# Using the digamma function for basis functions in mesh-free computational methods 

Bryce D. Wilkins ${ }^{\text {a,*, }}$, Theodore V. Hromadka II ${ }^{\text {b }}$<br>${ }^{\text {a }}$ School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, Pennsylvania 15213, United States of America<br>${ }^{\mathrm{b}}$ Department of Mathematical Sciences, United States Military Academy, 646 Swift Road, West Point, New York 10996, United States of America

## A R T I C L E I N F O

## Keywords:

Complex variable boundary element method (CVBEM)
Basis functions
Mesh-free methods
Applied complex variables
Potential flow


#### Abstract

We examine the utility of a new family of basis functions for use with the Complex Variable Boundary Element Method (CVBEM) and other mesh-free numerical methods for solving partial differential equations. The family of polygamma functions have found use in mathematics since as early as 1730 when James Stirling related the digamma function to the factorial function [1]. Now, we propose using the digamma function, as well as new variants of the digamma function, as basis functions for the CVBEM. This paper discusses technical aspects associated with using the digamma function as a CVBEM basis function. Then, we demonstrate the utility of the proposed basis function by applying it to a mixed boundary value problem of the Laplace type.


## 1. Introduction

Beginning with the introductory work of the 1600s, the digamma function-and, more broadly, the polygamma function-has been the subject of examination and utilization in difficult mathematical problems. For example, James Stirling used this type of function to relate to the factorial function in 1730 [1]. However, this function does not appear to have been used widely in applications of computational engineering mathematics. Now, with the advent of modern computational mathematics capabilities and visualization techniques, the ability for the digamma function to solve the Laplace equation enables it to be used in numerical methods for solving potential problems or problems that include the Laplace equation in their formulation.

In the current paper, we use the digamma function as an inspiration to formulate novel sets of basis function families for use with the meshfree Complex Variable Boundary Element Method (CVBEM). This new approach is shown to provide significant computational accuracy without a significant increase in the CPU time requirement versus the use of comparable basis functions such as the original CVBEM basis functions introduced in [2] and developed in [3]. Thus, it is demonstrated that the digamma function is useful as another basis function family for immediate use in approximations of boundary value problems of particular interest in computational methods.

While this work focuses directly on the use of this basis function family as it pertains to the CVBEM, the extension to other real-variable and mesh-free computational methods is direct. Furthermore, through the use of particular solutions, this solution approach may be extended
to several other PDE types such as the heat equation and wave equation as demonstrated in [4] and [5], respectively, and similar extensions are also possible through the use of particular solutions.

The novelty of this paper is in the following contributions:

1. Proposal of the digamma function as a suitable basis function for use with mesh-free numerical methods
2. Proposal of two new variants of the digamma function as basis functions
3. Assessment of the digamma basis functions along with their proposed variants by comparison with the more standard sets of basis of functions described in [6]
4. New formulation of the CVBEM for use with mixed boundary value problems

This work is an extension of and update to the work in [6], in which the authors sought to assess various basis functions in regard to their success in modeling a benchmark Dirichlet BVP of the Laplace type. In the present work, novel basis functions are now being assessed, and the demonstration problem is a more general mixed BVP of the Laplace type.

## 2. Literature review

Recent research related to the CVBEM has focused on the development of Node Positioning Algorithms (NPAs). In [7], the authors proposed an algorithm for constructing a model of $n \in \mathbb{Z}^{+}$functions, one function at a time, by evaluating the accuracy of each candidate pair of one node and two collocation points. The candidate pair corresponding

[^0]Table 1
Description of the mixed boundary value problem examined for comparing the efficacy of various basis function families for use in the CVBEM.

| Problem Domain: | $\begin{gathered} \Omega=\{(x, y): 0<x<8,0<y<5, \\ \text { and } \left.(x-5)^{2}+y^{2}>1\right\} \end{gathered}$ |
| :---: | :---: |
| Governing PDE: Boundary Conditions: | $\begin{aligned} & \nabla^{2} \psi=0 \\ & \begin{cases}\frac{\partial \phi}{\partial \eta}=0, & x=0 \\ \frac{\partial \phi}{\partial}=0, & y=0 \\ \frac{\partial \phi}{\partial n}=0, & (x-5)^{2}+y^{2}=1 \\ \phi(x, y)=\mathfrak{R}\left[z^{2}\right]=x^{2}-y^{2}, & \text { otherwise }\end{cases} \end{aligned}$ |
| Number of Candidate Computational Nodes: Number of Candidate Collocation Points: | 2000 2000 |

Table 2
Maximum error for each of the CVBEM models of the mixed BVP defined in Table 1. The basis functions include: the standard CVBEM basis functions $\left(z-z_{j}\right)\left[\ln \left(z-z_{j}\right)\right]_{\alpha_{j}}$; natural logarithm functions $\left[\ln \left(z-z_{j}\right)\right]_{\alpha_{j}}$; and simple poles $\frac{1}{z-z_{j}}$. The best-performing basis function family for each trial is highlighted in blue.

| Number of Basis Functions | Standard <br> Basis <br> Functions | Natural Logarithm | Simple Poles |
| :---: | :---: | :---: | :---: |
| 10 | $1.77922 \mathrm{e}+00$ | 5.077893e-01 | $2.60406 \mathrm{e}+00$ |
| 20 | $3.58833 \mathrm{e}-01$ | $4.976069 \mathrm{e}-02$ | $3.28793 \mathrm{e}-01$ |
| 30 | $1.67062 \mathrm{e}-03$ | $7.303782 \mathrm{e}-04$ | $1.06124 \mathrm{e}+00$ |
| 40 | $2.99314 \mathrm{e}-05$ | $1.170612 \mathrm{e}-03$ | $3.65187 \mathrm{e}-02$ |
| 50 | $1.01034 \mathrm{e}-06$ | $2.866278 \mathrm{e}-05$ | $2.47985 \mathrm{e}-02$ |

Table 3
Maximum error for each of the CVBEM models of the mixed BVP defined in Table 1. The basis functions include each of the three proposed variants of the digamma basis function. The best-performing basis function family for each trial is highlighted in blue.

| Number <br> of Basis <br> Functions | Digamma <br> Variant 0 | Digamma <br> Variant 1 | Digamma <br> Variant 2 |
| :--- | :--- | :--- | :--- |
| 10 | $3.61041 \mathrm{e}+00$ | $5.60733 \mathrm{e}-01$ | $2.20686 \mathrm{e}+00$ |
| 20 | $6.73311 \mathrm{e}-02$ | $\mathbf{9 . 4 3 8 5 2 e - 0 3}$ | $6.02101 \mathrm{e}-02$ |
| 30 | $1.93838 \mathrm{e}-02$ | $\mathbf{2 . 3 6 7 0 6 e - 0 4}$ | $2.31955 \mathrm{e}-03$ |
| 40 | $4.82301 \mathrm{e}-04$ | $\mathbf{3 . 8 3 2 7 4 e - 0 6}$ | $9.77183 \mathrm{e}-05$ |
| 50 | $1.64922 \mathrm{e}-04$ | $\mathbf{2 . 2 9 2 8 7 e - 0 7}$ | $1.73748 \mathrm{e}-06$ |

to the smallest maximum error of the approximation function based on the known boundary conditions was added to the model until $n$ pairs were selected.

The recent paper [8] improved upon the results of [7] by introducing a nodal position refinement procedure. This procedure re-assesses the location of each node in the CVBEM model to determine if a better location can be found given the current arrangement of nodes in the model. The recently-developed NPAs have demonstrated that the orientation and placement of the computational nodes has a significant effect upon the computational accuracy of CVBEM models in solving BVPs of, or related to, the Laplace type.

Our group's latest NPA research has focused on iteratively using either of the algorithms in [7] or [8] to progressively trim away unused areas of the candidate node space. The idea is to first establish a distribution of $N$ candidate nodes, denoted $\mathcal{N}_{1}$. Then, use an NPA to select a subset $n<N$ of the candidate nodes. Once the subset is identified, create a convex shape encompassing all of the NPA-selected nodes while ex-
cluding as many of the unused candidate nodes as possible. This convex shape becomes the new boundary of the candidate node space, which is denoted $\mathscr{N}_{2}$. Then, $\mathcal{N}_{2}$ is discretized with $N$ candidate nodes. The volume of $\mathscr{N}_{2}$ is less than or equal to the volume of $\mathscr{N}_{1}$. As both $\mathscr{N}_{1}$ and $\mathcal{N}_{2}$ are discretized with $N$ candidate nodes, $\mathscr{N}_{2}$ constitutes a denser (or possibly equally dense) discretization of the candidate node space than $\mathcal{N}_{1}$. This trimming process can be repeated as many times as desired. In this fashion, the volume of the candidate node space is reduced by trimming away localities where no or little success is achieved in satisfying the given problem boundary conditions.

Along with the development of NPAs, another avenue of recent research has to do with the selection of basis functions for use in the CVBEM approximation function. The original basis functions were derived from integration of the Cauchy integral equation and were developed in [2,3]. In [9-12], the authors examined the use of complex polynomials as basis functions, which resulted in a CVBEM approximation function based on the principles of Taylor series. In [13], the authors proposed using a finite Laurent series as the basis functions for a CVBEM approximation function. The Laurent series approximation function can be thought of as a CVBEM model with a single node defined as the expansion point of the series. Because the expansion point of the Laurent series is a pole of each of the basis functions, the expansion point was chosen to be located in the exterior of the problem domain in order to keep the approximation function analytic in the problem domain.

Originally, the singularities of the CVBEM basis functions were located on the problem boundary [9]. However, after the Laurent series work in [13], the idea of moving the singularities of the basis functions to the exterior of the problem domain was extended to the standard CVBEM basis functions, as demonstrated in [14]. The practice of locating the singularities of the basis functions in the exterior of the problem domain is utilized in the current iteration of the CVBEM software used in this work.

It is also noteworthy that the simple pole basis functions, as well as the more general class of rational functions, were recently examined in Lloyd Nicholas Trefethen's lecture given for SIAM's 2020 John von Neumann Prize [15]. These functions were specifically examined in regard to computational modeling of BVPs of the Laplace type. In the lecture, Trefethen examines the use of several types of rational functions such as simple poles and reciprocal-log functions as basis functions for approximating solutions to the Laplace equation. In the accompanying article, Trefethen concludes that "it seems that a new era of numerical computation with rational functions and other functions with singularities is arriving" [16].

Another development relevant to the present work has to do with modeling mixed BVPs using the CVBEM. Mixed boundary conditions can be used to model situations in which there is no flux of the potential function across one or more edges of the problem domain, such as considered in the primary demonstration problem of this paper. In [ 14,17$]$, it was noted that a zero-flux boundary condition on a given edge is equivalent to a streamline being located on that edge. These papers considered a problem geometry that naturally satisfied this condition with Dirichlet boundary conditions. Therefore, we will extend this work by explicitly stating how zero-flux conditions can be enforced in more general situations in which they are not naturally satisfied by Dirichlet boundary conditions.

## 3. Using digamma functions to model potential problems

In this section, we examine technical aspects pertaining to the use of digamma functions as basis functions for the CVBEM. A series representation for the digamma function is as follows:
$\psi(z)=-\gamma+\sum_{m=0}^{\infty} \frac{z-1}{(m+1)(m+z)}, \quad z \neq 0,-1,-2, \ldots$,
where $\gamma \approx 0.5772156$ denotes the Euler-Mascheroni constant. Furthermore, the $n^{\text {th }}$-order derivative ( $n>0$ ) of the digamma function is given


Fig. 1. Illustration of the rotation of a typical digamma axis. The angle $\alpha$ represents the clockwise rotation of the digamma axis from the negative real axis. The rotation angle is chosen so that the resulting digamma axis does not intersect the problem domain. This is necessary in order to ensure that the CVBEM approximation function is analytic within the problem domain, which guarantees that the real and imaginary parts of the approximation function will be harmonic in the problem domain.
as follows:
$\psi^{(n)}(z)=(-1)^{n+1} n!\sum_{k=0}^{\infty} \frac{1}{(k+z)^{n+1}}, \quad z \neq 0,-1,-2, \ldots$.
The set of derivatives of the digamma function are collectively called the polygamma functions. From the series definitions given in Eqs. (1) and (2), it can be shown that all of the polygamma functions, including the digamma function, are analytic at all points of $\mathbb{C}$ except for at the nonpositive integers at which they have a pole of order $n+1$. That the digamma function is analytic everywhere except for the non-positive integers makes it a suitable candidate family of basis functions for use with the CVBEM.

### 3.1. Rotation of the digamma axis

The digamma function has a pole of order 1 at all of the non-positive integers (see Fig. 2 for visualization). It is important to note that all of the poles are aligned with the negative real axis. We refer to the line along which these poles occur as the digamma axis. Thus, when the digamma function has not been rotated, we say that the digamma axis is aligned with the negative real axis. However, it is possible to rotate the digamma axis by an arbitrary angle. Rotating the digamma axis is most easily performed using polar coordinates: $z=|z| e^{i \theta}$, where $\theta=\arg (z)$ and $\theta \in(-\pi, \pi]$. Then, the rotation is performed as follows:
$\psi_{\alpha}(z)=\psi\left(|z| e^{i(\theta+\alpha)}\right)$.
A important consideration when using the CVBEM is ensuring that the approximation function is analytic within the problem domain. This is necessary so that the real and imaginary parts of the CVBEM approximation function are harmonic within the problem domain. As a result, in order to obtain a CVBEM approximation function that is analytic within the problem domain, it is necessary to rotate the digamma axis associated with each basis function so it does not intersect the problem domain, as illustrated in Fig. 1.

Each digamma function in this section is denoted as follows: $\psi_{\alpha}(z)$, where $\alpha$ is the angle of rotation of the digamma axis with $\alpha=0$ corresponding to the digamma axis being aligned with the negative real axis.

### 3.2. A Linear Combination of digamma functions

We now consider linear combinations of the digamma function. The functions examined in this section demonstrate translations of the base digamma function as well as rotations of the digamma axis. Each digamma function in this section is denoted as follows: $\psi_{\alpha_{j}}\left(z-z_{j}\right)$, where $\alpha_{j} \in \mathbb{R}$ is the clockwise rotation angle of the digamma axis from


Fig. 2. Contours of the real and imaginary parts of $\psi_{0}(z)$ with the digamma axis rotated clockwise by an angle of $\alpha=0$ radians.
the negative real axis, and $z_{j} \in \mathbb{C}$ defines the translation of the basis function. Note: the subscript $j$ is used to denote specific instantiations of a function from within a given basis function family. In the case of the digamma basis function family, each instantiation is identified by its translation as well as the rotation of its digamma axis.

In the context of the digamma basis functions, the points $z_{j}$ can be interpreted as the "first" pole along the digamma axis. For computational purposes, the points $z_{j}$ are interpreted as the computational nodes whose locations will be determined through the use of an 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. However, although the computational nodes do not have a physical meaning within the problem context, their selected locations do effect the ability of the CVBEM approximation function to satisfy the given boundary conditions. For this reason, it is important to use an NPA to determine suitable locations for the computational nodes.

### 3.3. Variations of the digamma basis function

In this section, we propose novel variations of the digamma basis function. The selection of these particular variations was inspired by the form of the original CVBEM basis functions as given in [3]. In [3], the authors showed that integration of the Cauchy integral equation with linear boundary segments leads to a linear combination of functions each of the form:
$\left(z-z_{j}\right)\left[\ln \left(z-z_{j}\right)\right]_{\alpha_{j}}$,
where $z_{j} \in \mathbb{C}$, and $\alpha_{j}$ is the rotation angle of the branch cut of the natural logarithm. The branch cut is rotated for the same purpose that the digamma axis is rotated. A key insight is to recognize the basis functions in Eq. (4) as products of two distinct functions; namely, $\left(z-z_{j}\right)$ and $\left[\ln \left(z-z_{j}\right)\right]_{\alpha_{j}}$. The following proposed variants of the original digamma function are thus inspired by the components of the original CVBEM basis functions:

1. Variant 0: the standard digamma function, denoted $\psi_{\alpha_{j}}\left(z-z_{j}\right)$
2. Variant 1: $\left(z-z_{j}\right) \psi_{\alpha_{j}}\left(z-z_{j}\right)$, depicted in Figs. 10 and 11


Fig. 3. Domain coloring of $\psi_{0}(z)$ with the digamma axis rotated clockwise by an angle of $\alpha=0$ radians.


Fig. 4. Contours of the real and imaginary parts of $\psi_{\pi / 2}(z)$ with the digamma axis rotated clockwise by an angle of $\alpha=\pi / 2$ radians.
3. Variant 2: $\left(z-z_{j}\right)\left[\ln \left(z-z_{j}\right)\right]_{\alpha_{j}} \psi_{\alpha_{j}}\left(z-z_{j}\right)$, depicted in Figs. 12 and 13

The Variant 2 basis function family incorporates both a branch cut as well as a digamma axis. Both of these need to be rotated so as not to intersect the problem domain. For simplicity, we have selected to rotate both by the same angle so that they are collinear. That is, they are both rotated by $\alpha_{j}$ radians. However, it is possible to rotate the branch cut and digamma axis by different angles, although we have not assessed the efficacy of this approach.


Fig. 5. Domain coloring of $\psi_{\pi / 2}(z)$ with the digamma axis rotated clockwise by an angle of $\alpha=\pi / 2$ radians.


Fig. 6. Linear combination of two digamma functions: $\psi_{0}(z)+\psi_{5 \pi / 4}(z+1+2 i)$.

### 3.4. Modeling potential problems

Let $\Omega \subseteq \mathbb{C}$ denote a simply connected domain, referred to as the problem domain. In $\Omega$, assume there exists a harmonic function, $\phi$, satisfying boundary conditions specified on $\partial \Omega$. Since $\phi$ is harmonic in $\Omega$ and satisfies the given boundary conditions on $\partial \Omega$, it is known as the target potential function. Specifically, $\phi$ is the function we seek to approximate with the CVBEM.

The CVBEM was originally formulated as a boundary integral equation method based on the Cauchy integral equation, which is given as


Fig. 7. Domain coloring of the linear combination of two digamma functions: $\psi_{0}(z)+\psi_{5 \pi / 4}(z+1+2 i)$.


Fig. 8. Linear combination of two digamma functions: $\psi_{\pi / 2}(z-2 i)+\psi_{\pi}(z-2)+$ $\psi_{3 \pi / 2}(z+2 i)$.
follows [3]:
$\omega(z)=\frac{1}{2 \pi i} \int_{\partial \Omega} \frac{\omega(\zeta) \mathrm{d} \zeta}{\zeta-z}, \quad z \in \Omega$.
In [3] it was shown that integration of the Cauchy integral equation using linear boundary elements connecting locations of known boundary conditions leads to a CVBEM approximation function of the form:
$\hat{\omega}(z)=\sum_{j=1}^{n} c_{j}\left(z-z_{j}\right) \ln \left(z-z_{j}\right), \quad z \in \Omega$.
In eqn. (6), we note that the branch cut of each natural logarithm is rotated so as to avoid intersection with the problem domain. Addition-


Fig. 9. Domain coloring of the linear combination of two digamma functions: $\psi_{\pi / 2}(z-2 i)+\psi_{\pi}(z-2)+\psi_{3 \pi / 2}(z+2 i)$.


Fig. 10. Digamma (variant 1). Contours of the real and imaginary parts of $z \psi_{0}(z)$ with the digamma axis rotated clockwise by an angle of $\alpha=0$ radians.
ally, an important realization is that, in fact, any analytic complex variable function can be used as a basis function in place of $\left(z-z_{j}\right) \ln \left(z-z_{j}\right)$ in eqn. (6). Hence, a more general expression for the CVBEM approximation function is as follows:
$\hat{\omega}(z)=\sum_{j=1}^{n} c_{j} g_{j}(z), \quad z \in \Omega$.
The complex variable functions $g_{j}(z): \Omega \rightarrow \mathbb{C}$ can be selected by the modeler subject to the condition that they are analytic in $\Omega$. The coefficients $c_{j} \in \mathbb{C}$ are complex numbers, each composed of two real constants; namely, $\alpha_{j}=\mathfrak{R}\left(c_{j}\right)$ and $\beta_{j}=\mathfrak{\Im}\left(c_{j}\right)$. The more general form of the CVBEM approximation function given in eqn. (7) is reminiscent of the


Fig. 11. Digamma (variant 1). Domain coloring of $z \psi_{0}(z)$ with the digamma axis rotated clockwise by an angle of $\alpha=0$ radians.


Fig. 12. Digamma (variant 2). Contours of the real and imaginary parts of $z[\ln (z)]_{0} \psi_{0}(z)$ with the digamma axis and the branch cut of the natural logarithm both rotated clockwise by an angle of $\alpha=0$ radians.
method of fundamental solutions [18] and the analytic element method [19].

The CVBEM approximation function consists of two real variable functions, denoted $\hat{\phi}: \mathbb{R}^{2} \rightarrow \mathbb{R}$ and $\hat{\psi}: \mathbb{R}^{2} \rightarrow \mathbb{R}$, respectively, such that $\hat{\omega}(z)=\hat{\phi}(x, y)+i \hat{\psi}(x, y)$. In many applications, $\hat{\phi}$ is interpreted as the approximate potential function, and $\hat{\psi}$ is interpreted as the approximate conjugate stream function. Furthermore, the real-valued functions $\hat{\phi}$ and $\hat{\psi}$ are related by the Cauchy-Riemann equations and can be shown to be harmonic within $\Omega$ [20]. Thus,
$\Delta \hat{\phi}(x, y)=0 \quad$ and $\quad \Delta \hat{\psi}(x, y)=0, \quad(x+i y) \in \Omega$.


Fig. 13. Digamma (variant 2). Domain coloring of $z[\ln (z)]_{0} \psi_{0}(z)$ with the digamma axis and the branch cut of the natural logarithm both rotated clockwise by an angle of $\alpha=0$ radians.

The CVBEM approach is to approximate the target potential function, $\phi$, by determining the coefficients of Eq. (7) in order to minimize $\|\phi-\hat{\phi}\|$ in a given norm. Hence, the real part of the CVBEM approximation function corresponds to the approximation of the target potential function. The imaginary part of the CVBEM approximation function is the conjugate stream function.

Since each $c_{j}$ corresponds to two real numbers, there are a total of $2 n$ real coefficient values to be determined in a CVBEM model. These coefficient values are selected so as to reduce the error of the CVBEM approximation function. Since the real and imaginary parts of $\hat{\omega}$ are harmonic functions within $\Omega$, the CVBEM approximation function does not have any error satisfying Laplace's equation within the problem domain. Hence, the error of the CVBEM approximation function occurs with respect to continuously satisfying the boundary conditions. Therefore, satisfying the boundary conditions is the main computational effort after the computational nodes of the basis functions have been determined.

This paper is primarily interested in assessing the effect of the selection of particular families of the $g_{j}(z)$ on the maximum error of the resulting CVBEM approximation function.

### 3.4.1. Demonstration of continuous basis functions

The results shown in Figs. 14-16 demonstrate that the real and imaginary parts of linear combinations of rotated digamma functions are continuous along a simple, closed curve such as the example problem boundary. In general, since the digamma function is analytic except at the poles on the digamma axis, linear combinations of the digamma function are continuous on the problem boundary (since each digamma axis has been translated and rotated away from the problem boundary). The target potential function will also be continuous along the problem boundary, which motivates the proper selection of coefficient values in the linear combination of digamma functions in order to approximate the value of the target potential function on the problem boundary.

## 4. Computational comparison of CVBEM basis functions

Since the CVBEM approximation function is formulated as a linear combination of analytic complex variable functions, both the real and imaginary parts of the approximation function satisfy Laplace's equa-


Fig. 14. An example problem domain defined by $(-1.5,1.5) \times(-1.5 i, 0.5 i)$. Three rotated digamma basis functions are considered. Each side of the problem boundary has a unique color that corresponds to the colors shown in Figs. 15 and 16.


Fig. 15. Values of the real part of the linear combination of digamma functions shown in Fig 15.
tion. Therefore, no computational effort is required to satisfy the governing PDE. Rather, the computational effort consists of two tasks:
noitemsep Determining suitable locations for the computational nodes using an NPA
noiitemsep Determining the coefficients of the CVBEM approximation function in order to minimize the maximum error of the CVBEM approximation function with respect to satisfying the given boundary conditions

For discussion on the latest NPAs that have been developed for use with the CVBEM, see [7] and [8]. In this section, we are concerned with the task of determining the coefficients of the CVBEM approximation function by solving a system of linear equations.


Fig. 16. Values of the imaginary part of the linear combination of digamma functions shown in Fig 15.


Fig. 17. Problem geometry for the example problem depicted in Section 4.3. Zero-flux Neumann conditions are enforced along the left and bottom edges of the problem boundary. Dirichlet conditions are enforced on the top and right edges of the problem boundary. For visual clarity, only $10 \%$ of the total candidate collocation points are shown.


Fig. 18. Flow net produced using the CVBEM approximation of the demonstration problem with 50 digamma basis functions (variant 1). The demonstration problem has been formulated as a mixed BVP with Dirichlet conditions on the top and right sides of the problem domain and zero-flux Neumann conditions on the left and bottom sides.


Fig. 19. View of the flow net near the obstacle. This flow net was produced using the CVBEM approximation of the demonstration problem with 50 digamma basis functions (variant 1).


Fig. 20. Magnified view of the flow net near the north pole of the obstacle (note: the left and right edges of the obstacle are not depicted in this figure). This flow net was produced using the CVBEM approximation of the demonstration problem with 50 digamma basis functions (variant 1 ).

### 4.1. CVBEM Formulation for mixed boundary value problems

As indicated in the discussion after Eq. (7), the coefficients of the CVBEM approximation function are complex and, hence, have both a real and an imaginary part: $c_{j}=\alpha_{j}+i \beta_{j}$. Likewise, the basis functions in Eq. (7) 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, it follows:

$$
\begin{align*}
\hat{\omega}(z) & =\sum_{j=1}^{n} c_{j} g_{j}(z) \\
& =\sum_{j=1}^{n}\left(\alpha_{j}+i \beta_{j}\right)\left(\lambda_{j}(x, y)+i \mu_{j}(x, y)\right)  \tag{9}\\
& =\sum_{j=1}^{n}\left[\alpha_{j} \lambda_{j}(x, y)-\beta_{j} \mu_{j}(x, y)+i\left[\alpha_{j} \mu_{j}(x, y)+\beta_{j} \lambda_{j}(x, y)\right]\right] .
\end{align*}
$$

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

$$
\begin{aligned}
\mathfrak{R}[\hat{\omega}(z)]=\hat{\phi}(x, y) & =\sum_{j=1}^{n} \alpha_{j} \lambda_{j}(x, y)-\beta_{j} \mu_{j}(x, y) \\
& =\lambda^{\top} \boldsymbol{\alpha}-\boldsymbol{\mu}^{\top} \boldsymbol{\beta}, \\
\mathfrak{J}[\hat{\omega}(z)]=\hat{\psi}(x, y) & =\sum_{j=1}^{n} \alpha_{j} \mu_{j}(x, y)+\beta_{j} \lambda_{j}(x, y) \\
& =\boldsymbol{\mu}^{\top} \boldsymbol{\alpha}+\lambda^{\top} \boldsymbol{\beta},
\end{aligned}
$$



Fig. 21. View of the flow net in the bottom left corner (i.e. the origin) of the problem geometry. The curvature of the target potential function is greatest in this region. This flow net was produced using the CVBEM approximation of the demonstration problem with 50 digamma basis functions (variant 1).


Fig. 22. Magnified view of the flow net near the left edge of the obstacle. This flow net was produced using the CVBEM approximation of the demonstration problem with 50 digamma basis functions (variant 1).
where

$$
\begin{aligned}
& \boldsymbol{\alpha}=\left[\begin{array}{c}
\alpha_{1} \\
\alpha_{2} \\
\vdots \\
\alpha_{n}
\end{array}\right], \quad \boldsymbol{\beta}=\left[\begin{array}{c}
\beta_{1} \\
\beta_{2} \\
\vdots \\
\beta_{n}
\end{array}\right], \\
& \boldsymbol{\lambda}=\left[\begin{array}{c}
\lambda_{1}(x, y) \\
\lambda_{2}(x, y) \\
\vdots \\
\lambda_{n}(x, y)
\end{array}\right], \quad \boldsymbol{\mu}=\left[\begin{array}{c}
\mu_{1}(x, y) \\
\mu_{2}(x, y) \\
\vdots \\
\mu_{n}(x, y)
\end{array}\right] .
\end{aligned}
$$

In matrix form, Eq. (10) is written in the following convenient form:

$$
\left[\begin{array}{c}
\hat{\phi}(x, y)  \tag{11}\\
\hat{\psi}(x, y)
\end{array}\right]=\left[\begin{array}{cc}
\lambda^{\top} & -\boldsymbol{\mu}^{\top} \\
\boldsymbol{\mu}^{\top} & \lambda^{\top}
\end{array}\right]\left[\begin{array}{l}
\boldsymbol{\alpha} \\
\boldsymbol{\beta}
\end{array}\right] .
$$



Fig. 23. Maximum absolute error of CVBEM models resulting from the use of digamma basis functions (variant 1) for approximations using $n=1, \ldots, 50$.

In Eq. (10), $\hat{\phi}$ and $\hat{\psi}$ are defined as linear combinations using the same coefficients, $\alpha$ and $\beta$. Thus, as soon as $\alpha$ and $\beta$ have been determined for either $\hat{\phi}$ or $\hat{\psi}$, the same coefficients can be used to compute the other function. This phenomenon is a manifestation of the CauchyRiemann equations and the fact that $\hat{\phi}$ and $\hat{\psi}$ are harmonic conjugates of each other. In practice, this is beneficial because once the target potential function is computed, it is not necessary to use post-processing software to compute the orthogonal stream function, which is necessary for popular domain discretization methods such as the Finite Element Method. Rather, using the CVBEM approach, the orthogonal stream function is obtained using the same coefficients, $\alpha$ and $\beta$, in a linear combinations given in Eq. (10).

One approach to determining the values of $\alpha$ and $\beta$ is to use collocation. Collocation leads to a system of $2 n$ equations in $2 n$ unknowns. The advantage of collocation is that the CVBEM approximation function will exactly satisfy the given boundary conditions at the locations of the $2 n$ collocation points in the absence of truncation and round-off errors. Another approach to determine the coefficients is to set up a least squares problem using more than $2 n$ of the available boundary data points. The least squares approach uses more boundary data; however, there is no guarantee that the resulting CVBEM approximation function will be exact at any location on the boundary. For our purposes, we prefer to use collocation because the CVBEM approximation function is guaranteed to be exact at the collocation points.

For Dirichlet boundary value problems, the unknown coefficients are determined by collocation of either the real or imaginary part of the CVBEM approximation function with the given boundary data. On the other hand, a mixed boundary value problem is formulated as follows. Suppose the problem boundary, $\partial \Omega$, is partitioned into a set $\partial \Omega_{\mathrm{D}}$ where Dirichlet boundary conditions are given and a set $\partial \Omega_{\mathrm{N}}$ where zero-flux Neumann conditions are given. The sets $\partial \Omega_{\mathrm{D}}$ and $\partial \Omega_{\mathrm{N}}$ satisfy the following: $\partial \Omega=\partial \Omega_{D} \cup \partial \Omega_{N}$ and $\partial \Omega_{D} \cap \partial \Omega_{N}=\varnothing$.

Let $N_{\mathrm{D}}$ denote the number of NPA-selected collocation points at locations where Dirichlet boundary conditions are enforced, and let $N_{\mathrm{N}}$ denote the number of NPA-selected collocation points at locations where zero-flux Neumann boundary conditions are enforced. Since collocation is being used, $N_{\mathrm{D}}$ and $N_{\mathrm{N}}$ must satisfy $N_{\mathrm{D}}+N_{\mathrm{N}}=2 n$. The equations corresponding to Dirichlet conditions are given by

$$
\begin{aligned}
\hat{\phi}\left(x_{i, \mathrm{D}}, y_{i, \mathrm{D}}\right) & =\sum_{j=1}^{n} \alpha_{j} \lambda_{j}\left(x_{i, \mathrm{D}}, y_{i, \mathrm{D}}\right)-\beta_{j} \mu_{j}\left(x_{i, \mathrm{D}}, y_{i, \mathrm{D}}\right) \\
& =\phi\left(x_{i, \mathrm{D}}, y_{i, \mathrm{D}}\right) \\
& \text { for } i=1, \ldots, N_{\mathrm{D}}, \quad\left(x_{i, \mathrm{D}}, y_{i, \mathrm{D}}\right) \in \partial \Omega_{\mathrm{D}}
\end{aligned}
$$

The equations corresponding to zero-flux Neumann conditions are given by

$$
\begin{align*}
\hat{\psi}\left(x_{i, \mathrm{~N}}, y_{i, \mathrm{~N}}\right) & =\sum_{j=1}^{n} \alpha_{j} \mu_{j}\left(x_{i, \mathrm{~N}}, y_{i, \mathrm{~N}}\right)+\beta_{j} \lambda_{j}\left(x_{i, \mathrm{~N}}, y_{i, \mathrm{~N}}\right)  \tag{13}\\
& =\text { const, } \\
& \text { for } i=1, \ldots, N_{\mathrm{N}}, \quad\left(x_{i, \mathrm{~N}}, y_{i, \mathrm{~N}}\right) \in \partial \Omega_{\mathrm{N}}
\end{align*}
$$

The locations on the boundary at which the Dirichlet and zero-flux Neumann conditions are enforced are conveniently written in vector form as follows:

$$
\begin{gather*}
\boldsymbol{x}_{\mathrm{D}}=\left[\begin{array}{c}
x_{1, \mathrm{D}} \\
x_{2, \mathrm{D}} \\
\vdots \\
x_{N_{\mathrm{D}}, \mathrm{D}}
\end{array}\right], \quad \boldsymbol{y}_{\mathrm{D}}=\left[\begin{array}{c}
y_{1, \mathrm{D}} \\
y_{2, \mathrm{D}} \\
\vdots \\
y_{N_{\mathrm{D}}, \mathrm{D}}
\end{array}\right],  \tag{14}\\
\boldsymbol{x}_{\mathrm{N}}=\left[\begin{array}{c}
x_{1, \mathrm{~N}} \\
x_{2, \mathrm{~N}} \\
\vdots \\
x_{N_{\mathrm{N}}, \mathrm{~N}}
\end{array}\right], \quad \text { and } \quad \boldsymbol{y}_{\mathrm{N}}=\left[\begin{array}{c}
y_{2, \mathrm{~N}} \\
\vdots \\
y_{N_{\mathrm{N}}, \mathrm{~N}}
\end{array}\right] .
\end{gather*}
$$

The desired matrix equation of size $2 n \times 2 n$ can be written in the following block matrix form. Namely,

$$
\begin{aligned}
& {\left[\begin{array}{ll}
\underbrace{\boldsymbol{\phi}\left(\boldsymbol{x}_{\mathrm{D}}, \boldsymbol{y}_{\mathrm{D}}\right)}_{N_{\mathrm{D}} \times 1} \\
\underbrace{\boldsymbol{f}}_{N_{\mathrm{N}} \times 1}
\end{array}\right]=[\begin{array}{ll}
\underbrace{\lambda\left(\boldsymbol{x}_{\mathrm{D}}, \boldsymbol{y}_{\mathrm{D}}\right)}_{N_{\mathrm{D}} \times n} & \underbrace{-\mu\left(\boldsymbol{x}_{\mathrm{D}}, \boldsymbol{y}_{\mathrm{D}}\right)}_{N_{N_{\mathrm{N}} \times n}} \\
\mu\left(\boldsymbol{x}_{\mathrm{N}}, \boldsymbol{y}_{\mathrm{N}}\right)
\end{array} \underbrace{\lambda\left(\boldsymbol{x}_{\mathrm{N}}, \boldsymbol{y}_{\mathrm{N}}\right)}_{N_{\mathrm{N}} \times n}]\left[\begin{array}{c}
\underbrace{\boldsymbol{\alpha}}_{n \times 1} \boldsymbol{\underbrace { } _ { n }} \boldsymbol{\beta}
\end{array}\right],} \\
& \text { where } f=\gamma\left[\begin{array}{c}
1 \\
1 \\
1 \\
\vdots \\
1
\end{array}\right], \quad \gamma \in \mathbb{R} \text {. }
\end{aligned}
$$

Lastly, it should be noted that the system of equations defined by Eq. (15) 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.

### 4.2. Error estimation of the CVBEM approximation function

As previously mentioned, the CVBEM approximation function is analytic within $\Omega$. Therefore, $\hat{\phi}$ and $\hat{\psi}$ satisfy Laplace's equation in $\Omega$. Consequently, the CVBEM approximation function does not have any error with regard to satisfying the governing PDE. Instead, error estimation pertains to assessing the ability of the CVBEM approximation function to satisfy the given boundary conditions. In particular, the approximation error is estimated separately for the Dirichlet and Neumann partitions of the boundary as follows:

$$
\begin{align*}
& \varepsilon_{\mathrm{D}}=\max _{(x, y) \in \partial \Omega_{\mathrm{D}}}|\phi(x, y)-\hat{\phi}(x, y)|,  \tag{16}\\
& \varepsilon_{\mathrm{N}}=\max _{(x, y) \in \partial \Omega_{\mathrm{N}}}|\gamma-\hat{\psi}(x, y)|,
\end{align*}
$$

where $\gamma$ is as defined in Eq. (15). Then, we define the maximum error of the CVBEM model as $\varepsilon=\max \left\{\varepsilon_{\mathrm{D}}, \varepsilon_{\mathrm{N}}\right\}$. Since $\hat{\omega}$ is a linear combination of analytic complex variable functions, both $\hat{\phi}$ and $\hat{\psi}$ can be evaluated continuously within $\Omega \cup \partial \Omega$. 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. A reasonable approximation of $\varepsilon$ can be obtained by computing the values of $\varepsilon_{\mathrm{D}}$ and $\varepsilon_{\mathrm{N}}$ at many locations along the problem boundary. In this work, we computed either $\varepsilon_{D}$ or $\varepsilon_{N}$ at a total of 2000 different reasonablyspaced locations on the problem boundary.

### 4.3. Example problem and results

In this section, we test the efficacy of several basis function families with regard to minimizing the maximum error of a CVBEM model of a benchmark mixed boundary value problem. The basis function families examined in this work are as follows:

1. Standard CVBEM functions: $\left(z-z_{j}\right) \ln \left(z-z_{j}\right)_{\alpha_{j}}$
2. Log functions: $\ln \left(z-z_{j}\right)_{\alpha_{j}}$
3. Simple poles $\frac{1}{z-z_{j}}$
4. Digamma Variant 0: $\psi_{\alpha_{j}}\left(z-z_{j}\right)$
5. Digamma Variant 1: $\left(z-z_{j}\right) \psi_{\alpha_{j}}\left(z-z_{j}\right)$
6. Digamma Variant 2: $\left(z-z_{j}\right)\left[\ln \left(z-z_{j}\right)\right]_{\alpha_{j}} \psi_{\alpha_{j}}\left(z-z_{j}\right)$

Of the candidate basis function families, the standard CVBEM functions and the simple poles were previously examined in [6]. The simple pole basis functions have also recently been examined in [16]. However, the assessment of the other four basis functions is unique to this work. The standard CVBEM basis functions and the simple poles are included in this work to compare the newly proposed basis function families with basis function families that have been successfully used in the past.

Furthermore, all of the basis functions have been selected because they each incorporate the use of computational nodes and are, therefore, compatible with node positioning algorithms. It was shown in [8] that node positioning algorithms can be used to reduce the maximum error of CVBEM models by several orders of magnitude. Other basis functions such as complex polynomials, which have been examined in several previous works [9-12,21], do not incorporate the use of computational nodes, and we have found that they are less reliable in practice.

For the CVBEM models developed in this section, the digamma axis is rotated at the same angle as the branch cut of the logarithm functions.

In the case of $n=10$, we found that the best-performing basis function family was the family of natural logarithms with rotated branch cuts. However, in the other situations examined, we found that the Digamma Variant 1 basis function family consistently resulted in the CVBEM approximation function with least maximum error. That the Digamma Variant 1 basis function family consistently out-performed the standard CVBEM basis functions suggests that these basis functions may have broad applicability to the mesh-free methods, and further research to determine this may be warranted.

## 5. Discussion and topics for future research

This paper can be viewed as an update of the work done in [6], which was focused on comparing the regularly-used basis functions with the CVBEM methodology in the context of a benchmark Dirichlet BVP of the Laplace type. In the present work, we propose the family of digamma functions as new basis functions for use with the CVBEM and other mesh-free computational methods. To this end, we examined the digamma function, denoted $\psi_{\alpha_{j}}\left(z-z_{j}\right)$, as well as two novel variants: $\left(z-z_{j}\right) \psi_{\alpha_{j}}\left(z-z_{j}\right)$ and $\left(z-z_{j}\right)\left[\ln \left(z-z_{j}\right)\right]_{\alpha_{j}} \psi_{\alpha_{j}}\left(z-z_{j}\right)$. These basis functions were compared to the standard CVBEM basis functions as well as simple pole basis functions.

The other sense in which this work is an update to [6] is the use of an NPA for the determination of computational nodes and collocation points. The recent work developing NPAs in $[7,8]$ has demonstrated the significance that the selection of node location and collocation point location has on the accuracy of the resulting CVBEM approximation. At the time of [6], these NPAs did not yet exist.

Lastly, the successful implementation of mixed boundary condition capabilities represents an important advancement in the state-of-the-art for CVBEM technology.

The good computational results obtained for our demonstration problem suggest it would be reasonable to explore other similar candidate basis function families, such as the more general polygamma function set, for making further progress in mesh-free numerical methods for solving partial differential equations.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

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

## References

[1] Stirling J. Methodus differentialis: sive tractatus de summatione et interpolatione serierum infinitarum. Typis Gul Bowyer; 1730.
[2] Hromadka II T, Guymon G. Application of a boundary integral equation to prediction of freezing fronts in soil. Cold Reg Sci Technol 1982;6:115-21.
[3] Hromadka II TV, Guymon GL. A complex variable boundary element method: development. Int J Numer Methods Eng 1984;20:25-37.
[4] Wilkins BD, Greenberg J, Redmond B, Baily A, Flowerday N, Kratch A, et al. An unsteady two-dimensional complex variable boundary element method. SCIRP Appl Math 2017;8(6):878-91.
[5] Wilkins BD, Hromadka II TV, Boucher R. A conceptual numerical model of the wave equation using the complex variable boundary element method. Appl Math (Irvine) 2017;8(5):724.
[6] Wilkins BD, Hromadka II TV, Johnson AN, Boucher R, McInvale HD, Horton S. Assessment of complex variable basis functions in the approximation of ideal fluid flow problems. Int J Comput Method Exp Measurement 2019;7(1):45-56.
[7] Demoes NJ, Bann GT, Wilkins BD, Grubaugh KE, Hromadka II TV. Optimization algorithm for locating computational nodal points in the method of fundamental solutions to improve computational accuracy in geosciences modeling. Professional Geol 2019.
[8] Wilkins BD, Hromadka II TV, McInvale J. Comparison of two algorithms for locating computational nodes in the complex variable boundary element method (cvbem). Int J Comput Method Exp Measurement 2020;8(4).
[9] Hromadka II T, Guymon G. Complex polynomial approximation of the laplace equation. ASCE J Hydraulic Eng 1984;110(3):329-39.
[10] Yen C, Hromadka II T. Complex polynomial approximation of two-dimensional potential problems using generalized fourier series. Microsoftw Eng 1986;2(2):75-81.
[11] Hromadka II T, Whitley R. Approximating three-dimensional steady-state potential flow problems using two-dimensional complex polynomials. Eng Anal Bound Elem 2005;29:190-4.
[12] Poler A, Bohannon A, Schowalter S, Hromadka II T. Using the complex polynomial method with mathematica to model problems involving the laplace and poisson equations. Numer Methods Partial Differ Equ 2009;25(3):657-67.
[13] Dean T, Hromadka II TV, Kastner T, Phillips M. Modeling potential flow using laurent series expansions and boundary elements. Numer Methods Partial Differ Equ 2010;28(2).
[14] Johnson AN, Hromadka II TV, Hughes MT, Horton SB. Modeling mixed boundary problems with the complex variable boundary element method (cvbem) using matlab and mathematica. Int J Comput Method Exp Measurement 2015;3(3):269-78.
[15] Trefethen L.N.. Rational functions. John von Neumann Lecture; 2020a.
[16] Trefethen LN. Rational functions and beyond. SIAM News 2020.
[17] Johnson AN, Hromadka II TV. Modeling mixed boundary conditions in a hilbert space with the complex variable boundary element method (cvbem). MethodsX 2015;2:292-305.
[18] Mathon R, Johnston R. The approximate solution of elliptic boundary-value problems by fundamental solutions. SIAM J Numer Anal 1977;14(4):638-50.
[19] Strack O. Theory and applications of the analytic element method. Rev Geophys 2003;41(2):1005.
[20] Brown JW, Churchill RV. Complex variables and applications. 8th. McGraw-Hill; 2009.
[21] Bohannon A, Hromadka II T. The complex polynomial method with a leastsquares fit to boundary conditions. Eng Anal Bound Elem 2009;33(8-9):1100-2. doi:10.1016/j.enganabound.2009.02.005.


[^0]:    * Corresponding author.

    E-mail addresses: bwilkins@andrew.cmu.edu (B.D. Wilkins), theodore.hromadka@westpoint.edu (T.V. Hromadka II).

