

# COMPARISON OF TWO ALGORITHMS FOR LOCATING COMPUTATIONAL NODES IN THE COMPLEX VARIABLE BOUNDARY ELEMENT METHOD (CVBEM)

BRYCE D. WILKINS<sup>1</sup>, THEODORE V. HROMADKA II<sup>2</sup> & JACKSON MCINVALLE<sup>3</sup>

<sup>1</sup> Carnegie Mellon University, USA.

<sup>2</sup> United States Military Academy, USA.

<sup>3</sup> Duke University, USA.

## ABSTRACT

In this paper, we introduce a new node positioning algorithm (NPA) for determining suitable locations of the computational nodes that are a typical feature of mesh reduction numerical methods for partial differential equations – specifically, the Complex Variable Boundary Element Method (CVBEM). The novelty of the introduced NPA is a ‘position refinement’ procedure, which facilitates the relocation of nodes that are already being used in the current CVBEM model when such relocation reduces the maximum error of the associated CVBEM model. The results of the new NPA (referred to as NPA2) are compared to the results obtained using the recent NPA described in [1] (referred to as NPA1). We compare NPA1 and NPA2 by modeling two example Dirichlet boundary value problems that have been selected due to having regions of extreme curvature in the analytic flow regime that are difficult to model computationally. Consequently, these problems serve as good benchmark problems for testing the efficacy of the current and future NPAs. Our empirical findings suggest that the use of NPA2 can reduce the maximum error of the associated CVBEM model by several orders of magnitude compared to the corresponding result obtained using NPA1.

*Keywords: applied complex variables, Complex Variable Boundary Element Method (CVBEM), mesh reduction methods, node positioning algorithms (NPAs), potential flow.*

## 1 INTRODUCTION

Mesh generation is often the most time-consuming step when modeling a partial differential equation (PDE) using popular domain discretization techniques such as the finite element method or the finite volume method [2]. For this reason, some recent research has been directed toward increasing the viability of mesh reduction numerical methods [3–7]. Many mesh reduction methods require the user to select locations for computational nodes (also referred to as source points) during the modeling process. For these particular numerical methods, the accuracy of the resulting boundary value problem (BVP) model depends on the locations of the computational nodes. Since these locations are determined by the modeler, it is useful to develop procedures for selecting suitable locations for the computational nodes, such that the resulting BVP model is accurate and can be calculated in a timely manner.

Several node-positioning algorithms (NPAs) have been proposed to address the problem of efficiently determining suitable locations for computational nodes. However, ‘a practical and efficient way of achieving (an optimal location of the computational nodes has) yet to be found’ [8]. Instead, the relevant research has primarily been focused on devising NPAs that result in BVP models with at least satisfactory – even if not necessarily optimal – accuracy. Likewise, this work does not attempt to provide an NPA that determines the optimal locations for computational nodes with respect to error minimization of the resulting BVP model. Rather, we propose an improvement to the NPA described in [1], which is referred to as NPA1 in this work. Our proposed NPA, referred to as NPA2, improves upon NPA1 by incorporating a new node position refinement procedure. The refinement procedure allows for the

possible replacement of already selected nodes when it is discovered that there exists a different candidate node whose use when replacing a node already in use would result in a BVP model with less error in the chosen norm.

The refinement procedure is implemented by temporarily exchanging an existing node of the BVP model with an unused node, one node at a time. If the considered exchange reduces the maximum error of the BVP model, then the exchange is made permanent. Otherwise, the original node is reincorporated in the BVP model and a new possible exchange is examined. Consequently, each time the refinement procedure is applied to an existing node, one of two possibilities will happen to the current BVP model:

1. The current BVP model is kept and there is no change in the approximation error of the model. This is the situation when no possible exchange would reduce the error of the BVP model.
2. An existing node is exchanged for a new node in a different location, resulting in a BVP model that more closely satisfies the given boundary conditions and, thereby, reduces the error of the BVP model.

Importantly, the second possibility is only implemented if the considered exchange of nodes reduces the approximation error of the BVP model. Hence, the refinement procedure has a monotonic, non-increasing effect on the approximation error of the BVP model.

In this work, NPAs 1 and 2 are coupled with the Complex Variable Boundary Element Method (CVBEM), which is a numerical solver for the Laplace equation and related PDEs [9–11]. The coupled CVBEM and NPA programs will be used to solve two Dirichlet BVPs of the Laplace equation, which have been selected due to the computational difficulty of modeling potential flow in areas where there is extreme curvature in the flow regime.

## 2 THE NODE POSITIONING ALGORITHM

Let  $\Omega \subseteq \mathbb{C}$  be a simply connected domain. The set  $\Omega$  denotes the problem domain and modeling area of interest. The first step of the NPA is to define a region, denoted  $\mathcal{N} \subseteq \mathbb{C}$ , where candidate computational nodes will be located.

It is usual to impose restrictions on where the candidate computational nodes can be located. For example, in the CVBEM, computational nodes must be located in the exterior of the problem domain. Furthermore, while original implementations of the CVBEM used computational nodes located exclusively on the boundary of the problem domain, such as in [12], newer implementations have focused on locating the computational nodes in the exterior of the set  $\Omega \cup \partial\Omega$ , as demonstrated in [13].

For real numbers  $a, b, c$ , and  $d$ , a common definition of  $\mathcal{N}$  is as follows:

$$\mathcal{N} = \{x + iy \in \mathbb{C} : a \leq x \leq b, c \leq y \leq d, x + iy \notin \Omega \cup \partial\Omega\}. \quad (1)$$

The next step is to discretize the region defined by  $\mathcal{N}$  into a finite set of candidate computational nodes. The number of candidate computational nodes that are used to discretize  $\mathcal{N}$  is a parameter that can be determined by the modeler, but we note that we have had success discretizing  $\mathcal{N}$  with between 250 and 1,000 candidate computational nodes. The number of candidate computational nodes is denoted  $n_c$ , and  $\mathcal{N}_{n_c}$  is used to denote a discretization of  $\mathcal{N}$  with  $n_c$  candidate nodes.

Let  $n \leq n_c$  denote the number of computational nodes that the user has specified for use in the formulation of the approximation function. The NPA is designed to select a subset of  $n$

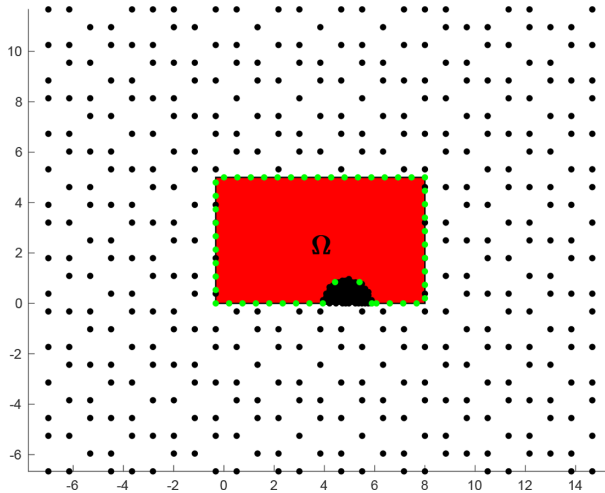


Figure 1: Example discretization  $\mathcal{N}_{500}$ . Candidate computational nodes are shown as black dots. Candidate collocation points are shown as green dots. For visualization purposes, only 50 collocation points are shown, but we typically use 1,000 such points. The modeling area of interest is shown in red.

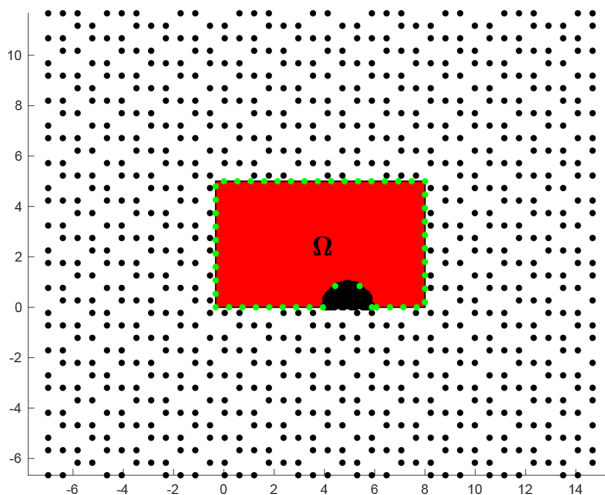


Figure 2: Example discretization  $\mathcal{N}_{1,000}$ . Candidate computational nodes are shown as black dots. Candidate collocation points are shown as green dots. For visualization purposes, only 50 collocation points are shown, but we typically use 1,000 such points. The modeling area of interest is shown in red.

nodes from the set  $\mathcal{N}_c$  of candidate nodes. The nodes selected by the NPA become the nodes used in the implementation of the mesh reduction PDE method. For both NPAs 1 and 2, the required  $n$  nodes are selected one at a time, as opposed to selecting multiple nodes, or all  $n$  nodes, at once. Consequently, when the NPA is determining the location of the  $i$ th node (with  $2 \leq i \leq n$ ), the locations of the  $(i - 1)$  already selected nodes and the locations of the  $2i$  already selected collocation points are considered fixed. Therefore, determining the location of the  $i$ th node is reduced to a single variable optimization problem where the only independent

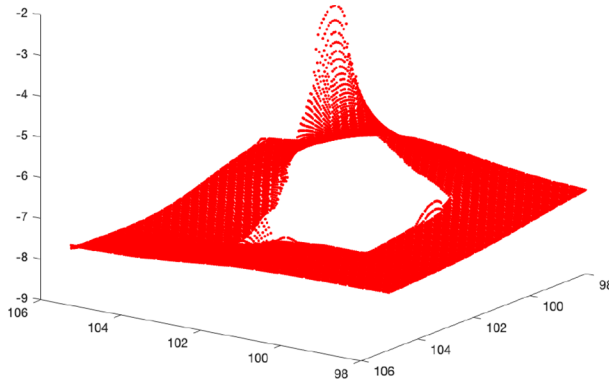


Figure 3: This is a representative illustration of the objective function for NPAs 1 and 2. In NPAs 1 and 2, we evaluate each candidate node to see what the maximum error of the CVBEM model would be if we were to use the corresponding basis function in the approximation function. This figure depicts the logarithm (base 10) of the maximum error of each potential CVBEM model corresponding to one of the currently available candidate node locations. The location of the  $i$ th node is determined by searching exhaustively for the candidate location that minimizes this objective function.

variable is the location of the  $i$ th node, and the objective function is the error of the corresponding BVP model.

After testing all of the currently available candidate computational nodes (i.e. the candidate computational nodes that are not presently being used in the CVBEM model) against the objective function, the  $i$ th node is selected as the candidate computational node that results in the BVP model of  $i$  total nodes with the least maximum error in the BVP model.

After determining the location of the  $i$ th node, the refinement procedure allows for the first  $i - 1$  nodes to be exchanged for different candidate nodes if the proposed swap would result in a BVP model with less error in the chosen norm. The value of each proposed swap is measured using the objective function described above.

The steps for implementing the new NPA are described below. These steps are presented specifically for use with the CVBEM, which requires two collocation points to be selected for each node that is selected. However, this algorithm can be adapted for use with other numerical PDE methods that only require selecting one collocation point for each computational node used in the model.

1. **Input:**  $n \leq n_c$ , which is the number of nodes that need to be selected. For the CVBEM,  $n$  is also the number of terms used in the linear combination of the approximation function.
2. **Create:** a set of candidate computational nodes,  $\mathcal{N}_{n_c}$ , and a set of candidate collocation points. The set of candidate collocation points must include at least  $2n$  points when applied to a CVBEM model.
3. **Initialize:** select two collocation points to be used in the model.
4. **Repeat (a)–(d) for  $k = 1, \dots, n$ :** construct the BVP model one node at a time by repeating the following steps.
  - a. Test all of the currently available candidate computational nodes to see which node

- will result in the approximation function of least error when added to the BVP model. Add the new computational node to the model. The model now has  $k$  computational nodes and  $2k$  collocation points.
- b. **Repeat (i)–(iii) for  $j = 1, \dots, M_1$ :** where  $M_1 \in \mathbb{Z}$  is the number of iterations of refinement to be performed after the selection of each new node.
    - i. Let  $p = \text{mod}(j, k)$  denote the index of the computational node whose location is being refined. If  $p = 0$ , then set  $p = k$ .
    - ii. Remove the  $p$ th-selected node from the CVBEM model. The model now has  $k - 1$  nodes and  $2k$  collocation points.
    - iii. Test all of the candidate computational nodes to see which node will result in the CVBEM model of least maximum error. Add the new computational node to the model replacing the  $p$ th-selected node that was removed in step 4(b)ii. The model now has  $k$  computational nodes and  $2k$  collocation points.
  - c. Use the  $k$ -term approximation function to measure the error of the BVP model with respect to fitting the known boundary conditions.
  - d. If  $k \neq n$ : Identify the two local maxima of the error function that have the greatest magnitude and add one collocation point to the BVP model corresponding to each of the two maxima. The model now has  $k$  computational nodes and  $2(k + 1)$  collocation points.
5. **Repeat (a)–(c) for  $j = 1, \dots, M_2$ :** where  $M_2 \in \mathbb{Z}$  is the number of iterations of refinement to be performed after selecting all  $n$  nodes.
    - a. Let  $p = \text{mod}(j, k)$  denote the index of the computational node whose location is being refined. If  $p = 0$ , then set  $p = k$ .
    - b. Remove the  $p$ th-selected node from the CVBEM model. The model now has  $k - 1$  nodes and  $2k$  collocation points.
    - c. Test all of the candidate computational nodes to see which node will result in the CVBEM model of the least maximum error. Add the new computational node to the model replacing the  $p$ th-selected node that was removed in step 5(b). The model now has  $k$  computational nodes and  $2k$  collocation points.

In the algorithm above, the refinement procedure is implemented twice: once at step 4(b) and for the second time at step 5. The implementation that occurs at step 4(b) refines the locations of the computational nodes as each new node is added to the model. The implementation that occurs at step 5 only refines the locations of the computational nodes after all  $n$  nodes have been selected. If a faster implementation of the NPA is desired, step 4(b) can be omitted.

### 2.1 Illustration of the refinement procedure

The critical difference between NPAs 1 and 2 is the refinement procedure. As previously mentioned, the refinement procedure allows for the possible replacement of already selected nodes when it is discovered that there exists a different candidate node whose use when replacing a node already in use would result in a BVP model with less error in the chosen norm. To illustrate this idea, we will walk through the refinement procedure being applied to a hypothetical BVP model of three computational nodes and six collocation points. The refinement procedure can be thought of as occurring in the following phases:

**Phase 1 – Initial orientation:** Figure 6 depicts the initial orientation of the algorithm-selected collocation points and algorithm-selected computational nodes of a hypothetical BVP model.

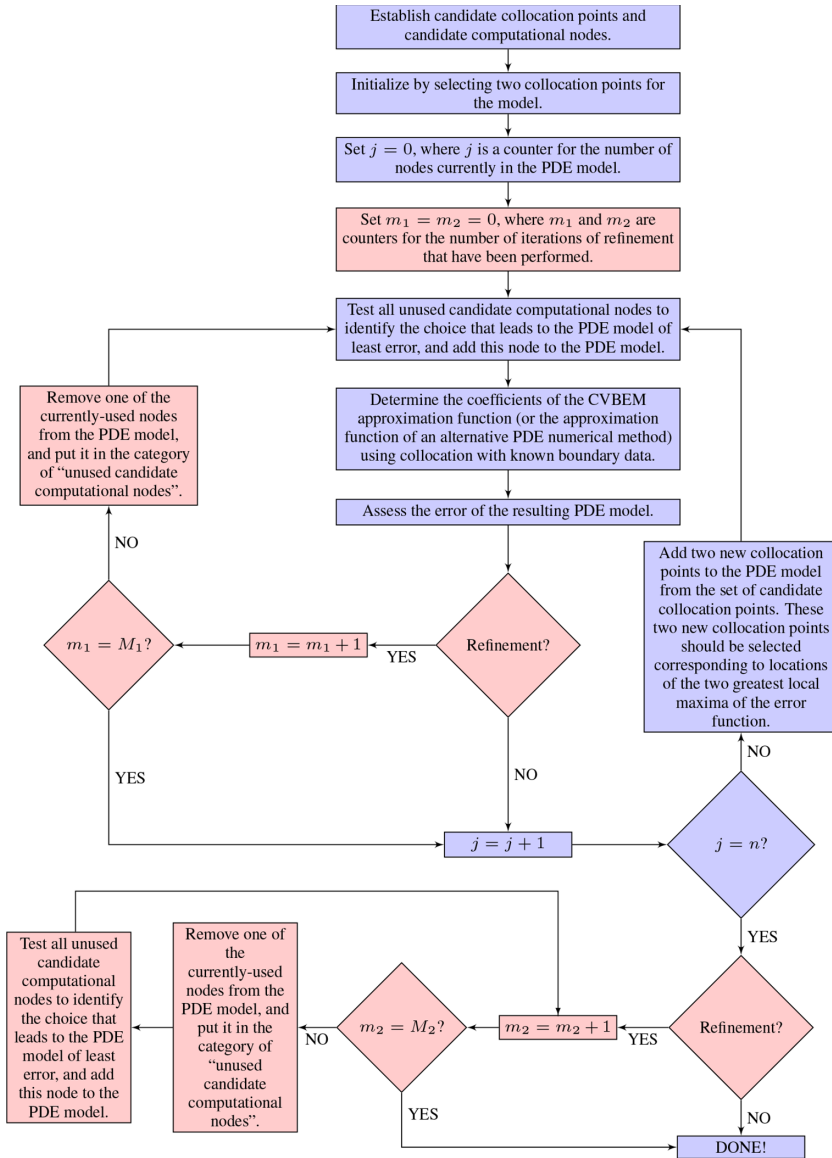


Figure 4: Flow chart depicting the steps of NPAs 1 and 2. The steps that are unique to NPA2 are colored red. The steps that are shared by both NPA1 and NPA2 are colored blue.  $M_1$  denotes the user-specified number of iterations of refinement to be performed upon the selection of each new node.  $M_2$  denotes the user-specified number of iterations of refinement to be performed after all  $n$  nodes have been identified.

**Phase 2 – Location updates:** The steps presented in Figs 7–10 show the locations of the algorithm-selected nodes changing during the refinement process as new BVP models are found that further reduce the approximation error.

**Phase 3 – No further location updates:** Figures 11 and 12 show that no new BVP models have been found that would further reduce the approximation error, so the





-  Current node under re-assessment.  
A better location has been found.
-  Node was re-assessed, and no better location found.
-  Fixed node
-  Fixed collocation point

Figure 5: Legend of symbols used in the refinement illustration in Figs. 6–13.

locations of nodes 2 and 3 are not changed. Once no more changes to the locations of the algorithm-selected nodes are made, the final positions of the algorithm-selected nodes are used.

**Phase 4 – Final orientation:** Figure 13 depicts the final orientation of the algorithm-selected nodes after the refinement procedure has been executed.

In both NPAs 1 and 2, nodes are selected one at a time until a BVP model of  $n$  nodes has been assembled. In NPA1, once a node is selected, it is permanently selected. However, in NPA2, we re-evaluate already selected nodes to see if a different node might further reduce the maximum error based on the selection of new nodes and collocation points that have occurred since the node in question was originally selected.

If a different node does, in fact, further reduce the error of the BVP model, then we exchange the old node for the new one that has been determined to reduce the error. While this search for a new node is being conducted, all of the other nodes and collocation points

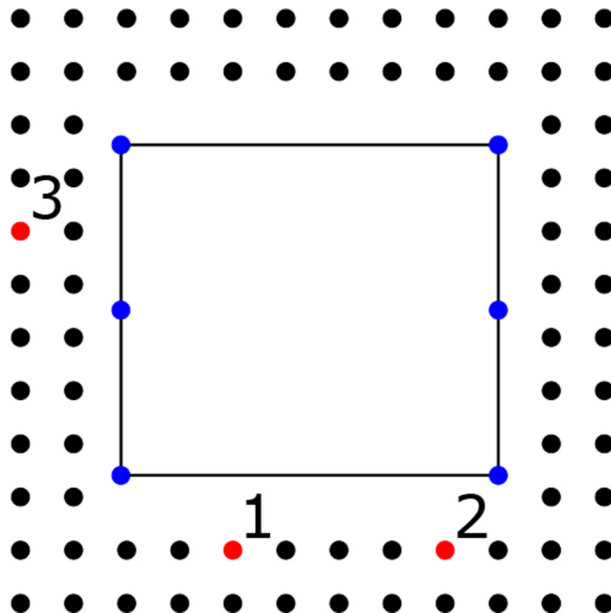


Figure 6: The starting situation after three nodes and six collocation points have been selected.

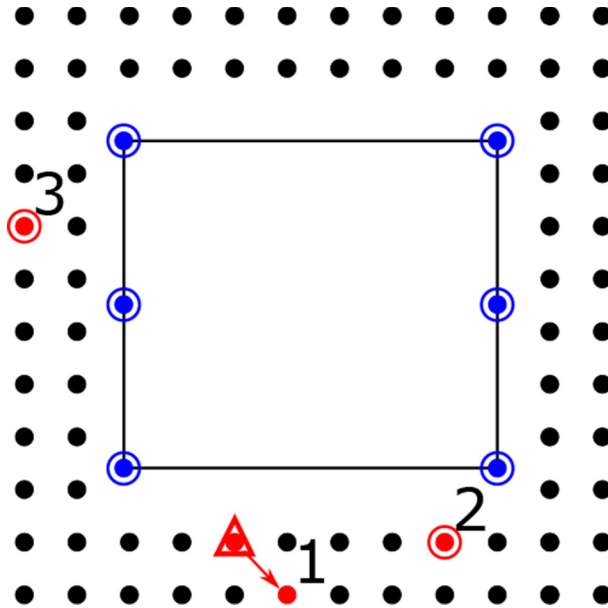


Figure 7: The position of Node 1 is being re-evaluated to see if a different position would further reduce the maximum error of the BVP model. In this case, a better position is found, so Node 1 is moved.

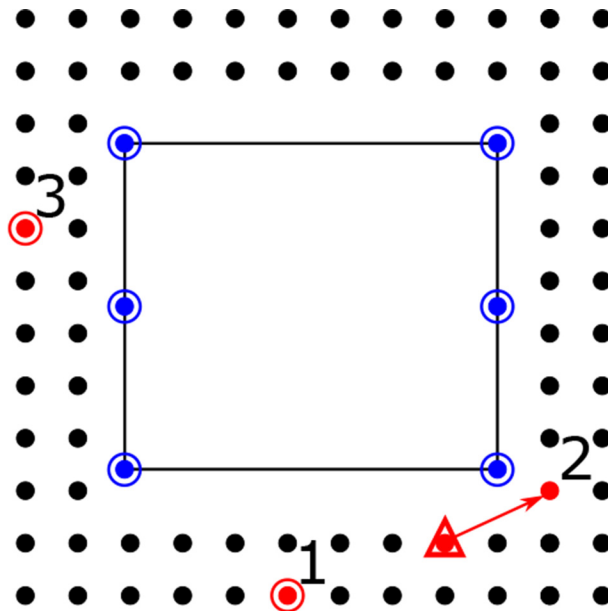


Figure 8: The position of Node 2 is being re-evaluated to see if a different position would further reduce the maximum error of the BVP model. In this case, a better position is found, so Node 2 is moved.



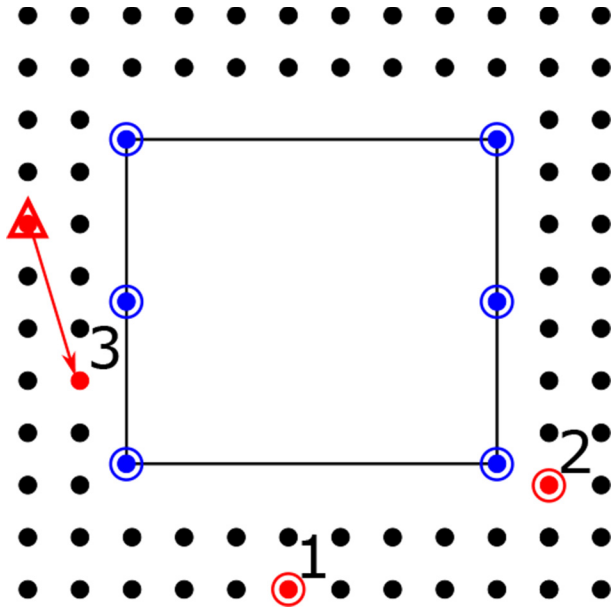


Figure 9: The position of Node 3 is being re-evaluated to see if a different position would further reduce the maximum error of the BVP model. In this case, a better position is found, so Node 3 is moved.

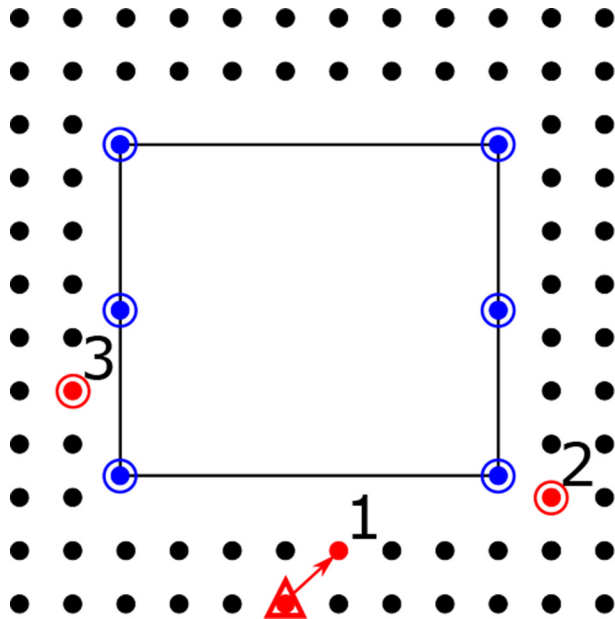


Figure 10: The position of Node 1 is being re-evaluated to see if a different position would further reduce the maximum error of the BVP model. In this case, a better position is found, so Node 1 is moved.

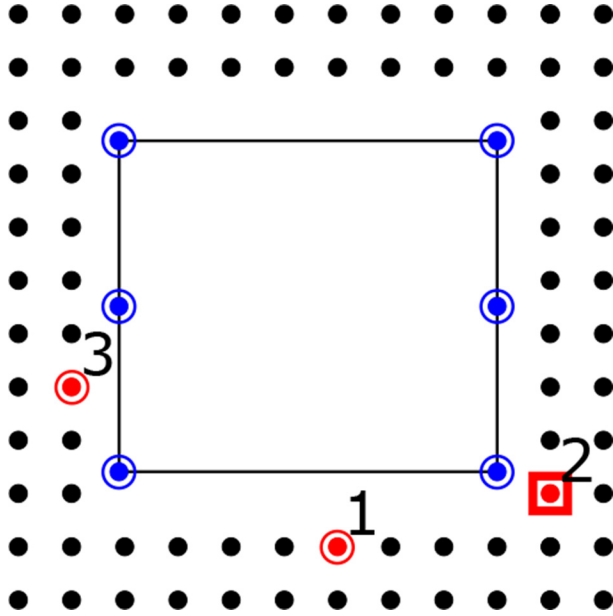


Figure 11: The position of Node 2 is being re-evaluated to see if a different position would further reduce the maximum error of the BVP model. In this case, NO better position is found, so Node 2 does not move.

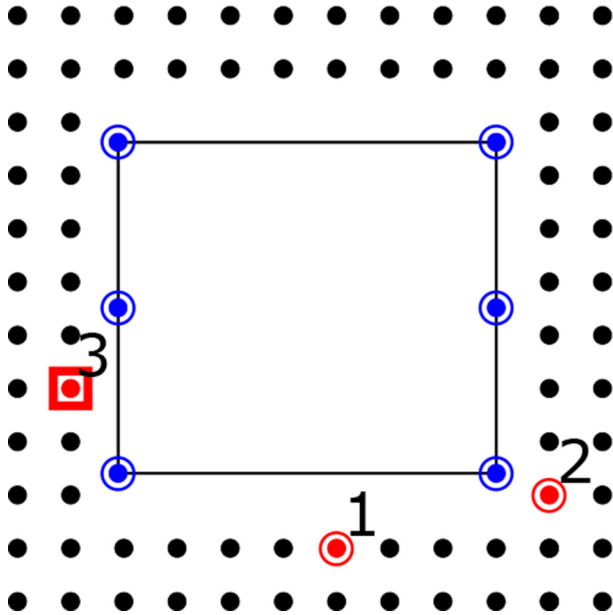


Figure 12: The position of Node 3 is being re-evaluated to see if a different position would further reduce the maximum error of the BVP model. In this case, NO better position is found, so Node 3 does not move.

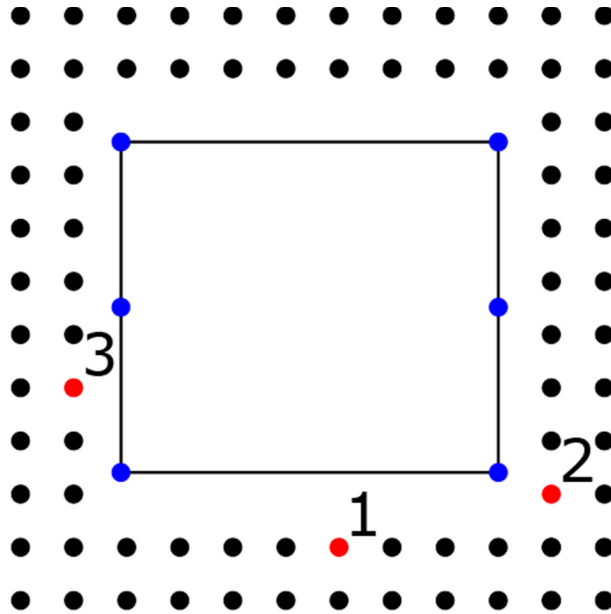


Figure 13: The final situation after the refinement procedure is complete. The next step is to determine the locations of two new collocation points and then to determine the location of Node 4. After these positions have been determined, the refinement procedure will be implemented again.

that are currently in the BVP model are considered to be fixed, so that the search amounts to a single variable optimization problem, which can be solved by exhaustively checking all candidate nodes.

### 3 APPLYING THE NPA TO A CVBEM MODEL

In this work, we demonstrate the NPA by coupling it with the CVBEM to determine the locations for the computational nodes for two potential flow problems. When applying the NPA to any PDE numerical approximation method, it is necessary to have a familiarity with the underlying method. Consequently, this section includes a brief review of the CVBEM methodology for the purpose of highlighting some of the considerations that are useful to make when integrating the NPA with a PDE numerical approximation method. Only a general overview of the CVBEM is given, and the reader is referred to [14–18] for more detailed descriptions of the CVBEM.

The CVBEM is a generalized boundary integral equation method based on the Cauchy integral theorem for analytic complex variable functions [12]. The CVBEM is related to other numerical techniques such as the real variable boundary element method, see [19, 20], and the method of fundamental solutions, which involve placing computational nodes along or near the boundary geometry and defining basis functions of particular types that typically solve the governing PDE under study. While the CVBEM was originally conceived to approximate the solution to two-dimensional Laplace BVPs, the method has since evolved to be capable of approximation of Laplace BVPs in three and higher spatial dimensions [21]. Furthermore, the CVBEM has also been expanded over the years to include capabilities of modeling both Dirichlet and mixed BVPs [13, 22].

Due to the CVBEM's ability to model a variety of problems of interest in science and engineering, as well as the CVBEM's compatibility with the use of computational nodes, it is a useful PDE numerical method for demonstrating the proposed NPA. The remainder of this section illuminates the relationship between the CVBEM basis functions and the computational nodes that are the focus of the NPA.

### 3.1 The CVBEM approximation function and basis functions

Let  $\Omega \subseteq \mathbb{C}$  denote a simply connected domain and  $\omega : \Omega \rightarrow \mathbb{C}$  be an analytic function of the complex variable  $z = x + iy$ , where  $i = \sqrt{-1}$  and  $x, y \in \mathbb{R}$ . The complex variable function  $\omega$  consists of two real variable functions, denoted  $\phi : \mathbb{R} \rightarrow \mathbb{R}^2$  and  $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ , respectively, such that  $\omega(z) = \phi(x, y) + i\psi(x, y)$ . Since  $\omega$  is analytic in  $\Omega$ ,  $\phi$  and  $\psi$  are related by the Cauchy–Riemann equations:

$$\frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} \quad \text{and} \quad \frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x}, \quad (x + iy) \in \Omega. \quad (2)$$

A consequence of eqn (2) is that both  $\phi$  and  $\psi$  are harmonic functions within  $\Omega$ . The key insight of the CVBEM approach is to create an approximation function, denoted  $\hat{\omega}$ , consisting of a linear combination of complex variable functions that are each analytic within  $\Omega$ , so that both the real and imaginary parts of  $\hat{\omega}$  satisfy Laplace's equation within  $\Omega$ :

$$\hat{\omega}(z) = \sum_{j=1}^n c_j g_j(z), \quad z \in \Omega. \quad (3)$$

Equation (3) is known as the CVBEM approximation function. The  $g_j(z) : \Omega \rightarrow \mathbb{C}$  are analytic within  $\Omega$  and are selected by the modeler (note: for brevity, we may refer to the set  $g_j(z)$ ,  $j = 1, \dots, n$ , as 'analytic functions' without the additional specification that the functions only need to be analytic within  $\Omega$ ). The coefficients  $c_j \rightarrow \mathbb{C}$  are complex numbers, each composed of two real constants,  $\alpha_j = \Re(c_j)$  and  $\beta_j = \Im(c_j)$ . Since each  $c_j$  corresponds to two degrees of freedom (d.o.f), there are a total of  $2n$  d.o.f to be determined in a CVBEM model. The particular values of these d.o.f are dependent on the geometry of the problem, the boundary conditions, the selected  $g_j(z)$ , and the locations of the computational nodes as determined by the NPA. Of note, in this paper, d.o.f is used exclusively to refer to the  $2n$  unknown values of the coefficients in eqn (3) and not to the locations of the candidate nodes or candidate collocation points for the NPA, or other potential d.o.f, such as the particular discretization of  $\mathcal{N}$ , among others.

Since the real and imaginary parts of  $\hat{\omega}$  are harmonic functions within  $\Omega$ , the CVBEM approximation function satisfies Laplace's equation within the problem domain. Hence, the only error in eqn (3) is due to error in continuously satisfying the boundary conditions. Therefore, the  $2n$  d.o.f are chosen so that the CVBEM approximation function satisfies the given boundary conditions as closely as possible within a given norm.

While various families of basis functions have been examined for use with the CVBEM approximation function, as in [23], only certain types of basis functions require the use of computational nodes. In order to demonstrate the NPA, we will use analytic functions obtained by direct numerical integration of the Cauchy integral equation, since these basis functions incorporate the use of computational nodes. The well-known Cauchy integral equation is

$$\omega(z) = \frac{1}{2\pi i} \oint_{\partial\Omega} \frac{\omega(\zeta) d\zeta}{\zeta - z}, \quad z \in \Omega, \tag{4}$$

and as demonstrated in [12], when straight line segments are used to discretize the boundary of the problem domain, the numerical integration of eqn (4) results in the following sum:

$$\hat{\omega}(z) = \sum_{j=1}^n c_j (z - z_j) \ln(z - z_j), \quad z_j \in \mathcal{N}, z \in \Omega. \tag{5}$$

The functions  $(z - z_j) \ln(z - z_j)$  are referred to as the basis functions of the CVBEM approximation function, and the points  $z_j$  are the branch points of these basis functions. The branch points are interpreted as the computational nodes whose locations will be determined by the NPA. These nodes are referred to as ‘computational nodes’ because they do not have a physical meaning within the problem context, and they only exist due to the use of these particular basis functions.

Lastly, we note that the locations of the computational nodes can be selected arbitrarily within  $\mathcal{N}$  by the NPA, provided that each node is in a distinct location, which is necessary for the basis functions to be linearly independent. The specific locations of the computational nodes that are determined by the NPA do not change the physical interpretation of the approximation function. Rather, the locations of the selected nodes primarily affect the degree to which the CVBEM approximation function can satisfy the given boundary conditions by collocation, least squares, or another boundary condition fitting technique.

### 3.2 Treatment of the basis function branch cuts

Each basis function used in this work is of the form  $(z - z_j) \ln(z - z_j)$ , where  $z_j \in \mathcal{N}$ . These functions have a branch cut emanating from  $z_j$ , along which the CVBEM approximation function is discontinuous. Therefore, in order to obtain a CVBEM approximation function that is continuous within the problem domain, it is necessary to rotate the branch cut associated with each basis function, such that it does not intersect the problem domain, as illustrated in Fig. 14.

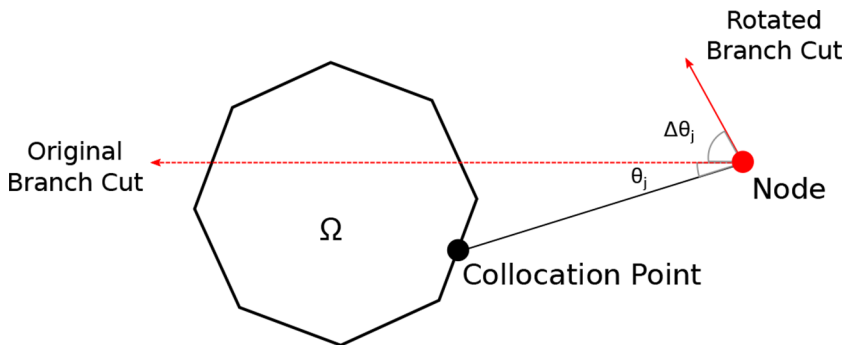


Figure 14: Illustration of the rotation of a typical branch cut. Some angle  $\Delta\theta_j$  is added to  $\theta_j$ , so that the branch cut corresponding to  $(z - z_j) \ln(z - z_j)$  does not intersect the problem domain or other branch cuts. The rotation angle,  $\Delta\theta_j$ , is related to  $\alpha$  by  $\Delta\theta_j = \pi - \alpha$ .

Rotating the branch cuts is most conveniently accomplished in polar coordinates. Using the usual conversion to polar coordinates, we have  $z - z_j = R_j e^{i\theta_j}$ , with  $\theta_j = \arg(z - z_j)$  often defined in the interval  $(-\pi, \pi]$  and  $R_j = |z - z_j|$ . Thus, the basis functions are represented in polar coordinates as follows:

$$\begin{aligned}
 (z - z_j) \ln(z - z_j) &= R_j e^{i\theta_j} \ln(R_j e^{i\theta_j}) && \theta_j \in (-\pi + \pi) \\
 &= R_j \left[ \cos(\theta_j) + i \sin(\theta_j) \right] \left[ \ln(R_j) + \ln(e^{i\theta_j}) \right], && \theta_j \in (-\pi + \pi) \\
 &= R_j \left[ \cos(\theta_j) + i \sin(\theta_j) \right] \left[ \ln(R_j) + i\theta_j \right], && \theta_j \in (-\pi + \pi) \\
 &= R_j \left[ \ln(R_j) \cos(\theta_j) - \theta_j \sin(\theta_j) \right], \\
 &\quad + iR_j \left[ \theta_j \cos(\theta_j) + \ln(R_j) \sin(\theta_j) \right], && \theta_j \in (-\pi + \pi)
 \end{aligned} \tag{6}$$

Rotating the branch cut is accomplished by redefining the interval of the angle to be  $\theta_{j,a} \in (\alpha, \alpha + 2\pi]$ , where  $\alpha$  is the desired angle of the branch cut. Thus, we are interested in mapping the interval  $(-\pi, \pi]$  to  $(\alpha, \alpha + 2\pi]$ . This is accomplished by the following:

$$\theta_{j,a} = \text{mod}(\theta_j - \alpha, 2\pi) + \alpha \tag{7}$$

The branch cuts are rotated counter-clockwise by the amount  $\Delta\theta_j$ , where  $\Delta\theta_j$  satisfies  $\alpha = \pi - \Delta\theta_j$ . The rotation angle  $\Delta\theta_j$  can be selected by the modeler as long as the rotated branch cut does not intersect the problem domain. Let  $\theta_{j,a}$  be defined as in eqn (7). Then, the basis functions with rotated branch cuts are given by:

$$\begin{aligned}
 (z - z_j) \ln(z - z_j) &= R_j \left[ \ln(R_j) \cos(\theta_{j,a}) - \theta_{j,a} \sin(\theta_{j,a}) \right] \\
 &\quad + iR_j \left[ \theta_{j,a} \cos(\theta_{j,a}) + \ln(R_j) \sin(\theta_{j,a}) \right], && \theta_{j,a} \in (\alpha, \alpha + 2\pi).
 \end{aligned} \tag{8}$$

### 3.3 Numerical implementation of the CVBEM

The ability to directly integrate eqn (4), which results in an approximation of the Cauchy integral equation as a finite sum, means that there is no need to do numerical quadrature in the computer implementation of the CVBEM procedure. Instead, after the NPA has been used to determine the locations of the collocation points and computational nodes that will be used in the CVBEM model, the only computational task is to compute the coefficients of the CVBEM approximation function. Determining these coefficients is done by solving a system of linear equations as will be detailed in this section.

These coefficients are complex, and hence have both a real and an imaginary part:  $c_j = \alpha_j + i\beta_j$ . Likewise, the basis functions in eqn (3) are complex variable functions, and hence also have both a real and an imaginary part:  $g_j(z) = \lambda_j(x, y) + i\mu_j(x, y)$ . Thus, from eqn (3), it follows:

$$\begin{aligned}
 \hat{\omega}(z) &= \sum_{j=1}^n c_j g_j(z) \\
 &= \sum_{j=1}^n (\alpha_j + i\beta_j)_j (\lambda_j(x, y) + i\mu_j(x, y)) \\
 &= \sum_{j=1}^n \left[ \alpha_j \lambda_j(x, y) - \beta_j \mu_j(x, y) + i \left[ \alpha_j \mu_j(x, y) + \beta_j \lambda_j(x, y) \right] \right].
 \end{aligned} \tag{9}$$

The real and imaginary parts of eqn (9) are, respectively,

$$\Re [\hat{\omega}(z)] = \hat{\phi}(x, y) = \sum_{j=1}^n a_j \lambda_j(x, y) - \beta_j \mu_j(x, y),$$

$$\Im [\hat{\omega}(z)] = \hat{\psi}(x, y) = \sum_{j=1}^n a_j \mu_j(x, y) + \beta_j \lambda_j(x, y).$$
(10)

In many applications,  $\hat{\phi}$  and  $\hat{\psi}$  are interpreted physically as the approximate potential and stream functions, respectively. Since  $\hat{\omega}$  is analytic within  $\Omega$ , we have

$$\Delta \hat{\phi}(x, y) = 0 \quad \text{and} \quad \Delta \hat{\psi}(x, y) = 0, \quad (x + iy) \in \Omega.$$
(11)

Notice in Eqn (10) that  $\hat{\phi}$  and  $\hat{\psi}$  are defined using the same coefficients and basis functions. Thus, as soon as one of either the approximate stream or potential functions is known, it is immediately possible to determine the other function. This important observation is a consequence of the Cauchy–Riemann equations, which can be used to determine the conjugate harmonic function up to an additive constant of any harmonic function. Thus, for example, the CVBEM can be used to obtain the approximate stream function using boundary data exclusively from the conjugate potential function, which is one of the useful properties of the CVBEM.

Collocation leads to a system of  $2n$  equations in  $2n$  unknowns. For Dirichlet BVPs, the unknown coefficients are determined by collocation of either the real or imaginary part of the CVBEM approximation function (depending on the nature of the given boundary conditions) with the given boundary data. In the example Dirichlet BVP that follows in Section 4.1, we assume boundary conditions are specified from the stream function,  $\psi$ . Consequently, we perform collocation with the imaginary part of the CVBEM approximation function. The set of  $2n$  equations is given as follows:

$$\hat{\psi}(x_i, y_i) = \sum_{j=1}^n a_j \mu_j(x_i, y_i) + \beta_j \lambda_j(x_i, y_i) = \psi(x_i, y_i), \quad i = 1, \dots, 2n$$

$$(x_i + y_i) \in \partial\Omega.$$
(12)

In eqn (12),  $\psi(x_p, y_p)$  denotes the boundary condition specified at  $(x_p, y_p)$ . By using collocation to determine the coefficients of the CVBEM approximation function, we have set  $\hat{\psi}(x_p, y_p) = \psi(x_p, y_p)$  at each of the  $2n$  collocation points. This guarantees that the CVBEM approximation function will be exact at the locations of the  $2n$  collocation points in the absence of truncation and round-off errors.

Lastly, it should be noted that the system of equations defined by eqn (12) results in a dense, non-symmetric matrix equation, which is more computationally difficult to solve than the sparse, symmetric matrix equations that are a usual feature of FEM models.

### 3.4 Error estimation

A convenience of modeling with the CVBEM is that complex variable theory provides a simple technique for determining the maximum error of the CVBEM approximation function in the  $l_\infty$  norm. Let  $\phi \in \mathbb{R}^2$  denote a harmonic function that is the analytic solution of the BVP of interest, and let  $\hat{\phi} = \Re[\omega(z)] \in \mathbb{R}^2$  denote the CVBEM approximation of  $\phi$ . Error estimation is based on the following two key observations:

1. The analytic solution,  $\phi$ , satisfies Laplace’s equation and is, therefore, harmonic in  $\Omega$ .

2. The CVBEM approximation function, denoted  $\hat{\omega}$ , is an analytic complex variable function. Consequently,  $\hat{\phi} = \Re[\hat{\omega}]$ , which is the CVBEM approximation of the target function  $\phi$ , is also harmonic in  $\Omega$ .

Now, we define the error function,  $\varepsilon(x, y)$ , as follows:

$$\varepsilon(x, y) = \phi(x, y) - \hat{\phi}(x, y) \quad (13)$$

Since the error function given in eqn (13) is the difference between two harmonic functions,  $\varepsilon(x, y)$  is itself harmonic in  $\Omega$ . Consequently, by the maximum modulus principle for harmonic functions,  $|\varepsilon(x, y)|$  attains its maximum on  $\partial\Omega$ . Hence, the maximum error of the CVBEM approximation function occurs on  $\partial\Omega$ :

$$\max_{(x,y) \in \partial\Omega} |\varepsilon(x, y)| \leq \max_{(x,y) \in \partial\Omega} |\varepsilon(x, y)|. \quad (14)$$

A reasonable approximation of  $\max_{(x,y) \in \partial\Omega} |\varepsilon(x, y)|$  can be obtained by computing the value of  $|\varepsilon(x, y)|$  at many locations along the problem boundary. As  $|\varepsilon(x, y)|$  is computed at more locations along the boundary, the estimation of  $\max_{\partial\Omega} |\varepsilon(x, y)|$  will improve, provided that the  $\partial\Omega$  error evaluation points are reasonably spaced.

### 3.5 Obtaining the flow nets

Since  $\hat{\omega}$  is a linear combination of analytic complex variable functions, both the real and imaginary parts of  $\hat{\omega}$ , which are  $\hat{\phi}$  and  $\hat{\psi}$ , respectively, can be evaluated continuously within  $\Omega$ . Consequently, continuous computational estimates of both the approximate potential function and the approximate stream function are provided within  $\Omega$  without the need for any interpolation or other post-processing of either  $\hat{\phi}$  or  $\hat{\psi}$ , which is a key benefit of the CVBEM that is not a typical feature of FEM models.

An important property of the CVBEM is that once  $\hat{\phi}$  has been computed, the corresponding conjugate function  $\hat{\psi}$  can be determined uniquely up to an additive constant as a consequence of the Cauchy–Riemann equations. Since  $\hat{\phi}$  and  $\hat{\psi}$  are related in this way, they are referred to as conjugate harmonic functions. As conjugate harmonic functions, their contour lines (level curves), which are the potential lines and streamlines, respectively, are orthogonal at their points of intersection throughout the domain. When these sets of contour lines are superimposed, the resulting plot is the standard flow net graphical display, which is a commonly used tool for visualizing potential flow in engineering as well as the applied sciences and mathematics [22, 24].

The flow nets produced in Section 4 were developed using the computer program MATLAB. The flow nets were created using the internal contour plot function of MATLAB. Specifically, we created a two-dimensional grid of points in the area where we wanted to produce a flow net. Then, we evaluated the real and imaginary parts of the CVBEM approximation function at each of the points in the grid. Then, using the  $x$  and  $y$  coordinates of the grid points as well as the values of the real and imaginary parts of the CVBEM approximation function at these points, we used the contour plot function to create the corresponding flow net.

We conclude this section with a remark about developing flow nets for BVPs of the Laplace equation in three or higher spatial dimensions. As previously mentioned, the flow nets that are developed with the CVBEM for two-dimensional problems are possible as a result of the Cauchy–Riemann equations, which state that the real and imaginary parts of an analytic



complex variable function are harmonic conjugates. However, the Cauchy–Riemann equations only apply to analytic complex variable functions in two spatial dimensions, and there is not a natural analog for analytic functions in higher spatial dimensions. Therefore, for problems in three or higher spatial dimensions, it is necessary to develop the conjugate harmonic function using the usual vector gradient techniques as a post-processing step, such as would be done when using a FEM model.

#### 4 DEMONSTRATION OF THE NEW NPA

In this section, we demonstrate the coupled CVBEM/NPA1 and CVBEM/NPA2 methodologies by modeling two Dirichlet BVPs. The first problem concerns potential flow over a half-cylinder, and the second problem concerns potential flow in a corner and over a half-cylinder. These problems are fundamental and are often used as components in the development of very sophisticated potential flow models. Moreover, these problems were selected because of the extreme curvature observed in areas of their analytic solutions, such as near the stagnation points. The curvature makes modeling these areas computationally difficult. Consequently, high-precision computational modeling is required in order to achieve satisfactory approximations of these potential flow problems.

In the following problems, we make the standard assumptions of an ideal and incompressible fluid with no vorticity. Under these conditions, the velocity potential describing the flow situation is a scalar function satisfying Laplace’s equation. Thus, the problems are well suited for modeling with the CVBEM. The example problems demonstrate the ability of NPA2 to reduce the CVBEM model error beyond the level achieved by the NPA1 result of equivalent model size.

Moreover, the availability of the analytic solutions for these example flow situations, as given in eqns (15) and (16), allows for a precise description of the computational error of the CVBEM model by direct comparison with the analytic solution. Thus, we can quantitatively assess the ability of both NPAs to determine suitable locations for the computational nodes of the CVBEM model by measuring the maximum errors of each outcome based on the given analytic solution.

##### 4.1 Example problem 1 – potential flow over a half-cylinder

The analytic representation of the velocity potential for this problem is given in [24] as

$$\omega(z) = z + 1/z, \quad \Im[z] \geq 0, \quad z \neq 0. \quad (15)$$

A formal description of the test problem follows in Table 1. Since the exact solution is analytic everywhere except at  $z = 0$ , the real and imaginary parts of  $\omega$  are harmonic functions in  $\mathbb{C} \setminus \{0\}$  and thus harmonic throughout  $\Omega \subset \mathbb{C} \setminus \{0\}$ .

Figures 17–21 depict various views of the CVBEM flow net approximation. In these figures, we have given emphasis to the north pole of the half-cylinder, as well as to the stagnation points because these are the regions of the flow situation in which the curvature of the analytic solution is most extreme.

As reported in Table 2, for  $n \geq 10$ , the new NPA improved the accuracy of the CVBEM models by at least an order of magnitude – and up to six orders of magnitude in one instance. The key ingredient to this better performance is the refinement procedure, which allows the locations of previously located nodes to change as the model develops. Thus, we conclude that the NPA2 consistently produces a more accurate CVBEM model than is produced using NPA1 when modeling this benchmark problem of potential flow over a half-cylinder.

Table 1: Example problem 1 – problem description.

Problem domain	$\Omega = f\{(x, y) : -3 \leq x \leq 3, 0 \leq y \leq 3, \text{ and } x^2 + y^2 \geq 1\}$
Governing PDE	$\nabla^2 \psi = 0$
Boundary conditions	$\psi(x, y) = \mathfrak{C}[z + \frac{1}{z}], (x, y) \in \partial\Omega$
Number of candidate computational nodes	1,000
Number of candidate collocation points	500

Table 2: Example problem 1 – maximum error and time elapsed for various CVBEM models.

Number of basis functions	Number of d.o.f.	Unrefined method (NPA1)		Refined method (NPA2)	
		Maximum error	Time elapsed (s)	Maximum error	Time elapsed (s)
10	20	2.376217e-02	2.600493	6.731285e-03	26.847856
20	40	1.324917e-03	5.413931	1.639780e-05	101.625993
30	60	2.123033e-05	10.021206	3.783824e-09	199.087752
40	80	3.277548e-07	11.846832	1.816325e-13	408.392388
50	100	6.777379e-11	16.865822	1.163514e-13	672.789040

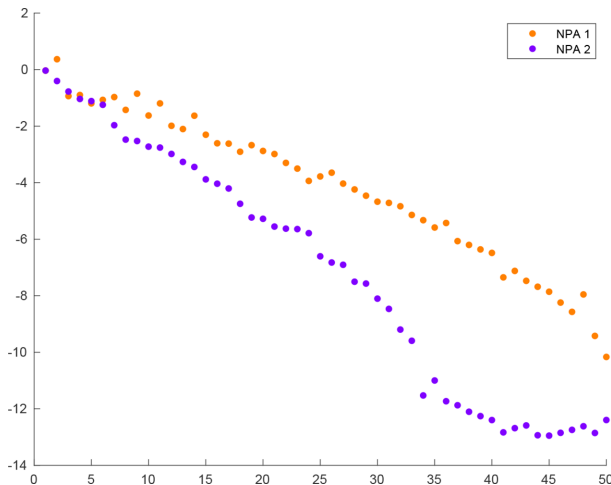


Figure 15: Maximum absolute error of CVBEM models resulting from the use of NPAs 1 and 2 for approximations using  $n = 1, \dots, 50$  terms in the approximation function. The horizontal axis denotes the number of terms used in each CVBEM model. The vertical axis measures the logarithm (base 10) of the maximum error for each CVBEM model.

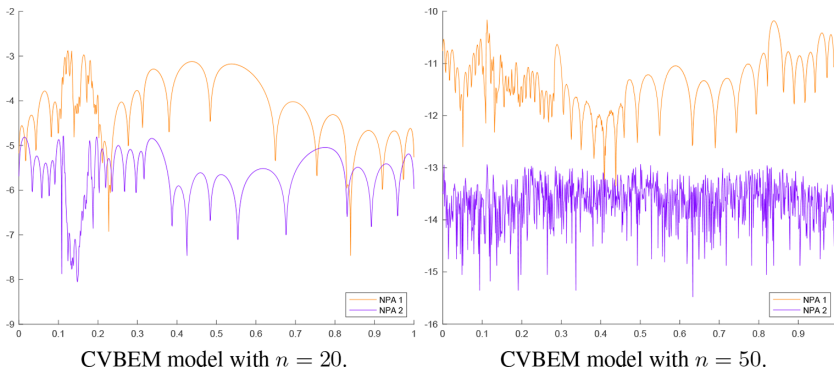


Figure 16: Comparison of approximation error measured on the boundary for CVBEM models resulting from the use of NPAs 1 and 2. The error on the boundary is mapped to the line segment  $[0, 1]$  on the horizontal axis. The vertical axis measures the logarithm (base 10) of the error for each CVBEM model.

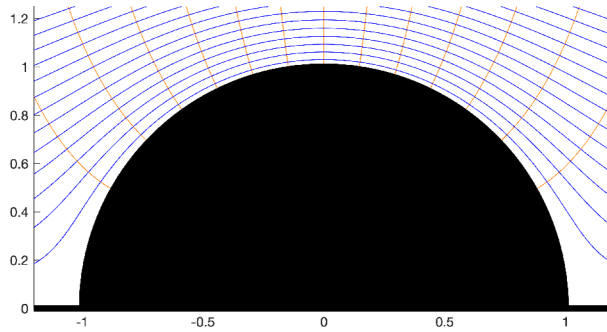


Figure 17: Magnified view of the CVBEM flow net approximation for the flow regime depicted near the half-cylinder obstacle. The curvature of the solution increases near the north pole of the half-cylinder and the stagnation points. Potential lines are shown in orange and streamlines are shown in blue.

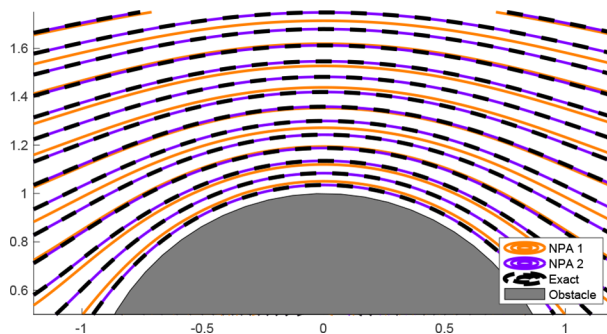


Figure 18: Comparison of the streamlines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . These are the streamlines near the north pole of the obstacle at  $(0, 1)$ .

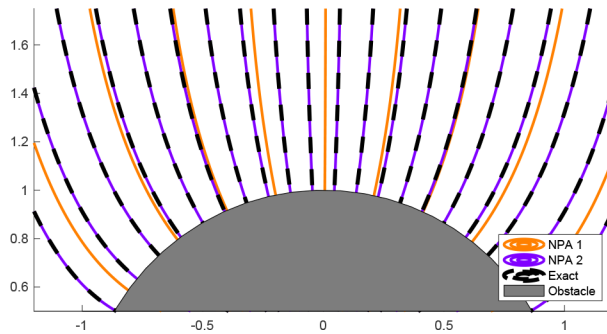


Figure 19: Comparison of the potential lines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . These are the potential lines near the north pole of the obstacle at (0, 1).

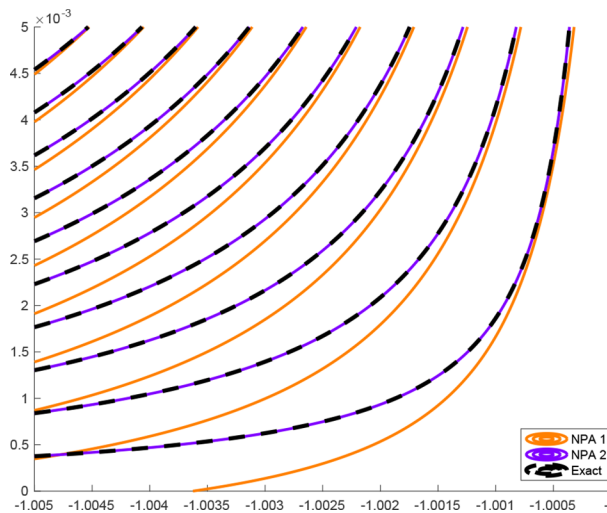


Figure 20: Comparison of the streamlines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . Notice that as the curvature of the solution becomes more extreme, NPA2 results in a CVBEM model that more accurately tracks the curvature of the analytic solution than NPA1. These are the streamlines near the stagnation point at (1, 0). The flow near the stagnation points approaches potential flow in a 90° bend, which is a challenging flow situation to model accurately.

#### 4.2 Example problem 2 – potential flow in a corner and over a half-cylinder

We consider another potential flow problem with Dirichlet boundary conditions. This problem models potential flow that turns in a corner and subsequently goes over a half-cylinder obstacle. The analytic representation of the velocity potential for this problem is given by

$$\omega(z) = z^2 + z + \frac{10}{z-5}, \quad z \neq 5, z \in \Omega. \tag{16}$$

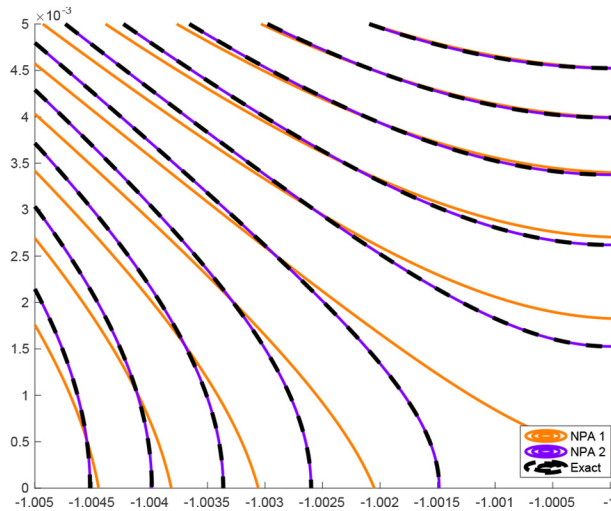


Figure 21: Comparison of the potential lines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . These are the potential lines near the stagnation point at  $(1, 0)$ . The flow near the stagnation points approaches potential flow in a  $90^\circ$  bend, which is a challenging flow situation to model accurately.

Table 3: Example problem 2 – problem description.

Problem domain	$\Omega = \{(x, y) : -0.325 \leq x \leq 8, 0 \leq y \leq 5$ and $(x - 4.9125)^2 + y^2 \geq 0.975^2\}$
Governing PDE	$\nabla^2 \phi = 0$
Boundary conditions	$\phi(x, y) = \Re \left[ z^2 + z + \frac{10}{z - 5} \right]$
Number of candidate computational nodes	250
Number of candidate collocation points	1,000

Figures 24–28 depict various views of the CVBEM flow net approximation. In these figures, we have given emphasis to the north pole of the half-cylinder, as well as to the corner in the bottom left of the problem domain because the curvature of the flow situation is most extreme in these areas.

As reported in Table 4, for  $n \geq 10$ , the new NPA improved the accuracy of the CVBEM models by at least an order of magnitude – and up to three orders of magnitude in two instances. The key ingredient to this better performance is the refinement procedure, which allows the locations of previously located nodes to change as the model develops. Thus, we conclude that the NPA2 consistently produces a more accurate CVBEM model than is produced using NPA1 when modeling this benchmark problem of potential flow in a corner and over a half-cylinder.

Table 4: Example problem 2 – maximum error and time elapsed for various CVBEM models.

Number of basis functions	Number of d.o.f.	Unrefined method (NPA1)		Refined method (NPA2)	
		Maximum error	Time elapsed (s)	Maximum error	Time elapsed (s)
10	20	1.757095e+00	1.178742	1.015156e-01	18.173322
20	40	2.165579e-02	2.326950	4.914030e-04	69.857409
30	60	2.325910e-04	3.681986	2.317752e-07	156.962888
40	80	2.979909e-06	5.477926	1.598643e-09	346.051462
50	100	1.362534e-08	6.828074	1.928342e-10	546.901546

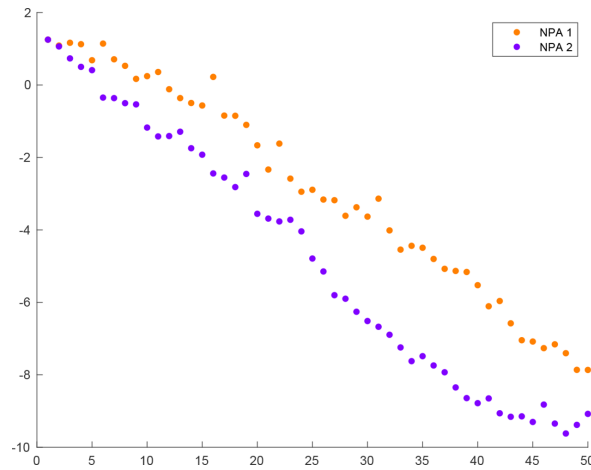


Figure 22: Maximum absolute error of CVBEM models resulting from the use of NPAs 1 and 2 for approximations using  $n = 1, \dots, 50$  terms in the approximation function. The horizontal axis denotes the number of terms used in each CVBEM model. The vertical axis measures the logarithm (base 10) of the maximum error for each CVBEM model.

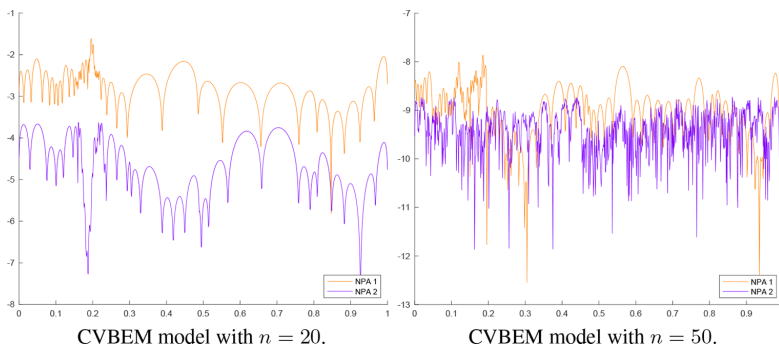


Figure 23: Comparison of approximation error measured on the boundary for CVBEM models resulting from the use of NPAs 1 and 2. The error on the boundary is mapped to the line segment  $[0, 1]$  on the horizontal axis. The vertical axis measures the logarithm (base 10) of the error for each CVBEM model.

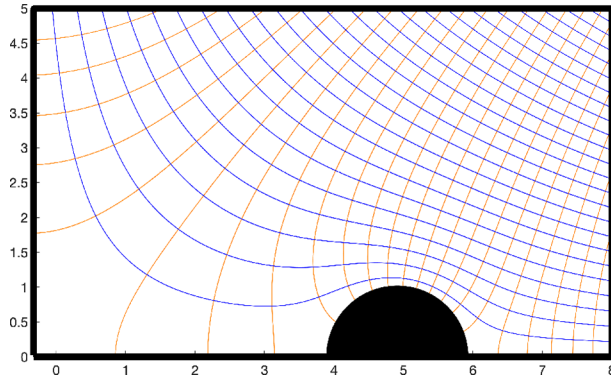


Figure 24: CVBEM approximation of the flow regime in the area of interest for example problem 2. The curvature of the flow situation is greatest near the origin and near the half-cylinder obstacle, which makes these areas difficult to model computationally. Potential lines are shown in orange and streamlines are shown in blue.

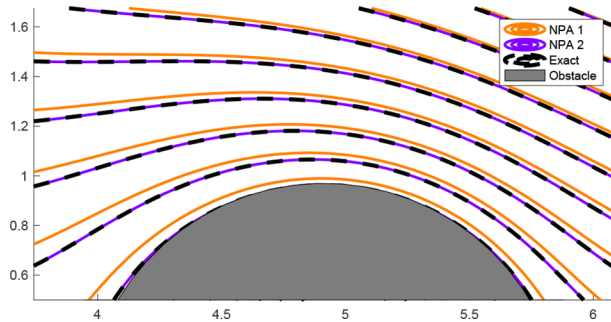


Figure 25: Comparison of the streamlines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . These are the streamlines near the north pole of the obstacle.

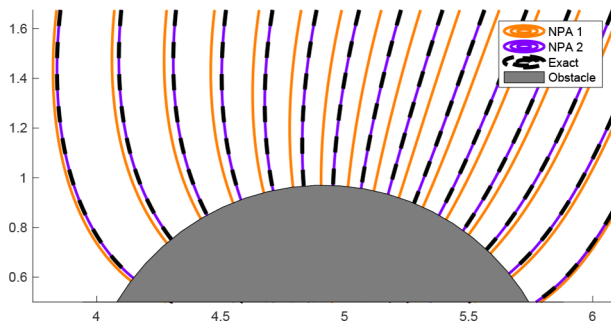


Figure 26: Comparison of the potential lines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . These are the potential lines near the north pole of the obstacle.

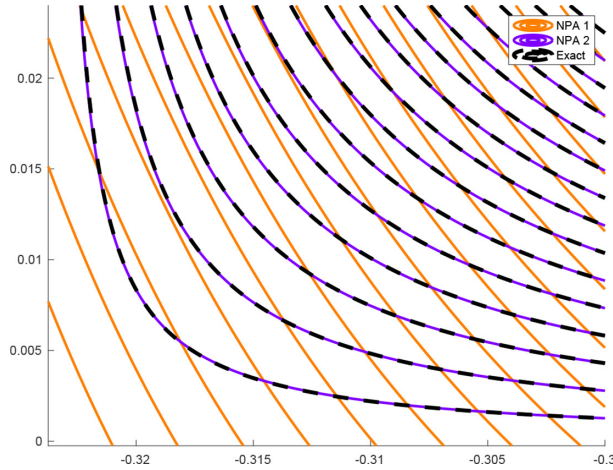


Figure 27: Comparison of the streamlines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . These are the streamlines near the lower left-hand corner of the problem domain. Notice that in this case, the streamlines resulting from the use of NPA2 much more accurately track the curvature of the analytic solution compared to the results from NPA1.

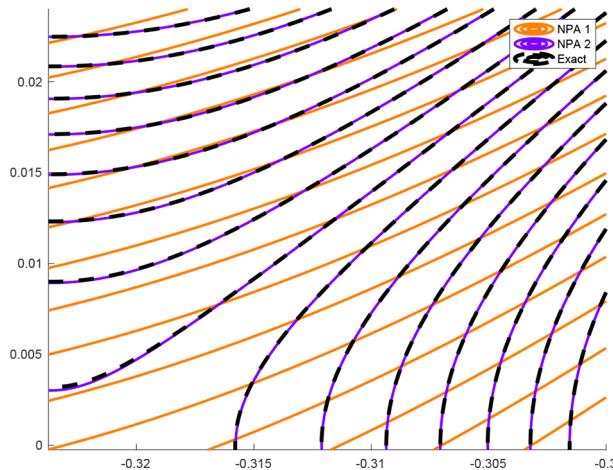


Figure 28: Comparison of the potential lines resulting from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with  $n = 20$ . These are the potential lines near the lower left-hand corner of the problem domain. Notice that in this case, the potential lines resulting from the use of NPA2 much more accurately track the curvature of the analytic solution compared to the results from NPA1.

## 5 DISCUSSION AND CONCLUSION

This paper proposes a new NPA, which we refer to as NPA2, for determining suitable locations of the computational nodes that are a common feature of mesh reduction methods for solving PDEs. The novelty of NPA2 is due to the refinement procedure that is described in Section 2. The key contribution of the refinement procedure is that it facilitates the exchange



of previously selected computational nodes when making such a change would result in a BVP model with less error. As a result, the refinement procedure has a monotonic non-increasing effect on the maximum error of the BVP model to which it is applied. The new NPA is compared to the algorithm presented in [1], which we refer to as NPA1.

In [1], the authors showed that the use of NPA1 could improve the accuracy of a coupled CVBEM/NPA1 model compared to a CVBEM model using an arbitrary node allocation scheme. The benefit of NPA1 was that nodes could now be used more efficiently to reduce the error of a CVBEM model, thereby rendering it unnecessary to use as many nodes or terms in a CVBEM approximation function to achieve a desired level of maximum error. With NPA2, we have continued this effort and shown that it is possible to obtain even more efficient uses of the computational nodes. Consequently, using NPA2, it is possible to obtain a desired level of maximum error using even fewer nodes than would be required when using NPA1.

In this work we coupled NPAs 1 and 2 with the CVBEM in order to model two Dirichlet BVPs of the Laplace type. The first example problem considers potential flow over a half-cylinder, and the second example problem considers potential flow in a corner followed by a half-cylinder. These demonstration problems involve the computational difficulty of modeling flow regimes with regions of extreme curvature near the stagnation points. The computational difficulty associated with these problems may qualify them as good benchmark problems for future evaluations of NPAs.

The maximum errors obtained using both NPAs 1 and 2 are reported for CVBEM models of various sizes. In general, we conclude that the use of NPA2 results in a CVBEM model that is more accurate than the model produced using NPA1.

Moreover, from the figures, it is observed that the streamline outcomes from NPA2 are a significantly closer plotting approximation to the streamline outcomes from the analytic solution. This result can be of immense importance when dealing with problems such as, for example, tracking the movement of groundwater contamination. The ability or inability to track such contamination can trigger environmental, health, and legal repercussions, which, therefore, makes the introduction of this improved NPA an important advancement in the modeling capabilities of such problems.

Finally, we emphasize that the new NPA is applicable to the large class of PDE numerical methods in which the accuracy of the BVP model depends on the locations of computational nodes or source points. For demonstration purposes, we chose to couple the NPA with the CVBEM, but the possible uses of the NPA are more general and include coupling with other mesh reduction numerical methods. Therefore, the successful coupling of NPA2 with the CVBEM that was demonstrated in this paper is only the first of potentially many more successful applications of the NPA that can be developed with other mesh reduction methods for PDEs and suggests the possibility for increased high-precision computational modeling with these methods.

#### ACKNOWLEDGEMENT

The authors are grateful to Hromadka & Associates, a consulting firm, for supporting this research.

#### REFERENCES

- [1] Demoes, N.J., Bann, G.T., Wilkins, B.D., Grubaugh, K.E. & Hromadka II, T.V., Optimization algorithm for locating computational nodal points in the method of fundamental solutions to improve computational accuracy in geosciences modeling. *The Professional Geologist*, 2019.

- [2] Bathe, K.J., *Finite Element Procedures*. Prentice Hall, Pearson Education, Inc., 1996.
- [3] Young, D., Chen, K., Chen, J. & Kao, J., A modified method of fundamental solutions with source on the boundary for solving laplace equations with circular and arbitrary domains. *Computer Modeling in Engineering and Sciences*, **19(3)**, pp. 197–221, 2007. <https://doi.org/10.1016/j.enganabound.2018.11.008>
- [4] Fornberg, B. & Flyer, N., Fast generation of 2-D node distributions for mesh-free PDE discretizations. *Computers and Mathematics with Applications*, **69**, pp. 531–544, 2015. <https://doi.org/10.1016/j.camwa.2015.01.009>
- [5] Liu, Y., Nie, Y., Zhang, W. & Wang, L., Node placement method by bubble simulation and its application. *Computer Modeling in Engineering and Sciences*, **55(1)**, pp. 89–109, 2010.
- [6] Shankar, V., Kirby, R.M. & Fogelson, A.L., Robust Node Generation for Mesh-free discretizations on irregular domains and surfaces. *SIAM Journal on Scientific Computing*, **40(4)**, pp. 2584–2608, 2018. <https://doi.org/10.1137/17m114090x>
- [7] Slak, J. & Kosec, G., Fast generation of variable density node distributions for mesh-free methods. *Boundary Elements and other Mesh Reductions Methods XLI*. WIT Press: Southampton and Boston, pp. 163–173, 2019.
- [8] Chen, C., Karageorghis, A. & Li, Y., On choosing the location of the sources in the MFS. *Numerical Algorithms (Springer)*, **72(1)**, pp. 107–130, 2016. <https://doi.org/10.1007/s11075-015-0036-0>
- [9] Hromadka II, T. & Guymon, G., Application of a boundary integral equation to prediction of freezing fronts in soil. *Cold Regions Science and Technology*, **6**, pp. 115–121, 1982. [https://doi.org/10.1016/0165-232x\(82\)90004-0](https://doi.org/10.1016/0165-232x(82)90004-0)
- [10] Wilkins, B.D., Greenberg, J., Redmond, B., Baily, A., Flowerday, N., Kratch, A., Hromadka II, T.V., Boucher, R., McInvale, H.D. & Horton, S., An unsteady two dimensional complex variable boundary element method. *SCIRP Applied Mathematics*, **8(6)**, pp. 878–891, 2017. <https://doi.org/10.4236/am.2017.86069>
- [11] Wilkins, B.D., Hromadka II, T.V. & Boucher, R., A conceptual numerical model of the wave equation using the complex variable boundary element method. *Applied Mathematics*, **8(5)**, p. 724, 2017. <https://doi.org/10.4236/am.2017.85057>
- [12] Hromadka II, T.V. & Guymon, G.L., A complex variable boundary element method: Development. *International Journal for Numerical Methods Engineering*, **20**, pp. 25–37, 1984. <https://doi.org/10.1002/nme.1620200104>
- [13] Johnson, A.N. & Hromadka II, T.V., Modeling mixed boundary conditions in a Hilbert space with the complex variable boundary element method (CVBEM). *MethodsX*, **2**, pp. 292–305, 2015. <https://doi.org/10.1016/j.mex.2015.05.005>
- [14] Hromadka II, T. & Pardoen, G.C., Application of the CVBEM to non-uniform St. Venant torsion. *Computer Methods in Applied Mechanics and Engineering*, **53(2)**, pp. 149–161, 1985. [https://doi.org/10.1016/0045-7825\(85\)90003-9](https://doi.org/10.1016/0045-7825(85)90003-9)
- [15] Hromadka II, T.V. & Lai, C., *The Complex Variable Boundary Element Method*. Springer-Verlag, New York, 1987.
- [16] Hromadka II, T. & Whitley, R., A new formulation for developing CVBEM approximation functions. *Engineering Analysis with Boundary Elements*, **18(1)**, pp. 39–41, 1996. [https://doi.org/10.1016/s0955-7997\(96\)00027-6](https://doi.org/10.1016/s0955-7997(96)00027-6)
- [17] Hromadka II, T.V. & Whitley, R.J., *Advances in the Complex Variable Boundary Element Method*. Springer, New York, 1998.

- [18] Hromadka II, T. & Whitley, R., *Foundations of the Complex Variable Boundary Element Method*. Springer, 2014.
- [19] Brebbia, C.A., *The Boundary Element Method for Engineers*. Wiley, 1978.
- [20] Brebbia, C. & Wrobel, L., Boundary element method for fluid flow. *Advances in Water Resources*, **2**, pp. 83–89, 1979. [https://doi.org/10.1016/0309-1708\(79\)90015-0](https://doi.org/10.1016/0309-1708(79)90015-0)
- [21] Hromadka II, T.V., *A Multi-Dimensional Complex Variable Boundary Element Method, volume 40 of Topics in Engineering*. WIT Press: Southampton and Boston, 2002.
- [22] Johnson, A.N., Hromadka II, T.V., Hughes, M.T. & Horton, S.B., Modeling mixed boundary problems with the Complex Variable Boundary Element Method (CVBEM) using matlab and mathematica. *International Journal of Computational Methods and Experimental Measurements*, **3(3)**, pp. 269–278, 2015. <https://doi.org/10.2495/cmeme-v3-n3-269-278>
- [23] Wilkins, B.D., Hromadka II, T.V., Johnson, A.N., Boucher, R., McInvale, H.D. & Horton, S., Assessment of complex variable basis functions in the approximation of ideal fluid flow problems. *International Journal of Computational Methods and Experimental Measurements*, **7(1)**, pp. 45–56, 2019. <https://doi.org/10.2495/cmeme-v7-n1-45-56>
- [24] Kirchhoff, R.H., *Potential Flows Computer Graphic Solutions*. CRC Press, 1985.