UC Irvine

UC Irvine Previously Published Works

Title

A computational approach to determining CVBEM approximate boundaries

Permalink

https://escholarship.org/uc/item/35q6n321

Authors

Johnson, Anthony N
Hromadka, TV
Carroll, M
et al.

Publication Date

2014-04-01

DOI

10.1016/j.enganabound.2013.12.011

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Engineering Analysis with Boundary Elements 41 (2014) 83–89

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

A computational approach to determining CVBEM approximate boundaries

Anthony N. Johnson [a,*], T.V. Hromadka II [a], M. Carroll [a], M. Hughes [a], L. Jones [a], N. Pappas [a], C. Thomasy [a], S. Horton [a], R. Whitley [b], M. Johnson [c]

[a] United States Military Academy, 601 Swift Road, West Point, NY 10996, United States
[b] University of California, Irvine, Irvine, CA 92697, United States
[c] University at Albany, State University of New York, 1400 Washington Ave, Albany, NY 12222, United States

ARTICLE INFO

Article history:
Received 12 September 2013
Accepted 31 December 2013
Available online 4 February 2014

Keywords:
Cauchy integral
Complex variables
Collocation
Dirichlet
Approximate boundary
Complex variable boundary element method (CVBEM)

ABSTRACT

The complex variable boundary element method (CVBEM) provides solutions of partial differential equations of the Laplace and Poisson type. Because the CVBEM is based upon convex combinations from a basis set of functions that are analytic throughout the problem domain, boundary, and exterior of the problem domain union boundary (except along branch cuts), both the real and imaginary parts of the CVBEM approximations satisfy the Laplace equation, leaving the modeling error reduction effort to be that of fitting the problem boundary conditions. In this paper, the approximate boundary approach is used to depict the goodness of fit between the CVBEM results and the problem boundary conditions. The approximate boundary is the locus of points where the CVBEM approximation function meets the values of the problem boundary conditions. Because of the collocation method, the approximate boundary necessarily intersects the problem boundary at least at the collocation points specified on the problem boundary. Consequently, adding nodes and collocation points on the problem boundary results in reducing the departure between the approximate boundary and the true problem boundary. Thus, the approximate boundary is developed by tracking level curves from the real and/or imaginary parts of the CVBEM approximation function.

Published by Elsevier Ltd.

1. Introduction

The complex variable boundary element method (CVBEM) is a coupled numerical analytic modeling technique that provides solutions to particular partial differential equations such as the Laplace and Poisson equations. The CVBEM has as its underpinnings the numerical integration of the Cauchy integral equation for analytic complex functions. Originally developed in 1981 as the CVBEM [6], the modeling technique has been the subject of numerous research papers and books (for example, see [7,8]). The methodology has been extended to three-dimensional problems and extensions to even higher spatial dimensions have been developed mathematically by Hromadka and Whitley [9]. The CVBEM has also been developed for problems using a Hilbert space setting as well as a collocation approach. Non-homogeneity, anisotropy, and other such topics have also been examined. Applications have been made in several areas of engineering including stress–strain, torsion, electrostatics, heat transfer, groundwater flow, contaminant transport in groundwater, freezing and thawing of algid soils, among others. It is noted that the CVBEM is not at all restricted to particular domain types such as typically found with other analytic methods such as Fourier series and so forth. Instead, the CVBEM can be applied to general industry level problems. Unlike domain type numerical methods such as the finite element method (FEM) or finite difference methods (FDM) that require nodal point specification upon the problem boundary and also throughout the domain of the problem, the CVBEM does not require nodes to be specified inside the problem domain. Furthermore, such domain methods do not develop a well-defined approximation function that can be evaluated throughout the problem domain (such as accomplished by the CVBEM), but instead provide point estimates at a set of nodal points. Domain methods like FEM and FDM do not have an equivalent modeling error display such as is provided by the CVBEM approximation function as applied to the approximation boundary approach used in this paper. This is because the CVBEM approximation is continuous over the problem domain and boundary, whereas the FEM and FDM are only point estimates at a finite set of particular points.

In this paper, the approximate boundary method for assessing the CVBEM in fitting problem boundary conditions is developed by using computer programs MATLAB and Mathematica. Of special interest are the matrix computational capabilities of MATLAB which

* Corresponding author. Tel.: 845 938 7685; Fax: 845 938 7690.
E-mail address: anthony.johnson@usma.edu (A.N. Johnson).

0955-7997/$ - see front matter Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.enganabound.2013.12.011

enable CVBEM models to be built involving more than a thousand nodes. Mathematica will be used for its internal graphical interface capabilities by way of MATlink, a Mathematica application module for seamless two-way communication and data transfer with MATLAB. This allows CVBEM matrix solutions to be developed by MATLAB whereas the plots of potential function and stream function isocontours are obtained using Mathematica. Consequently, the approximate boundary for use with the CVBEM is obtained from the Mathematica plots.

The approximate boundary is a geometric construct of an alternate boundary to the true problem boundary, where the CVBEM approximation function achieves the actual problem boundary condition values. The CVBEM satisfies the governing PDE for the Laplace and Poisson equations (in two or higher spatial dimensions), and fits the problem boundary conditions on the problem boundary either at collocation points where boundary condition values are specified, or in a least squares errors sense by using the usual inner product. In this paper, the collocation procedure [4] is used to develop the CVBEM approximation because the approximate boundary will intersect the true problem boundary at collocation points (i.e., the approximation achieves the boundary condition value at collocation points). Collocation points are then subsequently added to the true problem boundary at locations where departure is seen largest between the approximate and true boundaries. Adding or adjusting node and collocation points draws the approximate boundary closer to the true boundary, which intersects the true problem at the additional collocation points. Each CVBEM node is located either on the problem boundary or exterior of the boundary union in the enclosed domain. Because the CVBEM basis functions are products of complex polynomials with complex logarithms, logarithmic branch cuts are rotated with respect to the true boundary so that the branch cuts lie exterior of the problem domain. Accordingly, the resulting CVBEM approximation is analytic in the entire complex plane (or set of planes if three or higher dimensions are used) except at branch cuts and branch points of the nodal point logarithm functions. Therefore, the CVBEM approximation is analytic within the problem domain, on the problem boundary except at nodes (but continuous), and throughout the problem exterior except along branch-cuts. Isocontours may be developed from the CVBEM approximation using Mathematica, which correspond to level curves of the boundary condition values between collocation points for use in tracking the approximate boundary for the target problem. Collocation points are located at boundary corners, and close to but not on singularities of the boundary conditions. Generally, collocation point density is developed to increase at locations of high variation of boundary condition values.

The resulting approximate boundary by means of using a reasonable placement of collocation and nodal points is found to generally lay strikingly close to the true problem boundary, allowing one to surmise the conclusion that had the problem geometry and boundary conditions been the approximate boundary and associated conditions, then one has in-hand the solution to that alternative boundary value problem. Generally, one develops the approximate boundary (by adding or adjusting locations of nodes and/or collocation points) until the maximum departure from the true boundary is less than the construction tolerance for the prototype.

2. Mathematical development

The mathematical underpinnings of the CVBEM stem from the numerical integration of the Cauchy integral equation for an analytic function in a simply connected two-dimensional complex

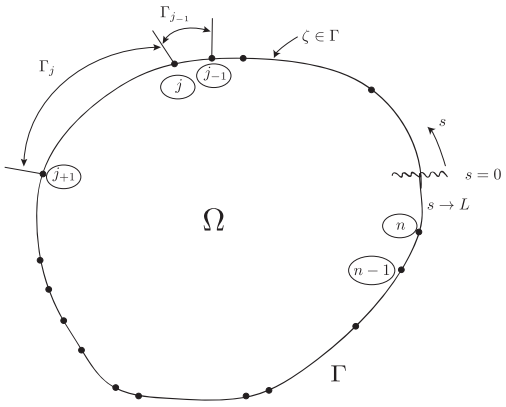

Fig. 1. CVBEM node and boundary element depiction.

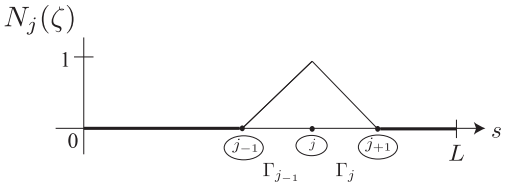

Fig. 2. Linear interpolation along problem boundary.

domain. Fig. 1 is a graphical depiction of the domain, Ω, of a simply connected region along with n nodal points on its boundary, Γ. For development purposes, nodes are assumed to be positioned on the problem boundary. (In application, nodes are typically made part of the optimization process where node location is an additional degree of freedom. Accordingly, nodes may be positioned exterior to the problem domain.) The boundary can be segmented into smaller sections or boundary elements denoted as Γ_j. Upon these subsections of the boundary an interpolating polynomial is applied. If one were to sever the boundary and lay it out flat from $s=0$ to $s=L$ as in Fig. 2, the values of the basis functions become apparent, attaining their maximum at node j and their minimum values at the nodes adjacent to j (i.e., nodes $(j-1)$ and $(j+1)$). Fig. 2 depicts the use of the usual basis functions. Higher order basis functions are examined in Hromadka and Whitley [8]. Using a piecewise continuous global interpolation function that is defined continuously on the problem boundary (that is assumed to be simple closed), the two-dimensional potential and stream functions are approximated on the problem boundary to the arbitrary level of accuracy. Eq. (1) is the Global Trial function used in the CVBEM. It is defined as a sum of basis functions multiplied individually by the corresponding nodal values. Depending on the nature of the Global Trial function, different variations of the CVBEM are possible.

$$G(\zeta) \equiv \sum_{j=1}^{n} N_j(\zeta)\overline{w}_j \tag{1}$$

For example, Bohannon and Hromadka [2] used a set of complex monomials to develop a Global Trial function that was a complex polynomial defined over the entire problem boundary, resulting in the complex polynomial method. The typical situation, however, is to use a set of basis functions that are polynomial interpolations between specified locations on the problem boundary, where such locations are defined by placement of nodal points. In such cases, the Global Trial function is a sum of such basis functions multiplied by complex coefficients [7]. The Global Trial function is then substituted into the Cauchy integral equation, resulting in the CVBEM approximation function, Eq. (2), defined over the problem boundary and the interior domain enclosed by the problem boundary. This

approximation is also analytic over the problem domain enclosed by the problem boundary. Given a basis function set of linear interpolating polynomials between nodes (all specified upon the problem boundary) as seen in Fig. 2, the CVBEM approximation function is seen to be a linear complex polynomial plus the sum of products of linear complex polynomials with complex logarithm functions as shown
in Eq. (3)

$$\hat{w}(z_0) = \frac{1}{2\pi i}\oint_{\Gamma} \frac{G(\zeta)\, d\zeta}{\zeta - z_0}; \quad z_0 \in \Omega, \;\; z_0 \notin \Gamma \tag{2}$$

$$\hat{w}(z) = P_1(z) + \sum_{j=1}^{n} \mathbb{C}_j (z - z_j)\, \mathrm{Ln}_j(z - z_j). \tag{3}$$

In (3) $\mathbb{C}_j$ are the complex constants $\mathbb{C}_j = \alpha_j + i\beta_j$; Ln_j is the notation for the complex logarithm function with argument $(z - z_j)$, with branch cut rotated not to intersect other branch cuts or the problem domain union boundary, and point z is not an element of any branch cut. If other polynomial basis functions are used, different CVBEM approximation functions result, but the overall constructs are similar to Eq. (3). Complex coefficient values may be estimated for a collocation approach. In all such cases, the real and also the imaginary parts of the resulting CVBEM approximation satisfy the Laplace equation over the problem domain.

If the CVBEM approximation function is viewed as a linear combination of analytic functions, then an inner-product may be used to expand the approximation function as a generalized Fourier series, with complex coefficients determined by using a Gram–Schmidt process.

Besides selection of the complex coefficients as degrees of freedom, the CVBEM approximation functions have additional degrees of freedom by virtue of selecting nodal point location, not only along the problem boundary, but also away from the boundary and exterior of the problem domain, similar to but different from the method of fundamental solutions [12,1]. Further, CVBEM adjustments include specification of the complex logarithm branch cuts for all the nodes such that the branch cuts do not intersect and lie exterior of the problem domain and problem boundary. With such branch cut orientation, the CVBEM approximation function is not only analytic over the problem domain enclosed by the problem boundary, but everywhere in the complex plane except along the branch cuts and at branch points (i.e., branch points being the CVBEM nodal points). Eq. (4), a polar coordinate form of Eq. (3), is one approach to incorporate the ability to rotate the logarithmic branch cuts of the individual nodes. Hence,

$$\hat{w}(z) = P_1(z) + \sum_{j=1}^{n} (\alpha_j + i\beta_j) R_j (\cos\ \theta_j + i\ \sin\ \theta_j)(\ln R_j + i\theta_j) \tag{4}$$

where $P_1(z) = \mathbb{C}_0 + \mathbb{C}_{-1} z$; α_j and β_j are the real components of the $\mathbb{C}_j$ complex constants; and R_j and θ_j are the polar coordinate representations with respect to node j. (It is further noted that the CVBEM has been extended to three-dimensional problems with virtually any reasonable geometry as the problem boundary. The CVBEM is not limited to specific geometries in order to be applied, like a typical Fourier series. Most geometries encountered in typical industrial or physical problems may be accommodated by the CVBEM.)

In this paper, the approach used for development of complex coefficients for use in Eq. (3) is collocation, where either the real or the imaginary part of the CVBEM approximation function is set to equal the problem boundary condition values at collocation point locations specified by the modeler. Of course, the real part of the CVBEM approximation corresponds to the usual potential function of the problem solution, and the imaginary part of the approximation corresponds to the stream function of the problem solution. Therefore, the approximation function's real or imaginary parts will be equal to the corresponding boundary condition values at collocation points specified along the problem boundary. Because the approximate boundary (as discussed in the prior section) is to be estimated as part of the CVBEM approximation effort, collocation point locations should typically include all corners of the boundary geometry. Additionally, at locations along the problem boundary where significantly above average variation in the boundary condition values is occurring, such as in the proximity of singularities or abrupt changes in boundary condition values, an increase in the density of collocation points provides a better CVBEM approximation due to better approximation of the boundary condition values along the problem boundary. More details regarding placement of nodes and collocation points is provided in the subsequent application sections below.

The implementation of the CVBEM onto computer programs Mathematica and MATLAB provides a breakthrough in the use of the CVBEM due to the significant increase in the number of nodal points possible in the CVBEM. Using these computer programs in tandem via MATlink, the authors have successfully applied the CVBEM to problems involving thousands of nodal points. This is important because the prior success of the CVBEM reported in the literature has been from applications involving typically several dozens of nodal points. Further research is needed in order to better quantify the magnitude of CVBEM problems that can be accommodated. The resulting much higher level of accuracy in matrix solutions enables an entirely new view of application of the CVBEM in industrial level problems.

3. Implementation using MATLAB

The CVBEM system matrix is the repository of all information surrounding the model domain and its simple-closed boundary. It is non-singular, well-posed and structurally suitable (i.e., well conditioned) for MATLAB to solve, and it does so both easily and most important efficiently. MATLAB uses routines from a collection of streamlined and highly optimized subroutines from LINPACK and EISPACK to perform numerical calculations [5,10]. Thus, it is optimized for operations involving matrices and vectors. This process is known as vectorization. It avoids the loop-based approaches to solving linear systems, which are more prone to error, require more lines of code to run (i.e., bulky), and have slower performance. Further, instead of defining `for` and `while` loops that incrementally increase the size of a data structure each time through the loop, MATLAB can preallocate large contiguous blocks of memory and move entire arrays into those blocks. This improves execution time because there is no need to repeatedly reallocate memory for growing matrix systems.

The CVBEM is coded in MATLAB as a function that performs four major tasks: retrieve geometry of the problem, calculate the nodal distances and angles, build the CVBEM system matrix, and solve for the unknown complex coefficients. The geometry of the problem is given as a $[(3n+3) \times 4]$ geometry matrix, which consists of $2n+3$ evaluation points and (n) nodes. The first column of the matrix distinguishes nodal point from evaluation point. In Fig. 3, the ASCII code 78, which corresponds to the character 'N',

Char	x coord	y coord	value
69.0000	1.0000	0	0
69.0000	1.0000	2.0000	0
69.0000	0	1.0000	100.0000
69.0000	2.0000	1.0000	0
69.0000	0	0.4444	100.0000
78.0000	2.2500	0.2955	3.1416

Fig. 3. Sample input geometry matrix with one node.

represents the nodes, while the ASCII code 69, which corresponds to the character 'E', represents the evaluation points. The second and third columns are the physical locations of the evaluation and nodal points in the x and y directions, respectively. Every point used in the mathematical calculation of the problem solution must have an x and y coordinate. The final column of the geometry matrix contains one of two types of information. If the row is a node coordinate, then the value of the fourth entry in that row will be the rotation value, $\Delta\theta$, which is used to rotate the branch cut of the particular node out of the domain space of the problem. This is a preliminary calculation taken by the modeler or it may be automatically computed for simply connected regions. If the row is an evaluation point, then the fourth entry in that row contains a known quantity at that point. The quantity may be real or imaginary. It may be a fixed constant, or simply the derivative at that point. Fig. 3 is a sample input geometry matrix of one node. Notice that the last row, a node, has a value of π, which is the $\Delta\theta$ rotation of the logarithmic branch cut out of the domain (by default branch cuts proceed to the left of nodes corresponding to a $\Delta\theta$ of zero). Retrieval of the information necessary to the mathematical formulation of the CVBEM system matrix is simply a matter of slicing the geometry matrix into column vectors and feeding it as input into our algorithm.

The next task for our CVBEM function is to calculate the nodal distances and angles. Fig. 4 depicts the location of an arbitrary evaluation point, z (denoted as a star), along the boundary of a two-dimensional domain. For each nodal point in the plane (note that nodes need not be on the actual boundary, and in some cases it may be better if they are not), the radius, R_j, and angles, θ_j, are calculated into column vectors. In Fig. 4, the n nodes are positioned around and on the problem boundary and are numbered in the counter-clockwise direction. MATLAB handles complex numbers efficiently. It does not require any special coding. Complex numbers are formed by simply appending `1i` to any term of a binomial representing the complex number. This makes specifying the z of the evaluation points simple since from the geometry matrix the x and y coordinates of each point are already given. To find the evaluation coordinate in the complex plane compute `z=x+y*1i`. The distance vector, R_j, is then calculated by taking the complex magnitude of $(z-z_j)$. We now have a single vector for any evaluation point (i.e., for each collocation point or node) on the boundary that contains the radii from the evaluation point to every node in the problem. Collecting all distance vectors together gives a $[(2n+3)\times n]$ R_j matrix which contains the radii to and from every evaluation point and node in the plane. The number of constants used in the CVBEM approximation function of Eq. (4) is $2n+3$ for n nodes and a linear basis function set. Subsequently,

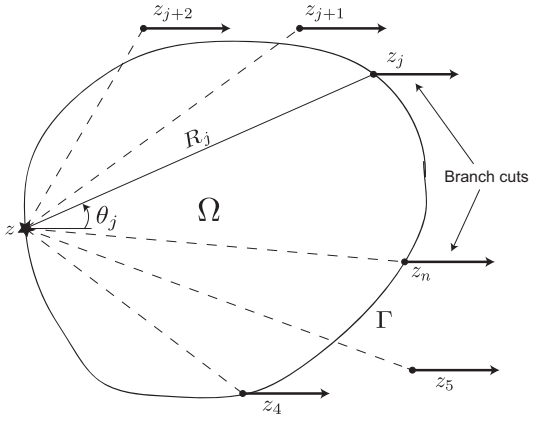

Fig. 4. A star representing the evaluation point along the boundary of a two-dimensional domain.

using the R_j matrix and the MATLAB `angle` function, calculate $(\theta_j+\Delta\theta_j)$, where $\Delta\theta_j$ represent the angles to rotate the branch cuts out of the domain.

With the R_j and the θ_j matrices defined, Euler's identity is applied. Because of MATLAB's vectorization property, $(z-z_j)$ is computed for the entire problem in one line, namely,

```
zminuszj=Rj.*exp(1i.*(thetaj)),
```

which is a code for $(z-z_j)=R_j e^{i\theta_j}$. The `(.*)` signals to the program to perform array multiplication, which is the element-by-element product of the arrays R_j and θ_j matrices. Having properly defined matrices for distances and angles in the form of $(z-z_j)$, the system matrix is constructed. Fig. 5 is the system matrix for a one node system (note that a one node system requires 5 evaluation points). The column vector in Fig. 5 represents the unknown coefficients of the approximating function. By setting this equal to the vector of known values at the evaluation points, we have a well-defined matrix system with a unique solution because the columns of the system matrix are linearly independent [11].

The CVBEM was used to calculate problems with nodes that ranged from 1 to over 1000. Solution of the CVBEM matrix system was assessed by tracking computational time. Fig. 6 is a plot of the number of nodes versus the time necessary to find a solution for the unknown coefficients. While from 1 to about 500 nodes show a slightly exponential rise in time, from 500 to 1000 nodes the computational time is nearly linear. In any event, calculating the solution to 1000 unknown coefficients took only 4 tenths of a second.

4. Demonstration problems

Example 4.1. A square domain with discontinuous boundary conditions.

The first example problem considered is the spatial domain of a unit square positioned in the first quadrant. Temperature values (i.e., potential function values) of 0° are specified along the square domain top, bottom, and right side. A temperature value of 100° is specified along the left side. Shown in Fig. 7 is a CVBEM plot of approximation function streamline contours using only 15 nodal points where the nodes are positioned as part of the CVBEM modeling procedure such as to visually reduce the departure between the approximate boundary and the true problem boundary. Two nodes are located closely together at the upper and lower left corners to better model the abrupt change in the value there. Also, in Fig. 7, one can see the CVBEM approximation streamline function as evaluated in the interior and also the exterior of the problem domain. The problem domain is not shown in Fig. 7; rather, the approximate boundary from the CVBEM is shown. The CVBEM approximate boundary was developed by program Mathematica given specified values for the CVBEM potential function level curves in the Mathematica automated plotting routines. Seen in Fig. 7 are some of the branch cuts.

The subject example problem is further analyzed by the CVBEM using a 512 evenly spaced node model. The resulting CVBEM approximation function, potential function and streamline function contours are shown plotted together in the flownet of Fig. 8.

$$\begin{bmatrix} 1.0000 & 1.0000 & 0.2443 & 0 & -0.3641 \\ 1.0000 & 1.0000 & -0.6634 & -2.0000 & 2.4484 \\ 1.0000 & 0 & 1.7160 & -1.0000 & 1.2871 \\ 1.0000 & 2.0000 & -0.9392 & -1.0000 & 0.1025 \\ 1.0000 & 0 & 1.8197 & -0.4444 & 0.2699 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

Fig. 5. CVBEM system matrix for one node.

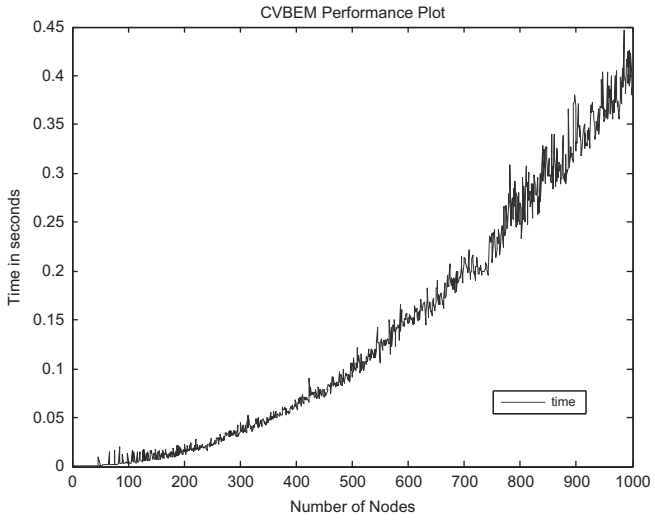

Fig. 6. Computational time versus number of nodes for CVBEM.

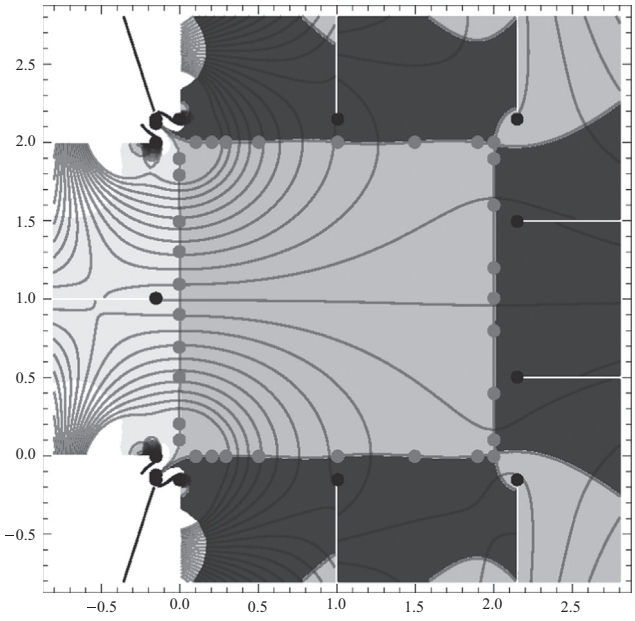

Fig. 7. Mathematica plot of the CVBEM approximate boundary for Example 4.1, including streamlines and branch cuts. Nodes shown as dots.

Also seen in Fig. 8 is the CVBEM approximate boundary as well as CVBEM approximation results located exterior of the problem domain. From Fig. 8, the effect of singularities in the potential function boundary condition values at the upper and lower left corners of the domain can be seen. Similar singularity effects are seen at the right side top and bottom corners even though the potential function boundary condition values remain a constant value of zero. From the CVBEM results seen in Figs. 7 and 8, the approximate boundary depicts a problem domain that is almost identical to the original problem domain and boundary.

In order to demonstrate the closeness between the CVBEM approximations as depicted in Fig. 8, versus the solution obtained from Fourier series, two pathlines are considered where both approximations are compared; namely, the diagonal pathline of $y=x$ that bisects the square domain from the lower left corner to the upper right corner, and also the horizontal pathline of $y=0.5$ that runs along the boundary of the lower quartile of the square domain. A plot of both nearly identical approximation functions along these

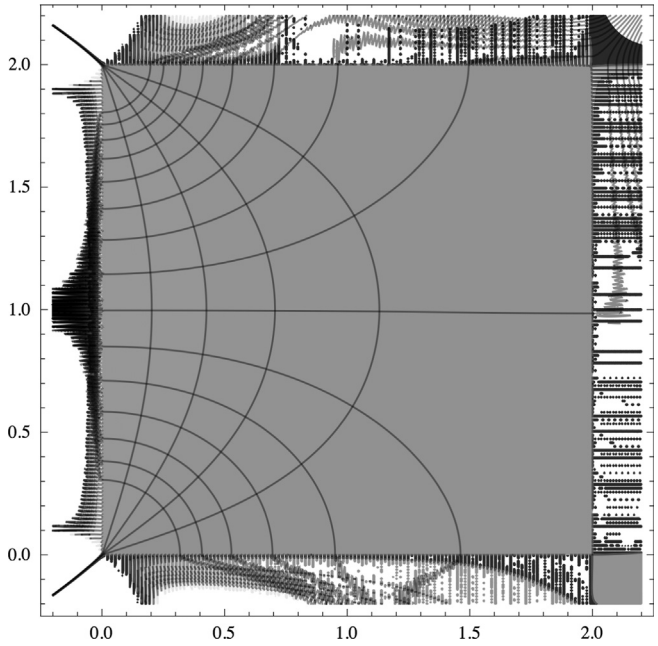

Fig. 8. Mathematica plot of the CVBEM approximate boundary, including display of branch cuts and CVBEM streamlines. $n=512$.

two pathlines is shown in Fig. 9a and b. It is noted that in Fig. 8, the flownet is plotted for the subject problem. The CVBEM produces both the streamlines and the orthogonal potentials. For the test problem, boundary conditions are specified as the potential values at a given (x, y) coordinate. Similarly, in Fig. 9a and b, values of the potential functions are compared along the prescribed pathlines. Further analysis, keeping in mind the maximum principle, estimates the departure from the true boundary by plotting the values on the boundary and comparing them to the true boundary of the problem. Fig. 10 represents values along the boundary of Example 4.1. Along the x-axis from zero to one represents one side of the domain, then two to three and so on. Three sides of the problem domain are set to 0° while the fourth side is set to 100°.

Example 4.2. Ideal fluid flow in degree bend.

Using the unit square domain in the first quadrant, ideal fluid flow in a 90° bend is described by the analytic function $w(z)=z^2$. In this example, there are no discontinuities as considered in Example 4.1 other than the boundary corners. The potential function is given by $\phi(x,y)=\mathrm{Re}[w(z)]=x^2-y^2$ and the streamline function is $\psi(x,y)=\mathrm{Im}[w(z)]=2xy$. The CVBEM is applied as described in Example 4.1. Modeling error throughout the domain is determined by a set of 1000 evaluation points per side on the boundary in the same way nodes and evaluation points are set up. The error at each evaluation point is determined by taking the known value at the point given by x^2-y^2 and subtracting the value given by the CVBEM at the same point. The L_2 error is given by squaring the exact value as defined by the problem minus the CVBEM given value. The L_∞ error is found by searching for the max difference between the value as defined by the function and the CVBEM value. As the number of nodes increases, the modeling error decreases as shown in Table 1.

Fig. 11 depicts the flow-net and approximate boundary for the ideal fluid flow problem of Example 4.2, with streamlines plotted as dashed lines and potentials plotted as solid lines. It is noted that the CVBEM develops both the potential and stream functions concurrently even though the problem is modeled as a Dirichlet problem involving only boundary conditions of the potential function.

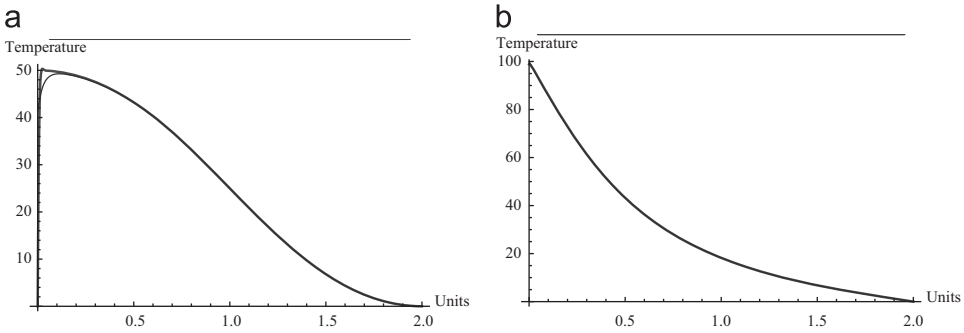

Fig. 9. Temperature values along two lines for Example 4.1. (a) Along the line $y=x$ and (b) along the line $y=\frac{1}{2}$.

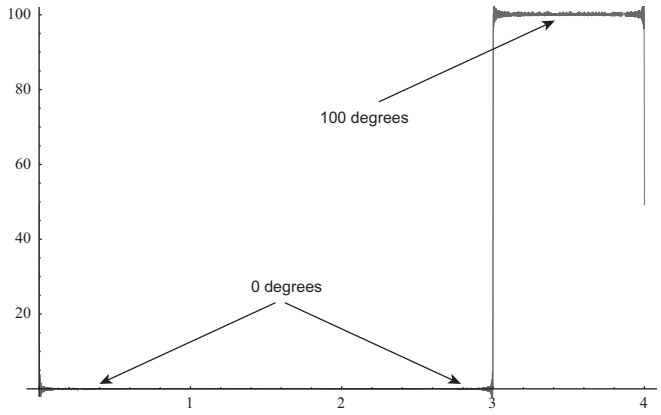

Fig. 10. CVBEM values along the boundary.

Table 1
Modeling error versus CVBEM nodes.

# of nodes	L_2	L_∞
8	0.259634	0.0135822
16	0.0820735	0.00523
32	0.021512	0.00177898
64	0.00436868	0.000495219
128	0.000563918	9.22912×10^{-05}
256	3.16339×10^{-05}	7.79644×10^{-06}
512	3.96644×10^{-07}	1.38827×10^{-07}

5. Analogy of the CVBEM to conformal mapping

In the development of the approximate boundary, the CVBEM approximation function can be viewed in terms of conformal mapping between the two domains of the original problem boundary and its associated enclosed domain, versus the approximate boundary and its associated enclosed domain. Under this interpretation, the CVBEM provides a conformal mapping of the problem boundary and the enclosed domain to an alternate geometry for the boundary and the enclosed domain. And, by the addition of nodes and collocation points, or by adjusting the location of nodes and collocation points, the conformal mapping is modified until the alternate geometry (i.e., "the approximate boundary") is visually "closer" to the true problem geometry. Given that the corresponding CVBEM model operates upon the approximate boundary, the modeling goal is to develop an approximate boundary that is as "close" as possible to the true problem boundary. This approach is assessing modeling error and provides a more visual representation.

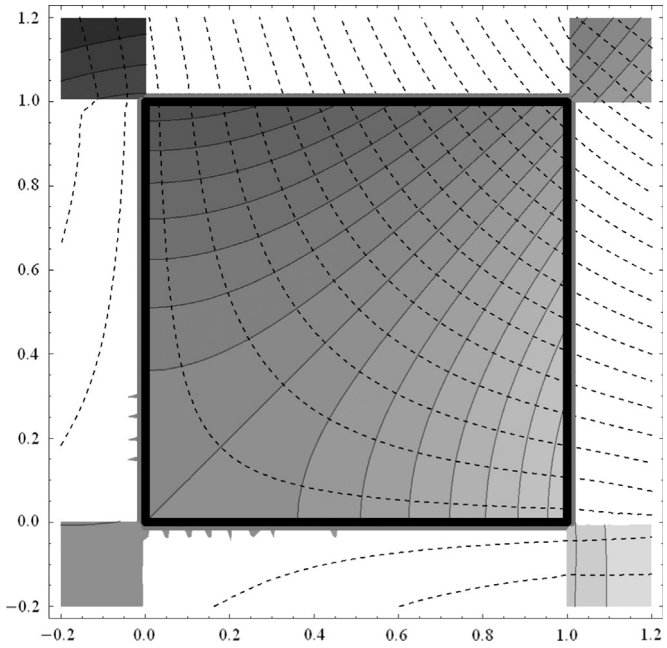

Fig. 11. CVBEM flow-net and approximate boundary for Example 4.2.

6. "Chasing" discontinuities in boundary conditions with the approximate boundary approach

Discontinuities in boundary conditions may occur in the problem formulation primarily through discontinuities in the boundary condition values or in their derivatives. Similarly, discontinuities may occur by the geometric construct of the problem boundary. Although it is commonplace to consider problem geometries with straight lines and corners, and with boundary condition values specified on the problem boundary as constant values along such straight lines and across corners, the reality is that such specifications may involve discontinuities. This is particularly true with boundary condition values specified as piecewise constants along the problem boundary. In the theory of Fourier series, piecewise constant boundary conditions are frequently encountered and the developed Fourier series converges to such piecewise constant values along the problem boundary except that a midpoint value is given by the series at the location of the step between piecewise constant values. However, although theoretically tractable, the computational approach is limited to partial sums of the Fourier series which is only a portion of the total series and, therefore, does not equal the exact solution. Similarly, there typically is a "Runge phenomenon" oscillation (see [3]) provided by these partial sums obtained from the Fourier series. In general, only the entire infinite term Fourier series is the exact solution to the boundary value problem. In the approximate boundary approach, the approximate

boundary also demonstrates oscillation and departure in the vicinity of singularities (such as may occur with boundary condition values or at problem geometry corners). To deal with such situations, additional nodes and collocation points are added to the CVBEM model at locations of singularities and at problem corners so as to intersect the approximate boundary with the true problem boundary. For example, in a problem situation involving an isosceles triangle, with a boundary condition value of 0 assigned along two sides and a value of 100 assigned along the third side, not only are there singularities created by the change in boundary condition values (from 0 to 100) at the corners, but also from the sharp corner geometry while holding the boundary condition value constant. In the CVBEM approximation, additional nodes and collocation points were added in the vicinity of the corners, improving computational approximation results as seen by a reduction in the departure between the approximate and true boundaries.

7. Conclusions

The CVBEM is extended towards use on problems involving over a thousand nodal points and collocation points, using computer programs MATLAB and MATLink as the computational engine and using program Mathematica for graphics. Given this computational capability, the approximate boundary technique for depicting CVBEM model error in fitting problem boundary conditions is shown to be a viable and useful approach, rather than developing the usual plots of function values versus specified boundary condition values along the problem boundary. The modeling approach now becomes one of creating a problem boundary and an enclosed domain that is "close" to the true problem boundary and the enclosed domain, which is a much more visual measure of modeling error and more tractable to locating where additional modeling nodes and collocation points should be added.

References

[1] Barrero-Gil A. The method of fundamental solutions without fictitious boundary for solving stokes problems. Comput Fluids 2012;62:86–90.
[2] Bohannon A, Hromadka T. The complex polynomial method with a least-squares fit to boundary conditions. Eng Anal Bound Elem 2009;33(8–9):1100–2.
[3] Boyd J. Six strategies for defeating the Runge phenomenon in Gaussian radial basis functions on a finite interval. Comput Math Appl 2010;60(12):3108–22.
[4] Dean T, Hromadka T. A collocation CVBEM using program Mathematica. Eng Anal Bound Elem 2009;34(2010):417–22.
[5] Golub G, Van Loan C. Matrix computations. 3rd ed. Baltimore, MD: Johns Hopkins University Press; 1996.
[6] Hromadka TV, Guymon GL. The complex variable boundary element method. Int J Numer Methods Eng 1984;20(1):25–37.
[7] Hromadka TV, Lai C. The complex variable boundary element method. New York: Springer-Verlag; 1987.
[8] Hromadka TV, Whitley RJ. Advances in the complex variable boundary element method. New York: Springer; 1998.
[9] Hromadka TV, Whitley RJ. Approximating three-dimensional steady-state potential flow problems using two-dimensional complex polynomials. Eng Anal Bound Elem 2005;29:190–4.
[10] Kwon Y, Bang H. The finite element method using MATLAB. 2nd ed. 2000 N.W. Corporate Blvd., Boca Raton, FL 33431: CRC Press LLC; 2000.
[11] Lay D. Linear algebra and its applications (book and CD-ROM). 3rd updated ed. Addison Wesley; 2005.
[12] Reutskiy. The method of approximate fundamental solutions (MAFS) for elliptic equations of general type with variable coefficients. Eng Anal Bound Elem 2012;36(6):985–92.