New Implementation Method for Essential Boundary Condition to Extended Element-Free Galerkin Method: Application to Nonlinear Problem*)

Ayumu SAITOH, Taku ITOH[1)], Nobuyuki MATSUI, Atsushi KAMITANI[2)] and Hiroaki NAKAMURA[3)]

University of Hyogo, Himeji 671-2280, Japan
[1)]*Seikei University, 3-3-1, Kichijoji-Kitamachi, Musashino 180-8633, Japan*
[2)]*Yamagata University, 4-3-16, Johnan, Yonezawa 992-8510, Japan*
[3)]*National Institute for Fusion Science, 322-6, Oroshi-cho, Toki 509-5292, Japan*

(Received 22 December 2010 / Accepted 22 April 2011)

A new method has been proposed for implementing essential boundary conditions to the Element-Free Galerkin Method (EFGM) without using the Lagrange multiplier. Furthermore, the performance of the proposed method has been investigated for a nonlinear Poisson problem. The results of computations show that, as interpolation functions become closer to delta functions, the accuracy of the solution is improved on the boundary. In addition, the accuracy of the proposed method is higher than that of the conventional EFGM. Therefore, it might be concluded that the proposed method is useful for solving the nonlinear Poisson problem.

Keywords: finite element method, element-free Galerkin method, Lagrange multiplier, nonlinear problem

DOI: 10.1585/pfr.6.2401089

1. Introduction

The finite element method (FEM) and the boundary element method (BEM) have been widely used as numerical methods for solving boundary-value problems of partial differential equations. By using these methods, several excellent results have been obtained in the fields of plasma physics and fusion science etc. However, both FEM and BEM have an inherent demerit: a target region must be divided into a set of elements before executing the FEM/BEM code.

In order to resolve the above difficulty, many meshless approaches have been proposed [1–3]. In the approaches, elements of a geometrical structure are no longer necessary and, hence, the preparation of input data is extremely simplified. However, meshless approaches are plagued by the following difficulty: the method for implementing the essential boundary condition is different according to meshless approaches. For example, as the implementation method, the Lagrange multiplier and the penalty method are used in the Element-Free Galerkin Method (EFGM) [1] and the meshless local Petrov-Galerkin method [2], respectively. If a new implementation method of the essential boundary condition were proposed without dependence on meshless approaches, the above demerit could be resolved completely.

The purpose of the present study is to propose a new method for implementing the essential boundary condition to the EFGM and to evaluate the performance of the proposed method by comparing with the conventional method.

2. Element-Free Galerkin Method

2.1 Discretization

Throughout the present study, we consider a two-dimensional nonlinear Poisson problem:

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

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

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

Here, Ω denotes a region bounded by a simple closed curve $\partial\Omega$ that satisfies the following relation: $\Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}} = \partial\Omega$ and $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \phi$. Furthermore, $\rho(u), \bar{u}$ and $\bar{q}$ are the known functions in Ω, on Γ_{D} and on Γ_{N}, respectively. In addition, $\boldsymbol{n}$ indicates an outward unit normal on $\partial\Omega$.

Since the inhomogeneous term $\rho(u)$ has a nonlinear dependence on the dependent variable u, the solution u is easily calculated by using the successive substitution method. In the kth step, we solve the following linear problem for $u^{(k+1)}$:

$$-\Delta u^{(k+1)} = \rho(u^{(k)}), \qquad \text{in } \Omega \tag{4}$$

$$u^{(k+1)} = \bar{u}, \qquad \text{on } \Gamma_{\mathrm{D}} \tag{5}$$

$$q^{(k+1)} = \bar{q}, \qquad \text{on } \Gamma_{\mathrm{N}} \tag{6}$$

where the superscript (k) is an iteration number label.

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

*) This article is based on the presentation at the 20th International Toki Conference (ITC20).

The above step is repeated until the termination condition $\|u^{(k+1)} - u^{(k)}\|/\|u^{(k+1)}\| \leq 10^{-10}$ is satisfied.

After straightforward calculations, the above linear problem is shown to be equivalent to the following weak form:

$$\forall w\big|_{\Gamma_D} = 0 : \iint_\Omega \nabla w \cdot \nabla u^{(k+1)} \, \mathrm{d}^2\boldsymbol{x} = \iint_\Omega w \, \rho(u^{(k)}) \, \mathrm{d}^2\boldsymbol{x} + \int_{\Gamma_N} w \, \bar{q} \, \mathrm{d}\ell. \quad (7)$$

In the conventional EFGM, the essential boundary condition (5) needs to be incorporated into the weak form (7) through the Lagrange multiplier. As a result, we can obtain

$$\begin{aligned} &\forall w, \forall \delta\lambda^{(k+1)} : \iint_\Omega \nabla w \cdot \nabla u^{(k+1)} \, \mathrm{d}^2\boldsymbol{x} \\ &- \iint_\Omega w \, \rho(u^{(k)}) \, \mathrm{d}^2\boldsymbol{x} - \int_{\Gamma_N} w \, \bar{q} \, \mathrm{d}\ell \\ &+ \int_{\Gamma_D} \delta\lambda^{(k+1)} \left[u^{(k+1)} - \bar{u}\right] \mathrm{d}\ell + \int_{\Gamma_D} w \, \lambda^{(k+1)} \, \mathrm{d}\ell = 0, \end{aligned} \quad (8)$$

where $\lambda^{(k+1)}$ is the Lagrange multiplier.

In order to discretize (8), let us first place $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ in Ω. Furthermore, $\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+2}, \cdots, \boldsymbol{x}_{N+M}$ are placed on $\partial\Omega$. Next, we define the shape functions ϕ_i's which are assigned to the nodes $\boldsymbol{x}_i$'s by using a moving least-square (MLS) approximation [1]. Finally, both $u^{(k+1)}$ and w are assumed to be contained in the functional space $V \equiv \mathrm{span}\,(\phi_1, \phi_2, \cdots, \phi_N)$, i.e., $u^{(k+1)}$ and w are assumed as

$$u^{(k+1)}(\boldsymbol{x}) = \sum_{i=1}^{N+M} \phi_i(\boldsymbol{x}) \, \hat{u}_i^{(k+1)}, \quad (9)$$

$$w(\boldsymbol{x}) = \sum_{i=1}^{N+M} \phi_i(\boldsymbol{x}) \, \hat{w}_i, \quad (10)$$

where $\hat{u}_i^{(k+1)}$ and $\hat{w}_i$ $(i = 1, 2, \cdots, N+M)$ are constants. On the other hand, both $\lambda^{(k+1)}$ and $\delta\lambda^{(k+1)}$ are assumed to be contained in the functional space $W \equiv \mathrm{span}\,(N_1, N_2, \cdots, N_M)$, i.e., $\lambda^{(k+1)}$ and $\delta\lambda^{(k+1)}$ can be expressed as

$$\lambda^{(k+1)}(s) = \sum_{i=1}^{M} N_i(s) \, \hat{\lambda}_i^{(k+1)}, \quad (11)$$

$$\delta\lambda^{(k+1)}(s) = \sum_{i=1}^{M} N_i(s) \, \delta\hat{\lambda}_i^{(k+1)}, \quad (12)$$

where $N_i(s)$ is an interpolation function and s is the arclength along $\partial\Omega$. In addition, $\hat{\lambda}_i^{(k+1)}$ and $\delta\hat{\lambda}_i^{(k+1)}$ $(i = 1, 2, \cdots, M)$ are constants.

Under the above assumptions, (8) is discretized as

$$\left[\begin{array}{c|c} A & B \\ \hline B^T & \mathrm{O} \end{array}\right] \left[\begin{array}{c} \hat{\boldsymbol{u}}^{(k+1)} \\ \hline \boldsymbol{\lambda}^{(k+1)} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f}^{(k)} \\ \hline \boldsymbol{g} \end{array}\right]. \quad (13)$$

Here, matrices A and B are defined by

$$A = \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \left(\iint_\Omega \nabla\phi_i \cdot \nabla\phi_j \, \mathrm{d}^2\boldsymbol{x} \right) \boldsymbol{e}_i \, \boldsymbol{e}_j^T, \quad (14)$$

$$B = \sum_{i=1}^{N+M} \sum_{j=1}^{M} \left(\int_{\Gamma_D} \phi_i \, N_j \, \mathrm{d}\ell \right) \boldsymbol{e}_i \, \boldsymbol{e}_j^T, \quad (15)$$

and nodal vectors, $\hat{\boldsymbol{u}}^{(k+1)}$ and $\boldsymbol{\lambda}^{(k+1)}$, correspond to $u^{(k+1)}$ and $\lambda^{(k+1)}$, respectively. In addition, $\boldsymbol{f}^{(k)}$ and $\boldsymbol{g}$ are defined as follows:

$$\boldsymbol{f}^{(k)} = \sum_{i=1}^{N+M} \left(\iint_\Omega \phi_i \, \rho(u^{(k)}) \, \mathrm{d}^2\boldsymbol{x} + \int_{\Gamma_N} \phi_i \, \bar{q} \, \mathrm{d}\ell \right) \boldsymbol{e}_i, \quad (16)$$

$$\boldsymbol{g} = \sum_{i=1}^{M} \left(\int_{\Gamma_D} N_i(s) \, \bar{u} \, \mathrm{d}\ell \right) \boldsymbol{e}_i. \quad (17)$$

As is apparent from (13), we can obtain the numerical solution by solving the linear system iteratively. In this way, the nonlinear Poisson problem is reduced to the problem in which (13) is solved repeatedly until the termination condition is fulfilled.

2.2 Shape function

As mentioned in 2.1, we must determine shape functions assigned to all nodes. In the EFGM, the shape functions $\phi_j(\boldsymbol{x})$ $(j = 1, 2, \cdots, N+M)$ are defined by means of the MLS approximation and its explicit form are expressed as follows:

$$\phi_j(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{x})^T C^{-1}(\boldsymbol{x}) \, \boldsymbol{b}_j(\boldsymbol{x}), \quad (18)$$

where $C(\boldsymbol{x})$ and $\boldsymbol{b}_j(\boldsymbol{x})$ are given by

$$C(\boldsymbol{x}) = \sum_{j=1}^{N+M} w\left(\frac{|\boldsymbol{x} - \boldsymbol{x}_j|}{R} \right) \boldsymbol{p}(\boldsymbol{x}_j) \, \boldsymbol{p}(\boldsymbol{x}_j)^T, \quad (19)$$

$$\boldsymbol{b}_j(\boldsymbol{x}) = w\left(\frac{|\boldsymbol{x} - \boldsymbol{x}_j|}{R} \right) \boldsymbol{p}(\boldsymbol{x}_j). \quad (20)$$

Moreover, $w(r)$ and R denote a weight function and its support radius, respectively. In addition, $\boldsymbol{p}(\boldsymbol{x})$ is defined by

$$\boldsymbol{p}(\boldsymbol{x})^T = [1, \; x, \; y]. \quad (21)$$

Several weight functions have been proposed so far [1]. In this study, we adopt the spline-type function [4]:

$$w(r) = \begin{cases} 1 - 6\,r^2 + 8\,r^3 - 3\,r^4 & ; r \leq 1, \\ 0 & ; r > 1. \end{cases} \quad (22)$$

3. Extended Element-Free Galerkin Method

In the conventional EFGM, the essential boundary condition (5) has been incorporated into the weak form (7) through the Lagrange multiplier. In this section, we explain a new method for implementing the essential boundary condition to the EFGM without using the Lagrange multiplier.

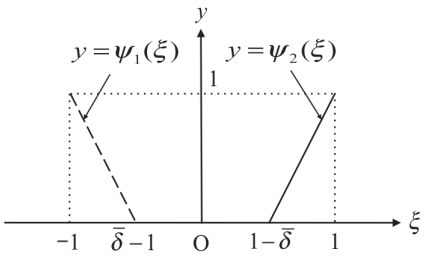

Fig. 1 Graphs of $\Psi_1(\xi)$ and $\Psi_2(\xi)$.

Let us first discretize the essential boundary condition (5). Obviously, (5) can be equivalent to the expression:

$$\int_{\Gamma_D} v(s)\left[u^{(k+1)}-\bar{u}\right]\mathrm{d}\ell = 0, \tag{23}$$

for $\forall v$. When v is assumed to be contained in the functional space W, we get

$$v(s) = \sum_{i=1}^{M} N_i(s)\,\hat{v}_i, \tag{24}$$

where $\hat{v}_i\,(i = 1, 2, \cdots, M)$ are all constants. By using (24), (23) can be discretized as

$$B^T\hat{\boldsymbol{u}}^{(k+1)} = \boldsymbol{g}. \tag{25}$$

Next, the weak form (7) can be similarly discretized as

$$\begin{aligned}&\forall\hat{\boldsymbol{w}}\ \text{s.t.}\ B\boldsymbol{w}=\boldsymbol{0}:\boldsymbol{w}^T[A\hat{\boldsymbol{u}}^{(k+1)}-\boldsymbol{f}^{(k)}]=0,\\ &\Leftrightarrow\exists\boldsymbol{\lambda}^{(k+1)}\in\boldsymbol{R}^M:A\hat{\boldsymbol{u}}^{(k+1)}-\boldsymbol{f}^{(k)}=B\boldsymbol{\lambda}^{(k+1)}.\end{aligned} \tag{26}$$

By combining (25) with (26), we can get the linear system (13). In this way, the nonlinear Poisson problem can be transformed to the linear system (13) without using the Lagrange multiplier. Throughout the present study, the above method is called the eXtended Element-Free Galerkin Method (X-EFGM).

As mentioned above, v is an arbitrary function. Therefore, $N_i(s)$ can be selected arbitrarily. For example, if $N_i(s)$ is given by the Lagrange interpolant, the X-EFGM becomes equivalent to the conventional EFGM. For calculating integrals easily, we adopt a dimensionless variable ξ. By using ξ, $N_{\sigma(e,j)}(\xi)$ is defined by $N_{\sigma(e,j)}(\xi) = (2/\bar{\delta}\,l_e)\Psi_j(\xi)$ (see Fig. 1). Here, $\sigma(e, j)$ and l_e are the global number of the jth local node in the eth integration cell and the length of eth integration cell, respectively.

4. Numerical Results

In this section, we investigate the performance of the X-EFGM by comparing with the conventional EFGM. Throughout the present study, the target region Ω is given by $\Omega \equiv (0, \pi/2) \times (0, \pi/2)$ and $\rho(u)$ is assumed as $\rho(u) = (u^{-3} + u)/2$. Obviously, the analytic solution of the nonlinear Poisson problem is given by $u = \sqrt{\sin(x + y)}$. Furthermore, we adopt a particular solution of $-\Delta u = 0$ as the initial solution $u^{(1)}$ and its explicit form is written as

$$u^{(1)} = -\cosh x \sin y + \cos x \sinh y. \tag{27}$$

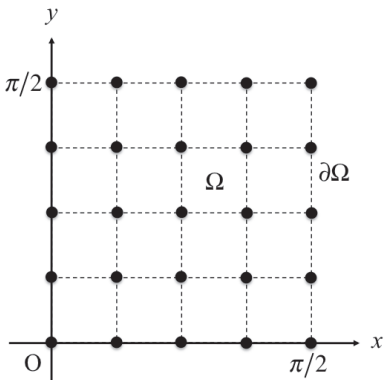

Fig. 2 The spatial distribution of the nodes, $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{N+M}$. Here, the symbol, •, denotes the nodes.

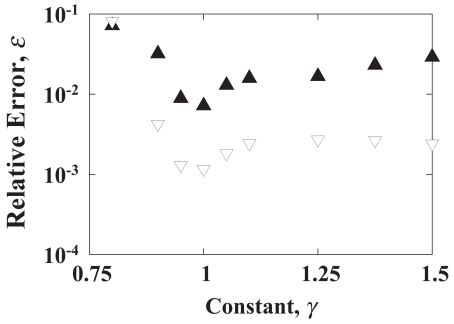

Fig. 3 Dependence of the relative error ε on the constant γ for the case with the EFGM. Here, ▲ : $N + M = 81$ and ▽ : $N + M = 529$.

After we divide Ω into K equal parts along not only the x-direction but also the y-direction, all grid points are adopted as the nodes (see Fig. 2). Furthermore, the support radius R is fixed as $R = \gamma(\pi/\sqrt{2}K)$ where γ is a constant. The relative errors are calculated for the case with EFGM, and they are plotted as functions of γ in Fig. 3. We see from this figure that the relative errors decrease monotonously with an increase in γ until $\gamma \approx 1$. After that, the relative errors are almost constant regardless of γ. This tendency does not depend on the total number of nodes. From the above results, we might say that the appropriate value of γ is $\gamma \approx 1$. Throughout this study, γ is fixed as $\gamma = 1.2$.

First, we investigate the influence of $N_i(\xi)$ on the accuracy of the X-EFGM for the Dirichlet problem. As the measure of the accuracy, we adopt the relative error defined by

$$\varepsilon_B = \frac{\sqrt{\int_{\partial\Omega}\left[u(\boldsymbol{x}(s))-\bar{u}\right]^2\mathrm{d}\ell}}{\sqrt{\int_{\partial\Omega}\bar{u}^2\,\mathrm{d}\ell}}. \tag{28}$$

The relative error is calculated as a function of the dimensionless support radius $\bar{\delta}$ and is depicted in Fig. 4. We see from this figure that the relative error decreases monotonously with a decrease in $\bar{\delta}$. This tendency does not depend on the total number $N+M$ of nodes. The above

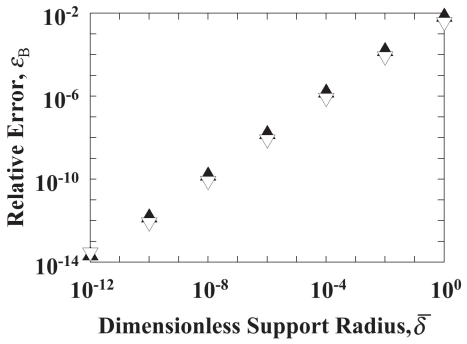

Fig. 4 Dependence of the relative error ε_B on the dimensionless support radius $\bar{\delta}$ for the case with $\gamma = 1.2$. Here, ▲: $N + M = 81$ and ▽: $N + M = 529$.

result shows that the accuracy of the X-EFGM is considerably improved for the case with $N_i(\xi) \approx \delta(\xi - \xi_i)$. For this reason, hereafter, $N_i(\xi)$ is fixed as $N_i(\xi) = \delta(\xi - \xi_i)$.

By comparing with the conventional EFGM, let us investigate the accuracy of the X-EFGM. As the measure of the accuracy of the numerical solution, we adopt the error:

$$\varepsilon_D(\boldsymbol{x}) \equiv \frac{\left|u_A(\boldsymbol{x}) - u_N(\boldsymbol{x})\right|}{\underset{\boldsymbol{x}}{\mathrm{Max}}\left|u_A(\boldsymbol{x})\right|}, \tag{29}$$

where subscript notations, A and N, are analytic and numerical solutions, respectively. Typical examples of the error distribution are shown in Figs. 5 (a) and 5 (b). We see from these figures that the accuracy of the X-EFGM is almost higher than that of the conventional EFGM in Ω. In particular, the accuracy is improved in the vicinity of $y = x$.

Next, we investigate the influence of the total number of nodes on both the accuracy of the X-EFGM and that of the conventional EFGM. The relative errors are calculated as a function of $N + M$ and are depicted in Fig. 6. We see from this figure that the relative errors are almost proportional to $(N + M)^{-\beta}$ among all methods and that the power indices β's satisfy $\beta \approx 1.2$ and $\beta \approx 1.1$ for the X-EFGM and the EFGM, respectively.

Finally, we investigate the calculation speed of the X-EFGM. To this end, the ratio τ_X/τ_E of the CPU time is calculated as a function of $N + M$ and is depicted in the inset of Fig. 6. Here, τ_X is the CPU time reguired for the X-EFGM, whereas τ_E is that for the conventional EFGM. We see from this figure that τ_X/τ_E is almost constant regardless of $N + M$. This means that the calculation speed of X-EFGM is almost equal to that of the conventional EFGM.

From these results, it might be concluded that the X-EFGM is useful for solving the nonlinear Poisson problem.

5. Conclusion

By using a new method for implementing the essential boundary condition to the EFGM, we have developed the X-EFGM. In addition, we have investigated its performance by comparing with the conventional EFGM. Con-

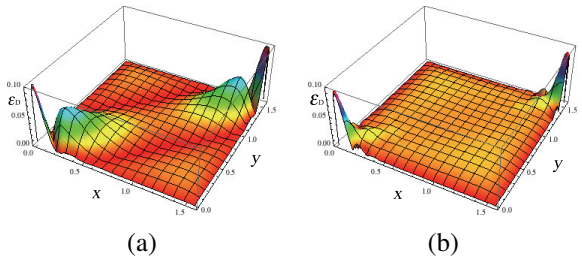

Fig. 5 The error distribution for the case with (a) the conventional EFGM and (b) the X-EFGM ($N + M = 121, \gamma = 1.2$).

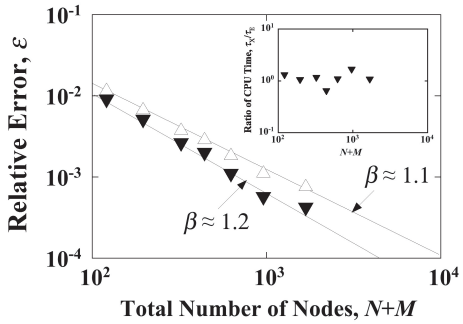

Fig. 6 Dependence of the relative error ϵ on the total number $N + M$ of nodes ($\gamma = 1.2$). Here, the symbols, ▼ and △ indicate the values for the X-EFGM and the conventional EFGM, respectively. The inset shows the dependence of the ratio τ_X/τ_E on the total number $N + M$ of nodes.

clusions obtained in the present study are summarized as follows.

1. When the value of $\bar{\delta}$ decreases, the accuracy of the numerical solution is drastically improved on the boundary.
2. The accuracy of the X-EFGM is higher than that of the conventional EFGM. Furthermore, the calculation speed of X-EFGM is almost equal to that of the conventional EFGM.

Acknowledgment

This work was partially supported by KAKENHI (No.20760059 and No.22360042) and the NIFS Collaboration Research Program (NIFS09KDBN003, NIFS10KKGS006 and NIFS10KNXN178).

[1] T. Belytchko, Y.Y. Lu and L. Gu, Int. J. Numer. Methods Eng. **37**, 229 (1994).
[2] S.N. Atluri and T. Zhu, Comput. Mech. **22**, 117 (1998).
[3] Y.X. Mukherjee and S. Mukherjee, Int. J. Numer. Methods Eng. **40**, 797 (1997).
[4] A. Saitoh, S. Nakata, S. Tanaka and A. Kamitani, Information **12**, 973 (2009) [in Japanese].