INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2001; **52**:417–432 (DOI: 10.1002/nme.217)

A comparison of domain integral evaluation techniques for boundary element methods

Marc S. Ingber*,†, Andrea A. Mammoli and Mary J. Brown

Department of Mechanical Engineering, University of New Mexico, Albuquerque, New Mexico 87131, U.S.A.

SUMMARY

In many cases, boundary integral equations contain a domain integral. This can be evaluated by discretization of the domain into domain elements. Historically, this was seen as going against the spirit of boundary element methods, and several methods were developed to avoid this discretization, notably dual and multiple reciprocity methods and particular solution methods. These involved the representation of the interior function with a set of basis functions, generally of the radial type. In this study, meshless methods (dual reciprocity and particular solution) are compared to the direct domain integration methods. The domain integrals are evaluated using traditional methods and also with multipole acceleration. It is found that the direct integration always results in better accuracy, as well as smaller computation times. In addition, the multipole method further improves on the computation times, in particular where multiple evaluations of the integral are required, as when iterative solvers are used. The additional error produced by the multipole acceleration is negligible. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: boundary element method; particular solution method; dual reciprocity method; multipole method; domain integral; meshless method

1. INTRODUCTION

The boundary element method originally was developed to analyze homogeneous linear partial differential equations for which a fundamental solution associated with the adjoint operator could either be determined or approximated. In these cases, the dimensionality of the problem could be reduced one order since the dependent variable could be represented solely by a boundary integral. In subsequent development, the boundary element method was extended to non-linear and non-homogeneous problems. However, for these problems, the boundary integral representation had to be supplemented with a domain integral.

*Correspondence to: Marc Ingber, Department of Mechanical Engineering, The University of New Mexico, Albuquerque, NM 87131, U.S.A.
†E-mail: ingber@me.unm.edu

Contract/grant sponsor: DOE, U.S.A., partially; contract/grant numbers: DE-FG03-97ER14778 and DE-FG03-97ER25332
Contract/grant sponsor: Sandia National Laboratories, partially; contract/grant number: DE-AC04-94AL85000

Received 31 July 2000
Revised 16 November 2000

The introduction of a domain integral in boundary element methods created additional complexity and CPU costs. Typical approaches for evaluating the domain integral involved discretizing the interior of the domain, approximating the non-homogeneous functions in the interior of the domain using Lagrangian interpolants, and performing numerical quadrature. Although this procedure was relatively robust, the additional discretization devalued one of the main attractions of using a boundary element method, namely, that only the boundary of the domain required a discretization.

Several methods for eliminating domain integrals associated with boundary element methods have been developed over the years. Nardini and Brebbia [1] proposed a generalization of the concept of particular integrals, which they interpreted as a localized particular solution approach in order to eliminate the domain integral in a free vibration problem. The method, which they called the dual reciprocity method (DRM), was extended to parabolic problems by Wrobel and Brebbia [2, 3]. Partridge *et al.* [4] wrote a book on the DRM detailing the use of the method for a wide variety of problems. The DRM has probably been the most popular of the methods to deal with domain integrals in boundary element methods as evidenced by the sheer mass of journal and conference papers written about the method.

Nowak and Brebbia [5, 6] developed a technique for analysis of transient heat transfer problems in which the domain integral was replaced by an infinite series of boundary integrals involving higher order fundamental solutions. They showed in their method which they called the multiple reciprocity method (MRM), that this series of integrals converged quickly and could be evaluated efficiently. Yeih *et al.* [7] used the MRM to study natural modes of an Euler–Bernoulli beam and Kamiya and Andoh [8] used the MRM for the Helmholtz equation. Power [9] extended the MRM to non-permanent Stokes flow problems.

Ahmad and Banerjee [10] used a closed form representation of a particular solution, again to eliminate a domain integral in a free vibration problem. Herry and Banerjee [11, 12] extended this work to several problems in elasticity by developing a technique to determine approximate particular solutions. Zheng *et al.* [13] solved a variety of non-homogeneous potential problems again by developing an approximate particular solution. Ingber and Phan-Thien [14] used a similar method to solve transient heat conduction problems.

Both the particular solution and the dual reciprocity methods require approximating a function in the interior of the domain with the use of radial basis functions. Further, a particular solution is required of the non-homogeneous adjoint operator set equal to the radial basis function. There has been much effort in recent years in developing the approximation theory using radial basis functions [15, 16]. Nevertheless, for interior functional distributions with steep gradients, it is often difficult to obtain good interior approximations. Further, the solution may be sensitive to the location of the radial basis functions [17]. In other cases, the particular solution may be ill-behaved causing further numerical problems [18].

Fast multipole methods have recently been applied to evaluate boundary integral equations. Applications of fast multipole methods for boundary element analysis have been presented by Korsmeyer [19] to study 3D potential problems, Allen [20] for 2D potential point-collocation and Galerkin BEMs, Gómez and Power [21] for Stokes flow problems, and Mammoli and Ingber [22] for suspension problems. In all of these applications, the use of the multipole methods were limited strictly to boundary integrals and not to any associated domain integrals.

In the current paper, four methods for evaluating domain integrals occurring in boundary element methods are compared. These methods include domain integration using Gaussian quadrature, domain integration using a fast multipole method, the dual reciprocity method,

and the particular solution method. The methods are compared in terms of ease of use, accuracy, robustness, and CPU requirements.

2. NUMERICAL FORMULATIONS

Two governing differential equations will be considered in this research, namely,

$$\nabla^2 u(x, y) = b(x, y) \tag{1}$$

$$\nabla^2 u(x, y) = h(u, x, y) \tag{2}$$

The particular form of the second differential equation (Equation (2)) is given by the non-homogeneous Helmholtz equation. The right hand side of this equation could be written in terms of only the independent variables x and y, and by using the fundamental solution associated with the Helmholtz operator, the development would be almost identical to the development for the Poisson equation (Equation (1)). The purpose here is to show that the methodologies discussed in this research can be applied in more general cases where there is no known fundamental solution.

2.1. Poisson equation

The boundary integral representation for the Poisson equation is given by:

$$c(\xi)u(\xi) + \int_\Gamma q^\star(\xi, \eta)u(\eta)\,\mathrm{d}\Gamma + \int_\Omega u^\star(\xi, \eta)b(\eta)\,\mathrm{d}\Omega = \int_\Gamma u^\star(\xi, \eta)q(\eta)\,\mathrm{d}\Gamma \tag{3}$$

where Γ is the boundary of the domain Ω. The benchmark problem considered for the Poisson equation has as its exact solution

$$u(x, y) = y^3 \exp(x) \tag{4}$$

and hence,

$$b(x, y) = (y^3 + 6y)\exp(x) \tag{5}$$

The problem is solved on the rectangular domain Ω given by $0 \leqslant x \leqslant 2$ and $0 \leqslant y \leqslant 1$. Unless otherwise stated, Dirichlet boundary conditions are specified on the side $x = 2$ and Neumann conditions are specified on the three remaining sides.

2.2. Non-homogeneous Helmholtz equation

A more complex problem is encountered in the case of the non-homogeneous Helmholtz equation, where the boundary integral equation becomes:

$$c(\xi)u(\xi) + \int_\Gamma q^\star(\xi, \eta)u(\eta)\,\mathrm{d}\Gamma + \int_\Omega u^\star(\xi, \eta)h[\eta, u(\eta)]\,\mathrm{d}\Omega = \int_\Gamma u^\star(\xi, \eta)q(\eta)\,\mathrm{d}\Gamma \tag{6}$$

The exact solution of the test problem chosen here is

$$u = 3x^3 y + 2x^2 y^2 - xy^3 \tag{7}$$

so that the right hand side of Equation (2) is given by

$$h(u,x,y) = 4x^2 + 4y^2 + 12xy + 3x^3y + 2x^2y^2 - xy^3 - u \tag{8}$$

For both the Poisson and non-homogeneous Helmholtz problems, the boundary integrals are evaluated by subdividing the boundary into boundary elements and approximating the source densities within the elements using linear Lagrangian interpolation functions. Since the focus of this research is on the evaluation of the domain integrals, all boundary integrals are evaluated analytically to remove quadrature error from those integrals. As discussed above, four methods for either evaluating or eliminating the domain integral in Equations (3) and (6) are considered in this research. A brief description of all four methods is given below.

2.3. *Classical domain integration*

The classical domain integration approach for the Poisson problem requires the direct evaluation of the domain integral in Equation (3) by subdividing the interior of the domain into quadrilateral finite elements and approximating $b(\eta)$ within the interior finite elements using isoparametric shape functions. Numerical quadrature is performed here using a 6×6 Gauss rule, which results in a sufficient level of accuracy. The weakly singular domain integrals that result when η approaches ξ are regularized by transforming from Cartesian to polar coordinates in the neighborhood of ξ. In this case, the domain integral simply adds to the load vector, and the system of equations that results from the discretization of Equation (3) applied to all collocation points is of the form:

$$\mathsf{A}\mathbf{x} = \mathbf{b} + \mathbf{f} \tag{9}$$

where A is a coefficient matrix, $\mathbf{b}$ is the vector that results from known boundary values of either u or q, and $\mathbf{f}$ is a vector that results from the domain integral. The solution of the system only requires the inversion of A. There are two difficulties with this approach: it requires a complete interior discretization, and the numerical evaluation of the domain integral. The difficulty of the discretization is a function of the complexity of the geometry. Meshing algorithms have progressed considerably in recent years, and most geometries can be meshed quite efficiently using unstructured meshes. In this paper, the latter concern is addressed. Using standard numerical quadrature, the operation count required to evaluate the vector $\mathbf{f}$ scales as $N \times (M + N)$, where N is the number of boundary collocation nodes and M is the number of interior nodes. As the problem size increases, the operation count for the domain integration begins to dominate.

In the case of the non-homogeneous Helmoltz equation, the values of u at the interior points are unknown, and the additional load vector $\mathbf{f}$ cannot be evaluated. One can now proceed in two directions:

1. *By iteration*: the vector $\mathbf{f}$ can be evaluated with an initial guess for u, allowing an approximate boundary solution to be found. Based on the approximate solution, new values for u can be found, leading to an improved estimate of $\mathbf{f}$. The iteration is repeated until convergence. The advantage of this method is speed and versatility, because potentially any function of u can be treated. The disadvantage is that convergence is not guaranteed.
2. *By collocation*: the values of u at the interior finite element nodes are introduced as additional unknowns. A linear system of equations are generated by collocating the BIE

at both the boundary element nodes and the interior finite element nodes:

$$\begin{bmatrix} \mathsf{A} & \mathsf{C} \\ \mathsf{B} & \mathsf{D} \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \end{pmatrix} \tag{10}$$

where the matrices A and C result from collocation points on the boundary and integration on the boundary and in the interior respectively, while the matrices B and D result from collocation points in the interior and integration on the boundary and in the interior respectively. The vectors $\mathbf{x}$ and $\mathbf{z}$ are the boundary unknowns and the domain values of u, respectively. The solution of the system yields the boundary solution and the domain values of u simultaneously. The advantage of this method over the iterative method is that a solution can always be found. The disadvantages are that the linear system to be solved is considerably larger, and that it is not simple to treat non-linear functions of u.

The second technique is investigated here, because the aim of this paper is to perform a fair comparison of the different approaches. The first method will be the subject of subsequent research.

2.4. *Multipole evaluation of the domain integral*

The multipole evaluation of the domain integral is essentially equivalent to the classical domain integration. However, in the present approach using the Barnes–Hut algorithm [23] for the evaluation of the domain integral, the operation count per collocation node is reduced from $O(M)$ for classical Gaussian quadrature to $O(\log M)$ for multipole evaluation. A consequence of the multipole evaluation of the domain integral is that, in the case of the Helmholtz equation, the matrices that result from the integration over the interior domain (C and D) are not stored explicitly, that is, the individual coefficients are not known. This precludes a direct evaluation of the solution to the linear system, and in fact one is limited to using solvers where only the forward matrix–vector product is available. Because of its robustness, a generalized minimal residual (GMRES) algorithm [24] is used here. Note that the largest matrix in the system, D, is diagonally dominant, leading to a well-conditioned linear system. Although this is not implemented here, further improvements to the matrix condition number could be obtained by left preconditioning with the matrix P_l, given by:

$$P_l = \begin{bmatrix} \mathsf{A}^{-1} & \mathsf{0} \\ \mathsf{0} & \mathsf{I} \end{bmatrix} \tag{11}$$

where I is an identity matrix of size M; this would lead to a fully diagonally dominant system.

The details of the Barnes–Hut multipole expansion have been explained in detail elsewhere, and will not be repeated at length here. The application of the multipole methods to domain integrals, as opposed to boundary integrals, is straightforward. However, a short explanation is given in the interest of clarity. Using a Taylor series expansion, the fundamental solution can be written as [21, 22]

$$\begin{aligned} u^{\star}(\xi,\eta) &= \sum_{q=0}^{\infty} \frac{\partial^q}{\partial\xi_{k_1}\partial\xi_{k_2}\ldots\partial\xi_{k_q}} u^{\star}(\xi,\eta_0) r_{k_1} r_{k_2} \ldots r_{k_q} \\ &= \sum_{q=0}^{n} \frac{\partial^q}{\partial\xi_{k_1}\partial\xi_{k_2}\ldots\partial\xi_{k_q}} u^{\star}(\xi,\eta_0) r_{k_1} r_{k_2} \ldots r_{k_q} + E_n(\xi,\varepsilon) \end{aligned} \tag{12}$$

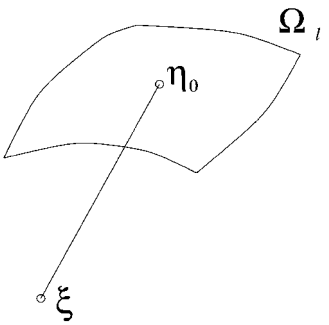

Figure 1. Subdomain Ω_l where the Taylor expansion of the kernel function around η_0 is integrated.

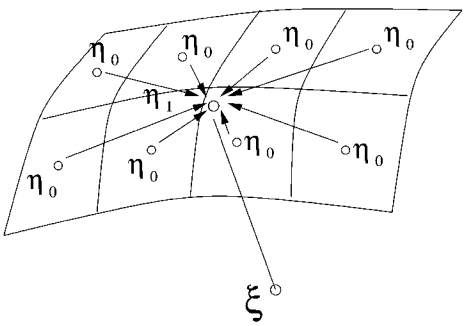

Figure 2. Schematic of moment tensor shift from a number of subdomain points η_0 to a common point η_1.

where $E_n(\xi, \varepsilon)$ is the error due to the truncation of the series to n terms, and r_i is the ith component of the distance between η and η_0.

The integral over a subdomain Ω_l, as shown in Figure 1, can be approximated by

$$\int_{\Omega_l} u^{\star}(\xi,\eta) h(\eta,u)\,\mathrm{d}\Omega \approx \sum_{q=0}^{n} \frac{\partial^q}{\partial\xi_{k_1}\partial\xi_{k_2}\ldots\partial\xi_{k_q}} u^{\star}(\boldsymbol{\xi},\eta_0) C^l_{k_1 k_2 \ldots k_q} \tag{13}$$

where the coefficients $C_{k_1 k_2 \ldots k_q}$ are moment tensors given by

$$C^l_{k_1 k_2 \ldots k_q} = \int_{\Omega_l} r_{k_1} r_{k_2} \ldots r_{k_q} h(\eta,u)\,\mathrm{d}\Omega \tag{14}$$

The variation of $h(\eta, u)$ is accounted for by appropriate interpolation functions, quadratic isoparametric in this case.

It is important to note that the moment tensors are solely a function of the local geometry, and not of the location of ξ, and that the shape of Ω_l is completely arbitrary. The integral over Ω_l can be evaluated by multiplying the moment tensor for the subdomain with the appropriate derivatives, evaluated for the pair of points (ξ, η_0).

The power of the multipole method comes from being able to economically calculate moment tensors about a point η_1 by modifying the moment tensors about point η_0, without the need to re-evaluate the integrals. By 'shifting' the moment tensors from the local point η_0 for each subdomain to a common point η_1, the integral from a number of subdomains can now be evaluated with a single tensor/derivative multiplication, as shown in Figure 2. The point η_1 is now the common point for a cluster of subdomains. The same procedure can be used to shift this point to an even higher level point, representing a cluster of clusters, and so on. The size of such clusters is limited by the size of the truncation error E_n, which is a function of the separation of the cluster from the collocation point ξ. Far away from ξ, larger clusters can be constructed. The formulae used to shift coefficients from one point to another are given in the recent work by Mammoli and Ingber [22] and Gómez and Power [21].

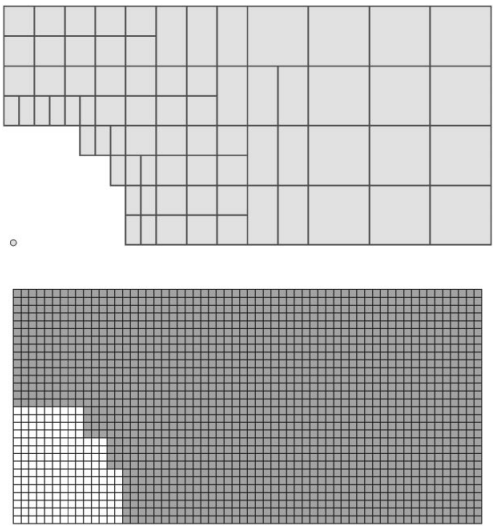

Figure 3. Clustering for the domain integral for the node highlighted by the dot. Notice that the number of clusters (light grey) is far smaller than the number of far-field elements that they replace (dark grey). Matrix entries are stored only for the near-field elements (white).

The domain integral is evaluated in two parts, a near field and a far field. In the near field, the classical domain integration is performed. In the far field, multipole expansions are used. The overall integration strategy can be summarized by

$$\int_{\Omega} u^{\star}(\xi,\eta)h(\eta,u)\,\mathrm{d}\Gamma_y = \sum_{i\in \mathrm{nf}} \int_{\Omega} u^{\star}(\xi,\eta)h(\eta,u)\,\mathrm{d}\Gamma + \sum_{j\in \mathrm{ff}} \sum_{q=0}^{n} C^{j}_{k_1 k_2 \ldots k_q} \frac{\partial^q}{\partial\xi_{k_1}\partial\xi_{k_2}\ldots\partial\xi_{k_q}} u^{\star}(\xi,\eta_j) \tag{15}$$

Each collocation node is associated with a list of near-field elements (nf) and a list of far-field (ff) clusters. The number of near-field elements in the list remains approximately constant as the problem size increases, while the number of far-field clusters grows as $\log M$, resulting in an $O(\log M)$ scaling overall. The procedure for the generation of clusters and cluster lists is well known and will not be repeated here. However, it is important to note that care must be taken in order to make the generation and storage of near-field and far-field element and cluster lists economical. The advantage of the multipole expansion can be severely curtailed if lists are generated and stored inefficiently. An example of a near-field element list and a far-field cluster list is shown in Figure 3.

2.5. Dual reciprocity method

The formulation implemented in this research for the dual reciprocity method is essentially the same as given by Partrtidge *et al.* [4] for the Poisson problem, but differs slightly for the case where the non-homogeneous term ($h(x, y, u)$ in Equation (2) is a function of the dependent

variable u. The formulation for the non-homogeneous Helmholtz equation is presented here with some comments on the simplifications afforded by the Poisson equation.

First, the non-homogeneous function $h(x, y, u)$ is approximated using radial basis functions. That is,

$$h(x, y, u) \approx \sum_{i=1}^{N} \alpha_i f_i \tag{16}$$

where N is the number of radial basis functions and the α_i's are coefficients to be determined and the particular radial basis functions chosen here are given by

$$f_i(x, y) = 1 + r_i \tag{17}$$

where r_i is the distance between the point (x, y) and the centre of the radial basis function (x_i, y_i). Although other radial basis functions could have been chosen, the present choice is rather common and of general applicability.

Particular solutions $\hat{u}_i$ are available so that

$$\nabla^2 \hat{u}_i = f_i \tag{18}$$

In particular, for the radial basis functions given above

$$\hat{u}_i = r_i^2/4 + r_i^3/9 \tag{19}$$

Now, using a second reciprocity on the domain integral in Equation (3), the boundary integral equation can be rewritten as

$$c(\xi)u(\xi) = \int_\Gamma [q^\star(\xi, \eta)u(\eta) - u^\star(\xi, \eta)q(\eta)]\, \mathrm{d}\Gamma(\eta) + \sum_{i=1}^{l} \alpha_i \left\{ \eta(\xi)\hat{u}_i(\xi) + \int_\Gamma [q^\star(\xi, \eta)\hat{u}_i(\eta) - u^\star(\xi, \eta)\hat{q}_i(\eta)]\, \mathrm{d}\Gamma(\eta) \right\} \tag{20}$$

where $\hat{q}_i = \partial \hat{u}_i / \partial n$.

For the Poisson problem, the left hand side of Equation (16) is known, and hence, the α_is can be determined in a variety of ways. In the present work, the α_is are determined by collocating Equation (16) at each radial basis function center (x_i, y_i), and then solving the dense system of equations. For the 2×1 domain considered, the (x_i, y_i)'s are chosen on a regular set of grid points which include all the boundary element nodes.

For the more general case where h is a function of the dependent variable u, the α_is must be determined as part of the overall solution algorithm. Partridge *et al.* [4] eliminate the α_is by essentially inverting the coefficient matrix associated with Equation (16) obtained through point collocation. In the present implementation, rather than eliminating the α_is, they are solved for along with the boundary unknowns. However, in order to do this, the centers of the radial basis functions can no longer coincide with boundary element nodes. That is, the (x_i, y_i)s are all chosen in the interior of the domain. After discretization, Equation (20) is collocated at each boundary element node and each radial basis function center to determine the linear equation set to solve for the α_is and the unknown boundary data.

2.6. Particular solution method

The particular solution method starts similarly to the dual reciprocity method by approximating the non-homogeneous terms in the governing equation using radial basis functions as given in Equation (16). The solution u is divided into a general solution, u^g, and a particular solution, u^p. That is,

$$u = u^g + u^p \tag{21}$$

An approximate particular solution is given by

$$u^p \approx \sum_{i=1}^{n} \alpha_i \hat{u}_i \tag{22}$$

Now, rather than using an additional reciprocity to eliminate the domain term in Equations (3) or (6), the particular solution is used to determine the appropriate boundary conditions for the homogeneous solution u^g. Further, since the governing equation for the general solution is simply the Laplace equation, the boundary integral representation for the general solution does not contain a domain integral. In particular, the boundary conditions for the general solution u^g are given by

$$u^g(x, y) = u(x, y) - u^p(x, y) \quad \text{for } (x, y) \in \Gamma_1 \tag{23}$$

$$q^g(x, y) = q(x, y) - q^p(x, y) \quad \text{for } (x, y) \in \Gamma_2 \tag{24}$$

where Γ_1 is the portion of the boundary where Dirichlet conditions are specified, Γ_2 is the portion of the boundary where Neumann conditions are specified, and q^p is determined by taking the normal derivative of Equation (22). In a similar fashion, Robin conditions can also be accommodated but, for simplicity, are not considered here.

Collocating the boundary integral equation for u^g at each boundary element node yields a linear system which can be written in matrix form as

$$[A_{ij}]\{u_j^g\} = [B_{ij}]\{q_j^g\} \tag{25}$$

Inserting Equations (23) and (24) into Equation (25) yields

$$[A_{ij}]\{u_j\} - [A_{ij}][\phi_{jk}]\{\alpha_k\} = [B_{ij}]\{q_j\} - [B_{ij}][\phi'_{jk}]\{\alpha_k\} \tag{26}$$

where ϕ_{jk} and ϕ'_{jk} represent the value of $\hat{u}_k$ and $\partial \hat{u}_k / \partial n$, respectively, at the centre of the jth radial basis function.

For the Poisson problem, the α_is can be determined in the same manner as discussed for the dual reciprocity method, and Equation (26) can be solved for the boundary unknowns. For the case where the non-homogeneous term is a function of the dependent variable u, the α_is must again be determined as part of the overall problem solution. In these cases, if there are N boundary element nodes and M radial basis functions, then Equation (26) represents N equations in $N + M$ unknowns. To close the algebraic system, the BIE (Equation (25)) is also collocated at the centres of the N radial basis functions yielding in matrix form

$$\{\eta_i u_i^o - \eta_i u_i^{op}\} + [C_{ij}]\{u_j\} - [C_{ij}][\phi_{jk}]\{\alpha_k\} = [D_{ij}]\{q_j\} - [D_{ij}][\phi'_{jk}]\{\alpha_k\} \tag{27}$$

where u_i^o and u_i^{op} represent the values of u and u^p at the centre of the ith radial basis function. These values of u and u^p can be written in terms of the α_is using the radial basis functions

f_i and the associated particular solutions $\hat{u}_i$. Hence, Equations (26) and (27) represent the system of linear equations to solve for the unknown boundary data and the unknown α_is.

3. NUMERICAL RESULTS

3.1. Poisson equation

In the case of the Poisson equation, the value of the function b in Equation (3) is known. As a result, the domain integral is only evaluated once. The number of degrees of freedom for the classical domain integration method is given by the number of boundary nodes. On the other hand, in the case of the dual reciprocity solution formulations, the α_i coefficients are evaluated by collocation at all radial basis function centres, requiring the solution of a large dense linear system, prior to evaluation of the boundary solution.

The results of interest in this section are the total time required to obtain the boundary solution, and the L_2 norm error, defined as

$$L_2 = \left[\int_\Omega (u - u_{\text{exact}})^2 \, \mathrm{d}\Omega \right]^{0.5} \tag{28}$$

The solution times are shown in Figure 4. The solution times with the classical domain integration method (standard and multipole accelerated) is comparable to the solution times for the dual reciprocity method and the particular solution method for small problem sizes. At larger problem sizes, the inversion of the large matrix produced by collocation at the internal points dominates the solution time for the dual reciprocity and particular solution methods. On a Compaq Alpha workstation with 640MB of RAM, the largest problems solved with classical domain integration could not be solved using the dual reciprocity and particular solution methods because of the amount of memory required to store the coefficient matrix used to determine the α_i's.

In the case of the direct domain integration, the multipole accelerated solution times are significantly smaller than with classical element by element integration, except for very small problems. Crossover occurs at approximately 100 boundary degrees of freedom. The times shown here are the total times required to arrive at a solution. However, the operations performed to arrive at a solution are different. In the case of the standard domain integration method, the majority of the time is spent in evaluating the domain integral. In the case of multipole acceleration, approximately half the time is spent evaluating the domain integral. The remainder of the time is spent in recursive domain subdivision and far-field and near-field list generation.

The L_2-norm error for the four domain integration methods is shown in Figure 5. The smallest errors are given by the domain integration methods. The multipole approximation does not introduce noticeable additional error in the results. Indeed, for largest problem size considered here, the multipole method results in superior accuracy compared to the direct domain integration technique. This counterintuitive result can be explained by the fact that some loss of accuracy results from a combination of single precision arithmetic and the very large number of calculations performed with the classical direct integration method. With the multipole method, the integral for large numbers of clustered elements is combined analytically, reducing roundoff error. For small problem sizes, the dual reciprocity and particular

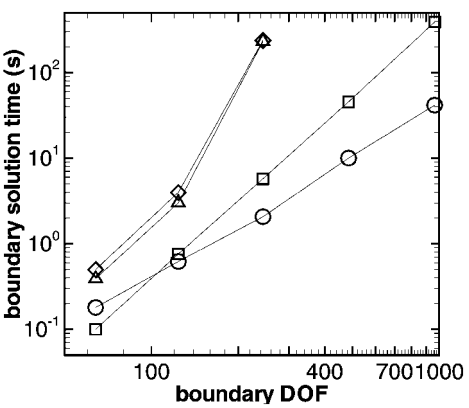

Figure 4. Boundary solution time as a function of problem size, for the dual reciprocity method (♢), the particular solution method (△), the regular domain integration method (□) and the domain integration method with multipole acceleration (◯).

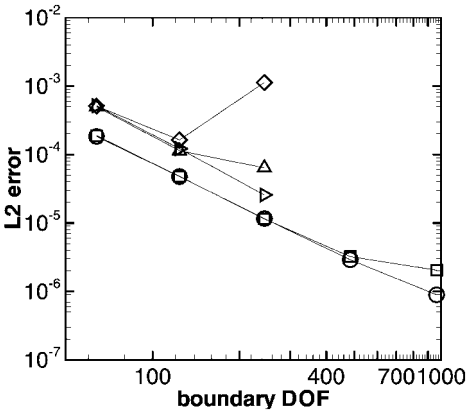

Figure 5. Error as a function of problem size, for the dual reciprocity method (♢), the dual reciprocity method with double precision arithmetic (▷), the particular solution method (△), the regular domain integration method (□) and the domain integration method with multipole acceleration (◯).

solution methods produce a similar error, between 2 and 3 times larger than with classical domain integration. For larger problem sizes, both the dual reciprocity method and the particular solution method produce large error when single precision arithmetic is used. In the case of the dual reciprocity method, the error for 244 boundary nodes is larger than for 64 nodes. This is due to the large condition number of the matrix used to determine the α_i's. When double precision arithmetic is used with the dual reciprocity method, the trend in the error vs DOF plot is the same as with the classical domain integration.

3.2. Non-homogeneous Helmholtz equation

As in the case of the Poisson equation, two issues are of importance, namely speed and accuracy. For both the particular solution method and the dual reciprocity method, fully populated linear systems of size $(M + N)\times(M + N)$ are generated. Potentially, these can be solved directly or iteratively. Unfortunately, the condition number of the matrices precludes effective iterative solution, and direct solution was used to solve the systems. It must be noted that the operation count is then proportional to the third power of the size of the system, and that very large systems cannot be solved efficiently.

For the case of regular direct integration, coefficient matrices can be formed, and the linear systems can be solved directly. The diagonal dominance of the matrix results in a small condition number, and the system is amenable to iterative solution. When multipole acceleration is used, individual matrix coefficients are not known, and certain types of iterative solution must be used. In order to directly compare the standard and multipole accelerated direct integral evaluation techniques, this class of problems was solved iteratively, using GMRES.

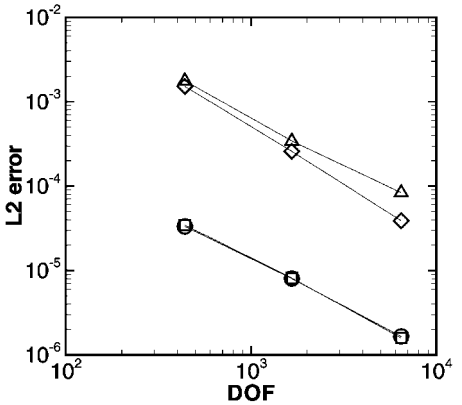

Figure 6. Error as a function of problem size, for the dual reciprocity method (◇), the particular solution method (△), the regular domain integration method (□) and the domain integration method with multipole acceleration (◯).

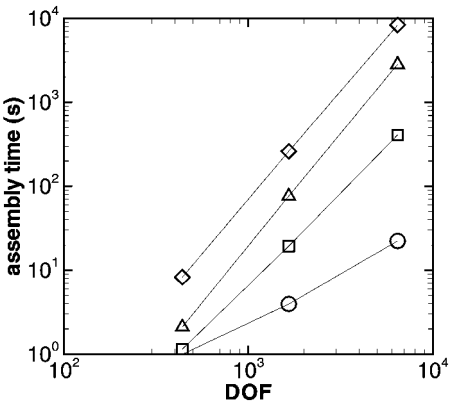

Figure 7. Assembly time as a function of problem size, for the dual reciprocity method (◇), the particular solution method (△), the regular domain integration method (□) and the domain integration method with multipole acceleration (◯). The codes were run on a Compaq Alpha XP1000.

The accuracy of the method is described by an L_2-norm, defined in Equation (28). The L_2-norm error for systems with degrees of freedom spanning more than a decade are shown in Figure 6. The dual reciprocity and particular solution methods display similar levels of accuracy, with the former being slightly more accurate, while the direct integration methods (regular and multipole accelerated) are almost two orders of magnitude more accurate. The fact that errors resulting from regular integration and multipole accelerated integration are virtually identical indicates that the multipole approximation is an extremely accurate one.

The assembly time for the different methods is shown in Figure 7. Assembly time is highest for the dual reciprocity method. The assembly time for the particular solution method is approximately a factor of 4 smaller compared to the dual reciprocity method, for all problem sizes. The regular direct integration assembly time is smaller and increases more slowly than for the previous methods. As could be expected, the assembly time with multipole acceleration is the smallest, more than two orders of magnitude lower than with the dual reciprocity method. The theoretical $O(N)$ scaling is not visible in this plot because these times include not only the near-field integration, but also recursive subdivision times and cluster list times.

Finally, the solution times are compared in Figure 8. The dual reciprocity and particular solution methods are solved in equal times. This is as expected, because a direct solution method is used to solve linear systems of equal size. The slope of the curve shows the $O(N^3)$ scaling characteristic of direct solution methods. The solution time for the regular direct integration method is larger in all cases because of the iterative solution method. Extrapolation of the curve indicates that it would become advantageous at approximately 10 000 degrees of freedom. Of course, since in this case all the matrix coefficients are stored, the linear system could be solved directly for smaller problems. Because the matrix size is the same as with the dual reciprocity and particular solution methods, the solution times would be identical. For the multipole accelerated direct domain integration, the break-even point with the dual

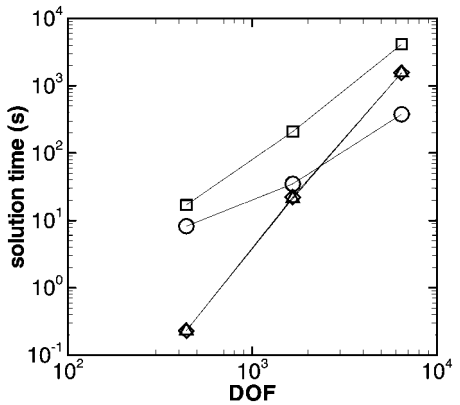

Figure 8. Solution time as a function of problem size, for the dual reciprocity method (◇), the particular solution method (△), the regular domain integration method (□) and the domain integration method with multipole acceleration (◯).

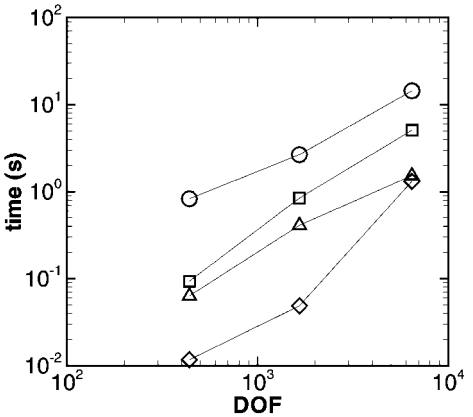

Figure 9. Breakdown of assembly time as a function of problem size for the multipole accelerated direct integration method: element moment generation (◇), recursive domain subdivision (△), list compilation (□) and near-field assembly (◯).

reciprocity/particular solution methods occurs at approximately 2000 degrees of freedom. The increasing slope of the curve is a result of the increasing number of GMRES iterations required to meet the convergence criterion. In this case, iterative solvers must be used with the multipole method for all problem sizes because the matrix coefficients are not known explicitly.

The assembly time with the multipole accelerated direct integration method reported in Figure 7 includes a number of different operations, namely the calculation of moment tensors for all domain elements, the recursive subdivision of the domain, the time taken to compile far-field cluster lists and the regular assembly time for the near-field elements. In a sense, part of the assembly actually takes place during the matrix multiplication time, when the moment tensors are multiplied by the trial solution vector and the results are shifted from the leaves of the binary tree to the root. Because it is not trivial to distinguish this process from the rest of the matrix multiplication time, this distinction will not be made here. Because the various steps involved in the assembly process in a multipole accelerated computation have the potential to become time-consuming, their relative importance is shown in Figure 9.

The generation time for element moments remains small compared to the other operations, until large problems are treated. This time scales as $O(P)$, where P is the number of finite elements. The calculation of the near-field coefficients is always the most time-consuming operation, although the recursive domain subdivision and list compilation processes become increasingly important. The list compilation and recursive domain splitting procedures appear to become more efficient with increasing problem size.

Another potential problem that may be encountered when using iterative solvers is that the condition number of the matrix may increase significantly with problem size, or the number of iterations required for convergence may increase too rapidly, resulting in poor scaling. To illustrate the effects of matrix condition number, the calculation was performed using

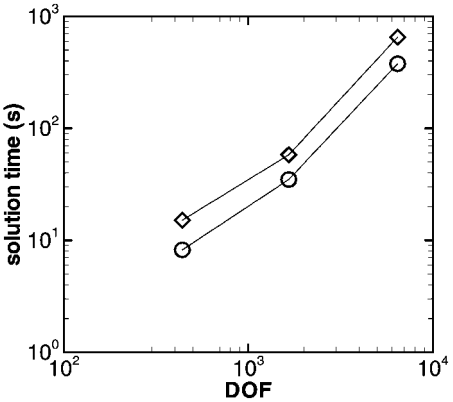

Figure 10. Solution time as a function of problem size for the multipole accelerated direct integration method: Dirichlet conditions on side $y = 0$ ($\diamondsuit$), Dirichlet conditions on side $x = 2$ ($\bigcirc$).

boundary conditions which result in matrices with different condition numbers. Inspection of Equation (6) shows that Dirichlet boundary conditions result in a first-kind integral equation, while Neumann conditions produce second-kind integral equations. The latter result in better conditioning of the A sub-matrix of Equation (10). In the case of mixed conditions, a larger proportion of Neumann conditions will result in better condition numbers. Solution times for the multipole accelerated direct integral method are compared in Figure 10 for the cases where Dirichlet conditions are imposed on the side $x = 2$ only (of length 1) versus the case where Dirichlet conditions are imposed on the side $y = 0$ only (of length 2) of the same computational domain used previously. The latter boundary conditions result in worse conditioning of the sub-matrix A.

There is a factor of two difference in the iteration time, due to the different number of iterations required for convergence. This illustrates the effect of the condition number of sub-matrix A on the iterative solution of the entire matrix. Preconditioning with the preconditioner described by Equation (11) should have the effect of reducing the number of iterations to convergence (which ideally would remain independent of the matrix size) and of removing the dependence of the number of iterations on the type of boundary conditions assigned.

4. CONCLUSIONS

Four methods for evaluating domain integrals associated with boundary element methods are compared in terms of accuracy and computational cost. The four methods include classical domain integration, domain integration by multipole expansions, the dual reciprocity method, and the particular solution method.

The dual reciprocity and particular solution methods were essentially the same in terms of both CPU requirements and accuracy for both the Poisson equation and non-homogeneous Helmholtz equation example problems. In fact, both of these methods are essentially equivalent. It is interesting to note that relatively little has been written in the literature for the particular solution method even though, in some respects, it is conceptually simpler and easier to implement than the dual reciprocity method.

The classical domain integration and domain integration by multipole expansion methods were essentially equivalent in terms of accuracy for both example problems. This simply shows that the multipole evaluation of the integrals was accurate. The classical domain integration method was typically much faster than the dual reciprocity and particular solution methods, especially as the number of radial basis functions became large. The domain integration by multipole expansions method had some overhead in terms of producing box lists and element moments which made it slower in terms of CPU for smaller problems. However, because of the $\log N$ scaling of the domain integral for a given collocation node, it became the most CPU efficient for large problems.

The interesting comparison of the four methods is in terms of their CPU costs and relative accuracy. The two domain integration methods were uniformly more accurate while typically requiring less CPU time compared to the dual reciprocity method and the particular solution method. In fact, for the two problems considered, the L_2-norm errors were anywhere from one to two orders of magnitude smaller for the domain integral methods.

Although not discussed in the results, the domain integration techniques are also superior in terms of memory requirements. In particular, for the Poisson problem, the matrices needed to evaluate the α_i's in the case of the dual reciprocity and particular solution methods limit the size of problems that can be treated. The direct domain integral evaluation methods do not require the storage of similarly sized matrices, and much larger problems can be treated. Further, in the case of the non-homogeneous Helmholtz equation problem, it can be shown that, for the multipole direct evaluation method, the required storage size scales as $O(M)$, as opposed to $O(M^2)$ for the classical direct integration method, rendering multipole integration methods even more efficient.

The main conclusions of this study are that the domain integration methods are superior in terms of CPU cost, memory requirements, and accuracy compared to the two meshless methods, namely, the dual reciprocity method and particular solution method. Further, this research has demonstrated the efficacy of evaluating domain integrals in boundary element methods using multipole acceleration. Multipole acceleration further reduces the CPU cost and memory requirements without impairing the accuracy compared with the traditional domain integration technique. In light of these conclusions, it is somewhat puzzling that the dual reciprocity method has enjoyed such popularity. The only advantage of the method is that it does not require a complete domain discretization. However, with the advanced preprocessors that are available today, this may not be that much of an advantage.

ACKNOWLEDGEMENTS

This work was partially supported by the United States Department of Energy (DOE) grants DE-FG03-97ER14778 and DE-FG03-97ER25332. This financial support does not constitute an endorsement by the DOE of the views expressed in this paper. This work was supported in part by Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the U.S. Department of Energy under Contract DE-AC04-94AL85000. The GMRES solver is used under license from CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse Cedex, France.

REFERENCES

1. Nardini D, Brebbia CA. A new approach to free vibration analysis using boundary elements. In *Boundary Elements*, vol. IV, Brebbia CA (ed), Springer-Verlag: Berlin, 1982; 312–348.

2. Wrobel LC, Brebbia CA. The dual-reciprocity boundary element formulation for non-linear diffusion problems. *Computational Methods in Applied Mechanical Engineering* 1987; **65**:147–164.
3. Wrobel LC, Brebbia CA. Boundary elements for nonlinear head conduction problems. *Communications in Applied Numerical Methods* 1988; **4**:617–622.
4. Partridge PW, Brebbia CA, Wrobel LC. *The Dual Reciprocity Boundary Element Method*. Elsevier Applied Mechanics: London, 1992.
5. Nowak AJ. The multiple reciprocity method of solving transient heat conduction problems. In *Advances in Boundary Elements*, vol. 2, Brebbia CA, Connor JJ (eds), Springer-Verlag: Berlin, 1989; 81–93.
6. Nowak AJ, Brebbia CA. The multiple reciprocity method: a new approach for transforming BEM domain integrals to the boundary. *Engineering Analysis* 1989; **6**(3):164–167.
7. Yeih W, Chen JT, Chang CM. Applications of dual MRM for determining the natural frequencies and natural modes of an Euler Bernoulli beam using the singular value decomposition method. *Engineering Analysis* 1999; **23**:339–360.
8. Kamiya N, Andoh E. A note on multiple reciprocity integral formulation for the Helmholtz equation. *Communications in Numerical Methods in Engineering* 1993; **9**:9–13.
9. Power H. A complete multiple reciprocity approximation for the non-permanent Stokes flow. In *Boundary Element Technology*, vol. IX, Brebbia CA, Kassab AJ (eds). Computational Mechanics Publications: Southampton, UK, 1994; 127–137.
10. Ahmad S, Banerjee PK. Free vibration analysis of BEM using particular integrals. *Journal of Engineering Mechanics* 1986; **112**:682–695.
11. Henry DP, Banerjee PK. A new boundary element formulation for two- and three-dimensional thermoelasticity using particular integrals. *International Journal for Numerical Methods in Engineering* 1988; **26**:2061–2077.
12. Henry DP, Banerjee PK. A new boundary element formulation for two- and three-dimensional elastoplasticity using particular integrals. *International Journal for Numerical Methods in Engineering* 1988; **26**:2079–2096.
13. Zheng R, Coleman CJ, Phan-Thien N. A boundary element approach for non-homogeneous potential problems. *Computers and Mathematics* 1991; **7**(4):279–288.
14. Ingber MS, Phan-Thien N. A boundary element approach for parabolic differential equations using a class of particular solutions. *Applied Mathematical Modelling* 1992; **16**:124–132.
15. Partridge PW. Radial basis approximation functions in boundary element dual reciprocity method. In *Boundary Element Technology*, vol. XIII, Chen CS, Brebbia CA, Pepper, DW (eds) WIT Press: Southampton, 1999; 325–334.
16. Goldberg MA, Chen CS. The theory of radial basis functions applied to the BEM for inhomogeneous partial differential equations. *Boundary Element Communications* 1994; **5**:57–61.
17. Ingber MS. A triple reciprocity boundary element method for transient heat conduction analysis. In *Boundary Element Technology*, vol. IX, Brebbia CA, Kassab AJ (eds), Elsevier Applied Science: Amsterdam, 1994; 41–49.
18. Li J, Ingber MS. A numerical study of the lateral migration of spherical particles in Poiseuille flow. *Engineering Analysis and Boundary Elements* 1994; **13**(1):83–92.
19. Korsmeyer FT, Yue DKP, Nabors K, White J. Multipole-accelerated preconditioned iterative methods for three-dimensional potential problems. In *Boundary Elements*, vol. XV, Brebbia CA, Rencis JJ (eds), Computational Mechanics Publications: Southampton, 1993; 517–527.
20. Allen EH, Herendeen JB, Levin PL, Petrangelo JH, Spasojevic M. In *A Galerkin Implementation of the Multipole Expansion Approach for Accurate and Fast Solution of First and Second Kind Fredholm Equations*, Brebbia CA, Rencis JJ (eds), Computational Mechanics Publications: Southampton, 1993; 485–500.
21. Gómez JE, Power H. A multipole direct and indirect BEM for 2D cavity flow at low Reynolds number. *Engineering Analysis of Boundary Elements* 1997; **19**:17–31.
22. Mammoli AA, Ingber MS. Stokes flow around cylinders in a bounded two-dimensional domain with multipole-accelerated boundary element methods. *International Journal for Numerical Methods in Engineering* 1999; **44**(7):875–895.
23. Barnes J, Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature* 1996; **324**:446–449.
24. Saad Y, Schultz M. GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Sciences Statistics and Computing* 1986; **7**:856–869.

