

Application of Two Dimensional Extended Boundary Node Method to Potential Problem

Ayumu SAITOH, Taku ITOH¹⁾, Atsushi KAMITANI²⁾, Nobuyuki MATSUI
and Hiroaki NAKAMURA³⁾

Graduate School of Engineering, University of Hyogo, Himeji 671-2280, Japan

¹⁾*Faculty of Science and Technology, Seikei University, Musashino 180-8633, Japan*

²⁾*Graduate School of Science and Engineering, Yamagata University, Yonezawa 992-8510, Japan*

³⁾*Department of Simulation Science, National Institute for Fusion Science, Toki 509-5292, Japan*

(Received 7 December 2009 / Accepted 25 February 2010)

The boundary node method has been reformulated without using any integration cells and its performance has been investigated by comparing with the dual reciprocal boundary element method (DRM). The results of computations show that the accuracy of the proposed method is superior to that of the DRM regardless of the number of a boundary node, a boundary condition and a boundary shape. In addition, when the number of boundary nodes exceeds a certain limit, the calculation speed of the proposed method becomes almost equal to that of the DRM.

© 2010 The Japan Society of Plasma Science and Nuclear Fusion Research

Keywords: boundary element method, boundary integral equation, boundary value problem, numerical analysis

DOI: 10.1585/pfr.5.S2108

1. Introduction

As is well known, the boundary-element method (BEM) is a powerful method for solving a potential problem. It has been widely used in the field of the nuclear fusion science and has yielded excellent results. However, the BEM has the inherent demerit. Before executing a BEM code, a boundary must be divided into a set of elements.

In order to resolve the above demerit, Mukherjee *et al.* have proposed the boundary-node method (BNM) [1]. Since the BNM is one of the meshless approaches, a preparation of input data can be extremely simplified. However, in the conventional BNM, integration cells must be employed for evaluating the influence coefficients. In other words, a concept of elements partly remains in the BNM.

In the field of computer graphics, the novel method has been recently proposed for representing the object surface [2,3]. In the method, the object surface has been represented in terms of an implicit function. If the above method were applied to the BNM, the demerit of the BNM could be completely resolved.

The purpose of the present study is to reformulate the BNM without using any integration cells and to numerically investigate the performance of the proposed method.

2. Boundary-Node Method

2.1 Discretization

As a potential problem, we consider a 2D Poisson problem:

$$-\Delta u = \rho \quad \text{in } \Omega, \quad (1)$$

$$u = \bar{u} \quad \text{on } \Gamma_D, \quad (2)$$

$$q \equiv \partial u / \partial n = \bar{q} \quad \text{on } \Gamma_N, \quad (3)$$

where Ω denotes a region bounded by a simple closed curve $\partial\Omega$. The boundary $\partial\Omega$ consists of two curves, Γ_D and Γ_N , which satisfy the following relations: $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$. In addition, ρ , \bar{u} and \bar{q} are known functions in Ω , on Γ_D and on Γ_N , respectively. Moreover, \mathbf{n} denotes an outward unit normal on $\partial\Omega$.

When (1) is transformed to an equivalent boundary integral equation, the equation contains not only boundary integrals but also a domain integral. In order to remain only boundary integrals, we assume that $\rho(\mathbf{x})$ is approximated as

$$\rho(\mathbf{x}) = \sum_{l=1}^{N+K} \alpha_l f(|\mathbf{x} - \mathbf{z}_l|/R), \quad (4)$$

where $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N$ are nodes on $\partial\Omega$, whereas $\mathbf{z}_{N+1}, \mathbf{z}_{N+2}, \dots, \mathbf{z}_{N+K}$ are nodes in Ω . Throughout the present study, the node on $\partial\Omega$ is called the boundary node. Moreover, R and α_l 's are all constants. For the function $f(r)$, we adopt a compactly supported radial basis function [4].

Under the approximation, the equation obtained by substituting (4) into (1) is shown to be equivalent to the following boundary integral equation:

$$\oint_{\partial\Omega} \frac{\partial w^*(\mathbf{x}(s), \mathbf{y})}{\partial n} [u(\mathbf{x}(s)) - u(\mathbf{y})] ds - \oint_{\partial\Omega} w^*(\mathbf{x}(s), \mathbf{y}) q(\mathbf{x}(s)) ds$$

author's e-mail: saitoh@eng.u-hyogo.ac.jp

$$\begin{aligned}
 &= \sum_{l=1}^{N+K} \alpha_l \left\{ \oint_{\partial\Omega} \frac{\partial w^*(\mathbf{x}(s), \mathbf{y})}{\partial n} [\hat{u}_l(\mathbf{x}(s)) - \hat{u}_l(\mathbf{y})] ds \right. \\
 &\quad \left. - \oint_{\partial\Omega} w^*(\mathbf{x}(s), \mathbf{y}) \hat{q}_l(\mathbf{x}(s)) ds \right\}. \quad (5)
 \end{aligned}$$

Here, $w^*(\mathbf{x}, \mathbf{y}) \equiv -(1/2\pi) \log|\mathbf{x} - \mathbf{y}|$. Furthermore, \hat{u}_l is a particular solution of $-\Delta \hat{u}_l = f(|\mathbf{x} - \mathbf{z}_l|/R)$ and \hat{q}_l denotes its normal derivative. In addition, s indicates an arclength along $\partial\Omega$.

Let us discretize (5) and its associated boundary conditions. To this end, the shape functions $\Phi_j(s)$'s are assigned to the boundary nodes, \mathbf{z}_i 's. As a result, u , q , \hat{u}_l and \hat{q}_l are assumed as

$$u(\mathbf{x}(s)) = \sum_{j=1}^N \Phi_j(s) u_j^*, \quad (6)$$

$$q(\mathbf{x}(s)) = \sum_{j=1}^N \Phi_j(s) q_j^*, \quad (7)$$

$$\hat{u}_l(\mathbf{x}(s)) = \sum_{j=1}^N \Phi_j(s) \hat{u}_{jl}, \quad (8)$$

$$\hat{q}_l(\mathbf{x}(s)) = \sum_{j=1}^N \Phi_j(s) \hat{q}_{jl}, \quad (9)$$

where u_j^* , q_j^* , \hat{u}_{jl} and \hat{q}_{jl} are constants.

Under the above assumptions, (5) and its associated boundary conditions can be discretized to a linear system [5]. Thus, the 2D Poisson problem is reduced to the problem in which the linear system is solved. By solving the linear system, we can determine \mathbf{u}^* and \mathbf{q}^* to get the distributions of u and q on $\partial\Omega$.

2.2 Shape function

As mentioned in 2.1, we must determine shape functions assigned to all boundary nodes. In the BNM, by using the Moving Least-Square approximation [3], the shape functions $\Phi_j(s)$ ($j = 1, 2, \dots, N$) are defined by

$$\Phi_j(s) = \mathbf{p}^T(s) \mathbf{A}^{-1}(s) \mathbf{c}_j(s). \quad (10)$$

Here, $\mathbf{A}(s)$ and $\mathbf{c}_j(s)$ are given by

$$\mathbf{A}(s) = \sum_{j=1}^N w(d(s, s_j)/R_j) \mathbf{p}(s_j) \mathbf{p}(s_j)^T, \quad (11)$$

$$\mathbf{c}_j(s) = w(d(s, s_j)/R_j) \mathbf{p}(s_j), \quad (12)$$

where $d(s, s')$ and R_j denote a distance between $\mathbf{x}(s)$ and $\mathbf{x}(s')$ along $\partial\Omega$ and a support radius, respectively, and $w(r)$ is defined by

$$w(r) = \begin{cases} 1 - 6r^2 + 8r^3 - 3r^4; & r \leq 1, \\ 0; & r > 1. \end{cases} \quad (13)$$

For the m -dimensional vector $\mathbf{p}(s)$, we adopt the following type:

$$\mathbf{p}(s)^T = \begin{cases} 1; & m = 1, \\ [1, s]; & m = 2, \\ [1, s, s^2]; & m = 3. \end{cases} \quad (14)$$

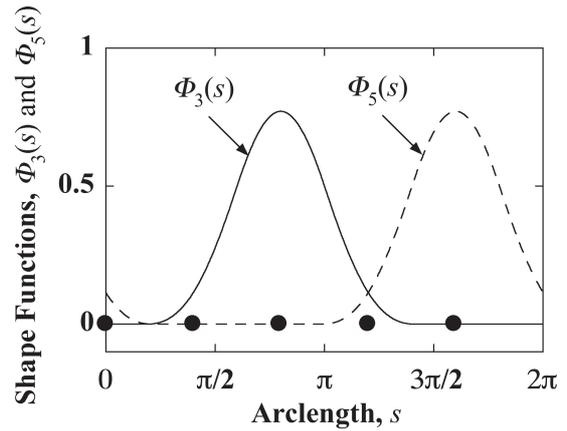


Fig. 1 The behavior of shape functions, $\Phi_3(s)$ and $\Phi_5(s)$. Here, the parameters are fixed as follows: $m = 2$ and $R_j = 2$. The symbol \bullet indicates the boundary node.

As an typical example, we compute shape functions, $\Phi_3(s)$ and $\Phi_5(s)$, from five boundary nodes placed on the boundary of a unit circle and its behavior is shown in Fig. 1. We see from this figure that both $\Phi_3(s)$ and $\Phi_5(s)$ are a smooth function with a period of 2π .

3. Extended Boundary-Node Method

In the conventional BNM, the boundary $\partial\Omega$ must be divided into a set of cells to evaluate the following contour integrals:

$$g_{ij} = \oint_{\partial\Omega} w^*(\mathbf{x}(s), \mathbf{x}(s_i)) \Phi_j(s) ds, \quad (15)$$

$$h_{ij} = \oint_{\partial\Omega} \frac{\partial w^*(\mathbf{x}(s), \mathbf{x}(s_i))}{\partial n} [\Phi_j(s) - \Phi_j(s_i)] ds. \quad (16)$$

In this sense, a concept of elements cannot be completely removed. In the present study, integrals are directly calculated by use of the vector equation of $\partial\Omega$. In this section, we propose the numerical method for determining the vector equation of $\partial\Omega$.

First, the implicit-function representation $g(\mathbf{x}) = 0$ is determined for the curve that passes through all boundary nodes. Next, we numerically solve the following ordinary differential equation:

$$\frac{d\mathbf{x}}{ds} = \mathbf{R}\left(\frac{\pi}{2}\right) \cdot \frac{\nabla g}{|\nabla g|}, \quad (17)$$

where $\mathbf{R}(\theta)$ denotes a tensor representing a rotation through an angle θ . Apparently, the analytic solution of (17) gives the vector equation $\mathbf{x} = \mathbf{x}(s)$ of the implicit-function representation $g(\mathbf{x}) = 0$. However, if the Runge-Kutta method is applied to (17), a large cost is necessary to obtain a high-precision solution of the vector equation.

In order to resolve the above difficulty, we propose the novel algorithm in which $\mathbf{x}^{(n+1)}$ is calculated from $\mathbf{x}^{(n)}$ by use of the following three steps (see Fig. 2).

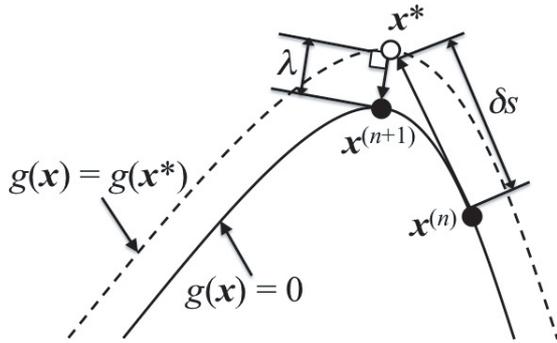


Fig. 2 The schematic view of the algorithm for solving (17).

- (i) An approximate solution of \mathbf{x}^* at the $(n + 1)$ th step is corrected by

$$\mathbf{x}^* = \mathbf{x}^{(n)} + R \left(\frac{\pi}{2} \right) \cdot \left[\frac{\nabla g}{|\nabla g|} \right]_{\mathbf{x}^{(n)}} \delta s. \quad (18)$$

Here, δs is a constant.

- (ii) For the purpose of determining an intersection of the straight line $\mathbf{x} = \mathbf{x}^* + \lambda(\nabla g)_{\mathbf{x}^*}$ and the curve $g(\mathbf{x}) = 0$, the nonlinear equation $G(\lambda) \equiv g(\mathbf{x}^* + \lambda(\nabla g)_{\mathbf{x}^*}) = 0$ is solved by using the Newton method.
- (iii) The numerical solution $\mathbf{x}^{(n+1)}$ is determined by $\mathbf{x}^{(n+1)} = \mathbf{x}^* + \lambda(\nabla g)_{\mathbf{x}^*}$.

After P data points, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(P)}$, are obtained by using the above algorithm, these points are interpolated by using the cubic spline. As a result, the vector equation $\mathbf{x} = \mathbf{x}(s)$ can be numerically determined. By means of the vector equation, boundary integrals can be easily evaluated without any integration cells.

The above technique is incorporated into the BNM. Throughout the present study, the resulting method is called the eXtended Boundary-Node Method (X-BNM).

4. Numerical Results

In this section, the performance of the X-BNM is investigated by comparing with the dual-reciprocal boundary-element method (DRM). In this study, we apply both two methods to a simple Poisson problem. $\partial\Omega$ is assumed as

$$\partial\Omega = \left\{ x \in \mathbf{R}^2 \mid g(\mathbf{x}) \equiv [x - \Delta(y/2)^2]^2 + (y/2)^2 - 1 = 0 \right\}, \quad (19)$$

where Δ is the triangularity. Furthermore, ρ, \bar{u} and \bar{q} are chosen so that the analytic solution of (1)-(3) is given by

$$u = 3e^{-(x^2+y^2)} - \cosh x \sin y + \cos x \sinh y. \quad (20)$$

In addition, Γ_N is a curve measured from \mathbf{z}_1 along $\partial\Omega$ and its length denotes s_N . Γ_D is defined as $\Gamma_D = \partial\Omega - \Gamma_N$. Throughout the present study, the support radius R_j is given by

$$R_j = \gamma \min \left(d(s_{\text{mod}(j+1,N)}, s_j), d(s_{\text{mod}(j-1,N)}, s_j) \right). \quad (21)$$

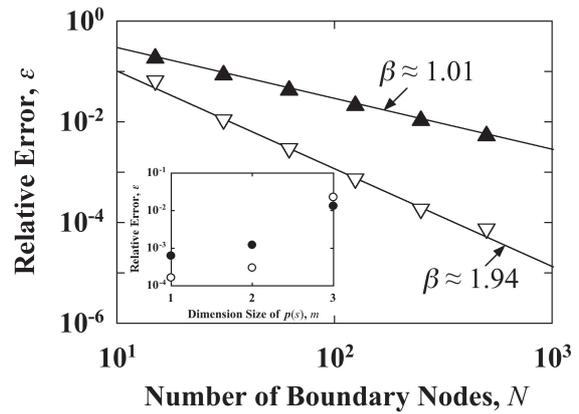


Fig. 3 Dependence of the relative error ϵ on the number N of the boundary nodes for the case with $\Delta = 0$. Here, the symbols, \blacktriangle and ∇ indicate the values for the DRM, the X-BNM ($m = 1$), respectively. The inset shows the dependence of ϵ on m for the same case. Here, \bullet : $N = 128$ and \circ : $N = 256$.

Here, γ is a constant. The number K of nodes in Ω , which satisfies $K \propto N^2$ is determined. In addition, the parameters are fixed as follows: $R = 1.5, \gamma = 1.6$.

As the measure of the accuracy of the numerical solution, we adopt the relative error defined by

$$\epsilon = \frac{\sqrt{\|u_A - u_N\|_2^2 + \|q_A - q_N\|_2^2}}{\sqrt{\|u_A\|_2^2 + \|q_A\|_2^2}}, \quad (22)$$

where subscript notations, A and N, are analytic and numerical solutions, respectively, and $\| \cdot \|_2$ denotes an Euclidean norm.

Let us first investigate the accuracy of the X-BNM and the DRM for the Dirichlet problem. The relative errors are calculated as a function of N and are depicted in Fig. 3. We see from this figure that the relative errors are almost proportional to $N^{-\beta}$ among all methods and that the power indices β 's satisfy $\beta \approx 1.01$ and $\beta \approx 1.94$ for the DRM and the X-BNM, respectively. The above results indicate that the accuracy of the X-BNM is much higher than that of the DRM. In the inset of Fig. 3, we see from this figure that the value of m does not affect the accuracy of the numerical solution. This result means that the accuracy of numerical solution does not depend on the selection of m . This is mainly because the support radius is much shorter than the arclength along the boundary. From the viewpoint of a calculation cost, m is fixed as $m = 1$ in the following numerical experiment.

We investigate the influence of the boundary shape on the accuracy of the solution for the Dirichlet problem. The relative errors are plotted as a function of Δ in Fig. 4. The accuracies of the DRM and the X-BNM monotonously increase with an increase in Δ . In addition, the accuracy of the X-BNM is much higher than that of the DRM regardless of Δ . From these results, in the X-BNM, we can ob-

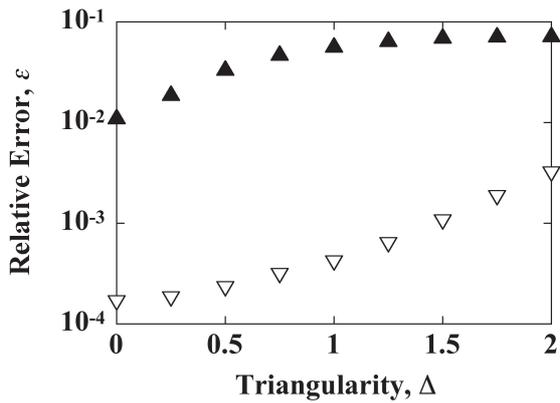


Fig. 4 Dependence of the relative error ϵ on the triangularity Δ for the Dirichlet problem. Here, the parameter is fixed as $N = 256$. The symbols, \blacktriangle and ∇ indicate the values for the DRM and the X-BNM, respectively.

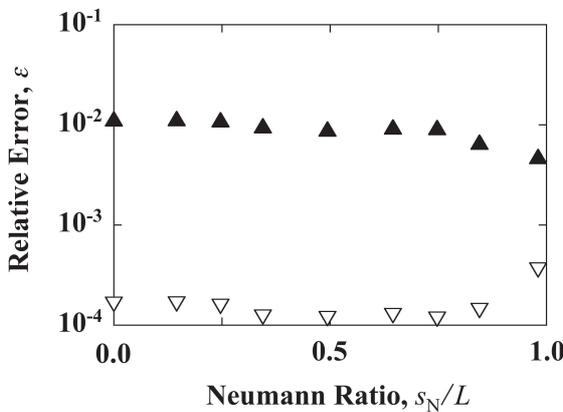


Fig. 5 Dependence of the relative error ϵ on the Neumann ratio s_N/L . Here, the parameters are fixed as $\Delta = 0$ and $N = 256$. The symbols, \blacktriangle and ∇ indicate the values for the DRM and the X-BNM, respectively.

tain the high-precision numerical solution even when the boundary shape is extremely concave.

Next, we investigate the influence of the kind of the boundary condition on the accuracy of the solution. In Fig. 5, we show the dependence of the relative error ϵ on the Neumann ratio s_N/L . Here, L denotes a length of $\partial\Omega$. This figure indicates that the relative error of the DRM decreases with an increase in the Neumann ratio. On the other hand, the accuracy of X-BNM is almost constant regardless of the Neumann ratio except for $s_N/L \approx 1$. Moreover, the accuracy of the X-BNM is much higher than that of the DRM. The above result shows the accuracy of the X-BNM is superior to that of the DRM regardless of the boundary condition.

Finally, let us compare the calculation speed of the X-BNM with that of the DRM. To this end, the ratio τ_X/τ_D of CPU times is calculated as a function of N and is depicted in Fig. 6. Here, τ_X and τ_D denote the CPU times for the X-BNM and that for the DNM, respectively. We see from this

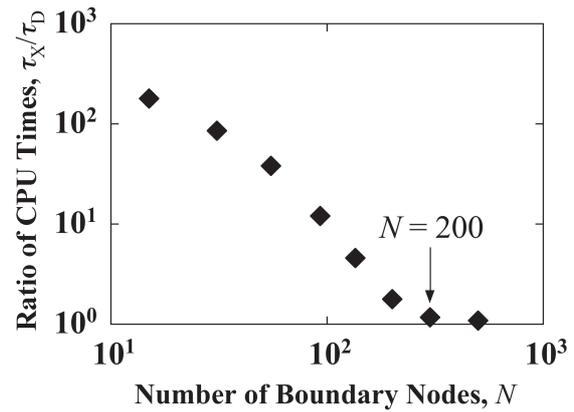


Fig. 6 Dependence of the ratio τ_X/τ_D on the number N of the boundary nodes for the case with $\Delta = 0$.

figure that τ_X/τ_D decreases monotonously with an increase in N until $\tau_X/\tau_D \approx 1$ is satisfied for $N \gtrsim 200$. This tendency shows that the speed of the X-BNM is almost equal to that of the DRM for $N \gtrsim 200$.

From the above results, we might conclude that the X-BNM is a powerful method for solving the potential problem.

5. Conclusion

By completely removing a concept of elements from the BNM, we have reformated the X-BNM. In addition, we have investigated its performance by comparing with the DRM. Conclusions obtained in the present study are summarized as follows.

- 1) For both the Dirichlet and the mixed-type problems, the accuracy of the X-BNM is much higher than that of the DRM.
- 2) Even when the boundary shape is concave, the X-BNM shows a much higher accuracy than the DRM.
- 3) When the number of boundary nodes exceeds a certain limit, the speed of the X-BNM becomes almost equal to that of the DRM. In this study, the value has become $N \approx 200$.

Acknowledgment

This work was partially supported by KAKENHI (No. 20760059) and the NIFS Collaboration Research Program (NIFS09KDBN003).

- [1] Y. X. Mukherjee and S. Mukherjee, *Int. J. Numer. Methods Eng.* **40**, 797 (1997).
- [2] G. Turk and J. F. O'Brien, *Proc. ACM SIGGRAPH 99*, 335 (1999).
- [3] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk and H. P. Seidel, *ACM Trans. Graph.* **40**, 463 (2003).
- [4] C. S. Chen, C. A. Brebbia and H. Power, *Commun. Numer. Methods Eng.* **15**, 137 (1999).
- [5] A. Saitoh, S. Nakata, S. Tanaka and A. Kamitani, *Information* **12**, 973 (2009).