Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations

S. N. Atluri, J. Y. Cho, H.-G. Kim

Abstract In this paper, the conventional moving least squares interpolation scheme is generalized, to incorporate the information concerning the derivative of the field variable into the interpolation scheme. By using this generalized moving least squares interpolation, along with the MLPG (Meshless Local Petrov–Galerkin) paradigm, a new numerical approach is proposed to deal with 4th order problems of thin beams. Through numerical examples, convergence tests are performed; and problems of thin beams under various loading and boundary conditions are analyzed by the proposed method, and the numerical results are compared with analytical solutions.

Introduction

In the past decade, a considerable attention has been given to meshless computational methods, due to their flexibility in solving boundary value problems, especially in problems with discontinuities, or with moving boundaries, or with severe deformations. The driving force for research on the meshless method is the desire to minimize, or alleviate, the human labor and error involved in meshing the entire structure. As a result, several so-called meshless methods have been proposed, such as smooth particle hydrodynamics (SPH) (Lucy, 1977), diffuse element method (DEM) (Nayroles et al., 1992), element free Galerkin method (EFG) (Belytschko et al., 1994; Organ et al., 1996), reproducing kernel particle method (RKPM) (Liu et al., 1995 and 1996), hp-clouds method (Duarte and Oden, 1996), partition of unity method (PUM) (Babuška and Melenk, 1997), local boundary integral equation method (LBIE) (Zhu, Zhang, and Atluri, 1998a, b), meshless local Petrov–Galerkin method (MLPG) (Atluri and Zhu, 1998a, b). Of these, as discussed in Atluri and Zhu (1998a, b), and in Zhu, Zhang and Atluri (1998a, b), only the MLPG and LBIE methods are truly meshless.

To be a truly meshless method, the two characteristics should be guaranteed: One is a non-element interpolation technique, and the other is a non-element approach for integrating the weak form. Most of the meshless methods are based on the non-element interpolation techniques, such as the Shepard interpolation technique (Shepard, 1968), moving least square interpolation (MLS) (Lancaster and Salkauskas, 1981), reproducing kernel particle method (RKPM), and the partition of unity method (PUM), which do not need any elements for constructing the interpolation functions for the unknown variables. However, because most of the so-called meshless methods, such as the EFG, RKPM, and hp-clouds method, still require a global background mesh for numerical integration of the global weak-form, they cannot be classified as being truly meshless. From this point of view, only the recently proposed LBIE and MLPG (meshless local Petrov–Galerkin) methods can be labeled as being truly meshless. As distinct from the other meshless methods based on the global weak form, the elegant paradigm of MLPG method is based on the local symmetric weak form (LSWF). Through the LSWF, one is naturally lead to a local non-element integration in local sub-domains such as spheres, cubes, and ellipsoids, in 3-D, without any difficulty.

Because of this pioneering truly meshless nature of the MLPG method, the present work is aimed at extending the MLPG method for 4th order boundary value problems governing thin beams or thin plates. Furthermore, in dealing with 4th order boundary value problems by a meshless computational method such as the EFG method, only a few works (Krysl and Belytschko, 1995 and 1996) were reported. Although only a thin beam will be addressed in this work, it is noted that the present approach is quite general and can easily deal with 2D plate problems.

In the 4th order boundary value problems, displacement and slope boundary conditions can be imposed at the same point, while such is impossible in 2nd order boundary value problems. Therefore, it is natural to introduce the slope as another independent variable in the interpolation schemes, in the 4th order problem. Due to this necessity, the conventional moving least square interpolation scheme is generalized, to incorporate the independent slope information, and it is used as a meshless interpolation technique in the MLPG method for 4th order boundary value problems. (It should be noted that the MLPG concept is independent of a meshless interpolation technique, and it can be combined with any meshless interpolation technique, such as PUM, or RKPM). To study the accuracy of the present method, convergence tests are carried out, and several problems of thin beams under various loading and boundary conditions are analyzed. From these tests, it is confirmed that the proposed

Received 7 February 1999

S.N. Atluri (✉), J.Y. Cho, H.-G. Kim
Center for Aerospace Research & Education, 7704 Boelter Hall, School of Engineering & Applied Science, UCLA, Los Angeles, CA. 90095-1600, USA

This work was supported by the Office of Naval Research, with Dr. Y.D.S. Rajapakse as the cognizant program official, by the FAA with Mr. Dy Le as the cognizant program official.

method gives quite accurate results and shows promising characteristics in the simulation of 4th order boundary value problems.

A review of the moving least squares interpolation

To achieve a non-element type interpolation, a meshless method uses a local interpolation or approximation, to represent the trial function, with the values (or the fictitious values) of the unknown variable at some randomly located nodes. The moving least squares interpolation is one such popular scheme (along with PUM, RKPM, Shepard function, etc.) which does not need any element information. Additionally, the required smoothness of the approximation function can be easily achieved by the moving least squares interpolation technique. Due to these reasons, the moving least squares technique may be a good candidate for approximating the unknown variables in boundary value problems.

In this section, the fundamental idea of the moving least square method is briefly reviewed, and it is generalized in the next section for 4th order boundary value problems.

Consider a continuous function u defined on a domain Ω, where the (fictitious) nodal values at the scattered points $\mathbf{x}_i$ $(1 \le i \le n)$ in Ω, that enter the interpolation, are given as $\hat{u}^i$. To approximate the distribution of function u in Ω, the global approximation form $u^h(\mathbf{x})$ is defined as follows.

$$\begin{aligned} u(\mathbf{x}) \cong u^h(\mathbf{x}) &= \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}) \\ &= \sum_{i=1}^{m} p_i(\mathbf{x})a_i(\mathbf{x}), \quad \text{for all } \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})]$ is a p-basis satisfying the conditions (Lancaster and Salkauskas, 1981) as

$$\text{(i) } p_1(\mathbf{x}) = 1 \tag{2a}$$

$$\text{(ii) } p_i(\mathbf{x}) \in C^r(\Omega), \quad i = 1, \ldots, m \tag{2b}$$

$$\begin{aligned} &\text{(iii) There exists } \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m\} \subset \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \text{ such that} \\ &\quad \{\mathbf{p}(\tilde{\mathbf{x}}_1), \ldots, \mathbf{p}(\tilde{\mathbf{x}}_m)\} \text{ is a linearly independent set.} \end{aligned} \tag{2c}$$

In Eq. (2b), $C^r(\Omega)$ denotes the set of functions, whose derivatives are continuous up to the r-th degree. For example, the $(m-1)$-th order polynomial p-basis in one dimension has the following form:

$$\mathbf{p}^{\mathrm{T}}(x) = [1, x, x^2, \ldots, x^{m-1}] \tag{3}$$

In two dimensions, a quadratic polynomial p-basis is written as

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

Additionally, the paper of Atluri and Zhu (1998a) can be referred to, for other forms of the p-basis in two and three dimensional problems.

The vector $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \ldots, a_m(\mathbf{x})]^{\mathrm{T}}$ is a vector of undetermined coefficients, whose values can vary according to the position $\mathbf{x} \in \Omega$. The coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ at each position $\mathbf{x} = \bar{\mathbf{x}}$ will be determined by a local weighted least square approximation $u_{\bar{\mathbf{x}}}(\mathbf{x})$ of the function $u(\mathbf{x})$, in a sufficiently small neighborhood $nbd(\bar{\mathbf{x}})$ of $\mathbf{x} = \bar{\mathbf{x}}$.

A local approximation $u_{\bar{\mathbf{x}}}(\mathbf{x})$, for each point $\bar{\mathbf{x}} \in \Omega$, is defined as

$$u(\mathbf{x}) \cong u_{\bar{\mathbf{x}}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\bar{\mathbf{x}}), \quad \text{for all } \mathbf{x} \in nbd(\bar{\mathbf{x}}) \tag{5}$$

In order that the local approximation is the best approximation to u, in a certain least square sense, the coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ is selected as the $m \times 1$ vector that minimizes the following weighted least square discrete L_2 error norm.

$$\begin{aligned} J_{\bar{\mathbf{x}}}(\mathbf{b}) &= \sum_{i=1}^{n} w_i(\bar{\mathbf{x}})\left[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{b} - \hat{u}^i\right]^2 \\ &= [\mathbf{P}\mathbf{b} - \hat{\mathbf{u}}]^{\mathrm{T}}\mathbf{w}(\bar{\mathbf{x}})[\mathbf{P}\mathbf{b} - \hat{\mathbf{u}}] \end{aligned} \tag{6}$$

That is, the coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ is selected to satisfy the following condition.

$$J_{\bar{\mathbf{x}}}(\mathbf{a}(\bar{\mathbf{x}})) \le J_{\bar{\mathbf{x}}}(\mathbf{b}), \quad \text{for all } \mathbf{b} \in R^m \tag{7}$$

In Eq. (6), $w_i(\mathbf{x})$ is the weight function associated with the position $\mathbf{x}_i$ of node i, and $w_i(\mathbf{x})$ is greater than 0 for all $\mathbf{x}$ in the support domain (i.e., the region of non-zero values) of $w_i(\mathbf{x})$ (which can in general be a sphere, a rectangular parallelepiped, or an ellipsoid in 3-D), and n denotes the number of nodes. For example, the support domain of the weight function $w_i(\mathbf{x})$ can be taken to be a sphere in 3-D; and the weight function $w_i(\mathbf{x})$ centered at each node $\mathbf{x}_i$ is usually 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 radius R_i, in order to preserve the local character of the MLS approximation.

The matrix $\mathbf{P}$ is an $n \times m$ matrix, and $\mathbf{w}(\bar{\mathbf{x}})$ is $n \times n$ diagonal matrix written as follows.

$$\mathbf{P} = [\mathbf{p}(\mathbf{x}_1), \mathbf{p}(\mathbf{x}_2), \ldots, \mathbf{p}(\mathbf{x}_n)]^{\mathrm{T}} \tag{8}$$

$$\mathbf{w}(\bar{\mathbf{x}}) = \begin{bmatrix} w_1(\bar{\mathbf{x}}) & 0 & \cdots & 0 \\ 0 & w_2(\bar{\mathbf{x}}) & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & w_n(\bar{\mathbf{x}}) \end{bmatrix} \tag{9}$$

And the vector $\hat{\mathbf{u}}$ denotes the vector of given fictitious values $\hat{u}^i$ of variable u at nodes i $(1 \le i \le n)$ as follows.

$$\hat{\mathbf{u}}^{\mathrm{T}} = \left[\hat{u}^1, \hat{u}^2, \ldots, \hat{u}^n\right] = [u(\mathbf{x}_1), u(\mathbf{x}_2), \ldots, u(\mathbf{x}_n)] \tag{10}$$

It is noted that the $\hat{u}^i$ $(1 \le i \le n)$ are not the nodal values of the approximation function $u^h(\mathbf{x})$.

The method to approximate the function by the moving least square method is sketched in Fig. 1. At each position $\mathbf{x} = \bar{\mathbf{x}}$, a local weighted least square approximation is found by using Eqs. (5)–(7), and its coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ is used in the global approximation form (1). Actually, it is same as the moving procedure of local approximation to obtain the global approximation, as stated in the previous work (Lancaster and Salkauskus, 1981).

A generalized moving least squares interpolation

As reviewed before, it should be noted that the moving least squares approximation is based only on the information of the values (fictitious values) of the variables at

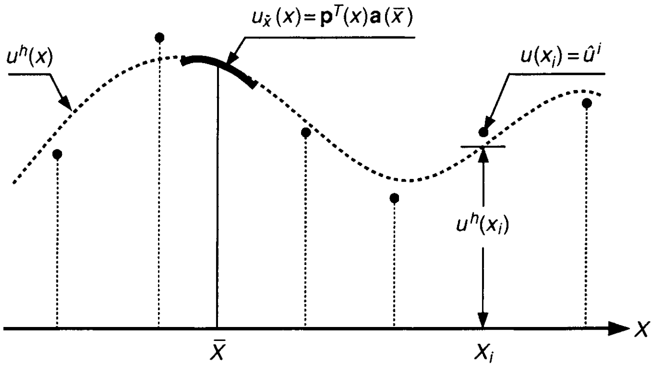

Fig. 1. Conceptual explanation of the moving least squares interpolation scheme

some scattered points. However, information concerning the derivatives of variables at some scattered points may be meaningful in some physical cases, and if they are used in an approximation procedure, it may give a better approximation result than the procedure that does not use the derivative information. From this point of view, the moving least square interpolation scheme is generalized in this paper, and the generalized approximation procedure is applied to analyze an Euler beam problem, which is a 4th order boundary value problem. We use a local symmetric weak form, and the MLPG, to construct this approximate solution.

We assume that the information concerning the derivatives up to an order l of the field variable, is available at some scattered points. Then one can expect that a reasonable interpolation procedure gives a good approximation result for the variables, as well as their derivatives up to l-th order. Therefore, it is reasonable that the derivative information be incorporated in the approximation procedure, if possible.

To approximate the function u in a domain Ω, the same global approximation form (1) as in the MLS scheme is defined in the present generalization also, and the same p-basis as in the MLS scheme, except condition (2c), is also used. However, a local approximation procedure, that is different from the MLS scheme, will be adopted to incorporate the given derivative information in the present generalized procedure.

For this purpose, the local approximation is carried out using the following weighted discrete H^l error norm instead of the weighted discrete L_2 error norm as in Eq. (6).

$$J^{(l)}_{\bar{\mathbf{x}}}(\mathbf{b}) = \sum_{i=1}^{n}\sum_{|\alpha|\le l} w_i^{(\alpha)}(\bar{\mathbf{x}})\left[D^\alpha \mathbf{p}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{b} - D^\alpha u(\mathbf{x}_i)\right]^2 \tag{11}$$

where a multi-index notation is used. It is noted that l is less than or equal to the minimum of the order of continuity of p-basis r and $(m-1)$. And the coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ at each position $\mathbf{x} = \bar{\mathbf{x}}$ is chosen so as to minimize the weighted discrete H^l error norm.

$$J^{(l)}_{\bar{\mathbf{x}}}(\mathbf{a}(\bar{\mathbf{x}})) \le J^{(l)}_{\bar{\mathbf{x}}}(\mathbf{b}), \quad \text{for all } \mathbf{b} \in R^m \tag{12}$$

Finally, the coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ obtained by the local minimization procedure is used for the global approximating function (1).

For example, the weighted discrete H^1 error norm in one-dimension assumes the form shown below.

$$\begin{aligned} J^{(1)}_{\bar{x}}(\mathbf{b}) &= \sum_{i=1}^{n}\sum_{|\alpha|\le 1} w_i^{(\alpha)}(\bar{x})\left[D^\alpha \mathbf{p}^{\mathrm{T}}(x_i)\mathbf{b} - D^\alpha u(x_i)\right]^2 \\ &= \sum_{i=1}^{n}\left\{ w_i^{(0)}(\bar{x})\left[\mathbf{p}(x_i)^{\mathrm{T}}\mathbf{b} - \hat{u}^i\right]^2 \right. \\ &\qquad \left. + w_i^{(1)}(\bar{x})\left[\frac{\mathrm{d}\mathbf{p}(x_i)^{\mathrm{T}}}{\mathrm{d}x}\mathbf{b} - \hat{\theta}^i\right]^2 \right\} \end{aligned} \tag{13}$$

where, $\hat{u}^i$, and $\hat{\theta}^i$ denote $u(x_i)$, and $\partial u(x_i)/\partial x$, respectively. Using the matrix notation, it can be rewritten as follows.

$$\begin{aligned} J^{(1)}_{\bar{x}}(\mathbf{b}) &= [\mathbf{Pb} - \hat{\mathbf{u}}]^{\mathrm{T}}\mathbf{w}^{(0)}(\bar{x})[\mathbf{Pb} - \hat{\mathbf{u}}] \\ &\quad + \left[\mathbf{P}_x\mathbf{b} - \hat{\mathbf{t}}\right]^{\mathrm{T}}\mathbf{w}^{(1)}(\bar{x})\left[\mathbf{P}_x\mathbf{b} - \hat{\mathbf{t}}\right] \\ &= \left\{\begin{bmatrix}\mathbf{P}\\ \mathbf{P}_x\end{bmatrix}\mathbf{b} - \begin{Bmatrix}\hat{\mathbf{u}}\\ \hat{\mathbf{t}}\end{Bmatrix}\right\}^{\mathrm{T}}\begin{bmatrix}\mathbf{w}^{(0)}(\bar{x}) & \mathbf{0}\\ \mathbf{0} & \mathbf{w}^{(1)}(\bar{x})\end{bmatrix} \\ &\quad \times\left\{\begin{bmatrix}\mathbf{P}\\ \mathbf{P}_x\end{bmatrix}\mathbf{b} - \begin{Bmatrix}\hat{\mathbf{u}}\\ \hat{\mathbf{t}}\end{Bmatrix}\right\} \\ &= \left[\mathbf{Qb} - \hat{\mathbf{d}}\right]^{\mathrm{T}}\mathbf{W}(\bar{x})\left[\mathbf{Qb} - \hat{\mathbf{d}}\right] \end{aligned} \tag{14}$$

where,

$$\mathbf{P}_x = \left[\frac{\partial\mathbf{p}(x_1)}{\partial x}, \frac{\partial\mathbf{p}(x_2)}{\partial x}, \ldots, \frac{\partial\mathbf{p}(x_n)}{\partial x}\right]^{\mathrm{T}} \tag{15a}$$

$$\hat{\mathbf{t}} = \left[\frac{\partial u(x_1)}{\partial x}, \frac{\partial u(x_2)}{\partial x}, \ldots, \frac{\partial u(x_n)}{\partial x}\right]^{\mathrm{T}} = \left[\hat{\theta}^1, \hat{\theta}^2, \ldots, \hat{\theta}^n\right]^{\mathrm{T}} \tag{15b}$$

$$\mathbf{w}^{(\alpha)}(\bar{x}) = \begin{bmatrix} w_1^{(\alpha)}(\bar{x}) & 0 & \cdots & 0 \\ 0 & w_2^{(\alpha)}(\bar{x}) & \vdots & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & w_n^{(\alpha)}(\bar{x}) \end{bmatrix} \tag{15c}$$

By applying the stationarity condition to the weighted discrete H^1 error norm, the coefficient vector $\mathbf{a}(\bar{x})$ can be obtained from the following matrix equation.

$$\mathbf{A}(\bar{x})\mathbf{a}(\bar{\mathbf{x}}) = \mathbf{B}(\bar{x})\hat{\mathbf{d}}$$

where,

$$\begin{aligned} \mathbf{A}(\bar{x}) &= \mathbf{Q}^{\mathrm{T}}\mathbf{W}(\bar{x})\mathbf{Q} = \mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0)}\mathbf{P} + \mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1)}\mathbf{P}_x \\ \mathbf{B}(\bar{x}) &= \mathbf{Q}^{\mathrm{T}}\mathbf{W}(\bar{x}) = \left[\mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0)}, \mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1)}\right] \end{aligned} \tag{16}$$

As in the one-dimensional case, the weighted discrete H^1 error norm in two-dimensional space has the following form:

$$J^{(1)}_{\bar{\mathbf{x}}}(\mathbf{b}) = \sum_{i=1}^{n}\sum_{|\alpha|\le 1} w_i^{(\alpha)}(\bar{\mathbf{x}})\left[D^\alpha \mathbf{p}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{b} - D^\alpha u(\mathbf{x}_i)\right]^2$$

$$= \sum_{i=1}^{n} \left\{ \begin{array}{l} w_i^{(0,0)}(\bar{\mathbf{x}})\left[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{b} - \hat{u}^i\right]^2 \\ + w_i^{(1,0)}(\bar{\mathbf{x}})\left[\frac{\partial \mathbf{p}(\mathbf{x}_i)}{\partial x}^{\mathrm{T}}\mathbf{b} - \hat{\theta}_x^i\right]^2 \\ + w_i^{(0,1)}(\bar{\mathbf{x}})\left[\frac{\partial \mathbf{p}(\mathbf{x}_i)}{\partial y}^{\mathrm{T}}\mathbf{b} - \hat{\theta}_y^i\right]^2 \end{array} \right\} \qquad (17)$$

where, $\hat{u}^i$, $\hat{\theta}_x^i$, and $\hat{\theta}_y^i$ denote $u(\mathbf{x}_i)$, $\partial u(\mathbf{x}_i)/\partial x$, and $\partial u(\mathbf{x}_i)/\partial y$, respectively. It can be also rewritten in matrix notation, as:

$$\begin{aligned} J_{\bar{\mathbf{x}}}^{(1)}(\mathbf{b}) &= [\mathbf{Pb} - \hat{\mathbf{u}}]^{\mathrm{T}}\mathbf{w}^{(0,0)}(\bar{\mathbf{x}})[\mathbf{Pb} - \hat{\mathbf{u}}] \\ &\quad + \left[\mathbf{P}_x\mathbf{b} - \hat{\mathbf{t}}_x\right]^{\mathrm{T}}\mathbf{w}^{(1,0)}(\bar{\mathbf{x}})\left[\mathbf{P}_x\mathbf{b} - \hat{\mathbf{t}}_x\right] \\ &\quad + \left[\mathbf{P}_y\mathbf{b} - \hat{\mathbf{t}}_y\right]^{\mathrm{T}}\mathbf{w}^{(0,1)}(\bar{\mathbf{x}})\left[\mathbf{P}_y\mathbf{b} - \hat{\mathbf{t}}_y\right] \\ &= \left\{ \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \\ \mathbf{P}_y \end{bmatrix}\mathbf{b} - \begin{Bmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{t}}_x \\ \hat{\mathbf{t}}_y \end{Bmatrix} \right\}^{\mathrm{T}} \\ &\quad \times \begin{bmatrix} \mathbf{w}^{(0,0)}(\bar{\mathbf{x}}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{w}^{(1,0)}(\bar{\mathbf{x}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{w}^{(0,1)}(\bar{\mathbf{x}}) \end{bmatrix} \\ &\quad \times \left\{ \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \\ \mathbf{P}_y \end{bmatrix}\mathbf{b} - \begin{Bmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{t}}_x \\ \hat{\mathbf{t}}_y \end{Bmatrix} \right\} \\ &= \left[\mathbf{Qb} - \hat{\mathbf{d}}\right]^{\mathrm{T}}\mathbf{W}(\bar{\mathbf{x}})\left[\mathbf{Qb} - \hat{\mathbf{d}}\right] \end{aligned} \qquad (18)$$

where,

$$\mathbf{P}_x = \left[\frac{\partial \mathbf{p}(\mathbf{x}_1)}{\partial x}, \frac{\partial \mathbf{p}(\mathbf{x}_2)}{\partial x}, \ldots, \frac{\partial \mathbf{p}(\mathbf{x}_n)}{\partial x}\right]^{\mathrm{T}} \qquad (19a)$$

$$\mathbf{P}_y = \left[\frac{\partial \mathbf{p}(\mathbf{x}_1)}{\partial y}, \frac{\partial \mathbf{p}(\mathbf{x}_2)}{\partial y}, \ldots, \frac{\partial \mathbf{p}(\mathbf{x}_n)}{\partial y}\right]^{\mathrm{T}} \qquad (19b)$$

$$\begin{aligned} \hat{\mathbf{t}}_x &= \left[\frac{\partial u(\mathbf{x}_1)}{\partial x}, \frac{\partial u(\mathbf{x}_2)}{\partial x}, \ldots, \frac{\partial u(\mathbf{x}_n)}{\partial x}\right]^{\mathrm{T}} \\ &= \left[\hat{\theta}_x^1, \hat{\theta}_x^2, \ldots, \hat{\theta}_x^n\right]^{\mathrm{T}} \end{aligned} \qquad (19c)$$

$$\begin{aligned} \hat{\mathbf{t}}_y &= \left[\frac{\partial u(\mathbf{x}_1)}{\partial y}, \frac{\partial u(\mathbf{x}_2)}{\partial y}, \ldots, \frac{\partial u(\mathbf{x}_n)}{\partial y}\right]^{\mathrm{T}} \\ &= \left[\hat{\theta}_y^1, \hat{\theta}_y^2, \ldots, \hat{\theta}_y^n\right]^{\mathrm{T}} \end{aligned} \qquad (19d)$$

By using the same minimization procedure as in the one-dimensional case, the coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ can be obtained from the following matrix equation.

$$\mathbf{A}(\bar{\mathbf{x}})\mathbf{a}(\bar{\mathbf{x}}) = \mathbf{B}(\bar{\mathbf{x}})\hat{\mathbf{d}}$$

where,

$$\begin{aligned} \mathbf{A}(\bar{\mathbf{x}}) &= \mathbf{Q}^{\mathrm{T}}\mathbf{W}(\bar{\mathbf{x}})\mathbf{Q} \\ &= \mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0,0)}\mathbf{P} + \mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1,0)}\mathbf{P}_x + \mathbf{P}_y^{\mathrm{T}}\mathbf{w}^{(0,1)}\mathbf{P}_y \\ \mathbf{B}(\bar{\mathbf{x}}) &= \mathbf{Q}^{\mathrm{T}}\mathbf{W}(\bar{\mathbf{x}}) = \left[\mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0,0)}, \mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1,0)}, \mathbf{P}_y^{\mathrm{T}}\mathbf{w}^{(0,1)}\right] \end{aligned} \qquad (20)$$

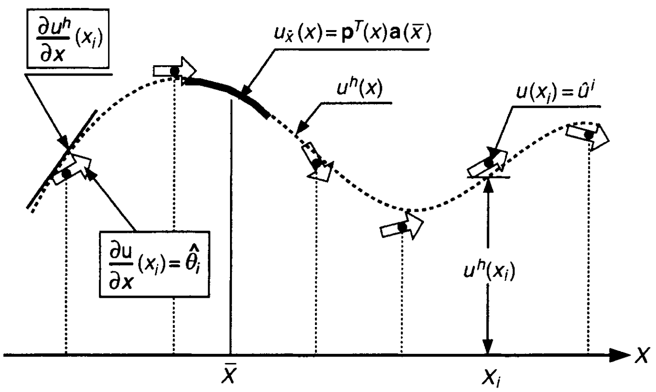
$\frac{\partial u^h}{\partial x}(x_i)$
$u_{\bar{x}}(x) = \mathbf{p}^T(x)\mathbf{a}(\bar{x})$
$u^h(x)$
$u(x_i) = \hat{u}^i$
$\frac{\partial u}{\partial x}(x_i) = \hat{\theta}_i$
$u^h(x_i)$
$\bar{X}$
X_i
X

Fig. 2. Conceptual explanation of a generalized moving least squares interpolation scheme

The GMLS (Generalized MLS) approximation is well defined only when the matrix A in Eqs. (16) and (20) is non-singular as in the MLS approximation. To guarantee a non-singular matrix A, it is necessary that the rank of Q be greater than or equal to the number of p-basis m, and at least m diagonal elements of weight function matrix $\mathbf{W}(\bar{\mathbf{x}})$ are non-zero. The approximation procedure is conceptually explained in Fig. 2.

The nodal basis functions from GMLS interpolation procedure

In the one-dimensional case, solving for $\mathbf{a}(x)$ from Eq. (16), and substituting it into Eq. (1), gives a relation which may be written in the form of a linear combination of nodal shape functions similar to that used in finite element method, as

$$\begin{aligned} u^h(x) &= \Psi_u^{\mathrm{T}}(x)\hat{\mathbf{u}} + \Psi_\theta^{\mathrm{T}}(x)\hat{\mathbf{t}} \\ &= \sum_{i=1}^{n} \hat{u}^i\psi_i^{(u)}(x) + \hat{\theta}^i\psi_i^{(\theta)}(x) \end{aligned} \qquad (21a)$$

where

$$\begin{aligned} \Psi_u^{\mathrm{T}}(x) &= \mathbf{p}^{\mathrm{T}}(x)\mathbf{A}^{-1}(x)\mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0)}(x) \\ \Psi_\theta^{\mathrm{T}}(x) &= \mathbf{p}^{\mathrm{T}}(x)\mathbf{A}^{-1}(x)\mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1)}(x) \end{aligned} \qquad (21b)$$

or

$$\begin{aligned} \psi_i^{(u)}(x) &= \sum_{j=1}^{m} p_j(x)\left[\mathbf{A}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0)}\right]_{ji} \\ \psi_i^{(\theta)}(x) &= \sum_{j=1}^{m} p_j(x)\left[\mathbf{A}^{-1}\mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1)}\right]_{ji} \end{aligned} \qquad (21c)$$

Similarly, the form of an interpolation function obtained from the generalized moving least squares method with discrete H^1 error norm, in two-dimensions can be written as follows.

$$\begin{aligned} u^h(\mathbf{x}) &= \Psi_u^{\mathrm{T}}(\mathbf{x})\hat{\mathbf{u}} + \Psi_{\theta_x}^{\mathrm{T}}(\mathbf{x})\hat{\mathbf{t}}_x + \Psi_{\theta_y}^{\mathrm{T}}(\mathbf{x})\hat{\mathbf{t}}_y \\ &= \sum_{i=1}^{n} \hat{u}^i\psi_i^{(u)}(\mathbf{x}) + \hat{\theta}_x^i\psi_i^{(\theta_x)}(\mathbf{x}) + \hat{\theta}_y^i\psi_i^{(\theta_y)}(\mathbf{x}) \end{aligned} \qquad (22a)$$

where

$$\begin{aligned}\Psi_u^{\mathrm{T}}(\mathbf{x}) &= \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0,0)}(\mathbf{x})\\ \Psi_{\theta_x}^{\mathrm{T}}(\mathbf{x}) &= \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1,0)}(\mathbf{x})\\ \Psi_{\theta_y}^{\mathrm{T}}(\mathbf{x}) &= \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}_y^{\mathrm{T}}\mathbf{w}^{(0,1)}(\mathbf{x})\end{aligned} \tag{22b}$$

or

$$\begin{aligned}\psi_i^{(u)}(\mathbf{x}) &= \sum_{j=1}^{m} p_j(\mathbf{x})\left[\mathbf{A}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{w}^{(0,0)}\right]_{ji}\\ \psi_i^{(\theta_x)}(\mathbf{x}) &= \sum_{j=1}^{m} p_j(\mathbf{x})\left[\mathbf{A}^{-1}\mathbf{P}_x^{\mathrm{T}}\mathbf{w}^{(1,0)}\right]_{ji}\\ \psi_i^{(\theta_y)}(\mathbf{x}) &= \sum_{j=1}^{m} p_j(\mathbf{x})\left[\mathbf{A}^{-1}\mathbf{P}_y^{\mathrm{T}}\mathbf{w}^{(0,1)}\right]_{ji}\end{aligned} \tag{22c}$$

In actual computations, various kinds of weight functions can be adopted for GMLS approximation procedure, as in the MLS approximation procedure. The required condition for the continuity of the approximating function can be easily satisfied by changing the weight function in the GMLS approximation procedure. In this work, we restrict ourselves to the weight functions that have the form of

$$w_i^{(\alpha)}(\mathbf{x}) = \begin{cases}\left(1 - \|\mathbf{x}-\mathbf{x}_i\|^2/R_i^2\right)^s, & \text{if } \|\mathbf{x}-\mathbf{x}_i\| \le R_i\\ 0, & \text{if } \|\mathbf{x}-\mathbf{x}_i\| > R_i\end{cases} \tag{23}$$

where R_i denotes the radius of support of weight function. If the derivatives of the p-basis in the GMLS approximation are continuous up to the r-th derivative, the resulting GMLS approximation function from this weight function is continuously differentiable up to the minimum of $(s-1)$ and r. One can also use other kinds of weight functions such as spline weight function or Gaussian weight function (Atluri and Zhu 1998a). However, it is noted that the weight function of (23) is infinitely differentiable at node $\mathbf{x}_i$, whereas conventional spline weight functions are not infinitely differentiable at node $\mathbf{x}_i$.

The GMLS nodal shape functions derived from minimizing the discrete H^1 error norm, in one-dimension and two-dimensions, are plotted in Figs. 3 and 4, respectively.

Local symmetric weak form of 4th order problems

The thin beam (Euler beam) equation is given by the following 4th order differential equation.

$$EIu'''' = f \quad \text{in global domain } \Omega \tag{24}$$

where u is transverse displacement, EI denotes the bending stiffness and f is distributed load over the beam. The boundary conditions are given at the global boundary, Γ, as

$$u(x) = \bar{u}(x) \text{ on } \Gamma_u, \text{ and } \frac{\partial u(x)}{\partial x} = \bar{\theta}(x) \text{ on } \Gamma_\theta \tag{25a}$$

$$M = \bar{M} \text{ on } \Gamma_M, \quad \text{and} \quad V = \bar{V} \text{ on } \Gamma_V \tag{25b}$$

where M and V denote the moment and the shear force, respectively. Γ_u, Γ_θ, Γ_M and Γ_V denote the boundary regions where displacement, slope, moment, and shear force are specified, respectively. The moment and shear force are related to the displacement through the equations:

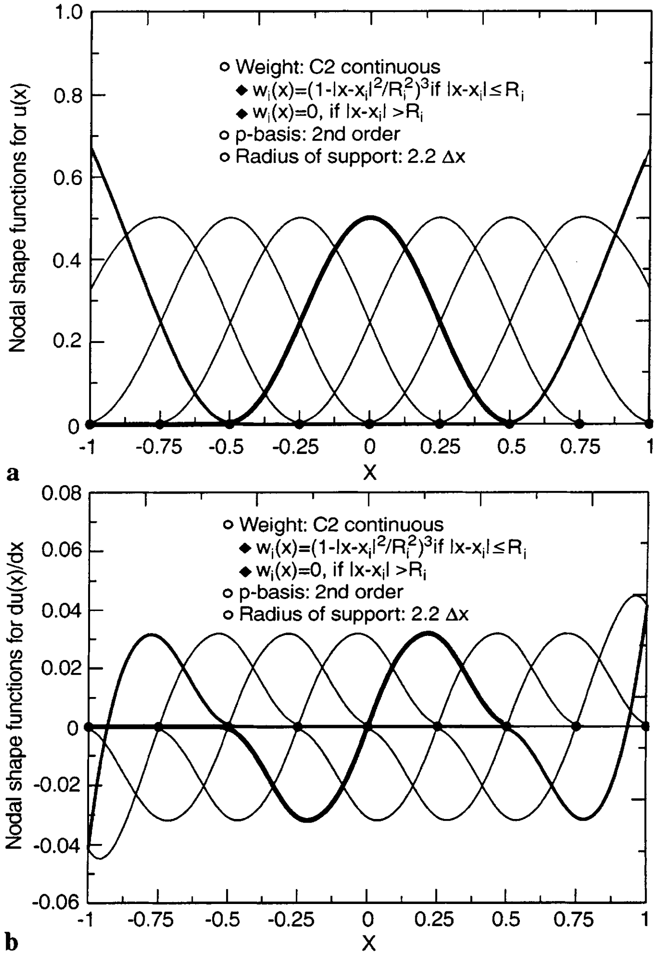

Fig. 3. **a** GMLS nodal shape functions for displacement $u(x)$, in one dimension; **b** GMLS nodal shape functions for slope $\mathrm{d}u(x)/\mathrm{d}x$, in one dimension

$$M = EIu'' \quad \text{and} \quad V = -EIu''' \tag{26}$$

Different from the other meshless methods, such as the element free Galerkin method, which are based on the global weak formulation over the entire domain Ω, a local weak form over a local sub-domain Ω_s located entirely inside the global domain Ω will be used in this study. It is noted that the local sub-domain can be of an arbitrary shape containing a point $\mathbf{x}$ in question. Even though a particular approximation of the local weak form will give the same resulting discretized equations as from the Galerkin approximation of global weak form, the local weak form will provide the clear concept for a local non-element integration, which does not need any background integration cell over the entire domain. And, it will lead to a natural way to construct the global stiffness matrix, not through the integration over a global domain, but through the integration over a local sub-domain.

To satisfy the equilibrium condition in a local subdomain Ω_s, in an average sense, the equilibrium equation is weighted by a test function v and integrated over the local subdomain. In this work, a penalty method is used to

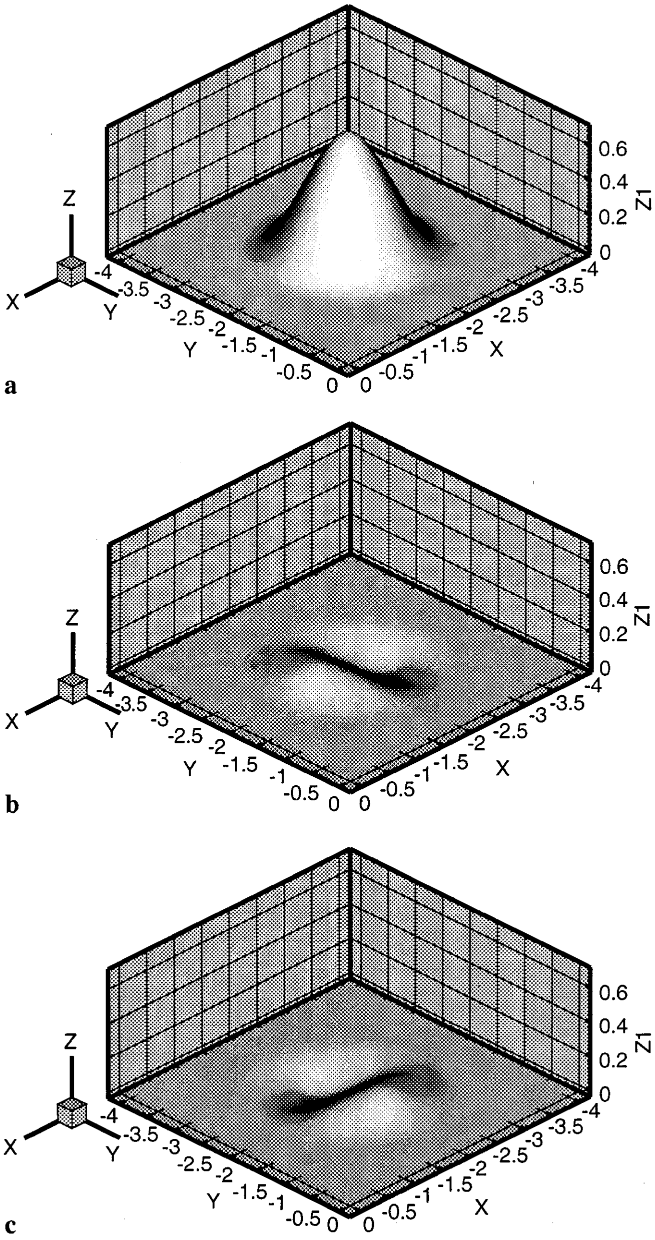

Fig. 4. **a** GMLS nodal shape function for displacement $u(\mathbf{x})$, in two dimensions. **b** GMLS nodal shape function for slope $\mathrm{d}u(\mathbf{x})/\mathrm{dx}$, in two dimensions. **c** GMLS nodal shape function for slope $\mathrm{d}u(\mathbf{x})/\mathrm{dy}$, in two dimensions

impose the essential boundary conditions (i.e., displacement and slope boundary conditions), because it is efficient and does not need any other additional unknown variables (Zhu and Atluri, 1998). The local weighted residual equation can be written as shown below.

$$0 = \int_{\Omega_s} (EIu'''' - f)v\,\mathrm{d}x \;+\; \alpha_u[(u-\bar{u})v]_{\partial\Omega_s\cap\Gamma_u} + \alpha_\theta\left[(u'-\bar{\theta})v'\right]_{\partial\Omega_s\cap\Gamma_\theta}, \quad \text{for all } v \tag{27}$$

where α_u and α_θ denote the penalty parameters to enforce displacement and slope boundary conditions, respectively. The boundary of the local sub-domain Ω_s is denoted by $\partial\Omega_s$. By using integration by parts, Eq. (27) is recast into a local symmetric weak form, as follows.

$$\begin{aligned}0 = &\int_{\Omega_s} EIu''v''\,\mathrm{d}x - \int_{\Omega_s} fv\,\mathrm{d}x - [\bar{n}EIu''v']_{\partial\Omega_s}\\ &+ [\bar{n}EIu'''v]_{\partial\Omega_s} + \alpha_u[(u-\bar{u})v]_{\partial\Omega_s\cap\Gamma_u}\\ &+ \alpha_\theta\left[(u'-\bar{\theta})v'\right]_{\partial\Omega_s\cap\Gamma_\theta}, \quad \text{for all } v\end{aligned} \tag{28}$$

where $\bar{n} = 1$ if boundary is on the right side of Ω_s, and $\bar{n} = -1$ if it is on the left side of Ω_s. Since the interior of the global domain $\mathring{\Omega}$, shear force boundary Γ_V, and displacement boundary Γ_u are mutually disjoint, and are related by $\Omega = \mathring{\Omega} \bigcup \Gamma_V \bigcup \Gamma_u$, the boundary of the sub-domain $\partial\Omega_s$ can be decomposed into disjoint subsets of $\partial\Omega_s \bigcap \mathring{\Omega}$, $\partial\Omega_s \bigcap \Gamma_V$, and $\partial\Omega_s \bigcap \Gamma_u$. By the same reason, it can be also decomposed into disjoint subsets of $\partial\Omega_s \bigcap \mathring{\Omega}$, $\partial\Omega_s \bigcap \Gamma_M$, and $\partial\Omega_s \bigcap \Gamma_\theta$. By using these decompositions, along with the boundary condition (25b) and Eq. (26), Eq. (28) can be rewritten as follows.

$$\begin{aligned}0 = &\int_{\Omega_s} EIu''v''\,\mathrm{d}x\\ &- \int_{\Omega_s} fv\,\mathrm{d}x + \alpha_u[(u-\bar{u})v]_{\partial\Omega_s\cap\Gamma_u} + \alpha_\theta\left[(u'-\bar{\theta})v'\right]_{\partial\Omega_s\cap\Gamma_\theta}\\ &- [\bar{n}\bar{M}v']_{\partial\Omega_s\cap\Gamma_M} - [\bar{n}\bar{V}v]_{\partial\Omega_s\cap\Gamma_V}\\ &- [\bar{n}EIu''v']_{\partial\Omega_s\cap\Gamma_\theta} + [\bar{n}EIu'''v]_{\partial\Omega_s\cap\Gamma_u}\\ &- [\bar{n}EIu''v']_{\partial\Omega_s\cap\mathring{\Omega}} + [\bar{n}EIu'''v]_{\partial\Omega_s\cap\mathring{\Omega}}, \quad \text{for all } v\end{aligned} \tag{29}$$

If we take the test function v whose values and derivatives are zero at $\partial\Omega_s \bigcap \mathring{\Omega}$, Eq. (29) is reduced to the following equation.

For all v such that $v = v' = 0$ at $\partial\Omega_s \cap \mathring{\Omega}$,

$$\begin{aligned}0 = &\int_{\Omega_s} EIu''v''\,\mathrm{d}x - \int_{\Omega_s} fv\,\mathrm{d}x + \alpha_u[(u-\bar{u})v]_{\partial\Omega_s\cap\Gamma_u}\\ &+ \alpha_\theta\left[(u'-\bar{\theta})v'\right]_{\partial\Omega_s\cap\Gamma_\theta} - [\bar{n}\bar{M}v']_{\partial\Omega_s\cap\Gamma_M} - [\bar{n}\bar{V}v]_{\partial\Omega_s\cap\Gamma_V}\\ &- [\bar{n}EIu''v']_{\partial\Omega_s\cap\Gamma_\theta} + [\bar{n}EIu'''v]_{\partial\Omega_s\cap\Gamma_u}\end{aligned} \tag{30}$$

The MLPG method, using the GMLS interpolation

Under the paradigm of MLPG method, the local symmetric weak form holds for arbitrary local sub-domains containing the point x in question inside the global domain Ω; and further, the shapes of local sub-domain can be chosen arbitrarily, such as spheres, cubes, and ellipsoids in 3-D. Therefore, if the nodal points x_i, and the supports of nodal shape functions for trial function are given, then the local symmetric weak form can be constructed for each local sub-domain Ω_s centered around each nodal point x_i. (It is noted that the support of nodal shape function is the same as the support of weight function in the GMLS interpolation procedure, as well as in MLS interpolation procedure.)

Because there is no restriction for the shape and size of the local sub-domains, the local sub-domain Ω_s can be taken to be different from the supports of nodal trial shape functions; and, as a special case, to be the same as the supports of nodal trial shape functions. In the MLPG method, the local sub-domain is assumed to be the support of nodal test function v, centered at a node i. If the size of local sub-domain is different from that of the support of nodal shape function for trial function; or if the nodal test function is different from the nodal trial function, the procedure becomes a Petrov–Galerkin approximation. On the other hand, if the size of local sub-domain is the same as that of the support of nodal shape function for the trial function; and, further, if exactly the same forms of nodal test and trial functions are used, it leads to the usual Galerkin approximation procedure. Therefore, the MLPG method is one of the most general methods, including the Petrov–Galerkin approximation procedure, as well as the Galerkin approximation procedure, as special cases. In this work, only the Galerkin approximation procedure is presented, even though the MLPG method is not confined to a Galerkin approximation.

We assume that the nodal points, and the sizes of supports of weight functions at each node, for the GMLS interpolation, are given. The symmetric weak form for each nodal point x_i is constructed as follows.

For all v such that $v = v' = 0$ at $\partial\Omega_s \cap \Omega$,

$$
\begin{aligned}
0 = {} & \int_{\Omega_s^{(i)}} EIu''v''\,\mathrm{d}x - \int_{\Omega_s^{(i)}} fv\,\mathrm{d}x + \alpha_u[(u-\bar{u})v]_{\partial\Omega_s^{(i)}\cap\Gamma_u} \\
& + \alpha_\theta\left[(u'-\bar{\theta})v'\right]_{\partial\Omega_s^{(i)}\cap\Gamma_\theta} - [\bar{n}\bar{M}v']_{\partial\Omega_s^{(i)}\cap\Gamma_M} - [\bar{n}\bar{V}v]_{\partial\Omega_s^{(i)}\cap\Gamma_V} \\
& - [\bar{n}EIu''v']_{\partial\Omega_s^{(i)}\cap\Gamma_\theta} + [\bar{n}EIu'''v]_{\partial\Omega_s^{(i)}\cap\Gamma_u} \qquad (31)
\end{aligned}
$$

where $\Omega_s^{(i)}$ denotes the local sub-domain, which, as a special case in the present study, is taken to be of the same size as the support of weight function $w_i(x)$ for x_i. Thus, in the present study, the local sub-domain $\Omega_s^{(i)}$ is the same as the support of the nodal trial function, as well as the support of the nodal test function.

The unknown displacement u in this local symmetric weak form is approximated by the nodal shape functions obtained through the GMLS interpolation procedure.

$$
u(x) \cong u^h(x) = \sum_{j=1}^{n} \left(\hat{u}^j \psi_j^{(u)}(x) + \hat{\theta}^j \psi_j^{(\theta)}(x) \right) \qquad (32)
$$

where $\hat{u}^j$ and $\psi_j^{(u)}(x)$ denote the fictitious nodal displacements and their corresponding GMLS nodal basis functions; and $\hat{\theta}^i$ and $\psi_i^{(\theta)}(x)$ denote the fictitious nodal slopes and their corresponding GMLS nodal shape functions. The test function v for the local sub-domain $\Omega_s^{(i)}$ is approximated by a linear combination of the nodal shape functions for nodal point x_i.

$$
v(x) \cong v^h(x) = \hat{v}^i \psi_i^{(u)}(x) + \hat{\beta}^i \psi_i^{(\theta)}(x) \quad \text{(no summation)} \qquad (33)
$$

where $\hat{v}^i$ and $\hat{\beta}^i$ denote the fictitious nodal displacement and the fictitious nodal slope, of the test function v, respectively. It is noted that the values and the derivatives of nodal test function $v^h(x)$ in Eq. (33) are zero at $\partial\Omega_s \bigcap \overset{\circ}{\Omega}$.

By substituting Eqs. (32) and (33) into the local symmetric weak form (31) gives the following discretized equation.

$$
\begin{aligned}
0 = {} & \sum_{j=1}^{n} \int_{\Omega_s^{(i)}} EI\left(\hat{v}^i \psi_i^{(u)''} + \hat{\beta}^i \psi_i^{(\theta)''}\right)\left(\hat{u}^j \psi_j^{(u)''} + \hat{\theta}^j \psi_j^{(\theta)''}\right)\mathrm{d}x \\
& - \int_{\Omega_s^{(i)}} \left(\hat{v}^i \psi_i^{(u)} + \hat{\beta}^i \psi_i^{(\theta)}\right) f\,\mathrm{d}x \\
& + \sum_{j=1}^{n} \alpha_u \Bigg[\left(\hat{v}^i \psi_i^{(u)} + \hat{\beta}^i \psi_i^{(\theta)}\right) \\
& \qquad \times \left(\hat{u}^j \psi_j^{(u)} + \hat{\theta}^j \psi_j^{(\theta)} - \bar{u}\right)\Bigg]_{\partial\Omega_s^{(i)}\cap\Gamma_u} \\
& + \sum_{j=1}^{n} \alpha_\theta \Bigg[\left(\hat{v}^i \psi_i^{(u)'} + \hat{\beta}^i \psi_i^{(\theta)'}\right) \\
& \qquad \times \left(\hat{u}^j \psi_j^{(u)'} + \hat{\theta}^j \psi_j^{(\theta)'} - \bar{\theta}\right)\Bigg]_{\partial\Omega_s^{(i)}\cap\Gamma_\theta} \\
& - \left[\left(\hat{v}^i \psi_i^{(u)'} + \hat{\beta}^i \psi_i^{(\theta)'}\right)\bar{n}\bar{M}\right]_{\partial\Omega_s^{(i)}\cap\Gamma_M} \\
& - \left[\left(\hat{v}^i \psi_i^{(u)} + \hat{\beta}^i \psi_i^{(\theta)}\right)\bar{n}\bar{V}\right]_{\partial\Omega_s^{(i)}\cap\Gamma_V} \\
& - \left[\bar{n}EI\left(\hat{v}^i \psi_i^{(u)'} + \hat{\beta}^i \psi_i^{(\theta)'}\right)\left(\hat{u}^j \psi_j^{(u)''} + \hat{\theta}^j \psi_j^{(\theta)''}\right)\right]_{\partial\Omega_s^{(i)}\cap\Gamma_\theta} \\
& + \left[\bar{n}EI\left(\hat{v}^i \psi_i^{(u)} + \hat{\beta}^i \psi_i^{(\theta)}\right)\left(\hat{u}^j \psi_j^{(u)'''} + \hat{\theta}^j \psi_j^{(\theta)'''}\right)\right]_{\partial\Omega_s^{(i)}\cap\Gamma_u}
\end{aligned}
\qquad (34)
$$

Because Eq. (34) should be satisfied for arbitrary $\hat{v}^i$ and $\hat{\beta}^i$, Eq. (34) can be rewritten as the following equation.

$$
0 = \mathbf{K}_i^{(\text{node})}\mathbf{d} + \mathbf{K}_i^{(\text{bdy})}\mathbf{d} - \mathbf{f}_i^{(\text{node})} - \mathbf{f}_i^{(bdy)} \qquad (35)
$$

where,

$$
\mathbf{d} = \left[\hat{u}^1, \hat{\theta}^1, \hat{u}^2, \hat{\theta}^2, \ldots, \hat{u}^n, \hat{\theta}^n\right]^{\mathrm{T}} \qquad (36a)
$$

$$
\mathbf{K}_i^{(\text{node})} = \left[\mathbf{k}_{i1}^{(\text{node})}, \mathbf{k}_{i2}^{(\text{node})}, \ldots, \mathbf{k}_{in}^{(\text{node})}\right] \qquad (36b)
$$

$$
\mathbf{K}_i^{(\text{bdy})} = \left[\mathbf{k}_{i1}^{(\text{bdy})}, \mathbf{k}_{i2}^{(\text{bdy})}, \ldots, \mathbf{k}_{in}^{(\text{bdy})}\right] \qquad (36c)
$$

$$
\mathbf{k}_{ij}^{(\text{node})} = EI \begin{bmatrix} \int_{\Omega_s^{(i)}} \psi_i^{(u)''} \psi_j^{(u)''}\,\mathrm{d}x & \int_{\Omega_s^{(i)}} \psi_i^{(u)''} \psi_j^{(\theta)''}\,\mathrm{d}x \\ \int_{\Omega_s^{(i)}} \psi_i^{(\theta)''} \psi_j^{(u)''}\,\mathrm{d}x & \int_{\Omega_s^{(i)}} \psi_i^{(\theta)''} \psi_j^{(\theta)''}\,\mathrm{d}x \end{bmatrix} \qquad (36d)
$$

$$
\begin{aligned}
\mathbf{k}_{ij}^{(\text{bdy})} = {} & \alpha_u \begin{bmatrix} \psi_i^{(u)}\psi_j^{(u)} & \psi_i^{(u)}\psi_j^{(\theta)} \\ \psi_i^{(\theta)}\psi_j^{(u)} & \psi_i^{(\theta)}\psi_j^{(\theta)} \end{bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_u} \\
& + \bar{n}EI \begin{bmatrix} \psi_i^{(u)}\psi_j^{(u)'''} & \psi_i^{(u)}\psi_j^{(\theta)'''} \\ \psi_i^{(\theta)}\psi_j^{(u)'''} & \psi_i^{(\theta)}\psi_j^{(\theta)'''} \end{bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_u} \\
& + \alpha_\theta \begin{bmatrix} \psi_i^{(u)'}\psi_j^{(u)'} & \psi_i^{(u)'}\psi_j^{(\theta)'} \\ \psi_i^{(\theta)'}\psi_j^{(u)'} & \psi_i^{(\theta)'}\psi_j^{(\theta)'} \end{bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_\theta}
\end{aligned}
$$

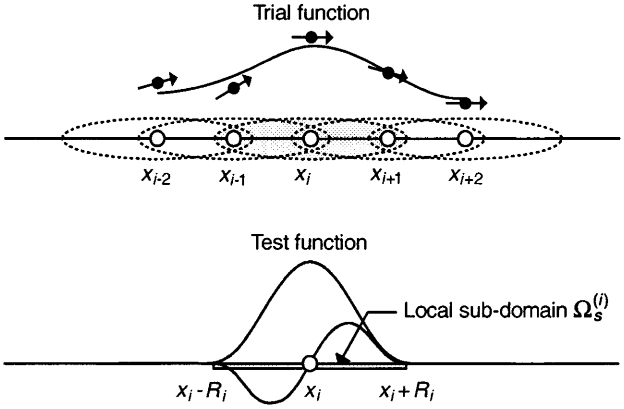

Fig. 5. The LSWF concept for constructing the discretized equation for the node i in MLPG

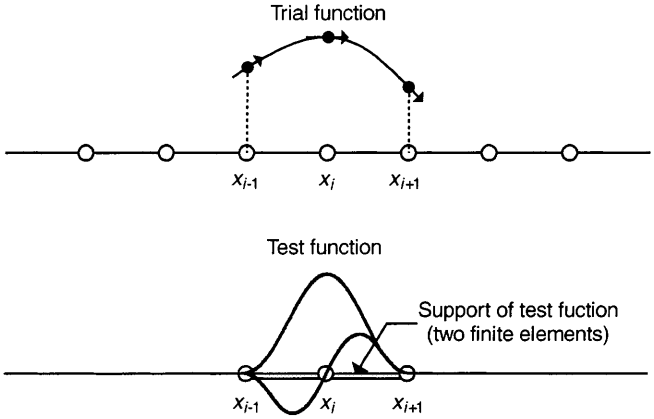

Fig. 6. The LSWF concept for constructing the discretized equation in the Galerkin finite element method

$$-\bar{n}EI\begin{bmatrix} \psi_i^{(u)'}\psi_j^{(u)''} & \psi_i^{(u)'}\psi_j^{(\theta)''} \\ \psi_i^{(\theta)'}\psi_j^{(u)''} & \psi_i^{(\theta)'}\psi_j^{(\theta)''} \end{bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_\theta} \tag{36e}$$

$$\mathbf{f}_i^{(\text{node})} = \begin{Bmatrix} \int_{\Omega_s^{(i)}} \psi_i^{(u)} f \,\mathrm{d}x \\ \int_{\Omega_s^{(i)}} \psi_i^{(\theta)} f \,\mathrm{d}x \end{Bmatrix} \tag{36f}$$

$$\mathbf{f}_i^{(\text{bdy})} = \bar{n}\bar{M}\begin{Bmatrix} \psi_i^{(u)'} \\ \psi_i^{(\theta)'} \end{Bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_M} + \bar{n}\bar{V}\begin{Bmatrix} \psi_i^{(u)} \\ \psi_i^{(\theta)} \end{Bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_V} + \alpha_u\begin{Bmatrix} \bar{u}\psi_i^{(u)} \\ \bar{u}\psi_i^{(\theta)} \end{Bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_u} + \alpha_\theta\begin{Bmatrix} \bar{\theta}\psi_i^{(u)'} \\ \bar{\theta}\psi_i^{(\theta)'} \end{Bmatrix}_{\partial\Omega_s^{(i)}\cap\Gamma_\theta} \tag{36g}$$

In this equation, one can easily notice that the numerical integration in one local sub-domain gives two equations, without any global background integration mesh. The LSWF concept for constructing the discretized equation is presented in Fig. 5. It is noted that the number of nodal variables coupled in Eq. (35) changes, according to the size of local sub-domain, while the number of coupled nodal variables in the usual finite element method for 4th order problems is six, as shown in Fig. 6.

Because the above relation (35) should hold for every local sub-domain $\Omega_s^{(i)}$, the same kind of equations can be obtained for each local sub-domain $\Omega_s^{(i)}$. Finally, we can obtain the following matrix equation for the discrete system, by collecting the equations obtained from each local sub-domain $\Omega_s^{(i)}$, without any element assembly.

$$\mathbf{Kd} = \mathbf{f} \tag{37}$$

It is noted again that it is sufficient to integrate in each local sub-domain, without any global background integration mesh, and an entry of the global stiffness matrix can be obtained directly without any element matrix assembly.

Numerical integration algorithm

In this section, the numerical integration scheme of the proposed method is discussed. Because both the GMLS nodal shape functions, as well as the MLS nodal shape functions, are not polynomial functions, it is difficult to integrate the weak form corresponding to either the GMLS shape functions or the MLS shape functions accurately, by using a conventional numerical integration scheme such as the Gaussian quadrature rule. Because of this reason, more integration points are usually needed to obtain acceptable numerical results, as compared to the usual finite element method. Moreover, an accurate integration becomes more difficult, because the Gaussian quadradure rule is based on an interpolation function that is infinitely differentiable in the integration domain, whereas the GMLS nodal shape function and MLS nodal shape function are not infinitely differentiable in the integration domain. Actually, the higher order derivatives of GMLS and MLS nodal shape functions, which are higher than the order of continuity of the weight function, are discontinuous at the support boundaries. In Fig. 7, the discontinuities, arising in the higher derivatives of GMLS nodal shape function, are presented. In the calculation of the nodal shape function in Fig. 7, a C^2 continuous weight function, and a second order polynomial p-basis are used. The C^2 continuous weight function used in this example has the form of

$$w_i^{(0)}(x) = w_i^{(1)}(x) = \begin{cases} \left(1 - |x - x_i|^2/R_i^2\right)^3, & \text{if } |x - x_i| \le R_i \\ 0, & \text{if } |x - x_i| > R_i \end{cases} \tag{38}$$

Because the derivatives of the weight function, higher than the second order, are discontinuous at the boundaries of supports of sub-domains, the derivatives higher than the second order derivative may show discontinuities at the boundaries of supports of sub-domains, as denoted by scissors in Fig. 7. To handle this difficult problem in numerical integration, an integration procedure that uses the information of all boundaries of supports of sub-domains is proposed in this work. Consider the arrangement of sub-domains shown in Fig. 8. Then the higher derivatives of the GMLS nodal shape function, in a sub-domain Ω_s, will be discontinuous at each boundary of the other support. Therefore it is natural that integrations in a sub-domain are performed in each divided sub-region, after

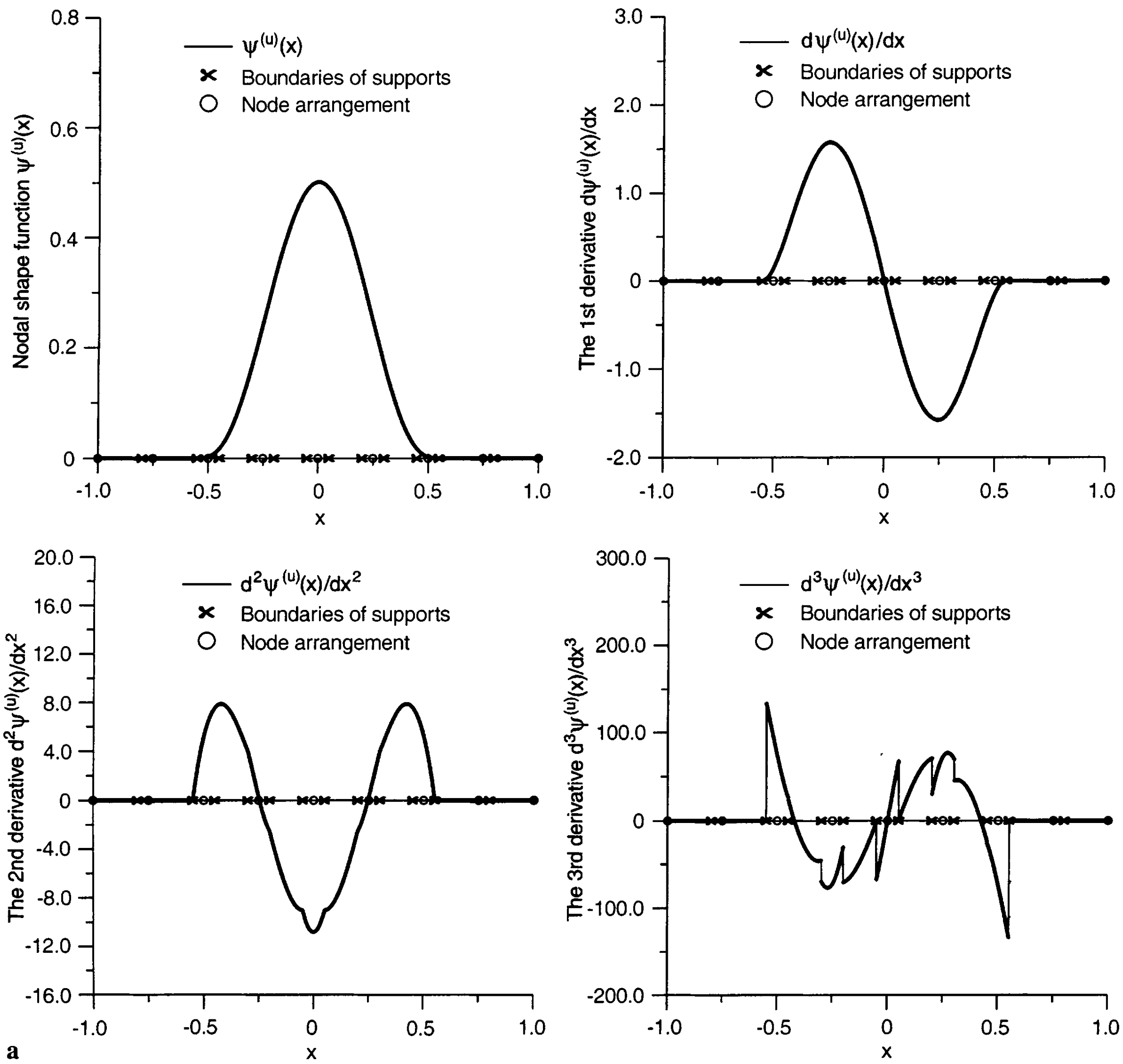

Fig. 7. **a** C^2 continuous nodal shape function, and its derivatives, for displacement. **b** C^2 continuous nodal shape function, and its derivatives, for slope

dividing the sub-domain by boundaries of supports of other nodes. To evaluate the entry of stiffness matrix $\mathbf{k}_{ij}^{(\mathrm{node})}$, related to node i and node j, it is sufficient to integrate only in the sub-regions in which both of nodal shape functions are non-zero, because the integrand is zero in the other regions. This situation is sketched in Fig. 9.

Numerical tests

By using the presently developed GMLS approximation procedure, with a local symmetric weak form and MLPG, several numerical examples are worked out to investigate the numerical characteristics of the proposed method. The calculated results are compared with analytical solutions. A 10 point Gaussian quadrature rule is used in each sub-region of intersection of sub-domains to evaluate the entry of the stiffness matrix accurately.

Convergence test

Convergence tests are carried out, for the problem of a cantilevered thin beam under a uniformly distributed load. To observe the convergence, three relative error norms are measured. They are defined as follows.

Relative L_2 error norm:

$$\frac{\sqrt{\int_\Omega |u_{\mathrm{num}} - u_{\mathrm{exact}}|^2 \, \mathrm{d}x}}{\sqrt{\int_\Omega |u_{\mathrm{exact}}|^2 \, \mathrm{d}x}} \tag{39}$$

Relative H^1 error norm:

$$\frac{\sqrt{\int_\Omega |u_{\mathrm{num}} - u_{\mathrm{exact}}|^2 + |u'_{\mathrm{num}} - u'_{\mathrm{exact}}|^2 \, \mathrm{d}x}}{\sqrt{\int_\Omega |u_{\mathrm{exact}}|^2 + |u'_{\mathrm{exact}}|^2 \, \mathrm{d}x}} \tag{40}$$

Relative H^2 error norm:

$$\frac{\sqrt{\int_\Omega |u_{\mathrm{num}} - u_{\mathrm{exact}}|^2 + |u'_{\mathrm{num}} - u'_{\mathrm{exact}}|^2 + |u''_{\mathrm{num}} - u''_{\mathrm{exact}}|^2 \, \mathrm{d}x}}{\sqrt{\int_\Omega |u_{\mathrm{exact}}|^2 + |u'_{\mathrm{exact}}|^2 + |u''_{\mathrm{exact}}|^2 \, \mathrm{d}x}} \tag{41}$$

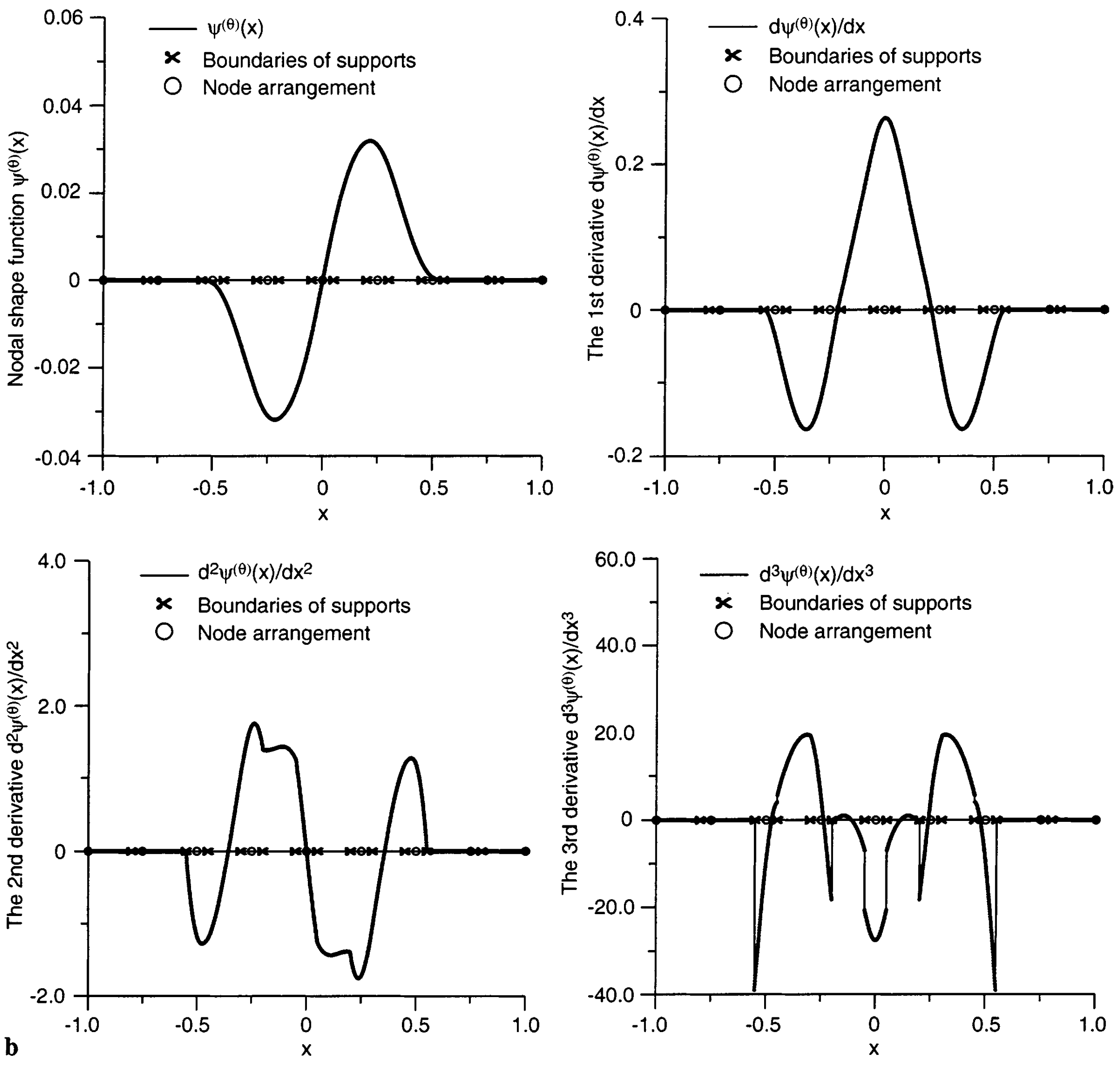

Fig. 7. Continued

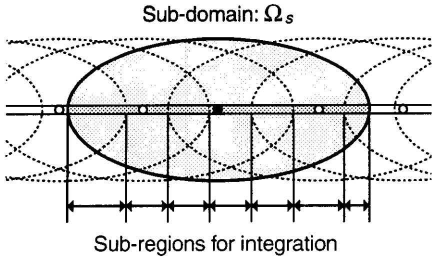

Fig. 8. Sub-domain Ω_s, divided by the boundaries of supports of other sub-domains

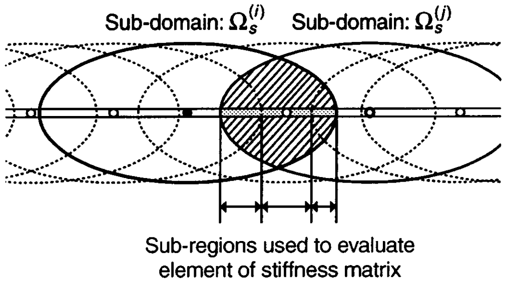

Fig. 9. Integration domain to evaluate the terms of stiffness matrix $\mathbf{k}_{ij}^{(\text{node})}$ related to node i and node j

In Fig. 10, the second order polynomial p-basis, C^2 continuous weight, and the value of 2.7 times nodal distance (2.7Δx) for radius of support, are used. The results show that the convergence rates of L_2, H^1 and H^2 error norms are 2.98, 2.62, 1.48, respectively. In Fig. 11, the order of p-basis is increased to a third order polynomial. From the results, it is observed that the convergence rate is greatly improved. Especially, the convergence rate of L_2 norm is increased from 2.98 to 4.31. However, it should be noted that the computational burden is increased, as the order of p-basis is increased, because a larger matrix **A** in Eq. (21b) should be inverted to obtain the higher order p-basis GMLS nodal shape function. In Fig. 12, the weight function is changed to a C^4 continuous weight function, holding the other parameters to be the same as those adopted in the simu-

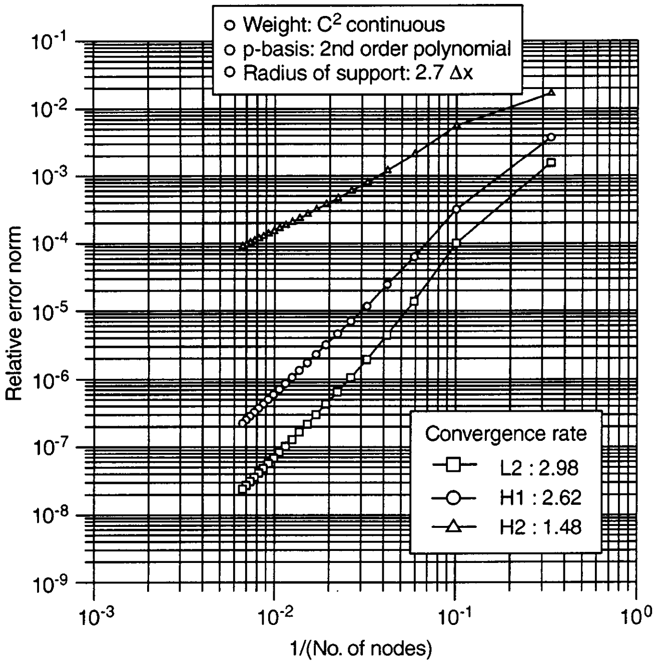

Fig. 10. Convergence rates for a cantilevered beam under uniform loading, using second order polynomial p-basis, and a C^2 continuous weight function, with the value of $2.7\Delta x$ for the radius of support

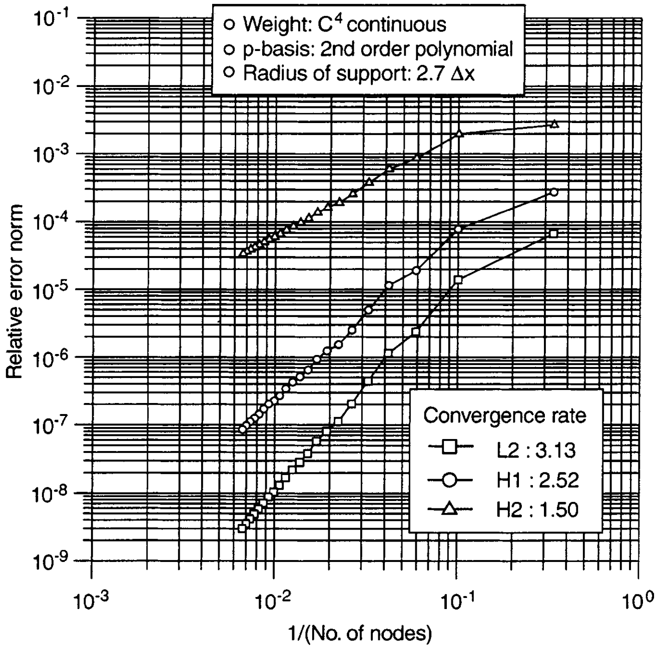

Fig. 12. Convergence rate for cantilevered beam under uniform loading, using second order polynomial p-basis, and a C^4 continuous weight function, with the value of $2.7\Delta x$ for the radius of support

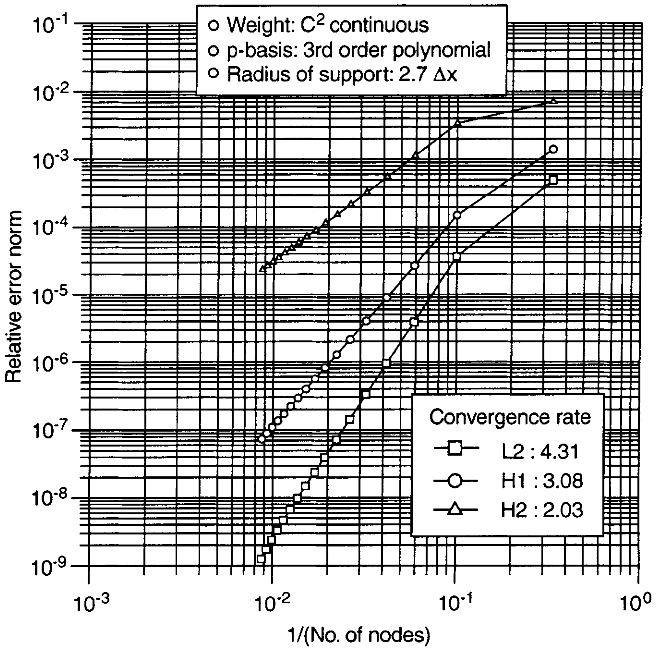

Fig. 11. Convergence rate for cantilevered beam under uniform loading, using third order polynomial p-basis, and a C^2 continuous weight function, with the value of $2.7\Delta x$ for the radius of support

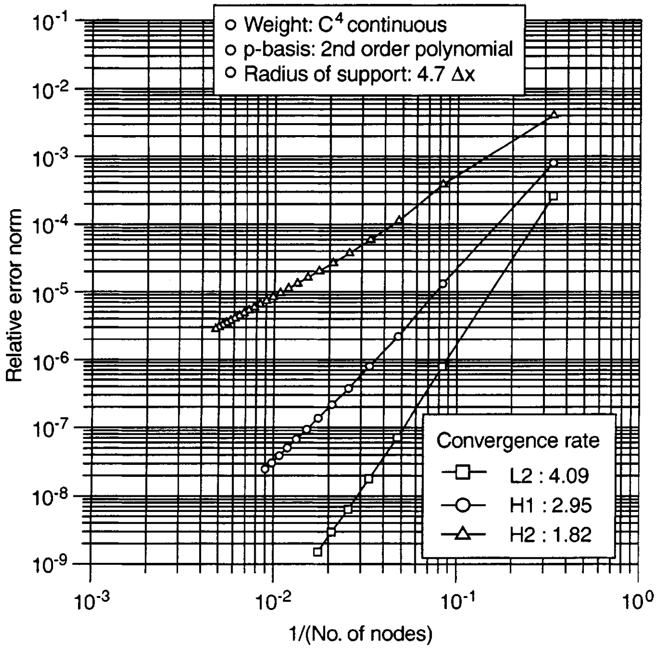

Fig. 13. Convergence rate for cantilevered beam under uniform loading, using second order polynomial p-basis, and a C^4 continuous weight function, with the value of $4.7\Delta x$ for the radius of support

lation of Fig. 10. It shows that the convergence rate is similar to the case of C^2 continuous weight function. However, the magnitude of the relative error norm becomes much smaller than in the case of a C^2 continuous weight function. It is noted that a C^4 continuous weight function can be easily used, because there is no additional computational cost to increase the order of continuity of weight function. In Fig. 13, the radius of support (sub-domain size) is increased to 4.7 times nodal distance ($4.7\Delta x$) without changing the other parameters of Fig. 12. Because more nodes are considered in one sub-domain as the radius of support is increased, both

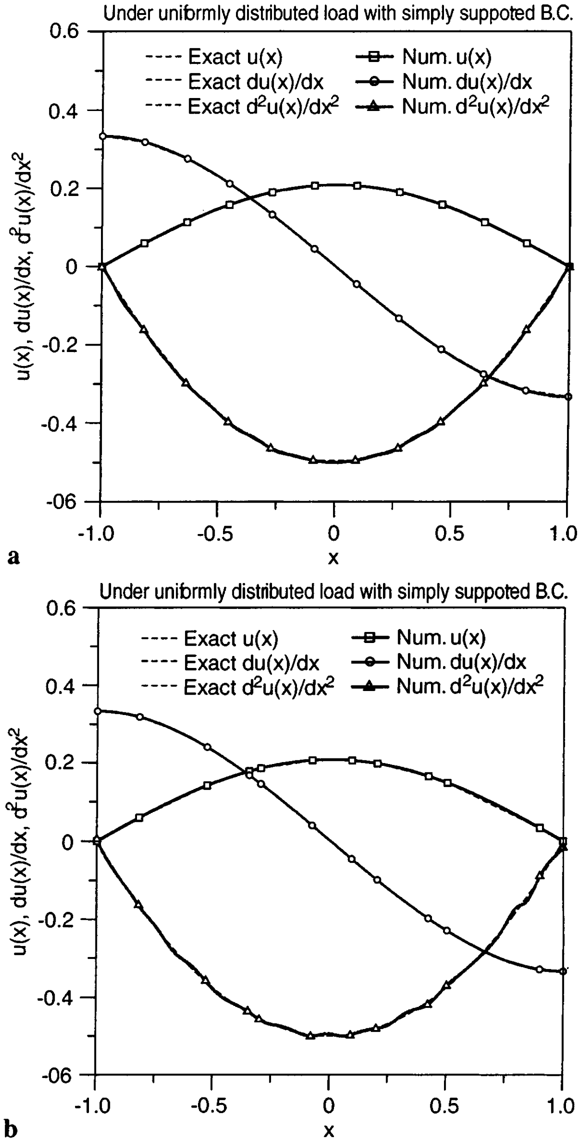

Fig. 14. **a** Numerical solution of a simply supported thin beam under uniformly distributed load, by using uniformly spaced nodes. **b** Numerical solution of a simply supported thin beam under uniformly distributed load, by using irregularly spaced nodes

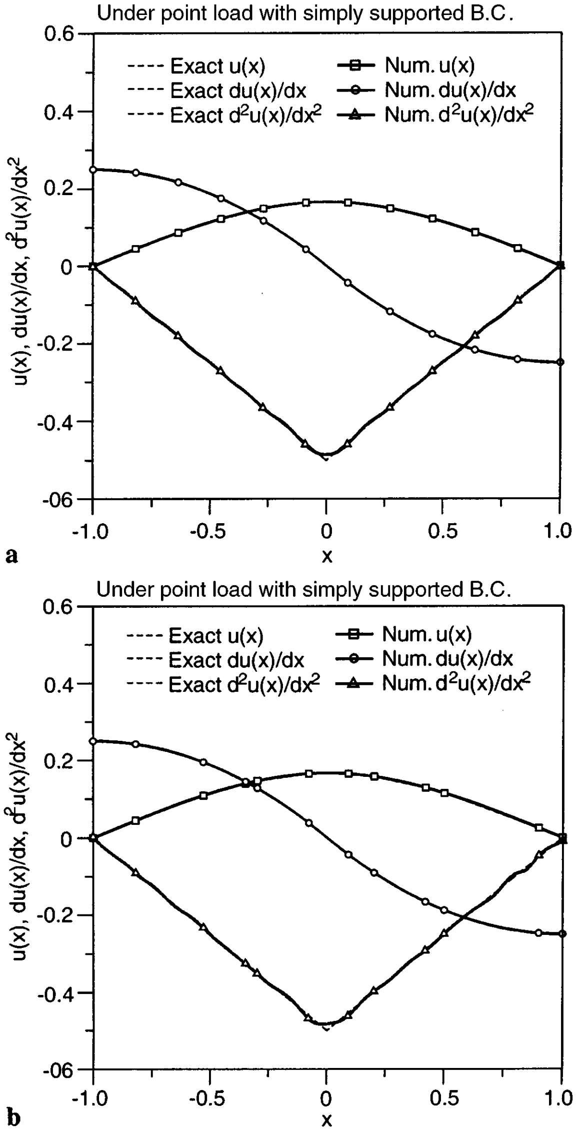

Fig. 15. **a** Numerical solution of a simply supported thin beam under a point load at the center of the beam, by using uniformly spaced nodes. **b** Numerical solution of a simply supported thin beam under a point load at the center of the beam, by using irregularly spaced nodes

the convergence rate and the magnitude of the error norm are drastically increased. The convergence rates of the L_2 error norm, the H^1 error norm, and the H^2 error norm are close to 4, 3, and 2, respectively, even though the second order polynomial p-basis is used in the computation.

Thin beam under various loading and boundary conditions

In Fig. 14, a thin beam under a uniformly distributed load is analyzed by the present method. Simply supported boundary conditions are imposed. For the simulation, the values of $2.7\Delta x$ for the radius of support of weight function; and a C^4 continuous weight function are used. The second order polynomial p-basis is adopted, and 12 nodes are used. From the results, it can be observed that both the cases of a uniform node distribution, and an irregular node distribution, give accurate results. Although the error of the second derivative in irregular node distribution is a little bit larger than in the uniform node distribution case, the case of irregular node distribution also shows excellent agreements with the exact solution.

In Fig. 15, the loading condition of Fig. 14 is changed to that of a point load at the center of the beam, while the other conditions of Fig. 14 are preserved. The simulated results also show good agreements with the analytical solution. In the vicinity of the center of the beam, the maximum error of the second derivative occurs, because of the kink in the analytical solution for the second derivative.

In Figs. 16 and 17, a thin beam with clamped boundary condition is considered. The uniformly distributed load is depicted in the case of Fig. 16, and the linearly

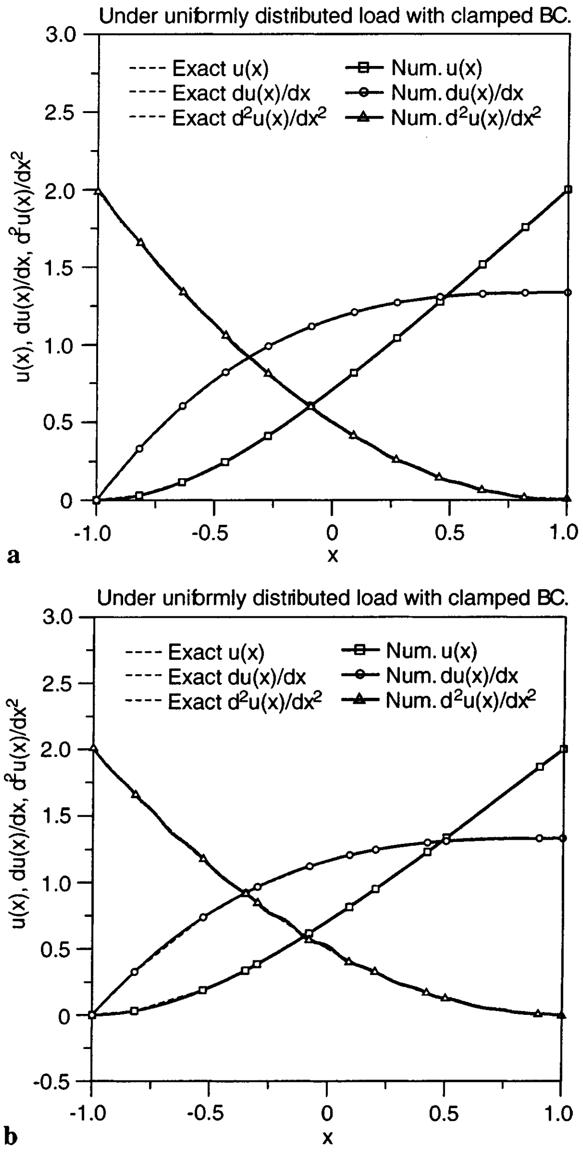

Fig. 16. a Numerical solution of a clamped thin beam under uniformly distributed load, by using uniformly spaced nodes. b Numerical solution of a clamped thin beam under uniformly distributed load, by using irregularly spaced nodes

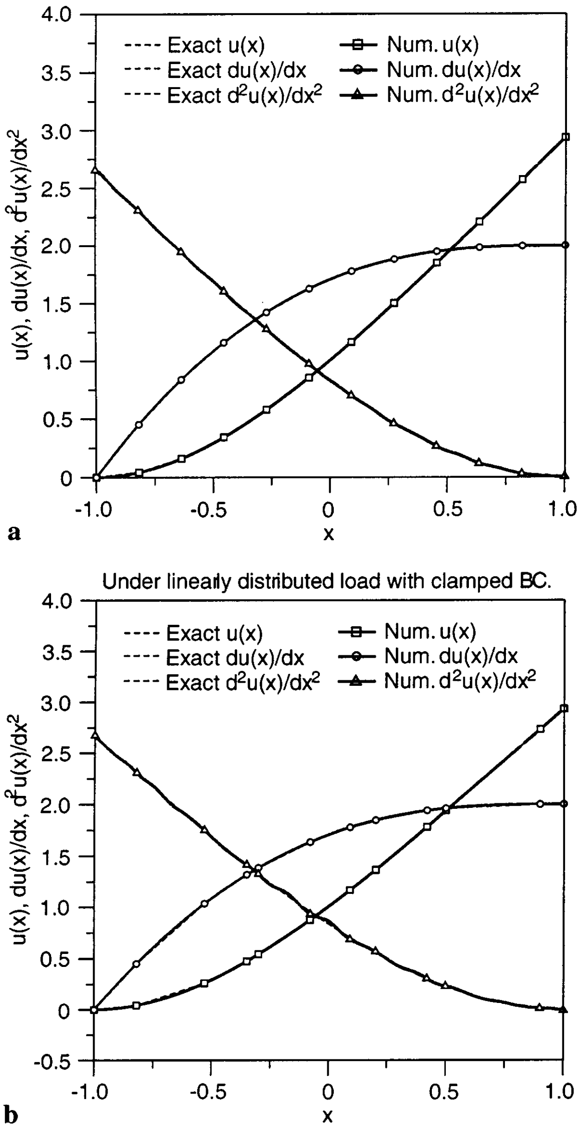

Fig. 17. a Numerical solution of a clamped thin beam under linearly distributed load, by using uniformly spaced nodes. b Numerical solution of a clamped thin beam under linearly distributed load, by using irregularly spaced nodes

distributed load is depicted in the case of Fig. 17. Similar to the case of a simply-supported boundary condition, it can be observed that the numerical solutions are highly accurate.

Concluding remarks

In this paper, the MLPG method is extended to solve the 4th order boundary problems of thin beams. For the meshless interpolation scheme in the MLPG method, the conventional moving least squares interpolation scheme is generalized. (As noted before, the MLPG concept is independent of the interpolation scheme, and it can be combined with other meshless interpolation schemes.) In this generalized moving least squares (GMLS) interpolation procedure, the slope is introduced as an additional independent variable, through the modification of local approximation in the MLS interpolation scheme. For an accurate integration of the global stiffness matrix, a new non-element local integration scheme is proposed, and it is implemented in the present numerical algorithm.

To study the accuracy of the proposed method, convergence tests are performed. Additionally, several problems of thin beams under various loading and boundary conditions are analyzed, and compared with analytical solutions. From these numerical results, it is confirmed that the proposed method shows promising characteristics in dealing with 4th order boundary value problems.

References

Atluri SN, Zhu T (1998a) A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. Comp. Mech. 22:117–127

Atluri SN, Zhu T (1998b) A new meshless local Petrov–Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. Comput. Modeling Simul. Eng. 3:187–196

Babuška I, Melenk J (1997) The partition of unity method. Int. J. Num. Meth. Eng. 40:727–758

Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. Int. J. Num. Meth. Eng. 37:229–256

Duarte CA, Oden JT (1996) An h-p adaptive method using clouds. Comp. Meth. Appl. Mech. Eng. 139:237–262

Krysl P, Belytschko T (1995) Analysis of thin plates by the element-free Galerkin method. Comp. Mech. 17:26–35

Krysl P, Belytschko T (1996) Analysis of thin shells by the element-free Galerkin method. Int. J. Solids Struct. 33:3057–3078

Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. Math. Comp. 37:141–158

Liu W, Jun S, Zhang Y (1995) Reproducing kernel particle methods. Int. J. Num. Meth. Fluids 20:1081–1106

Liu WK, Chen Y, Chang CT, Belytschko T (1996) Advances in multiple scale kernel particle methods. Comp. Mech. 18:73–111

Lucy LB (1977) A numerical approach to the testing of the fission hypothesis. The Astro. J. 8:1013–1024

Nayroles B, Touzot G, Villon P (1992) Generalizing the finite element method: diffuse approximation and diffuse elements. Comp. Mech. 10:307–318

Organ D, Fleming M, Terry T, Belytschko T (1996) Continuous meshless approximations for nonconvex bodies by diffraction and transparency. Comp. Mech. 18:225–235

Shepard D (1968) A two-dimensional function for irregularly spaced data. Proc. of ACM Nat'l Conf., pp. 517–524

Zhu T, Atluri SN (1998) A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. Comp. Mech. 21:211–222

Zhu T, Zhang JD, Atluri SN (1998a) A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Comp. Mech. 21:223–235

Zhu T, Zhang JD, Atluri SN (1998b) A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. Comp. Mech. 22:174–186