Development of Modified Extended Boundary-Node Method -New Approach for Determining Data Points-

Kenta Miyashita[1], Ayumu Saitoh[1], Taku Itoh[2], Atsushi Kamitani[3],
Naotake Kamiura[1], and Nobuyuki Matsui[1]

[1]Graduate School of Enginnering, University of Hyogo, Japan
[2]Faculty of Science and Technology, Seikei University, Japan
[3]Graduate School of Science and Enginnering, Yamagata University, Japan

1 Introduction

The Boundary-Node Method (BNM) [1] has been so far proposed for solving the boundary-value problem of a partial differential equation and has yielded excellent results in the field of engineering [1, 2]. It is a primary merit of the BNM that elements of a geometrical structure are no longer necessary. However, the BNM has the following difficulty: a boundary must be divided into a set of integration cells to calculate contour integrals. If the BNM were extended to the method without integration cells, the demerit of the BNM can be resolved completely.

For the purpose of resolving the above demerit, Saitoh *et al.* proposed the eXtended BNM (X-BNM) and investigated its performance numerically [3]. As a result, they showed that the accuracy of the X-BNM is much higher than that of the standard BNM for the Poisson problem. However, when the boundary shape is strongly concave, the accuracy of the X-BNM is drastically degraded.

The purpose of the present study is to modify the X-BNM for suppressing the accuracy degradation and to compare the performance of the modified X-BNM with that of the conventional one.

2 Extended Boundary-Node Method

Throughout the present study, we adopt a two-dimensional (2-D) Laplace problem:

$$-\Delta u = 0, \qquad \text{in } \Omega \tag{1}$$

$$u = \bar{u}, \qquad \text{on } \Gamma_{\mathrm{D}} \tag{2}$$

$$q \equiv \partial u/\partial n = \bar{q}, \qquad \text{on } \Gamma_{\mathrm{N}} \tag{3}$$

where Ω denotes a domain bounded by a simple closed curve $\partial\Omega$. Here, $\partial\Omega$ consists of two parts, Γ_{D} and Γ_{N}, which satisfy the following relations: $\Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}} = \partial\Omega$ and $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \phi$. In addition, $\bar{u}$ and $\bar{q}$ denote known functions on Γ_{D} and on Γ_{N}, respectively, and $\boldsymbol{n}$ indicates an outward unit normal on $\partial\Omega$.

As is well known, (1) can be easily transformed to the following boundary integral equation:

$$\oint_{\partial\Omega} \frac{\partial w^*(\boldsymbol{x}(s), \boldsymbol{y})}{\partial n} \left[u(\boldsymbol{x}(s)) - u(\boldsymbol{y}) \right] ds - \oint_{\partial\Omega} w^*(\boldsymbol{x}(s), \boldsymbol{y})\, q(\boldsymbol{x}(s))\, ds = 0, \tag{4}$$

where $w^*(\boldsymbol{x}, \boldsymbol{y}) \equiv -\log|\boldsymbol{x} - \boldsymbol{y}|/2\pi$. Moreover, s indicates an arclength along $\partial\Omega$.

In order to discretize (4) and its associated boundary conditions, N boundary nodes, $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N$, are placed on $\partial\Omega$. Furthermore, shape functions, $\Phi_1, \Phi_2, \cdots, \Phi_N$, are defined by means of the Moving Least-Squares (MLS) approximation [3]. In addition, u and q are assumed as

$$u\left(\boldsymbol{x}(s)\right) = \sum_{j=1}^{N} \Phi_j(s)\, u_j^*, \quad q\left(\boldsymbol{x}(s)\right) = \sum_{j=1}^{N} \Phi_j(s)\, q_j^*. \tag{5}$$

where u_j^* and q_j^* $(j = 1, 2, \cdots, N)$ are all constants.

Under the above assumptions, (4) and its associated boundary conditions can be discretized to a linear system. In this way, the 2-D Laplace problem is reduced to the problem in which the linear system is solved.

In the X-BNM, contour integrals of (4) are directly calculated by use of the vector equation of $\partial\Omega$. To this end, the vector equation is determined as follows. First, the implicit-function representation $f(\boldsymbol{x}) = 0$ is determined for the curve passing through all boundary nodes. Next, we numerically solve the following ordinary differential equation:

$$\frac{d\boldsymbol{x}}{ds} = \boldsymbol{R}\left(\frac{\pi}{2}\right) \cdot \frac{\nabla f}{|\nabla f|}, \tag{6}$$

where $\boldsymbol{R}(\theta)$ denotes a tensor representing a rotation through an angle θ. Finally, the resulting P data points, $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(P)}$, are interpolated with the cubic spline to get the vector equation $\boldsymbol{x} = \boldsymbol{g}(s)$. In the next section, we explain the numerical solution of (6) in detail.

3 New Approch for Determing Data Points

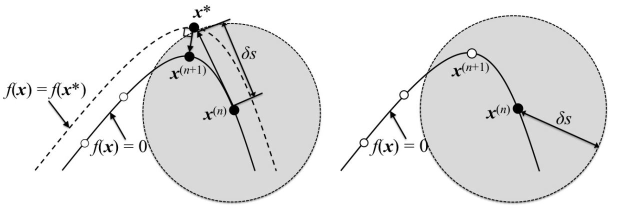
By solving (6), we can determine P data points. However, even if the higher-order Runge-Kutta method is applied to (6), the numerical solution does not always satisfy $f(\boldsymbol{x}) = 0$. So as to resolve this problem, Saitoh *et al.* proposed the algorithm in which $\boldsymbol{x}^{(n+1)}$ is calculated from $\boldsymbol{x}^{(n)}$ by use of the following three steps (see Fig. 1(a)):

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

$$\boldsymbol{x}^* = \boldsymbol{x}^{(n)} + \boldsymbol{R}\left(\frac{\pi}{2}\right) \cdot \left[\frac{\nabla f}{|\nabla f|}\right]_{\boldsymbol{x}^{(n)}} \delta s \tag{7}$$

Here, δs is a constant.

(ii) In order to calculate an intersection of the straight line $\boldsymbol{x} = \boldsymbol{x}^* + \lambda(\nabla f)_{\boldsymbol{x}^*}$ and the curve $f(\boldsymbol{x}) = 0$, the nonlinear equation $G(\lambda) \equiv f\left(\boldsymbol{x}^* + \lambda(\nabla f)_{\boldsymbol{x}^*}\right) = 0$ is solved by using the Newton method.



(a) (b)

Fig. 1. Schematic view of the approach for determining data points for the case where a boundary node (a) is not contained and (b) is contained in the δs-neighborhood of $\boldsymbol{x}^{(n)}$. Here, the symbol, ∘, denotes boundary nodes.

(iii) The numerical solution $\boldsymbol{x}^{(n+1)}$ is determined by $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^* + \lambda(\nabla f)_{\boldsymbol{x}^*}$.

By using the above approach, although the curve $\boldsymbol{x} = \boldsymbol{g}(s)$ passes through all data points, boundary nodes do not always locate on the curve. This is because any boundary nodes may not be contained in data points to be interpolated. In order to overcome the above problem, we use the following approach for determining data points: if a boundary node exists in the δs-neighborhood of $\boldsymbol{x}^{(n)}$, it is employed as $\boldsymbol{x}^{(n+1)}$ instead of solving (6) numerically (see Fig. 1(b)). Throughout the present study, the X-BNM with the above correction is called the modified X-BNM.

4 Numerical Results

In this section, we compare the performance of the modified X-BNM with the conventional one. As an example problem, we adopt the 2-D Laplace problem over $\Omega \equiv \left\{ (x, y) \,\middle|\, \left[x - \Delta(y/2)^2\right]^2 + (y/2)^2 < 1 \right\}$ with the Dirichlet condition: $u = \cosh \pi x \sin \pi y + \sinh \pi x \cos \pi y$ on $\partial\Omega$.

Let us first investigate both the accuracy of the modified X-BNM and that of the conventional one. The relative errors are calculated as a function of N and are plotted in Fig. 2. The relative error of the modified X-BNM is diminished in almost proportion to $N^{-1.35}$. On the other hand, the relative error of the conventional X-BNM decreases with an increase in N for the case with $N \lesssim 50$, whereas it drastically increases with N for $N \gtrsim 50$. In addition, the accuracy of the modified X-BNM is almost equal to that of the conventional one for the case with $N \lesssim 50$.

Next, we investigate the influence of the boundary shape on the accuracy of the numerical solution. To this end, the relative errors are calculated as functions of the triangularity Δ and are plotted in Fig. 3(a). This figure indicates that the accuracy of the modified X-BNM is much higher than that of the conventional one for the case with $\Delta \gtrsim 1$.

Finally, we compare the speed of the modified X-BNM with that of the conventional one. The ratio $\tau_{\mathrm{M}}/\tau_{\mathrm{C}}$ of the CPU time is calculated as a function of Δ and is depicted in Fig. 3(b). Here, τ_{M} and τ_{C} are the CPU time required for the modified X-BNM and that for the conventional one, respectively. We see from this figure that the speed of the modified X-BNM is almost equal to that of the conventional one.

From these results, we can conclude that the performance of the modified X-BNM is superior to that of the conventional one.

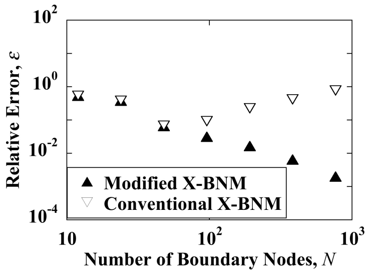
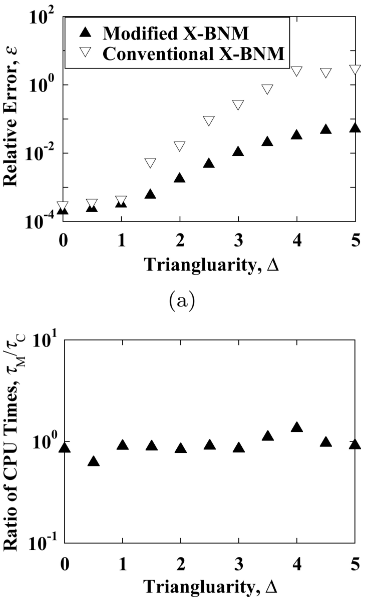


Fig. 2. Dependence of the relative error ε on the number N of boundary nodes ($\Delta = 3$).



(a)

(b)

Fig. 3. Dependence of (a) the relative error ε and (b) the ratio $\tau_{\mathrm{M}}/\tau_{\mathrm{C}}$ of CPU times on the triangularity Δ ($N = 256$).

5 Conclusion

We have modified the X-BNM so that the accuracy degradation may not be caused by a boundary shape and, have numerically investigated the performance of the modified X-BNM by comparing with the conventional X-BNM. Conclusions obtained in this paper are summarized as follows.

1. the accuracy of the modified X-BNM is much higher than that of the conventional one, when the boundary shape is strongly concave.
2. the speed of the modified X-BNM is almost equal to that of the conventional one.

References

[1] Y.X. Mukherjee and S. Mukherjee, "The boundary node method for potential problems," *Int. J. Numer. Methods Eng.*, vol. 40, pp. 797-815, 1997.

[2] V.S. Kothnur, S. Mukherjee and Y.X. Mukherjee, "Two-dimensional linear elasticity by the boundary node method," *Int. J. Solids Struc.*, vol. 36, pp. 1129-1147, 1998.

[3] A. Saitoh, N. Matsui, T. Itoh and A. Kamitani, "Development of 2-D meshless approaches without using integration cells," *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1222-1225, 2011.