Using Series Basis Functions with the Complex Variable Boundary Element Method

Bryce D. Wilkins[1],
Theodore V. Hromadka II[2]

[1]Carnegie Mellon University
[2]Distinguished Professor, United States Military Academy

bwilkins@andrew.cmu.edu

theodore.hromadka@westpoint.edu

July 6, 2021

Section 1

The CVBEM Methodology with Digamma and Polygamma Basis Functions

The General CVBEM Approximation Function

- The CVBEM approximation function is a linear combination of complex variable functions that are analytic within a given problem domain, Ω:

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

- where
  - $c_j = \alpha_j + i\beta_j$ are complex coefficients,
  - $g_j(z)$ are analytic complex variable basis functions,
  - n is the number of basis functions being used in the approximation
- In the collocation approach, each term in the approximation function corresponds to one node and two collocation points.

Some possible basis functions:

- $(z - z_j)\ln_{\alpha_j}(z - z_j)$
- $1/(z - z_j)$
- $(z - z_j)^j$
- Digamma
- Polygamma
- And more!

Hromadka II, T.V., Guymon, G.L., A Complex Variable Boundary Element Method: Development. *International Journal for Numerical Methods in Engineering*, pp. 25-37, 1984.

Digamma Basis Functions

A series representation for the digamma function is as follows:

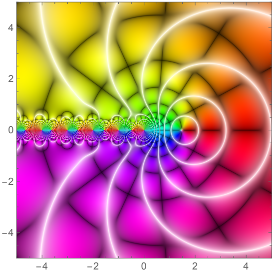
$$\psi(z) = -\gamma + \sum_{m=0}^{\infty} \frac{z-1}{(m+1)(m+z)}, \quad z \neq 0, -1, -2, \ldots, \qquad (2)$$

where $\gamma \approx 0.5772156$ denotes the Euler-Mascheroni constant.

- Using the digamma functions, the CVBEM approximation function is as follows:

$$\hat{\omega}(z) = \sum_{j=1}^{n} c_j \psi_{\alpha_j}(z - z_j). \qquad (3)$$

- For each basis function, the point z_j is the location of the "first" singularity of the digamma function. The points z_j are referred to as computational nodes.



Wilkins, B.D., Hromadka II, T.V., Using the digamma function for basis functions in mesh-free computational methods. *Engineering Analysis with Boundary Elements*, 2021 (in-press).

Rotation of the Digamma Axis

The digamma axis is the line containing all of the singularities of the digamma function. By default, the axis is aligned parallel to the negative real axis. Therefore, we need to rotate this axis so that singularities of the basis functions do not occur in the problem domain.

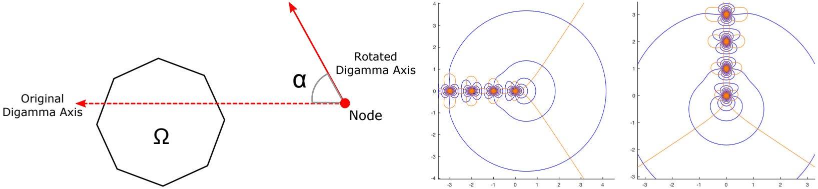


Figure: Rotation of the digamma axis. Our standard approach is to rotate the digamma axis radially away from the center of the problem domain. The node depicted in the figure refers to the point z_j of the basis function, and α is the rotation angle.

Node Position Algorithm (NPA)

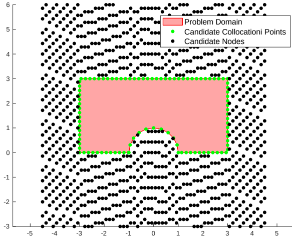


Figure: The NPA begins by establishing candidate nodes in the exterior of the problem domain. Candidate nodes are tested one-at-a-time to determine which node contributes the most to reducing the maximum error of the CVBEM approximation function.

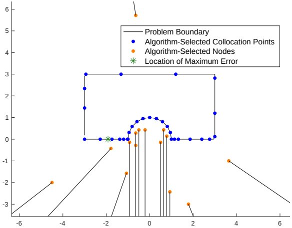


Figure: After a node is selected, the maximum error of the resulting CVBEM model is assessed on the problem boundary. Two new collocation points are added at the two highest local maxima of the error function on the boundary.

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*, pp. 6-12, 2019.

Node Position Refinement

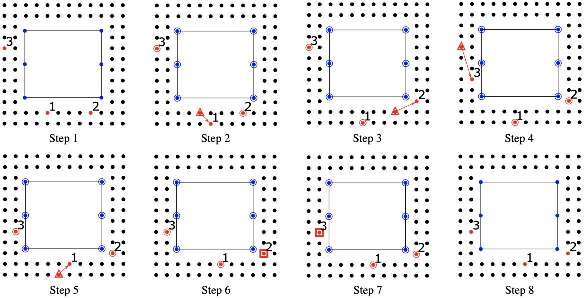


Figure: A refinement procedure is used, which allows for the possibility of exchanging a previously-selected node for a new node when the new node would further reduce the maximum error of the CVBEM model given the current selection of nodes and collocation points. The refinement procedure monotonically decreases the error of the CVBEM approximation function.

Wilkins, B.D., Hromadka II, T.V. & McInvale, J., Comparison of Two Algorithms for Locating Nodes in the Complex Variable Boundary Element Method (CVBEM). *International Journal of Computational Methods and Experimental Measurements*, pp. 289-315, 2020.

Adapting the NPA to the Polygamma Basis Functions

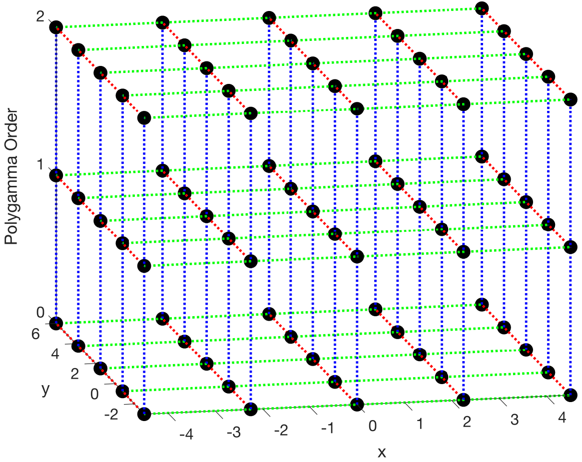


Figure: When using polygamma basis functions, the NPA has to determine the x and y coordinates of the node as well as the order of the polygamma basis function to use. This results in a 3D search, as indicated in the figure. Each layer (in the vertical direction) of the candidate nodes corresponds to a different order of the polygamma function.

Mechanics of the CVBEM

The CVBEM approximation function:

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

Let:

$$\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix}\beta_1\\ \beta_2\\ \vdots\\ \beta_n\end{bmatrix},$$

$$\boldsymbol{\lambda} = \begin{bmatrix}\lambda_1(x,y)\\ \lambda_2(x,y)\\ \vdots\\ \lambda_n(x,y)\end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix}\mu_1(x,y)\\ \mu_2(x,y)\\ \vdots\\ \mu_n(x,y)\end{bmatrix}.$$

The real and imaginary parts of the CVBEM approximation function:

$$\begin{aligned}\Re\left[\hat{\omega}(z)\right] = \hat{\phi}(x,y) &= \sum_{j=1}^{n} \alpha_j\lambda_j(x,y) - \beta_j\mu_j(x,y) \\ &= \boldsymbol{\lambda}^\top\boldsymbol{\alpha} - \boldsymbol{\mu}^\top\boldsymbol{\beta}, \\ \Im\left[\hat{\omega}(z)\right] = \hat{\psi}(x,y) &= \sum_{j=1}^{n} \alpha_j\mu_j(x,y) + \beta_j\lambda_j(x,y) \\ &= \boldsymbol{\mu}^\top\boldsymbol{\alpha} + \boldsymbol{\lambda}^\top\boldsymbol{\beta}.\end{aligned}$$

In matrix form:

$$\begin{bmatrix}\hat{\phi}(x,y)\\ \hat{\psi}(x,y)\end{bmatrix} = \begin{bmatrix}\boldsymbol{\lambda}^\top & -\boldsymbol{\mu}^\top\\ \boldsymbol{\mu}^\top & \boldsymbol{\lambda}^\top\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix}.$$

Handling the Mixed Boundary Conditions

The Dirichlet boundary conditions:

$$\begin{aligned}\hat{\phi}(x_{i,\mathrm{D}}, y_{i,\mathrm{D}}) &= \sum_{j=1}^{n} \alpha_j \lambda_j(x_{i,\mathrm{D}}, y_{i,\mathrm{D}}) - \beta_j \mu_j(x_{i,\mathrm{D}}, y_{i,\mathrm{D}}) \\ &= \phi(x_{i,\mathrm{D}}, y_{i,\mathrm{D}}), \\ &\text{for} \quad i = 1, \ldots, N_\mathrm{D}, \quad (x_{i,\mathrm{D}}, y_{i,\mathrm{D}}) \in \partial\Omega_\mathrm{D}.\end{aligned}$$

Let:

$$\boldsymbol{x}_\mathrm{D} = \begin{bmatrix} x_{1,\mathrm{D}} \\ x_{2,\mathrm{D}} \\ \vdots \\ x_{N_\mathrm{D},\mathrm{D}} \end{bmatrix}, \quad \boldsymbol{y}_\mathrm{D} = \begin{bmatrix} y_{1,\mathrm{D}} \\ y_{2,\mathrm{D}} \\ \vdots \\ y_{N_\mathrm{D},\mathrm{D}} \end{bmatrix},$$

$$\boldsymbol{x}_\mathrm{N} = \begin{bmatrix} x_{1,\mathrm{N}} \\ x_{2,\mathrm{N}} \\ \vdots \\ x_{N_\mathrm{N},\mathrm{N}} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{y}_\mathrm{N} = \begin{bmatrix} y_{1,\mathrm{N}} \\ y_{2,\mathrm{N}} \\ \vdots \\ y_{N_\mathrm{N},\mathrm{N}} \end{bmatrix}.$$

The Neumann boundary conditions:

$$\begin{aligned}\hat{\psi}(x_{i,\mathrm{N}}, y_{i,\mathrm{N}}) &= \sum_{j=1}^{n} \alpha_j \mu_j(x_{i,\mathrm{N}}, y_{i,\mathrm{N}}) + \beta_j \lambda_j(x_{i,\mathrm{N}}, y_{i,\mathrm{N}}) \\ &= \text{const}, \\ &\text{for} \quad i = 1, \ldots, N_\mathrm{N}, \quad (x_{i,\mathrm{N}}, y_{i,\mathrm{N}}) \in \partial\Omega_\mathrm{N}.\end{aligned}$$

In matrix form:

$$\begin{bmatrix} \underbrace{\boldsymbol{\phi}(\boldsymbol{x}_\mathrm{D}, \boldsymbol{y}_\mathrm{D})}_{N_\mathrm{D} \times 1} \\ \underbrace{\boldsymbol{f}}_{N_\mathrm{N} \times 1} \end{bmatrix} = \begin{bmatrix} \underbrace{\boldsymbol{\lambda}(\boldsymbol{x}_\mathrm{D}, \boldsymbol{y}_\mathrm{D})}_{N_\mathrm{D} \times n} & \underbrace{-\boldsymbol{\mu}(\boldsymbol{x}_\mathrm{D}, \boldsymbol{y}_\mathrm{D})}_{N_\mathrm{D} \times n} \\ \underbrace{\boldsymbol{\mu}(\boldsymbol{x}_\mathrm{N}, \boldsymbol{y}_\mathrm{N})}_{N_\mathrm{N} \times n} & \underbrace{\boldsymbol{\lambda}(\boldsymbol{x}_\mathrm{N}, \boldsymbol{y}_\mathrm{N})}_{N_\mathrm{N} \times n} \end{bmatrix} \begin{bmatrix} \underbrace{\boldsymbol{\alpha}}_{n \times 1} \\ \underbrace{\boldsymbol{\beta}}_{n \times 1} \end{bmatrix},$$

$$\text{where } \boldsymbol{f} = \gamma \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \gamma \in \mathbb{R}.$$

Section 2

Example Problem and Results

Example Problem Details

Problem Domain:	$\Omega = \left\{(x,y):\ 0 < x < 8,\ 0 < y < 5,\ \text{and } (x-5)^2 + y^2 > 1\right\}$
Governing PDE:	$\nabla^2 \phi = 0$
Boundary Conditions:	$\begin{cases} \frac{\partial \phi}{\partial n} = 0, & x = 0 \\ \frac{\partial \phi}{\partial n} = 0, & y = 0 \\ \frac{\partial \phi}{\partial n} = 0, & (x-5)^2 + y^2 = 1 \\ \phi(x,y) = \Re\left[z^2\right] = x^2 - y^2, & \text{otherwise} \end{cases}$
Number of Candidate Computational Nodes:	2,000
Number of Candidate Collocation Points:	2,000

CVBEM Modeling Outcomes

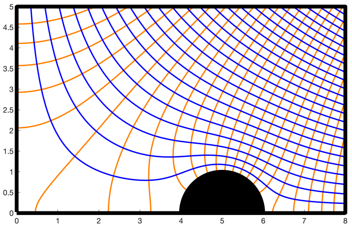


Figure: CVBEM-produced flownet depicting the entire problem domain. The CVBEM model was developed using polygamma series-type basis functions.

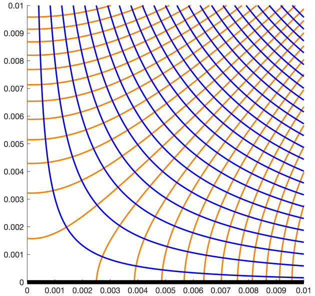


Figure: CVBEM-produce flownet near the origin where we observe potential flow in a 90-degree bend. Here, the flow situation is computationally difficult to model because of the relatively extreme curvature of the flow regime.

CVBEM Modeling Outcomes (continued)

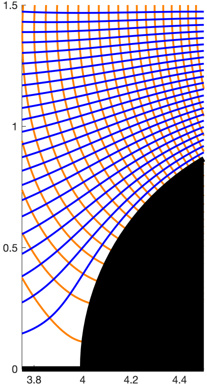


Figure: CVBEM-produced flownet near the left edge of the cylindrical obstacle.

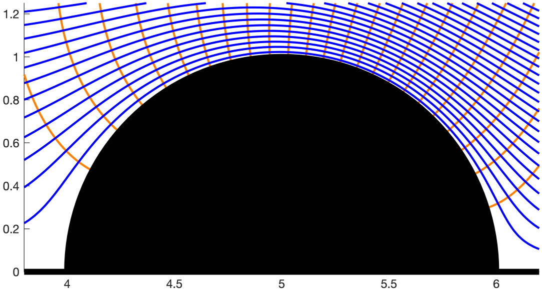


Figure: CVBEM-produced flownet near the obstacle. The north pole of the obstacle, as well as the left and right edges of the obstacle, are areas of interest due to the relatively extreme nature of the curvature of the flow situation in those areas.

Error Results

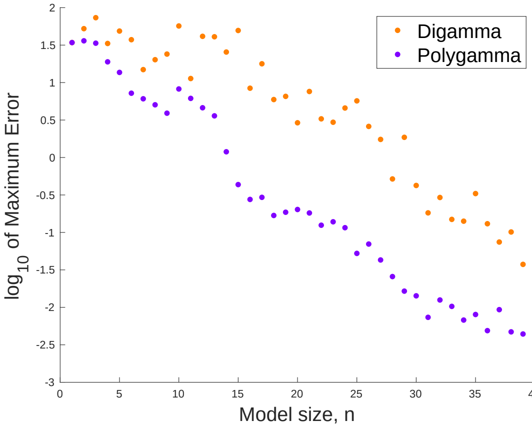


Figure: Maximum absolute error of CVBEM models created using digamma basis functions and polygamma basis functions (orders 0 through 9). The maximum error is reported as each new node is added to the CVBEM model up to a total of 40 nodes. After $n = 10$, it is clear that the polygamma basis functions lead to CVBEM models consistently more accurate than the CVBEM approximations using digamma basis functions.

Section 3

Final Thoughts - New Polygamma Search Technique

Line-search approach

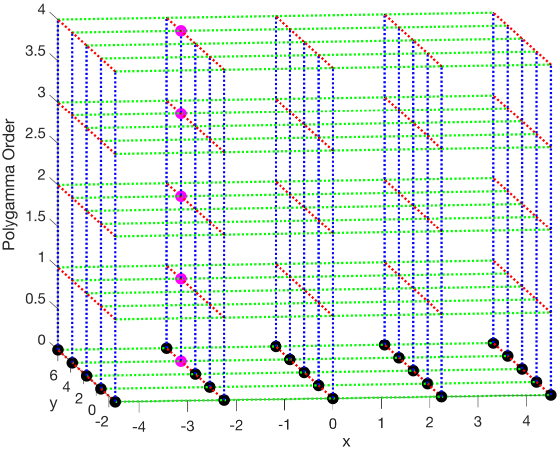


Figure: An alternative to the 3D search method is to apply the standard 2D search method while only considering digamma basis functions. Once the location of the highest-performing digamma basis function is determined, then that location can be held fixed as the NPA optimizes the order of the polygamma function to use in the CVBEM model at that location.

Questions