Improvement of Node Positioning Algorithms for the Complex Variable Boundary Element Method Using a Position Refinement Procedure

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

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

bdwilkins95@gmail.com

November 18, 2020

Overview

Section 1

Summary of CVBEM Methodology

CVBEM Fundamentals

Theorem (The Cauchy Integral Formula)

Let $\Gamma = \partial\Omega$ be a simple closed contour, and suppose $\omega(z)$ is analytic on $\Omega \cup \partial\Omega$. Then, for any point, $z \in \Omega$,

$$\omega(z) = \frac{1}{2\pi i} \oint_{\Gamma} \frac{\omega(\zeta) d\zeta}{\zeta - z}. \tag{1}$$

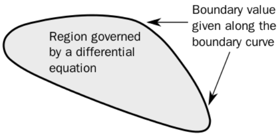


Figure: General boundary value problem suitable for modeling with the Cauchy integral equation. The boundary is a simple, closed contour.

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.

$$\hat{\omega}(z) = \sum_{j=1}^{n} c_j g_j(z), \qquad z \in \Omega, \tag{2}$$

where

- $c_j \in \mathbb{C}$ are complex coefficients,
- $g_j(z)$ are the complex variable basis functions being used in the approximation,
- n is the number of basis functions being used in the approximation

We note there are $2n$ degrees of freedom since each complex coefficient has unknown real and imaginary parts as follows:

$$c_j = \alpha_j + i\beta_j.$$

Problem Formulation

When straight line segments are used to discretize the boundary of the problem domain, the integration of the Cauchy integral formula results in the following sum, which is known as the CVBEM approximation function:

$$\hat{\omega}(z) = \sum_{j=1}^{n} c_j (z - z_j) \ln(z - z_j).$$

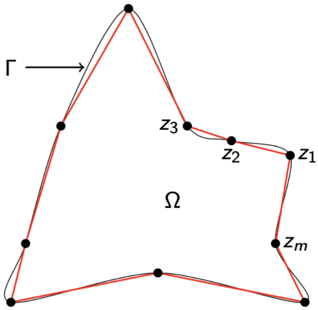


Figure: The boundary is discretized using a set of interpolation points. The interpolation points can be connected using straight line segments to create a polygonal representation.

The CVBEM Modeling Procedure

- The points z_j are the branch points of the logarithm (with branch cuts rotated) and are often referred to as computational nodes.
- Collocation with known boundary conditions is used to determine the coefficients of the approximation function.
- Once the coefficients are known, the approximate equipotential and stream lines can be evaluated continuously in the plane as the real and imaginary parts of the CVBEM function, respectively.

The CVBEM approximation function is as follows:

$$\hat{\omega}(z) = \sum_{j=1}^{n} c_j (z - z_j) \ln(z - z_j).$$

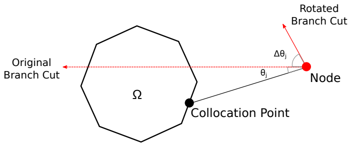


Figure: Rotation of a typical branch cut.

Section 2

The Position Refinement Procedure

NPA1

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.

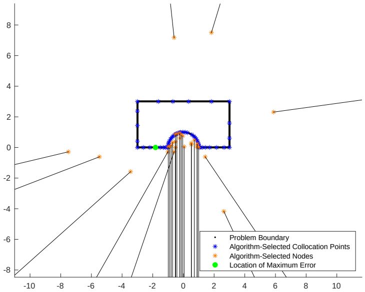


Figure: Nodes and collocation points are selected so as to decrease error in fitting boundary conditions.

NPA2

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*, in press.

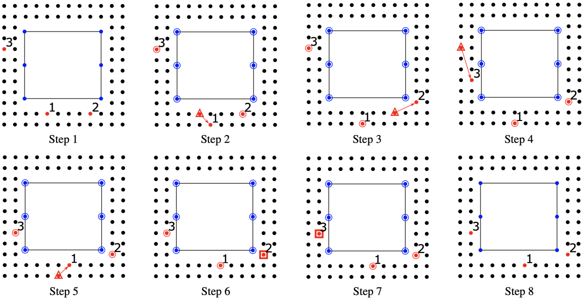


Figure: A refinement procedure is added, which allows for the re-location of previously located nodes.

NPA1 vs NPA2

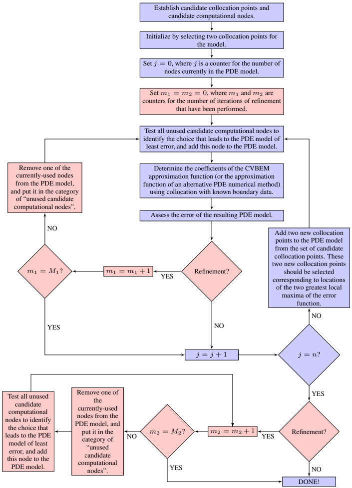


Figure: Flow chart depicting the steps of NPAs 1 and 2. The steps that are unique to NPA2 are colored red. The steps that are shared by both NPA1 and NPA2 are colored blue. M_1 denotes the user-specified number of iterations of refinement to be performed upon the selection of each new node. M_2 denotes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

Section 3

Example Problem and Results

Example Problem Details

Table: Example Problem 2 - Problem Description

Problem Domain:	$\Omega = \left\{ (x, y) : -0.325 \leq x \leq 8,\ 0 \leq y \leq 5, \text{ and } (x - 4.9125)^2 + y^2 \geq 0.975^2 \right\}$
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

Analytic Solution

- The example problem considers potential flow around a cylinder with the analytic solution given by:

$$\omega(z) = z^2 + z + \frac{10}{z-5}$$

- The flow regime approaches potential flow in a 90-degree bend near $(0,0)$.
- Certain areas of this problem are difficult to model computationally because of the extreme curvature of the flow regime.

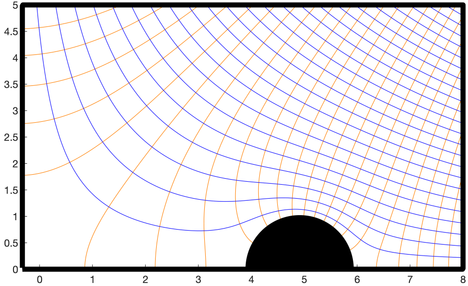


Figure: Analytic solution used for comparison between NPA1 and NPA2. There are areas of extreme curvature in the flow situation near $(0,0)$, as well as near the cylinder obstacle.

NPA Comparisons

Computational results from the use of NPAs 1 and 2 to determine the nodes and collocation points of a CVBEM model with $n = 20$.

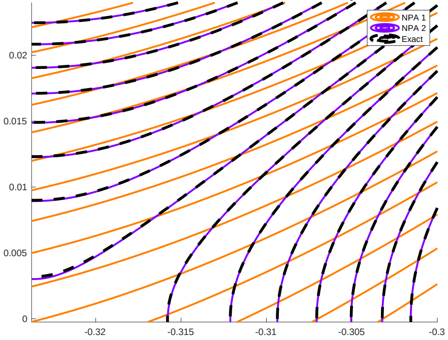


Figure: Streamlines.

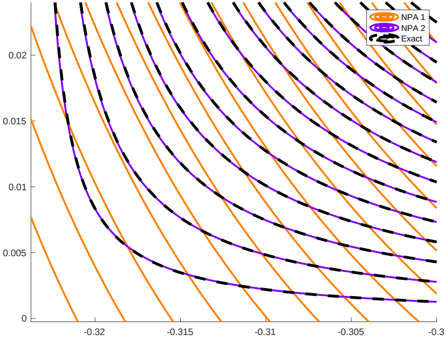
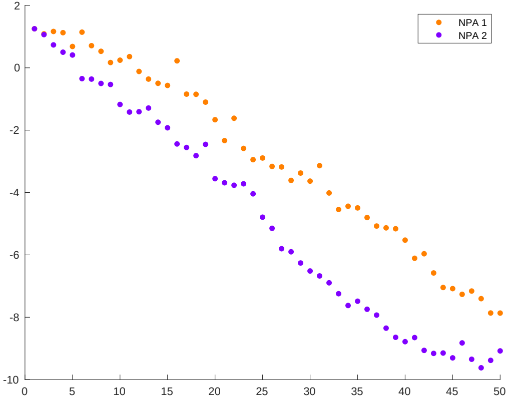


Figure: Potential lines.

Error Results

Figure: Maximum absolute error of CVBEM models resulting from the use of NPAs 1 and 2 as each new node is added up to a total of 50 nodes. After $n = 5$, it is clear that the NPA2 approximation is several orders of magnitude more accurate than the NPA1 approximation.



Time Results

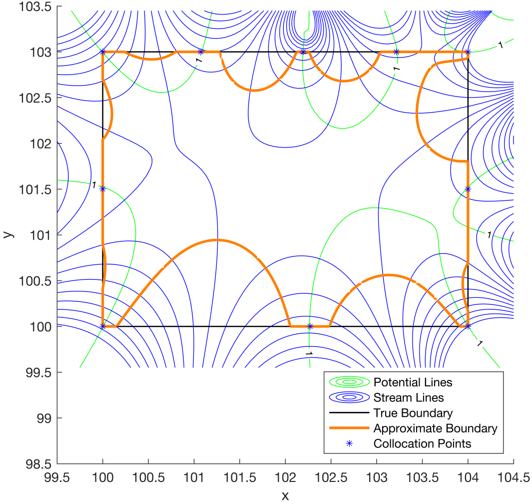
Number of Basis Functions	Number of d.o.f.	Unrefined Method (NPA1): Maximum Error	Unrefined Method (NPA1): Time Elapsed (sec)
10	20	1.757095e+00	1.178742
20	40	2.165579e-02	2.326950
30	60	2.325910e-04	3.681986
40	80	2.979909e-06	5.477926
50	100	1.362534e-08	6.828074
Number of Basis Functions	Number of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

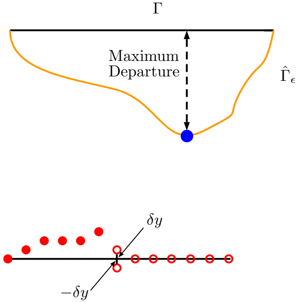
Table: Maximum error and time elapsed for various CVBEM models of a Dirichlet boundary value problem.

Section 4

Final Thoughts - The Approximate Boundary Method

The Approximate Boundary Method





Questions