 CMC, vol.5, no.2, pp.99-116, 2007

How to Achieve Kronecker Delta Condition in Moving Least Squares Approximation along the Essential Boundary

Jin Yeon Cho[1]

Abstract: A novel way is proposed to fulfill Kronecker delta condition in moving least squares (MLS) approximation along the essential boundary. In the proposed scheme, the original MLS weight is modified to boundary interpolatable (BI) weight based on the observation that the support of weight function is exactly the same as the support of MLS nodal shape function. The BI weight is zero along the boundary edges except the edges containing the nodal point associated with the concerned weight. In order to construct the BI weight from the original weight, concept of edge distance function is introduced, and the BI weight construction procedure is presented in detail. Furthermore, it is explained theoretically why the MLS nodal shape functions obtained by BI weights satisfy Kronecker delta condition along the boundary edges. To identify the validity and usefulness of the proposed BI MLS approximation scheme through numerical tests, the scheme is applied to the model problems with rectangular domain and complex shaped domain. Through the tests, theoretical prediction is identified numerically, and it is confirmed that one can handle the essential and natural boundary conditions through the proposed BI MLS scheme in exactly the same manner used in traditional finite element methods.

Keyword: BI (Boundary Interpolatable) Weight, Edge Distance Function, Essential Boundary Condition, Kronecker Delta Condition, Moving Least Squares Approximation.

[1] Associate Professor, Department of Aerospace Engineering, INHA University, 253 Yonghyun-Dong, Nam-Gu, Incheon, 402-751, Korea. E-mail: cjy@inha.ac.kr

1 Introduction

Most of the meshless (or mesh-free) analysis methods rely on the meshless approximation methods. Through the meshless approximations, the shape functions for unknown variables, which are utilized in variational weak forms or strong forms of ordinary or partial differential equations, can be constructed only with the nodal points with no aid of well-defined mesh. The feature of independency of meshes in constructing the shape functions gives various potentials in dealing with engineering problems. And the drawbacks frequently encountered in applying the finite element methods, such as human labor-intensive meshing, degradation of solution accuracy according to the element distortions, difficulty in tracking the moving boundary, burden of re-meshing during large deformation, locking, element mismatching, and others, have been expected to be eliminated or alleviated by adopting the meshless (or mesh-free) approaches.

With the anticipation, considerable research efforts have been given to the field of meshless method and its application to the analyses of several engineering problems. As a result, various methods have been proposed in the context of meshless analysis to alleviate the drawbacks of traditional analysis methods dependent on mesh (or grid) such as finite element, finite volume, finite difference, and others. Some of these are SPH (smoothed particle hydrodynamics) [Lucy (1977)], Generalized Finite Difference [Liszka and Orkisz (1980)], DEM (diffuse element method) [Nayroles, Touzot, and Villion (1992)], EFG (element free Galerkin method) [Belytschko, Lu, and Gu (1994)], Generalized Finite Element [Babuska and Melenk (1997)], MLPG (meshless local Petrov-Galerkin method)

[Atluri and Zhu (1998)], MLS displacement welding [Cho, et al. (2005)], and others [Atluri (2005)].

Even though those have their own salient features, all of the methods have the common character. It is that they utilize the nodal shape functions obtained from the meshless approximation schemes in order to approximate the unknown variables in weak form or strong form of differential equations. Consequently, all of the meshless analysis methods inherit the common character from the meshless approximation, and the character makes it possible to eliminate or alleviate the mesh-related drawbacks in the traditional numerical analysis methods such as finite element, finite volume, finite difference, and so on.

Among the meshless approximation methods, the moving least squares scheme [Lancaster and Salkauskas (1981)] may be one of the most representative approximation techniques. Its lowest version is the same as Shepard interpolation [Shepard (1968)], and it is essentially the same as the reproducing kernel particle method (RKPM) [Liu, Jun, and Chang (1995); Atluri (2005)]. For its generalized version, one can see references [Atluri, Cho, and Kim (1999); Cho and Atluri (2001)].

However, due to the character of diffuseness in meshless approximations including the moving least squares scheme, most of the meshless approximations lose the exact interpolation property, differing from Lagrange interpolation functions which are usually adopted in finite element method. And the lack of exact interpolation property directly leads to the difficulty in enforcement of essential boundary conditions when we apply meshless analysis methods in solving differential equations, although enforcement of essential boundary conditions in traditional finite element method is very simple task. Because of this drawback, several approaches have been proposed to resolve this unexpected trouble. Some of these are Lagrange multiplier method [Belytschko, Lu, and Gu (1994)], coupling to finite elements [Krongauz and Belytschko (1996)], the penalty method [Zhu and Atluri (1998)], the mixed (or partial) transformation method [Atluri, Kim, and Cho (1999); Chen and Wang (2000)], and so on.

Lagrange multiplier method requires additional unknowns and does not preserve the positive definiteness and banded structure of system matrix. Method of coupling to finite elements requires well-defined meshes (or elements). The mixed (or partial) transformation method requires additional computation and bookkeeping for transformation of boundary values, and it needs additional computing cost to preserve the symmetry of system matrix. In case of penalty method, numerical solution is very sensitive to the penalty parameter, although it is easy to implement and preserves the positive definiteness and banded structure of system matrix. In this sense, the trouble in enforcing the essential boundary condition is not eliminated completely, and most of the analysis methods based on meshless approximation scheme still have troubles and inconveniences in enforcing the essential boundary condition.

Under this background, this work aims to completely eliminate the trouble in enforcing the essential boundary conditions in meshless methods, focusing on the moving least squares method. For the purpose, by using the newly devised edge distance function and modified weight function, a novel boundary interpolatable (BI) moving least squares scheme is proposed, where the obtained meshless nodal shape function has the exact interpolation property along the boundary of interest. The exact interpolation property obtained by the newly proposed scheme makes it possible to enforce the boundary conditions in meshless analysis methods with no difficulty just like in conventional finite element methods.

2 Original Moving Least Squares Approximation

Let us assume that a continuous function $u(\mathbf{x})$ is defined on a domain Ω, and its nodal values $u(\mathbf{x}_I)$ at the scattered nodal points $\mathbf{x}_I$ $(1 \leq I \leq N)$ in the domain Ω are given as $\hat{u}^I$ $(1 \leq I \leq N)$. Then, in the moving least squares method, the following global approximation form (1) is defined in order to approximate the continuous function $u(\mathbf{x})$ only with the given nodal values $\hat{u}^I$ at the scat-

tered points $\mathbf{x}_I$.

$$u(\mathbf{x}) \cong u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) = \sum_{k=1}^{m} p_k(\mathbf{x})a_k(\mathbf{x}) \quad (1)$$

where $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \cdots, p_m(\mathbf{x})]$ is a r-times differentiable basis and $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \cdots, a_m(\mathbf{x})]^T$ is a vector of undetermined coefficient functions, whose values are varying according to the position $\mathbf{x} \in \Omega$. The basis $\mathbf{p}(\mathbf{x})$ is selected to contain constant '1', and to be linearly independent over some set of m among the given N points in Ω [Lancaster and Salkauskas (1981)].

For example, the $(m-1)$-th order monomial basis in one dimension is written as follows.

$$\mathbf{p}^T(x) = [1, x, x^2, \cdots, x^{m-1}] \quad (2)$$

In two dimensions, linear and quadratic monomial bases have the following forms, respectively.

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y] \quad (3)$$

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y, x^2, xy, y^2] \quad (4)$$

The coefficient vector $\mathbf{a}(\overline{\mathbf{x}})$ at each position $\mathbf{x} = \overline{\mathbf{x}}$ will be determined by a local weighted least squares approximation $u_{\overline{\mathbf{x}}}(\mathbf{x})$ of the function $u(\mathbf{x})$. In a sufficiently small neighborhood of each point $\overline{\mathbf{x}} \in \Omega$, the local approximation $u_{\overline{\mathbf{x}}}(\mathbf{x})$ is defined by the form (5), and the coefficient vector $\mathbf{a}(\overline{\mathbf{x}})$ is selected to fulfill the minimizing condition (6).

$$u_{\overline{\mathbf{x}}}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\overline{\mathbf{x}}) \quad (5)$$

$$J_{\overline{\mathbf{x}}}(\mathbf{a}(\overline{\mathbf{x}})) \leq J_{\overline{\mathbf{x}}}(\mathbf{b}), \text{ for all } \mathbf{b} \in R^m \quad (6)$$

where $J_{\overline{\mathbf{x}}}(\mathbf{b})$ is defined by the weighted least squares L_2 error norm as shown in Eq. (7).

$$J_{\overline{\mathbf{x}}}(\mathbf{b}) = [\mathbf{P}\mathbf{b} - \overline{\mathbf{u}}]^T \mathbf{W}(\overline{\mathbf{x}}) [\mathbf{P}\mathbf{b} - \overline{\mathbf{u}}] \quad (7)$$

where,

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_N) \end{bmatrix} = \left[\begin{array}{ccc} p_1(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ \hdashline p_1(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \hdashline & \vdots & \\ \hdashline p_1(\mathbf{x}_N) & \cdots & p_m(\mathbf{x}_N) \end{array}\right] \quad (8)$$

$$\overline{\mathbf{u}} = \begin{pmatrix} u(\mathbf{x}_1) \\ u(\mathbf{x}_2) \\ \vdots \\ u(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} \hat{u}^1 \\ \hat{u}^2 \\ \vdots \\ \hat{u}^N \end{pmatrix} \quad (9)$$

$$\mathbf{W}(\overline{\mathbf{x}}) = \begin{bmatrix} W_1(\overline{\mathbf{x}}) & 0 & \cdots & 0 \\ 0 & W_2(\overline{\mathbf{x}}) & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & W_N(\overline{\mathbf{x}}) \end{bmatrix} \quad (10)$$

The $N \times m$ matrix $\mathbf{P}$ consists of basis, and the vector $\overline{\mathbf{u}}$ denotes the vector of given values $\hat{u}^I$ of variable u at nodes I $(1 \leq I \leq N)$. The $N \times N$ diagonal matrix $\mathbf{W}(\overline{\mathbf{x}})$ is composed of weight functions. The weight function $W_I(\mathbf{x})$ is associated with the position $\mathbf{x}_I$ of node I. The weight function $W_I(\mathbf{x})$ is selected to be non-negative for all $\mathbf{x}$, and the region of non-zero values is called the support.

The coefficient vector $\mathbf{a}(\overline{\mathbf{x}})$, which satisfies the minimizing condition (6), is determined by applying the stationarity condition to the weighted discrete error norm as shown below.

$$\left[\mathbf{P}^T\mathbf{W}(\overline{\mathbf{x}})\mathbf{P}\right]\mathbf{a}(\overline{\mathbf{x}}) = \left[\mathbf{P}^T\mathbf{W}(\overline{\mathbf{x}})\right]\overline{\mathbf{u}} \quad (11)$$

A local weighted least squares approximation is found by solving Eq. (11) for $\mathbf{a}(\overline{\mathbf{x}})$, and the coefficient vector $\mathbf{a}(\overline{\mathbf{x}})$ is used to construct the global approximation (1) at each position $\mathbf{x} = \overline{\mathbf{x}}$. This is the moving procedure of local approximation to obtain the global approximation, as stated in the previous work [Lancaster and Salkauskas (1981)]. The method to approximate the function by the moving least squares method is sketched in Fig. 1.

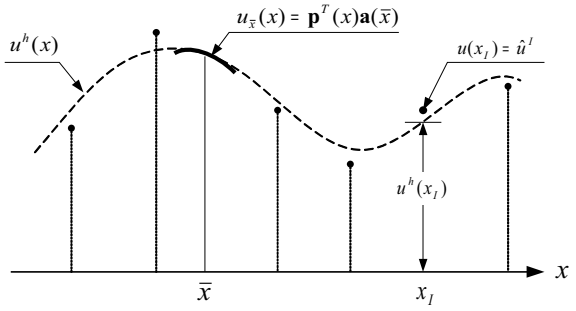

Figure 1: Construction of the moving least squares approximating function

Additionally, the global approximation may be rewritten in the form of a linear combination of nodal shape functions similar to that used in finite element method as shown in Eq. (12)

$$u^h(\mathbf{x}) = \Psi^T(\mathbf{x})\overline{\mathbf{u}} = \sum_{I=1}^{N} \hat{u}^I \psi_I(\mathbf{x}) \tag{12}$$

where,

$$\Psi^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{P}\right]^{-1} \mathbf{P}^T\mathbf{W}(\mathbf{x})$$
$$\psi_I(\mathbf{x}) = \sum_{k=1}^{m} p_k(\mathbf{x}) \left[\left[\mathbf{P}^T\mathbf{W}(\mathbf{x})\mathbf{P}\right]^{-1}\mathbf{P}^T\mathbf{W}(\mathbf{x})\right]_{kI} \tag{13}$$

3 Boundary Interpolatable Moving Least Squares Approximation

3.1 Properties of MLS approximation

As widely known, the value of MLS approximation at nodal point $\mathbf{x}_I$ is not the same as the given data $\hat{u}^I = u(\mathbf{x}_I)$ at location $\mathbf{x}_I$.

$$u^h(\mathbf{x}_I) \neq \hat{u}^I = u(\mathbf{x}_I) \tag{14}$$

Therefore, MLS nodal shape functions lose Kronecker delta property. And consequently, it becomes difficult to enforce the essential boundary condition unlike the traditional finite element methods.

$$\psi_I(\mathbf{x}_J) \neq \delta_{IJ} \tag{15}$$

In Fig. 2, one can see that the nodal shape functions obtained from the moving least squares approximation procedure do not have Kronecker delta property. The reason for losing Kronecker delta property is that the value of weight function $W_J(\mathbf{x})$ associated with nodal points $\mathbf{x}_J (J \neq I)$ is not zero at the location of nodal point of interest $\mathbf{x}_I$.

Based on the observation, singular weight functions have been utilized to equip MLS nodal shape function with Kronecker delta property in previous work [Lancaster and Salkauskas (1981)]. In this singular weight approach, MLS nodal shape function becomes to have Kronecker delta property in nodal points. However, because nodal

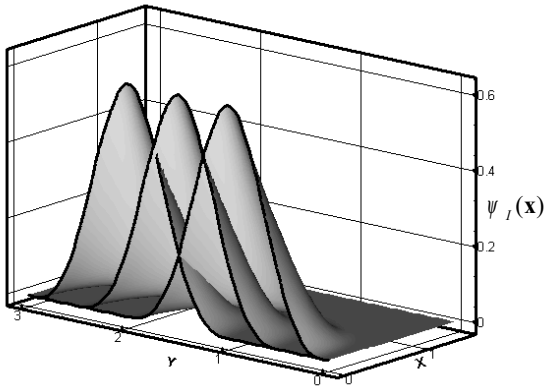

Figure 2: Non-Kronecker delta property of MLS nodal shape functions

shape functions have the property similar to original ones along the boundary except the position of nodal points, it is still difficult to completely enforce the essential boundary conditions along the boundary unlike the conventional finite element methods. Because of this limitation, the essential boundary condition is enforced not through the boundary edges (or faces) but through the boundary nodal points in the singular weight approach. Additionally, the quality of MLS approximation with singular weight is degraded compared with the original non-singular weight MLS approximation.

In actual computations, various kinds of weight functions can be adopted for MLS approximation, and the required smoothness of MLS can be easily achieved by changing the weight function $W_I(\mathbf{x})$ in the MLS approximation. Usually the weight function $W_I(\mathbf{x})$ centered at each node $\mathbf{x}_I$ is adopted to be positive and non-zero if the distance between node $\mathbf{x}_I$ and $\mathbf{x}$ is less than a specified radius R_I, and to be zero if the distance is greater than or equal to the given radius R_I.

In Eq. (16), typical form of weight function is presented.

$$W_I(\mathbf{x}) = \begin{cases} 1 - 3\left(\frac{d_I}{R_I}\right)^2 + 2\left(\frac{d_I}{R_I}\right)^3, & \text{if } 0 \leq d_I < R_I \\ 0, & \text{if } d_I \geq R_I \end{cases} \tag{16}$$

where R_I denotes the radius of support of weight function and d_I denotes the distance between the

point $\mathbf{x}$ and nodal point $\mathbf{x}_I$. It is noted that the weight function (16) is C^1 continuous (differentiable) function.

If the derivatives of MLS basis $\mathbf{p}(\mathbf{x})$ are continuous up to the r-th derivative, the resulting function from MLS approximation is continuously differentiable up to the minimum of r and the smoothness order of weight function. Therefore, if one uses the weight function (16) along with infinitely differentiable monomial basis, one can obtain a C^1 continuous (differentiable) moving least squares approximating function.

Also one can use other kinds of weight functions such as Gaussian weight function [Alturi (2005)] as shown in Eq. (17).

$$W_I(\mathbf{x}) = \begin{cases} \frac{e^{\left(-(d_I/c_I)^{2k}\right)} - e^{\left(-(R_I/c_I)^{2k}\right)}}{1 - e^{\left(-(R_I/c_I)^{2k}\right)}}, & \text{if } 0 \le d_I < R_I \\ 0, & \text{if } d_I \ge R_I \end{cases} \tag{17}$$

where c_I is a constant controlling the shape of weight function. For k=1, the weight function becomes C^0continuous over the entire domain. In this case of k=1, one may obtain a C^0 continuous MLS approximating function at most.

Also one may notice that the regions of non-zero values of weight functions (16) and (17) are local circular (or spherical) regions with a specified radius R_I. This local character of weight function results in the locality of MLS nodal shape function, which directly leads to the sparseness of system matrix for the considered differential equation.

Furthermore, it should be noted that the support (region of non-zero values) of MLS nodal shape function becomes exactly the same as the support of weight function associated with the nodal point of interest. Thus, the support of MLS nodal shape function can be tailored according to the support of weight function. Fig. 3 shows MLS nodal shape functions, which are corresponding to the weight functions with circular and rectangular supports.

In summary, MLS nodal shape function has the following intrinsic properties.

1) In general, MLS nodal shape functions do not have Kronecker delta property.

2) One may obtain the required smoothness of MLS nodal shape function through the weight function.

3) One can tailor the local support of MLS nodal shape function by changing the support of weight function.

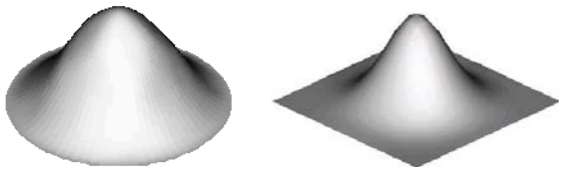

Figure 3: MLS nodal shape functions with circular support and rectangular support

3.2 Edge Distance Function, Modified Weight Function (Boundary Interoplatable (BI) Weight)

In real practice, most of the geometric engineering models constructed by CAD tools are represented by well-defined boundary edges(lines) or faces(surfaces) as presented in Fig. 4.

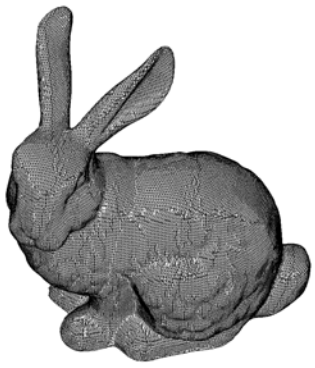

Figure 4: Geometric model represented by well-defined faces

In this point of view, boundary edge(line) or face(surface) is essential to represent the geometric model precisely even in meshless approach, and it is natural to utilize the information of boundary edge or face for dealing with geometric model regardless of meshless or mesh-based

approaches. Based on this practical observation, the information of boundary edge (or face) is utilized to tailor the support of weight function in this work. The brief concept for tailoring the support of weight function (consequently, the support of MLS nodal shape function) through the edge information is as follows.

Let us consider a weight function for x_I, which is not zero on the boundary, as shown in Fig. 5. Then the resulting MLS nodal shape function $\psi_I(x)$ has non-zero value on the boundary as noted previously. However if we multiply the edge distance function $D_e(x)$ in Fig. 6 to the original weight function, then we can make the modified BI (boundary interpolatable) weight which is zero on the boundary as shown in Fig. 7. And as a result, we can obtain MLS nodal shape function which has zero value on the boundary.

By using the concept, we can construct the boundary interpolatable (BI) MLS nodal shape functions which satisfy Kronecker delta condition. Depending upon whether the nodal point is located in interior domain or boundary region, BI weight construction procedure is changed as follows.

Let us consider the nodal point x_I located in the interior domain. Suppose that the support of original weight function for node x_I includes boundary points as shown in Fig. 8. Then all of the edge distance functions corresponding to the included boundary edges are multiplied to the original weight function in order to construct BI weight function. If the support of original weight for the interior node x_I does not include boundary point as shown in Fig. 9, then BI weight can be adopted as the original weight $W_I(x)$ or $W_I(x)D_{e1}(x)D_{e2}(x)$ since both have zero value on the boundary. The first case corresponds to a local edge finding algorithm where only the edges intersecting with the support of weight are considered in constructing BI weight. The second case is a global edge finding algorithm which considers all of the edges.

If the nodal point x_I is located on the boundary, the procedure for constructing BI weight is slightly changed from the case of interior nodal point. In this case, the edge distance function as-

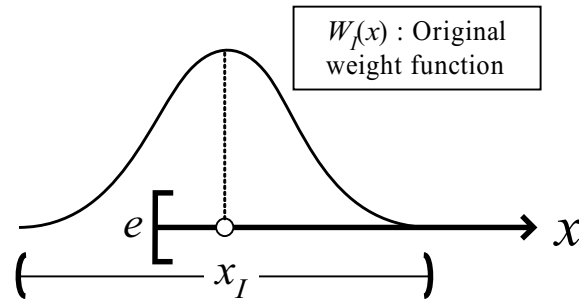

Figure 5: Original weight function associated with the nodal point x_I

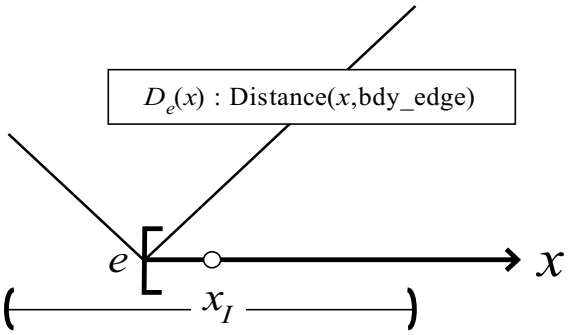

Figure 6: Edge distance function $D_e(x)$ for edge e

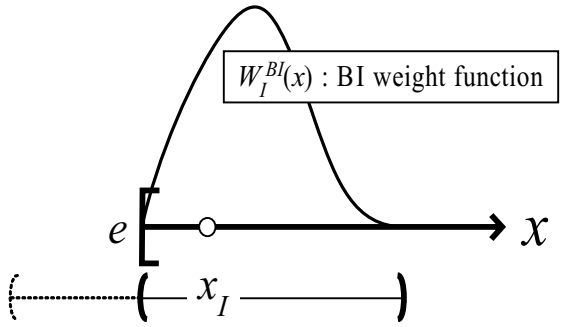

Figure 7: BI (boundary interpolatable) weight constructed by multiplying the edge distance function to the original weight

sociated with the nodal point x_I is excluded, even though the nodal point x_I itself is also the boundary point. As sketched in Fig. 10, if the support of weight function for nodal point x_I includes the other boundary point e_2, the edge distance function, which corresponds to the boundary point e_2, is multiplied to the original weight function in order to make the BI weight. If no boundary point is included in the support of weight function except the nodal point x_I as shown in Fig. 11, then BI weight can be adopted as the original weight $W_I(x)$ or $W_I(x)D_{e2}(x)$ since both are zero

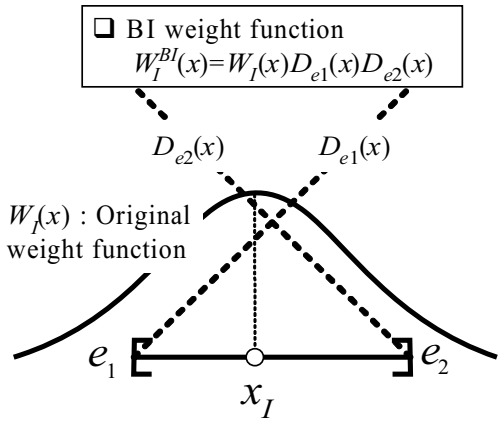

Figure 8: Constructing BI weight for interior node with support intersecting with boundary

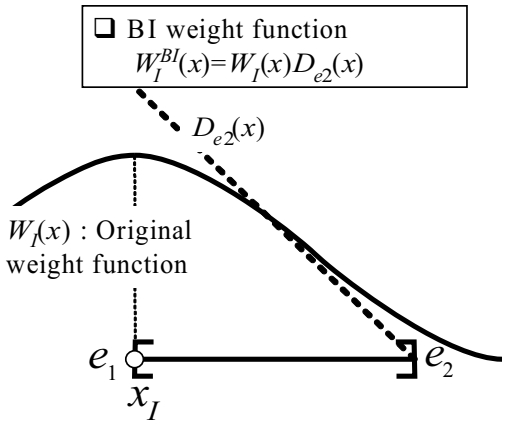

Figure 10: Constructing BI weight for boundary node with support intersecting with the other boundary

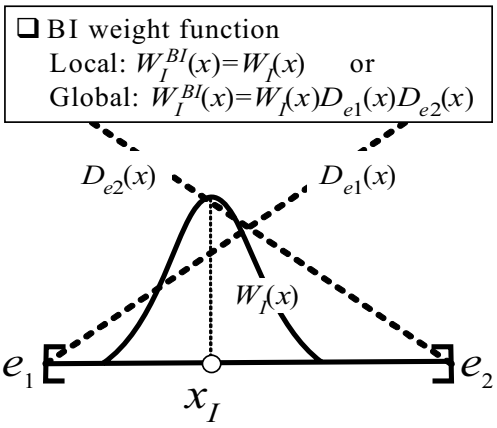

Figure 9: Constructing BI weight for interior node with support which does not intersect with boundary

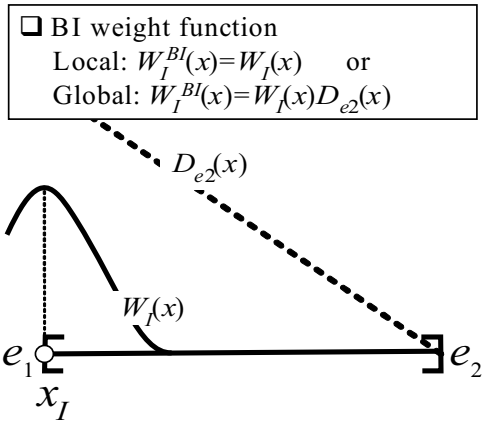

Figure 11: Constructing BI weight for boundary node with support which does not intersect with the other boundary

on the boundary except the nodal point x_I. The first and second cases are corresponding to local and global edge finding algorithms for boundary nodal point, respectively. Here, it is noted that local edge finding algorithm, where only the edges intersecting with support of weight function are considered, is more efficient than global edge finding algorithm in practical problems.

Differing from one dimension, boundary in two or three dimensions is composed of collection of points. Therefore, edge distance functions in two and three dimensions should be defined in different form. In this work, edge distance functions in two and three dimensions are defined as the minimum distance between the point $\mathbf{y}$ in boundary edge (or face in 3D) and the point of interest $\mathbf{x}$ in domain Ω as denoted in (18).

$$D_e(\mathbf{x}) = \min_{\mathbf{y}\in Edge\_e} \|\mathbf{y}-\mathbf{x}\| \tag{18}$$

Edge distance functions in two and three dimensions are presented graphically in Fig. 12 and 13, respectively. This concept may be also extended for general curved lines or surfaces.

Furthermore, one can utilize not only the form of (18) but also its variants (19) in order to convert the original weight into the modified BI weight, since its variants presented in (19) are also zero

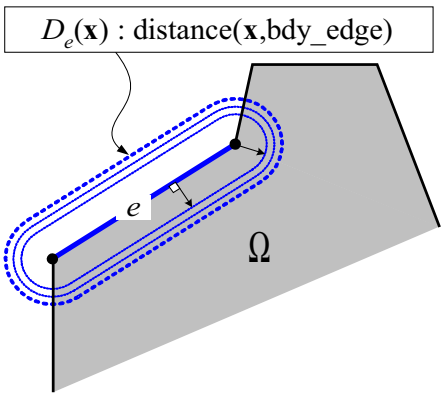

Figure 12: Edge distance function in two dimensions

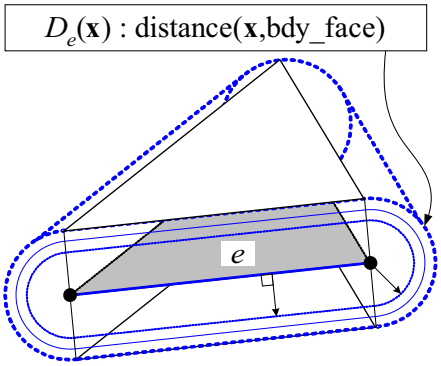

Figure 13: Edge distance function in three dimensions

on the boundary.

$$f(D_e(\mathbf{x}))(f(0)=0 \quad \text{and} \quad f(D)\geq 0) \tag{19}$$

where $f(D)$ may be spline, power, or other function forms. The simplest form among the variants of edge distance function may be the form of $cD_e(\mathbf{x})$ with positive normalizing coefficient c.

By using the aforementioned edge distance function or its variants, one can tailor the support of weight function in two and three dimensions. In case of 2D and 3D, algorithm for constructing BI weight is slightly extended from the one dimensional case, since the boundary of 2D and 3D is not a point but a set of points. In BI weight construction algorithm for 2D and 3D, the original weight is made be zero for the boundary edges (or faces) except the edges (or faces) containing the nodal point of interest, whereas the original weight is made be zero only for the boundary points different from the concerned nodal point in one dimension.

And consequently, one can obtain the MLS nodal shape function which is completely zero along the boundary edges except the edges containing the nodal point of interest. The BI weight construction algorithm in two and three dimensions is shown below. It is noted that the following algorithm can be also utilized in one dimension without modification if we consider the boundary point as a boundary edge in one dimension.

[Algorithm for BI weight construction]

- ❑ Identify the set of candidate boundary edges C_E
- ❑ Locate the position $\mathbf{x}$ and nodal point $\mathbf{x}_I$
- ❑ **Local**) Find the edges which do not contain the nodal point $\mathbf{x}_I$ and intersect with support of $W_I(\mathbf{x})$

  $$E_I = \{\, e \in C_E \,|\, e \cap \text{supp}(W_I(\mathbf{x})) \neq \phi,\ \mathbf{x}_I \notin e\}$$

  or Global) Find the boundary edges which do not contain the nodal point $\mathbf{x}_I$

  $$E_I = \{\, e \in C_E \,|\, \mathbf{x}_I \notin e\}$$

- ❑ Calculate the edge distance function

  $$D_e(\mathbf{x})(e \in E_I) \text{ or } f(D_e(\mathbf{x}))(e \in E_I)$$

- ❑ Construct the BI weight

  $$W_I^{BI}(\mathbf{x}) = W_I(\mathbf{x}) \prod_{e \in E_I} D_e(\mathbf{x})$$
  $$\text{or } W_I^{BI}(\mathbf{x}) = W_I(\mathbf{x}) \prod_{e \in E_I} f(D_e(\mathbf{x}))$$

As mentioned previously, the local edge finding algorithm is more efficient than global edge finding algorithm especially for 2D and 3D problems. In local edge finding algorithm, the original weights for the majority of nodal points are utilized without any modification. Therefore the increased computational cost compared with the original MLS approximation is proportional only

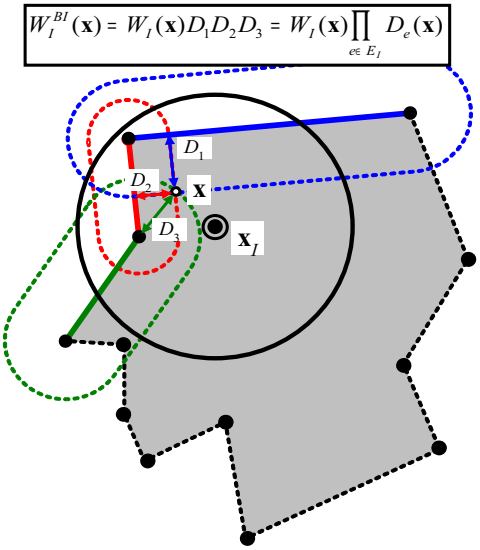

Figure 14: Construction of BI weight for interior nodal point by using local edge finding algorithm

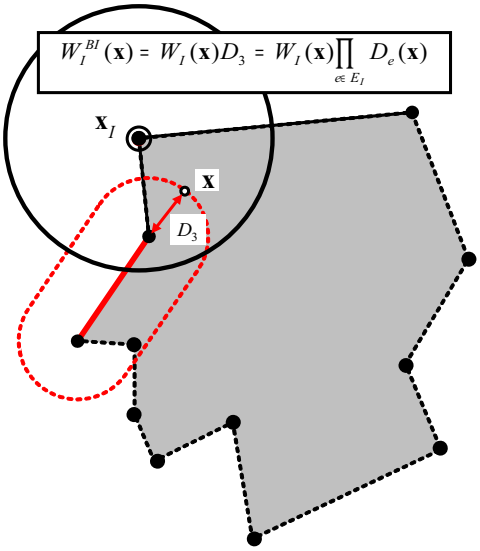

Figure 15: Construction of BI weight for boundary nodal point by using local edge finding algorithm

to the small number of nodal points near the boundary of interest, such as the essential or natural boundary.

Fig. 14 and Fig. 15 present graphically how to construct BI weights in 2D associated with interior nodal point and boundary nodal point through the local edge finding algorithm, respectively.

In Fig. 16 and Fig 17, the original weight function and the corresponding BI weight function for

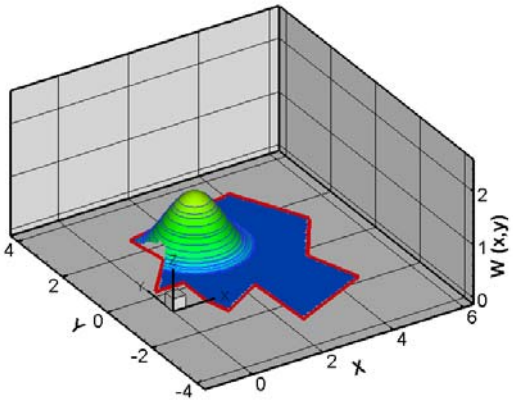

Figure 16: Original weight function associated with interior nodal point

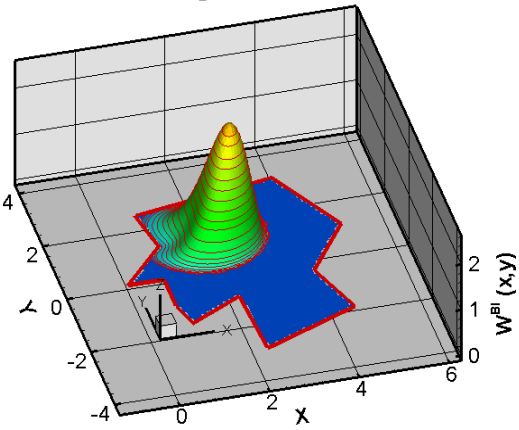

Figure 17: BI weight function for interior nodal point (by using the normalized edge distance function $cD_e(x)$ and local edge finding algorithm)

interior nodal point are presented. To construct BI weight, the normalized edge distance function $cD_e(\mathbf{x})$ is utilized along with local edge finding algorithm. One may easily observe that the obtained BI weight values are becoming zero along the boundary.

Fig. 18 and Fig. 19 show the original weight function and the corresponding BI weight function for nodal point located on the boundary, respectively. Like Fig. 17, the normalized edge distance function $cD_e(\mathbf{x})$ is utilized along with local edge finding algorithm in order to construct BI weight presented in Fig. 19. Fig. 20 shows the BI weight function obtained by using local edge finding algorithm and variant form of edge distance function $f(D_e(\mathbf{x})) = c_1D_e(\mathbf{x})(1-c_2D_e(\mathbf{x}))^2$. Here, c_1

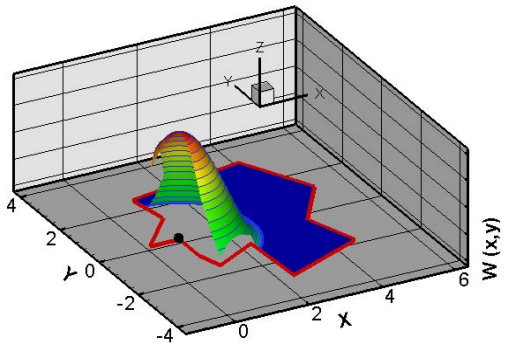

Figure 18: Original weight function associated with the nodal point located on boundary

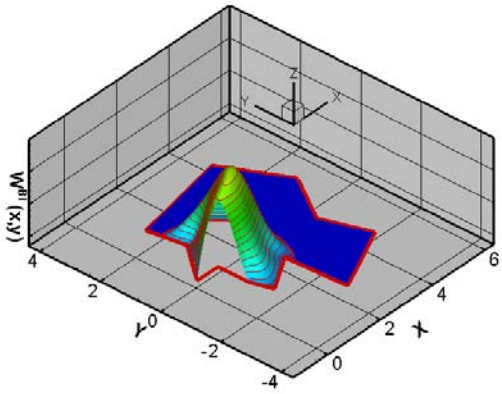

Figure 19: BI weight function for the boundary nodal point (by using the normalized edge distance function $cD_e(x)$ and local edge finding algorithm)

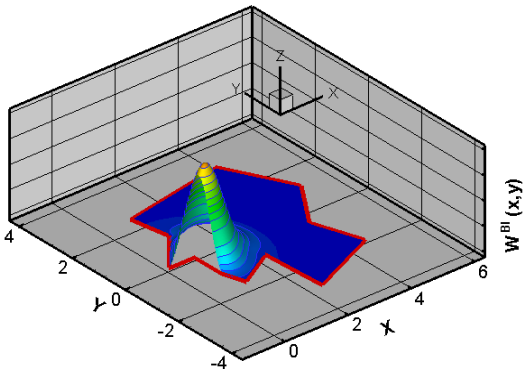

Figure 20: BI weight function for boundary nodal point (by using the variant form of edge distance function $c_1D_e(x)(1-c_2D_e(x))^2$ and local edge finding algorithm)

and c_2 denote positive normalizing coefficients.

In both cases of Fig. 19 and Fig. 20, we can observe that the BI weight values along the boundary become zero except the edges containing the nodal point of interest. Also from Fig. 21 and Fig. 22, one can see that the BI weight obtained by $c_1D_e(\mathbf{x})(1-c_2D_e(\mathbf{x}))^2$ is less diffusive compared with the BI weight from $cD_e(\mathbf{x})$.

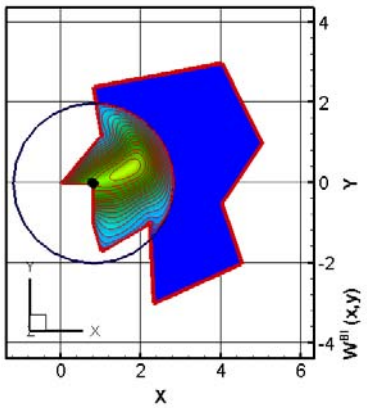

Figure 21: BI weight function for boundary nodal point (by using $cD_e(x)$ and local edge finding algorithm)

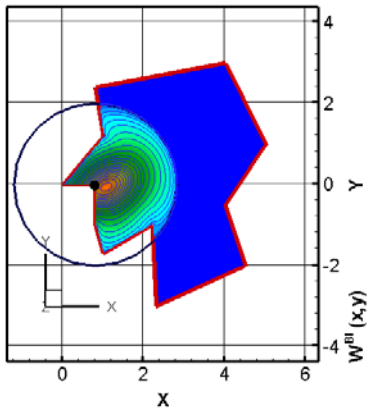

Figure 22: BI weight function for boundary nodal point (by using $c_1D_e(x)(1-c_2D_e(x))^2$ and local edge finding algorithm)

3.3 *BI (Boundary Interpolatable) MLS Nodal Shape Function*

For the interior node, the associated BI weight function has zero value along the boundary. And for the boundary node $\mathbf{x}_I$, the corresponding BI weight function is zero along the boundary edges except the edges containing the nodal point $\mathbf{x}_I$. Therefore, all of the BI weight functions become zero along the boundary edge e except the BI

weights associated with the nodal points which are located on the concerned edge e as sketched in Fig. 23.

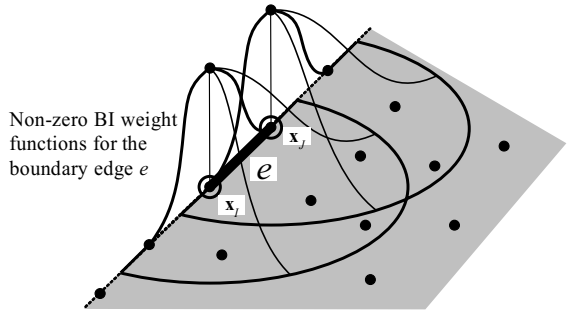

Figure 23: Non-zero BI weight functions for the boundary edge e

As a result, the matrix $\mathbf{P}^T\mathbf{W}(\overline{\mathbf{x}})\mathbf{P}$ may become singular along the boundary edge, since sufficient number of non-zero weights is not available on the boundary. And one may not determine the coefficient vector $\mathbf{a}(\overline{\mathbf{x}})$ by direct inversion because of the rank deficiency of $\mathbf{P}^T\mathbf{W}(\overline{\mathbf{x}})\mathbf{P}$.

Therefore, in this work the coefficient vector $\mathbf{a}(\overline{\mathbf{x}})$ is determined by the limit of sequence rather than the conventional way as denoted in (20).

$$\mathbf{a}(\overline{\mathbf{x}}) = \lim_{k\to\infty} \mathbf{a}_k(\overline{\mathbf{x}}) \tag{20}$$

And the sequence $\mathbf{a}_k(\overline{\mathbf{x}})$is defined as follows

$$\begin{aligned}&\left[\mathbf{P}^T\mathbf{W}(\overline{\mathbf{x}}+(1/k)\mathbf{n})\mathbf{P}\right]\mathbf{a}_k(\overline{\mathbf{x}})\\ &=\left[\mathbf{P}^T\mathbf{W}(\overline{\mathbf{x}}+(1/k)\mathbf{n})\right]\overline{\mathbf{u}}\end{aligned} \tag{21}$$

where, $\mathbf{n}$ denotes an arbitrary inward unit vector, and the point $\overline{\mathbf{x}}+(1/k)\mathbf{n}$ is located in local neighborhood of $\overline{\mathbf{x}}$. It is noted that there exists the limit of sequence (21) because the sequence is Cauchy sequence defined on the real finite dimensional normed linear space [Bartle and Sherbert (1982)].

Consider BI MLS approximation with the linear basis $\mathbf{p}(x,y) = [1,x,y]^T$ and BI weights in 2 dimensional domain Ω represented by straight edges. Then the following proposition regarding the exact interpolation property is satisfied.

Proposition. *Suppose that the boundary edge e of interest is composed of nodal points $\mathbf{x}_1$ and $\mathbf{x}_2$, and it is represented by linear equation $0 = g(\mathbf{x}) = \alpha+\beta x+\gamma y$.*

Then the value $u^h(\overline{\mathbf{x}})$ of BI MLS function at point $\overline{\mathbf{x}} = \lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2$ $(0\le\lambda\le 1)$ along the boundary edge e is represented by the convex combination of the given nodal values $\hat{u}^1$ and $\hat{u}^2$.

(i.e., $u^h(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2) = \lambda\hat{u}^1+(1-\lambda)\hat{u}^2$, $\forall\ 0\le\lambda\le 1$)

Proof. At point $\overline{\mathbf{x}} = (\overline{x},\overline{y}) = \lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2$ $(0<\lambda<1)$ located on the interior of boundary edge e, the sequence (21) to define MLS approximation can be written in the following Taylor expansion form (22), because the values of BI weights $W_I^{BI}(\overline{\mathbf{x}})(3\le I\le N)$ are zero along the boundary edge except the BI weights $W_1^{BI}(\overline{\mathbf{x}})$ and $W_2^{BI}(\overline{\mathbf{x}})$ associated with the nodal points $\mathbf{x}_1$ and $\mathbf{x}_2$ contained in the concerned boundary edge e.

$$\mathbf{A}_k(\overline{\mathbf{x}})\mathbf{a}_k(\overline{\mathbf{x}}) = \mathbf{B}_k(\overline{\mathbf{x}})\overline{\mathbf{u}} \tag{22a}$$

where,

$$\begin{aligned}
\mathbf{A}_k(\overline{\mathbf{x}}) &= \left[\begin{array}{c:c:c} A_{11} & A_{12} & A_{13}\\ \hdashline A_{21} & A_{22} & A_{23}\\ \hdashline A_{31} & A_{32} & A_{33}\end{array}\right] + O(k^{-2})\\
A_{11} &= W_1^{BI}(\overline{\mathbf{x}}) + W_2^{BI}(\overline{\mathbf{x}})\\
&\quad + \sum_{I=1}^{N}\left(\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}\\
A_{12} &= x_1W_1^{BI}(\overline{\mathbf{x}}) + x_2W_2^{BI}(\overline{\mathbf{x}})\\
&\quad + \sum_{I=1}^{N}\left(x_I\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}\\
A_{13} &= y_1W_1^{BI}(\overline{\mathbf{x}}) + y_2W_2^{BI}(\overline{\mathbf{x}})\\
&\quad + \sum_{I=1}^{N}\left(y_I\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}\\
A_{21} &= x_1W_1^{BI}(\overline{\mathbf{x}}) + x_2W_2^{BI}(\overline{\mathbf{x}})\\
&\quad + \sum_{I=1}^{N}\left(x_I\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}\\
A_{22} &= x_1^2W_1^{BI}(\overline{\mathbf{x}}) + x_2^2W_2^{BI}(\overline{\mathbf{x}})\\
&\quad + \sum_{I=1}^{N}\left(x_I^2\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}\\
A_{23} &= x_1y_1W_1^{BI}(\overline{\mathbf{x}}) + x_2y_2W_2^{BI}(\overline{\mathbf{x}})\\
&\quad + \sum_{I=1}^{N}\left(x_Iy_I\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}
\end{aligned} \tag{22b}$$

$$
\begin{aligned}
A_{31} &= y_1 W_1^{BI}(\overline{\mathbf{x}}) + y_2 W_2^{BI}(\overline{\mathbf{x}}) \\
&\quad + \sum_{I=1}^{N} \left(y_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
A_{32} &= x_1 y_1 W_1^{BI}(\overline{\mathbf{x}}) + x_2 y_2 W_2^{BI}(\overline{\mathbf{x}}) \\
&\quad + \sum_{I=1}^{N} \left(x_I y_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
A_{33} &= y_1^2 W_1^{BI}(\overline{\mathbf{x}}) + y_2^2 W_2^{BI}(\overline{\mathbf{x}}) \\
&\quad + \sum_{I=1}^{N} \left(y_I^2 \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k}
\end{aligned}
$$

$$
\mathbf{B}_k(\overline{\mathbf{x}}) =
\left[\begin{array}{ccc:c} B_{11} & B_{12} & B_{13} & \cdots \; B_{1I} \\ \hdashline B_{21} & B_{22} & B_{23} & \cdots \; B_{2I} \\ \hdashline B_{31} & B_{32} & B_{33} & \cdots \; B_{3I} \end{array}\right] + O(k^{-2})
$$

$$
\begin{aligned}
B_{11} &= W_1^{BI}(\overline{\mathbf{x}}) + \frac{\partial W_1^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \\
B_{12} &= W_2^{BI}(\overline{\mathbf{x}}) + \frac{\partial W_2^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \\
B_{1I} &= \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \quad (3 \le I \le N) \\
B_{21} &= x_1 W_1^{BI}(\overline{\mathbf{x}}) + x_1 \frac{\partial W_1^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \\
B_{22} &= x_2 W_2^{BI}(\overline{\mathbf{x}}) + x_2 \frac{\partial W_2^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \\
B_{2I} &= x_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \quad (3 \le I \le N) \\
B_{31} &= y_1 W_1^{BI}(\overline{\mathbf{x}}) + y_1 \frac{\partial W_1^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \\
B_{32} &= y_2 W_2^{BI}(\overline{\mathbf{x}}) + y_2 \frac{\partial W_2^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \\
B_{3I} &= y_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \quad (3 \le I \le N)
\end{aligned} \tag{22c}
$$

and, where, $O(k^{-2})$ means that the remainder is the order of k^{-2}.

If we multiply γ to the third row and add α-multiple of the first row and β-multiple of the second row to the third row, then we can obtain the relation (23).

$$
\tilde{\mathbf{A}}_k(\overline{\mathbf{x}}) \mathbf{a}_k(\overline{\mathbf{x}}) = \tilde{\mathbf{B}}_k(\overline{\mathbf{x}}) \overline{\mathbf{u}} \tag{23a}
$$

where,

$$
\tilde{\mathbf{A}}_k(\overline{\mathbf{x}}) =
\left[\begin{array}{c:c:c} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} \\ \hdashline \tilde{A}_{21} & \tilde{A}_{22} & \tilde{A}_{23} \\ \hdashline \tilde{A}_{31} & \tilde{A}_{32} & \tilde{A}_{33} \end{array}\right] + O(k^{-2})
$$

$$
\begin{aligned}
\tilde{A}_{1I} &= A_{1I} \quad (1 \le I \le 3) \\
\tilde{A}_{2I} &= A_{2I} \quad (1 \le I \le 3) \\
\tilde{A}_{31} &= g(\mathbf{x}_1) W_1^{BI}(\overline{\mathbf{x}}) + g(\mathbf{x}_2) W_2^{BI}(\overline{\mathbf{x}}) \\
&\quad + \sum_{I=1}^{N} \left(g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
\tilde{A}_{32} &= x_1 g(\mathbf{x}_1) W_1^{BI}(\overline{\mathbf{x}}) + x_2 g(\mathbf{x}_2) W_2^{BI}(\overline{\mathbf{x}}) \\
&\quad + \sum_{I=1}^{N} \left(x_I g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
\tilde{A}_{33} &= y_1 g(\mathbf{x}_1) W_1^{BI}(\overline{\mathbf{x}}) + y_2 g(\mathbf{x}_2) W_2^{BI}(\overline{\mathbf{x}}) \\
&\quad + \sum_{I=1}^{N} \left(y_I g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k}
\end{aligned} \tag{23b}
$$

$$
\tilde{\mathbf{B}}_k(\overline{\mathbf{x}}) =
\left[\begin{array}{ccc:c} \tilde{B}_{11} & \tilde{B}_{12} & \tilde{B}_{13} & \cdots \; \tilde{B}_{1N} \\ \hdashline \tilde{B}_{21} & \tilde{B}_{22} & \tilde{B}_{23} & \cdots \; \tilde{B}_{2N} \\ \hdashline \tilde{B}_{31} & \tilde{B}_{32} & \tilde{B}_{33} & \cdots \; \tilde{B}_{3N} \end{array}\right] + O(k^{-2})
$$

$$
\begin{aligned}
\tilde{B}_{1I} &= B_{1I} \quad (1 \le I \le N) \\
\tilde{B}_{2I} &= B_{2I} \quad (1 \le I \le N) \\
\tilde{B}_{31} &= \left(g(\mathbf{x}_1) W_1^{BI}(\overline{\mathbf{x}}) + g(\mathbf{x}_1) \frac{\partial W_1^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \right) \\
\tilde{B}_{32} &= \left(g(\mathbf{x}_2) W_2^{BI}(\overline{\mathbf{x}}) + g(\mathbf{x}_2) \frac{\partial W_2^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \right) \\
\tilde{B}_{3I} &= g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \frac{1}{k} \quad (1 \le I \le N)
\end{aligned} \tag{23c}
$$

Here, we can assume that $x_1 \neq x_2$ without loss of generality. If not, it is sufficient to change the roles of x and y coordinates, since both of x and y coordinates for $\mathbf{x}_1$ and $\mathbf{x}_2$ cannot be same simultaneously. Also, it is noted that the solution $\mathbf{a}_k(\overline{\mathbf{x}})$ is not changed by elementary row operations.

Since $\mathbf{x}_1$ and $\mathbf{x}_2$ are located on the concerned edge e, $g(\mathbf{x}_1) = \alpha + \beta x_1 + \gamma y_1$ and $g(\mathbf{x}_2) = \alpha + \beta x_2 + \gamma y_2$ are zero in Eq. (23b) and Eq. (23c). After substituting zero for $g(\mathbf{x}_1)$ and $g(\mathbf{x}_2)$, multiplying k to the third row and taking k to be infinity yield the final form which may be solved with no diffi-

culty.

$$\tilde{\mathbf{A}}_\infty(\overline{\mathbf{x}})\mathbf{a}(\overline{\mathbf{x}}) = \tilde{\mathbf{B}}_\infty(\overline{\mathbf{x}})\overline{\mathbf{u}} \tag{24a}$$

$$\tilde{\mathbf{A}}_\infty(\overline{\mathbf{x}}) = \left[\begin{array}{c:c:c} \tilde{A}^\infty_{11} & \tilde{A}^\infty_{12} & \tilde{A}^\infty_{13} \\ \hdashline \tilde{A}^\infty_{21} & \tilde{A}^\infty_{22} & \tilde{A}^\infty_{23} \\ \hdashline \tilde{A}^\infty_{31} & \tilde{A}^\infty_{32} & \tilde{A}^\infty_{33} \end{array}\right]$$

$$\begin{aligned}
\tilde{A}^\infty_{11} &= W_1^{BI}(\overline{\mathbf{x}}) + W_2^{BI}(\overline{\mathbf{x}}) \\
\tilde{A}^\infty_{12} &= x_1 W_1^{BI}(\overline{\mathbf{x}}) + x_2 W_2^{BI}(\overline{\mathbf{x}}) \\
\tilde{A}^\infty_{13} &= y_1 W_1^{BI}(\overline{\mathbf{x}}) + y_2 W_2^{BI}(\overline{\mathbf{x}}) \\
\tilde{A}^\infty_{21} &= x_1 W_1^{BI}(\overline{\mathbf{x}}) + x_2 W_2^{BI}(\overline{\mathbf{x}}) \\
\tilde{A}^\infty_{22} &= x_1^2 W_1^{BI}(\overline{\mathbf{x}}) + x_2^2 W_2^{BI}(\overline{\mathbf{x}}) \\
\tilde{A}^\infty_{23} &= x_1 y_1 W_1^{BI}(\overline{\mathbf{x}}) + x_2 y_2 W_2^{BI}(\overline{\mathbf{x}}) \\
\tilde{A}^\infty_{31} &= \sum_{I=3}^{N} \left(g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \\
\tilde{A}^\infty_{32} &= \sum_{I=3}^{N} \left(x_I g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \\
\tilde{A}^\infty_{33} &= \sum_{I=3}^{N} \left(y_I g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right)
\end{aligned} \tag{24b}$$

$$\tilde{\mathbf{B}}_\infty(\overline{\mathbf{x}}) = \left[\begin{array}{c:c:c:c:c} W_1^{BI}(\overline{\mathbf{x}}) & W_2^{BI}(\overline{\mathbf{x}}) & 0 & \cdots & 0 \\ \hdashline x_1 W_1^{BI}(\overline{\mathbf{x}}) & x_2 W_2^{BI}(\overline{\mathbf{x}}) & 0 & \cdots & 0 \\ \hdashline 0 & 0 & \tilde{B}^\infty_{33} & \cdots & \tilde{B}^\infty_{3N} \end{array}\right] \tag{24c}$$

$$\tilde{B}^\infty_{3I} = g(\mathbf{x}_I) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \quad (1 \le I \le N)$$

Further, if we rearrange the first and second equations obtained from (24) for $a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_I + a_3(\overline{\mathbf{x}})y_I (I = 1,2)$, then we can obtain the relation (25).

$$\begin{aligned}
&\left[\begin{array}{c:c} W_1^{BI}(\overline{\mathbf{x}}) & W_2^{BI}(\overline{\mathbf{x}}) \\ \hdashline x_1 W_1^{BI}(\overline{\mathbf{x}}) & x_2 W_2^{BI}(\overline{\mathbf{x}}) \end{array}\right] \\
&\cdot \left\{\begin{array}{c} a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_1 + a_3(\overline{\mathbf{x}})y_1 \\ \hdashline a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_2 + a_3(\overline{\mathbf{x}})y_2 \end{array}\right\} \\
&= \left[\begin{array}{c:c} W_1^{BI}(\overline{\mathbf{x}}) & W_2^{BI}(\overline{\mathbf{x}}) \\ \hdashline x_1 W_1^{BI}(\overline{\mathbf{x}}) & x_2 W_2^{BI}(\overline{\mathbf{x}}) \end{array}\right] \left\{\begin{array}{c} \hat{u}^1 \\ \hdashline \hat{u}^2 \end{array}\right\}
\end{aligned} \tag{25}$$

Since x_1 is not the same as x_2 from the assumption, Eq. (25) is solvable, and the solution is given by

$$a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_I + a_3(\overline{\mathbf{x}})y_I = \hat{u}^I \quad (I = 1,2) \tag{26}$$

On the other hand, for the point $\overline{\mathbf{x}} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ $(0 < \lambda < 1)$ on the interior of boundary edge e, the value of MLS approximation can be rewritten as shown below.

For $(0 < \lambda < 1)$

$$\begin{aligned}
u^h(\overline{\mathbf{x}}) &= a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})\overline{x} + a_3(\overline{\mathbf{x}})\overline{y} \\
&= \lambda \left[a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_1 + a_3(\overline{\mathbf{x}})y_1 \right] \\
&\quad + (1-\lambda) \left[a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_2 + a_3(\overline{\mathbf{x}})y_2 \right]
\end{aligned} \tag{27}$$

Therefore, substituting Eq. (26) into Eq. (27) yields the relation (28) which is a linear combination of the given nodal values $\hat{u}^1$ and $\hat{u}^2$.

$$u^h(\overline{\mathbf{x}}) = \lambda \hat{u}^1 + (1-\lambda)\hat{u}^2 \quad (0 < \lambda < 1) \tag{28}$$

If the location of interest $\overline{\mathbf{x}}$ is $\mathbf{x}_1$(or $\mathbf{x}_2$) located at the end of edge, the derivation procedure should be changed slightly from the case of point located on the interior of edge. For the point $\overline{\mathbf{x}} = \mathbf{x}_1$, $W_2^{BI}(\overline{\mathbf{x}})$ is zero, and the second row of Eq. (24) becomes x_1-multiple of the first row. Thus, Eq. (24) becomes singular. In this case, x_1-multiple of the first row of (23) is subtracted from the second row of (23) before taking k infinity. Then one can obtain the following equation.

$$\hat{\mathbf{A}}_k(\overline{\mathbf{x}})\mathbf{a}_k(\overline{\mathbf{x}}) = \hat{\mathbf{B}}_k(\overline{\mathbf{x}})\overline{\mathbf{u}} \tag{29a}$$

where,

$$\hat{\mathbf{A}}_k(\overline{\mathbf{x}}) = \left[\begin{array}{c:c:c} \hat{A}_{11} & \hat{A}_{12} & \hat{A}_{13} \\ \hdashline \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{23} \\ \hdashline \hat{A}_{31} & \hat{A}_{32} & \hat{A}_{33} \end{array}\right] + O(k^{-2})$$

$$\begin{aligned}
\hat{A}_{11} &= W_1^{BI}(\overline{\mathbf{x}}) + \sum_{I=1}^{N} \left(\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
\hat{A}_{12} &= x_1 W_1^{BI}(\overline{\mathbf{x}}) + \sum_{I=1}^{N} \left(x_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
\hat{A}_{13} &= y_1 W_1^{BI}(\overline{\mathbf{x}}) + \sum_{I=1}^{N} \left(y_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
\hat{A}_{21} &= \sum_{I=1}^{N} \left((x_I - x_1) \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
\hat{A}_{22} &= \sum_{I=1}^{N} \left((x_I - x_1) x_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k} \\
\hat{A}_{23} &= \sum_{I=1}^{N} \left((x_I - x_1) y_I \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \right) \frac{1}{k}
\end{aligned} \tag{29b}$$

$$\hat{A}_{31} = g(\mathbf{x}_1)W_1^{BI}(\overline{\mathbf{x}}) + \sum_{I=1}^{N}\left(g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}$$

$$\hat{A}_{32} = x_1 g(\mathbf{x}_1)W_1^{BI}(\overline{\mathbf{x}}) + \sum_{I=1}^{N}\left(x_I g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}$$

$$\hat{A}_{33} = y_1 g(\mathbf{x}_1)W_1^{BI}(\overline{\mathbf{x}}) + \sum_{I=1}^{N}\left(y_I g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right)\frac{1}{k}$$

$$\hat{\mathbf{B}}_k(\overline{\mathbf{x}}) = \left[\begin{array}{c:ccc} \hat{B}_{11} & \hat{B}_{12} & \hat{B}_{13} & \cdots & \hat{B}_{1N} \\ \hdashline 0 & \hat{B}_{22} & \hat{B}_{23} & \cdots & \hat{B}_{2N} \\ \hdashline \hat{B}_{31} & \hat{B}_{32} & \hat{B}_{33} & \cdots & \hat{B}_{3N} \end{array}\right] + O(k^{-2})$$

$$\begin{aligned} \hat{B}_{11} &= W_1^{BI}(\overline{\mathbf{x}}) + \frac{\partial W_1^{BI}(\overline{\mathbf{x}})}{\partial n}\frac{1}{k} \\ \hat{B}_{1I} &= \frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\frac{1}{k} \quad (2 \le I \le N) \\ \hat{B}_{2I} &= (x_I - x_1)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\frac{1}{k} \quad (2 \le I \le N) \\ \hat{B}_{31} &= g(\mathbf{x}_1)W_1^{BI}(\overline{\mathbf{x}}) + g(\mathbf{x}_1)\frac{\partial W_1^{BI}(\overline{\mathbf{x}})}{\partial n}\frac{1}{k} \\ \hat{B}_{3I} &= g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\frac{1}{k} \quad (2 \le I \le N) \end{aligned} \tag{29c}$$

Finally, since $g(\mathbf{x}_1)$ and $g(\mathbf{x}_2)$ are zero, multiplying k to the second and third rows, and taking k to be infinity yields the equation shown below.

$$\hat{\mathbf{A}}_\infty(\overline{\mathbf{x}})\mathbf{a}(\overline{\mathbf{x}}) = \hat{\mathbf{B}}_\infty(\overline{\mathbf{x}})\overline{\mathbf{u}} \tag{30a}$$

where,

$$\hat{\mathbf{A}}_\infty(\overline{\mathbf{x}}) = \left[\begin{array}{c:c:c} W_1^{BI}(\overline{\mathbf{x}}) & x_1 W_1^{BI}(\overline{\mathbf{x}}) & y_1 W_1^{BI}(\overline{\mathbf{x}}) \\ \hdashline \hat{A}_{21}^{\infty} & \hat{A}_{22}^{\infty} & \hat{A}_{23}^{\infty} \\ \hdashline \hat{A}_{31}^{\infty} & \hat{A}_{32}^{\infty} & \hat{A}_{33}^{\infty} \end{array}\right]$$

$$\begin{aligned} \hat{A}_{21}^{\infty} &= \sum_{I=1}^{N}\left((x_I - x_1)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right) \\ \hat{A}_{22}^{\infty} &= \sum_{I=1}^{N}\left((x_I - x_1)x_I\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right) \\ \hat{A}_{23}^{\infty} &= \sum_{I=1}^{N}\left((x_I - x_1)y_I\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right) \\ \hat{A}_{31}^{\infty} &= \sum_{I=3}^{N}\left(g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right) \\ \hat{A}_{32}^{\infty} &= \sum_{I=3}^{N}\left(x_I g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right) \\ \hat{A}_{33}^{\infty} &= \sum_{I=3}^{N}\left(y_I g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n}\right) \end{aligned} \tag{30b}$$

$$\hat{\mathbf{B}}_\infty(\overline{\mathbf{x}}) = \left[\begin{array}{c:ccc} W_1^{BI}(\overline{\mathbf{x}}) & 0 & 0 & \cdots & 0 \\ \hdashline 0 & \hat{B}_{22}^{\infty} & \hat{B}_{23}^{\infty} & \cdots & \hat{B}_{2N}^{\infty} \\ \hdashline 0 & 0 & \hat{B}_{33} & \cdots & \hat{B}_{3N} \end{array}\right]$$

$$\begin{aligned} \hat{B}_{2I}^{\infty} &= (x_I - x_1)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \\ \hat{B}_{3I}^{\infty} &= g(\mathbf{x}_I)\frac{\partial W_I^{BI}(\overline{\mathbf{x}})}{\partial n} \end{aligned} \tag{30c}$$

And rearrangement of the first equation in (30) for $a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_1 + a_3(\overline{\mathbf{x}})y_1$ is written as

$$W_1^{BI}(\overline{\mathbf{x}})\left(a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_1 + a_3(\overline{\mathbf{x}})y_1\right) = W_1^{BI}(\overline{\mathbf{x}})\hat{u}^1 \tag{31}$$

where $a_1(\overline{\mathbf{x}}) + a_2(\overline{\mathbf{x}})x_1 + a_3(\overline{\mathbf{x}})y_1$ is the same as $u^h(\mathbf{x}_1)$ and $W_1^{BI}(\overline{\mathbf{x}})$ is equal to $W_1^{BI}(\mathbf{x}_1)$, since $\overline{\mathbf{x}}$ is equal to $\mathbf{x}_1$. Therefore, Eq. (31) reveals that the value of MLS approximation $u^h(\mathbf{x}_1)$ is exactly the same as $\hat{u}^1$, because $W_1^{BI}(\mathbf{x}_1)$is non-zero. The same is also true for the other node $\mathbf{x}_2$. Consequently, we can confirm that the relation (32) is satisfied along the whole boundary edge e.

$$\begin{aligned} &\forall\ \overline{\mathbf{x}} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \quad (0 \le \lambda \le 1), \\ &u^h(\overline{\mathbf{x}}) = \lambda\hat{u}^1 + (1-\lambda)\hat{u}^2 \end{aligned} \tag{32}$$

□

The relation (32) is the exact interpolation property which is not satisfied in original MLS approximation. This relation of convex combination implies more than Kronecker delta condition. Therefore the BI MLS nodal shape functions satisfy Kronecker delta condition along the boundary. However it is noted that mere Kronecker delta condition does not guarantee the relation (32) generally.

Furthermore, one can handle the boundary condition in exactly the same manner of conventional finite element method because of this property, although such is difficult only with Kronecker delta property for the nodal points like in the singular weight approach.

Similar to the value of coefficient vector $\mathbf{a}(\overline{\mathbf{x}})$, its directional derivative $d\mathbf{a}(\overline{\mathbf{x}})$ on the boundary can

be defined in the sense of limit as follows.

$$d\mathbf{a}(\overline{\mathbf{x}}) = \lim_{k\to\infty} d\mathbf{a}_k(\overline{\mathbf{x}}) \tag{33}$$

where the sequence of derivative $d\mathbf{a}_k(\overline{\mathbf{x}})$ is defined as follows.

$$\mathbf{A}_k(\overline{\mathbf{x}})d\mathbf{a}_k(\overline{\mathbf{x}}) = -d\mathbf{A}_k(\overline{\mathbf{x}})\mathbf{a}_k(\overline{\mathbf{x}}) + d\mathbf{B}_k\overline{\mathbf{u}} \tag{34}$$

It is noted that the expression for directional derivative $d\mathbf{a}(\overline{\mathbf{x}})$ can be also obtained through a similar method used to obtain the expression for value of $\mathbf{a}(\overline{\mathbf{x}})$.

4 Numerical Examples

4.1 Rectangular Domain

In this section, the proposed BI MLS approximation scheme is applied to the rectangular domain as shown in Fig. 24.

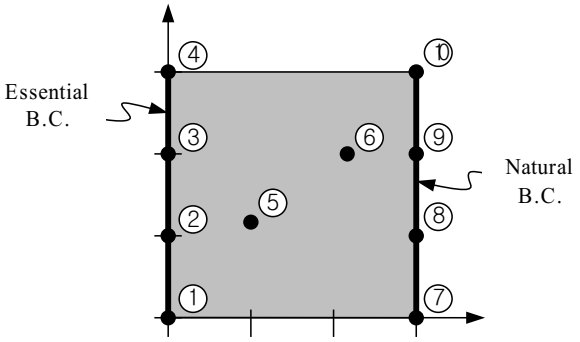

Figure 24: Rectangular domain and nodal points for MLS approximation

The total number of nodal points, to which data values $\hat{u}^I$ are assigned, is ten. The left and right sides of the model are given as essential boundary and natural boundary, respectively.

For approximation, linear monomial basis (1, x, y) and C^1 continuous weight function (16) are utilized, and the radius of support of weight function is adopted as 2.9.

The BI weight functions are constructed by using edge distance function $c_1 D_e(\mathbf{x})(1-c_2 D_e(\mathbf{x}))^2$ along with local edge finding algorithm in order that the BI weight values are zero along the edges which are included in the essential and natural boundaries. The edges in bottom and upper sides are not considered as the candidate boundary edges in the BI weight construction procedure. Therefore the BI weight values are not zero on the bottom and upper sides in this example.

In Fig. 25 and 26, the BI MLS nodal shape functions for interior nodal points are presented, and it can be observed that the values of BI MLS nodal shape functions are completely zero along the essential and natural boundary, whereas it is not zero along the upper and bottom sides.

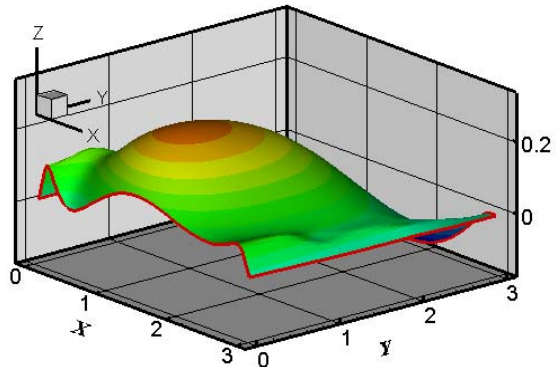

Figure 25: The 5th BI MLS nodal shape function

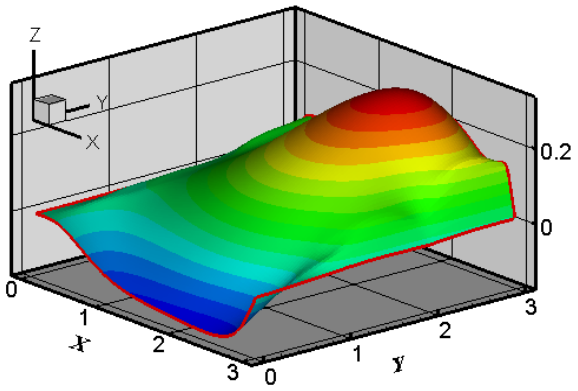

Figure 26: The 6th BI MLS nodal shape function

Fig. 27 shows the BI MLS nodal shape functions associated with the 5^{th}, 6^{th}, 7^{th}, and 8^{th} nodal points. One can observe that the functions are completely zero along the essential boundary edges.

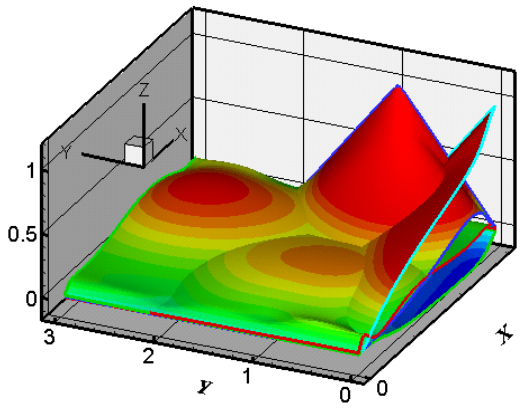

Figure 27: The 5th, 6th, 7th, and 8th BI MLS nodal shape functions

In Fig. 28, the BI MLS nodal shape functions, associated with nodal points on the natural boundary, are presented. From the results, it can be confirmed the BI MLS nodal shape functions have Kronecker delta property as predicted theoretically in previous section.

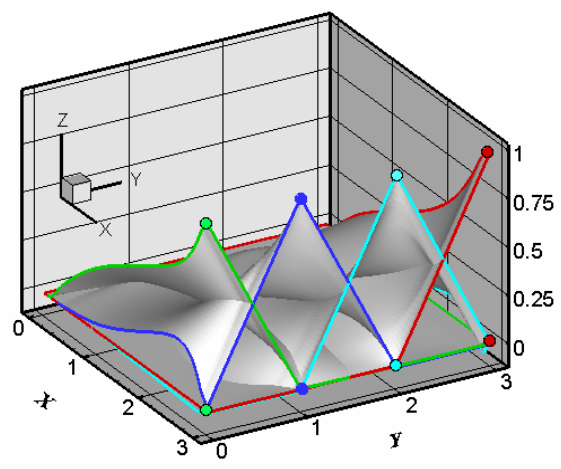

Figure 28: The 7th, 8th, 9th, and 10th BI MLS nodal shape functions along the natural boundary edges

Further, it can be known that one may enforce the natural boundary condition as well as the essential boundary condition in exactly the same manner of traditional finite element method.

4.2 Complex Shaped Domain

To demonstrate practical applicability and usefulness of the BI MLS approximation scheme, the proposed scheme is applied to the complex shaped domain which looks like human face. In Fig. 29 the domain and nodal points are presented.

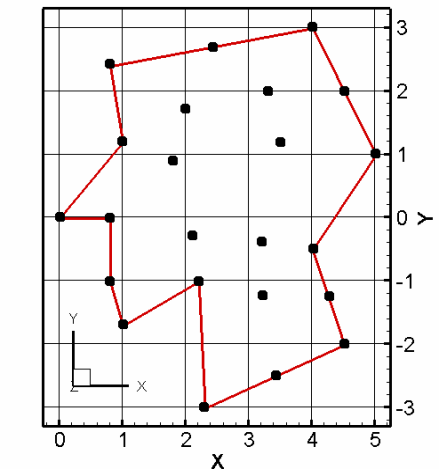

Figure 29: Domain and nodal points for BI MLS approximation

In BI MLS approximation, linear monomial basis (1, x, y) and C^1 continuous weight function (16) are utilized, and the radius of support of weight is chosen as 2.0. The BI weight functions are constructed by using edge distance function $c_1 D_e(\mathbf{x})(1 - c_2 D_e(\mathbf{x}))^2$ and local edge finding algorithm, and all of the boundary edges are considered as candidate boundary edges.

In Fig. 30, the BI MLS nodal shape function for interior nodal point is presented, and one can observe that the interior BI MLS nodal shape function becomes completely zero along the boundary. Fig. 31 shows the BI MLS nodal shpae functions associated with the nodal points located on the boundary. Likewise in previous example, it can be confirmed numerically that the BI MLS shape function on the boundary edge is the same as the linear Lagrange interpolation along the boundary, as predicted in theoretical approach.

From the numerical observation on the complicated model, it is identified that the pro-

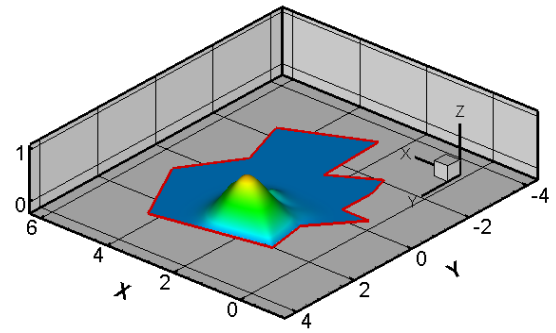

Figure 30: BI MLS nodal shape function associated with interior nodal point

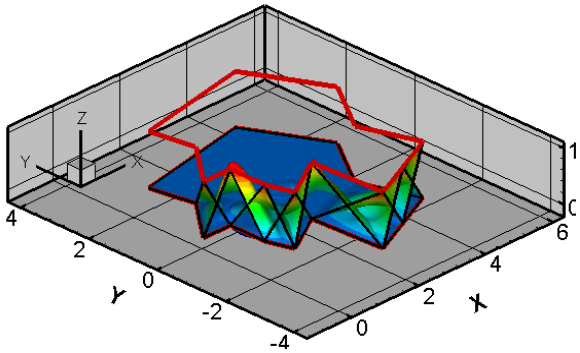

Figure 31: BI MLS nodal shape function associated with boundary nodal points

posed BI MLS approximation scheme can be also efficiently utilized for complex geometric domain, in order to construct nodal shape functions, which satisfy Kronecker delta condition along the boundary, without mesh.

Furthermore, it can be known that there is no intrinsic trouble in dealing with the model with cuts along which the functions may have jumps, since both of arbitrary convexity and concavity can be handled by the proposed scheme with no difficulty.

Additionally, it is noted that the situation is similar to the given examples although there are a lot of nodal points in the problem domain, because usually the size of support of weight is adopted to be a multiple of nodal distance and the number of nodal points contained in the support of weight for each nodal point is maintained with no regard to the total number of nodal points in problem domain.

5 Conclusions

In this work, a novel approach is proposed in order to make meshless nodal shape functions which satisfy Kronecker delta condition along the boundary, after investigating the critical characteristics of original moving least squares approximation. In the proposed approach, the original MLS weight is modified to BI(boundary interpolatable) weight. In BI weight function, its values are zero along the boundary edges except the edges containing the corresponding nodal point. And consequently the resulting BI MLS nodal shape function becomes zero along the boundary edges if the edges do not contain the corresponding nodal point, since the support of MLS weight is exactly the same as the support of MLS nodal shape function.

For constructing the BI weight efficiently, concept of edge distance function is introduced, and the algorithm to construct the BI weight is presented. The algorithm can be applied to 2D and 3D as well as 1D.

Because the number of non-zero weights is not sufficient to calculate the boundary value of BI MLS function through the direct inversion of matrix, the boundary value of BI MLS function is defined in the sense of limit. And based on the definition, it is proved theoretically that the BI MLS approximation scheme has the exact interpolation property along the boundary edges, which guarantees Kronecker delta condition along the boundary edges.

Further, two test problems are worked out for rectangular model and complex shaped model in order to justify the validity and usefulness of the proposed BI MLS scheme in numerical way. Test results are the same as predicted theoretically. The BI MLS nodal shape function obtained by the proposed scheme is exactly the same as linear Lagrange interpolation along the boundary edges.

Also numerical example shows that it is sufficient to select only the limited number of edges on the essential boundary as candidate edges in BI weight construction procedure in order to fulfill Kronecker delta condition along the essential boundary.

From the theoretical and numerical results, it can be confirmed that one can handle the essential and natural boundary conditions through the proposed BI MLS scheme in exactly the same manner used in traditional finite element methods.

Acknowledgement: This work was supported by INHA UNIVERSITY Research Grant.

References

Atluri, S.N. (2005): *Methods of Computer Modeling in Engineering & the Sciences, Vol.* 1. Tech Science Press, Forsyth, GA.

Atluri, S.N.; Cho, J.Y.; Kim, H.G. (1999): Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations. *Computational Mechanics*, Vol. 24, pp. 334-347.

Atluri, S.N.; Kim, H.G.; Cho, J.Y. (1999): A critical assessment of thetruly meshless local Petrov-Galerkin (MLPG), and local boundary integral equation (LBIE) methods. *Computational Mechanics*, Vol. 24, pp. 348-372.

Atluri, S.N.; Zhu, T. (1998): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, Vol. 22, pp. 117-127.

Babuska, I.; Melenk, J. (1997): The partition of unity method. *International Journal for Numerical Methods in Engineering*, Vol. 40, pp. 727-758.

Bartle, R.; Sherbert, D. (1982): *Introduction to Real Analysis*. John Wiley & Sons, New York.

Belytschko, T.; Lu, Y.Y.; Gu, L. (1994): Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 229-256.

Chen, J.S.; Wang, H.P. (2000): New boundary condition treatment in meshfree computation of contact problems. *Computer Method in Applied Mechanics and Engineering*, Vol. 187, pp. 441-468.

Cho, J.Y.; Atluri, S.N. (2001): Analysis of shear flexible beams, using the meshless local Petrov-Galerkin method, based on a locking-free formulation. *Engineering Computations*, Vol. 18, pp. 215-240.

Cho, J.Y.; An, J.M.; Song, Y.M.; Lee, S; Choi, D.W. (2005): Coupled analysis of independently modeled finite element substructures by moving least squares displacement welding technique. *CMES: Computer Modeling in Engineering & Sciences*, Vol. 9, pp. 1-17.

Kronguz, Y.; Belytschko, T. (1996): Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Method in Applied Mechanics and Engineering*, Vol. 131, pp. 133-145.

Lancaster, P.; Salkauskas, K. (1981): Surfaces generated by moving least squares methods. *Mathematics of Computation*, Vol. 37, pp. 141-158.

Liszka, T.; Orkisz, J. (1980): The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures*, Vol. 11, pp. 83-95.

Liu, W.; Jun, S.; Zhang, Y. (1995): Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluid*, Vol. 20, pp. 1081-1106.

Lucy, L.B. (1977): A numerical approach to the testing of the fission hypothesis. *The Astro. J.*, Vol. 8, pp. 1013-1024.

Nayroles, B.; Touzot, G.; Villon, P. (1992): Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, Vol. 10, pp. 307-318.

Shepard, D. (1968): A two-dimensional function for irregularly spaced data. *Proceeding of ACM National Conference*, pp. 517-524.

Zhu, T.; Atluri, S.N. (1998): A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. *Computational Mechanics*, Vol. 21, pp. 211-222.