Approximating Three-Dimensional Potential Problems Using the Complex Variable Boundary Element Method (CVBEM): Applications

T. V. Hromadka II
Department of Mathematics
California State University
Fullerton, California 92634

Received 30 January 2000; accepted 12 May 2000

In the current research, the primary focus is to extend the CVBEM to solving potential problems in three dimensions (3D). This is achieved by applying the CVBEM to three coupled projections of the 3D problem domain, in 2D planes, and then superimposing the resulting corresponding 2D CVBEM solutions. The new 3D CVBEM technique is also applied towards improving 3D problem approximations, which are based on the usual 3D boundary element method (BEM) techniques, by approximating the 3D BEM residual error. Finally, a technique to extend a 3D problem geometry into higher geometric dimensions is introduced, and a corresponding numeric error reduction technique is advanced for use in superimposing multiple dimension approximations to improve 3D approximations. © 2000 John Wiley & Sons, Inc. Numer Methods Partial Differential Eq 16: 535–560, 2000

Keywords: CVBEM; complex variable boundary elements; multidimensional

I. INTRODUCTION

The complex variable boundary element method (CVBEM) is a numeric technique for use in solving, in an approximation sense, two-dimensional (2D) potential problems or 2D Poisson problems. The considerations of multiply connected regions (i.e., problem domains that contain holes) and various types of boundary conditions (e.g., flux type or Dirichlet type) have been examined in detail in various publications, including the recent book of Hromadka and Whitley [1]. Applications of the CVBEM to a wide variety of practical problems is also reported in the literature (see cited reference). Ever since the introduction of the CVBEM [2], the CVBEM has been limited to solving potential problems in 2D.

In the current research, the primary focus is to extend the CVBEM to solving potential problems in three dimensions (3D). This is achieved by applying the CVBEM to three coupled projections of

Correspondence to: Dr. T. V. Hromadka, II, Dept. of Mathematics, California State University, Fullerton, CA 92634 (e-mail: thromadka@exponent.com)

the 3D problem domain, in 2D planes, and then superimposing the resulting 2D CVBEM solutions. The new 3D CVBEM technique is also applied towards improving 3D problem approximations, which are based on the usual 3D boundary element method (BEM) techniques (e.g., [3]), by approximating the 3D BEM residual error. Finally, a technique to extend a 3D problem geometry into higher geometric dimensions is introduced, and a corresponding numeric error reduction technique is advanced, for use in superimposing multiple dimension approximations to improve 3D approximations.

II. MATHEMATICAL FORMULATION OF PROBLEM

The three-dimensional (3D) potential problem setting considered is to solve the PDE

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \text{ in } \Omega \tag{1}$$

with Dirichlet boundary conditions $\phi = \phi_b$ on Γ, where Ω is the 3D domain with boundary Γ; ϕ is the 3D potential function; and x, y, z are the usual cartesian coordinates. For development purposes, Ω is assumed to be simply connected (no holes interior of Ω).

III. GRAMM–SCHMIDT APPROXIMATION SETTING (HILBERT SPACE)

The approximation technique being considered is to define a particular set of basis functions, and then to determine the best approximation in a least squares error minimization sense (i.e., a Hilbert space setting; see [4]). In this case, the best approximation is determined by the usual Gramm–Schmidt process, with respect to the assembled set of basis functions in matching the value of the boundary conditions, ϕ_b, on Γ (again, see above-cited reference).

A. Gramm–Schmidt Integration Process

The Gramm–Schmidt integration process involves the use of integrals of the form, known as an inner-product, (f, g), given by

$$(f, g) = \int_\Gamma f g d\Gamma, \tag{2}$$

where f and g are integrable on Γ. In general, (2) is solved numerically, and the approach used herein is as follows:

Step 1. Define a set of "integration points," $\{p_i\}$, of uniform density, on the problem boundary Γ. Number the integration points from 1 to NI. (The uniform density of the $\{p_i\}$ on Γ can be relaxed, if the integral of (2) is extended to include a weighting function.)

Step 2. Develop a GEOMETRY vector, of dimension NI, composed of the coordinates of p_i, $i = 1, 2, \ldots, NI$.

Step 3. For any function used in (2), say f, develop a vector F, of dimension NI, composed of the values $F = \{f(p_i); i = 1, 2, \cdots NI\}$ (that is, develop a column or row vector).

Step 4. To approximate (2), use the vector dot product

$$(f, g) \cong (F, G) = \sum_{i=1}^{NI} f(p_i)g(p_i)\Delta\Gamma, \tag{3}$$

where $\Delta\Gamma$ is the measure of the incremental boundary associated to p_i. Note that as $NI \to \infty$ and $\Delta\Gamma \to 0$, that $(F, G) \to (f, g)$. Also, the requirement that the set of integration points be uniformly distributed on Γ can be readily relaxed by defining $\Delta\Gamma$, in (3), as a function of the p_i.

B. Gramm–Schmidt Orthonormalization Process

Given a set of m linearly independent basis functions (i.e., basis dimension, m), noted as $\{f_j; j = 1, 2, \ldots, m\}$, orthonormalization is achieved by using the Gramm–Schmidt process, except now we are in a vector representation of the set $\{f_j\}$, noted as $\{F_j\}$, and we use the vector dot product given in (3). The resulting orthonormalized vectors are $\{H_j; j = 1, 2, \ldots, m\}$.

C. Determining the Best Approximation to ϕ_b on Γ

To approximate ϕ_b, on Γ, another vector, ϕ_B, of dimension, NI, is developed as $\phi_B = \{\phi_b(p_i); i = 1, 2, \ldots, NI\}$. The best approximation of ϕ_b on Γ, noted as ϕ^*, is given by

$$\phi^* = \sum_{j=1}^{m} (H_j, \phi_B)H_j. \tag{4}$$

By back-substitution, ϕ^* can be rewritten in terms of the original vectors, F_j, giving

$$\phi^* = \sum_{j=1}^{m} \alpha_j F_j, \tag{5}$$

where the α_j are the coefficients determined from the usual Gramm–Schmidt back-substitution process from the (H_j, ϕ_B) values [4]. Note that as $NI \to \infty$ and $\Delta\Gamma \to 0$, the $\alpha_j \to \beta_j$, where β_j is the Gramm–Schmidt coefficient corresponding to the original basis function, f_j, and in the original space spanned by the $\{f_j\}$,

$$\phi^* = \sum_{j=1}^{m} \beta_j f_j. \tag{6}$$

D. Generalization to Other Spatial Dimensions

The above procedures are applicable to any spatial dimension. In our setting, the focus of the approximation is to match the boundary values, ϕ_b, on Γ, and, hence, we minimized the norm $\|\phi^* - \phi_b\|$ on Γ, where $\|\cdot\|$ is the usual least squares norm value. But Ω and Γ can be extended to any higher geometric dimension, the only difference in the above development being the expression of the GEOMETRY vector coordinates, p_i, in the appropriate geometric dimension.

IV. APPROXIMATING 3D POTENTIAL FUNCTIONS USING 3D BASIS FUNCTIONS

In the previous section, the focus of the Gramm–Schmidt procedure was to minimize $\|\phi^* - \phi_b\|$ on Γ. No attention was paid to how ϕ^* relates to the interior of Ω. In this section, the choice of

basis functions is addressed such that ϕ^* exactly solves the PDE operator equation, $L(\cdot) = 0$, where L is given by

$$L(\cdot) = \frac{\partial^2(\cdot)}{\partial x^2} + \frac{\partial^2(\cdot)}{\partial y^2} + \frac{\partial^2(\cdot)}{\partial z^2}, \tag{7}$$

and where $(\cdot)$ is a function that is twice differentiable with respect to x, y, and z as used in (7).

A. 3D Basis Functions of the Form, $1/R$

In 3D, the Green's function $1/R_j$ is a potential function, where $R_j^2 = (x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2$ and $x_j = (x_j, y_j, z_j)$ is an arbitrary point in space or "node," and $R_j > 0$. Thus, for $N3D$ nodes, defined exterior of $\Omega \cup \Gamma$, an approximation function of $\phi(x, y, z)$, as used in (1), is given by $\hat{\phi}_A$, where

$$\hat{\phi}_A = \sum_{j=1}^{N3D} a_j / R_j, \tag{8}$$

where the a_j are coefficients. The choice of the a_j is determined by minimizing the norm:

$$\|\hat{\phi}_A - \phi_b\| \text{ on } \Gamma, \tag{9}$$

arriving at the best approximation, noted as $\hat{\phi}_A^*$. Note that we now have $\hat{\phi}_A^*$ approximates ϕ_b on Γ and $L\hat{\phi}_A^* = 0$ in Ω.

V. APPROXIMATING 3D POTENTIAL FUNCTIONS USING LOWER DIMENSION BASIS FUNCTIONS

A. 1D Geometry Basis Functions

Other types of basis functions can be used other than the form used in (8), including functions that are developed in other geometry dimensions, but yet satisfy (1).

For example, in a one-dimension geometry, or 1D, extended for each of the 3 axes used in a 3D geometry, consider the set of functions $\{1, x, y, z\}$. These functions all satisfy (1) and can be used to increase the basis dimension, m, in developing a ϕ^*.

In a two-dimension geometry, or 2D, consider 2D polynomials generated from the real or imaginary parts of complex variable polynomials z^n, where $z = x + iy, i = \sqrt{-1}$. (There should be no confusion over the notation z being the complex variable vs. z being the third-dimensional coordinate.)

B. 2D Geometry Polynomial Basis Functions

For example, the complex function z^2 includes the 2D real functions, in the (x, y) 2D geometry plane of $(x^2 - y^2)$ and (xy). Note that analogous 2D functions can be developed in the (x, z) and (y, z) 2D planes as well, which for the equivalent second-degree complex polynomial results in the functions $\{(x^2 - y^2), xy, (x^2 - z^2), xz, (y^2 - z^2), yz\}$. All these functions are generated in a 2D geometry sense, yet can be extended into the 3D geometry, and satisfy (1). (It is noted that the six functions listed above are not all linearly independent and do not all survive the orthonormalization process.) By increasing the degree of the complex polynomial in each of the three 2D planes, an arbitrarily large basis can be developed for use in (6). (It is noted that the 1D geometry extended basis functions, stated in the above, are elements of the 2D geometry extended

polynomial basis functions. It is also noted that these particular basis functions do not involve the use of nodal points.)

C. 2D Geometry CVBEM Basis Functions

Many other types of 2D geometry basis functions are also available. In this section, the complex variable boundary element method (CVBEM) is utilized to generate 2D geometry basis functions to be extended into the 3D geometry.

Unlike the 2D polynomial basis functions, but similar to the $1/R_j$ 3D form of basis functions, the CVBEM requires the use of nodal points. Details regarding the CVBEM can be found in numerous articles and books, including [1]; consequently, these details are not repeated here. For our purposes, it is sufficient to state the form of the CVBEM basis functions (for the case of a linear global trial function used in the CVBEM). Section VIII.C contains further information regarding the CVBEM.

The CVBEM involves the sum of products of complex coefficients multiplied by analytic functions of the form,

$$\hat{\omega}(z) = \sum_{j=1}^{N} C_j(z - z_j)Ln_j(z - z_j), \tag{10}$$

where $\hat{\omega}(z)$ is the CVBEM approximation in the (x, y) plane; N is the number of CVBEM basis functions; C_j are complex constants; z_j is $x_j + iy_j$, the (x, y) plane coordinate of node j; and Ln_j is the complex natural logarithm with branch cut oriented to lie exterior of Ω and not intersecting with other such branch cuts emanating from other CVBEM nodes. (Again note that z is notation for the third coordinate in a 3D geometry, and also denotes the complex variable.) From Euler's formula,

$$z - z_j = r_j e^{i\theta_j}, \tag{11}$$

where r_j is the 2D radial distance from z to z_j, and θ_j is the radial angle measured counterclockwise from the branch cut defined at CVBEM node j. From (10) and (11), the associated 2D CVBEM basis functions are of the form

$$a_j r_j(\cos\theta_j \ln r_j - \theta_j \sin\theta_j)$$

or

$$b_j r_j(\sin\theta_j \ln r_j + \theta_j \cos\theta_j), \tag{12}$$

where the a_j and b_j are real constants.

By simply rewriting the complex variable in terms of the other spatial coordinates using $x + iz$ or $y + iz$, the CVBEM basis functions, (12), are readily extended into the other 2D (x, z) and (y, z) planes, resulting in another set of extended 2D basis functions for use in the 3D problem setting of (7) and (9), to be applied analogous to the extended 2D complex polynomial formulation described previously.

The CVBEM formulation involves the use of three sets of 2D plane nodal points, one set of nodes being defined for each of the three 2D planes. The approach used is described as follows:

Step 1. Define the 3D problem domain to have geometry coordinates all greater than zero (i.e., if necessary, simply translate the 3D $\Omega \cup \Gamma$).

Step 2. Project the 3D $\Omega \cup \Gamma$ onto the (x, y) 2D plane. The 2D projected domain is denoted by Ω_{xy} with boundary Γ_{xy}. Similarly, project $\Omega \cup \Gamma$ onto the 2D (x, z) and (y, z) planes, resulting in Ω_{xz} and Γ_{xz}, and Ω_{yz} and Γ_{yz}, respectively.

Step 3. Define CVBEM nodes in the (x, y) 2D plane, all located exterior of Ω_{xy} (but arbitrarily close or on Γ_{xy}). Similarly, define other CVBEM node sets corresponding to Γ_{xz} and Γ_{yz}, respectively.

Step 4. Develop CVBEM basis functions, of the form of (12), with respect to each node, corresponding to the Γ_{xy}, Γ_{xz}, and Γ_{yz} surfaces, in the $(x, y), (x, z)$, and (y, z) planes, respectively.

Upon completion of the above steps, a set of extended 2D CVBEM basis functions result that satisfy (1) and have potential function components in all three geometry dimensions of the 3D $\Omega \cup \Gamma$. That is, analogous to the above complex polynomial extension into 3D, the CVBEM is now extended into 3D. The applications section provides further details on the implementation of this procedure.

VI. USING EXTENDED 2D CVBEM BASIS FUNCTIONS IN APPROXIMATING 3D POTENTIAL FUNCTIONS

Two techniques for using the extended 2D CVBEM basis functions are as follows.

A. Technique 1: 3D PDE Solver

The extended 2D CVBEM basis functions can be written in a more generalized form as

$$\phi_j^{\alpha\beta} = [r_j(\cos\theta_j \ln r_j - \theta_j \sin\theta_j)]^{\alpha\beta}$$

$$\psi_j^{\alpha\beta} = [r_j(\sin\theta_j \ln r_j + \theta_j \cos\theta_j)]^{\alpha\beta}, \tag{13}$$

where superscripts $\alpha\beta$ refer to any of the $(x, y), (x, z)$, or (y, z) 2D planar coordinates. That is, ϕ_i^{xy} and ψ_i^{xy} refer to CVBEM basis functions defined with respect to the j^{th} CVBEM nodal point that is located exterior of the 2D projection, Ω_{xy} of the 3D problem domain, Ω, onto the (x, y) plane. Thus, for n_1, n_2, and n_3 CVBEM nodes defined appropriately (i.e., exterior or on the 2D projected surfaces Γ_{xy}, Γ_{xz}, and Γ_{yz}, respectively) in each of the $(x, y), (x, z)$, and (y, z) 2D planes, respectively, the 3D CVBEM approximation function is given by

$$\begin{aligned}\hat{\phi}(x, y, z) = &\sum_{j=1}^{n_1} (a_j^{xy}\phi_j^{xy} + b_j^{xy}\psi_j^{xy}) \\ &+ \sum_{j=1}^{n_2} (a_j^{xz}\phi_j^{xz} + b_j^{xz}\psi_j^{xz}) + \sum_{j=1}^{n_3} (a_j^{yz}\phi_j^{yz} + b_j^{yz}\psi_j^{yz}),\end{aligned} \tag{14}$$

where the $a_j^{\alpha\beta}$ and $b_j^{\alpha\beta}$ are constants to be determined by the Gramm–Schmidt process described previously. It is noted that for simplicity, a 3D approximator can be readily developed in terms of only using the $\phi_j^{\alpha\beta}$ or the $\psi_j^{\alpha\beta}$ functions, rather than both forms of basis functions. It is also noted that, from (14),

$$L(\hat{\phi}(x, y, z)) = 0 \text{ in } \Omega \tag{15}$$

and $\|\hat{\phi}(x, y, z) - \phi_b\|$ is minimized on Γ by the Gramm–Schmidt procedure. An application of Technique 1 is provided in Section VIII.

B. Technique 2: 3D Numerical Error Reduction

Technique 2 involves the concept of coupling both the more standard 3D approximation techniques, such as use of (8), with the extended 2D approximation techniques introduced in the above, such as the use of (14). The following steps are used:

Step 1. Solve (1) using the 3D solver, such as (8), and develop the best approximation for this step, denoted as ϕ_A^*. Hereafter this is called the e approximation "baseline" solution, because a dimension k solver is being used to approximate an order k PDE.

Step 2. Using ϕ_A^*, develop the boundary condition error of

$$e_A = \phi_b - \phi_A^* \text{ on } \Gamma. \tag{16}$$

Step 3. Using the extended CVBEM technique, in 3D, solve the PDE of (1), but instead of using ϕ_b as the problem boundary conditions defined on Γ, use e_A, of (16), as the boundary conditions, and develop the best approximation for this step, denoted as ϕ_B^*.

Step 4. Couple the results of the above steps by adding ϕ_A^* and ϕ_B^*,

$$\phi^* = \phi_A^* + \phi_B^*. \tag{17}$$

Then $L(\phi^*) = 0$ in Ω, and $\|\phi^* - \phi_b\|$ is further reduced in value.

An application of Technique 2 is provided in Section VIII.

It is noted that the above steps could be applied on an individual (x, y), (y, z), and (x, z) 2D planar basis, one at a time, rather than in combination, in order to further increase the 3D solver basis function dimension without increasing computational effort in solving matrix systems.

VII. APPROXIMATING 3D POTENTIAL FUNCTIONS USING HIGHER DIMENSION BASIS FUNCTIONS

A. Approximating 2D Potential Functions Using 3D Basis Functions

Consider a 2D Potential problem,

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \text{ in } \Omega \tag{18}$$

with $\phi = \phi_b$ on the boundary of Ω, denoted by Γ, where now $\Omega \cup \Gamma$ is a 2D geometry. Assume that Ω is simply connected and Γ is a simple closed contour. A method to approximately solve (18) is to extend the 2D problem domain and boundary into 3D, redefine boundary conditions on the extended 3D domain, and solve a 3D PDE system analogous to the 2D PDE, equivalent to (1), given by

$$\left.\begin{aligned} \tfrac{\partial^2 \phi}{\partial x^2} + \tfrac{\partial^2 \phi}{\partial y^2} + \tfrac{\partial^2 \phi}{\partial z^2} = 0 \\ \tfrac{\partial \phi}{\partial z} = 0 \end{aligned}\right\} \text{ in } \Omega', \tag{19}$$

where Ω' is the 3D domain developed from extending the 2D domain. To arrive at Ω', and its associated boundary (in the 3D geometry), Γ', the following steps are undertaken:

Step 1. Define integration points (see Section III.A) on the 2D boundary, Γ, and also define integration points in the interior of the 2D domain, Ω. Note that $\Omega \cup \Gamma$ lies in the (x, y) plane.

Step 2. The new dimension to be attached, namely the z coordinate, is now coupled to each integration point, defined in above Step 1, in the positive orthogonal direction. Let $z = 0$ at each integration point defined in the original $\Omega \cup \Gamma$ domain, resulting in the 3D coordinate $(x_j, y_i, 0)$ for each such integration point.

Step 3. Extend rays, from each integration point of Step 1, into the third dimension, in the positive direction, from $z = 0$ to $z = L$, where L is the desired length of 3D "tube" being extended from the original 2D domain. Note that rays extend into the positive third dimension from integration points located on the original boundary and in the interior of the 2D $\Omega \cup \Gamma$. Also note that, algorithmically, all that is required to attach the new dimension is to simply add another coordinate dimension.

Step 4. Define new integration points, perhaps at an equal increment spacing, say Δz, along each ray, from Step 3, until $z = L$. At each spacing location, a new set of integration points now exist that are geometrically identical in pattern to the original set of integration points, now corresponding to $z = 0$.

We now have a 3D "tube" in (x, y, z) such that each 2D slice in (x, y), where each slice is taken orthogonal to the 3rd dimension axis, results in identical 2D shapes and integration point spacing. Note that the 2D interior, Ω, becomes a boundary surface of the 3D extension, Γ'. Also note that at $z = L$, the boundary surface appears identical in geometry to the boundary surface at $z = 0$; that is, the 2D boundaries of the 3D tube ends are identical to any 2D slice of the tube. We say that the set of integration points that lie on a particular ray are part of a "string." Note that the number of strings equals the sum of the number of integration points defined on the original 2D Γ and the number of integration points defined in the interior of the 2D Ω.

We now have arrived at a 3D geometric extension of the 2D problem domain. We can approximately solve the 3D PDE system of (19) with boundary conditions defined as follows:

Step 1. For each string attached to the original Γ, define $\phi_b(x_j, y_j, z_j) = \phi_b(x_j, y_j, 0)$. That is, the value of ϕ_b is a known constant on each string attached to Γ.

Step 2. For each string originating from the interior of Ω, satisfy $\partial\phi/\partial z = 0$ by requiring ϕ to be constant along each string. Note that for such "interior" strings, the constant needed is solved as part of the numerical solution (which, of course, is the goal of the original 2D Potential problem). This provides for a "zero flux" along the newly attached orthogonal geometry dimension.

Given the 3D domain and boundary as extended from the 2D domain and boundary, and given the extended PDE system and corresponding boundary conditions (of which the interior strings can be used or, as an alternative, simply set $\phi(x_j, y_j, 0) = \phi(x_j, y_j, L)$ for integration points interior of Ω), we can now solve the new 3D problem, using Technique 2 of Section VI.B, that is equivalent to the original 2D problem. Then, we arrive at the best approximation, in 3D, noted again as ϕ_A^*. Solving for ϕ_B^*, as in Section VI.B the best approximation is $\phi^* = \phi_A^* + \phi_B^*$.

B. Extending a Dimension kPotential Problem Setting Into Dimension $(k+1)$

An objective of Section VII.A is to introduce the underpinnings of how to apply the geometry extension concepts, introduced in Section V, towards using still higher geometric dimensions. The above methods provide overall generality to other geometric dimensions of basis functions. To present such a generalization, the extension of a 3D potential problem into a 4D problem

setting is now examined. The extension into even higher dimensions directly follows from the procedural steps presented. To better describe the procedures, the 2D PDE problem of Section VII.A is carried forward in its developed 3D problem setting, to be extended into a 4D problem setting.

Three key steps are involved in extending a k-dimension potential problem into a $(k+1)$ dimension setting (where k is a positive integer):

Step 1. Define a PDE system, in dimension $(k+1)$, equivalent to the PDE system of dimension k.

Step 2. Develop a dimension $(k+1)$ geometry "tube," denoted as $\Omega^{k+1} \cup \Gamma^{k+1}$, from the dimension k problem domain, denoted as $\Omega^k \cup \Gamma^k$.

Step 3. Define dimension $(k+1)$ boundary conditions on Γ^{k+1}, equivalent to the original dimension k boundary conditions, and corresponding to the dimension $(k+1)$ PDE system.

With respect to Step 1, the Section VII.A 2D problem PDE is

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0, \text{ in } \Omega^2. \tag{20}$$

When extended into 3D, the PDE of (20) becomes the PDE system (see (19)),

$$\left.\begin{aligned} \tfrac{\partial^2 \phi}{\partial x^2} + \tfrac{\partial^2 \phi}{\partial y^2} + \tfrac{\partial^2 \phi}{\partial z^2} &= 0 \\ \tfrac{\partial \phi}{\partial z} &= 0 \end{aligned}\right\} \text{ in } \Omega^3. \tag{21}$$

And when extended into 4D, the PDE system of (21) becomes

$$\left.\begin{aligned} \tfrac{\partial^2 \phi}{\partial x^2} + \tfrac{\partial^2 \phi}{\partial y^2} + \tfrac{\partial^2 \phi}{\partial z^2} + \tfrac{\partial^2 \phi}{\partial s^2} &= 0 \\ \tfrac{\partial \phi}{\partial z} &= 0 \\ \tfrac{\partial \phi}{\partial s} &= 0 \end{aligned}\right\} \text{ in } \Omega^4, \tag{22}$$

where s is notation for the added fourth dimension geometry coordinate. From (20)–(22), adding dimensions simply adds terms to the PDE operator $L(\cdot)$, with additional zero gradient conditions imposed on the new dimensions.

With respect to Step 2, the 3D geometry "tube," as developed by extending the original 2D problem domain and boundary into 3D, can now be extended into a 4D "tube" analogous to the previous geometry extension procedure. Again, a dimension 4 coordinate is attached to all 3D coordinates, resulting in $(x_j, y_j, z_j, 0)$ coordinates for each integration point j used on the boundary and in the interior for the 3D tube. Note that for the 4th dimension, $s = 0$ at this stage of the analysis. Next, rays are extended into the positive 4th dimension, from $s = 0$ to $s = L_s$, emanating from each 3D integration point on Γ^3 and in Ω^3. The integration points that are located in the interior of Ω^3 (i.e., the integration points defined along the strings connecting both ends of the 3D tube, and do not lie on the boundary of the 3D tube, Γ^3) are also extended into 4D, to become part of Γ^4, when $s = 0$ or $s = L_s$, using the same procedure. We now have the 4D problem boundary, Γ^4.

By defining new integration points on each 4D string, that emanate from Γ^3, at increments Δs, Γ^4 is prepared to perform numeric integration operations. If new integration points are also defined along the strings located in the interior of Ω^4 (i.e., strings emanating from the integration points located in the interior of Ω^3), then the interior of Ω^4 is prepared to be extended in dimension 5.

Note that the 4D geometry "tube" boundary has end boundaries defined by, simply, the set of all integration points with 4D coordinates $(x_j, y_j, z_j, 0)$ at one end and the set of all points with coordinates (x_j, y_j, z_j, L_s) at the other end. The 4D strings have coordinates (x_j, y_j, z_j, s), where $0 \leq s \leq L_s$. Integration points are defined in the interior of Ω^4, along strings, using the same incremental spacing, Δs, used along Γ^4 strings. Note that any 3D "slice" of the 4D tube, taken orthogonal to the 4th dimensional axis, s, is identical in geometric shape and distribution of integration point locations to the 4D tube end boundaries (i.e., the original 3D shape and distribution of integration points). By comparison of this exercise to that accomplished in Section VII.A, the geometric extension procedure from dimension k to dimension $(k+1)$ follows accordingly.

It is seen that the above geometry dimension extension procedure is simply the "stacking" of dimension k "slices," along the "vertical" $(k+1)$ dimension direction, where each slice is a copy of the original $\Omega^k \cup \Gamma^k$, offset by a spatial increment in the $(k+1)$ dimension. Additionally, all integration points of each slice are in "vertical" alignment with respect to dimension $(k+1)$, and lie along a straight line in dimension $(k+1)$, called a dimension $(k+1)$ string.

With respect to Step 3, all strings used in extending $\Omega^k \cup \Gamma^k$ into $\Omega^{k+1} \cup \Gamma^{k+1}$ have the property that ϕ is a constant value (i.e., ϕ is an equipotential) along an individual string. Consequently, any integration point located on Γ^{k+1} either has a ϕ_b value defined from Γ^k, or has a zero flux boundary condition due to being located at the end boundary of the dimension $(k+1)$ tube.

C. Dimension $(k+1)$ Potential Basis Functions

Given the $\Omega^{k+1} \cup \Gamma^{k+1}$ with boundary conditions, ϕ_b^{k+1}, and PDE system in dimension $(k+1)$, a dimension $(k+1)$ approximation function of the form of (8) is used. In dimension $(k+1)$, R_j^{k+1} is the dimension $(k+1)$ Euclidean norm (analogous to 2D and 3D), measured from node j located in the $(k+1)$ dimension space, but exterior of $\Omega^{k+1} \cup \Gamma^{k+1}$.

Define $L^{k+1}(\cdot)$ by

$$L^{k+1}(\cdot) = \sum_{i=1}^{k+1} \frac{\partial^2(\cdot)}{\partial u_i^2}, \tag{23}$$

where u_i is the i^{th} dimension coordinate in the $(k+1)$ dimension geometry. Then

$$L^{k+1}(R_j^{k+1})^{(k-1)} = 0, \tag{24}$$

and a suitable approximation function is

$$\hat{\phi}^{k+1} = \sum_{j=1}^{N_{k+1}} a_j / (R_j)^{(k-1)}, \tag{25}$$

where the goal is to minimize, as before $\|\hat{\phi}^{k+1} - \phi_b^{k+1}\|$.

D. Multiple Dimension Numerical Error Reduction

Analogous to Section VI.B, the highest dimension solver used forms the approximation baseline solution. The application of successively smaller dimension approximations, each targeted to evaluating the residual error remaining in matching boundary conditions, and then adding all dimension solutions together, forms the total approximation function.

An application of solving a 2D potential problem by coupling 2D, 3D, and 4D approximations is contained in Section VIII.

VIII. APPLICATIONS

A. Introduction

In this section, we investigate the effectiveness of approximating a potential function $f(x, y, z)$ by a generalized Fourier series whose basis functions include potentials of the form

$$\zeta_i(x, y, z) = \frac{1}{\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}}, \quad i = 1, \ldots, n, \tag{26}$$

where $\{P_i(x_i, y_i, z_i)\}_{i=1}^{N}$ is a predetermined set of points in R^3 called 3D nodes.

Initially, f is approximated by the ζ_i's discretely in a least-squares sense on the surface of a cube, a process that emulates solving a normal R^3 boundary-value problem. This is referred to as the 3D Solver.

The error from the above approximation, in turn, is least-squares approximated using 2D CVBEM potential basis functions generated in the xy, xz, and yz planes.

An advantage of this process is that the basis of the generalized Fourier series is composed of potential functions of three distinct types; it should prove useful in the approximation of a variety of different functions.

B. 3D Solver Approximation of a Function

Integration Points. A number of equally spaced points (called *integration* points) are selected on each face of a cube (called the *primary* cube). These points are arranged in an $n \times n$ pattern on each face (Fig. 1), resulting in a total of $6n^2$ integration points. At these integration points the function f is evaluated to produce the boundary conditions of the problem. These function values are stored in a $6n^2 \times 1$ column vector called F.

3D Nodes. The ζ_i 3D potential basis functions are uniquely defined by the choice of the 3D nodes. The 3D nodes are defined on a larger cube that is centered around the primary cube. These nodes are arranged in an equally spaced $m \times m$ pattern on each face. (Note: must have $m \leq n$ or the resulting system is underdetermined). The ζ_i's generated by such nodes are, of course, linearly independent.

The size of the 3D cube is related to that of the primary cube by the value of *offset* (Fig. 2).

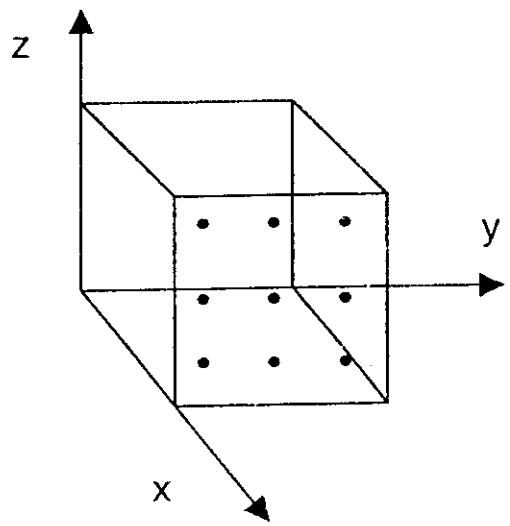


FIG. 1. Placement of integration points.

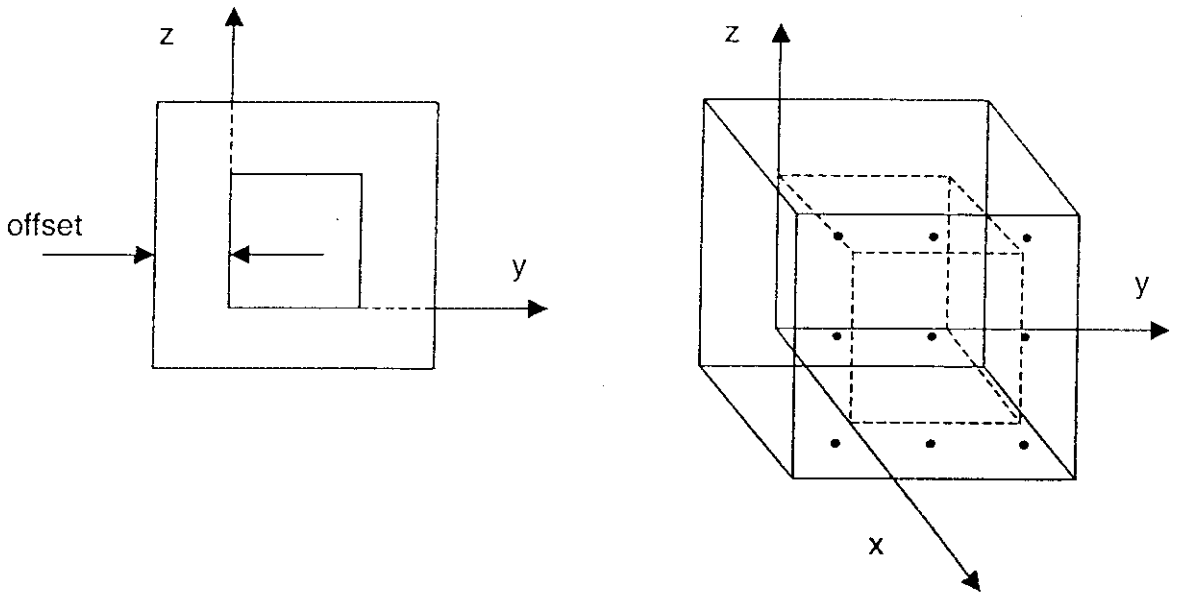


FIG. 2. Basis function node placement.

Each ζ_i basis function is evaluated at the integration points (in the same order as was done for the function f) and the results are stored in the columns of a $6n^2 \times 6m^2$ matrix D (for $3D$). To discretely approximate f with a linear combination of the ζ_i's, solve the system

$$Dx = F. \tag{27}$$

In most cases, this system is overdetermined $(m < n)$. The least-squares solution is found by orthonormalizing D using the Gramm–Schmidt method. Let M represent the orthonormalized D. Then the normal equations are

$$M^T(Mv - F) = 0$$

or

$$M^T Mv = M^T F. \tag{28}$$

Since $M^T M = I$, then the solution in terms of the orthonormal basis (columns) is

$$v = M^T F. \tag{29}$$

The coefficient vector x from the original problem can now be found by solving the upper-triangular system

$$M^T Dx = v. \tag{30}$$

Hence the vector $\psi = Dx$ is the $6n^2 \times 1$ column vector that most closely approximates the vector F in the subspace spanned by the basis function vector representations.

Note that this process of using Gramm–Schmidt and then back-substituting to find x is an efficient way to solve this least-squares system.

Error Evaluation. The error evaluation in the continuous case over each 2D face A is

$$e_1 = \left[\int_A (f - f_{\text{APPROX}})^2 dA\right]^{\frac{1}{2}}. \tag{31}$$

To approximate this in the discrete case over the entire cube, the following is used:

$$e_1 \approx \left[\sum_{j=1}^{6n^2} [F_j - \psi_j]^2 \frac{s^2}{n^2}\right]^{\frac{1}{2}}$$

$$= \frac{s}{n}\left[\sum_{j=1}^{6n^2} [F_j - \psi_j]^2\right]^{\frac{1}{2}}$$

$$= \frac{s}{n}\,\|E_1\|_2, \tag{32}$$

where F_j and ψ_j are the j^{th} elements of F and ψ, respectively; $E_1 = F - \psi$ (a $6n^2 \times 1$ column vector); and s is the length of one side of the cube.

C. CVBEM Approximation of 3D Solver Error

The objectives here are to provide a discussion of the general theory of the complex variable boundary element method (CVBEM), to explain how to discretely solve real-valued boundary value problems in R^2 using CVBEM, and to explain how this method can be used to estimate the error resulting from the 3D Solver in (R^3).

Consider a function $\varphi(z) = \alpha(z) + i\beta(z)$ in the complex plane defined on a simply connected region Ω bounded by a simple closed curve Γ, with φ analytic on $\Omega \cup \Gamma$. By Cauchy's Integral,

$$\varphi(z) = \frac{1}{2\pi i}\int_\Gamma \frac{\varphi(\zeta)d\zeta}{\zeta - z}. \tag{33}$$

More simply, Let Γ be a square boundary, and consider a set of n nodes on Γ, (Fig. 3).

Let $G(z)$ be a global trial function defined and continuous on Γ, and composed of a linear combination of independent nodal basis functions [4]. An approximation to φ is

$$\hat{\varphi}(z) = \frac{1}{2\pi i}\int_\Gamma \frac{G(\zeta)d\zeta}{\zeta - z}. \tag{34}$$

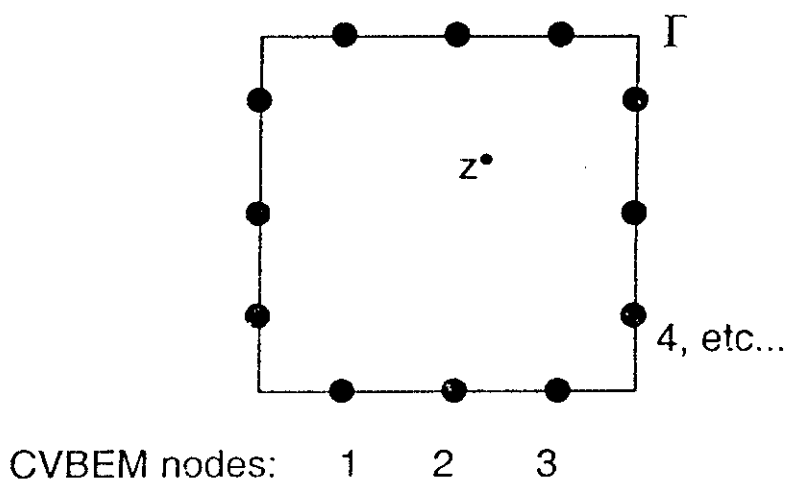


FIG. 3. CVBEM node placement.

This integral can be approximated (discretely) by

$$\hat{\varphi}(z) = \sum_{j=1}^{n} c_j (z - z_j) \ln(z - z_j), \tag{35}$$

where z_j is the j^{th} node coordinates. Note that the summand argument can be rewritten as

$$\begin{aligned} c_j(z - z_j)\ln(z - z_j) &= (a_j + ib_j)(z - z_j)\ln(z - z_j) \\ &= (a_j + ib_j)(\mathrm{Re}^{i\theta})\ln(\mathrm{Re}^{i\theta}) \quad (\text{where } R = R_j \text{ and } \theta = \theta_j) \\ &= (a_j + ib_j)[[R\ln(R)\cos\theta - R\theta\sin\theta] + i[R\ln(R)\sin\theta - R\theta\cos\theta]]. \end{aligned} \tag{36}$$

Extracting the real part, we obtain an approximation of $\alpha(z)$

$$\hat{\alpha}(z) = \sum_{j=1}^{n} [a_j[R\ln(R)\cos\theta - R\theta\sin\theta] - b_j[R\ln(R)\sin\theta - R\theta\cos\theta]]. \tag{37}$$

In this presentation, since the two functions of R and θ above are closely related, only functions of the form

$$g_j(R, \theta) = [R\ln(R)\cos\theta - R\theta\sin\theta] \tag{38}$$

are considered as basis functions for use in the approximation of the real-valued function $\alpha(z)$. It should also be noted here that these basis functions are themselves potentials.

To implement the CVBEM method, the n nodes and their associated *branch cuts* must be chosen. A branch cut is a ray, which emanates from a node and does not intersect the 2D boundary except at the node itself. There is only one branch cut assigned to each node, from which all angles θ are measured for that node. Branch cuts are necessary to preserve continuity of the g_j's.

To simplify the process, all CVBEM nodes are chosen at equally spaced intervals with the pattern of node points identical on each side of Γ. The branch cuts are selected to be at outward normal angles to the problem boundary (Fig. 4).

Integration points must also be chosen on Γ. Again, to simplify matters, integration points are chosen in the same manner as the nodes. No integration point, however, should coincide with a node (since $\ln(0)$ is not defined). In addition, if q represents the number of integration points chosen, it is desirable to ensure that $q \geq n$.

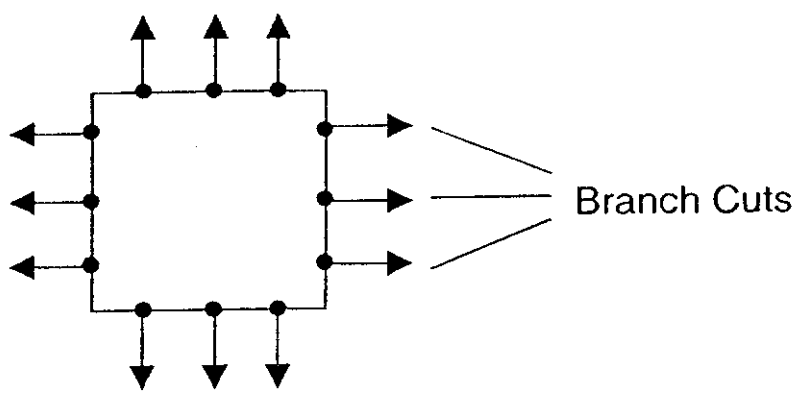


FIG. 4. Branch cuts for CVBEM nodes.

Let C (for CVBEM) be the $q \times n$ matrix whose columns are the g_j's evaluated at the integration points. Figure 5 illustrates the quantities involved in the evaluation of a g_j at an integration point.

The function f to be approximated can either be given explicitly or defined discretely on the integration points. If given explicitly, the function must be evaluated at all the integration points, resulting in a $q \times 1$ column vector E.

If the function is given discretely on the integration points (as it would be in a real-world problem), E is already known. For our purposes here, $E = E_1$, the error column vector $F - \psi$ from the 3D Solver.

In either case, the system

$$C_y = E_1 \tag{39}$$

can be solved as before to obtain a least-squares solution so that

$$e_2 = \|E_1 - C_y\|_2 \tag{40}$$

is a minimum.

CVBEM IN R^3. To handle the error from the 3D Solver, the xy, xz, and yz planes are all used separately as CVBEM 2D platforms. Because the CVBEM basis functions are limited to the plane, the following convention is used.

Since all integration points in R^3 are essentially "stacked" in columns on the surface of the cube, they can be substituted in the CVBEM by their projection onto the appropriate plane (Fig. 6).

To simplify this process, the same CVBEM node pattern is used in all three planes (xy, xz, and yz).

D. Linear Approximation of CVBEM Error

For completeness as a final error reducing measure, the error vector

$$E_2 = E_1 - Cy \tag{41}$$

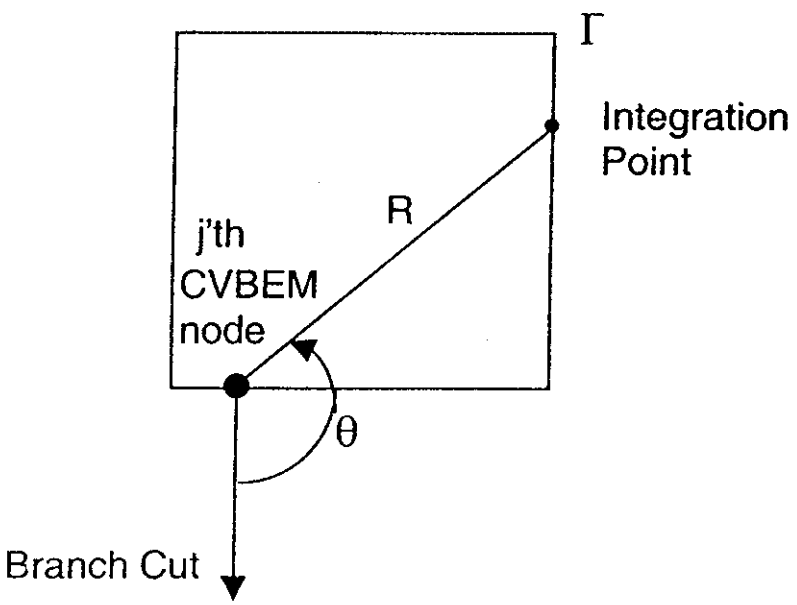


FIG. 5. Branch cut parameters.

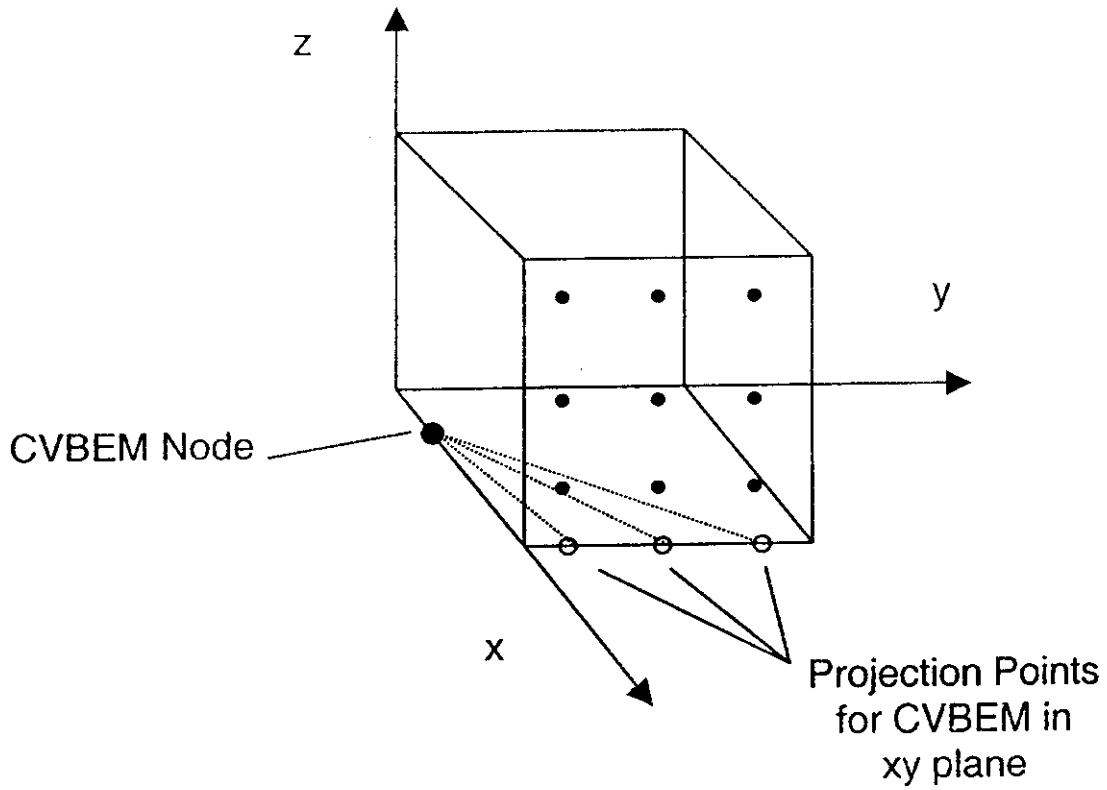


FIG. 6. CVBEM node geometry parameters.

from the CVBEM method is least-squares approximated by the linear potential basis $\{1, x, y, z\}$. Let E_3 represent the error vector from this last approximation. Then E_3 is the error vector for the complete process (3D Solver, CVBEM, and Linear method) and the 2-norm error for the entire process is

$$e_3 = \|E_3\|_2. \tag{42}$$

It is noted that use of the 10 basis functions provides little help in the overall process due to the minimization in a Hilbert Space setting.

E. MATLAB Program Features

The entire system presented has been programmed in MATLAB. Some of the features include (varies among the three programs presented):
Input:

1. User-defined array sizes for integration points, 3D nodes, and CVBEM nodes.
2. User-defined primary cube size.
3. User-defined 3D Solver cube size.

Output:

1. Graphical plots of functions, solvers, and errors.
2. Screen display of pertinent norms and cooperature vectors (optional).
3. Contour potential function, e.g., plots of plane slices through the primary cube (slices parallel to xy, xz, and yz axis available only).

F. Two Potential Function Approximations

Two separate potential problems were tackled with the 3D Solver/CVBEM/Linear method programs. Because of computing limitations, the number of integration points that were used in analyzing these functions was limited to 100 per side (600 total).

Problem 1. Using a primary cube in R^3 with side length 5 units, three singularities were placed at the three points $P_1(5, 0, -2)$. $P_2(0, -.1, 5)$ and $P_3(0, 5, 6)$, by virtue of invoking the potential function:

$$f_1(x, y, z) = \frac{-500}{\sqrt{(x-5)^2 + (y-0)^2 + (z+2)^2}} + \frac{10000}{\sqrt{(x-0)^2 + (y+0.1)^2 + (z-5)^2}} + \frac{100}{\sqrt{(x-0)^2 + (y-5)^2 + (z-6)^2}}.$$

The first term of f_1 emulates a cold temperature sink; the second, a very hot source; and the third, a mild source.

Example 1. The function f_1 can be seen to be of similar type to the ζ_i's from the 3D Solver. To use the 3D Solver to its full power, 100 3D nodes per side (600 total) were used in this first example. 36 CVBEM nodes were used per plane (108 total).

The results of the program run appear in Fig. 7. As can be seen by the graph and the norms, the 3D Solver does a nice job of approximating f_1. Note that the CVBEM does reduce the error slightly. The Linear method does not have much effect.

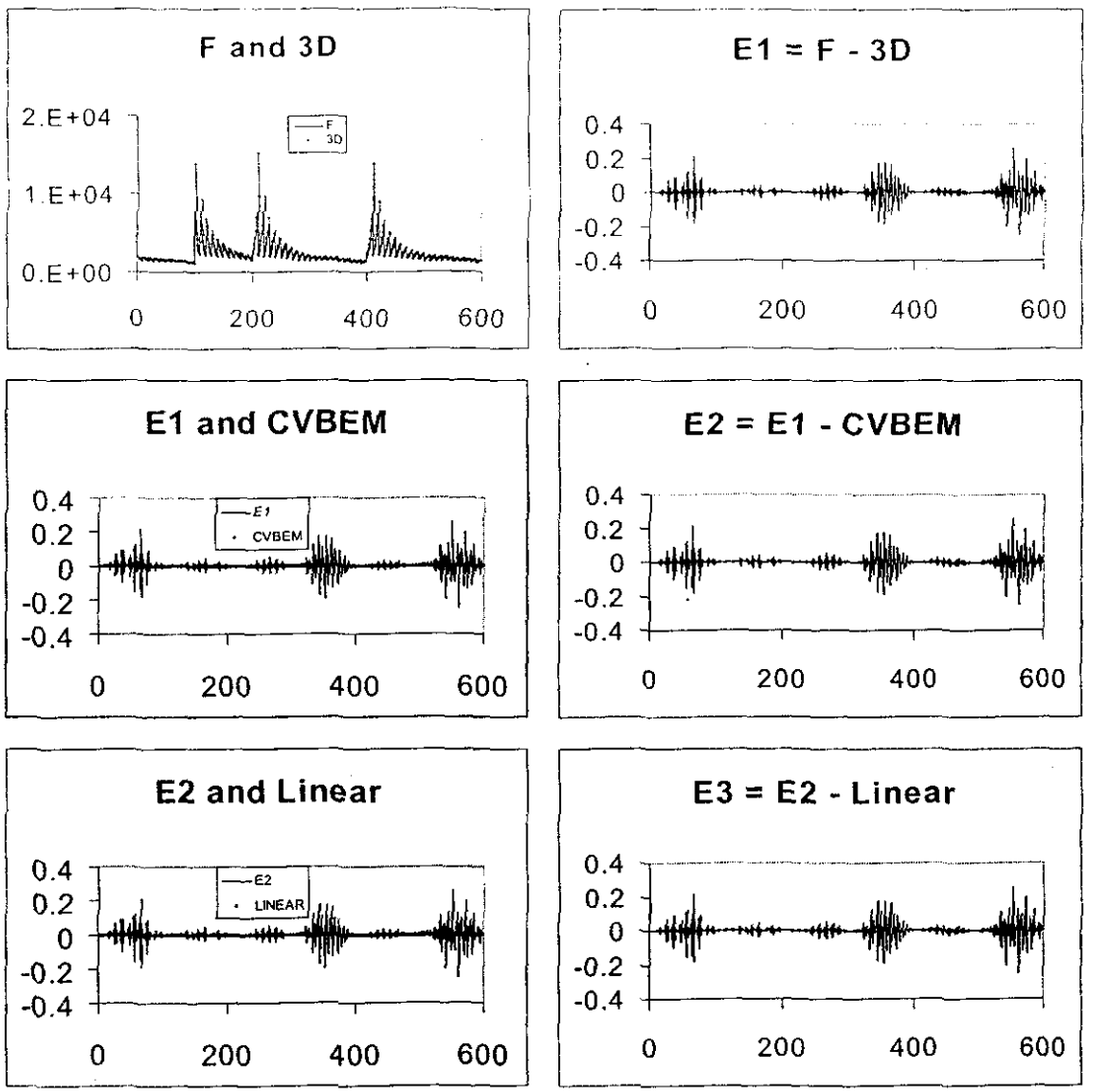
Figure 8 provides graphs of a slice through the primary cube parallel to the xy plane at a height of 4.9 units above it. Slice graphs are given for 3D Solver only, 3D Solver plus CVBEM, 3D Solver plus CVBEM plus Linear method, and finally for the actual function f_1. All slice graphs in this example look identical to the graph of f_1. The actual errors involved are supplied by the norms in the bottom half of the figure.

Example 2. To ascertain the effect of the CVBEM in approximating the error from the 3D Solver, the number of 3D nodes was reduced to one per side (6 total). In a number of program experiments, the error on the *primary cube* from the 3D Solver has been reduced by up to 60% by varying the number of CVBEM nodes used. Table I shows that, for the conditions given, the CVBEM errors on a slice *decrease* as the number of CVBEM nodes increase.

From Table I, the experiment using 36 CVBEM nodes per plane (and 6 3D nodes) was quite successful. Figure 9 illustrates that the 3D Solver left a large error on the primary cube. The same

TABLE I.

Number of CVBEM nodes (per plane)	CVBEM E_2 error on cube ($\times 10^3$)	CVBEM E_2 error on slice ($\times 10^3$)
4	2.92	6.75
8	2.64	6.28
16	2.37	5.89
24	2.27	5.66
32	2.25	5.60
36	2.24	5.54



**********LEAST-SQUARES PROGRAM**********

FUNCTION: 1/R1 + 1/R2 + 1/R3
Side of Cube: 5
Offset for 3D Basis: 2
Number of Integration Points: 600
Number of 3D Basis Functions: 600
Number of CVBEM Basis Functions: 108

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 3.84975111e-001
REL: ||F-Dx||/||F|| = 2.76946833e-005

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| = 3.84962730e-001
REL: ||E1-Cy||/||E1|| = 9.99967838e-001

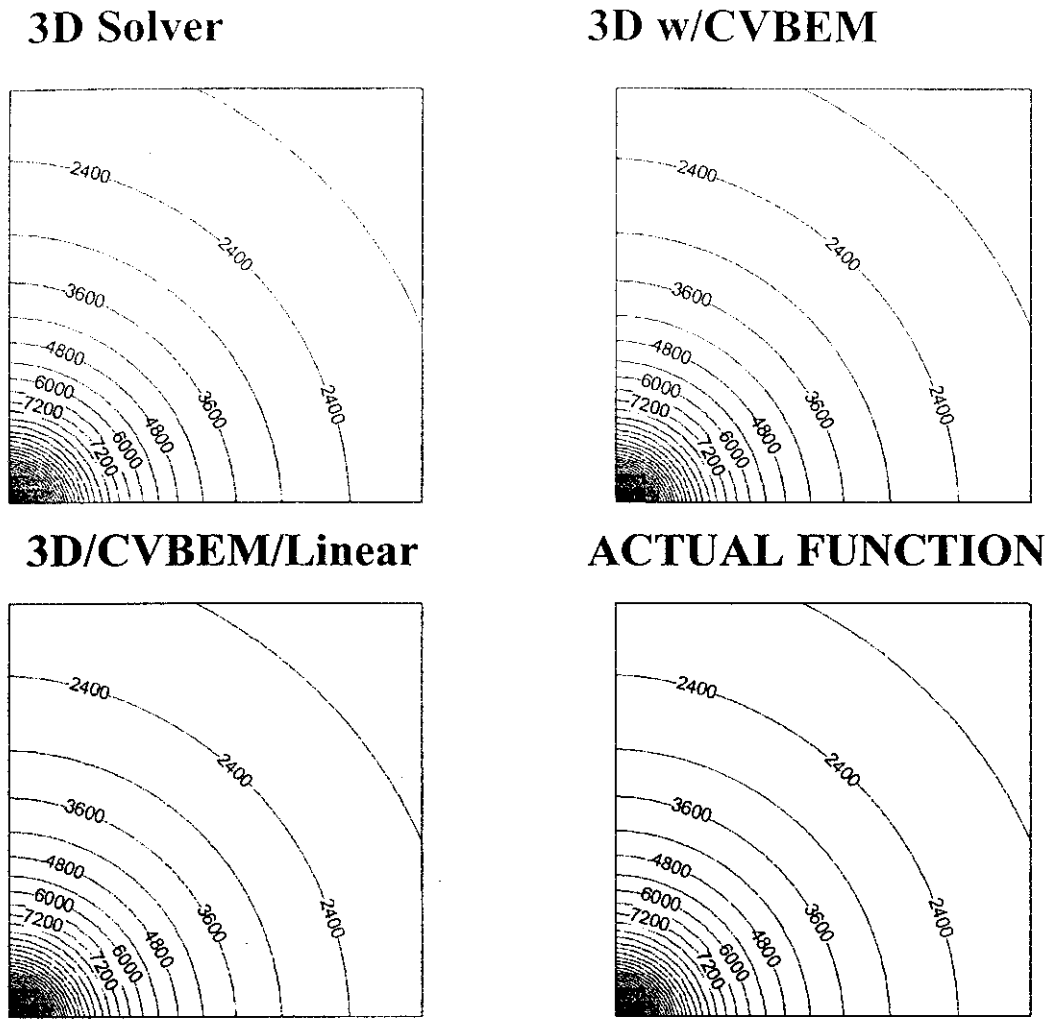
COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 3.84962730e-001
REL: ||F-Dx-Cy||/||F|| = 2.76937926e-005

LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = 3.84962730e-001
REL: ||E2-Lz||/||E2|| = 1.00000000e+000

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| = 3.84962730e-001
REL: ||F-Dx-Cy-Lz||/||F|| = 2.76937926e-005

FIG. 7.

figure shows that the CVBEM decreased this 3D error by 51%. Figure 10 confirms that the 3D slice error was cut down 43% by the CVBEM process. Note that the 3D slice is in error. Most of the high bands of temperature are missing or misplaced. On the other hand, the CVBEM slice has improved the approximation. These are impressive results considering that f_1 is quite different from the CVBEM basis functions. And again, as before, the Linear method has little effect.



Enter # for plane parallel to slice (xy = 1, xz = 2, yz = 3): 1
Enter constant variable value (slice constant): 4.9
Enter plot saturation value n(nxn plot saturation): 30

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 8.77519282e+002
REL: ||F-Dx||/||F|| = 4.38163460e-002

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| = 8.77519121+e002
REL: ||E1-Cy||/||E1|| = 9.99999817e-001

COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 8.77519121e+002
REL: ||F-Dx-Cy||/||F|| = 4.38163380e-002

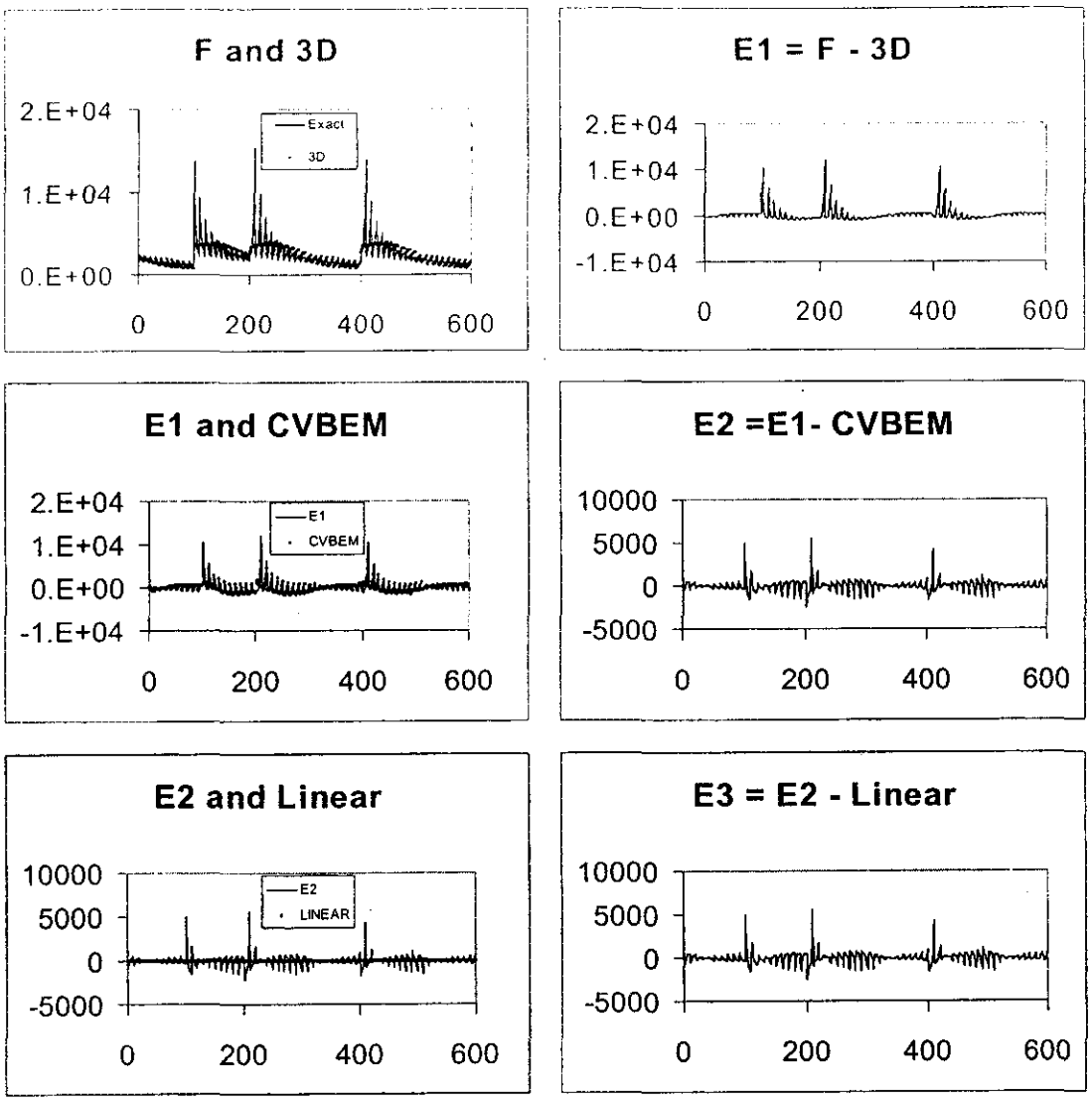
LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = 8.77519121e+002
REL: ||E2-Lz||/||E2|| = 1.00000000e+000

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| = 8.77519121e+002
REL: ||F-Dx-Cy-Lz||/||F|| =4.38163380e-002

FIG. 8.

Problem 2. The second function approximated by the program was the potential function

$$\begin{aligned} f_2(x,y,z) &= \mathrm{Re}[(x+iy)^4 + (x+iz)^4 + (y+iz)^4] \\ &= 2[x^4 + y^4 + z^4 - 3x^2y^2 - 3x^2z^2 - 3y^2z^2]. \end{aligned}$$



**********LEAST-SQUARES PROGRAM**********
FUNCTION: 1/R1 + 1/R2 + 1/R3
Side of Cube: 5
Offset for 3D Basis: 2
Number of Integration Points: 600
Number of 3D Basis Functions: 600
Number of CVBEM Basis Functions: 108

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 6.04741746e+003
REL: ||F-Dx||/||F|| = 4.35044517e-001

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| = 2.97870260e+003
REL: ||E1-Cy||/||E1|| = 4.92557793e-001

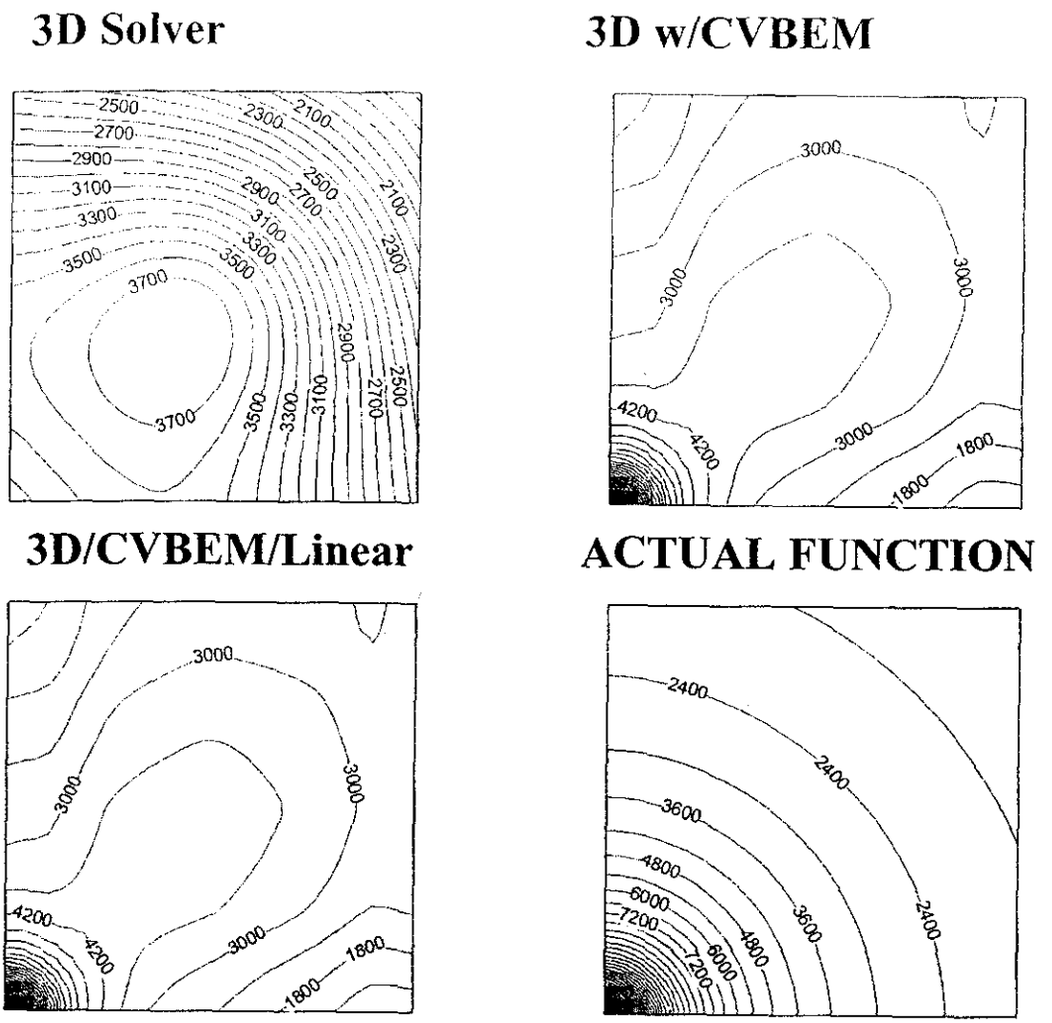
COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 2.97870260e+003
REL: ||F-Dx-Cy||/||F|| = 2.14284567e-001

LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = 2.97870259e+003
REL: ||E2-Lz||/||E2|| =9.999999999e-001

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| =2.978702590e+003
REL: ||F-Dx-Cy-Lz||/||F|| = 2.14284567e-001

FIG. 9.

Example 3. In this example, the 3D solver was put to full use. The results on the primary cube are given in Fig. 11. The 3D Solver approximated f_2 extremely well (note the absolute error of 2.8×10^{-2}). The CVBEM reduced this error a bit more (again 36 CVBEM nodes per plane or 108 total were used). The Linear method did nothing.



Enter # for plane parallel to slice (xy = 1, xz = 2, yz = 3): 1
Enter constant variable value (slice constant): 4.9
Enter plot saturation value n(nxn plot saturation): 30

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 1.10717497e+004
REL: ||F-Dx||/||F|| = 5.52835277e-001

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| =6.37106924e+003
REL: ||E1-Cy||/||E1|| = 5.75434726e-001

COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 6.37106924e+003
REL: ||F-Dx-Cy||/||F|| = 3.18120616e-001

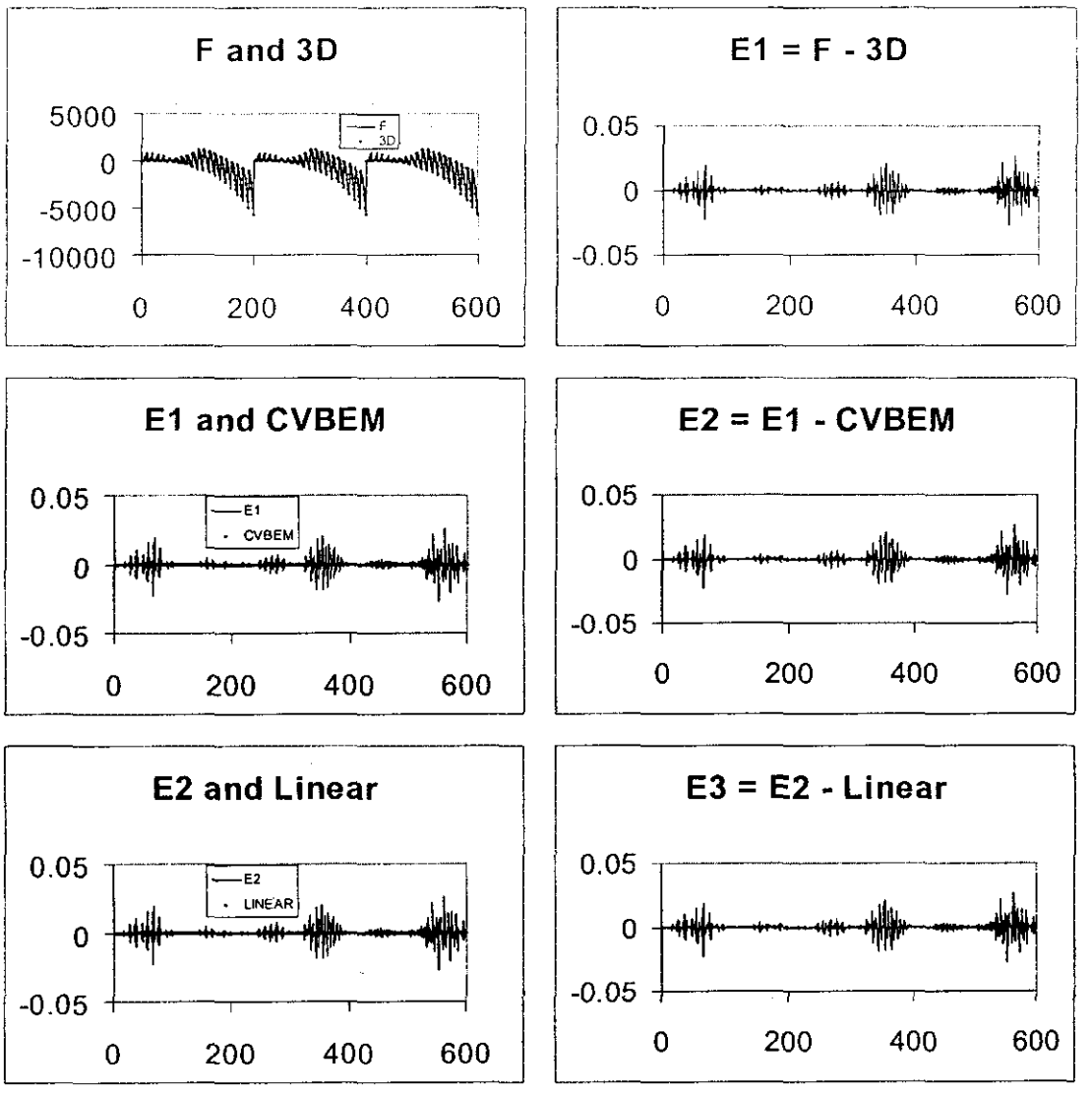
LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = 6.37107149e+003
REL: ||E2-Lz||/||E2|| = 1.00000035e+000

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| = 6.37107149e+003
REL: ||F-Dx-Cy-Lz||/||F|| =3.18120728e-001

FIG. 10.

In Fig. 12, all slices appear to be identical. The norms show that the CVBEM slightly decreased the error on the slice.

Example 4. Here the 3D nodes were decreased again to a total of only 6 to test the effectiveness of the CVBEM process. The results are even more impressive than those from Example 2. Figure 13 shows that the error on the primary cube left by the 3D Solver was quite large. The CVBEM



**********LEAST-SQUARES PROGRAM**********
FUNCTION: Re(Z^4)
Side of Cube: 5
Offset for 3D Basis: 2
Number of Integration Points: 600
Number of 3D Basis Functions: 600
Number of CVBEM Basis Functions: 108

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 2.75475111e-002
REL: ||F-Dx||/||F|| =1.262946833e-003

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| =2.753962730e-001
REL: ||E1-Cy||/||E1|| = 9.99962565e-001

COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 2.753962730e-001
REL: ||F-Dx-Cy||/||F|| = 1.262846833e-003

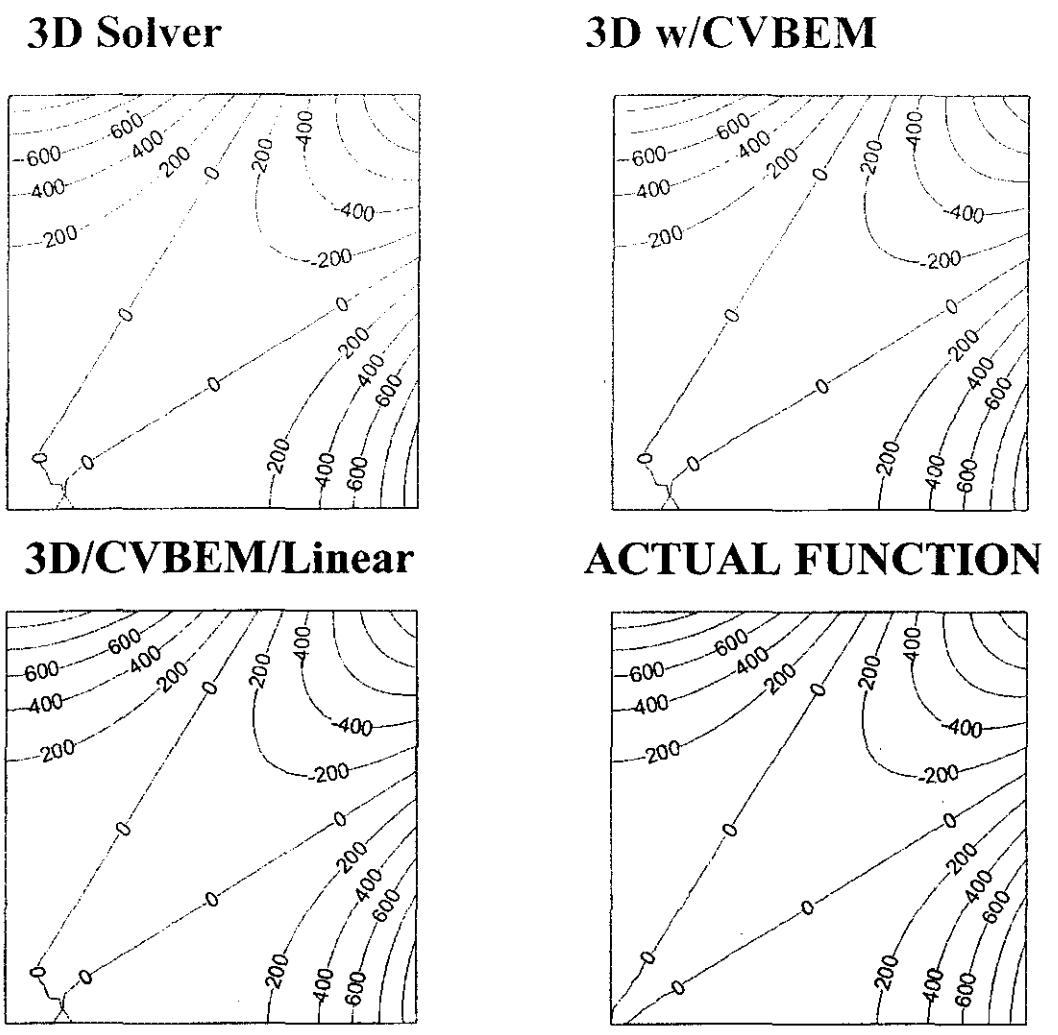
LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = = 2.753962730e-001
REL: ||E2-Lz||/||E2|| = 1.00000000e+000

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| = = 2.753962730e-001
REL: ||F-Dx-Cy-Lz||/||F|| = 1.262846833e-003

FIG. 11.

applied to this error decreased it by 96%. And, as in all of the cases so far, the Linear method has little error reduction effect.

The slice graphs provided in Fig. 14 show similar results. The 3D Solver graph looks nothing like the slice of the actual function f_2. The CVBEM plot, however, appears nearly identical to that of f_2. As evidenced by the norms, the CVBEM cut down the 3D Solver error on the slice by 91%.



Enter # for plane parallel to slice (xy = 1, xz = 2, yz = 3): 2
Enter constant variable value (slice constant): .1
Enter plot saturation value n(nxn plot saturation): 30

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 5.90898304e-002
REL: ||F-Dx||/||F|| = 3.36752460e-005

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| =5.90897314e-002
REL: ||E1-Cy||/||E1|| = 9.99998325e-001

COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 5.90897314e-002
REL: ||F-Dx-Cy||/||F|| = 3.36751896e-005

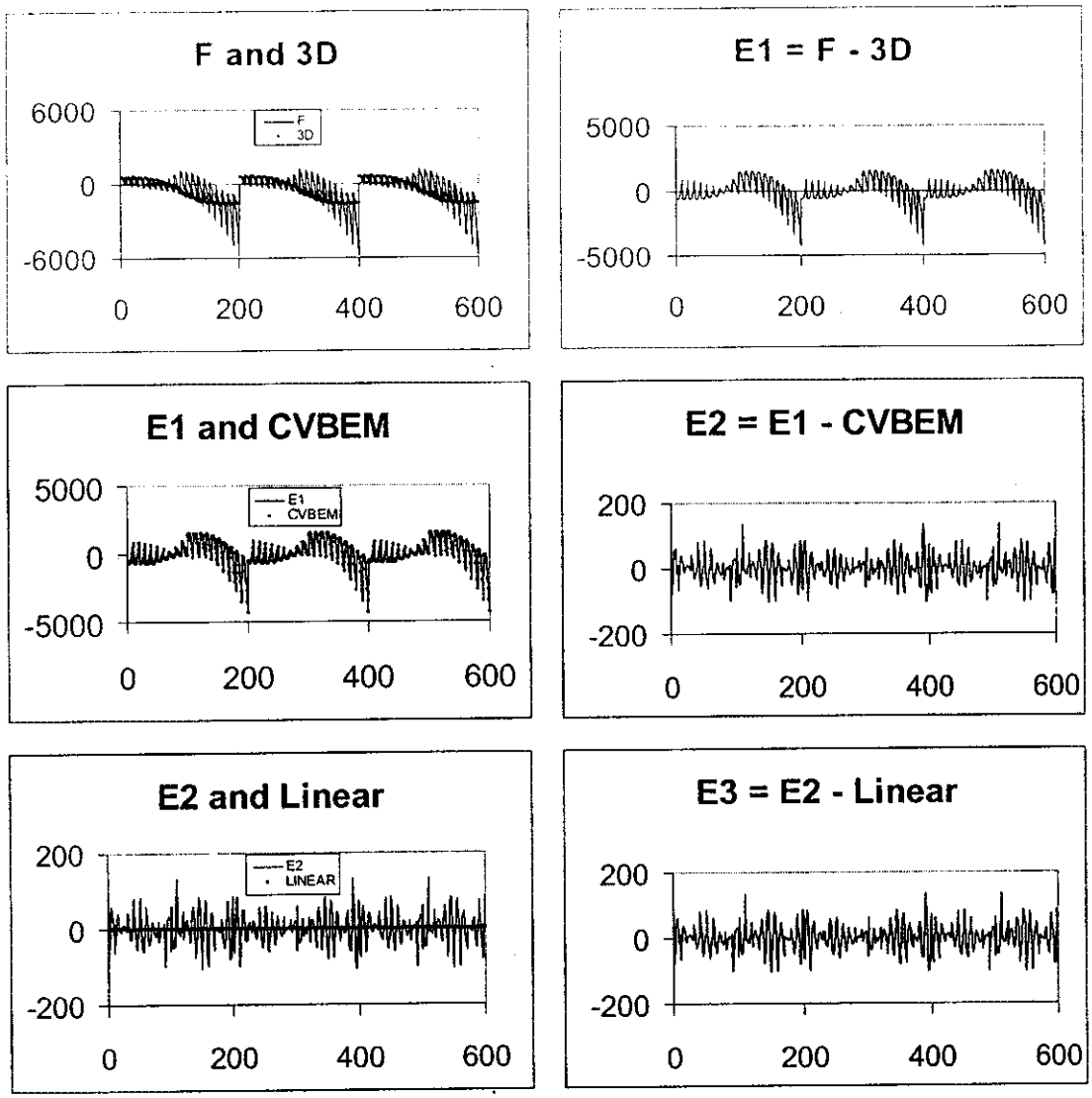
LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = 5.90897314e-002
REL: ||E2-Lz||/||E2|| = 1.00000000e+000

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| = 5.90897314e-002
REL: ||F-Dx-Cy-Lz||/||F|| =3.36751896e-005

FIG. 12.

IX. CONCLUSIONS

In this article, the primary focus is to extend the CVBEM to solving potential problems in three dimensions (3D). This is achieved by applying the CVBEM to three coupled projections of the 3D problem domain, in 2D planes, and then superimposing the resulting corresponding



**********LEAST-SQUARES PROGRAM**********
FUNCTION: Re(Z^4)
Side of Cube: 5
Offset for 3D Basis: 2
Number of Integration Points: 600
Number of 3D Basis Functions: 600
Number of CVBEM Basis Functions: 108

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 4.72877840e+003
REL: ||F-Dx||/||F|| = 7.05025115e-001

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| = 2.09605544e+002
REL: ||E1-Cy||/||E1|| = 4.43255162e-002

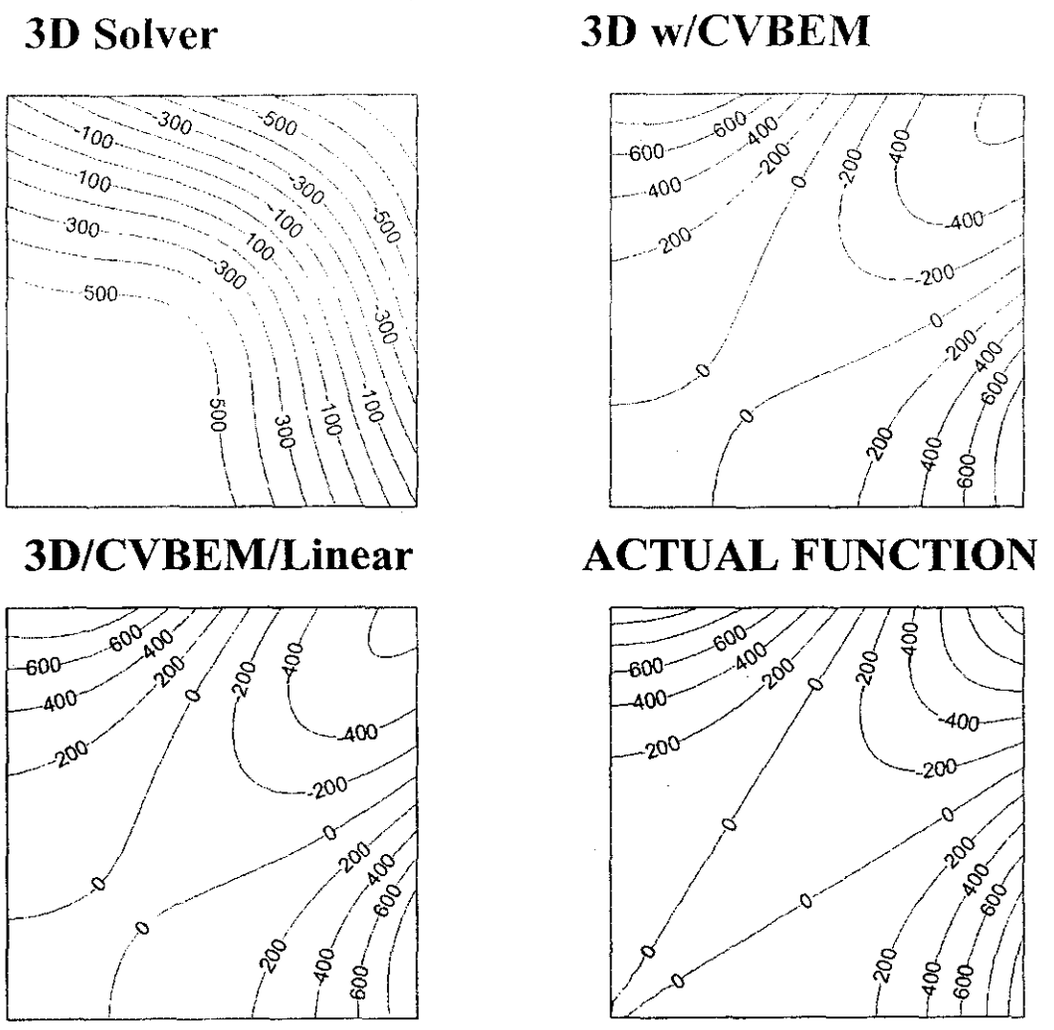
COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 2.09605544e+002
REL: ||F-Dx-Cy||/||F|| = 3.12506022e-002

LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = = 2.09605541e+002
REL: ||E2-Lz||/||E2|| = 9.99999988e-001

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| = = 2.09605541e+002
REL: ||F-Dx-Cy-Lz||/||F|| = 3.12506018e-002

FIG. 13.

2D CVBEM solutions. The new 3D CVBEM technique is also applied towards improving 3D problem approximations, which are based on the usual 3D boundary element method (BEM) techniques, by approximating the 3D BEM residual error. Finally, a technique to extend a 3D problem geometry into higher geometric dimensions is introduced, and a corresponding numeric



Enter # for plane parallel to slice (xy = 1, xz = 2, yz = 3): 2
Enter constant variable value (slice constant): .1
Enter plot saturation value n(nxn plot saturation): 30

3D SOLVER ERROR ANALYSIS:
ABS: ||F-Dx|| = 2.41419819e+003
REL: ||F-Dx||/||F|| = 1.37584958e+000

CVBEM ERROR ANALYSIS:
ABS: ||E1-Cy|| =2.12903631e+002
REL: ||E1-Cy||/||E1|| = 8.81881327e-002

COMBINED 3D/CVBEM:
ABS: ||F-Dx-Cy|| = 2.12903631e+002
REL: ||F-Dx-Cy||/||F|| = 1.21333605e-001

LINEAR BASIS ANALYSIS:
ABS: ||E2-Lz|| = 2.12901476e+002
REL: ||E2-Lz||/||E2|| = 9.99989881e-001

COMBINED 3D/CVBEM/LINEAR:
ABS: ||F-Dx-Cy-Lz|| = 2.12901476e+002
REL: ||F-Dx-Cy-Lz||/||F|| =1.21332377e-001

FIG. 14.

error reduction technique is advanced, for use in superimposing multiple dimension approximations to improve 3D approximations. The application problems demonstrate the advantage and utility in using multidimensional approximations in solving various dimensional potential problems.