2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

Speed Improvement of Extended Boundary Node Method by Using Adaptive Cross Approximation

Akihiro Nishimura[1)], Ayumu Saitoh[2)], Taku Itoh[3)], Nobuyuki Matsui[1)], and Atsushi Kamitani[2)]

1) Graduate School of Engineering, University of Hyogo
2167, Shosha, Himeji, Hyogo 671-2280, Japan
2) Graduate School of Science and Engineering, Yamagata University
4-3-16, Johnan, Yonezawa, Yamagata 992-8510, Japan
3) College of Industrial Technology, Nihon University
1-2-1, Izumi-cho, Narashino, Chiba 226-8575, Japan
Email: saitoh@yz.yamagata-u.ac.jp

Abstract—The extended boundary node method with the adaptive cross approximation has been proposed. As the future work, we will investigate its performance numerically.

1. Introduction

Recently, the boundary node method (BNM) [1], which is one of boundary-type meshless methods, has been proposed. As the feature of the BNM, a boundary does not need to be divided into a set of elements before executing the BNM code. In addition, a smooth numerical solution is obtained because the shape function is determined by using the moving least-squares approximation. However, the BNM has the demerit: integration cells must be used for calculating matrix elements.

In order to resolve the above demerit, the BNM has been reformulated without using integration cells. Throughout the present study, the method is called the extended BNM (X-BNM) [2]. The results of computations have shown that the accuracy of the X-BNM is much higher than that of the dual reciprocity boundary element method (DRM) [3, 4].

In spite of a high usefulness of the X-BNM, the calculation speed for obtaining the numerical solution is extremely slow. This is because the coefficient matrix of the resulting linear system becomes asymmetric and dense. In effect, this means that the X-BNM is difficult to be applied to a large-scale simulation.

The purpose of the present study is to develop the X-BNM with the adaptive cross approximation (ACA) [5, 6] and to investigate the performance numerically by applying it to the Grad-Shafranov (G-S) equation.

2. X-BNM

2.1. Boundary Integral Equation

For simplicity, we consider a boundary-value problem of the G-S equation in the cylindrical coordinate (r, z):

$$-\hat{L}\psi = rg \qquad \text{in } \Omega, \tag{1}$$

$$\psi = \bar{\psi} \qquad \text{on } \partial\Omega, \tag{2}$$

where Ω denotes a domain bounded by a simple closed curve $\partial\Omega$ in the r-z plane. Furthermore, g and $\bar{\psi}$ are a known function in Ω and on $\partial\Omega$, respectively. In addition, $\hat{L}$ denotes the G-S operator defined by

$$\hat{L} \equiv \frac{\partial^2}{\partial z^2} + r\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}\right).$$

By following the standard manner of the DRM, we assume that g is approximated as

$$g(r, z) = \sum_{l=1}^{N+M} \alpha_l \, \bar{g}_l(r, z), \tag{3}$$

where N, M and α_l are the number of boundary nodes, the number of poles and the lth coefficient, respectively. Furthermore, $\bar{g}_l(r, z)$ is defined by

$$\bar{g}_l(r, z) = \frac{2}{c}\left(1 + \frac{r_l}{r} - 2\frac{d_l}{c}\right)e^{-\frac{d_l}{c}},$$

where d_l is defined by $d_l \equiv (r - r_l)^2 + (z - z_l)^2$ and (r_l, z_l) denotes the lth pole. Moreover, c is a constant.

By substituting (3) into the right-hand side of (1), the G-S equation is transformed to the equivalent boundary-only integral equation:

$$\begin{aligned} & c(\boldsymbol{y})\psi(\boldsymbol{y}) \\ & \quad - \oint_{\partial\Omega} \frac{1}{r}\left(w^* \frac{\partial\psi(\boldsymbol{x}(s))}{\partial n} - \frac{\partial w^*}{\partial n}\psi(\boldsymbol{x}(s))\right) ds \\ & = \sum_{l=1}^{L} \alpha_l \Bigg\{ c(\boldsymbol{y})\hat{\psi}_l(\boldsymbol{y}) \\ & \quad - \oint_{\partial\Omega} \frac{1}{r}\left(w^* \frac{\partial\hat{\psi}_l(\boldsymbol{x}(s))}{\partial n} - \frac{\partial w^*}{\partial n}\hat{\psi}_l(\boldsymbol{x}(s))\right) ds \Bigg\}. \end{aligned} \tag{4}$$

Here, w^* and $\partial w^*/\partial n$ denote the fundamental solution of $-\hat{L}\psi = r\delta(\boldsymbol{x}(s) - \boldsymbol{y})$ and its normal derivative, respectively. Furthermore, $\hat{\psi}_l$ and $\partial\hat{\psi}_l/\partial n$ are the particular solution of $-\hat{L}\hat{\psi}_l = f_l$ and its normal derivative, respectively. In addition, s indicates an arclength along $\partial\Omega$ and $c(\boldsymbol{y})$ is a shape coefficient defined by

$$c(\boldsymbol{y}) = -\oint_{\partial\Omega} \frac{1}{r}\frac{\partial w^*(\boldsymbol{x}(s), \boldsymbol{y})}{\partial n} ds.$$

It must be noted here that $c(\boldsymbol{y}) = 1$ is exactly satisfied for $\boldsymbol{y} \in \Omega$. This suggests that, for the case with $\boldsymbol{y} \in \Omega$, the solution $\psi(\boldsymbol{y})$ can be calculated from the solution ψ and its normal derivative $\partial\psi/\partial n$ on the boundary $\partial\Omega$. Therefore, we have only to obtain ψ and $\partial\psi/\partial n$ on $\partial\Omega$.

2.2. Discretization

For the purpose of obtaining ψ and $\partial\psi/\partial n$ on $\partial\Omega$, let us discretize (4) and (2). If N nodes are placed on $\partial\Omega$, RPIM shape functions ϕ_i's [7] can be easily determined. Furthermore, $\psi, \partial\psi/\partial n, \hat{\psi}_l$ and $\partial\hat{\psi}_l/\partial n$ are assumed as

$$\psi(\boldsymbol{x}(s)) = \sum_{i=1}^{N} \phi_i(s)\,\psi_i,$$

$$\frac{\partial\psi(\boldsymbol{x}(s))}{\partial n} = \sum_{i=1}^{N} \phi_i(s)\,q_i,$$

$$\hat{\psi}_l(\boldsymbol{x}(s)) = \sum_{i=1}^{N} \phi_i(s)\,\hat{\psi}_i^l,$$

$$\frac{\partial\hat{\psi}_l(\boldsymbol{x}(s))}{\partial n} = \sum_{i=1}^{N} \phi_i(s)\,\hat{q}_i^l,$$

where ψ_i and q_i are the solution on ith boundary node and its normal derivative, respectively. Furthermore, $\hat{\psi}_i^l$ and $\hat{q}_i^l$, are the lth particular solution on ith boundary node and its normal derivative, respectively.

Under the aforementioned assumptions, (4) and (2) are discretized as the following equation:

$$G\boldsymbol{q} = H\boldsymbol{u} - \left[H\hat{U} - G\hat{Q}\right]F^{-1}\boldsymbol{\alpha}. \tag{5}$$

Here, $\boldsymbol{u}$, $\boldsymbol{q}$ and $\boldsymbol{\alpha}$ are defined by

$$\boldsymbol{u} = \sum_{i=1}^{N} \psi_i\,\boldsymbol{e}_i,$$

$$\boldsymbol{q} = \sum_{i=1}^{N} q_i\,\boldsymbol{e}_i,$$

$$\boldsymbol{\alpha} = \sum_{i=1}^{M} \alpha_i\,\boldsymbol{e}_i^*,$$

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. In addition, H, G, $\hat{U}$, $\hat{Q}$ and F are given by

$$H = \sum_{i=1}^{N}\sum_{j=1}^{N}\left\{\oint_{\partial\Omega} \frac{\partial w^*(\boldsymbol{x}(s), \boldsymbol{x}_i)}{\partial n}\left[\phi_j(s) - \delta_{i,j}\right] ds\right\}\boldsymbol{e}_i\,\boldsymbol{e}_j^T, \tag{6}$$

$$G = \sum_{i=1}^{N}\sum_{j=1}^{N}\left\{\oint_{\partial\Omega} w^*(\boldsymbol{x}(s), \boldsymbol{x}_i)\phi_j(s)\,ds\right\}\boldsymbol{e}_i\,\boldsymbol{e}_j^T, \tag{7}$$

$$\hat{U} = \sum_{i=1}^{N}\sum_{j=1}^{M} \hat{\psi}_i^l\,\boldsymbol{e}_i\,\boldsymbol{e}_j^{*T},$$

$$\hat{Q} = \sum_{i=1}^{N}\sum_{j=1}^{M} \hat{q}_i^l\,\boldsymbol{e}_i\,\boldsymbol{e}_j^{*T},$$

$$F = \sum_{i=1}^{M}\sum_{j=1}^{M} f_j(\hat{\boldsymbol{x}}_i)\,\boldsymbol{e}_i^*\,\boldsymbol{e}_j^{*T},$$

where $\boldsymbol{x}_i$ and $\hat{\boldsymbol{x}}_i$ are the ith boundary node and ith poles, respectively. In this way, the boundary-value problem of the G-S equation is reduced to the problem in which (5) is solved.

2.3. Determination of Matrices H and G

The BNM needs integration cells for calculating contour integrals in (6) and (7). In contrast, in the X-BNM, contour integrals in (6) and (7) are directly calculated by use of a vector equation of the boundary $\partial\Omega$.

The vector equation of $\partial\Omega$ is determined by means of the following three steps. First, we determine the implicit-function representation $f(\boldsymbol{x}) = 0$ [8,9] for the curve passing through all boundary nodes. Next, we numerically solve the following ordinary differential equation:

$$\frac{d\boldsymbol{x}}{ds} = \boldsymbol{R}\left(\frac{\pi}{2}\right)\cdot\frac{\nabla f}{|\nabla f|}, \tag{8}$$

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. In this way, we can get the vector equation $\boldsymbol{x} = \boldsymbol{g}(s)$. Since the vector equation of $\partial\Omega$ is represented as a function of s, we can easily calculate contour integrals in (6) and (7) without using integration cells in the X-BNM.

Even if the higher-order Runge-Kutta method is applied to (8), the numerical solution does not always satisfy $f(\boldsymbol{x}) = 0$. In order to resolve the above problem, Saitoh *et al.* proposed the algorithm in which $\boldsymbol{x}^{(n+1)}$ is calculated from $\boldsymbol{x}^{(n)}$ by use of the following four steps:

Step1 If the boundary node exists in the δs-neighborhood of $\boldsymbol{x}^{(n)}$, it is employed as $\boldsymbol{x}^{(n+1)}$ instead of the execution of Steps 2-4. Here, δs is a constant.

Step2 An approximate solution of $\boldsymbol{x}^*$ at the $(n+1)$th step is modified by

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

Step3 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 $f(\boldsymbol{x}^* + \lambda(\nabla f)_{\boldsymbol{x}^*}) = 0$ is solved by using the Newton method.

Step4 The numerical solution $\boldsymbol{x}^{(n+1)}$ is determined by $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^* + \lambda(\nabla f)_{\boldsymbol{x}^*}$.

The above three steps are repeated until the following termination conditions:

(i) $\tilde{G}(\boldsymbol{x}^{(n-1)}) < 0$ and $\tilde{G}(\boldsymbol{x}^{(n)}) > 0$.
Here, $\tilde{G}(\boldsymbol{x})$ is defined by

$$\tilde{G}(\boldsymbol{x}) \equiv \left\{\boldsymbol{R}\left(\frac{\pi}{2}\right) \cdot \left[\frac{\nabla f}{|\nabla f|}\right]_{\boldsymbol{x}}\right\} \cdot \left(\boldsymbol{x} - \boldsymbol{x}^{(1)}\right).$$

(ii) $\left|\boldsymbol{x}^{(n)} - \boldsymbol{x}^{(1)}\right| < \gamma \left|\boldsymbol{x}^{(2)} - \boldsymbol{x}^{(1)}\right|$,
where γ is a constant such that $\gamma \simeq 1$.

are fulfilled.

2.4. RPIM Shape Function

In many meshless methods, the moving least-squares (MLS) approximation has been generally adopted as one of the interpolation schemes. This reason is that shape functions can be generated by using only the geometrical location of boundary nodes. However, note that the shape function with the MLS approximation (MLS shape function) does not satisfy the delta function property. In other words, the MLS shape function $\Phi_i^{\mathrm{M}}(s)$ fulfills $\Phi_i^{\mathrm{M}}(s_j) \neq \delta_{i,j}$ where $\delta_{i,j}$ is the Kronecker's delta. This means that both $\psi(\boldsymbol{x}(s_j)) = \psi_j$ and $q(\boldsymbol{x}(s_j)) = q_j$ do not satisfied. Therefore, the number of unknowns equals twice as much as the number of boundary nodes.

In order to resolve the demerit of the MLS shape function, the interpolation scheme used in the RPIM has been proposed. By using the radial basis function $r_i(s)$ and the monomial basis function $p_i(s)$, the shape function can be determined. Then, the curve passing through all boundary nodes is assumed as the approximate function. The approximate function $f^{\mathrm{h}}(s)$ of $f(s)$ in the influence domain can be written as

$$f^{\mathrm{h}}(s) = h_i(s)\left[\boldsymbol{r}^T(s)\,\boldsymbol{b}(s) + \boldsymbol{p}^T(s)\,\boldsymbol{a}(s)\right]. \tag{9}$$

Here, $h_i(s)$ is given by

$$h_i(s) = H\left(1 - |s - s_i|/R_i\right),$$

where $\boldsymbol{r}(s) \equiv [r_1(s), r_2(s), \cdots, r_N(s)]^T$. In addition, R_i and $\boldsymbol{b}(s)$ denote a ith support radius and a N-dimensional vector such that all components are functions of s, respectively.

In order to determine $\boldsymbol{a}(s)$ and $\boldsymbol{b}(s)$, we enforce the interpolation to satisfy the given value at boundary nodes as

$$\begin{bmatrix} R(s) & P(s) \\ P^T(s) & \mathrm{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{b}(s) \\ \boldsymbol{a}(s) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix}, \tag{10}$$

where $R(s)$ and $P(s)$ are defined by

$$R(s) = \sum_{i=1}^{N} h_i(s)\,\boldsymbol{r}(s_i)\,\boldsymbol{r}^T(s_i),$$

$$P(s) = \sum_{i=1}^{N} h_i(s)\,\boldsymbol{e}_i\,\boldsymbol{p}^T(s_i).$$

By solving (10) and substituting it into (9), we can get

$$f^{\mathrm{h}}(s) = \sum_{i=1}^{N} \Phi_i^{\mathrm{R}}(s)\,f(s_i), \tag{11}$$

where

$$\Phi_i^{\mathrm{R}}(s) = \left[\boldsymbol{r}^T(s), \boldsymbol{p}^T(s)\right] \begin{bmatrix} R(s) & P(s) \\ P^T(s) & \mathrm{O} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{e}_i \\ \boldsymbol{0} \end{bmatrix}. \tag{12}$$

Throughout the present study, $\Phi_i^{\mathrm{R}}(s)$ is called the RPIM shape function. Since it has the Kronecker's delta function property, the number of unknowns is equal to the number of boundary nodes.

3. Fast Calculation of Matrix-Vector Production

As is well known, coefficient matrices G and H become asymmetric and dense. Therefore, we cannot solve (5) by using stationary iterative methods. For this reason, the GMRES method has been so far adopted as the solver of (5) [10]. In order to further accelerate, the ACA is applied to matrix-vector multiplications in the GMRES method.

In the ACA, a hierarchical structure, which is called a cluster, is generated by dividing a coefficient matrix into a sub-matrix based on a location of the focussed boundary node. When the cluster distance is near, the sub-matrix is stored as the usual matrix. In contrast, the sub-matrix is approximated as a potential low-rank matrix for the case where the cluster distance is far. When the ratio of the number of the potential low-rank matrices to the number of clusters is large, the matrix-vector production is computed fast. As a result, the speed of the X-BNM becomes fast.

4. Conclusion

We have proposed the approach for improving the performance of the X-BNM by applying of the ACA. By using the proposed approach, it will be able to apply the X-BNM to a large-scale simulation.

As the future work, we will investigate its performance numerically.

Acknowledgment

This work was supported by Japan Society for the Promotion of Science under Grant-in-Aid for Young Scientists (B) (No. 25870630) and Scientific Research (C) (No. 26520204).

References

[1] Y.X. Mukherjee *et al.*, "The boundary node method for potential problems," *Int. J. Numer. Methods Eng.*, vol. 40, iss. 6, pp. 797-815, 1997.

[2] 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].

[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, iss. 5, pp. 1222-1225, May 2011.

[4] A. Saitoh, K. Miyashita, T. Itoh, A. Kamitani, T. Isokawa, N. Kamiura and N. Matsui, "Accuracy improvement of extended boundary-node method," *IEEE Trans. Magn.*, vol. 49, iss. 5, pp. 1601-1604, 2013.

[5] M. Bebendorf, "Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems," *Springer-Verlag*, 2008.

[6] S. Kurz, O. Rain and S. Rjasanow, "The adap- tive cross approximation technique for the 3-D bound- ary element method" *IEEE Trans. Magn.*, vol. 38, iss. 2, pp. 421-424, 2002.

[7] J.G. Wang and G.R. Liu, "A point interpolation meshless method based on radial basis functions," *Int. J. Numer. Meth. Eng.*, vol. 54, iss. 11, pp. 1623-1648, 2002.

[8] G. Turk and J.F. O'Brien, "Modelling with implicit surfaces that interpolate," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 855-873, 2002,

[9] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk and H.P. Seidel, "Multi-level partition of unity implicits," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 463-470, 2003.

[10] A. Saitoh, and A. Kamitani, "GMRES with New Preconditioning for Solving BEM-Type Linear System" *IEEE Trans. Magn.*, vol. 40, iss. 2, pp. 1084-1087, 2004.