

# An accelerated surface discretization-based BEM approach for non-homogeneous linear problems in 3-D complex domains

Jian Ding<sup>1</sup>, Wenjing Ye<sup>1,\*</sup>,<sup>†</sup> and L. J. Gray<sup>2</sup>

<sup>1</sup>*Woodruff School of Mechanical Engineering, Georgia Institute of Technology,  
Atlanta, GA 30332-0405, U.S.A.*

<sup>2</sup>*Computer Science and Mathematics Division, Oak Ridge National Laboratory, U.S.A.*

## SUMMARY

For non-homogeneous or non-linear problems, a major difficulty in applying the boundary element method (BEM) is the treatment of the volume integrals that arise. An accurate scheme that requires no volume discretization is highly desirable. In this paper, we describe an efficient approach, based on the precorrected-FFT technique, for the evaluation of volume integrals resulting from non-homogeneous linear problems. In this approach, the 3-D uniform grid constructed initially to accelerate surface integration is used as the baseline mesh for the evaluation of volume integrals. As such, no volume discretization of the interior problem domain is necessary. Moreover, with the uniform 3-D grid, the matrix sparsification techniques (such as the precorrected-FFT technique used in this work) can be extended to accelerate volume integration in addition to surface integration, thus greatly reducing the computational time. The accuracy and efficiency of our approach are demonstrated through several examples. A 3-D accelerated BEM solver for Poisson equations has been developed and has been applied to a 3-D multiply-connected problem with complex geometries. Good agreement between simulation results and analytical solutions has been obtained. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: Poisson equation; BEM; volume integral; fast algorithm; precorrected-FFT technique

## 1. INTRODUCTION

The need for efficient solutions to large-scale problems, such as those encountered in micro-electromechanical systems (MEMS), has led to the development of fast algorithms. Integral-based methods have the advantage in reduced dimensionality and problem size and thus are good candidates as efficient methods for large-scale problems with complex 3-D geometries.

\*Correspondence to: Wenjing Ye, Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405, U.S.A.

<sup>†</sup>E-mail: wenjing.ye@me.gatech.edu

Contract/grant sponsor: NSF; contract/grant number: CCR 0306664

Contract/grant sponsor: U.S. Department of Energy; contract/grant number: DE-AC05-00OR22725

*Received 19 July 2004*

*Revised 28 October 2004*

*Accepted 31 January 2005*

Moreover, this type of approaches can be attractive for problems with unbounded domains and/or moving boundaries, etc. One drawback of integral-based methods is that they generate fully populated system matrices. Solving them requires  $O(n^3)$  operations ( $n$  is the system size) if direct methods are employed or  $O(n^2)$  operations if iterative methods are used. Recent development of matrix sparsification techniques such as fast multipole method (FMM) [1, 2] and precorrected-FFT (p-FFT) technique [3–5] has reduced the computational complexity to  $O(n \log n)$  or  $O(n)$ . Based on the accelerated boundary element method (BEM), fast solvers for electrostatic problems [6], Stokes problems [7–10], etc. have been developed and applied successfully in solving practical problems, for example, in the design of very large scale integration (VLSI) circuits [11] and in the modelling of complex MEMS devices [12].

It is perhaps fair to say that to date most applications of the BEM have been limited to linear and homogeneous problems. This is largely due to the difficulties associated with the evaluation of volume integrals in the boundary integral formulations that result from non-homogeneous or non-linear problems. Efficient and accurate treatment of these volume integrals is the major bottleneck in the extension of the BEM to non-linear or non-homogeneous problems. The challenge and the significance of this problem have attracted the attention of many researchers, and over last several decades, various approaches have been proposed and developed [13–28]. For a detailed review, readers are referred to a paper by Hsiao *et al.* [28]. In summary, these approaches can be classified into two main categories: domain discretization-free methods (or meshless methods) and cell-based direct integration schemes. Domain discretization-free methods such as particular solution methods (PSM) [14, 16] and dual reciprocity methods (DRM) [13, 19] or most recently multiple reciprocity formulation (MRM) [17] use different techniques to either eliminate the volume integrals [14, 16] or (approximately) transform them into boundary integrals [13, 19]. The major advantage of this type of methods is that the boundary-only nature of the BEM is retained. However, as pointed out in References [24, 28] and also based on our own experience, the quality of these methods depends on the quality of the radial basis functions (RBF) approximation. Finding the optimal methods for approximations is still an active research topic. In addition, very large, fully populated and often poorly conditioned matrices are generated [26], resulting in a significantly high computational cost.

Cell-based direct integration schemes employ an interior volume mesh to directly perform the integration. A major advantage of this type of schemes is the high accuracy. Such approaches may also be efficient, as matrix sparsification techniques can be employed to accelerate volume integration. In a paper by Ingber *et al.* [24], a comparison of the different integration schemes has been performed. It was found that the classical cell-based direct integration method coupled with FMM may be significantly better (in terms of computational accuracy and efficiency) than both the dual reciprocity method and PSM. However, a major disadvantage of cell-based direct integration schemes is the requirement of volume discretization of the problem domain. Unless the volume integral exists only in a small region of the problem domain or the domain is of simple geometry, such approaches would lose the main advantage of the BEM, namely the boundary-only discretization and therefore would be less attractive for problems with complex geometries. In an attempt to reduce the complexity caused by volume discretization, auxiliary domain subtraction method [26, 28] has been developed to simplify the mesh generation for multiply-connected problems. Such a method is effective only when the inclusions or holes are of simple geometries, for which volume discretization can be easily performed.

It seems that an ideal approach for the treatment of volume integrals would be the one that has the accuracy and the efficiency of the cell-based direct integration methods and yet does

not have the high cost associated with volume discretization. In this paper, we describe a new approach based on the p-FFT technique for the evaluation of volume integrals. This approach uses the 3-D uniform grid built initially for the purpose of rapid evaluation of surface integrals as the baseline volume cells to perform volume integration. The advantage of this approach is two-fold. First no volume discretization of the interior domain is needed. The required inputs are the surface mesh and the uniform 3-D grid that encompasses/encloses the problem domain. Second, with the uniform 3-D grid, volume integration can be accelerated the same way as surface integration, resulting in a fast volume integral evaluation.

In order to illustrate the proposed approach clearly, we first give a brief introduction of the p-FFT technique. This technique is used to accelerate both surface and volume integrals in this work. A description of the proposed volume-integration approach is presented next followed by a collection of numerical techniques for the implementation of this approach. In Section 3, results for volume integrals evaluated on a simple geometry, namely a sphere, are presented first to illustrate the accuracy and efficiency of the proposed schemes. A 3-D Poisson solver has been developed and applied to a 3-D multiply-connected problem and the results are compared with analytic solutions. Finally, we summarize our work in Section 4.

## 2. VOLUME-INTEGRATION APPROACH

### 2.1. The boundary integral formulation

For the purpose of clarity, we use the Poisson equation to illustrate our approach. The general form of the Poisson equation reads

$$\Delta u = b(\mathbf{x}) \quad (1)$$

where  $u$  is the potential and  $b$  is the source term or inhomogeneous term. The Poisson equation in Equation (1) is subject to appropriate boundary conditions, and can be cast into a boundary integral formulation

$$c(\mathbf{x})u(\mathbf{x}) + \int_{\partial\Omega} \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{y})} u(\mathbf{y}) dS(\mathbf{y}) - \int_{\partial\Omega} G(\mathbf{x}, \mathbf{y}) \frac{\partial u(\mathbf{y})}{\partial n(\mathbf{y})} dS(\mathbf{y}) = - \int_{\Omega} G(\mathbf{x}, \mathbf{y}) b(\mathbf{y}) dv(\mathbf{y}) \quad (2)$$

where

$$c(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega \\ 0, & \mathbf{x} \notin (\Omega \cup \partial\Omega) \\ \alpha, & \mathbf{x} \in \partial\Omega \end{cases}$$

In (2),  $\Omega$  is the domain of the problem with boundary  $\partial\Omega$ ,  $n(\mathbf{y})$  is the unit outward normal vector at the field point  $\mathbf{y}$ ,  $\mathbf{x}$  is the evaluation point and  $G(\mathbf{x}, \mathbf{y})$  is the Green's function of the Laplace operator. In a 3-D space, it is given by

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi r} \quad \text{where } r = |\mathbf{x} - \mathbf{y}|$$

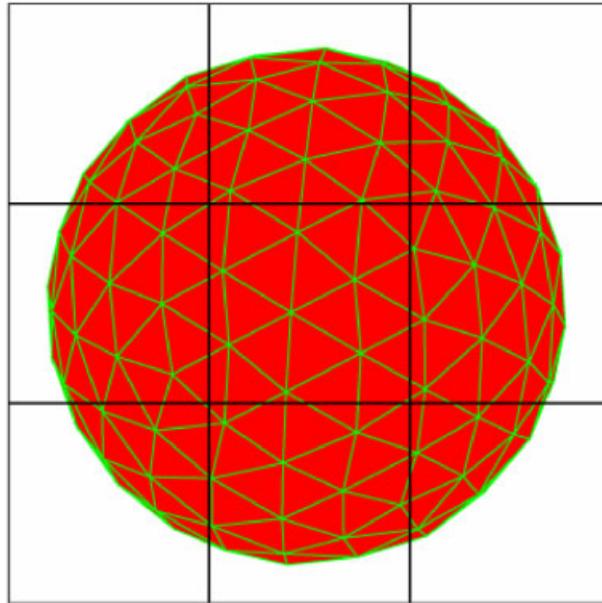


Figure 1. Side view of a sphere encompassed by a parallelepiped with a  $3 \times 3 \times 3$  array of cubes.

### 2.2. The precorrected-FFT technique

In the p-FFT technique, a parallelepiped is constructed to enclose a 3-D problem domain after it has been discretized into  $n$  surface panels. This parallelepiped is then subdivided into a  $k \times l \times m$  array of small cubes so that each small cube contains only a few panels. Figure 1 shows a discretized sphere, with the associated space subdivided into an  $3 \times 3 \times 3$  array of cubes. It should be pointed out that the surface panels and the p-FFT cubes can intersect with each other. There is no need to maintain any consistency between the surface panels and the cubes.

The acceleration of surface integration is achieved by exploiting the fact that the kernels in the surface integrals such as those in Equation (2) have piecewise-smooth convolutional form. Thus with the aid of the uniform grid formed by the cubes in the parallelepiped, these integrals can be computed approximately using the Fast Fourier Transform technique. To ensure accuracy, such an approximation is only employed for far-field interactions (i.e. integrals in which the evaluation point is far away from the field panel). For nearby interactions, direct evaluation is required. For a detailed description of this technique, readers are referred to Reference [3].

### 2.3. Approach for volume integration

The evaluation of the volume integral on the right-hand side of (2) requires volume discretization of  $\Omega$  if a standard cell-based direct integration scheme is employed. In our approach, we use the uniform 3-D FFT grid (see Figure 1), which was set initially for accelerating the surface integration, to perform the volume integration. To do so, let  $\tilde{b}(\mathbf{y}) = b(\mathbf{y})$  if  $\mathbf{y} \in \Omega$  and  $\tilde{b}(\mathbf{y}) = 0$  if  $\mathbf{y} \in B \setminus \Omega$ , where  $B$  denotes the uniform grid. The volume integral in (2)  $\int_{\Omega} b(\mathbf{y})G(\mathbf{x}, \mathbf{y}) d\mathbf{v}(\mathbf{y})$

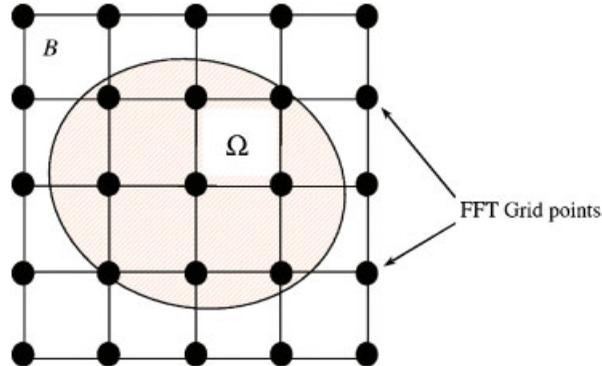


Figure 2. 2-D illustration of relationship between problem domain and the uniform FFT grid—shaded area is the problem domain  $\Omega$ ,  $B$  is the uniform grid.

can then be replaced by a volume integral performed on the grid, i.e.  $\int_B \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) dv(\mathbf{y})$ . Since  $B$  consists of a set of cubes (see Figure 2), this integral is also equivalent to the sum of integrals evaluated on each cube, i.e.

$$\int_B \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) dv(\mathbf{y}) = \sum_i \int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) dv(\mathbf{y}) \quad (3)$$

where  $C_i$  is the  $i$ th cube of  $B$ . Now the integration domain has been transferred from the complex problem domain ( $\Omega$ ) to a regular grid ( $B$ ) consisting of a set of cubes.

*2.3.1. Evaluation of  $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) dv(\mathbf{y})$ .* If volume integrals present in the integral formulation are also of convolutional form (which is true for many engineering problems), the p-FFT technique can be used to accelerate volume integration in addition to surface integration. For example, the integral  $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) dv(\mathbf{y})$  with evaluation point  $\mathbf{x}$  located inside cube  $C_j$  can be approximated by

$$\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) dv(\mathbf{y}) \approx \sum_{\mu} \mathbf{W}_{\mu} \left( \int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu}, \mathbf{y}) dv(\mathbf{y}) \right) \quad (4)$$

in which  $\mathbf{x}_{\mu}$  represents  $\mu$ th grid point on cube  $C_j$  and  $W_{\mu}$  is an interpolation operator that interpolates function value at  $\mathbf{x}$  based on values at surrounding grid points ( $\mathbf{x}_{\mu}$ ).

The integral  $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu}, \mathbf{y}) dv(\mathbf{y})$  in Equation (4) can be further approximated as

$$\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu}, \mathbf{y}) dv(\mathbf{y}) \approx \sum_v G(\mathbf{x}_{\mu}, \mathbf{y}_v) \cdot \tilde{b}(\mathbf{y}_v) = \sum_v G(\mathbf{x}_{\mu}, \mathbf{y}_v) \cdot \int_{C_i} P_v(\tilde{b}(\mathbf{y})) dv(\mathbf{y}) \quad (5)$$

In (5),  $P_v$  is a projection operator that projects the source term  $\tilde{b}$  located at  $\mathbf{y}$  onto the  $v$ th grid point ( $\mathbf{y}_v$ ) located on cube  $C_i$ . As such the original integral  $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) dv(\mathbf{y})$  is now replaced by

$$\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu}, \mathbf{y}) dv(\mathbf{y}) \approx \sum_{\mu} \mathbf{W}_{\mu} \sum_v G(\mathbf{x}_{\mu}, \mathbf{y}_v) \cdot \tilde{b}(\mathbf{y}_v) \quad (6)$$

In summary, three major steps are involved in the evaluation of  $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) d\nu(\mathbf{y})$ :

- Projection: project  $\tilde{b}(\mathbf{y})$  to produce  $\tilde{b}(\mathbf{y}_\nu)$  according to Equation (5).
- Convolution: evaluate  $\sum_\nu \tilde{G}(\mathbf{x}_\mu, \mathbf{y}_\nu) \cdot \tilde{b}(\mathbf{y}_\nu)$  using FFT.
- Interpolation: obtain  $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) d\nu(\mathbf{y})$  using values at surrounding grid points ( $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}_\mu, \mathbf{y}) d\nu(\mathbf{y})$ ) by interpolation as illustrated in Equation (4).

Obviously, the above procedure will produce accurate results only when the evaluation cube  $C_j$  and the field cube  $C_i$  are reasonably far away. When they are close, direct evaluation of  $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}, \mathbf{y}) d\nu(\mathbf{y})$  is to be performed.

**2.3.2. Implementation schemes.** The general scheme for the implementation of the accelerated approach to evaluate volume integrals is similar to that of surface integrals [3, 8]. A polynomial interpolation scheme is used to project sources inside cubes to the surrounding grid points as well as to interpolate values from grid points back to the evaluation points. For the convolution, the fast Fourier transform technique is used. The major difference between volume and surface integration, and the most challenging task, is the direct evaluation of volume integrals on the cubes. Such calculations are needed in the direct calculation of nearby interactions and also in the projection step.

If the cube is totally inside the problem domain  $\Omega$  (i.e.  $\tilde{b}(\mathbf{y}) = b(\mathbf{y})$ ), regular Gauss quadrature may be employed to evaluate the volume integral if it is not singular. When the cube intersects with  $\partial\Omega$  or when the integral is singular, special schemes must be developed to accurately and efficiently compute the integrals.

Consider a volume integral defined on a cube that intersects with the problem domain  $\Omega_i$ . The integrand is non-zero inside  $\Omega_i$  (shaded region shown in Figure 3(a)) and zero outside  $\Omega_i$ . The only given information about  $\Omega_i$  is the surface mesh (triangular panels shown in

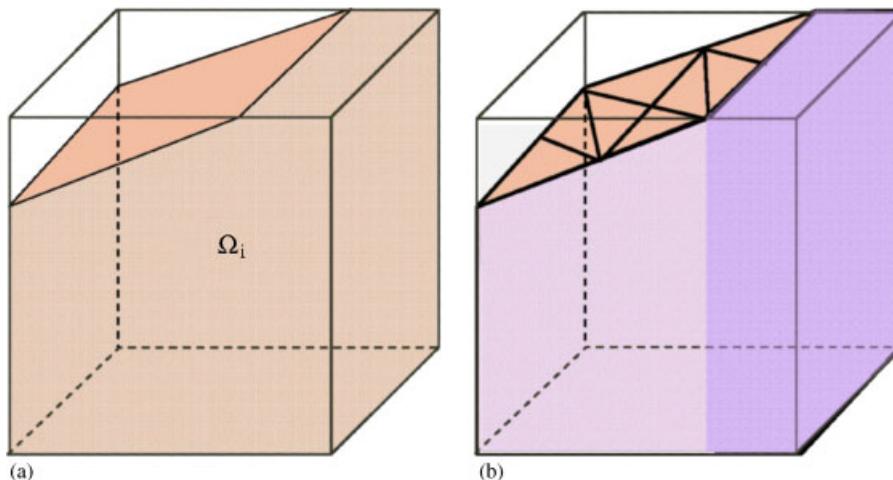


Figure 3. (a) A partially filled boundary cube,  $\Omega_i$  is the problem domain; and (b) a partially filled cube with surface panels.

Figure 3(b)). To evaluate this integral, a ‘projection + transformation’ scheme that uses only the existing surface elements has been developed. This scheme can also be readily extended to accurately evaluate singular integrals.

*2.3.2.1. The ‘projection + transformation’ scheme.* In the ‘projection + transformation’ scheme, prisms are formed by projecting surface elements to one side of the cube. The projection direction is determined automatically in the code based on the consideration of efficiency. For example, the optimal projection strategy for the case shown in Figure 3(b) is to project panels down to the bottom face of the cube. Further division of the prism along the projection direction is performed if necessary to maintain a good aspect ratio of prisms. These prisms serve as the fine ‘volume elements’ and integration is performed on these elements. With proper addition or subtraction of integrals over these prisms and the compensation volume (i.e. the volume which is not covered by prisms, for example, the purple region on the right in Figure 3(b)), the desired volume integral on the cube can be obtained. For the case shown in Figure 3, the integral over  $\Omega_i$  is the summation of the same integral evaluated on prisms and on the compensation volume.

To evaluate integrals on prisms, if the integral is non-singular, a coordinate transformation is applied to transform prisms into regular domains shown in Figure 4 and Gaussian quadrature is then employed to perform the integration.

For singular integrals, i.e. the integrand approaches to infinity at certain point (when  $\mathbf{x} \in C_i$ ), the above scheme will not produce accurate results even when the Gauss points do not coincide with the singular point. To improve accuracy, the singularity is first removed from the integral before the Gaussian quadrature is applied. This is achieved by performing an additional prism subdivision before transformation, and then applying a transformation whose Jacobian cancels the singularity. As illustrated in Figure 5, after projection, the singular prism is separated into three smaller prisms (Figure 5(a)), each with the evaluation point as one of its vertices (Figure 5(a)). A transformation similar to the one described in Reference [29] is then performed

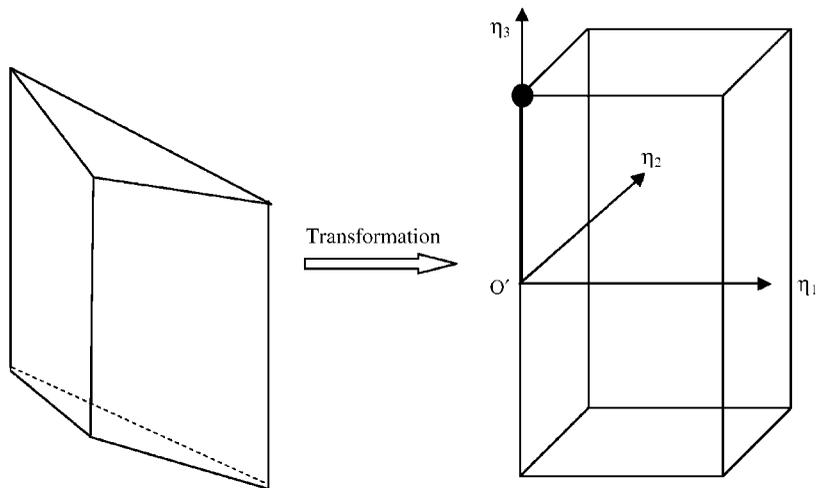


Figure 4. Transformation of a prism into a regular domain.

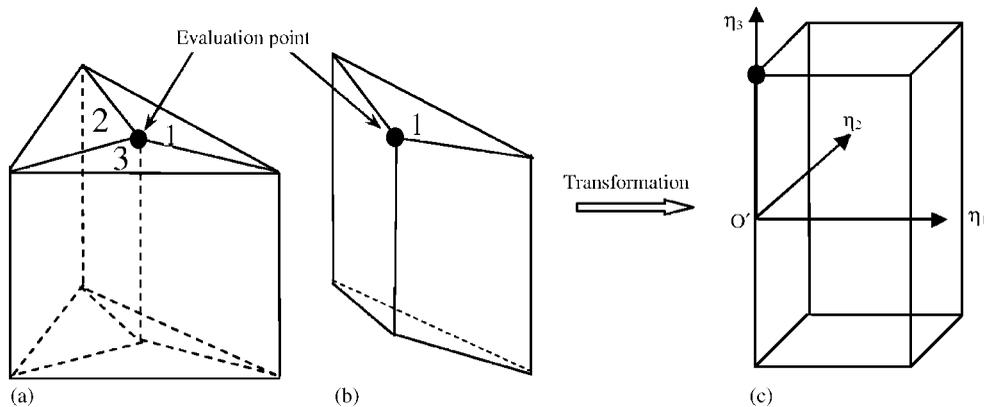


Figure 5. (a) A singular prism being divided into three smaller prisms; (b) a smaller prism with the evaluation point located on one of its vertices; and (c) coordinate transformation: prism 1 is transformed into a regular domain. Similar transformations can be done for prisms 2 and 3.

for each smaller prism (Figure 5(b)) to remove the singularity analytically (Figure 5(c)) and Gaussian quadrature is then applied to evaluate the resulting non-singular integrals.

*2.3.2.2. Issues in the implementation of the 'projection + transformation' scheme.* In the implementation of the 'projection + transformation' scheme, one issue has arisen due to the fact that the surface elements and the p-FFT cubes can intersect with each other. As shown in Figure 6, both surface panels  $\Gamma_1$  and  $\Gamma_2$  cross the cube boundaries. Based on the locations of their centroids (light grey dots (red dots in online version) in Figure 6), panel  $\Gamma_1$  is assigned to cube  $C_2$  and  $\Gamma_2$  is assigned to cube  $C_4$ . When the projection directions in two neighbouring cubes are different, it is possible to have a region on which the integration has been performed twice. For example, in Figure 6, if the panels in cubes  $C_1$  and  $C_2$  are projected down while panels in cube  $C_4$  are projected to the left, the small region shaded by strips is double counted which will cause errors (so-called 'overlapping' errors) in the volume integration if no care is taken.

The current strategy to deal with this issue is to truncate the panels that cross the cube boundaries (Figure 7) and to form a new set of surface panels. This is done by identifying the intersections between panels and the cube, which can be readily obtained since the geometries of both the panel and cube are given. The projection is then performed on the new set of panels. This approach guarantees that in the new mesh, no panel will cross the cube boundaries and therefore the scenario described above will not occur. One drawback of such an approach is that it increases the number of panels. As shown in Figure 7, one panel has become three panels after truncation. Nevertheless the increased number of panels will not dramatically increase the computational cost since the new mesh is only used for the evaluation of volume integrals on the boundary cubes. Integration on interior cubes does not need any mesh and surface integration is still performed using the original surface mesh.

Another implementation issue is the identification of the compensation volume. As mentioned before, the compensation volume is part of the problem domain ( $\Omega_i$ ) that is not covered by the prisms formed from surface panels. This identification is accomplished by utilizing the new

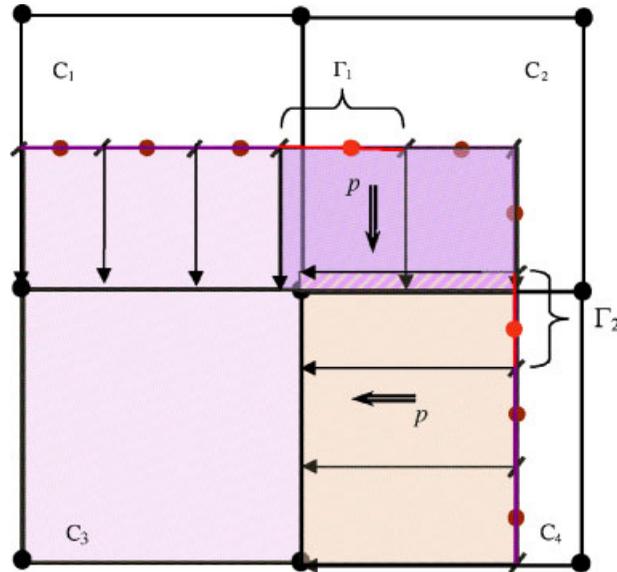


Figure 6. Illustration of scenario in which overlapping errors may occur.

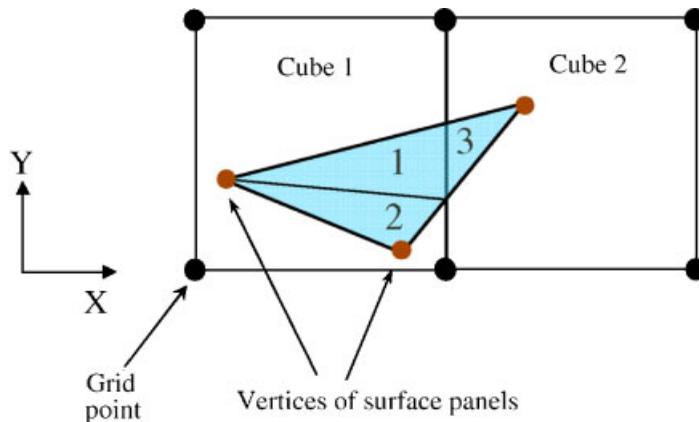


Figure 7. Panel truncation using cube boundaries

surface mesh to find the intersection curves of the problem domain and the cube as shown in Figure 8. Integration is then performed by the ‘projection + transformation’ scheme for non-singular integrals as described above.

**2.3.3. Complexity of volume integration.** The complexity analysis of volume integration based on the p-FFT accelerated technique is similar to that of surface integration. In the direct approach (as shown in Figure 9(a)), for each evaluation point,  $m$  volume integrals need to be

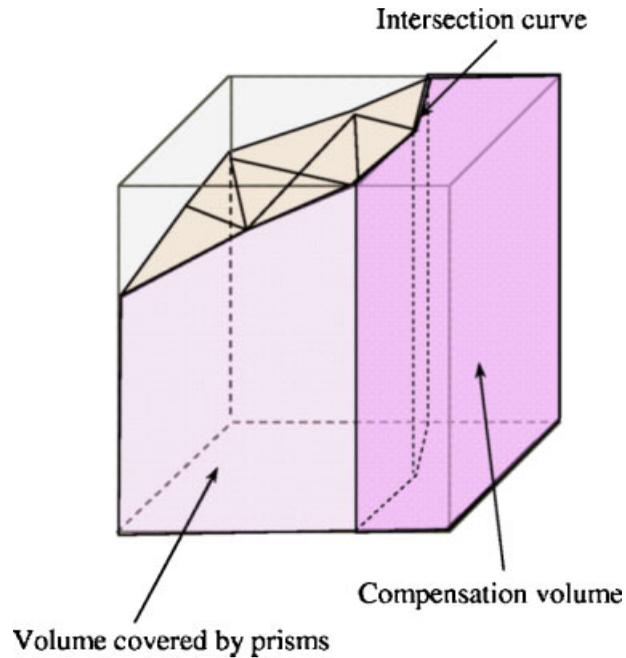


Figure 8. Illustration of the compensation volume.

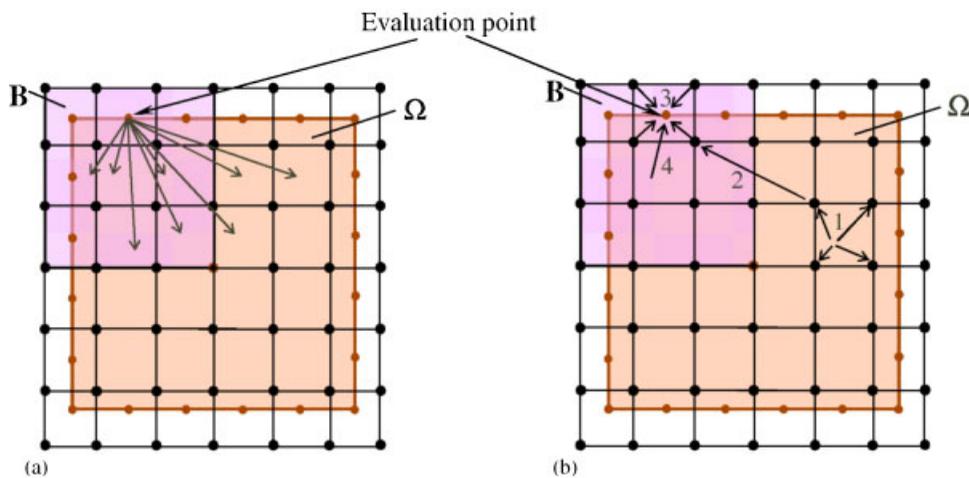


Figure 9. (a) Illustration for direct approach; and (b) illustration for p-FFT acceleration: (1) projection, (2) convolution, (3) interpolation, (4) nearby interactions.

performed, where  $m$  represents the number of cubes. Thus the complexity of volume integration for  $n$  evaluation points is  $O(nm)$ . The p-FFT accelerated approach illustrated in Figure 9(b) requires  $O(m) + O(m \log m) + O(n)$  operations to perform the same integration. About  $O(m)$

operations are used to perform projection. The convolution step takes about  $O(m \log m)$  operations. Both interpolation and direct calculations for nearby interactions require  $O(n)$  operations.

When the number of evaluation points  $n$  which is related to the number of panels is larger than or equivalent to the number of cubes  $m$  (typically this is the case), acceleration in volume integration based on the p-FFT technique is achieved.

#### 2.4. Numerical implementation scheme for solving the Poisson equation

The integral equation shown in Equation (2) was solved using a BEM together with the p-FFT acceleration technique. For simplicity, a piecewise constant collocation scheme is used to discretize the integral equation. The surface of the structure is discretized into small panels. On each panel,  $u$  and  $\partial u/\partial n$  are assumed to be constant. A system of equations for the panel unknowns is then derived by insisting that the integral equations are satisfied at each panel centroid. The result is a linear system which relates the known quantities ( $u$  in our example) to the unknown quantities ( $q = \partial u/\partial n$ ), as in Equation (7)

$$\mathbf{G} \cdot \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{Bmatrix} + \mathbf{F} \cdot \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{Bmatrix} \quad (7)$$

In (7), the load vector  $d$  consists of the volume integrals shown in (2). The linear system in (7) is solved using GMRES [30] and the procedure is accelerated using the p-FFT technique.

### 3. RESULTS AND DISCUSSION

#### 3.1. Volume integration

**3.1.1. Singular integrals—convergence study.** The proposed prism subdivision combined with ‘projection + transformation’ scheme for handling singular volume integrals was tested by integrating  $1/\|\mathbf{x} - \mathbf{y}\|$  on the singular prism shown in Figure 5(b) with  $\mathbf{x}$  being the evaluation point and  $\mathbf{y}$  being the field point. The convergence plot is shown in Figure 10 together with the results obtained by applying Gauss quadrature directly without removing the singularity (direct Gauss quadrature scheme). With order of 19, the result obtained by the proposed scheme converges to at least 4 digits while the direct Gauss quadrature scheme fails to converge even with the order of 100. This clearly indicates the influence of singularity on the integration and the success of the proposed scheme for handling the singular integrals.

**3.1.2. Non-singular integrals—direct method versus accelerated method.** The accuracy of the p-FFT accelerated technique when applied to evaluate volume integrals is tested on several field cubes that have different distances from the evaluation points. Two approaches are compared here: direct evaluation using Gauss quadrature and the p-FFT accelerated integration. The integral tested is again  $\int_{C_i} (1/\|\mathbf{x} - \mathbf{y}\|) dv(y)$ , evaluated on cubes  $C_i$ ,  $i = 1, 2, 3, 4$ . Both  $C_1$  and  $C_2$  are fully filled cubes, corresponding to the cases when the cubes are totally inside

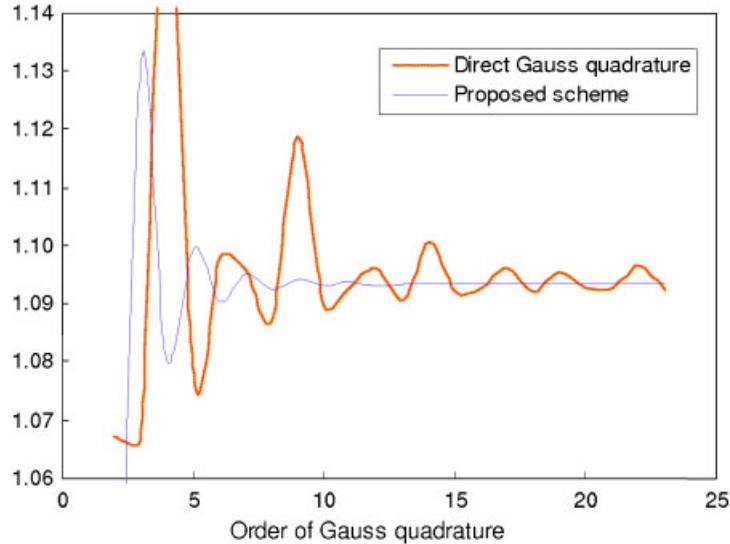


Figure 10. Convergence of evaluation schemes for singular volume integrals.

Table I. Accuracy of  $\int_{C_i} (1/\|\mathbf{x} - \mathbf{y}\|) dv(\mathbf{y})$ .

Test cubes	Distance	Direct GQ	p-FFT	Error
$C_1$	0.61	0.0952561	0.0952363	0.0208%
$C_2$	1.1867	0.0606677	0.0606677	0.0000%
$C_3$	1.6117	0.0106495	0.0106488	0.0066%
$C_4$	2.1734	0.000117512	0.00011751	0.0009%

the problem domain.  $C_3$  and  $C_4$  are chosen to be partially filled cubes to test cases when the cubes intersect with the domain boundary. Thus the integrand is zero in some parts of  $C_3$  and  $C_4$ . In all the cases considered, the evaluation point is the same and is located at  $(-0.68449, -0.451469, -0.565426)$ . Table I compares the results of  $\int_{C_i} (1/\|\mathbf{x} - \mathbf{y}\|) dv(\mathbf{y})$  obtained from the direct Gauss quadrature (GQ) approach and the approximate method using the p-FFT technique. The relative errors together with the distances from the evaluation point to the cube in each case are also listed. It is clear from the table that errors introduced by the acceleration are negligible in all cases shown and in general errors decrease when the distance between the cube and the evaluation point increases. The reason that errors in  $C_3$  and  $C_4$  are slightly larger than that in  $C_2$  is due to the fact that in  $C_3$  and  $C_4$  the integrands are not smooth (zero in parts of the cubes).

In general, the accuracy of the p-FFT technique depends on the accuracy of the interpolation scheme used to project sources to grid points and to interpolate values back to the evaluation point. If the polynomial interpolation scheme is used, the acceleration error depends on the order of the polynomial order of polynomial interpolation (the number of grid points), the location of the cube relative to the boundary and of course the complexity of the source term.

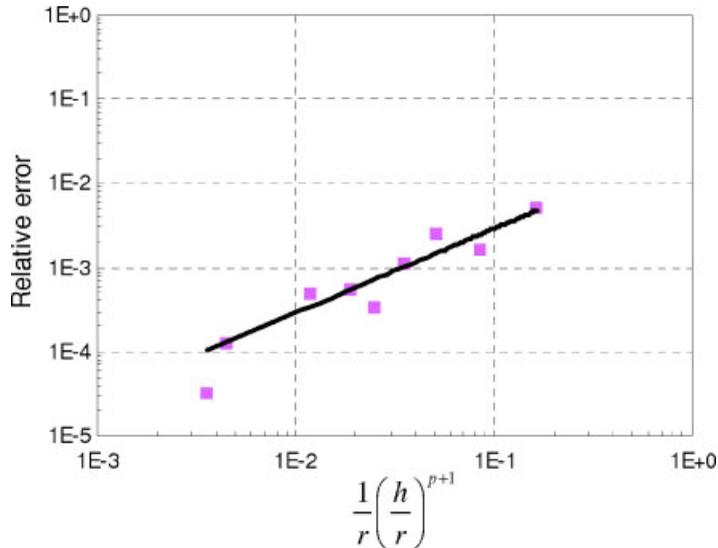


Figure 11. Log-log plot of relative error of  $\int_{C_i} (1/\|\mathbf{x} - \mathbf{y}\|) dv(\mathbf{y})$  versus  $1/r \cdot (h/r)^{p+1}$ , where  $p = 1$ .

For a source term of  $1/r$ , the error introduced in the approximation is roughly proportional to  $1/r \cdot (h/r)^{p+1}$ , where  $r$  is the distance between a evaluation point and a field cube,  $h$  is the space between two nearest grid points and  $p$  is the order of polynomial interpolation. This is supported by Figure 11 in which the relative errors of  $\int_{C_i} (1/\|\mathbf{x} - \mathbf{y}\|) dv(\mathbf{y})$  evaluated at different cubes as a function of  $1/r \cdot (h/r)^{p+1}$  are plotted. As expected, a linear relationship with a slope of one in the log-log plot is observed.

**3.1.3. Complete volume integral.** The accuracy and efficiency of the proposed volume-integration scheme were tested on  $\int_{\Omega} (1/\|\mathbf{x} - \mathbf{y}\|) dv(\mathbf{y})$  for  $\Omega$  a solid sphere with a radius of 1, shown in Figure 12. The evaluation point  $\mathbf{x}$  is the centroid of first surface panel in the data structure. Three approaches are compared.

- I. Direct calculation using the ‘projection + transformation’ scheme. Since there is no approximation, results obtained by using this approach serve as the reference and errors are calculated based on it.
- II and III. Acceleration for volume integration
  - II. Integration over the singular cube and its nearest neighbours are performed directly. Integration on all other cubes is performed approximately.
  - III. Only the singular cube is treated directly.

Results of the integrals evaluated using the different schemes are listed in Tables II and III. Three different surface discretizations were used, formed based on three original sets of surface panels with the total number of panels being 192, 768 and 3072, respectively. Figure 12(a) shows one original set of 768 surface panels. Figure 12(b) shows the corresponding mesh after truncation and Figure 12(c) shows the FFT grid on the mesh in (b). Recall that the purpose

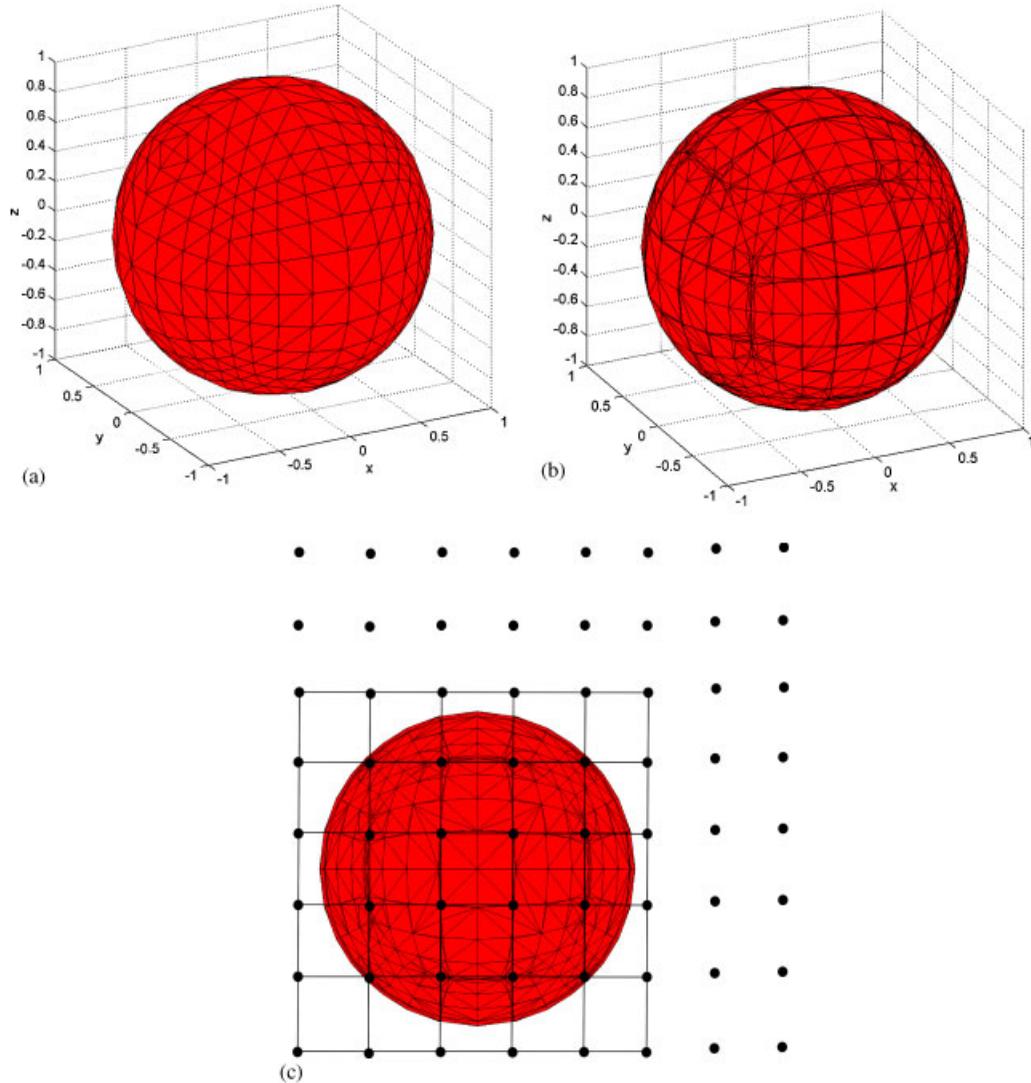


Figure 12. A solid sphere domain: (a) original surface mesh (768 panels); (b) new surface mesh (2244 panels); and (c) sphere with the 3-D uniform grid.

in forming a new set of panels is to eliminate the overlapping errors. In the case of 192, no panels intersect with cubes therefore the new set of panels is the same as the original one. From Table II, it is clear that errors introduced by approximation are negligible particularly when approach II is used. In the case of 192, the error is zero when II was used and 0.63% when III was used. This is due to the fact that in this case the total number of cubes is 8, i.e. 2/side. Thus all the cubes are the nearest neighbours to each other and according to II, they are calculated directly. In III, however, only the singular cube is evaluated directly, resulting in

Table II. Accuracy of different integration schemes.

No. of surface panels	Evaluation point (x)	I	II	Error (II - I)/I	III	Error (III - I)/I
192	(-0.686838, -0.439644, -0.550755)	0.633815	0.633815	0%	0.637835	0.63%
2244	(-0.68449, -0.451469, -0.565426)	0.658263	0.658288	0.04%	0.654849	0.52%
5376	(-0.715191, -0.654795, -0.237605)	0.664413	0.664473	0.01%	0.660061	0.66%

Table III. CPU time of different schemes (s).

No. of surface panels	I	II	Ratio (II/I)	III	Ratio (III/I)
192	0.0194	0.0194	100%	0.0151	77.8%
2244	0.2074	0.1859	89.6%	0.0338	16.3%
5376	0.7969	0.2863	35.9%	0.0340	4.26%

a different value from those obtained from I and II. In the cases of 768 and 3072, the number of cubes per side is five. Thus there are cubes that are not the nearest neighbours to each other. This is the reason why there are differences between the results obtained by different schemes.

Table III gives a comparison of the computational cost associated with volume integration of each method. All the calculations were performed on a PC with Pentium 4 1.8 GHz processor. The savings in CPU time when acceleration is performed is evident from the results and more efficiency is achieved when the discretization is finer.

Overall, the case study (i.e.  $\int_{\Omega} (1/\|\mathbf{x} - \mathbf{y}\|) dv(\mathbf{y})$  on a solid sphere) shown in this section has demonstrated the accuracy and efficiency of the accelerated approach.

### 3.2. Poisson problems

Based on the proposed volume-integration scheme, a 3-D Poisson solver has been developed. This solver uses a uniform 3-D grid (FFT grid) as the coarse mesh to perform volume integration. Same grid is also used to accelerate the evaluation of both surface and volume integrals. The accuracy and efficiency of the solver are demonstrated through two examples described in the following sections. In both cases, Dirichlet boundary conditions are prescribed. The solutions found through simulations are the normal flux at the boundaries. Results are compared with the exact solutions and errors measured using the  $L_2$ -norm (Equation (8)) are reported.

$$\text{Error} = \sqrt{\sum_{i=1}^N \left( \frac{\partial u^s}{\partial n} - \frac{\partial u^e}{\partial n} \right)^2 ds_i} / \sqrt{\int \left( \frac{\partial u^e}{\partial n} \right)^2 ds} \quad (8)$$

where superscript s indicates simulation and e represents the exact solution.

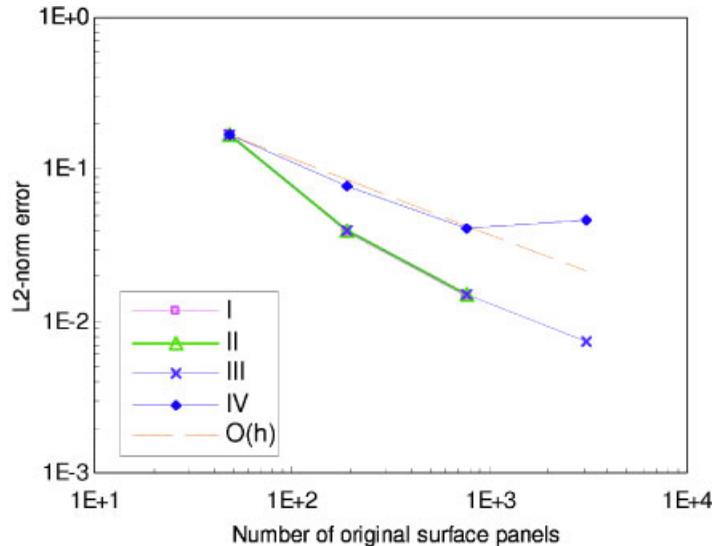


Figure 13. The  $L_2$ -norm errors in  $\partial u/\partial n$  of Poisson Example 1.

3.2.1. *Example 1—a solid sphere with radius 1 (Figure 12).* The problem solved in this example is  $\nabla^2 u = 1$  with the boundary conditions of  $u = x^2/2$ . The analytic solution for the normal flux at the boundary is  $\partial u/\partial n = x \cdot n_x$  where  $n_x$  is the  $x$ -component of the outward normal vector. Similar to the previous case study, four approaches are compared:

- I. Direct calculation for both surface and volume integration, uses only the original surface mesh (Figure 12(a)).
- II. Acceleration for surface integration only, uses original surface mesh (Figure 12(a)) for surface integration and new mesh (Figure 12(b)) for volume integration.
- III and IV. Acceleration for both surface and volume integration, uses original surface mesh (Figure 12(a)) for surface integration and new mesh for volume integration (Figure 12(b)).
- III. Integration on the singular cube and its nearest neighbours is performed directly. Integration on all other cubes is performed approximately.
- IV. Only the singular cube is treated directly.

The errors in the normal flux at the boundaries are plotted in Figure 13, together with a line of  $O(h)$ , where  $h$  is the panel size. There are no data points corresponding to the original discretization of 3072 for approaches I and II due to exhaustion of machine memory. It is interesting to see that errors in approaches I–III are very close. This further confirms our previous observation that errors caused by acceleration are negligible. In addition, the errors seem to come mostly from the surface integration. Figure 14 shows a similar error plot for the Laplace problem (only results from the accelerated approach are plotted). The level of errors is comparable with that of the Poisson problem. The convergence rate in all the approaches except IV is around  $O(h)$ . This is consistent with the constant element used in this work.

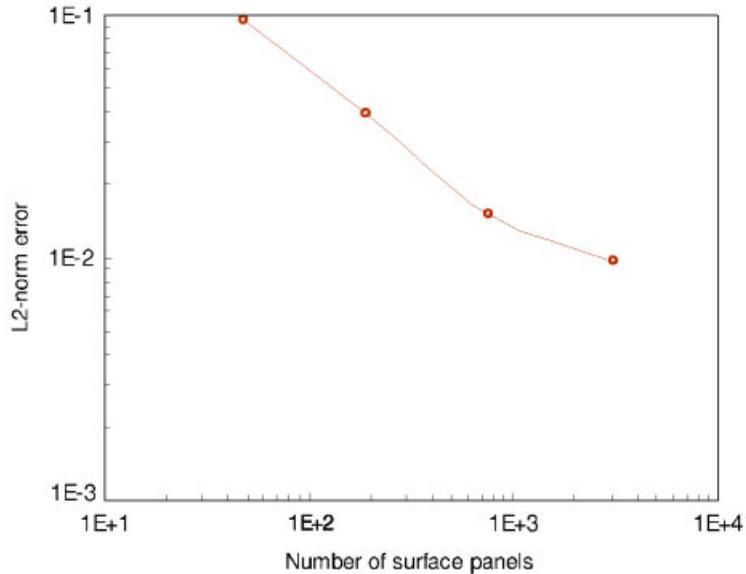


Figure 14.  $L_2$ -norm errors in  $\hat{\partial}u/\hat{\partial}n$  of the Laplace problem.

Approach IV, on the other hand, fails to converge in this example. Despite the great savings in the CPU time, this approach is therefore not recommended. It is worth pointing out that in all the calculations, the order of Gauss quadrature formula used is 3. Obviously, the accuracy will further improve when a higher order of Gauss quadrature formula is used. For complicated source terms, high-order quadrature rules are recommended to ensure accuracy.

Figure 15 gives the CPU times used for solving the Poisson problem using the different approaches. Recall that the only input to the code is the surface mesh. Thus the recorded CPU time includes the time for setting up the FFT grid, generating new mesh, etc. The data points at 3072 for I and II are extrapolated. At coarse discretizations particularly in the cases of 48 and 192, the accelerated schemes are more costly. This is because the additional overhead needed for generating new meshes, performing projection, convolution and interpolation offsets or even overpowers the time saved by the acceleration. When the discretization becomes finer, the savings by the acceleration picks up, resulting in a much less CPU time. Thus, the acceleration scheme is, as expected, more efficient for large-scale problems.

**3.2.2. Example 2—a solid ellipsoid with two spherical exclusions.** This example is similar to the benchmark problem 1 studied in Reference [28] except a solid ellipsoid is used instead of a parallelepiped region (see Figure 16). The lengths of the semi-axes of the solid ellipsoid are  $a=2$ ,  $b=4$ ,  $c=2$ , and the diameter for the spheres is 1 with centres at  $(0, -0.5, 0)$  and  $(0, 1, 0)$ , respectively.

The governing equation is

$$\nabla^2 u = (2y^3 + 6y)e^{(x+z)} \quad (9)$$

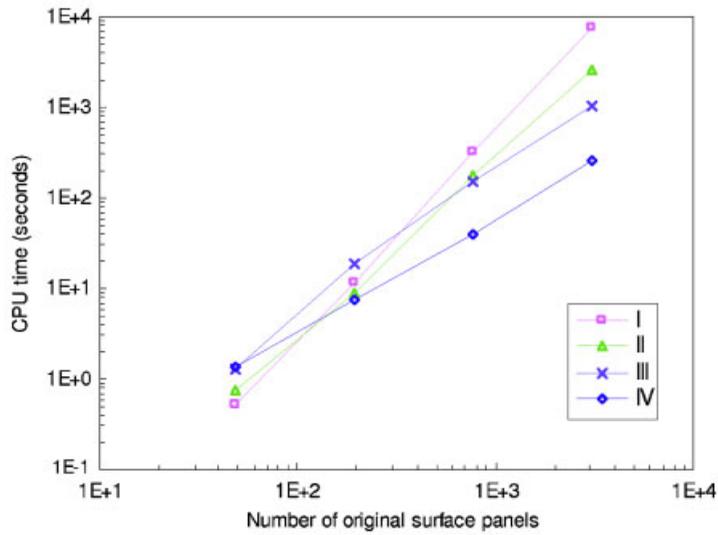


Figure 15. CPU time for solving the Poisson equation on a spherical domain.

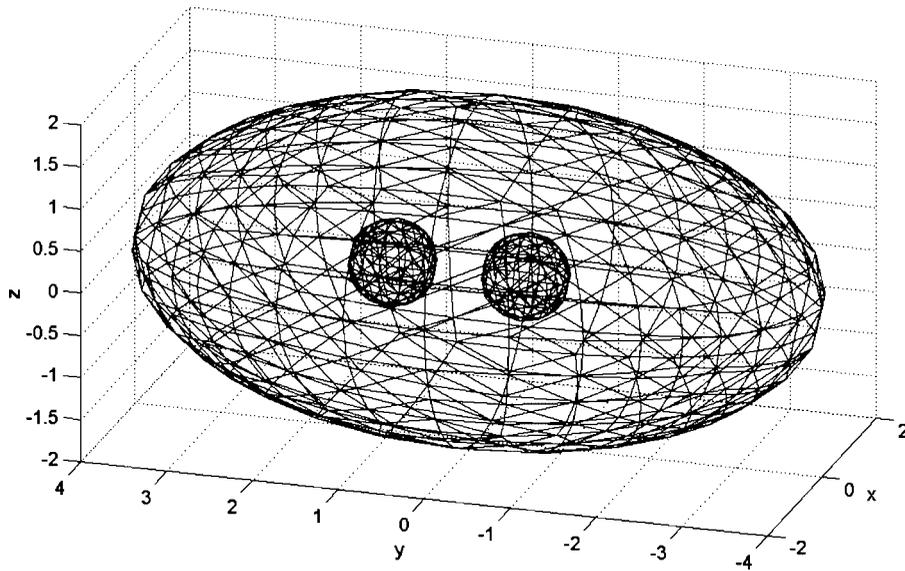


Figure 16. Surface mesh of the problem domain for Example 2—a solid ellipsoid with two spherical exclusions. The total number of surface panels shown is 1152.

One solution to this problem is shown in Equation (10). This solution is used to prescribe the Dirichlet boundary conditions (i.e.  $u = y^3 e^{(x+z)}$ ) at all surfaces (ellipsoid and two spheres). It also serves as the reference for the calculation of errors (measured in  $L_2$ -norm) in the

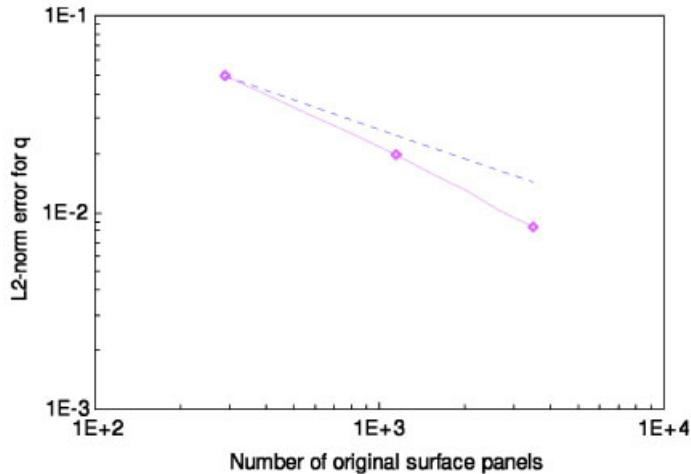


Figure 17. Accuracy and convergence for Poisson Example 2.

simulated  $\partial u / \partial n$

$$u = y^3 e^{(x+z)}, \quad q = \frac{\partial u}{\partial n} = e^{(x+z)} (y^3 n_x + 3y^2 n_y + y^3 n_z) \quad (10)$$

Figure 17 plots the error versus the number of panels as well as a dash line that represents the linear convergence, i.e.  $E \approx O(h)$ . Three discretizations were tested (one is shown in Figure 16). With a total of 3456 constant surface elements (3072 for the ellipsoid and 192 for each sphere), the error is reduced to 0.92% and the associated overall CPU time is 1029 s. The reported error for a similar problem in Reference [28] is around 2.3% and the CPU time is 750 s using the auxiliary domain method (ADM). (The authors have shown in Reference [28] that the ADM performs better than the Dual Reciprocity Method for this particular case.) Although it is difficult to directly compare the performance of our approach and the ADM based on these results, different computers, different boundary conditions, etc. Nevertheless, these results indicate that our approach is accurate and efficient. Again, the convergence rate is around  $O(h)$ .

#### 4. CONCLUSIONS

In this paper, we described an efficient numerical approach for computing volume integrals present in the boundary integral formulations of non-homogeneous linear problems with 3-D complex domains. A 3-D uniform grid constructed initially to accelerate surface integration is used for the evaluation of volume integrals. As such, no volume discretization of the interior problem domain is necessary. In addition, the precorrected-FFT technique is extended to accelerate volume integration, resulting in a much reduced computational time. Several techniques have been developed to handle issues arising from the implementation. In particular a ‘projection+transformation’ scheme is used to accurately evaluate singular volume integrals and to perform integration on partially filled cubes. The accuracy and efficiency of these techniques

are demonstrated through several examples. It has found that the errors caused by the acceleration are very small and more efficiency is achieved when the discretization is finer, indicating the developed approach is suitable for large-scale problems.

An accelerated BEM solver for Poisson equations has been developed based on our approach. In this solver, the only required input is the surface mesh of the problem domain. The accuracy, efficiency and convergence of this solver are first demonstrated through an example with a spherical domain. A 3-D multiply-connected problem was then solved to illustrate the capability of the solver for handling problems with complex geometries. The results were compared with those obtained using the ADM on a similar problem. Although it is difficult to judge which method performs better, due to the differences between the two studied cases, the good performance of our approach is nevertheless clearly demonstrated. In addition the approach developed in this paper is applicable to any domain geometry.

#### ACKNOWLEDGEMENTS

The authors would like to acknowledge valuable discussions with Jacob White at MIT. Work at Georgia Tech was supported by an NSF grant (CCR0306664). This research was also supported in part by the Applied Mathematical Sciences Research Program of the Office of Mathematical, Information, and Computational Sciences, U.S. Department of Energy under contract DE-AC05-00OR22725 with UT-Battelle, LLC.

#### REFERENCES

1. Greengard L. *The Rapid Evaluation of Potential Fields in Particle Systems*. MIT Press: Cambridge, MA, 1988.
2. Greengard L, Rokhlin V. New version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica* 1997; 229–269.
3. Phillips JR, White J. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and System* 1997; **16**(10):1059–1072.
4. Bruno OP, Kunyansky LA. A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests and applications. *Journal of Computational Physics* 2001; **169**(1):80–110.
5. Pierce AP, Napier JAL. A spectral multipole method for efficient solution of large-scale boundary element models in electrostatics. *International Journal for Numerical Methods in Engineering* 1995; **38**:4009–4034.
6. Nabors K, White J. Fastcap: a multipole accelerated 3-D capacitance extraction program. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and System* 1991; **10**(11):1447–1459.
7. Aluru NR, White J. A fast integral equation technique for analysis of microflow sensors based on drag force calculation. *International Conference on Modeling and Simulation of Microsystems* 1998; 283–286. (Available at <http://www.cr.org/publications/index.html>).
8. Ye W, Kanapka J, Wang X, White J. Efficiency and accuracy improvements for faststokes, a precorrected-FFT accelerated 3-D Stokes solver. *International Conference on Modeling and Simulation of Microsystems* 1998; 502–505. (Available at <http://www.cr.org/publications/index.html>).
9. Ye W, Wang X, White J. A fast 3D solver for unsteady flow with applications to micro-electro-mechanical systems. *International Conference on Modeling and Simulation of Microsystems* 1999; 518–521. (Available at <http://www.cr.org/publications/index.html>).
10. Ye W, Wang X, White J. Fast stokes solver for generalized flow problems. *International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators* 2000; 524–527.
11. Silveira LM, Elfadel IM, White JK, Chilukuri M, Kundert KS. Efficient frequency-domain modeling and circuit simulation of transmission lines. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology—Part B: Advanced Packaging, Special issue on Electrical Performance of Electrical Packaging* 1994; **17**(4):505–513.
12. Ye W, Wang X, Hemmert W, Freeman D, White J. Air damping in lateral oscillating micro resonators: a numerical and experimental study. *Journal of Microelectromechanical Systems* 2003; **12**(5):1–10.

13. Nardini D, Brebbia CA. A new approach to free vibration analysis using boundary elements. *Applied Mathematical Modelling* 1983; **7**(3):157–162.
14. Ahmed S, Banerjee PK. Free vibration analysis of BEM using particular integrals. *Journal of Engineering Mechanics* 1986; **112**:682–695.
15. Ingber MS. An efficient boundary element method for a class of parabolic differential equations using discretization in time. *Numerical Methods for Partial Differential Equations* 1987; **3**(3):187–197.
16. Herry D, 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.
17. Nowak AJ, Brebbia CA. The multiple-reciprocity method. A new approach for transforming BEM domain integrals to the boundary. *Engineering Analysis with Boundary Elements* 1989; **6**(3):164–167.
18. Ingber MS, Phan-Thien N. A boundary element approach for parabolic differential equations using a class of particular solutions. *Applied Mathematical Modelling* 1992; **16**(3):124–132.
19. Partridge PW, Brebbia CA, Wrobel LC. *The Dual Reciprocity Boundary Element Method*. Elsevier: London, 1992.
20. Ingber MS. A triple reciprocity boundary element method for transient heat conduction analysis. *International Conference on Boundary Element Technology* 1994; 41–49.
21. Zhang JJ, Afagh FF, Tan CL. Boundary integral transformation of body-force volume integral in BEM for 2D anisotropic elasticity. *Mechanics Research Communications* 1996; **23**(2):227–232.
22. Goldberg MA, Chen CS, Bowman H, Power H. Some comments on the use of radial basis functions in the dual reciprocity method. *Computational Mechanics* 1998; **21**(2):141–148.
23. Katsikadelis JT, Nerantzaki MS. The boundary element methods for nonlinear problems. *Engineering Analysis with Boundary Elements* 1999; **23**:365–373.
24. Ingber MS, Mammoli A, Brown M. A comparison of domain integral evaluation techniques for boundary element methods. *International Journal for Numerical Methods in Engineering* 2001; **52**(4):417–432.
25. Gao XW. The radial integration methods for evaluation of domain integrals with boundary-only discretization. *Engineering Analysis with Boundary Elements* 2002; **26**:905–916.
26. Mammoli AA. Solution of non-linear integral equations in complex geometries with auxiliary integral subtraction. *International Journal for Numerical Methods in Engineering* 2002; **55**:1115–1128.
27. Gao XW. Boundary element analysis in thermoelasticity with and without internal cells. *International Journal for Numerical Methods in Engineering* 2003; **57**(7):975–990.
28. Hsiao SC, Mammoli AA, Ingber MS. The evaluation of domain integrals in complex multiply-connected three-dimensional geometries for boundary element methods. *Computational Mechanics* 2003; **32**:226–233.
29. Nagarajan A, Mukherjee S. Mapping method for numerical evaluation of two-dimensional integrals with  $1/r$  singularity. *Computational Mechanics* 1993; **12**:19–26.
30. Saad Y, Schultz M. GMRES: a generalized minimal residual algorithm for solving symmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:856–869.