ELSEVIER

Engineering Analysis with Boundary Elements 27 (2003) 57–71

ENGINEERING ANALYSIS *with* BOUNDARY ELEMENTS

www.elsevier.com/locate/enganabound

A boundary cloud method with a cloud-by-cloud polynomial basis

Gang Li[a], N.R. Aluru[b,*]

[a]*Department of Mechanical and Industrial Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

[b]*Department of General Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

Received 18 April 2002; revised 6 September 2002; accepted 18 September 2002

Abstract

We have recently presented a boundary cloud method (BCM) [Comput. Meth. Appl. Mech. Engng 191 (2002) 2337], which combines boundary integral formulations with scattered point interpolation techniques. A generalized least-squares approach, which requires information about the outward normal to the boundary, is employed to construct interpolation functions. Since an outward normal is not well defined for geometries with corners for 2D problems (or for corners and edges for 3D problems), points could not be placed at corners when discretizing the surface of the object. In this paper, we introduce a new implementation of the BCM, which uses a varying base interpolating polynomial to construct interpolation functions. The key idea is to define an appropriate polynomial basis which ensures linear completeness. The polynomial basis can change from cloud to cloud depending on the definition of the cloud at each point. The new implementation can handle points at corners and is much simpler and at least an order of magnitude faster compared to our original implementation. The original implementation can be more accurate and can give higher order convergence rates, but is limited because it cannot handle points at corners. Numerical results comparing the original and the new implementation are shown for several potential and electrostatic problems.

Keywords: Boundary cloud method; Least-squares approach; Cloud by cloud; Polynomial basis; Meshless method

1. Introduction

Boundary integral formulations [2], especially when combined with fast algorithms based on multi-pole expansions [3], fast Fourier transform (FFT) [4] and singular value decomposition (SVD) [5,6], are powerful computational techniques for rapid analysis of exterior problems. Boundary integral formulations have the advantage of reducing the original dimensionality of the problem. For example, boundary integral formulations reduce a 2D problem to a 1D problem and a 3D problem to a 2D problem. The computational complexity of fast boundary element methods is typically $O(N \log N)$, where N is the number of panels or elements used to discretize the surface of the object. A major burden in fast boundary element methods is the mesh generation task, which can be very expensive and complicated for complex objects.

Boundary integral formulations when combined with scattered point or meshless techniques (see Refs. [7–9] for overview on meshless methods) are attractive computational methods for exterior problems as they alleviate the mesh generation requirements. Recently, several meshless methods for boundary-only analysis have been proposed in the literature. Some of the methods include the boundary node method [10–12], the hybrid boundary node method [13] and the boundary knot method [14]. The boundary node method is a combined boundary integral/meshless approach for boundary only analysis of partial differential equations. In a boundary node method, given a scattered set of points, meshless interpolation functions are constructed by using a moving least-squares approach [15]. A straight-forward use of Cartesian coordinates in a moving least-squares approach can lead to singular coefficient matrices if the points on the boundary lie along a straight line (see Ref. [1] for a more detailed discussion). The boundary node method circumvents this problem by employing a cyclic coordinate (or curvilinear coordinates

* Corresponding author.
E-mail address: aluru@uiuc.edu (N.R. Aluru).

0955-7997/03/$ - see front matter © 2002 Published by Elsevier Science Ltd.
PII: S0955-7997(02)00110-8

in 3D) in the moving least-squares approach to construct interpolation functions. Recently, we have introduced a boundary cloud method (BCM), which is also a combined boundary-integral/scattered point approach for boundary only analysis of partial differential equations. The BCM employs a generalized least-squares approach to construct interpolation functions. The generalized least-squares approach enables the use of Cartesian coordinates, which is attractive. However, since the approach requires information about the outward normal to the boundary, points could not be placed at the corners (for 2D) or along the edges and corners (for 3D) as the normal at these points is not well-defined.

In this paper, we introduce a new implementation of BCM where points can be placed along corners and edges. The basic idea is to use a varying polynomial basis to eliminate the singularity encountered in the moving least-squares approach when all the points lie along a straight line. Specifically, when a weighting function is centered at a point, a cloud for that point is defined. If all the points in the cloud lie along a straight line (or in the same plane for 3D), the moment matrix becomes singular when a complete linear base interpolating polynomial is used, i.e. $\{1, x, y\}$. However, if the basis is chosen carefully, then the singularity can be avoided. For example, instead of using $\{1, x, y\}$, the use of either $\{1, x\}$ or $\{1, y\}$ depending on the definition of the cloud can ensure the linear completeness as well as eliminate the problems encountered in the moving least-squares approach. Compared to the generalized least-squares approach, the use of a varying polynomial basis is attractive for two reasons: first, since the varying basis approach does not require information about the outward normal, points can be placed at corners (and edges in 3D); second, the varying basis approach is much simpler and faster to use to compute interpolation functions. However, as demonstrated in the results section, the generalized least-squares approach is more accurate and produces higher convergence rates compared to the varying basis approach.

The rest of the paper is outlined as follows: Section 2 summarizes the governing equations and the boundary integral formulations for potential and electrostatic problems; Section 3 summarizes the generalized least-squares approach and introduces the varying basis approach to compute interpolation functions; Section 4 describes the numerical implementation of the method; numerical results for potential and electrostatic problems are shown in Section 5; conclusions are given in Section 6.

2. Governing equations and boundary-integral formulations

We will focus on two-dimensional problems in this paper, but the approach can be extended for three-dimensional problems. The governing equations along with the boundary integral formulations are summarized below.

2.1. Potential problem

Consider an arbitrary domain Ω, as shown in Fig. 1, on which the potential equation along with the boundary conditions is to be solved, i.e.

$$\nabla^2 u = 0 \qquad \text{in } \Omega \tag{1}$$

$$u = g \qquad \text{on } \Gamma_g \tag{2}$$

$$q = \frac{\partial u}{\partial n} = h \qquad \text{on } \Gamma_h \tag{3}$$

where u is the unknown potential; Γ_g, the portion of the boundary where Dirichlet boundary conditions are specified; Γ_h, the portion of the boundary where Neumann boundary conditions are specified; g and h, the prescribed Dirichlet and Neumann boundary conditions, respectively; q, the normal derivative of u; n is the outward normal vector.

A boundary integral equation for the potential problem is given by (see, e.g. Ref. [2] for details)

$$\int_{\mathrm{d}\Omega} \left\{ \ln r(P,Q) q(Q) - \frac{\partial \ln r(P,Q)}{\partial n_Q} [u(Q) - u(P)] \right\} \mathrm{d}S_Q = 0 \tag{4}$$

where $\ln(r)$ is the Green's function for the 2D potential equation, P and Q are the source and field points, respectively, $r(P,Q)$ is the distance between P and Q and n_Q is the outward normal vector at a field point Q on boundary $\mathrm{d}\Omega$.

2.2. Electrostatics

Consider a two conductor system as shown in Fig. 2. Ω_1 and Ω_2 denote the geometry of conductors 1 and 2, respectively, $\mathrm{d}\Omega_1$ and $\mathrm{d}\Omega_2$ denote the surface or boundary of conductors 1 and 2, respectively, and $\bar{\Omega}$ is the domain exterior to Ω_1 and Ω_2. A potential of g_1 is applied on conductor 1 and a potential of g_2 is applied on conductor 2. The objective is to compute the surface charge density on

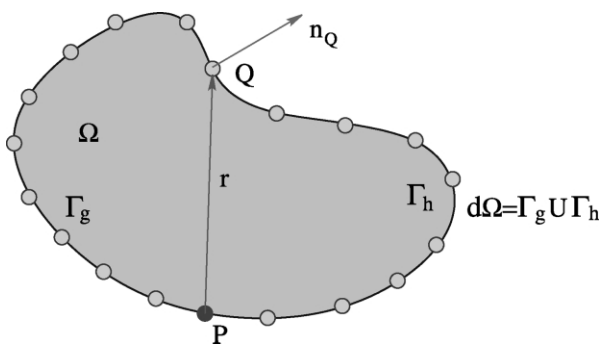

Fig. 1. An arbitrary domain on which the potential equation is to be solved.

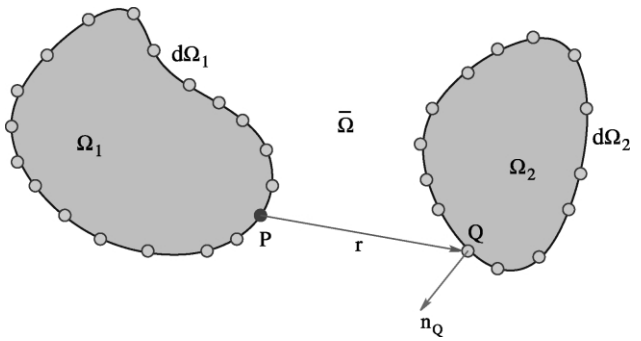

Fig. 2. A two conductor electrostatic system.

the two conductors. The governing equation along with the boundary conditions for the exterior electrostatic problem is given by [16]

$$\nabla^2\phi = 0 \qquad \text{in } \bar{\Omega} \tag{5}$$

$$\phi = g_1 \qquad \text{on } d\Omega_1 \tag{6}$$

$$\phi = g_2 \qquad \text{on } d\Omega_2 \tag{7}$$

A boundary integral equation for the electrostatic problem is given by [17]

$$\phi(P) = \int_{d\Omega} G(P,Q)\sigma(Q)dS_Q + C \tag{8}$$

$$\int_{d\Omega} \sigma(Q)dS_Q = C_T \tag{9}$$

where σ is the unknown surface charge density; P, the source point; Q, the field point; G is the Green's function and $d\Omega = d\Omega_1 \cup d\Omega_2$. In 2D electrostatics, $G(P,Q) = -\ln|P-Q|/(2\pi\epsilon_0)$, where $|P-Q|$ is the distance between the source point P and the field point Q, ϵ_0 is the permittivity of the free space, C_T is the total charge of the system ($C_T = 0$) and C is an unknown variable which needs to be computed. The unknown variable C arises because the potential at infinity is not zero for two-dimensional problems.

3. Scattered point interpolation

The first step in the BCM is to construct interpolation functions over a scattered set of points. As shown in Fig. 1, the surface of the object is represented by scattered points. The objective is to approximate unknown quantities by interpolation functions. The Hermite-type approach that was introduced in Ref. [1] is first quickly summarized before the varying basis approach is presented to construct interpolation functions.

3.1. Hermite-type interpolation

In a Hermite-type interpolation, given a point t, the unknown and its normal derivative in the vicinity of the point t are approximated by

$$u(x,y) = \mathbf{p}^T(x,y)\mathbf{a}_t \tag{10}$$

$$q(x,y) = \frac{\partial \mathbf{p}^T(x,y)}{\partial n}\mathbf{a}_t \tag{11}$$

where $\mathbf{p}$ is the base interpolating polynomial; $\mathbf{a}_t$, the unknown coefficient vector for point t; n is the direction of the outward normal to the boundary (note that n can be different at every point). The linear basis interpolating polynomial and its normal derivatives are given by

$$\mathbf{p}^T(x,y) = [\,1 \quad x \quad y\,] \qquad m = 3 \tag{12}$$

$$\frac{\partial \mathbf{p}^T(x,y)}{\partial n} = \begin{bmatrix} 0 & \frac{\partial x}{\partial n} & \frac{\partial y}{\partial n} \end{bmatrix} = [\,0 \quad \cos(\hat{nx}) \quad \cos(\hat{ny})\,] \qquad m = 3 \tag{13}$$

where $(\hat{nx})$ and $(\hat{ny})$ are the angles between the outward normal direction and the positive x- and y-axis, respectively.

For a point t, the unknown coefficient vector $\mathbf{a}_t$ is computed by minimizing

$$J_t = \sum_{i=1}^{NP} w_i(x_t,y_t)[\mathbf{p}^T(x_i,y_i)\mathbf{a}_t - \hat{u}_i]^2 + \sum_{i=1}^{NP} \tilde{w}_i(x_t,y_t) \times \left[\frac{\partial \mathbf{p}^T(x_i,y_i)}{\partial n}\mathbf{a}_t - \hat{q}_i\right]^2 \tag{14}$$

where NP is the number of nodes, $w_i(x_t,y_t)$ is the weighting function centered at (x_t,y_t) and evaluated at node i whose coordinates are (x_i,y_i). The weighting function for the second term in Eq. (14) is different from the weighting function for the first term to account for dimensionality analysis. Typically, $\tilde{w}_i = w_i$ for convenience. This means that the scaling factor is set to one. $\hat{u}_i$ and $\hat{q}_i$ are nodal parameters. The weighting function is nonzero when the location of node i is within a certain distance from point (x_t,y_t). Thus, for a point (x_t,y_t), the weighting function is nonzero only for a few other nodes in the vicinity of it. The region where the weighting function is nonzero is called a cloud. The weighting function is typically a cubic spline or a Gaussian function. In this paper, a cubic spline is used whose definition is given by

$$w_i(x_t) = \frac{1}{d_x}\hat{w}\left(\frac{x_t - x_i}{d_x}\right) \tag{15}$$

$$\hat{w}(z) = \begin{cases} 0 & z < -2 \\ (z)^3/6 & -2 \le z \le -1 \\ 2/3 - z^2(1 + z/2) & -1 \le z \le 0 \\ 2/3 - z^2(1 - z/2) & 0 \le z \le 1 \\ -(z)^3/6 & 1 \le z \le 2 \\ 0 & z > 2 \end{cases} \tag{16}$$

where $z = (x_t - x_i)/d_x$ and d_x denotes the support size of the weighting function in the x-direction. A multi-dimensional weighting function can be constructed as a product of one-dimensional weighting functions. In two dimensions, the weighting function is given by

$$w_i(x_t, y_t) = \frac{1}{d_x}\hat{w}\left(\frac{x_t - x_i}{d_x}\right)\frac{1}{d_y}\hat{w}\left(\frac{y_t - y_i}{d_y}\right) \tag{17}$$

where d_y is the support size in the y direction.

By minimizing J_t the necessary interpolation functions can be computed. Once the interpolation functions are computed, the unknowns are evaluated by

$$u(x,y) = \sum_{I=1}^{\mathrm{NP}} \tilde{M}_I(x,y)\hat{u}_I + \sum_{I=1}^{\mathrm{NP}} \tilde{N}_I(x,y)\hat{q}_I \tag{18}$$

$$q(x,y) = \sum_{I=1}^{\mathrm{NP}} \tilde{S}_I(x,y)\hat{u}_I + \sum_{I=1}^{\mathrm{NP}} \tilde{T}_I(x,y)\hat{q}_I \tag{19}$$

where $\tilde{M}_I(x,y)$, $\tilde{N}_I(x,y)$, $\tilde{S}_I(x,y)$ and $\tilde{T}_I(x,y)$ are the interpolation functions and are given by

$$\tilde{M}_I(x,y) = \mathbf{p}^{\mathrm{T}}(x,y)\mathbf{C}_t^{-1}\mathbf{p}(x_I,y_I)w_I(x_t,y_t) \tag{20}$$

$$\tilde{N}_I(x,y) = \mathbf{p}^{\mathrm{T}}(x,y)\mathbf{C}_t^{-1}\mathbf{p}'(x_I,y_I)w_I(x_t,y_t) \tag{21}$$

$$\tilde{S}_I(x,y) = \frac{\partial \mathbf{p}^{\mathrm{T}}(x,y)}{\partial n}\mathbf{C}_t^{-1}\mathbf{p}(x_I,y_I)w_I(x_t,y_t) \tag{22}$$

$$\tilde{T}_I(x,y) = \frac{\partial \mathbf{p}^{\mathrm{T}}(x,y)}{\partial n}\mathbf{C}_t^{-1}\mathbf{p}'(x_I,y_I)w_I(x_t,y_t) \tag{23}$$

Note that $\tilde{M}_I(x,y)$, $\tilde{N}_I(x,y)$, $\tilde{S}_I(x,y)$ and $\tilde{T}_I(x,y)$ are multi-valued. We limit the interpolation functions to a single value by computing the interpolations at the point where the weighting function is centered (see Refs. [1,19,20] for details).

3.2. Varying basis least-squares approximation

In this section, we introduce a varying basis least-squares approach to approximate the unknown quantity, i.e.

$$u(x,y) = \mathbf{p}_v^{\mathrm{T}}(x,y)\mathbf{b}_t \tag{24}$$

where $\mathbf{p}_v$ is the varying base interpolating polynomial and $\mathbf{b}_t$ is the unknown coefficient vector for point t. To construct the varying basis interpolation functions, clouds are classified into two types: singular and nonsingular. When all the points inside a cloud lie along a straight line, the cloud is defined as singular, otherwise, it is nonsingular. As shown in Fig. 3, the cloud of point 1 is singular and the cloud for point 2 is nonsingular as the points do not lie along a straight line. In this paper, we use a linear polynomial basis. The base interpolating polynomial is given by

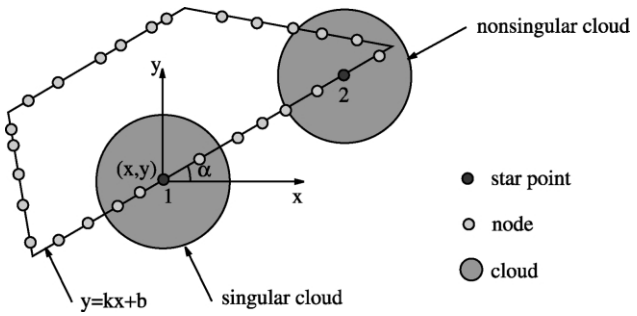

Fig. 3. Definition of singular and nonsingular cloud.

$$\mathbf{p}_v^{\mathrm{T}}(x,y) = \begin{cases} [\,1 \quad x \quad y\,] & m = 3 \text{ (basis for nonsingular cloud)} \\ [\,1 \quad x\,] \text{ or } [\,1 \quad y\,] & m = 2 \text{ (basis for singular cloud)} \end{cases} \tag{25}$$

For a point t, the unknown coefficient vector $\mathbf{b}_t$ is computed by minimizing

$$J_t = \sum_{i=1}^{\mathrm{NP}} w_i(x_t,y_t)[\mathbf{p}_v^{\mathrm{T}}(x_i,y_i)\mathbf{b}_t - \hat{u}_i]^2 \tag{26}$$

where NP is the number of nodes, $w_i(x_t,y_t)$ is the weighting function centered at (x_t,y_t) and evaluated at node i whose coordinates are (x_i,y_i) and $\hat{u}_i$ is a nodal parameter. The weighting function is again the cubic spline defined in Eqs. (15)–(17).

The stationary of J_t leads to

$$(\mathbf{P}_v^{\mathrm{T}}\mathbf{W}\mathbf{P}_v)\mathbf{b}_t = \mathbf{P}_v^{\mathrm{T}}\mathbf{W}\hat{\mathbf{u}} \tag{27}$$

Eq. (27) can be rewritten as

$$\mathbf{C}_t'\mathbf{b}_t = \mathbf{A}_t'\hat{\mathbf{u}} \tag{28}$$

$$\mathbf{b}_t = \mathbf{C}_t'^{-1}\mathbf{A}_t'\hat{\mathbf{u}} \tag{29}$$

where $\mathbf{C}_t'$ is an $m \times m$ matrix and $\mathbf{A}_t'$ is an $m \times \mathrm{NP}$ matrix, whose definitions are given by

$$\mathbf{C}_t' = \mathbf{P}_v^{\mathrm{T}}\mathbf{W}\mathbf{P}_v \tag{30}$$

$$\mathbf{A}_t' = \mathbf{P}_v^{\mathrm{T}}\mathbf{W} \tag{31}$$

where

$$\mathbf{P}_v = \begin{bmatrix} \mathbf{p}_v^{\mathrm{T}}(x_1, y_1) \\ \mathbf{p}_v^{\mathrm{T}}(x_2, y_2) \\ \vdots \\ \mathbf{p}_v^{\mathrm{T}}(x_{\mathrm{NP}}, y_{\mathrm{NP}}) \end{bmatrix}, \qquad \hat{\mathbf{u}} = \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_{\mathrm{NP}} \end{bmatrix},$$

$$\mathbf{W} = \begin{bmatrix} w_1(x_t, y_t) & 0 & 0 & 0 \\ 0 & w_2(x_t, y_t) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & w_{\mathrm{NP}}(x_t, y_t) \end{bmatrix} \tag{32}$$

As shown in Ref. [1], when a cloud is singular, the moment matrix $\mathbf{C}_t'$ becomes singular if the full polynomial basis $[1\ x\ y]$ is used. In such cases, the appropriate base interpolating polynomial can be determined from the information of points contained in the cloud. For example, if all the points in the singular cloud are defined by a line $y = \text{constant}$, then $\mathbf{p}_v^{\mathrm{T}} = [1\ x]$ can be used. If the singular cloud is defined by a line $x = \text{constant}$, then $\mathbf{p}_v^{\mathrm{T}} = [1\ y]$ can be used. In the general case of a cloud defined by $y = ax + b$, either $\mathbf{p}_v^{\mathrm{T}} = [1\ x]$ or $\mathbf{p}_v^{\mathrm{T}} = [1\ y]$ can be used. Substituting the definition of $\mathbf{b}_t$ from Eq. (29) into Eq. (24), the approximation for the unknown can be rewritten as

$$u(x, y) = \mathbf{p}_v^{\mathrm{T}}(x, y)\mathbf{b}_t = \mathbf{p}_v^{\mathrm{T}}(x, y)\mathbf{C}_t'^{-1}\mathbf{A}_t'\hat{\mathbf{u}} \tag{33}$$

In short form,

$$u(x, y) = \mathbf{N}(x, y)\hat{\mathbf{u}} \tag{34}$$

where $\mathbf{N}(x, y)$ is $1 \times \mathrm{NP}$ interpolation vector given by

$$\mathbf{N}(x, y) = \mathbf{p}_v^{\mathrm{T}}(x, y)\mathbf{C}_t'^{-1}\mathbf{A}_t' \tag{35}$$

$\mathbf{N}(x, y)$ is again multi-valued and the interpolation function can be limited to a single value by computing the interpolations at the point where the weighting function is centered. Eq. (35) can be rewritten in a matrix form as,

$$\mathbf{u} = \mathbf{N}\hat{\mathbf{u}} \tag{36}$$

where $\mathbf{N}$ is an $\mathrm{NP} \times \mathrm{NP}$ interpolation matrix. At any point (x, y), the unknowns are evaluated by

$$u(x, y) = \sum_{I=1}^{\mathrm{NP}} N_I(x, y)\hat{u}_I \tag{37}$$

where

$$N_I(x, y) = \mathbf{p}_v^{\mathrm{T}}(x, y)\mathbf{C}_t'^{-1}\mathbf{p}_v(x_I, y_I)w_I(x_t, y_t) \tag{38}$$

The normal derivative of u can be interpolated by using the same interpolation function, i.e.

$$q(x, y) = \sum_{I=1}^{\mathrm{NP}} N_I(x, y)\hat{q}_I \tag{39}$$

Remarks.

1. A fixed least-squares approach [18] is used to illustrate the construction of interpolation functions by a varying basis approach. A moving least-squares approach can also be used to define interpolation functions. The interpolation functions constructed by either the fixed least-squares approach or by the moving least-squares approach are identical. However, the derivatives of the interpolation function can be different. Since the derivatives of the interpolation function are not required for the work presented in the paper, both moving least-squares and fixed least-squares approaches produce identical results.
2. Both the Hermite-type approach and the varying basis approach avoid singular moment matrices when all the points in the cloud lie along a straight line. Unlike the Hermite-type approach, the varying basis approach overcomes the difficulty by appropriately redefining the basis for the singular clouds.
3. Both the Hermite-type and the varying basis interpolations functions are meshless interpolation functions. No connectivity information among the scattered points is required to compute the interpolation functions.
4. Both the Hermite-type and the varying basis interpolants are consistent, i.e. any linear combination of the basis polynomial can be interpolated exactly. The varying basis interpolants are consistent even when the basis is reduced. For example, assigning to each $\hat{u}_i$ the values of basis functions gives

   $$\hat{\mathbf{u}} = \mathbf{P}_v \tag{40}$$

   Substituting Eq. (40) to Eq. (33)

   $$u(x, y) = \mathbf{p}_v^{\mathrm{T}}(x, y)\mathbf{C}_t'^{-1}\mathbf{A}_t'\hat{\mathbf{u}} = \mathbf{p}_v^{\mathrm{T}}(x, y)\mathbf{C}_t'^{-1}\mathbf{P}_v^{\mathrm{T}}\mathbf{W}\mathbf{P}_v$$
   $$= \mathbf{p}_v^{\mathrm{T}}(x, y)\mathbf{C}_t'^{-1}\mathbf{C}_t' = \mathbf{p}_v^{\mathrm{T}}(x, y) \tag{41}$$

   Without losing generality, assume a singular cloud is represented by a reduced basis $\mathbf{p}_v^{\mathrm{T}} = [1\ x]$ when all the points in the cloud lie along a straight line defined by $y = kx$. Assuming the unknown u takes the form $u = a + bx + cy$ (a, b and c are constants), for the points in the singular cloud, u is given by $u = a + (b + kc)x$. Since any linear combination of $\mathbf{p}_v^{\mathrm{T}}(x, y) = [1\ x]$ can be interpolated exactly, the unknown u is interpolated exactly by the reduced basis $[1\ x]$.
5. The varying basis least-squares technique is a much simpler approach compared to the Hermite-type approach. The varying basis approach is a much faster technique to compute interpolation functions compared to the Hermite-type approach. For a typical potential problem with 512 nodes, the solution of the new implementation is about 10 times faster than the original Hermite-type approach.
6. In a Hermite-type approach, the outward normal to the boundary is required to compute interpolation functions. Hence, points could not be placed at corners (or edges in 3D) where the normal to the boundary is not

well-defined. A varying basis approach eliminates this problem as it does not require the outward normal and points can be placed at corners (and edges in 3D).

7. In a Hermite-type approach, the normal derivative of the unknown (i.e. q) is also interpolated exactly when the unknown function (i.e. u) is interpolated exactly. For example, if the base interpolating polynomial is linear, and if u is linear then a constant q is interpolated exactly. If q is constant, but is discontinuous from one boundary edge to the other boundary edge, it can still be interpolated exactly in a Hermite-type approach. A varying basis approach does not possess this property and typically error is introduced in approximating a constant q. An example demonstrating this aspect is discussed in Section 5.

4. Numerical implementation

As described in Ref. [1], the boundary of the domain is discretized into cells for integration purpose. Each cell contains a certain number of nodes and the number of nodes can vary from cell to cell. The concept of a cell is different from that of element or panel in boundary-element methods. The cell can be of any shape or size and the only restriction is that the union of all the cells equal to the boundary of the domain.

4.1. Potential problem

The discretized boundary integral equation for the potential problem is given by

$$\sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} \ln r(P,Q_k) q(Q_k) \mathrm{d}S_Q - \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} \frac{\partial \ln r(P,Q_k)}{\partial n_{Q_k}} \times u(Q_k)\mathrm{d}S_Q + u(P) \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} \frac{\partial \ln r(P,Q_k)}{\partial n_{Q_k}} \mathrm{d}S_Q = 0 \quad (42)$$

where NC is the total number of cells and Q_k is the field point in cell k.

Substituting Eqs. (37) and (39) into Eq. (42) for each node, the final matrix form of the BCM with a varying basis is given by

$$\mathbf{F}\hat{\mathbf{u}} = \mathbf{G}\hat{\mathbf{q}} \quad (43)$$

$$\mathbf{u} = \mathbf{N}\hat{\mathbf{u}} \quad (44)$$

$$\mathbf{q} = \mathbf{N}\hat{\mathbf{q}} \quad (45)$$

where $\mathbf{F}$ and $\mathbf{G}$ are NP × NP matrices given by

$$F(i,j) = \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} \frac{\partial \ln r(P_i,Q_k)}{\partial n_{Q_k}} N_j(Q_k)\mathrm{d}S_Q - N_j(P_i) \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} \frac{\partial \ln r(P_i,Q_k)}{\partial n_{Q_k}} \mathrm{d}S_Q \quad (46)$$

$$G(i,j) = \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} \ln r(P_i,Q_k) N_j(Q_k)\mathrm{d}S_Q \quad (47)$$

In a general mixed boundary value problem, either u or q is specified at each point. By substituting all the known quantities given by the boundary conditions into a vector $\mathbf{x}$, a 2NP × 2NP linear system is obtained

$$\begin{bmatrix} \mathbf{F} & -\mathbf{G} \\ \mathbf{N}' & \mathbf{N}'' \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix} \quad (48)$$

where $\mathbf{N}'$ and $\mathbf{N}''$ are obtained by appropriately selecting the rows of $\mathbf{N}$ corresponding to the known us and qs. $\hat{\mathbf{u}}$ and $\hat{\mathbf{q}}$ are computed by solving the above linear system. Once $\hat{\mathbf{u}}$ and $\hat{\mathbf{q}}$ are known, $\mathbf{u}$ and $\mathbf{q}$ can be computed from Eqs. (44) and (45).

4.2. Electrostatic problem

The discretized boundary integral equation for exterior electrostatics is given by

$$\phi(P) = \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} G(P,Q_k)\sigma(Q_k)\mathrm{d}S_Q + C \quad (49)$$

$$\sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} \sigma(Q_k)\mathrm{d}S_Q = C_T \quad (50)$$

The varying basis interpolation of the surface charge density is given by

$$\sigma(x,y) = \sum_{I=1}^{\mathrm{NP}} N_I(x,y)\hat{\sigma}_I \quad (51)$$

Substituting Eq. (51) into Eqs. (49) and (50) for each source node, the final matrix form is given by

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (52)$$

where

$$\begin{cases} A(i,j) = \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} G(P_i,Q_k)N_j(Q_k)\mathrm{d}S_Q & i = 1,\ldots,\mathrm{NP} \\ A(\mathrm{NP}+1,j) = \sum_{k=1}^{\mathrm{NC}} \int_{\mathrm{d}S_k} N_j(Q_k)\mathrm{d}S_Q & j = 1,\ldots\mathrm{NP} \\ A(i,\mathrm{NP}+1) = 1 & i = 1,\ldots,\mathrm{NP} \\ A(\mathrm{NP}+1,\mathrm{NP}+1) = 0 \end{cases} \quad (53)$$

$$\mathbf{x} = \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{\mathrm{NP}} \\ C \end{Bmatrix}, \qquad \mathbf{b} = \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \sigma_{\mathrm{NP}} \\ C_T \end{Bmatrix} \quad (54)$$

The integrals in Eqs. (46), (47), and (53) are weakly singular and near singular when the source point P and the field point

Q coincide or very close to each other, respectively. Regular Gauss quadrature becomes inaccurate in these cases. Special treatment is employed to compute the various integrals. Given a source point P, cells are classified into three categories: (1) for cells which are far away from the source point, a regular Gauss quadrature is used to compute the integral, (2) for cells which are nearby to the source point, the weights of the nodes are computed more accurately by using the Nyström [21] and the SVD [22] techniques, and (3) for a cell which includes the source point, the integral is computed analytically. Extensive details on computing these integral can be found in Ref. [1].

5. Numerical examples

In this section we present results for several problems using the BCM with varying base interpolating polynomial. The results are compared with the results of BCM with Hermite-type interpolations. Unless otherwise mentioned we use the 2D cubic spline weighting function as given in Eq. (17). The convergence of the method is measured by using a global error measure

$$\epsilon = \frac{1}{|u^{(e)}|_{\max}} \sqrt{\frac{1}{\mathrm{NP}} \sum_{I=1}^{\mathrm{NP}} [u_I^{(e)} - u_I^{(c)}]^2} \tag{55}$$

where ϵ is the error in the solution and the superscripts (e) and (c) denote, respectively, the exact and the computed solutions.

5.1. 2D Potential examples

Consider the solution of the potential equation on various domains as shown in Fig. 4. Fig. 4(a) is a 1×1 square

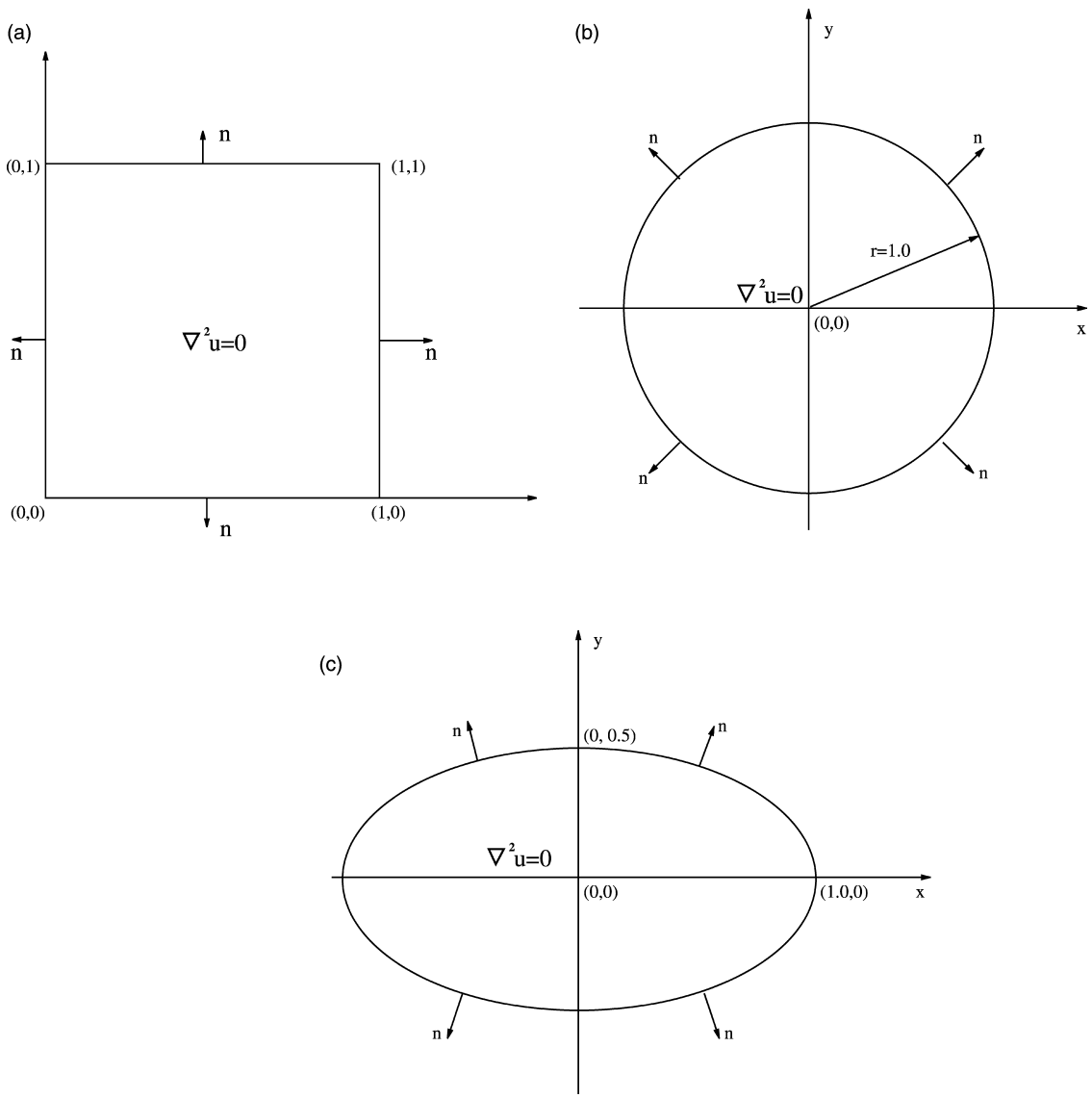

Fig. 4. (a) 1×1 square domain on which the potential equation is solved, (b) a circular domain, and (c) an elliptical domain.

domain where Dirichlet or Neumann boundary conditions are specified along the four edges of the boundary. Four examples with different boundary conditions are presented. For a uniform distribution of nodes with a linear basis, the cloud sizes are chosen to be $r_x = r_y = 1.17\Delta s$, where Δs is the spacing between the points. A uniform cell structure is used. Fig. 4(b) is a circular domain with a radius of 1.0. A Dirichlet problem is solved over the circular domain. Nodes are uniformly distributed along the boundary of the domain. A uniform cell structure with two Gauss points per cell is considered. The cloud sizes are chosen to be $r_x = r_y = 1.17$ Δs where Δs is the perimeter of the circle divided by the number of points. Fig. 4(c) is an elliptical domain and the boundary of the domain is described by

$$x = a\cdot\cos(\theta), \qquad y = b\cdot\sin(\theta) \tag{56}$$

where $a = 1.0$ and $b = 0.5$. A Neumann problem is solved over the elliptical domain. The cell structure and the cloud size are identical to those used for the circular domain.

The first example is a Dirichlet problem on the square domain with a linear exact solution for u and a constant exact solution for q. Dirichlet boundary conditions are specified along the four edges of the boundary. The boundary conditions are

$$u = x + y \qquad \text{for } x = 0,\ y = 0,\ x = 1,\ y = 1 \tag{57}$$

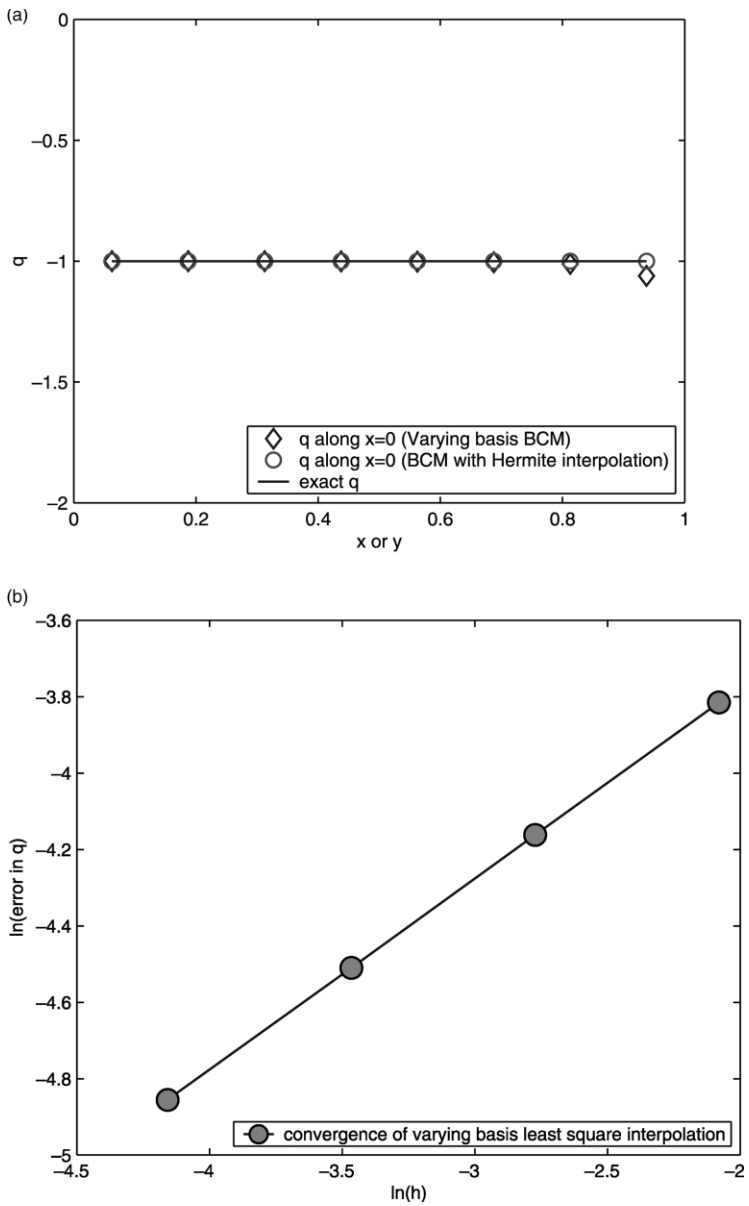

Fig. 5. (a) Comparison of the computed and exact solution, (b) convergence of q with varying basis BCM. Note that no points are put at the corners. The boundary is defined between 0 and 1 and no points are put at either 0 or 1. Even though points can be put at the corners with the new implementation, no points were put at the corners in order to compare the solution with the original implementation.

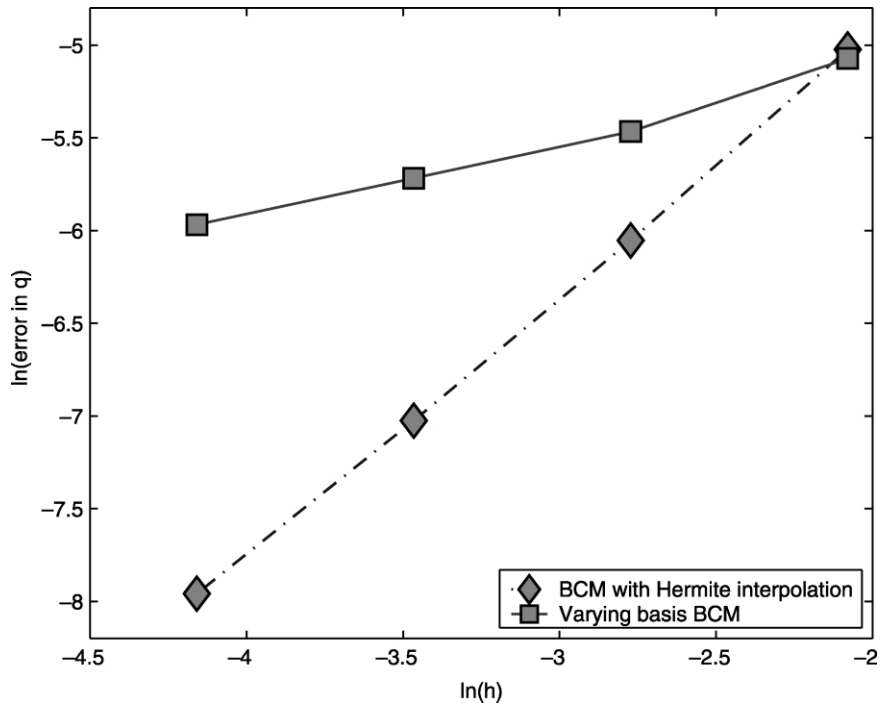

Fig. 6. Comparison of convergence of BCM with varying basis interpolation and Hermite-type interpolation for a Dirichlet problem with quadratic exact solution for u.

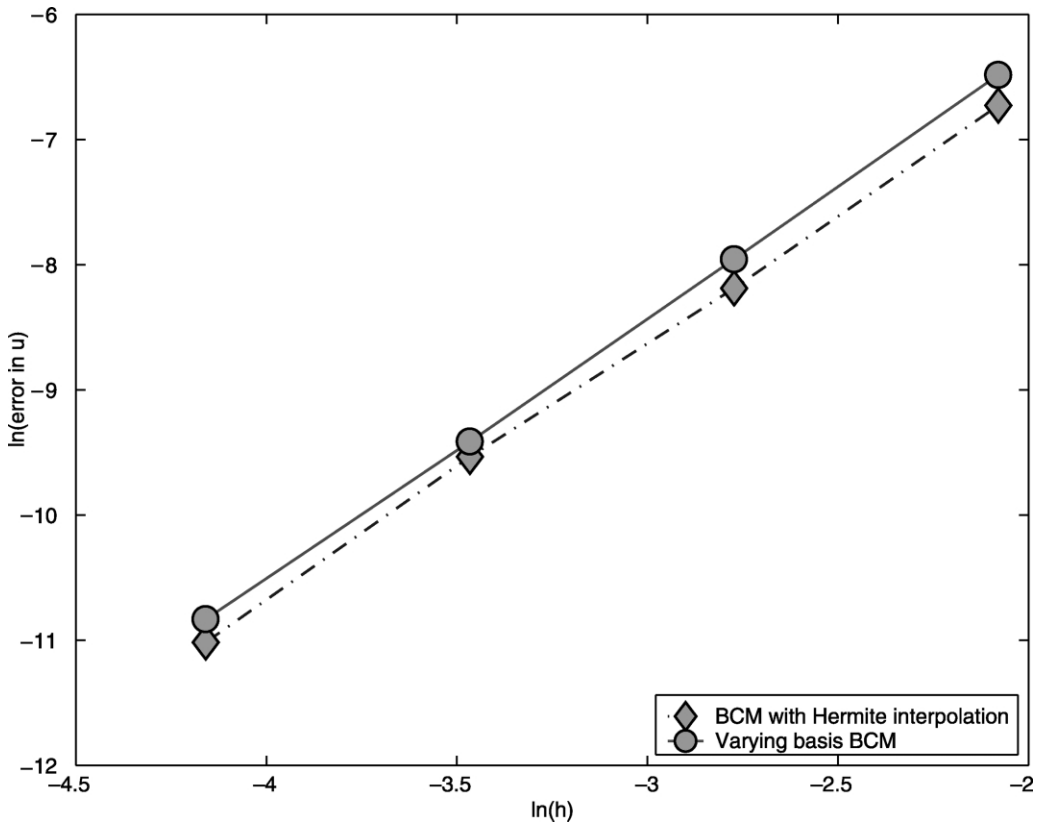

Fig. 7. Comparison of convergence of BCM with varying basis interpolation and Hermite-type interpolation for a Neumann problem with cubic exact solution for u.

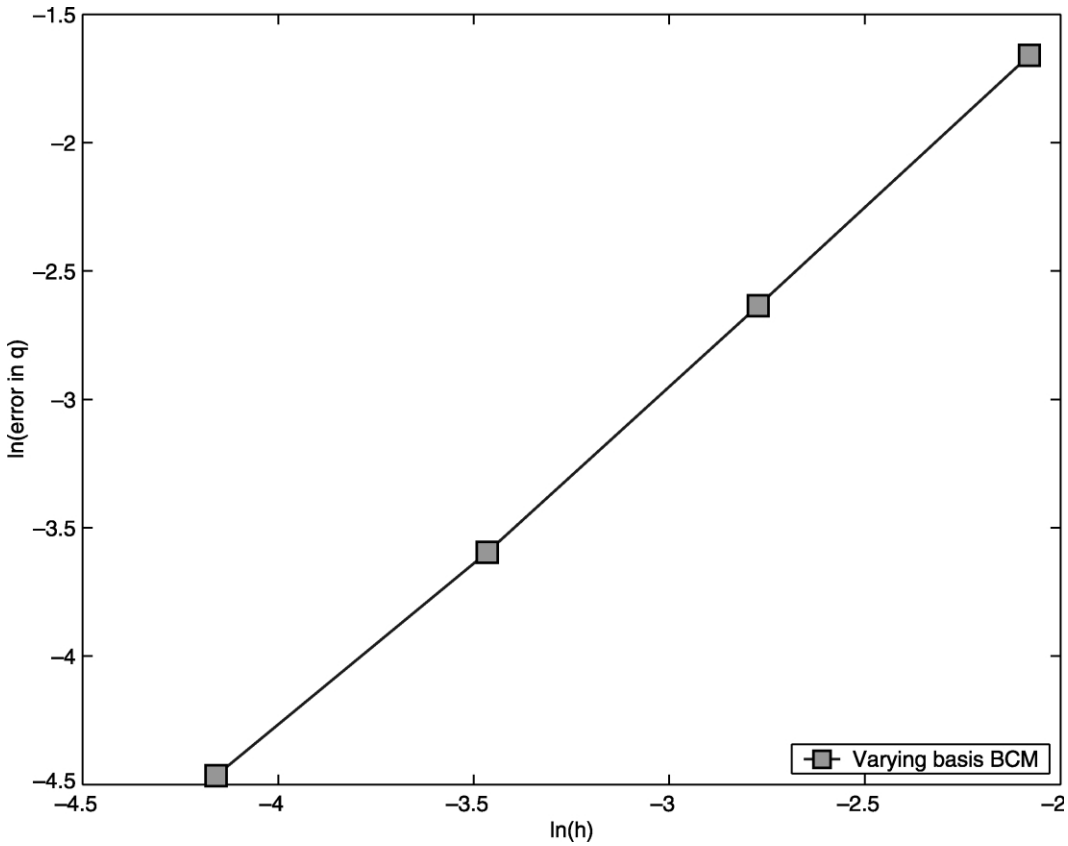

Fig. 8. Convergence of BCM with varying basis interpolation and a random point distribution for a Dirichlet problem with quadratic exact solution for u.

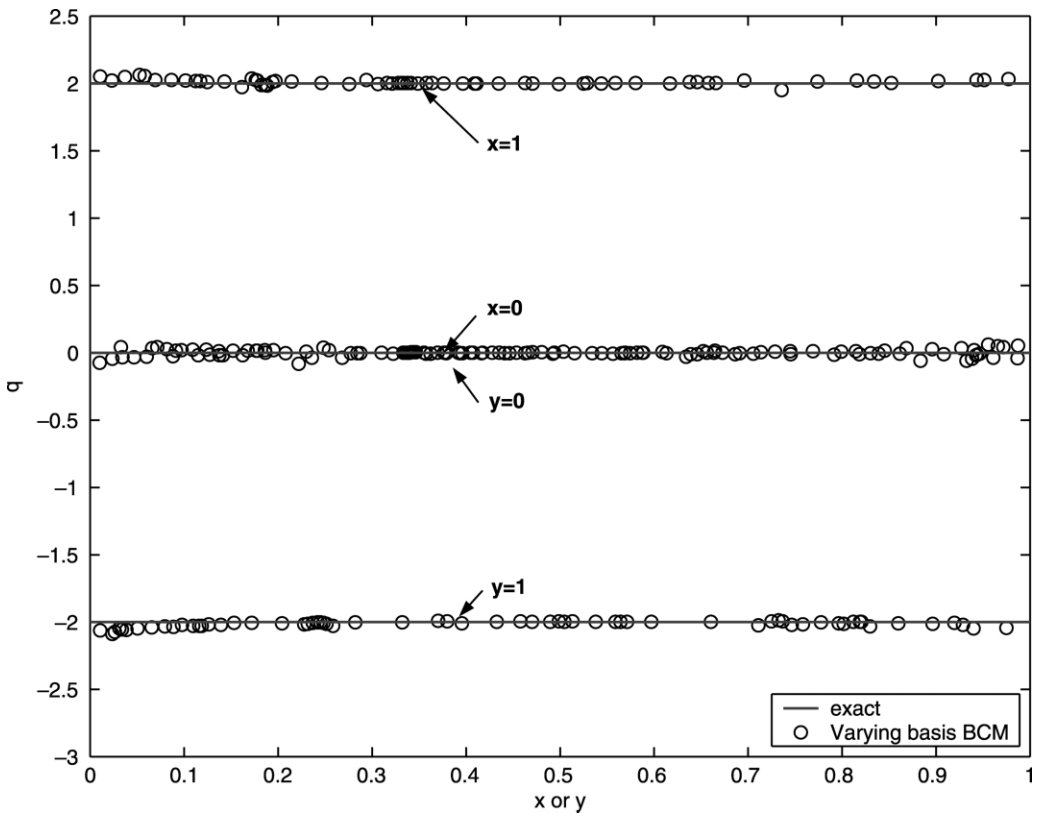

Fig. 9. Comparison of results of BCM with varying basis interpolation and exact solution for a Dirichlet problem with quadratic exact solution for u.

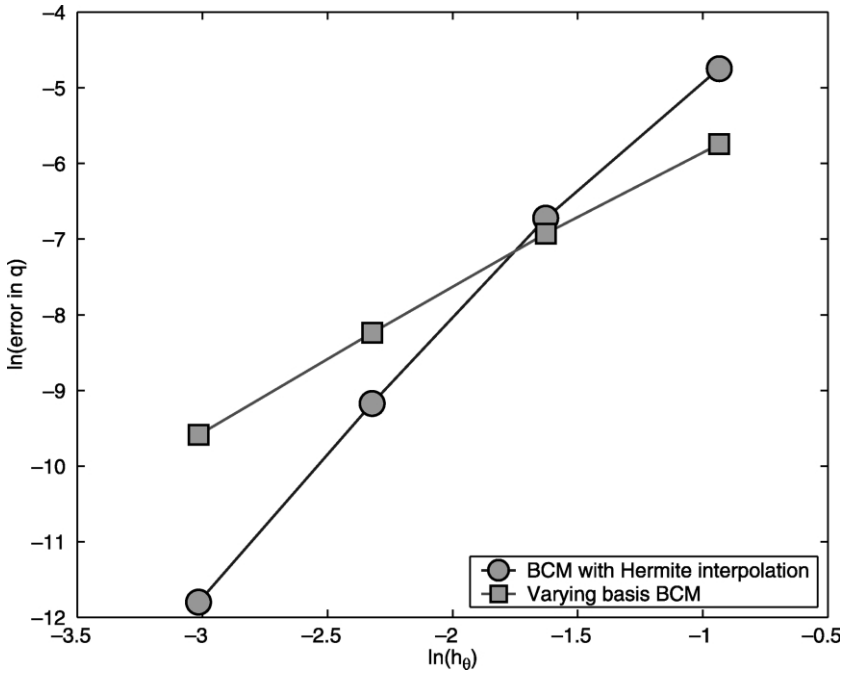

Fig. 10. Convergence of BCM for a Dirichlet problem on a circular domain.

The exact solution for q is given by

$$q = \frac{\partial u}{\partial n} = \begin{cases} -1 & \text{for } y = 0, \ \ x = 0 \\ 1 & \text{for } x = 1, \ \ y = 1 \end{cases} \tag{58}$$

Note that the normal derivative of u is discontinuous at points $(0,1)$ and $(1,0)$. At these two points, q jumps from -1 to 1. For this example, Gaussian quadrature is used for all the cells. However, the weights of the Gauss points are

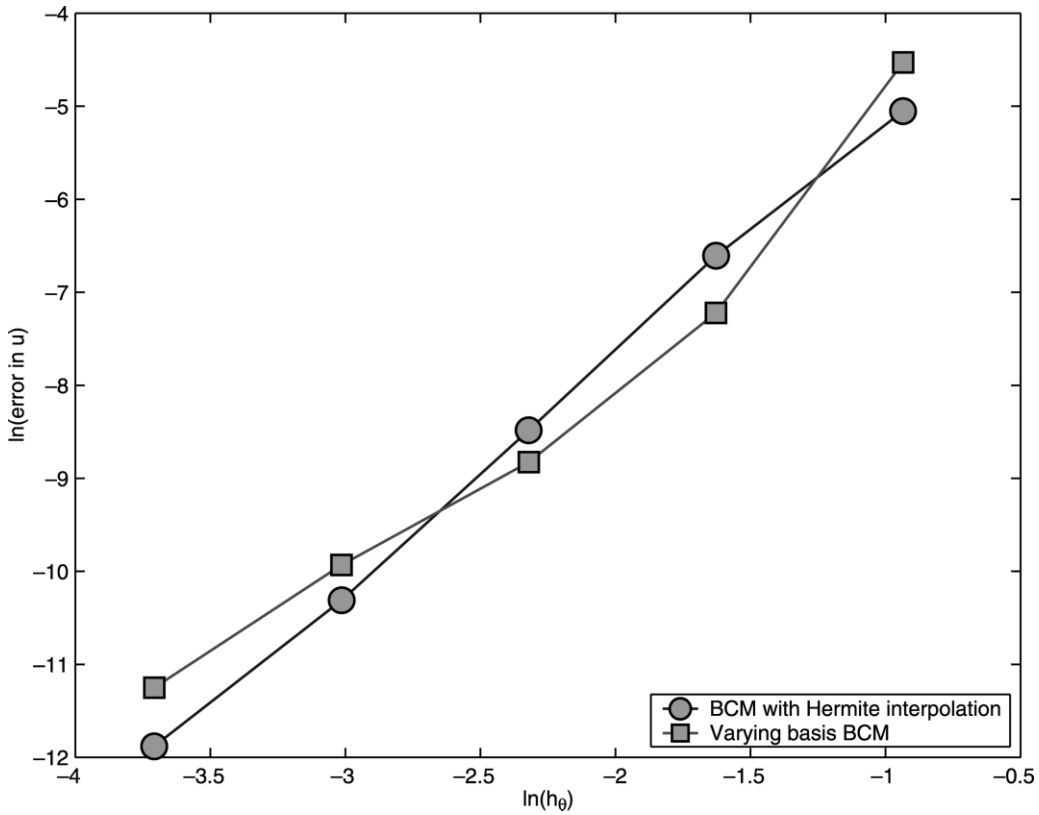

Fig. 11. Convergence of BCM for a Neumann problem on an elliptical domain.

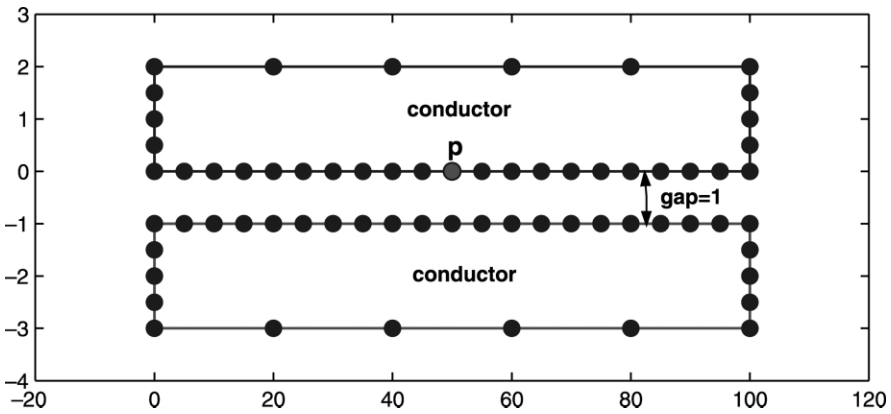

Fig. 12. A two plate conductor system. Also shown is the discretization of the geometry into points.

computed such that the integration of $\ln(P,Q)$, $x\cdot\ln(P,Q)$ and $y\cdot\ln(P,Q)$ are exact. Thus, in this case, the numerical error comes only from the interpolation. We obtain the exact solution by using the Hermite-type interpolations. However, interpolation error arises near the two corners where q is discontinuous when the varying basis least-squares interpolation is used. Comparison of the numerical results and the exact solution is shown in Fig. 5(a). When the point distribution is refined (8, 16, 32, 64 points per edge and 2 points per cell), the numerical error is reduced. As shown in Fig. 5(b), the convergence rate is 0.5. This example clearly shows one advantage of the Hermite-type interpolation, i.e. the capability to capture the discontinuity in the normal derivative of the solution around the corner.

The second example is a Dirichlet problem on the square domain with a quadratic solution for u. Dirichlet boundary conditions are specified along the four edges of the boundary. The boundary conditions are

$$u = x^2 - y^2 \qquad \text{for } x = 0,\ y = 0,\ x = 1,\ y = 1 \tag{59}$$

The exact solution for q is given by

$$q = \frac{\partial u}{\partial n} = \begin{cases} 0 & \text{for } y = 0,\ x = 0 \\ 2 & \text{for } x = 1 \\ -2 & \text{for } y = 1 \end{cases} \tag{60}$$

The exact solution of u is quadratic which is outside the linear base interpolating polynomial. This problem is solved by the varying basis BCM with a uniform distribution of 8, 16, 32 and 64 points per edge and 2 points per cell to study the convergence behavior. The accurate integration distance, defined in Ref. [1], is set to be 0.5. A comparison of convergence of the BCM with varying basis and the BCM with Hermite-type interpolation is shown in Fig. 6. The normal derivative of the potential is not continuous at the points $(1,0)$, $(0,1)$ and $(1,1)$. This discontinuity is not captured by the varying basis BCM. The convergence rate of the varying basis BCM is found to be 0.43. In comparison, the Hermite interpolation functions give a better performance by including the normal derivative of the basis into the interpolation. The convergence rate of the BCM with Hermite-type interpolation is 1.41.

The third example is a Neumann problem on the square domain with a cubic exact solution for u. The specified boundary conditions are

$$\frac{\partial u}{\partial n} = \begin{cases} -3y^2 & \text{for } x = 0 \\ -3x^2 & \text{for } y = 0 \\ 3y^2 + 6y - 3 & \text{for } x = 1 \\ 3x^2 + 6x - 3 & \text{for } y = 1 \end{cases} \tag{61}$$

$$u = -x^3 - y^3 + 3x^2y + 3xy^2 \quad \text{for } x = 0.97,\ y = 0 \tag{62}$$

The exact solution for u is given by

$$u = -x^3 - y^3 + 3x^2y + 3xy^2 \qquad \text{for } x = 0,\ y = 0,\ x = 1,\ y = 1 \tag{63}$$

The convergence rate of the BCM with varying basis and the BCM with Hermite-type interpolations is shown in Fig. 7 employing a uniform distribution of 8, 16, 32 and 64 points per edge and 4 points per cell. The accurate integration distance is set to be 0.3. The convergence rate of the BCM with varying basis is found to be 2.09. The convergence rate is found to be 2.06 with the Hermite-type interpolation. Note that for this example the potential, u, and its normal derivative, q, are continuous along the boundary.

The fourth example is identical to the second example except that the nodes are distributed randomly on the boundary. An accurate integration distance of 0.3 is used for this example. The convergence plot with a random point distribution of 8, 16, 32 and 64 nodes per edge is shown in Fig. 8. The exact solution and the numerical solution obtained with a random distribution of 64 nodes per edge are compared in Fig. 9. The cloud size in this case is $r_x = r_y = 1.67\Delta s$, where Δs is the average spacing between the points. The convergence rate is found to be 1.35.

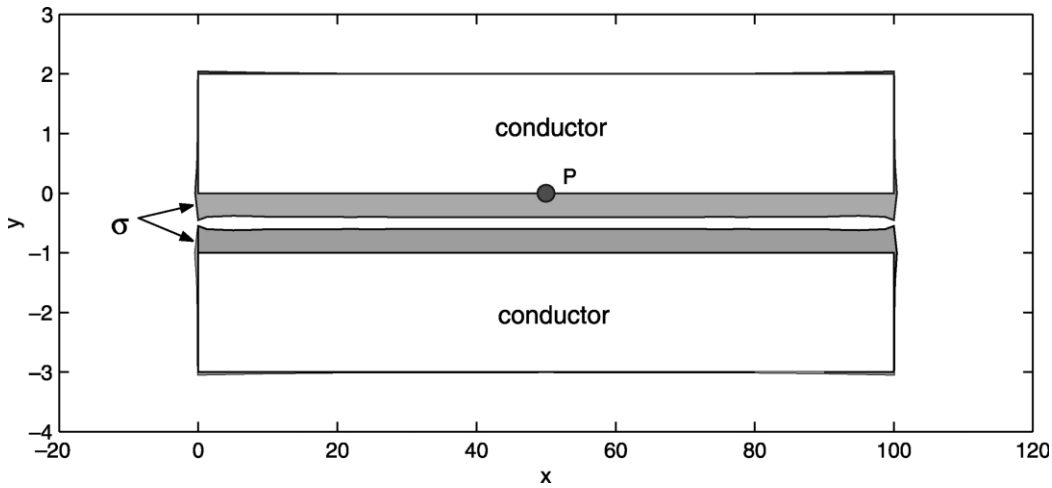

Fig. 13. Surface charge density distribution on the conductors obtained by the varying basis BCM.

The next example is a Dirichlet potential problem on the circular domain shown in Fig. 4(b). For the Dirichlet problem, the boundary conditions are

$$u = r\cdot\cos(\theta) = x \qquad \text{on all the boundary} \tag{64}$$

The exact solution for q is

$$\frac{\partial u}{\partial n} = \cos(\theta) \qquad \text{on all the boundary} \tag{65}$$

The accurate integration distance for the circular domain is set to be 0.8. The convergence of the BCM is shown in Fig. 10. The convergence rate is found to be 3.61 for the Hermite-type interpolation and 1.85 for varying basis interpolation. The varying basis interpolation shows a lower convergence rate compared to the regular Hermite-type interpolation.

The next example is a Neumann potential problem on the elliptical domain shown in Fig. 4(c). The boundary

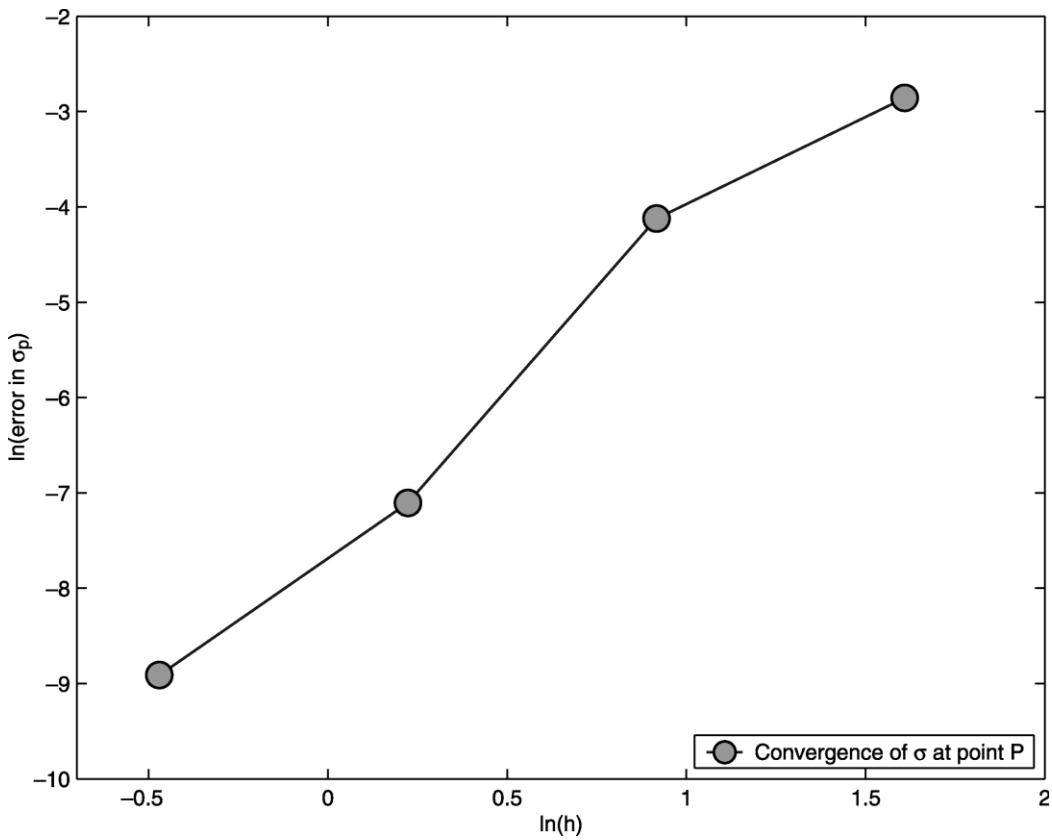

Fig. 14. Convergence of the surface charge density at point $p(\sigma_p)$.

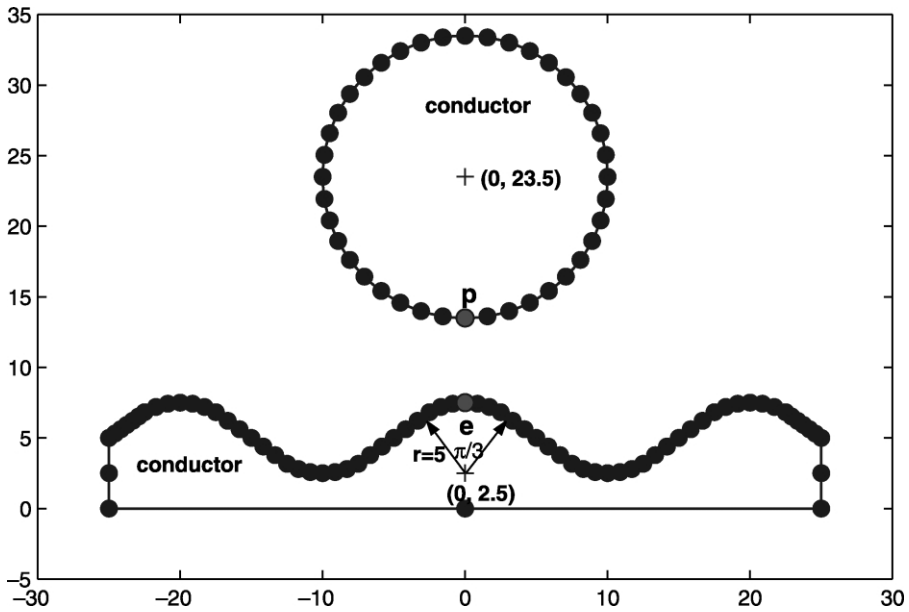

Fig. 15. A two conductor system with a complex geometry for the ground plane. Also shown is the point distribution.

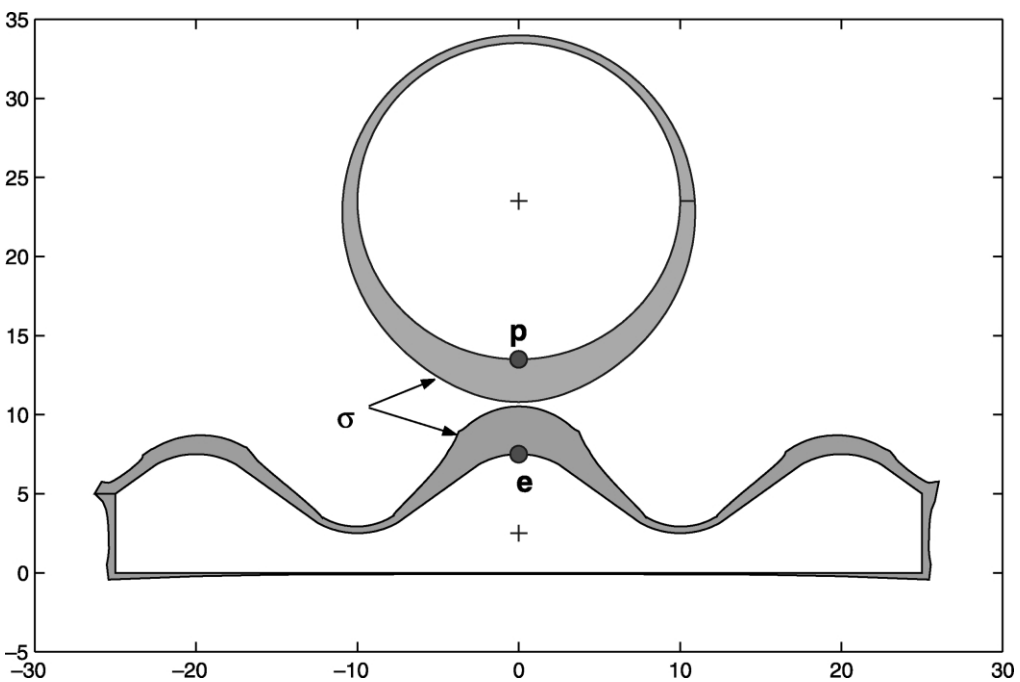

Fig. 16. Surface charge density distribution on the conductors obtained by the varying basis BCM.

conditions are

$$\frac{\partial u}{\partial n} = \cos(\hat{nx}) = \frac{b\cdot\cos(\theta)}{\sqrt{b^2\cos^2(\theta) + a^2\sin^2(\theta)}} \quad (66)$$

on all the boundary

$$u = x \quad \text{at two points on the boundary} \quad (67)$$

The exact solution for u is

$$u = x \quad \text{on all the boundary} \quad (68)$$

The accurate integration distance for the elliptical domain is set to be 0.8. The convergence of the BCM is found to be 2.61 (Fig. 11) for the Hermite-type BCM and 2.58 for the varying basis BCM.

5.2. 2D Exterior electrostatic analysis

The first example is a two plate conductor system as shown in Fig. 12. Each conductor has dimensions of 100×2 and the gap between the two plates is 1. The applied voltage is set to be 1 for the upper plate and 0 for the lower plate. The permittivity of the free space ϵ_0 is also set to be 1 for simplicity. Since the plate is very long compared to the gap the surface charge density σ at the center point p is given by

$$\sigma = \frac{\epsilon_0 V_{ab}}{d} \tag{69}$$

where V_{ab} is the potential difference between the two plates and d is the gap. The surface charge density distribution is shown in Fig. 13 and the convergence of the surface charge density at point p is shown in Fig. 14.

The second example shows that the BCM with the varying basis interpolation is attractive for solving complicated MEMS problems. Shown in Fig. 15 is a circular conductor located above a wave shaped ground plane. The surface charge density distribution obtained by the varying basis BCM is shown in Fig. 16. The computed surface charge densities at points p and e are 0.18 and 0.198, respectively.

6. Conclusion

A new implementation of the BCM is presented in this paper. The new implementation uses a varying polynomial basis when constructing the interpolation functions. Depending on whether the cloud is singular or non-singular, the base interpolating polynomial is defined to satisfy the linear completeness condition. The new implementation can handle points at corners and edges and is much simpler to implement and faster compared to our original implementation. Numerical results indicate that the Hermite-type BCM is more accurate and typically produces higher order convergence rate compared to the varying basis BCM.

Acknowledgements

This work was supported by the National Science Foundation under Grants CCR-9875671 and CCR-0121616.

References

[1] Li G, Aluru NR. Boundary cloud method: a combined scattered point/boundary integral approach for boundary-only analysis. Comput Meth Appl Mech Engng 2002;191(21/22):2337–70.
[2] Kane JH. Boundary element analysis in engineering continuum mechanics. Englewood Cliffs, NJ: Prentice-Hall; 1994.
[3] Greengard L, Rokhlin V. A fast algorithm for particle simulations. J Comput Phys 1987;73(2):325–48.
[4] Phillips JR, White JK. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. IEEE Trans Comput-Aid Des Integr Circuits Syst 1997;16(10):1059–72.
[5] Kapur S, Long DE. IES^3: a fast integral equation solver for efficient 3-dimensional extraction. IEEE Computer Aided Design, 1997, Digest of Technical Papers, IEE/ACM International Conference; 1997. p. 448–55.
[6] Shrivastava V, Aluru NR. A fast boundary cloud method for exterior 2-D electrostatics. Int J Numer Meth Engng 2003.
[7] Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. Comput Meth Appl Mech Engng 1996;139:3–47.
[8] Li S, Liu WK. Meshfree and particle methods and their applications. Appl Mech Rev 2002;55:1–34.
[9] Atluri SN. The meshless local Petrov-Galerkin (MLPG) method. Tech Science Press; 2002.
[10] Mukherjee YX, Mukherjee S. The boundary node method for potential problems. Int J Numer Meth Engng 1997;40:797–815.
[11] Chati MK, Mukherjee S. The boundary node method for three-dimensional problems in potential theory. Int J Numer Meth Engng 2000;47:1523–47.
[12] Chati MK, Mukherjee S. The boundary node method for three-dimensional linear elasticity. Int J Numer Meth Engng 1999;46:1163–84.
[13] Zhang J, Yao Z, Li H. A hybrid boundary node method. Int J Numer Meth Engng 2002;53(4):751–63.
[14] Chen W. Symmetric boundary knot method. Engng Anal Bound Elem 2002;26(6):489–94.
[15] Lancaster P, Salkauskas K. Surface generated by moving least squares methods. Math Comput 1981;37:141–58.
[16] Jackson JD. Classical electrodynamics, 3rd ed. New York: Wiley; 1999.
[17] Shi F, Ramesh P, Mukherjee S. On the application of 2D potential theory to electrostatic simulation. Commun Numer Meth Engng 1995;11:691–701.
[18] Onate E, Idelsohn S, Zienkiewicz OC, Taylor RL, Sacco C. A stabilized finite point method for analysis of fluid mechanics problems. Comput Meth Appl Mech Engng 1996;139:315–46.
[19] Aluru NR, Li G. Finite cloud method: a true meshless technique based on a fixed reproducing kernel approximation. Int J Numer Meth Engng 2001;50(10):2373–410.
[20] Jin X-Z, Li G, Aluru NR. On the equivalence between least-squares and kernel approximation in meshless methods. Comput Model Engng Sci 2001;2(4):447–62.
[21] Kapur S, Long DE. High-order Nyström schemes for efficient 3-D capacitance extraction. 38th International Conference on Computer-Aided Design, San Jose, CA, USA; 1998. p. 178–85.
[22] Golub GH, Van Loan CF. Matrix computations. Baltimore MD: Johns Hopkins University Press; 1989.