An Adaptive Mesh-Independent Numerical Integration for Meshless Local Petrov-Galerkin Method

Jin Yeon Cho*, Young Burm Jee
Department of Aerospace Engineering, College of Engineering, Inha University,
253 Yong-Hyun Dong, Nam Gu, Incheon, 402-751, Korea

In this paper, an adaptive numerical integration scheme, which does not need non-overlapping and contiguous integration meshes, is proposed for the MLPG (Meshless Local Petrov-Galerkin) method. In the proposed algorithm, the integration points are located between the neighboring nodes to properly consider the irregular nodal distribution, and the nodal points are also included as integration points. For numerical integration without well-defined meshes, the Shepard shape function is adopted to approximate the integrand in the local symmetric weak form, by the values of the integrand at the integration points. This procedure makes it possible to integrate the local symmetric weak form without any integration meshes (non-overlapping and contiguous integration domains). The convergence tests are performed, to investigate the present scheme and several numerical examples are analyzed by using the proposed scheme.

Key Words : Meshless Method, Numerical Integration, MLPG, Irregular Nodal Distribution

1. Introduction

Because of the flexibility in dealing with engineering and physical problems without geometrical meshes, meshless approaches are receiving considerable attention in the field of computational modeling and simulation of engineering problems. As a result, several methods have been proposed under the concept of a so-called meshless approach, 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), finite point method (Oñate et. al., 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 ; Atluri, Cho and Kim, 1999 ; Atluri, Kim and Cho, 1999), and the method of finite sphere (De and Bathe, 2001).

In a truly meshless approach, there are two germane aspects. One is a non-element interpolation technique. The other is a non-element approach for integrating the weak form, which was not recognized until the work of Atluri and Zhu (1998a, b). To be a true meshless, a meshless method should equip with not only a non-element interpolation scheme, but also a non-element scheme for integrating the weak form.

Of course, most of the so-called meshless methods in literature are based on non-element interpolation techniques, such as the Shepard interpolation technique (Shepard, 1968), moving least squares interpolation (MLS) (Lancaster and Salkauskas, 1981), reproducing kernel particle method (RKPM), the partition of unity method (PUM), and the generalized moving least squares

* Corresponding Author,
E-mail : cjy@inha.ac.kr
TEL : +82-32-860-7353; **FAX** : +82-32-865-5401
Department of Aerospace Engineering, College of Engineering, Inha University, 253 Yong-Hyun Dong, Nam Gu, Incheon, 402-751, Korea. (Manuscript **Received** September 5, 2002; **Revised** March 3, 2003)

interpolation (GMLS) (Atluri, Cho, and Kim, 1999a), which do not need any elements for constructing the interpolation functions for the unknown variables.

However, most of the meshless methods still rely on global meshes for integration purpose, except for the MLPG method. Additionally, contrary to the major reason for the use of a meshless method, the issue of the use of randomly distributed nodes in the problem domain has not been sufficiently dealt with in previous works. Most of the previously published papers deal with nodes that are uniformly distributed in the problem domain.

In view of this situation, an adaptive numerical integration scheme for MLPG method is proposed in this paper, in order to increase the flexibility in dealing with randomly distributed nodes, while preserving the non-element integration nature of MLPG method.

2. Moving Least Squares Interpolation

In this section, the moving least squares interpolation technique is reviewed. 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 smoothness required for the approximation function can be easily achieved by the moving least squares interpolation technique. Thanks to these merits, the moving least squares technique may be thought as a good candidate for approximating the unknown variables in boundary value problems.

Consider a continuous function u defined in 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:

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

where $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \cdots, p_m(\mathbf{x})]$ is a p-basis. For example, the $(m-1)$-th order polynomial p-basis in one dimension has the following form:

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

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

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

Similarly, a quadratic polynomial p-basis is written as

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

Meanwhile, $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \cdots, a_m(\mathbf{x})]^T$ is a vector of undetermined coefficients, whose values can vary with 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 $\bar{\mathbf{x}}$.

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

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

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 weighted least square discrete L_2-error norm defined by

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

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

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

In Eq. (6), $w_i(\mathbf{x})$ are the weight functions associated with the position $\mathbf{x}_i$ of node i such that $w_i(\mathbf{x})$ are 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 functions $w_i(\mathbf{x})$ can be spheres in 3-D ; and the weight functions $w_i(\mathbf{x})$ centered at each node $\mathbf{x}_i$ are usually adopted to be non-zero positive if the distance between nodes $\mathbf{x}_i$ and $\mathbf{x}$ is less than a specified radius R_i, but 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.

On the other hand, the $n \times m$ matrix $\mathbf{P}$ and the $n \times n$ diagonal matrix $\mathbf{W}(\bar{\mathbf{x}})$ are defined respectively by

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

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

And the fictitious values $\hat{u}^i$ of variable u at nodes i $(1 \leq i \leq n)$ are contained in the vector $\hat{\mathbf{u}}$ such that

$$\begin{aligned}\hat{\mathbf{u}}^T &= [\hat{u}^1, \hat{u}^2, \cdots, \hat{u}^n] \\ &= [u(\mathbf{x}_1), u(\mathbf{x}_2), \cdots, u(\mathbf{x}_n)]\end{aligned} \tag{10}$$

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

The method to approximate an one-dimension function by the moving least squares method is sketched in Fig. 1. At each position $\mathbf{x}=\bar{\mathbf{x}}$, the coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ for local approximation can be obtained by applying the stationarity condition to the weighted discrete L_2 error norm (6). The normal equation for minimizing the weighted discrete L_2 error norm can be written in the following form

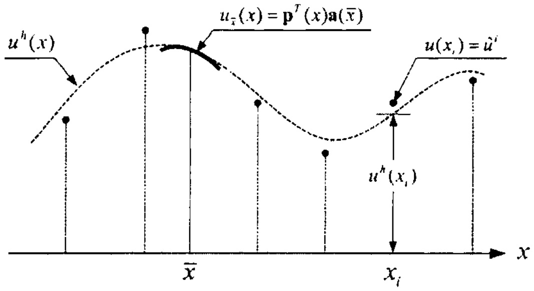


Fig. 1 Conceptual explanation of the moving least squares interpolation scheme

$$\mathbf{A}(\bar{\mathbf{x}})\mathbf{a}(\bar{\mathbf{x}}) = \mathbf{B}(\bar{\mathbf{x}})\hat{\mathbf{u}} \tag{11}$$

where

$$\begin{aligned}\mathbf{A}(\bar{\mathbf{x}}) &= \mathbf{P}^T\mathbf{W}(\bar{\mathbf{x}})\mathbf{P} \\ \mathbf{B}(\bar{\mathbf{x}}) &= \mathbf{P}^T\mathbf{W}(\bar{\mathbf{x}})\end{aligned} \tag{12}$$

The coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ is used in the global approximation form (1). For convenience, $\bar{\mathbf{x}}$ is replaced by $\mathbf{x}$ in the global approximation, because a local approximation point $\bar{\mathbf{x}}$ can be extended to all points in the entire domain. This is usually known as the concept of moving procedure.

3. Nodal Basis Functions from MLS Interpolation Procedure

Solving Eq. (11) for $\mathbf{a}(\mathbf{x})$ and substituting it into Eq. (1) give 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

$$u^h(\mathbf{x}) = \Psi^T(\mathbf{x})\hat{\mathbf{u}} = \sum_{i=1}^{n} \hat{u}^i \psi_i(\mathbf{x}) \tag{13a}$$

where

$$\Psi^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}^T\mathbf{W}(\mathbf{x}) \tag{13b}$$

or

$$\psi_i(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})[\mathbf{A}^{-1}\mathbf{P}^T\mathbf{W}]_{ji} \tag{13c}$$

In actual computations, various kinds of weight functions can be adopted for MLS approximation procedure. The condition required for the continuity of the approximating function can be easily satisfied by changing the weight function in the MLS approximation procedure. In this work, we use the 4-th order spline weight function defined by

$$w_i(\mathbf{x}) = \begin{cases} 1-6\left(\dfrac{d_i}{R_i}\right)^2+8\left(\dfrac{d_i}{R_i}\right)-3\left(\dfrac{d_i}{R_i}\right)^4, & \text{if } d_i \equiv \|\mathbf{x}-\mathbf{x}_i\| \leq R_i \\ 0 & , \text{if } d_i \equiv \|\mathbf{x}-\mathbf{x}_i\| > R_i \end{cases} \tag{14}$$

where R_i denotes the radius of support of weight function. It is noted that the 4-th order spline weight function (14) is C^1 continuous over the entire domain Ω.

4. Local Symmetric Weak form and Discretization

In this section, the local symmetric weak form is reviewed for the two-dimensional Poisson's equation :

$$\nabla^2 u(\mathbf{x}) = f(\mathbf{x}) \text{ in global domain } \Omega \tag{15}$$

where u is potential and $f(\mathbf{x})$ is a given source function. The boundary conditions are specified along the entire boundary $\Gamma = \Gamma_u \cup \Gamma_q$, as

$$u = \bar{u} \text{ on } \Gamma_u \tag{16a}$$

$$q(=\nabla u \cdot n) = \bar{q} \text{ on } \Gamma_q \tag{16b}$$

where $\bar{u}$ and $\bar{q}$ are the prescribed potential and normal flux, respectively, and $\mathbf{n}$ denotes the outward unit vector normal the boundary Γ.

In the present approach, a local weak form over a local sub-domain Ω_S will be used instead of global weak form. It is noted that the local sub-domain Ω_S is contained inside the global domain Ω, and it can be of an arbitrary shape containing a point $\mathbf{x}$ in question.

To satisfy the equilibrium condition in a local subdomain Ω_S, in the average sense, the equilibrium equation is weighted by a test function v and integrated over the local subdomain Ω_S. In this work, a penalty method is used to impose the essential boundary conditions, because it does not need any additional unknown variables (Zhu and Atluri, 1998). The local weighted residual equation can be written as

$$\begin{aligned} 0 = &\int_{\Omega_S} (\nabla^2 u - f)\, v d\Omega \\ &- \alpha_u \int_{\partial\Omega_S \cap \Gamma_u} (u - \bar{u})\, v d\Gamma \end{aligned}, \text{ for all } v \tag{17}$$

where α_u denotes a penalty parameter to enforce the essential boundary condition. The boundary of the local sub-domain Ω_S is denoted by $\partial\Omega_S$. By integrating by parts, Eq. (17) is recast into a local symmetric weak form given by

$$\begin{aligned} 0 = &-\int_{\Omega_S} \nabla u \cdot \nabla v d\Omega - \int_{\Omega_S} f v d\Omega + \int_{\partial\Omega_S} (\mathbf{n} \cdot v \nabla u)\, d\Gamma \\ &- \alpha_u \int_{\partial\Omega_S \cap \Gamma_u} (u - \bar{u})\, v d\Gamma \end{aligned}, \text{ for all } v \tag{18}$$

Since the interior of the global domain $\mathring{\Omega}$, the essential boundary Γ_u, and the flux boundary Γ_q are mutually disjoint. And, since those are related by $\bar{\Omega} = \mathring{\Omega} \cup \Gamma_u \cup \Gamma_q$, the boundary of the sub-domain $\partial\Omega_S$ can be decomposed into disjoint subsets of $\partial\Omega_S \cap \mathring{\Omega}$, $\partial\Omega_S \cap \Gamma_u$, and $\partial\Omega_S \cap \Gamma_q$. By using this decomposition, together with the boundary condition (16b), Eq. (18) can be rewritten as follows :

$$\begin{aligned} 0 = &\int_{\Omega_S} \nabla u \cdot \nabla v d\Omega + \int_{\Omega_S} f v d\Omega - \int_{\partial\Omega_S \cap \mathring{\Omega}} (\mathbf{n} \cdot v \nabla u)\, d\Gamma - \int_{\partial\Omega_S \cap \Gamma_q} v \bar{q} d\Gamma \\ &- \int_{\partial\Omega_S \cap \Gamma_u} (\mathbf{n} \cdot v \nabla u)\, d\Gamma + \alpha_u \int_{\partial\Omega_S \cap \Gamma_u} (u - \bar{u})\, v d\Gamma \end{aligned}, \text{ for all } v \tag{19}$$

If we restrict our attention to the test function v which vanishes at $\partial\Omega_S \cap \mathring{\Omega}$, then, Eq. (19) is reduced to

$$\begin{aligned} 0 = &\int_{\Omega_S} \nabla u \cdot \nabla v d\Omega + \int_{\Omega_S} f v d\Omega - \int_{\partial\Omega_S \cap \Gamma_q} v \bar{q} d\Gamma \\ &- \int_{\partial\Omega_S \cap \Gamma_u} (\mathbf{n} \cdot v \nabla u)\, d\Gamma + \alpha_u \int_{\partial\Omega_S \cap \Gamma_u} (u - \bar{u})\, v d\Gamma \end{aligned}, \text{ for all } v \tag{20}$$

Since the local symmetric weak form (20) holds for arbitrary local sub-domains containing the point $\mathbf{x}$ in question inside the global domain Ω, we can construct each local symmetric weak form for each local sub-domain Ω_S centered around each nodal point $\mathbf{x}_i$. Because there is no restriction for the shape and size of the local sub-domains, the local sub-domain Ω_S can be taken differently from the supports of nodal trial shape functions. As a special case, it can 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 furthermore, if exactly the same forms are used for nodal test and trial functions, it becomes the usual Galerkin approximation procedure. In this work, only the Galerkin approximation procedure is presented, even though the MLPG method is not limited to a Galerkin approximation.

We assume that the nodal points and the sizes of supports of weight functions at each node are given for the MLS interpolation. The symmetric weak form for each nodal point $\mathbf{x}_i$ is constructed as follows :

For all v that vanishes at $\partial\Omega_s \cap \mathring{\Omega}$,

$$0=\int_{\Omega_S^{(i)}} \nabla u \cdot \nabla v d\Omega + \int_{\Omega_S^{(i)}} f v d\Omega - \int_{\partial\Omega_S^{(i)} \cap \Gamma_q} v \bar{q} d\Gamma \qquad (21)$$
$$- \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} (\mathbf{n} \cdot v \nabla u) d\Gamma + \alpha_u \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} (u - \bar{u}) v d\Gamma$$

where $\Omega_S^{(i)}$ denotes the local sub-domain. As a special case in the present study, which is taken to be of the same size as the support of weight function $w_i(\mathbf{x})$ for $\mathbf{x}_i$. 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 variable u in this local symmetric weak form is approximated by the nodal shape functions obtained through the MLS interpolation procedure,

$$u(\mathbf{x}) \cong u^h(\mathbf{x}) = \sum_{j=1}^{n} \hat{u}^j \psi_j(\mathbf{x}) \qquad (22)$$

where $\hat{u}^j$ and $\psi_j(\mathbf{x})$ denote the fictitious nodal potential values and the corresponding MLS nodal basis 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 $\mathbf{x}_S$ such that

$$v(\mathbf{x}) \cong v^h(\mathbf{x}) = \hat{v}^i \psi_i(\mathbf{x}) \qquad (23)$$

where $\hat{v}^i$ denotes the fictitious nodal value of the test function v. It is noted that the values of nodal test function $v^i(\mathbf{x})$ in Eq. (23) are zero at $\partial\Omega_S \cap \mathring{\Omega}$.

By substituting Eqs. (22) and (23) into the local symmetric weak form (21), we obtain the following discretized equation given by

$$0=\sum_{j=1}^{n} \int_{\Omega_S^{(i)}} \hat{v}^i \nabla \psi_i \cdot \nabla \psi_j \hat{u}^j d\Omega - \sum_{j=1}^{n} \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} \hat{v}^i (\mathbf{n} \cdot \psi_i \nabla \psi_j) \hat{u}^j d\Gamma$$
$$+ \sum_{j=1}^{n} \alpha_u \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} \hat{v}^i \psi_i \psi_j \hat{u}^j d\Gamma + \int_{\partial\Omega_S^{(i)}} \hat{v}^i \psi_i f d\Omega \qquad (24)$$
$$- \int_{\partial\Omega_S^{(i)} \cap \Gamma_q} \hat{v}^i \psi_i \bar{q} d\Gamma - \alpha_u \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} \hat{v}^i \psi_i \bar{u} d\Gamma, \text{ for all } \hat{v}^i$$

Because Eq. (24) is satisfied for arbitrary $\hat{v}^i$, Eq. (24) can be rewritten in the form of

$$\mathrm{K}_i^{(node)} \hat{\mathbf{u}} = \mathbf{f}_i^{node} \text{ in } \Omega_S^{(i)} \qquad (25a)$$

$$\sum_{j=1}^{n} \left(\int_{\Omega_S^{(i)}} \nabla \psi_i \cdot \nabla \psi_j d\Omega - \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} (\mathbf{n} \cdot \psi_i \nabla \psi_j) d\Gamma + \alpha_u \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} \psi_i \psi_j d\Gamma \right) \hat{u}^j$$
$$= - \int_{\Omega_S^{(i)}} \psi_i f d\Omega + \int_{\partial\Omega_S^{(i)} \cap \Gamma_q} \psi_i \bar{q} d\Gamma + \alpha_u \int_{\partial\Omega_S^{(i)} \cap \Gamma_u} \psi_i \bar{u} d\Gamma \qquad (25b)$$

By collecting the equations for individual each sub-domains $\Omega_S^{(i)}$, we can construct the matrix equation

$$\mathbf{K}\hat{\mathbf{u}} = \mathbf{f} \qquad (26)$$

5. Numerical Integration Algorithm

In this section, an adaptive mesh-independent numerical integration procedure is proposed for the numerical integration of the local symmetric weak form in the MLPG method. It is well known that the prerequisites for numerical integration are the determination of locations of sampling points, the corresponding weights, and the corresponding shapes of the integration domains. The integration domains can be non-overlapping or overlapping with each other. Depending upon this character of integration domains, meshless methods may be classified into two categories.

In the MLPG method (Atluri and Zhu, 1998a, b), the supports of nodal test functions are taken as integration domains which overlap with each other, and regularly distributed integration points in the sub-domains are used as shown in Fig. 2. This integration procedure has many advantages, such as the true meshless character (Atluri and Zhu, 1998a). However, it has some inflexibility in the sampling of integration points when we deal with randomly distributed nodes, because the sampling points for integration are always located regularly in the integration domain, irrespective of nodal distribution.

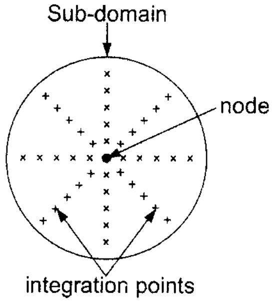


Fig. 2 Regular arrangement of integration points over a sub-domain

To increase the flexibility in MLPG method, without sacrificing its non-element integration character, it is desirable to locate the integration points by taking into account the non-uniformity of the given nodal distribution, while preserving the meshless nature at the same time that does not need integration meshes (non-overlapping integration domains). If the integration points are sampled adequately to consider a given irregular nodal distribution, the next remaining question is how we can define the integration domains for sampling points, without using integration meshes. One of the answers to this query may be the non-element interpolation technique itself, such as the Shepard or the MLS (moving least squares) interpolations. In this line of thought, an alternative non-element numerical integration procedure is proposed here.

Let us assume that the sampling points are given as $\mathbf{x}_k^q(1\leq k\leq N_q)$, in order to integrate the function $A(\mathbf{x})$ over the whole domain Ω:

$$\int_{\Omega} A(\mathbf{x})\, d\Omega \tag{27}$$

Then, the integrand $A(\mathbf{x})$ can be approximated by diffuse interpolation functions such as the Shepard function or the MLS functions,

$$A(\mathbf{x}) \cong \sum_{k=1}^{N_q} A(\mathbf{x}_k^{(q)})\, \varphi_k(\mathbf{x}) \tag{28}$$

where $\varphi_k(\mathbf{x})$ are the shape functions obtained from non-element interpolation procedure. The whole integration domain Ω can be decomposed into the supports of the shape functions $\varphi_k(\mathbf{x})$,

$$\int_{\Omega} A(\mathbf{x})\, d\Omega \cong \sum_{k=1}^{N_q} A(\mathbf{x}_k^{(q)}) \int_{\Delta_k} \varphi_k(\mathbf{x})\, d\Omega \equiv \sum_{k=1}^{N_q} A(\mathbf{x}_k^{(q)})\, w_k^{(q)} \tag{29}$$

where Δ_k denotes the supports of the shape functions $\varphi_k(\mathbf{x})$ inside Ω. The support Δ_k of each function $\varphi_k(\mathbf{x})$ overlaps with each other and satisfies

$$\Omega = \bigcup_{k=1}^{N_q} \Delta_k \tag{30}$$

If the integration points are chosen, the weight value $w_k^{(q)}$ can be obtained by integrating the corresponding $\varphi_k(\mathbf{x})$. In the present work, the Shepard interpolation is chosen to approximate the integrand $A(\mathbf{x})$, because the Shepard shape function is much easier to evaluate than the MLS shape function. In other words, no matrix factorization $\varphi_k(\mathbf{x})$ is required to evaluate the Shepard shape function, unlike in the case of the MLS shape functions. In this work, the Shepard function is integrated numerically. Although this numerical evaluation of the weight value is an additional computational burden, which is minor compared to the main integration procedure where the MLS function should be evaluated through the complex computational procedure.

It is noted that the same integration points $\mathbf{x}_k^{(q)}$ and the same weight $w_k^{(q)}$ are used for every subdomain $\Omega_s^{(i)}$. By using the proposed integration scheme, the volume integrals in Eq. (25b) are approximated as follows:

$$\begin{aligned}
&\int_{\Omega_s^{(i)}} \nabla\psi_i \cdot \nabla\psi_j d\Omega \cong \sum_{k=1}^{N_q} \nabla\psi_i(\mathbf{x}_k^{(q)}) \cdot \nabla\psi_j(\mathbf{x}_k^{(q)})\, w_k^{(q)} \\
&\int_{\Omega_s^{(i)}} \psi_i f d\Omega \cong \sum_{k=1}^{N_q} \psi_i(\mathbf{x}_k^{(q)})\, f(\mathbf{x}_k^{(q)})\, w_k^{(q)}
\end{aligned} \tag{31}$$

In this work, the trapezoidal rule is used for boundary integration. To properly consider the irregular nodal distribution, we locate the integration points uniformly between the neighboring nodes. Nodes i and j are defined to be neighboring nodes with each other, if the intersection of their supports is a non-empty set. In this work, when we find the neighboring nodes in this work, a fictitious radius of support which is 1.5 times larger than the real radius of support is used to include sufficient information of nodal distribution. The nodal points are also selected as integration points. Fig. 3 shows how to place the integration points when we have an irregular nodal

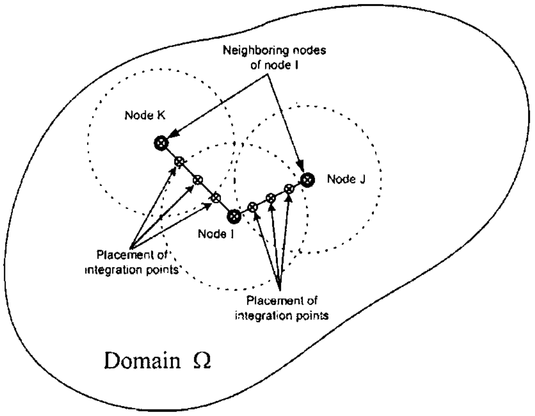

Neighboring nodes of node I
Node K
Node J
Node I
Placement of integration points
Placement of integration points
Domain Ω

Fig. 3 Placement of integration points according to nodal distribution

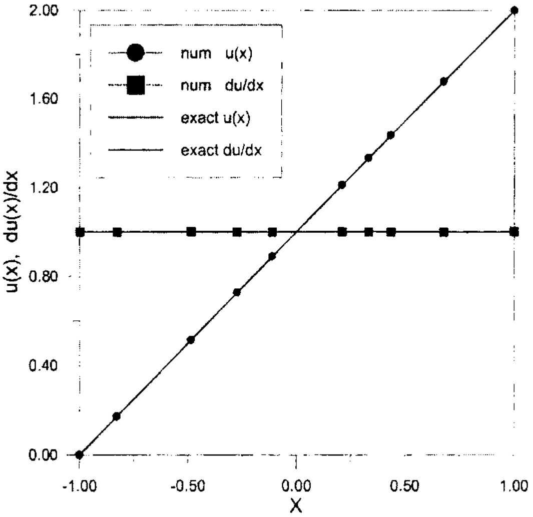


Fig. 4 Numerical solution of a bar under a tip load, by using irregularly distributed nodes

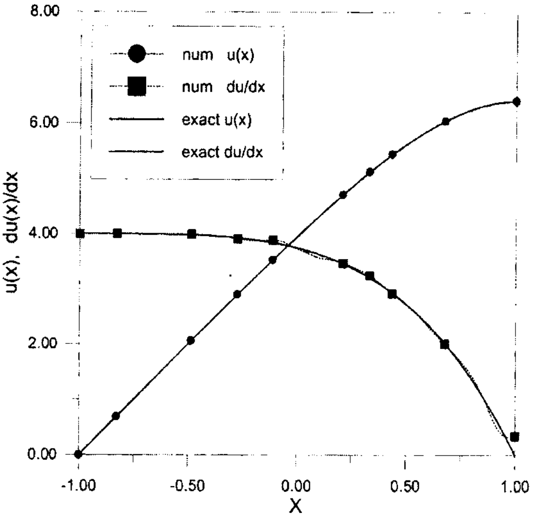


Fig. 5 Numerical solution of a bar under a cubic distributed load, by using irregularly distributed nodes

distribution. Furthermore, if the distance between integration points is smaller than a given tolerance, then the integration points are merged together to avoid excessive large number of integration points. The size of support Δ_k of shape function $\varphi_k(\mathbf{x})$ is selected to be small as possible, while preserving the condition in Eq. (30).

6. Numerical Tests

By using the presently developed numerical integration procedure, with a local symmetric weak form and MLPG, several numerical examples are worked out to investigate the numerical characteristics of the proposed integration method. As well, the results are compared with analytical solutions.

6.1 Analyses of bar problems

A bar under a tip load is analyzed with irregularly distributed nodes, as shown in Fig. 4, and a bar under a cubic distributed load is analyzed with irregularly spaced nodes, as shown in Fig. 5. In the computation, the linear polynomial p-basis and the 4-th order spline-weight function are used. The radius of support of node x_i is chosen 2.4 times as large as the maximum of $|x_i - x_{i-1}|$ and $|x_{i+1} - x_i|$. For the numerical integration, 6 points are placed between neighboring nodes, and the nodal points are included as integration points. To merge the integration points for avoiding a large number of integration points, the tolerance is selected set by $0.1\times$ (length of bar)/(no. of nodes-1)/7. The size of support of shape function $\varphi_k(x)$ for integration point $x_k^{(q)}$ for numerical integration is chosen 0.7 times as large as the maximum of $|x_k^{(q)} - x_{k-1}^{(q)}|$ and $|x_{k+1}^{(q)} - x_k^{(q)}|$. From the result of Fig. 4, it is identified that the constant strain condition

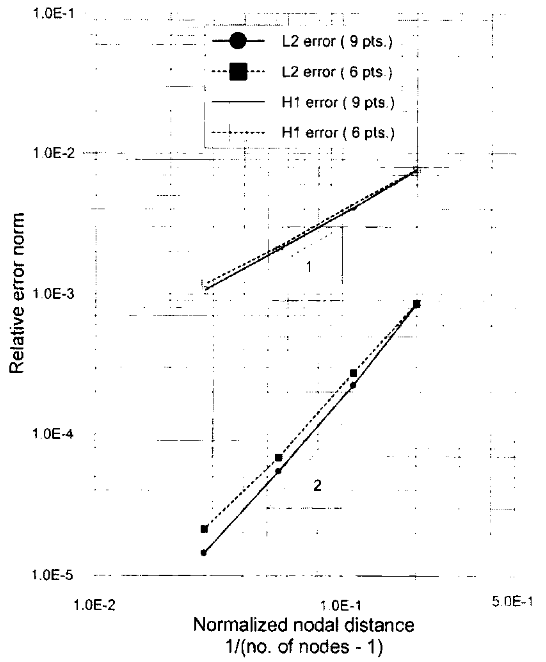


Fig. 6 Convergence results of a bar under a linearly distributed load

is properly preserved by the present algorithm. Fig. 5 shows a little deviation in the derivative at the tip of bar. However, the calculated displacement shows excellent agreement with the exact solution, even though irregularly distributed nodes are used to analyze the problem.

In Fig. 6, convergence results are given, for the problem of a bar under a linearly distributed load. The linear polynomial p-basis, the 4-th order spline-weight function, and the value of 2.4 times nodal distance ($2.4\Delta x$) for radius of support are adopted in numerical simulation. For the numerical integration, 6 points (or 9 points) are placed regularly between neighboring nodes, and the size of support of shape function $\varphi_k(x)$ for integration point $x_k^{(q)}$ is chosen 0.7 times as large as the maximum of $|x_k^{(q)}-x_{k-1}^{(q)}|$ and $|x_{k+1}^{(q)}-x_k^{(q)}|$. In cases of 6 points and 9 points, the tolerances for merging the integration points are selected as 0.1 × (length of bar)/(no. of nodes-1)/7 and 0.1 × (length of bar)/(no. of nodes-1)/10, respectively. To observe the convergence, two relative error norms are defined, respectively

$$\text{Relative } L_2 \text{ error norm}: \frac{\sqrt{\int_\Omega |u_{num}-u_{exact}|^2 dx}}{\sqrt{\int_\Omega |u_{exact}|^2 dx}} \tag{32}$$

Relative H^1 error norm :

$$\frac{\sqrt{\int_\Omega |u_{num}-u_{exact}|^2+|u'_{num}-u'_{exact}|^2 dx}}{\sqrt{\int_\Omega |u_{exact}|^2+|u'_{exact}|^2 dx}} \tag{33}$$

The results show that the convergence rates L_2 and H^1 error norms are approximately 2 and 1, respectively. It is also shown that the convergence rate is improved, in proportional to the number of integration points.

6.2 Poisson's equation in 2D

The convergence test is performed for the Poisson's equation. Regular nodal distributions of 25(5×5), 81(9×9), 289(17×17), and 1089 (33×33) are used to study the convergence of the present algorithm. The source function in the Poisson's equation is given by $f(x, y)=12y^2(x^4-16)+12x^2(y^4-16)$. To measure the convergence rate, we use the relative L_2 and H^1 error norms defined respectively by

$$\text{Relative } L_2 \text{ error norm}: \frac{\sqrt{\int_\Omega |u_{num}-u_{exact}|^2 d\Omega}}{\sqrt{\int_\Omega |u_{exact}|^2 d\Omega}} \tag{34}$$

Relative H^1 error norm :

$$\frac{\sqrt{\int_\Omega |u_{num}-u_{exact}|^2+|\nabla u_{num}-\nabla u_{exact}|^2 d\Omega}}{\sqrt{\int_\Omega |u_{exact}|^2+|\nabla u_{exact}|^2 d\Omega}} \tag{35}$$

The radius of support of nodal trial shape function $\psi_i(\mathbf{x})$ is taken as $2.52\Delta\mathbf{x}$ ($\Delta\mathbf{x}$ denotes nodal distance like in Fig. 8(a)). The linear p-basis, and the 4-th order spline weight function are used. In numerical integration, 3 points (or 5 points) are placed between neighboring nodes, and the nodal points are selected as integration points. To avoid the excessive number of integration points, the integration points are merged together if the distance is smaller than $0.15\Delta\mathbf{x}$. The size of support of shape function $\varphi_k(\mathbf{x})$ for integration point $\mathbf{x}_k^{(q)}$ is chosen as $\Delta\mathbf{x}/3$ (or

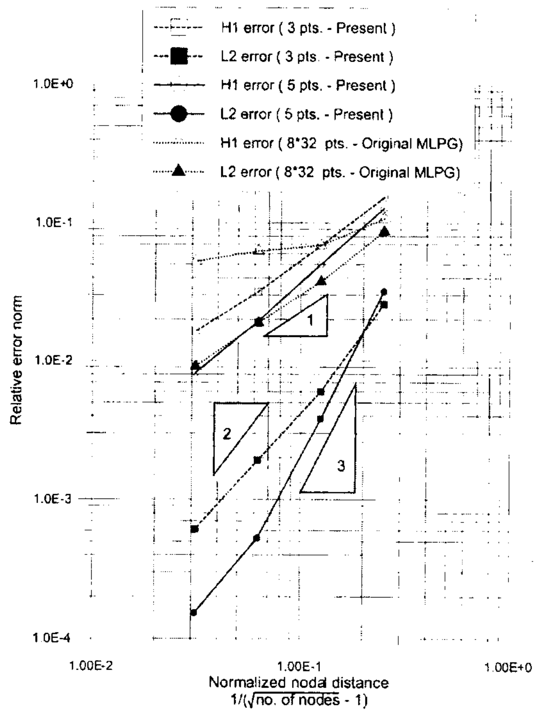
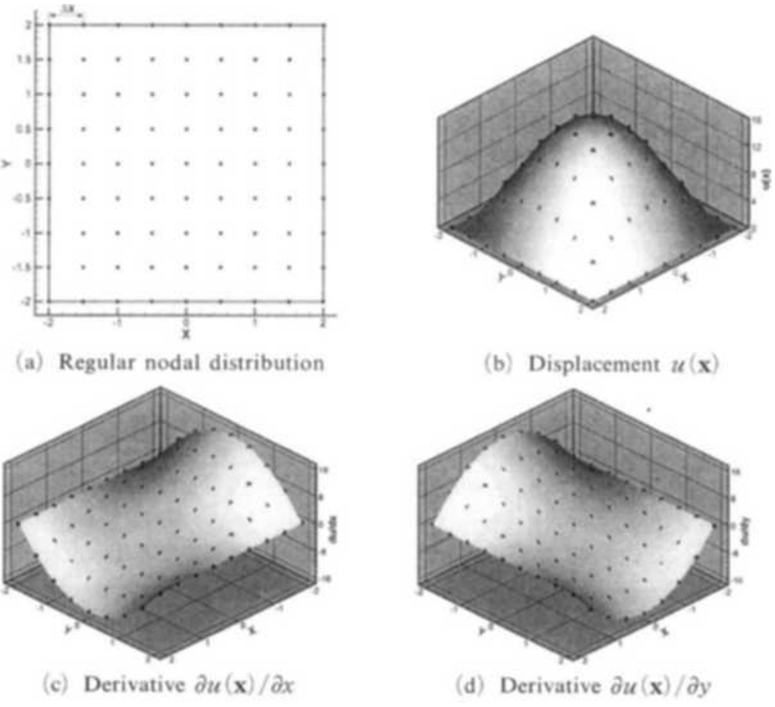


Fig. 7 Convergence rates of the Poisson's equation

$\Delta \mathbf{x}/5$). Like as in Fig. 6, the results shown in Fig. 7 also show that the convergence rate is improved as the number of integration points increases. In the case of 5 integration points between neighboring nodes, the convergence rates of L_2 and H^1 error norms become much higher than 2 and 1, respectively, although only linear p-basis is adopted in the simulation. And the convergence rates of the proposed scheme are compared with those of the original MLPG 8×32 integration scheme (8 points Gaussian quadrature in the radial direction and 32 points trapezoidal integration in the circumferencial direction). In case of the present 5 points integration, the total number of integration points is around 25 times of the number of nodes, and in case of the original integration, the number of integraion points in each local sub-domain is $256(8 \times 32)$ points. Even though the number of integration points per node in the present method is less than that in the original MLPG method, one can observe that the proposed integration scheme gives improved and



(a) Regular nodal distribution

(b) Displacement $u(\mathbf{x})$

(c) Derivative $\partial u(\mathbf{x})/\partial x$

(d) Derivative $\partial u(\mathbf{x})/\partial y$

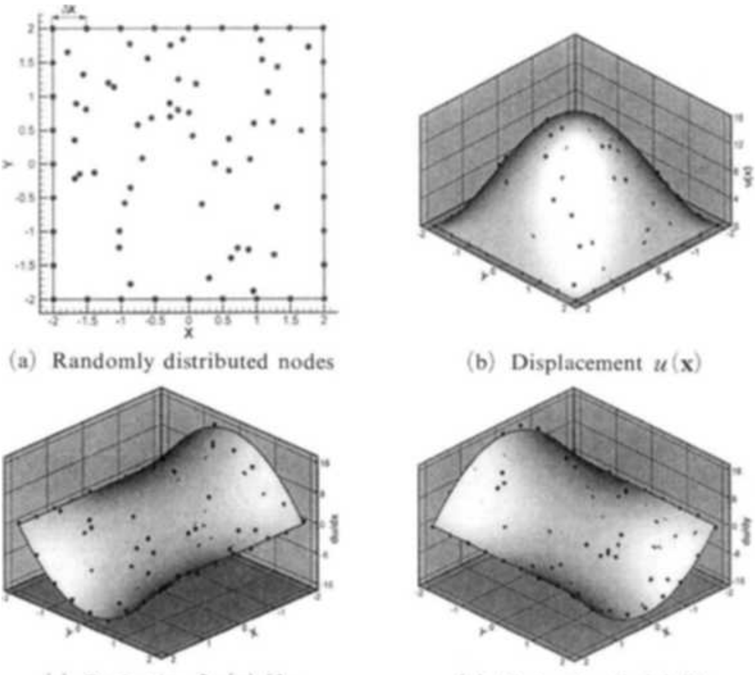
Fig. 8 (a) Regular nodal distribution (L_2 relative error norm : 0.15%) (b) Displacement $u(\mathbf{x})$ obtained by regularly distributed nodes (c) Derivative $\partial u(\mathbf{x})/\partial x$ obtained by regularly distributed nodes (d) Derivative $\partial u(\mathbf{x})/\partial y$ obtained by regularly distributed nodes robust convergence rates compared to the original integration scheme.

To observe the difference in the solutions between regularly distributed nodes and randomly distributed nodes, the Poisson's equation with the source function of $f(x, y) = (2x^2 + 2y^2 - 16)$ is analyzed. In the simulations of Fig. 8 and Fig. 9, the linear p-basis, 81 nodes, the 4-th order spline weight function, and $2.52\Delta\mathbf{x}$ of radius of support of nodal trial shape function $\psi_i(\mathbf{x})$ are adopted. The nodal distance $\Delta\mathbf{x}$ in each case is shown in Fig. 8(a) and Fig. 9(a), resepectively. For the integration, 5 points arc placed between neighboring nodes. The tolerance is taken as $0.15\ \Delta\mathbf{x}$ to merge the integration points, and the nodal points are also included as integration points. The size of support of shape function $\varphi_k(\mathbf{x})$ for integration point $\mathbf{x}_k^{(q)}$ is chosen as $\Delta\mathbf{x}/5$. The exact solution is presented as surface, and the numerical solution is denoted by points. The displacements from both cases show good agreements with the exact solution. The relative L_2 error norms of regular and irregualr cases are 0.15%, and 1.26%, respectively. However, it is observed that the derivatives near the boundary in the irregular case show large error compared to the regular case.

A problem with a local high gradient is simulated with randomly located nodes. The source function is given by

$$f(x, y) = e^{(4-x^2)(4-y^2)-16}\left(4x^2(4-y^2)^2 - 2(4-y)^2 + 4y^2(4-x^2)^2 - 2(4-x^2)\right) \quad (36)$$

In the region of high gradient, more nodes are randomly located compared to other region. The randomly located nodes are presented in Fig. 10. In the simulation of Fig. 11, linear p-basis, 160 nodes, and the 4-th order spline weight function are utilized. The radius of support of the nodal trial shape function $\psi_i(\mathbf{x})$ for each node is adopted as $2.52\Delta\mathbf{x}$ (The distance of $\Delta\mathbf{x}$ is shown in Fig. 10). For the integration, 5 points are placed between neighboring nodes. As mentioned before, if the integration points selected are too close to each other, they are merged together. The tolerance for merging the integration points is



(a) Randomly distributed nodes

(b) Displacement $u(\mathbf{x})$

(c) Derivative $\partial u(\mathbf{x})/\partial x$

(d) Derivative $\partial u(\mathbf{x})/\partial y$

Fig. 9 (a) Randomly distributed nodes (L_2 relative error norm : 1.26%) (b) Displacement $u(\mathbf{x})$ obtained by regularly distributed nodes (c) Derivative $\partial u(\mathbf{x})/\partial x$ obtained by regularly distributed nodes (d) Derivative $\partial u(\mathbf{x})/\partial y$ obtained by regularly distributed nodes adopted as 0.15$\Delta\mathbf{x}$, and the nodal points are also included in integration points. The radius of support of shape function $\varphi_k(\mathbf{x})$ for integration point $\mathbf{x}_k^{(q)}$ is chosen as $\Delta\mathbf{x}/5$. In Fig. 11, the numerical results are compared with the analytical solution. The exact solution is presented as surface, and the numerical solution is denoted by points. Even thouth there is some error in the vicinity of the peak point, we can identify that the present algorithm gives an acceptable solution when we deal with randomly located nodes. In the simulation of Fig. 12, the radius of support of nodal trial shape function $\psi_i(\mathbf{x})$ is decreased to 2.0$\Delta\mathbf{x}$, while preserving the other parameters adopted in the simulation of Fig. 11. The result presented in Fig. 12 shows that the numerical results is greatly improved in the vicinity of the peak point, as the support size of nodal trial function is decreased. From the comparison between the solutions of Fig. 11 and Fig. 12, it can be noticed that a smaller support size of nodal trial shape function gives an improved numcrical solution in the region where the solution is changed suddenly. To assess the accuracy of the proposed scheme, the number of integration points required in the original MLPG integration to achieve the same degree of accuracy for the example of randomly distributed nodal points in Fig. 12 is presented in Fig. 13(a) and Fig. 13(b). In the present case, the total number of integration points is around 10 times the number of nodes, and in the original MLPG integration

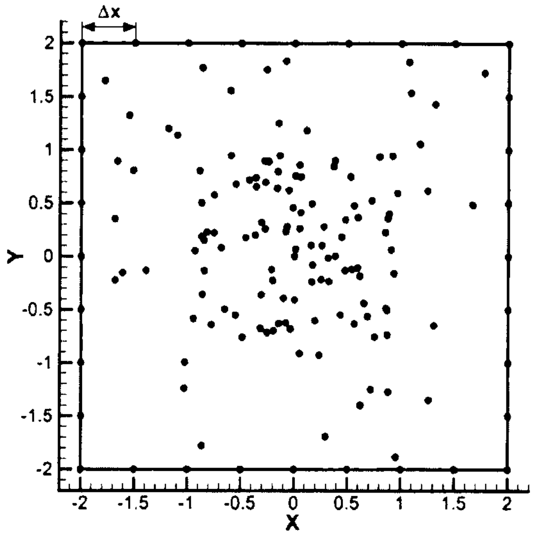


Fig. 10 Randomly distributed nodes

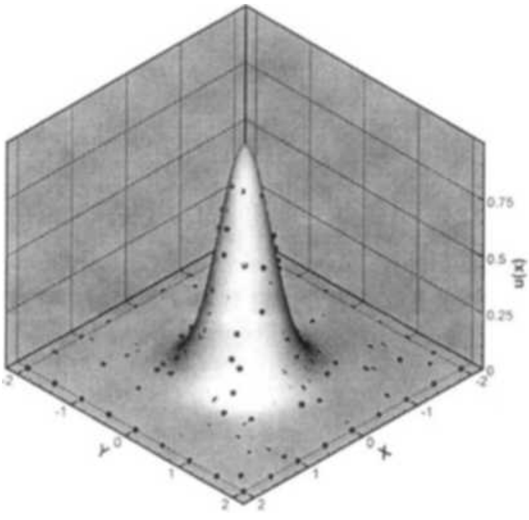


Fig. 11 Comparison between the exact solution and the numerical solution obtained by using 2.52$\Delta\mathbf{x}$ of radius of support of nodal trial shape function (L_2 relative error norm : 5.91 %, Relative error norm at the center : 8.94%)

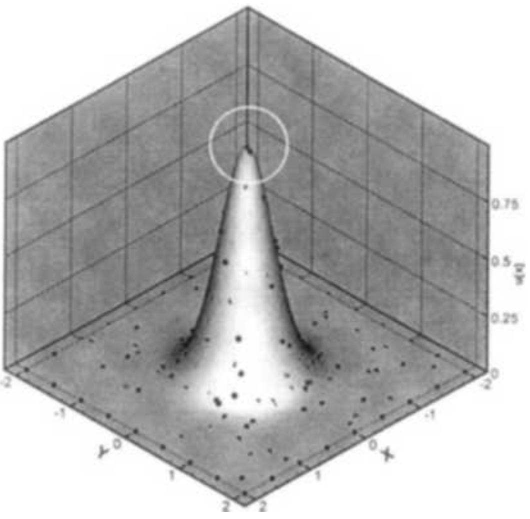


Fig. 12 Comparison between the exact solution and the numerical solution obtained by using 2.0$\Delta\mathbf{x}$ of radius of support of nodal trial shape function (L_2 relative error norm : 4.46 %, Relative error norm at the center : 1.32%)

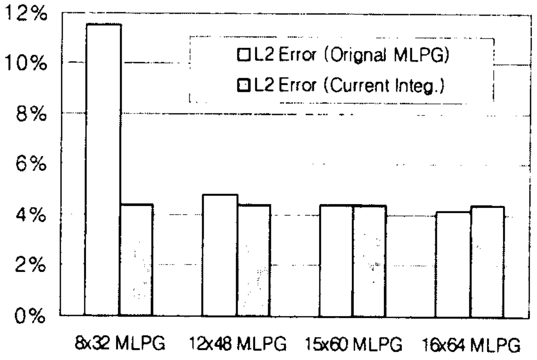


Fig. 13(a) The number of integration points required in the original MLPG integration to achieve the same degree of L_2 error norm accuracy for the example of randomly distributed nodal points

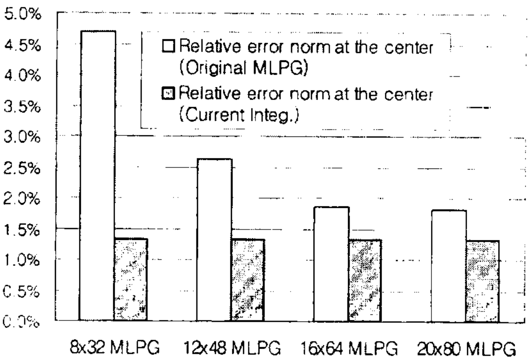


Fig. 13(b) The number of integration points required in the original MLPG integration to achieve the same degree of relative error norm accuracy at the center (peak point) for the example of randomly distributed nodal points

more than 900(15×60) integration points are required in each local sub-domain to attain the comparable L_2 error norm accuracy 4.46%. In case of relative error norm $|u_{num}-u_{ext}|/|u_{ext}|$ at the center (peak point), comparable accuracy 1.32% is not obtained even by 1600(20×80) original integration as shown in Fig. 13(b). The comparison shows that the proposed method is more flexible in handling the randomly distributed nodal points.

Concluding Remarks

In MLPG (Meshless Local Petrov-Galerkin) method (Atluri and Zhu, 1998a, b), the supports of nodal test functions are taken as integration domains which overlap with each other, and regularly distributed points in the sub-domains are used for numerical integration of the local symmetric weak form. This integration procedure has many advantages, such as truly meshless character (Atluri and Zhu, 1998), compared to other integration schemes.

In the present work, an adaptive mesh-independent numerical integration algorithm is proposed for the MLPG method, to increase the flexibility in dealing with irregular nodal distribution. To properly consider the irregular nodal distribution, the integration points are located between the neighboring nodes, and the nodal points are also selected as integration points. If the distance between integration points is smaller than a given tolerance, then the integration points are merged together to avoid excessive large number of integration points. Furthermore, to preserve the non-element integration nature, we adopted the Shepard function to approximate the integrand in the local symmetric weak form by the values of integrand at the integration points. This procedure makes it possible to integrate the local symmetric weak form without any non-overlapping meshes for arbitrarily given integration points.

By using the proposed scheme, convergence tests are performed, and several numerical examples are worked out. Through the numerical results, it is convinced that the present algorithm gives reasonable solution when we deal with randomly distributed nodes as well as regularly distributed nodes. However, the proposed algorithm to place the integration points is only one of various possibilities, and may not be the best one. Therefore, further research efforts are required to find the optimal integration points from which more robust and accurate solution can be obtained with a minimum computational cost.

Acknowledgment

This work was supported by grant No. R05-2002-000-00717-0 from the Basic Research Program of the Korea Science & Engineering Foundation. Authors would like to acknowledge the financial support from the Korea Science & Engineering Foundation.

References

Atluri, S. N., Cho, J. Y. and Kim, H. G., 1999, "Analysis of Thin Beams, Using the Meshless Local Petrov-Galerkin Method, with Generalized Moving Least Squares Interpolations," *Comp. Mech.*, Vol. 24, pp. 334~347.

Atluri, S. N., Kim, H. G. and Cho, J. Y., 1999, "A Critical Assessment of the Truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods," *Comp. Mech.*, Vol. 24, pp. 348~372.

Atluri, S. N. and Zhu, T., 1998a, "A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics," *Comp. Mech.*, Vol. 22, pp. 117~127.

Atluri, S. N. and Zhu, T., 1998b, "A new meshless local Petrov-Galerkin (MLPG) Approach to Nonlinear Problems in Computer Modeling and Simulation," *Comput. Modeling Simul. Eng.*, Vol. 3, pp. 187~196.

Babuška, I. and Melenk, J., 1997, "The Partition of Unity Method," *Int. J. Num. Meth. Eng.*, Vol. 40, pp. 727~758.

Belytschko, T., Lu, Y. Y. and Gu, L., 1994, "Element-free Galerkin Methods," *Int. J. Num. Meth. Eng.*, Vol. 37, pp. 229~256.

De, S. and Bathe, K. J., 2001, "Towards an Efficient Meshless Computational Technique : the Method of Finite Spheres," *Eng. Comp.*, Vol. 18, pp. 170~192.

Duarte, C. A. and Oden, J. T., 1996, "An h-p Adaptive Method Using Clouds," *Comp. Meth. Appl. Mech. Eng.*, Vol. 139, pp. 237~262.

Lancaster, P. and Salkauskas, K., 1981, "Surfaces Generated by Moving Least Squares Methods," *Math. Comp.*, Vol. 37, pp. 141~158.

Liu, W. K., Jun, S. and Zhang, Y., 1995, "Reproducing Kernel Particle Methods," *Int. J. Num. Meth. Fluids*, Vol. 20, pp. 1081~1106.

Liu, W. K., Chen, Y., Chang, C. T. and Belytschko, T., 1996, "Advances in Multiple Scale Kernel Particle Methods," *Comp. Mech.*, Vol. 18, pp. 73~111.

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. and Villon, P., 1992, "Generalizing the Finite Element Method : Diffuse Approximation and Diffuse Elements," *Comp. Mech.*, Vol. 10, pp. 307~318.

Organ, D., Fleming, M., Terry, T. and Belytschko, T., 1996, "Continuous Meshless Approximations for Nonconvex Bodies by Diffraction and Transparency," *Comp. Mech.*, Vol. 18, pp. 225~235.

Oñate, E., Idelsohn, S., Zienkiewicz, O. C. and Taylor, R. L., 1996, "A Finite Point Method in Computational Mechanics. Applications to Convective Transport and Fluid Flow," *Int. J. Num. Meth. Eng.*, Vol. 39, pp. 3839~3866.

Shepard, D., 1968, "A Two-Dimensional Function for Irregularly Spaced Data," *Proceeding of ACM National Conference.*, pp. 517~524.

Zhu, T. and Atluri, S. N., 1998, "A Modified Collocation Method and a Penalty Formulation for Enforcing the Essential Boundary Conditions in the Element Free Galerkin Method," *Comp. Mech.*, Vol. 21, pp. 211~222.

Zhu, T., Zhang, J. D. and Atluri, S. N., 1998a, "A Local Boundary Integral Equation (LBIE) Method in Computational Mechanics, and a Meshless Discretization Approach," *Comp. Mech.*, Vol. 21, pp. 223~235.

Zhu, T., Zhang, J. D. and Atluri, S. N., 1998b, "A Meshless Local Boundary Integral Equation (LBIE) Method for Solving Nonlinear Problems," *Comp. Mech.*, Vol. 22, pp. 174~186.