Acceleration Technique for Extended Boundary Node Method

Ayumu Saitoh, Taku Itoh, Nobuyuki Matsui, Atsushi Kamitani
Graduate School of Engineering, University of Hyogo,
2167, Shosha, Himeji, Hoagy 671-2280, Japan
saitoh@eng.u-hyogo.ac.jp

***Abstract*—The acceleration technique for extended boundary-node method (X-BNM) has been proposed and its performance has been numerically investigated. The results of computations show the X-BNM with the radial point interpolation method is always faster than the X-BNM with the moving least-squares approximation regardless of the number of boundary nodes.**

***Index Terms*—Boundary value problems, Partial differential equations, Green's function methods, Integral equations, Least squares approximation**

I. Introduction

Several boundary-type meshless methods [1]-[5] have been proposed for solving the boundary-value problem of the partial differential equation and have yielded excellent results in the fields of science and engineering. Since the boundary elements of a geometrical structure are no longer necessary among all boundary-type meshless methods, the input data preparation can be extremely simplified.

In the Boundary-Node Method (BNM) [1] is one of the boundary-type meshless methods. In spite of a high usefulness, the BNM is plagued by two demerits. First of all, integration cells must be adopted for calculating matrix elements. This means that a concept of elements partly remains in the BNM. Second, a shape function lacks the Kronecker's delta function property. This causes the following problem: the number of unknowns is equal to twice the number of boundary nodes.

In our previous work, the BNM has been reformulated without using integration cells and its performance has been investigated numerically [5], [6]. This method is called the eXtended BNM (X-BNM). This work shows that the accuracy of the X-BNM is much higher than that of the dual reciprocity boundary element method. In addition, we have modified the X-BNM for improving the accuracy degradation due to a complex boundary shape [7]. In this way, the first demerit of the BNM can be completely removed by using the X-BNM. However, the second demerit of the BNM has been included in the X-BNM yet.

Recently, the Radial Point Interpolation Method (RPIM) [8] which is one of interpolation methods has been proposed. The RPIM has an advantage that the shape function possesses the Kronecker's delta function property. Furthermore, the modified RPIM has been also proposed for accelerating the computation of the shape functions [9], [10]. If the RPIM were applied to the X-BNM, the second demerit of the BNM could be removed from the X-BNM completely.

The purpose of the present study is to apply the RPIM to the X-BNM and to investigate the performance of the proposed method numerically.

II. Shape Function

In the X-BNM, we assume that a solution and its normal derivative can be expanded by using shape functions. Hence, we must define shape functions which are assigned to boundary nodes. In this section we summarize two methods to generate shape functions from N boundary nodes.

The Moving Least-Squares Approximation (MLSA) is well known as one of interpolation methods. The approximate function $f^{\mathrm{h}}(s)$ of a function $f(s)$ can be written by

$$f^{\mathrm{h}}(s) = \boldsymbol{p}^T(s)\,\boldsymbol{a}(s), \tag{1}$$

where s indicates an arclength along the boundary and $\boldsymbol{p}(s) = [1, s, \cdots, s^{(m-1)}]^T$. $\boldsymbol{a}(s)$ can be determined by minimizing the following functional:

$$J[\boldsymbol{a}(s)] = \sum_{i=1}^{N} w_i(s)[\boldsymbol{p}^T(s_i)\,\boldsymbol{a}(s) - f(s_i)],$$

where $w_i(s)$ and s_i denote a weight function and an arclength to the ith boundary node, respectively.

After minimizing $J[\boldsymbol{a}(s)]$, we can obtain

$$\Phi_i(s) = \boldsymbol{p}^T(s)\,A^{-1}(s)\,\boldsymbol{b}_i(s), \tag{2}$$

where $A(s)$ and $\boldsymbol{b}_i(s)$ are defined by

$$A(s) = \sum_{i=1}^{N} w_i(s)\,\boldsymbol{p}(s_i)\,\boldsymbol{p}^T(s_i),$$
$$\boldsymbol{b}_i(s) = w_i(s)\,\boldsymbol{p}(s_i).$$

Note that the shape function $\Phi_i(s)$ in (2) does not satisfy the following relation: $\Phi_i(s_j) = \delta_{i,j}$ where $\delta_{i,j}$ is the Kronecker's delta. Therefore, the number of unknowns equals twice as much as the number of boundary nodes.

In order to resolve the demerit of the MLSA, the RPIM has been proposed. In the RPIM, a combination of the radial basis function $r_i(s)$ and the monomial basis function $p_i(s)$ is used for the interpolation of the field distribution around the ith boundary node. The approximate function $f^{\mathrm{h}}(s)$ can be rewritten by

$$f^{\mathrm{h}}(s) = \boldsymbol{r}^T(s)\,\boldsymbol{a} + \boldsymbol{p}^T(s)\,\boldsymbol{b},$$

where $\boldsymbol{r}(s) = [r_1(s), r_2(s), \cdots, r_N(s)]^T$.

In order to determine $\boldsymbol{a} \in \boldsymbol{R}^N$ and $\boldsymbol{b} \in \boldsymbol{R}^m$, we enforce the interpolation to satisfy the given value at nodes as

$$\begin{bmatrix} R_0 & P_0 \\ P_0^T & \mathrm{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix}. \tag{3}$$

Here, R_0, P_0 and $\boldsymbol{f}$ are defined by

$$R_0 = \sum_{i=1}^{N} \sum_{j=1}^{N} r_i(s_j)\, \boldsymbol{e}_i\, \boldsymbol{e}_j^T,$$

$$P_0 = \sum_{i=1}^{N} \sum_{j=1}^{m} s_i^{(m-1)}\, \boldsymbol{e}_i\, \boldsymbol{e}_j^{*T},$$

$$\boldsymbol{f} = \sum_{i=1}^{N} f(s_i)\, \boldsymbol{e}_i,$$

where $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N\}$ and $\{\boldsymbol{e}_1^*, \boldsymbol{e}_2^*, \cdots, \boldsymbol{e}_m^*\}$ are the orthonormal system of the N-dimensional vector space and that of the m-dimensional vector space, respectively.

By solving (3), we can get

$$\boldsymbol{\Phi}(s) = \left[\boldsymbol{r}^T(s), \boldsymbol{p}^T(s)\right] \begin{bmatrix} R_0 & P_0 \\ P_0^T & \mathrm{O} \end{bmatrix}^{-1}. \tag{4}$$

where $\boldsymbol{\Phi}(s) = [\Phi_1(s), \Phi_2(s), \cdots, \Phi_N(s)]^T$. Since shape function $\Phi_i(s)$ exactly fulfills the following relation: $\Phi_i(s_j) = \delta_{i,j}$, the number of unknowns is equal to the number of boundary nodes.

In this way, the demerit of the MLSA is removed completely. Throughout the present study, the X-BNM with the MLSA and the X-BNM with RPIM are referred to as the X-BNM(MLSA) and the X-BNM(RPIM), respectively.

III. Numerical Results

In this section, we investigate the performance of the X-BNM(MLSA) and the X-BNM(RPIM). As an example problem, we adopt the 2-D Laplace problem over $\Omega \equiv \left\{x^2 + (y/2)^2 < 1\right\}$ with the boundary condition: $-\cosh \pi x \sin \pi y + \sinh \pi x \cos \pi y$ on the boundary. All numerical experiments are executed on an Intel Core 2 Duo, 1.86 GHz processor.

The 10 points Gaussian quadrature is employed as the integration method and the number m of order is fixed as $m = 2$. In addition, the ith support radius R_i is defined by $R_i = \gamma \min\left(\left|s_{\bmod (i+1, N)} - s_i\right|, \left|s_{\bmod (i-1, N)} - s_i\right|\right)$ where γ is fixed as $\gamma = 1.47$.

In the MLSA, the weight function is given by

$$w_i(s) = \omega\left(|s - s_i| / R_i\right),$$
$$\omega(r) = H(1 - r)(1 - 6r^2 + 8r^3 - 3r^4),$$

where $H(x)$ denotes the Heaviside step function. In the RPIM, the radial basis function is given by

$$r_i(s) = \rho\left(|s - s_i| / R_i\right),$$
$$\rho(r) = H(1 - r)(1 - r)^3(3r + 1).$$

Table I
Computational Cost for X-BNM.

(a) $N = 256$

	CPU time [s]		
Methods	Matrix in Assembly	Solution of Linear System	Total
X-BNM(MLSA)	2.23×10^{-1}	8.37×10^{-1}	1.06×10^{0}
X-BNM(RPIM)	4.98×10^{-1}	5.26×10^{-2}	5.51×10^{-1}

(b) $N = 512$

	CPU time [s]		
Methods	Matrix in Assembly	Solution of Linear System	Total
X-BNM(MLSA)	7.18×10^{-1}	7.90×10^{0}	8.62×10^{0}
X-BNM(RPIM)	7.49×10^{0}	8.41×10^{-1}	8.33×10^{0}

(c) $N = 1024$

	CPU time [s]		
Methods	Matrix in Assembly	Solution of Linear System	Total
X-BNM(MLSA)	2.53×10^{0}	1.46×10^{2}	1.49×10^{2}
X-BNM(RPIM)	1.08×10^{2}	8.40×10^{0}	1.16×10^{2}

Let us compare the speed of the X-BNM(RPIM) with that of the X-BNM(MLSA). The CPU time of the X-BNM(RPIM) and that of the X-BNM(MLSA) are shown in Tables I(a)-I(c). The CPU time of the X-BNM(RPIM) required for the matrix assembly increases drastically. Nevertheless, the X-BNM(RPIM) is always faster than the X-BNM(MLSA) regardless of the value of N.

From these results, we can conclude that the RPIM is useful to accelerate the X-BNM.

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, Mar. 1997.

[2] T. Zhu, J.-D. Zhang and S.N. Atluri, "A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach," *Comput. Mech.*, vol. 41, pp. 223-235, Apr. 1998.

[3] T. Zhu, "A new meshless regular local boundary integral equation (MRLBIE) approach," *Int. J. Numer. Methods Eng.*, vol. 46, pp. 1237-1252, Nov. 1999.

[4] J. Zhang, Z. Yao and H. Li, "A hybrid boundary node method," *Int. J. Numer. Methods Eng.*, vol. 53, pp. 751-763, Nov. 2001.

[5] A. Saitoh, S. Nakata, S. Tanaka and A. Kamitani, "Development of cell-independent dual-reciprocal boundary-node method," *Information*, vol. 12, no. 5, pp. 973-984, Sep. 2009 [in Japanese].

[6] 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, May 2011.

[7] A. Saitoh, K. Miyashita, T. Itoh, A. Kamitani, T. Isokawa, N. Kamiura and N. Matsui, "Accuracy improvement of extended boundary-node method," *Proceedings of The 15th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC2012)*, p. 247, Nov. 2012.

[8] J.G.Wang and G.R. Liu, "A point interpolation meshless method based on radial basis functions," *Int. J. Numer. Meth. Eng.*, vol. 54, no. 11, pp. 1623-1648, Aug. 2002.

[9] S. Nakata, "Meshless analysis with non-uniformly distributed nodes using hierarchical cell structure," *AIP Conf. Proc.*, vol. 936, pp. 378-381, Sep. 2007.

[10] S. Nakata, Y. Takeda, N. Fujita, S. Ikuno, "Parallel algorithm for meshfree radial point interpolation method on graphics hardware," *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1206-1209, May 2011.