. The global matrix is indeed well conditioned for the integral equation of the second kind, and thus we obtain converged solutions in just a few iterations for all boundary element meshes considered in this example. Finally, for the third example problem involving mixed boundary conditions, multi-grid iterations are mandatory to accelerate the convergence.

3.2. Problem 1

We evaluate the following multi-integral

$$w(\xi) = \int_0^1 T(x) g(x-\xi) dx \qquad \text{for} \qquad \xi \in [0,1] \tag{23}$$

involving the log-singular kernel $g(x - \xi)$. For the oscillatory function $u(x) = \cos(\lambda x)$, the multi-integral in Eq. (23) permits a closed form expression, which is used to evaluate the numerical solution error

$$\|w\| = \max|w_{\text{num}} - w_{\text{ex}}| \qquad \text{for} \qquad \xi \in [0,1] \tag{24}$$

The numerical results are summarized in Table 1. Here, we present MLBEM solutions and compare them with the direct multi-integration. Note that the latter operation is extremely time-consuming for large N_0, and thus only relatively small numbers of degrees of freedom are presented in Table 1 for the direct evaluation. Meanwhile, the MLBEM formulation allows extremely fast and accurate evaluations of $w(\xi)$ for N_0 well beyond a million. The run-times increase at an approximate rate of 1.09 for large numbers of degrees of freedom. We note an excellent agreement between MLBEM and predicted (Eq. 13) values of m.

3.3. Problem 2

Next, let us consider steady-state heat diffusion in a unit square with four boundary patches. We utilize an equal number of boundary elements for each boundary patch. Consider an oscillatory solution of the form:

$$T_{\text{ex}}(x_1, x_2) = \frac{\cos(\lambda x_1)\cosh(\lambda x_2)}{\cosh(\lambda)} \tag{25}$$

satisfying the governing heat diffusion in Eq. (1). We specify normal heat fluxes on all boundaries and look for boundary temperatures at any boundary collocation node. For the Neumann boundary value problem, we introduce an L_∞ error norm $\|T\|$ to monitor the numerical solution error

$$\|T\| = \max|T_{\text{num}} - T_{\text{ex}}| \qquad \text{for} \qquad \xi^{(m)} \in \Gamma_T \tag{26}$$

Since the global matrix is well-conditioned for this problem, the convergence of the bi-conjugate gradient method (BCGM) is virtually independent of the mesh size. Therefore, no multi-grid acceleration is needed for the integral equation of the second kind. Table 2 summarizes the comparisons between the conventional and fast boundary element methods for this problem. Notice that the conventional iterative algorithm prohibits solutions with more than 20 000 degrees of freedom due to memory restrictions. Meanwhile, the MLBEM formulation permits accurate solutions for more than four million degrees of freedom. We emphasize that these solutions are obtained on a moderate desktop workstation in just under an hour.

Table 2
Numerical errors and computer requirements for conventional and fast BEM: Problem 2, unit square, $\lambda = 1875\pi$, $\sim$ implies estimated values

	$\|T\|$		CPU time (s)		Memory (Mb)	
N_0	BCGM	MLBEM	BCGM	MLBEM	BCGM	MLBEM
8196	0.244	0.232	1.96e3	6.02	537	1.43
16 388	0.139	0.142	8.44e3	8.84	2,148	2.49
32 772	–	4.03e–2	~3.4e4	23.1	~8.6e3	4.68
65 540	–	1.09e–2	~1.4e5	44.2	~3.4e4	8.90
131 076	–	3.27e–3	~5.4e5	82.4	~1.4e5	17.3
262 148	–	8.65e–4	~2.2e6	170	~5.5e5	35.0
524 292	–	2.07e–4	~8.6e6	399	~2.2e6	68.7
1 048 580	–	6.31e–5	~3.5e7	845	~8.8e6	135
2 097 156	–	1.87e–5	~1.4e8	1732	~3.5e7	270
4 194 308	–	4.48e–6	~5.5e8	3091	~1.4e8	542

Table 3
Computer requirements for conventional and fast BEM: Problem 3, unit square, $\lambda = 1000\pi$, $\sim$ implies estimated values

	CPU time (s)		Memory (Mb)	
N_0	Direct	MLBEM	Direct	MLBEM
65 540	~1.2e7	641	~3.4e4	15.4
131 076	~9.2e7	2.47e3	~1.4e5	34.2
262 148	~7.4e8	6.94e3	~5.5e5	59.6
524 292	~5.9e9	1.81e4	~2.2e6	135
1 048 580	~4.7e10	4.61e4	~8.8e6	294

3.4. Problem 3

Finally, we re-visit Problem 2 and utilize mixed boundary conditions. Namely, we specify temperatures on vertical boundaries and normal heat fluxes elsewhere. This problem leads to an ill-conditioned matrix, and, thus, we use a C-cycle multi-grid to accelerate the convergence. We note that the converged solutions are then obtained in just a few cycles. Table 3 presents comparisons between MLBEM and direct BEM solutions using Gauss elimination. For values of N_0 presented in Table 3, no direct solutions were possible. Meanwhile, the MLBEM algorithm permits solutions for over a million degrees of freedom.

5. Conclusions

A multi-level boundary element solver [10] has been extended to provide an extremely robust, efficient, and fast computational tool. We introduced two important modifications into the code, namely the implementation of a pre-evaluation of the total correction from the fine-level meshes and the facilitation of fast matrix-vector multiplications using a sliding correction stencil for the discrete coefficients. These amendments have led to dramatic improvements of the modified multi-level algorithm and permitted solutions on boundary element meshes that are three orders of magnitude finer than those possible using conventional boundary element methods.

In this study, we present analysis of truncation errors for the MLBEM and investigate the complexity of the algorithm. We show that the optimal complexity of the multi-level method is of the order $O(N \ln N)$. Then, we study performance of the MLBEM on three model problems. We compare the CPU time and memory requirements for the fast multi-level and conventional BEMs and demonstrate exceptional performance of the fast BEM. We have been able to model the example steady heat diffusion problem involving more than four million degrees of freedom on a workstation in less than one hour.

6. Acknowledgment

The work described in this paper was partially supported by the National Science Foundation under grant CMS 0084632. The authors gratefully acknowledge this support.

References

[1] Barnes J, Hut P. A hierarchical force calculation algorithm. Nature 1986;324:446–449.
[2] Grama A, Kumar A, Sameh A. Parallel hierarchical solvers and preconditioners for boundary element methods. SIAM Sci Comput 1998;20:337–358.

[3] Beylkin G, Coifman R, Rokhlin V. Fast wavelet transforms and numerical algorithms. Commun Pure Appl Math 1991;44:141–183.
[4] Alpert B, Beylkin G, Coifman R, Rokhlin V. Waveletlike bases for the fast solution of second-kind integral equations. SIAM J Sci Comput 1993;14:159–184.
[5] Rokhlin V. Rapid solution of integral equations of classical potential theory. J Computat Phys 1985;60:187–207.
[6] Greengard L, Rokhlin V. A fast algorithm for particle simulations. J Computat Phys 1987;73:325–348.
[7] Rahola J. Diagonal forms of the translation operators in the fast multipole algorithm for scattering problems. BIT 1996;36:333–358.
[8] Cheng H, Greengard L, Rokhlin V. A fast adaptive multipole algorithm in three dimensions. J Computat Phys 1999;155:468–498.
[9] Darve E. The fast multipole method: numerical implementation. J Computat Phys 2000;160:195–240.
[10] Grigoriev MM, Dargush GF. A multi-level boundary element method for two-dimensional steady heat diffusion. Numerical Heat Transfer Part B Fundamentals 2004;46:329–356.
[11] Brandt A, Lubrecht AA. Multilevel matrix multiplication and fast solution of integral equations. J Computat Phys 1990;90:348–370.
[12] Venner CH, Lubrecht AA. Multi-level Methods in Lubrication. Elsevier, Amsterdam, 2000.
[13] Lubrecht AA, Ioannides E. A fast solution of the dry contact problem and the associated subsurface stress field using multilevel techniques. ASME J Tribol 1991;113:128–133.
[14] Polonsky IA, Keer LM. A numerical method for solving rough contact problems based on the multi-level multi-summation and conjugate gradient techniques. Wear 1999;231:206–219.
[15] Grigoriev MM, Dargush GF. A fast multi-level boundary element method for the Helmholtz equation. Comput Meth Appl Mech Engng 2004;193:165–203.
[16] Wang CH, Grigoriev MM, Dargush GF. A fast multi-level convolution boundary element method for transient diffusion problems. Int J Numer Meth Engng (to appear).
[17] Grigoriev MM, Dargush GF. A multi-level boundary element method for Stokes flows in irregular two-dimensional domains. Comput Meth Appl Mech Engng (to appear).
[18] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical Recipes in C. The Art of Scientific Computing. Cambridge: Cambridge University Press, 1992.
[19] Brandt A. Multi-level adaptive solutions to boundary-value problems. Math Computat 1977;31:333–390.