 CMES, vol.3, no.1, pp.11-51, 2002

The Meshless Local Petrov-Galerkin (MLPG) Method: A Simple & Less-costly Alternative to the Finite Element and Boundary Element Methods

Satya N. Atluri[1] **& Shengping Shen**[1]

Abstract: A comparison study of the efficiency and accuracy of a variety of meshless trial and test functions is presented in this paper, based on the general concept of the meshless local Petrov-Galerkin (MLPG) method. 5 types of trial functions, and 6 types of test functions are explored. Different test functions result in different MLPG methods, and six such MLPG methods are presented in this paper. In all these six MLPG methods, absolutely no meshes are needed either for the interpolation of the trial and test functions, or for the integration of the weak-form; while other meshless methods require background cells. Because complicated shape functions for the trial function are inevitable at the present stage, in order to develop a fast and robust meshless method, we explore ways to avoid the use of a domain integral in the weak-form, by choosing an appropriate test function. The MLPG5 method (wherein the local, nodal-based test function, over a local sub-domain Ω_s (or Ω_{te}) centered at a node, is the Heaviside step function) avoids the need for both a domain integral in the attendant symmetric weak-form as well as a singular integral. Convergence studies in the numerical examples show that all of the MLPG methods possess excellent rates of convergence, for both the unknown variables and their derivatives. An analysis of computational costs shows that the MLPG5 method is less expensive, both in computational costs as well as definitely in human-labor costs, than the FEM, or BEM. Thus, due to its speed, accuracy and robustness, the MLPG5 method may be expected to replace the FEM, in the near future.

keyword: MLPG, test function, trial function, MLS, RBF

[1] Center for Aerospace Research & Education
7704 Boelter Hall
University of California, Los Angeles
Los Angeles, CA 90095-1600, USA

1 Introduction

Meshless methods, as alternative numerical approaches to eliminate the well-known drawbacks in the finite element and boundary element methods, have attracted much attention in the past decade, due to their flexibility, and due to their potential in negating the need for the human-labor intensive process of constructing geometric meshes in a domain. Such meshless methods are especially useful in those problems with discontinuities or moving boundaries. The main objective of the meshless methods is to get rid of, or at least alleviate the difficulty of, meshing and remeshing the entire structure; by only adding or deleting nodes in the entire structure, instead. Meshless methods may also alleviate some other problems associated with the finite element method, such as locking, element distortion, and others.

The initial idea of meshless methods dates back to the smooth particle hydrodynamics (SPH) method for modeling astrophysical phenomena (Gingold and Monaghan, 1977). The research into meshless methods has become very active, only after the publication of the Diffuse Element Method by Nayroles, Touzot & Villon (1992). Several so-called meshless methods [Element Free Galerkin (EFG)] by Belytschko, Lu & Gu (1994); Reproducing Kernel Particle Method (RKPM) by Liu, Chen, Uras & Chang (1996); the Partition of Unity Finite Element Method (PUFEM) by Babuska and Melenk (1997); hp-cloud method by Duarte and Oden 1996; Natural Element Method (NEM) by Sukumar, Moran, & Belytschko (1998); Meshless Galerkin methods using Radial Basis Functions (RBF) by Wendland (1999); have also been reported in literature since then. The major differences in these meshless methods, all of which may be classified as Galerkin methods, come only from the techniques used for interpolating the trial function. Even though no mesh is required in these methods for the interpolation of the trial and test functions for the solution variables, the use of shadow elements is inevitable in these meth-

ods, for the integration of the symmetric weak-form, or of the 'energy'. Therefore, these methods are not truly meshless.

Recently, two truly meshless methods, the meshless local boundary integral equation (LBIE) method, and the meshless local Petrov-Galerkin (MLPG) method, have been developed in Zhu, Zhang & Atluri (1998a, b), Atluri & Zhu (1998a, b) and Atluri, Kim & Cho (1999), for solving linear and non-linear boundary problems. Both these methods are truly meshless, as no finite element/or boundary element meshes are required in these two approaches, either for purposes of interpolation of the trial and test functions for the solution variables, or for the purpose of integration of the weak-form (either "symmetric" or "unsymmetric"). All pertinent integrals can be easily evaluated over over-lapping, regularly shaped, domains (in general, spheres in three-dimensional problems) and their boundaries. In fact, The LBIE approach can be treated simply as a special case of the MLPG approach (Atluri, Kim & Cho, 1999). Remarkable successes of the MLPG method have been reported in solving the convection-diffusion problems [Lin & Atluri (2000)]; fracture mechanics problems [Kim & Atluri (2000), Ching & Batra (2001)]; Navier-Stokes flows [Lin & Atluri (2001)]; and plate bending problems [Gu & Liu (2001), and Long & Atluri (2002)].

In summary, the MLPG is a truly meshless method, which involves not only a meshless interpolation for the trial functions (such as MLS, PU, Shepard function or RBF), but also a meshless integration of the weak-form (i.e. all integrations are always performed over overlapping and regularly shaped sub-domains such as spheres, parallelopipeds, and ellipsoids in 3-D). In the conventional Galerkin method, the trial and test functions are chosen from the same function-space. In MLPG, the nodal trial and test functions can be different: the nodal trial function may correspond to any one of MLS, PU, Shepard function, or RBF types of interpolations; and the test function may be totally different, and may correspond to any one of MLS, PU, Shepard function, RBF, a Heaviside step function, a Dirac delta function, the Gaussian weight function of MLS, a special form of the fundamental solution to the differential equation, or any other convenient function, in the support domain, Ω_{te}, of the test function. Furthermore, the physical sizes of the supports (Ω_{tr} and Ω_{te}, respectively) as well as shapes of the nodal trial and test functions may be different. These features make the MLPG method very flexible. The MLPG method, based on a local formulation, can include all the other meshless methods based on global formulation, as special cases.

Effective implementations of meshless methods are the key to their success. Meshless methods still require a considerable improvement, before they are equal to or excel the convenience of the finite element method in computational science and engineering. The great challenges appear to lie in improving the speed and robustness of the meshless methods, to match the corresponding levels of the low-order finite elements. Numerical integration plays an important role in the convergence of numerical solutions of meshless methods. Unfortunately, the nodal shape functions from meshless interpolations, such as MLS, are highly complex in nature; and this makes an accurate numerical integration of the weak form highly difficult, especially in the conventional Galerkin type methods. Due to the fact that the computational cost is too high, and large numbers of Gaussian points are needed to obtain a convergent result in the Galerkin equivalent (Atluri, Kim & Cho (1999)) of the MLPG, we henceforth discard the notion of using Galerkin type approximations (i.e. trial and test functions belonging to the same function spaces) in the MLPG method. Also, although only boundary integrals appear in the LBIE method, these boundary integrals may contain singularities, to which special attention should be paid [Sladek, Sladek & Atluri (2000)]. In order to take a full advantage of the concept of the MLPG method, we try to

1. Use mainly the Petrov-Galerkin type approximations, i.e. the nodal trial and test functions are decidedly different.

2. Simplify the integrand in the weak-form over the local, nodal-based test function domain;

3. Avoid the domain integral in the weak-form, if possible, over the local, nodal-based test-function domain.

The second idea implies that we should choose a suitable interpolation that makes the nodal (trial function) shape function simple. The third idea means that we should choose a suitable local nodal-based test function over a regular-shaped local domain [centered at each node, such as a sphere, parallelopiped, and ellipsoid in 3-D], in or-

der to make the domain integral in the weak-form disappear. In this paper, we will study the efficiency and accuracy of a variety of meshless nodal trial and test functions, using the three main ideas mentioned above. MLS (or, entirely equivalently, RKPM), PU, Shepard function, and RBF interpolations are selected to be the trial functions in this paper. Unfortunately, the nodal shape functions from all these interpolations are highly complex (although the one from Shepard function is relatively simpler), which means that the integrand in the weak-form, for Galerkin-type method, including the Galerkin equivalent of the MLPG, cannot be simplified at the present stage. Six different nodal-based local test functions are also selected, leading to six different MLPG methods. Based on the MLPG concept, we label these variants of the MLPG method as MLPG1, MLPG2, MLPG3, MLPG4, MLPG5, and MLPG6, respectively. Among them, there are three methods that avoid the domain integral in the weak-form, over the nodal test-function domain Ω_s: MLPG2 (wherein the local, nodal-based test function, over a local sub-domain Ω_s centered at a node, is the Dirac's Delta function); MLPG4 (wherein the local, nodal-based test function, over a local sub-domain Ω_s centered at a node, is the modified fundamental solution to the differential equation); and MLPG5 (wherein the local, nodal-based test function, over a local sub-domain Ω_s centered at a node, is the Heaviside step function). MLPG4 (which is synonymous with the LBIE) involves singular integrals; while the collocation method, (i.e. MLPG2), is notorious for the sensitivity of the solution to the choice of proper collocation points. However, MLPG5 does not involve either a domain, or a singular integral, to generate the stiffness matrix; it only involves the regular boundary integral. Thus, it is a highly promising MLPG method. The numerical examples in Section 7 also illustrate that the MLPG5 method is fast, accurate and robust. We believe that the MLPG (especially the MLPG5) method holds a great promise to replace the finite element method, as a method of choice, someday in the not-too distant future.

The paper is organized as follows. In Section 2, we introduce the 3 global weak forms, which are the basis of the Finite Element Method (FEM), an Element Free Galerkin method (EFG), the global collocation method, and the global boundary integral equation (or boundary element) method. In Section 3 are derived three corresponding *local weak forms*, which form the basis of the meshless local Petrov-Galerkin (MLPG) method. Section 4 discusses *five* meshless interpolations for trial functions: 1. moving least square (MLS); 2. Shepard function; 3. the partition of unity method (PU); 4. reproducing kernel particle methods (RKPM); and 5. radial basis functions (RBF). In Section 5, *six different* nodal-based local test functions are selected; and thus *six different* MLPG methods are developed. Based on the MLPG concept, these variants of the MLPG method are labeled as MLPG1, MLPG2, MLPG3, MLPG4, MLPG5, and MLPG6, respectively. Section 6 deals with the implementation of the essential boundary conditions in these six MLPG methods. In Section 7, some numerical results are presented, to compare the efficiency and accuracy of the present MLPG methods. Section 8 qualitatively estimates the computational costs in the present MLPG methods, and compares these costs with that of the classical FEM. The paper concludes in Section 9.

2 The Global Weak Forms

Consider the linear Poisson's equation (in a global domain Ω_s bounded by Γ), which can be written as

$$\nabla^2 u(\mathbf{x}) = p(\mathbf{x}) \quad \mathbf{x} \in \Omega \tag{1}$$

where p is a given source function, and the domain Ω is enclosed by $\Gamma = \Gamma_u \cup \Gamma_q$, with boundary conditions

$$u = \overline{u} \quad \text{on} \quad \Gamma_u, \tag{2a}$$

$$\frac{\partial u}{\partial n} \equiv q = \overline{q} \quad \text{on} \quad \Gamma_q \tag{2b}$$

where $\overline{u}$ and $\overline{q}$ are the prescribed potential and normal flux, respectively, on the boundary Γ_u, and on the boundary Γ_q; and n is the outward normal direction to the boundary Γ. We will consider *three weak formulations* of the differential Eq.(1).

In the Galerkin finite element, and in element free Galerkin [Belytschko, Lu, & Gu, 1994] methods, which are based on the global Galerkin formulation, one uses the global weak form over the entire domain Ω, to solve the problem numerically. For a general discussion of the alternate "symmetric" and "unsymmetric" weak forms as the basis of a variety of numerical methods, such as the collocation, sub-domain, least squares, Finite Volume, Finite Element & Boundary Element Methods, see Atluri (2002).

A global unsymmetric weak formulation (GUSWF1) of the problem may be written as

$$\int_{\Omega} (\nabla^2 u - p) v d\Omega = 0 \tag{3}$$

where u is the trial function, and v is the test function. This GUSWF requires that u be at least C^1 continuous, while v may be discontinuous, and hence is the label, "unsymmetric weak formulation", in as much as the continuity requirements on u and v are not the same. In this case, the collocation approach or other approaches, such as the penalty approach, are used to impose both the essential as well as natural boundary conditions.

Using $(\nabla^2 u)v = u_{,ii}v = (u_{,i}v)_{,i} - u_{,i}v_{,i}$, and the divergence theorem, we obtain the following global symmetric weak formulation (GSWF),

$$\int_{\Gamma} u_{,i} n_i v d\Gamma - \int_{\Omega} (u_{,i}v_{,i} + pv) d\Omega - \alpha \int_{\Gamma_u} (u - \overline{u})\, v d\Gamma = 0 \tag{4}$$

where α is a penalty parameter, which is used to impose the essential boundary conditions. The GSWF requires that both u and v be C^0 continuous, and hence the label "symmetric weak form". Imposing the natural boundary condition (2b) and noticing that $u_{,i}n_i = \partial u / \partial n \equiv q$ in Eq. (4), we obtain,

$$\int_{\Gamma_u} qv d\Gamma + \int_{\Gamma_q} \overline{q} v d\Gamma - \int_{\Omega} (u_{,i}v_{,i} + pv) d\Omega - \alpha \int_{\Gamma_u} (u - \overline{u})\, v d\Gamma = 0 \tag{5}$$

In this case, the collocation approach, or other approaches such as the penalty approach, is used to impose only the essential boundary condition a priori. The Galerkin Finite Element Method (GFEM) is based on the Global Symmetric Weak Form (GSWF), Eq. (5).

Using $(\nabla^2 u)v = u_{,ii}v = (u_{,i}v)_{,i} - (uv_{,i})_{,i} + uv_{,ii}$, and the divergence theorem twice, yields another "global unsymmetric weak formulation" (GUSWF2),

$$\int_{\Gamma} u_{,i} n_i v d\Gamma - \int_{\Gamma} u v_{,i} n_i d\Gamma + \int_{\Omega} u \nabla^2 v d\Omega - \int_{\Omega} pv d\Omega = 0 \tag{6}$$

This GUSWF requires that v be at least C^1 continuous, while u may be discontinuous. Imposing the natural boundary condition (2b) and essential boundary condition (2) in Eq. (6), we obtain,

$$\int_{\Gamma_q} \overline{q} v d\Gamma + \int_{\Gamma_u} qv d\Gamma - \int_{\Gamma_q} u v_{,i} n_i d\Gamma - \int_{\Gamma_u} \overline{u} v_{,i} n_i d\Gamma + \int_{\Omega} u \nabla^2 v d\Omega - \int_{\Omega} pv d\Omega = 0 \tag{7}$$

The major dilemma in meshless methods, based on the global weak forms revolves around how to evaluate the integrals in the (3-7). In many methods, such as the EFG method, the background cells are used for integration. As a matter of fact, the background cell integration does not lead to a truly meshless method.

3 MLPG: the Local Weak Forms

Many of the so-called meshless methods, such as the EFG (element free Galerkin) method, are based on the global weak form over the entire domain Ω. In the MLPG, however, a local weak form over a local sub-domain Ω_s, which is located entirely inside the global domain Ω is used. This is the most distinguishing feature of the MLPG. It is noted that the local sub-domain Ω_s can be of an arbitrary shape. Even though a particular approximation of the local weak form will give the same resulting discretized equations as from the Galerkin approximation of the global weak form, the local weak form will provide a clear concept for a local meshless integration of the weak-form, which does not need any background integration cells over the entire domain. Also, it will lead to a natural way to construct the global stiffness matrix: not through the integration over a contiguous mesh, and by assembly of the stiffness matrices of the elements in the mesh, but through the integration over local sub-domains. These local sub-domains do not form a contiguous mesh globally; but these disjointed local sub-domains may overlap each other. In contrast to the conventional Galerkin finite element formulations, which are based on the global weak form, the MLPG method (Atluri and Zhu, 1998) stems from a weak form over a sub-domain Ω_s inside the global domain Ω as shown in Fig. 1, *where the domain of support of the test function Ω_{te}^I is synonymous with the sub-domain Ω_s.* A local unsymmetric weak formulation (LUSWF1) of

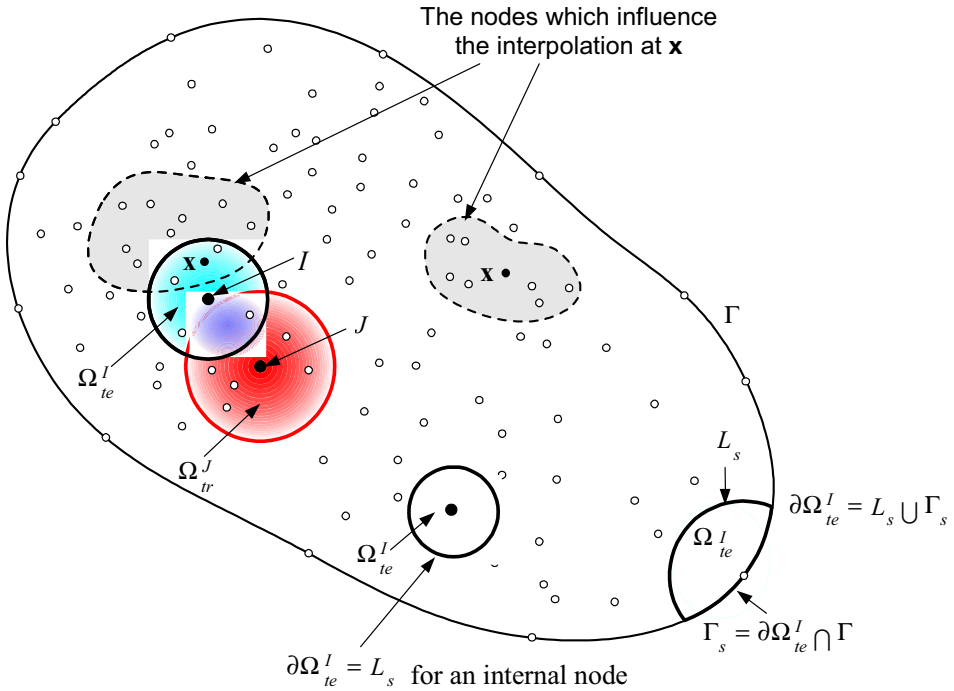


Figure 1 : Schematics of the MLPG method

the problem (1), corresponding to the GUSWF1, may be written as

$$\int_{\Omega_s} (\nabla^2 u - p) v d\Omega = 0 \tag{8}$$

The local symmetric weak formulation (LSWF), corresponding to GSWF, can be written as

$$\int_{L_s} qvd\Gamma + \int_{\Gamma_{su}} qvd\Gamma + \int_{\Gamma_{sq}} \overline{q}vd\Gamma - \int_{\Omega_s} (u_{,i}v_{,i} + pv) d\Omega$$
$$- \alpha \int_{\Gamma_u} (u - \overline{u}) v d\Gamma = 0 \tag{9a}$$

in which, Γ_{sq} is a part of $\partial\Omega_s$, over which the natural boundary condition is specified. In general, $\partial\Omega_s = \Gamma_s \cup L_s$, with Γ_s being a part of the local boundary located on the global boundary, and let L_s being the other part of the local boundary over which no boundary condition is specified, i.e., $\Gamma_s = \partial\Omega_s \cap \Gamma$, (see Fig. 1). For a subdomain located entirely within the global domain, there is no intersection between $\partial\Omega_s$ and Γ, $L_s = \partial\Omega_s$ and the integrals over Γ_{sq} and Γ_{su} vanish.

It is noted that in a 3-D problem, Ω_s can be: (i) a sphere; (ii) an ellipsoid; (iii) a cube, or any other simple shape. Correspondingly, in a 2-D problem, Ω_s can be: (i) a circle; (ii) an ellipse; (iii) a rectangle, or any other convenient regular or irregular shape. In order to simplify the above equation, we can select a test function v such that it vanishes over L_s, which is, usually, a circle (for an internal node) in a 2D problem, or the circular arc (for a node on the global boundary Γ. Therefore the first integral in Eq. (9a) vanishes. Using this test function and rearranging Eq. (9a), we obtain the following local weak form (LSWF),

$$\int_{\Omega_s} u_{,i}v_{,i} d\Omega - \int_{\Gamma_{su}} qvd\Gamma + \alpha \int_{\Gamma_u} uvd\Gamma =$$
$$\int_{\Gamma_{sq}} \overline{q}vd\Gamma - \int_{\Omega_s} pvd\Omega + \alpha \int_{\Gamma_u} \overline{u}vd\Gamma \tag{9b}$$

Another local unsymmetric weak formulation (LUSWF2) of the problem (1) corresponding to GUSWF2 can be written as

$$\int_{L_s} qvd\Gamma - \int_{L_s} uv_{,i}n_i d\Gamma + \int_{\Gamma_{sq}} \overline{q}vd\Gamma + \int_{\Gamma_{su}} qvd\Gamma - \int_{\Gamma_{sq}} uv_{,i}n_i d\Gamma - \int_{\Gamma_{su}} \overline{u}v_{,i}n_i d\Gamma + \int_{\Omega_s} u\nabla^2 vd\Omega - \int_{\Omega_s} pvd\Omega = 0 \quad (10a)$$

Also, by selecting a test function v such that it vanishes over L_s, we obtain the following local weak form

$$\int_{\Gamma_{su}} qvd\Gamma - \int_{L_s} uv_{,i}n_i d\Gamma - \int_{\Gamma_{sq}} uv_{,i}n_i d\Gamma + \int_{\Omega_s} u\nabla^2 vd\Omega = \int_{\Omega_s} pvd\Omega - \int_{\Gamma_{sq}} \overline{q}vd\Gamma + \int_{\Gamma_{su}} \overline{u}v_{,i}n_i d\Gamma \quad (10b)$$

It should be noted that these local weak forms (8)-(10) hold, irrespective of the size and shape of $\partial\Omega_s$. Therefore, we can choose a simple regular shape for Ω_s and thus for $\partial\Omega_s$. The most regular shape should be an n-dimensional sphere for a boundary value problem defined on an n-dimensional space. Thus, an n-dimensional sphere (or a part of an n-dimensional sphere for a boundary node), is chosen in our development (see Fig. 1).

With the local weak form for any point $\mathbf{x}$, the problem (1) becomes one as if we are dealing with a localized boundary value problem over an n-dimensional sphere Ω_s. The radius of the sphere will thus affect the solution. The equilibrium equation and the boundary conditions are satisfied, a posteriori, in all local sub-domains and on their Γ_s, respectively. Theoretically, as long as the union of all local domains covers the global domain, i.e., $\cup\Omega_s \supset \Omega$, the equilibrium equation and the boundary conditions will be satisfied, a posteriori, in the global domain Ω and on the boundary Γ, respectively.

In this paper, the Petrov-Galerkin method is used in each local sub-domain. Unlike in the conventional Galerkin method in which the trial and test functions are chosen from the same space, the Petrov-Galerkin method uses the trial function and test function from different spaces. In particular, the test functions need not vanish on the boundary where the essential boundary conditions are specified.

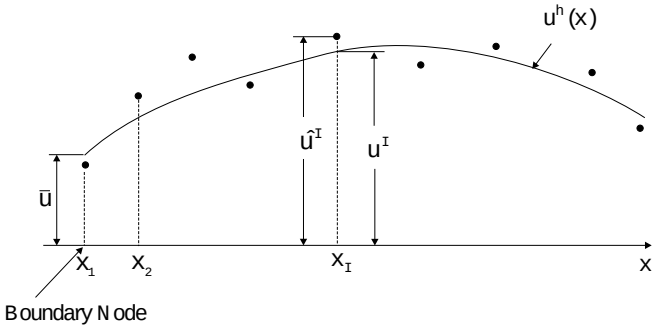


Figure 2 : The distinction between u^I and $\hat{u}^l$.

4 Meshless Approximation of Trial Functions

In this section, various available methods of approximating a *trial function* over an arbitrary domain, using only its value at a finite number of points in the domain, without, however, using a mesh, are discussed. In general, in a meshless method, which still preserves the local character of the numerical implementation, one 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. A variety of local interpolation schemes that interpolate the data at randomly scattered points in two or more independent variables is available.

4.1 Moving least –square method (MLS)

The moving least-square method is generally considered to be one of the best schemes to interpolate data with a reasonable accuracy. The MLS interpolation does not pass through the nodal data, as shown in Fig. 2. Here we give a brief summary of the MLS approximation. For details of the MLS approximation, see Belytschko, Lu & Gu (1994) and Atluri, Kim & Cho (1999).

Consider a sub-domain $\Omega_{\mathbf{x}}$, which is defined as the neighborhood of a point $\mathbf{x}$ and denoted as the domain of definition of the MLS approximation for the trial function at $\mathbf{x}$, and which is located in the problem domain Ω. To approximate the distribution of the function u in $\Omega_{\mathbf{x}}$, over a number of randomly located nodes $\{\mathbf{x}_I\}$, I=1, 2,... N, the moving least squares approximant $u^h(\mathbf{x})$ of u, $\forall\mathbf{x} \in \Omega_{\mathrm{x}}$, can be defined by

$$u^h(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\,\mathbf{a}(\mathbf{x})\;\forall\mathbf{x} \in \Omega_{\mathbf{x}} \quad (11)$$

where $\mathbf{p}^{\mathrm{T}}(\mathbf{x})=[p_1(\mathbf{x}),\ p_2(\mathbf{x}),\ \ldots,\ p_m(\mathbf{x})]$ is a complete monomial basis, m is the number of terms in the basis; we denote by t the highest-order polynomial which is completely include in the basis, the relation between m, and t can be described as: $m=\frac{(t+1)(t+2)}{2}$ in 2 dimension, while $m=\frac{(t+1)(t+2)(t+3)}{6}$ in 3 dimension [Usually, for l-dimension problem, we have $m=\frac{(t+1)\cdots(t+l)}{1\cdots l}$]; and $\mathbf{a}(\mathbf{x})$ is a vector containing coefficients $a_j(\mathbf{x})$, j=1, 2, $\ldots m$ which are functions of the space co-ordinates $\mathbf{x}=[x^1, x^2, x^3]^{\mathrm{T}}$. The commonly used bases in 2-D or 3-D are the linear basis, due to their simplicity. In Zhu, Zhang & Atluri (1998a, b) and Atluri & Zhu, (1998a, b), both the linear and quadratic basis are used, and the results show that both bases possess high accuracy. *In the present research, we just use the linear basis* $\mathbf{p}^{\mathrm{T}}(\mathbf{x})=[1, x^1, x^2]$ *for 2-D*. In fact, the linear basis assures that the MLS approximation has the linear completeness. Thus, it can reproduce any smooth function *and its first derivative* with arbitrary accuracy, as the approximation is refined. It is also possible to use other functions in a basis. *For example, in problems with singular solutions, singular functions can be included in the basis*.

The coefficient vector $\mathbf{a}(\mathbf{x})$ is determined by minimizing a weighted discrete L_2 norm, which can be defined as

$$J(\mathbf{a}(\mathbf{x}))=\sum_{I=1}^{N} w_I(\mathbf{x})\left[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_I)\,\mathbf{a}(\mathbf{x})-\hat{u}^I\right]^2 \tag{12}$$

where $w_I(\mathbf{x})$ is a weight function associated with the node I, with $w_I(\mathbf{x})>0$ for all $\mathbf{x}$ in the support of $w_I(\mathbf{x})$, $\mathbf{x}_I$ denotes the value of $\mathbf{x}$ at node I, N is the number of nodes in $\Omega_{\mathbf{x}}$ for which the weight functions $w_I(\mathbf{x})>0$.

Here it should be noted that $\hat{u}^I$, I=1, 2,$\ldots N$, in equation (12), are the fictitious nodal values, and not the actual nodal values of the unknown trial function $u^h(\mathbf{x})$, in general (see Fig. 2 for a simple one-dimensional case for the distinction between u^I and $\hat{u}^I$).

Solving for $\mathbf{a}(\mathbf{x})$ by minimizing J in equation (12), and substituting it into equation (11), give a relation which may be written in the form of an interpolation function similar to that used in the FEM, as

$$u^h(\mathbf{x})=\sum_{I=1}^{N}\phi^I(\mathbf{x})\,\hat{u}^I,\quad u^h(\mathbf{x}_I)\equiv u^I\neq\hat{u}^I,\quad \mathbf{x}\in\Omega_{\mathbf{x}} \tag{13}$$

where

$$\phi^I(\mathbf{x})=\sum_{j=1}^{m} p_j(\mathbf{x})\left[\mathbf{A}^{-1}(\mathbf{x})\,\mathbf{B}(\mathbf{x})\right]_{jI} \tag{14}$$

with the matrix $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ being defined by:

$$\mathbf{A}(\mathbf{x})=\sum_{I=1}^{N} w_I(\mathbf{x})\,\mathbf{p}(\mathbf{x}_I)\,\mathbf{p}^{\mathrm{T}}(\mathbf{x}_I) \tag{15}$$

$$\mathbf{B}(\mathbf{x})=\left[w_1(\mathbf{x})\,\mathbf{p}(\mathbf{x}_1),w_2(\mathbf{x})\,\mathbf{p}(\mathbf{x}_2),\cdots,w_N(\mathbf{x})\,\mathbf{p}(\mathbf{x}_N)\right] \tag{16}$$

The partial derivatives of $\phi^I(\mathbf{x})$ are obtained as

$$\phi^I_{,k}=\sum_{j=1}^{m}\left[p_{j,k}\left(\mathbf{A}^{-1}\mathbf{B}\right)_{jI}+p_j\left(\mathbf{A}^{-1}\mathbf{B}_{,k}+\mathbf{A}^{-1}_{,k}\mathbf{B}\right)_{jI}\right] \tag{17}$$

in which $\mathbf{A}^{-1}_{,k}=\left(\mathbf{A}^{-1}\right)_{,k}$ represents the derivative of the inverse of A with respect to x^k, which is given by

$$\mathbf{A}^{-1}_{,k}=-\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1} \tag{18}$$

The second partial derivatives of $\phi^I(\mathbf{x})$ (which are used in the MLPG2 and MLPG3 methods to be discussed later in this paper) are obtained as

$$\begin{aligned}\phi^I_{,kl}=\sum_{j=1}^{m}\,&[p_{j,kl}\left(\mathbf{A}^{-1}\mathbf{B}\right)_{jI}+p_{j,k}\left(\mathbf{A}^{-1}\mathbf{B}_{,l}+\mathbf{A}^{-1}_{,l}\mathbf{B}\right)_{jI}\\&+p_{j,l}(\mathbf{A}^{-1}\mathbf{B}_{,k}+\mathbf{A}^{-1}_{,k}\mathbf{B})_{jI}\\&+p_j(\mathbf{A}^{-1}\mathbf{B}_{,kl}+\mathbf{A}^{-1}_{,kl}\mathbf{B}+\mathbf{A}^{-1}_{,l}\mathbf{B}_{,k}+\mathbf{A}^{-1}_{,k}\mathbf{B}_{,l})_{jI}]\end{aligned} \tag{19}$$

with

$$\begin{aligned}\mathbf{A}^{-1}_{,kl}=&-\mathbf{A}^{-1}\mathbf{A}_{,l}\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{A}_{,kl}\mathbf{A}^{-1}\\&+\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}\mathbf{A}_{,l}\mathbf{A}^{-1}\end{aligned} \tag{20}$$

The MLS approximation is well defined, only when the matrix in Eq. (15) is non-singular. $\phi^I(\mathbf{x})$ is usually called the shape function of the MLS approximation, corresponding to the nodal point $\mathbf{x}_I$. From Eqs. (14) and (16), it may be seen that $\phi^I(\mathbf{x})=0$ when $w_I(\mathbf{x})=0$. The fact that $\phi^I(\mathbf{x})$ vanishes, for $\mathbf{x}$ not in the support of nodal point $\mathbf{x}_I$ preserves the local character of the moving least squares approximation. The nodal shape function is complete up to the order of the basis $\mathbf{P}(\mathbf{x})$. *The smoothness of the nodal shape function is determined by that of the basis, as well as that of the weight function.*

The choice of the weight function is more or less arbitrary, as long as the weight function is positive and continuous. Both Gaussian and spline weight functions with

compact supports can be considered in the present work. The Gaussian weight function corresponding to node I may be written as

$$w_I(\mathbf{x}) = \begin{cases} \frac{\exp\left[-(d_I/r_I)^{2k}\right]-\exp\left[-(r_I/c_I)^{2k}\right]}{1-\exp\left[-(r_I/c_I)^{2k}\right]}, & 0 \le d_I \le r_I \\ 0, & d_I \ge r_I \end{cases} \tag{21}$$

where $d_I = |\mathbf{x}-\mathbf{x}_I|$ is the distance from node $\mathbf{x}_I$ to point $\mathbf{x}$, c_I is a constant controlling the shape of the weight function w_I (and therefore the relative weights), and r_I is the size of the support for the weight function w_I (and thus determines the support of node $\mathbf{x}_I$).

A 4th-order spline weight function is defined as

$$w_I(\mathbf{x}) = \begin{cases} 1-6\left(\frac{d_I}{r_I}\right)^2+8\left(\frac{d_I}{r_I}\right)^3-3\left(\frac{d_I}{r_I}\right)^4, & 0 \le d_I \le r_I \\ 0, & d_I \ge r_I \end{cases} \tag{22}$$

The size of the support, r_I of the weight function w_I associated with node I should be chosen such that r_I should be large enough to have sufficient number of nodes covered in the domain of definition of every sample point ($N \ge m$), in order to ensure the regularity of $\mathbf{A}$. A very small r_I may result in a relatively large numerical error, while using the Gauss numerical quadrature to calculate the entires in the system stiffness matrix. On the other hand, r_I should also be small enough to maintain the local character of the MLS approximation. It can be easily seen that the spline weight function given in (22) possesses C^1 continuity. *So, with the use of the linear basis $\boldsymbol{P}(\boldsymbol{x})$, and the 4th-order spline for $w_I(\boldsymbol{x})$, the MLS, Shepard, and PU shape functions, and the corresponding trial functions are C^1 continuous over the entire domain.*

In Atluri and Zhu (1998b) and Zhu, Zhang & Atluri(1998a), both the Gaussian weight function and the spline weight function were used. Both of them possess high accuracy. However, the definition of all constants c_I in Eq. (21) is more or less arbitrary, and they do affect the computational results significantly. When inappropriate c_I are used in the Gaussian weight function, the results may become very unsatisfactory. The optimal choice of these parameters is still an open research topic. So, in this study, we just use the 4th-order spline weight function.

A generalization of the MLS interpolation scheme using the data for the derivative of a function, in addition to the value of the function itself, at a finite number of nodes, can be found in Atluri, Cho & Kim (1999).

4.2 Shepard function

If the MLS nodal shape functions (14) only represent a globally constant function, i.e. m=1 in the basis $\mathbf{P}(\mathbf{x})$, we obtain the so called Shepard function (Shepard, 1968) given by

$$\phi_0^I(\mathbf{x}) = \frac{w_I(\mathbf{x})}{\sum_{J=1}^{N} w_J(\mathbf{x})} \tag{23}$$

The Shepard functions satisfy the zeroth order completeness [Here, by zeroth-order completeness implies the ability to represent a constant function exactly (m=1 in the basis $\mathbf{P}(\mathbf{x})$); first-order completeness implies the ability to represent a complete linear function in x^1, x^2 (i.e., m=3 in the basis $\mathbf{P}(\mathbf{x})$]. The Shepard shape functions have a simpler structure than the higher order MLS shape functions, but they are still in a rational form. The Shepard function in the MLS interpolation is one of the partition of unity functions.

4.3 Partition of Unity methods (PU)

Meshless methods can also be based on partitions of unity. This viewpoint has been developed by Bauska and Melenk (1997). The interpolation can be defined as

$$u^h(\mathbf{x}) = \sum_{I=1}^{N}\sum_{j=1}^{m} \phi_0^I(\mathbf{x})\, p_j(\mathbf{x})\, \beta^{jI} \tag{24}$$

where $p_j(\mathbf{x})$ are the polynomials [$p_1(\mathbf{x}) = 1$; $p_2(\mathbf{x}) = x^1$; $p_3(\mathbf{x}) = x^2$, etc], $\phi_0^I(\mathbf{x})$ are defined in Eq. (23) and β^{jI} are the coefficients. The consistency in the above shape function depends on the order of polynomials in $p_j(\mathbf{x})$. Note that this method requires at least three unknowns per node, in order to attain linear consistency; whereas, an MLS approximation of order 1 [i.e., linear basis, with m=3 in $\mathbf{P}(\mathbf{x})$] requires only one unknown per node for linear consistency. In computations, $p_j(\mathbf{x})$ is generally replaced by $p_j(\mathbf{x}\text{-}\mathbf{x}_I)$ for better conditioning.

PU method is not a good choice for present MLPG method, because the test function also needs m unknowns

per node, or indirectly, m test functions are needed in each Ω_s; otherwise we cannot obtain sufficient equations to determine the unknowns. In addition, it is more time-consuming to form and solve the resulting algebraic equations, because, in this case, the dimension of the stiffness matrix will be m times of that in the case when MLS or Shepard functions are used. However, the PU can be used for the Galerkin method, due to the fact that the test function can be taken as same as the trial function. But it is still more time consuming. In Section 5, we will find that a large number of Gaussian points are needed to guarantee a convergent result in the Galerkin method, which is very expensive; *hence, we will not conduct the numerical analysis for PU interpolation in this paper*.

4.4 Reproducing Kernel Particle Methods (RKPM)

The reproducing kernel particle method (RKPM) is developed in Liu, Chen, Uras & Chang (1996). Here we just give a brief summary of the RKPM approximation.

The shape function in RKPM was derived from the reproducing conditions. In order to reproduce the linear basis function exactly, the following conditions are required:

$$\phi \cdot \mathbf{P}_1 = 1 \tag{25a}$$

$$\phi \cdot \mathbf{P}_2 = 0 \tag{25b}$$

$$\phi \cdot \mathbf{P}_3 = 0 \tag{25c}$$

where

$$\phi = \left[\begin{array}{cccc} \phi^1 & \phi^2, & \cdots, & \phi^N \end{array} \right] \tag{26a}$$

$$\mathbf{P}_1^T = \left[\begin{array}{cccc} p_1(\mathbf{x}_1), & p_1(\mathbf{x}_2), & \cdots, & p_1(\mathbf{x}_N) \end{array} \right] = \left[\begin{array}{cccc} 1, & 1, & \cdots & 1 \end{array} \right] \tag{26b}$$

$$\mathbf{P}_2^T = \left[\begin{array}{cccc} p_2(\mathbf{x}_1,) & p_2(\mathbf{x}_2), & \cdots, & p_2(\mathbf{x}_N) \end{array} \right] = \left[\begin{array}{cccc} x_1^1, & x_2^1, & \cdots, & x_N^1 \end{array} \right] \tag{26c}$$

$$\mathbf{P}_3^T = \left[\begin{array}{cccc} p_3(\mathbf{x}_1,) & p_3(\mathbf{x}_2), & \cdots, & p_3(\mathbf{x}_N) \end{array} \right] = \left[\begin{array}{cccc} x_1^2, & x_2^2, & \cdots, & x_N^2 \end{array} \right] \tag{26d}$$

In (26a-d), the superscripts for ϕ, and the subscripts for x^1 and x^2, refer to the nodes, 1, $\cdots, N$. It is noted that, here, the origin is shifted to the point of evaluation.

The linear RKPM shape function can be chosen as

$$\phi^I(\mathbf{x}) = \sum_{j=1}^{3} \alpha_j(\mathbf{x})\, q_j(\mathbf{x}_I)\, w_I(\mathbf{x}) \tag{27}$$

where $q_i(\mathbf{x})$ are three linearly independent functions. The coefficients α_j can be determined by equations (25). If we let $q_i(\mathbf{x})$ be the same as the linear basis $p_i(\mathbf{x})$ in the MLS, and the weight function also to be the same as that in the MLS, we can obtain

$$\phi^I(\mathbf{x}) = \sum_{j=1}^{3} p_j(\mathbf{x}) \left[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\right]_{jI} \tag{28}$$

which is the same as that in MLS. The detailed procedure to show this equality can be found in Belytschko et al. (1996).

Therefore, if we choose the same weight function for the RKPM and impose the consistency requirement of order k, the resulting shape function is identical to the MLS approximation of order k [note that, in the present notation, k=0 implies m=1 in $\mathbf{P}(\mathbf{x})$; k=1 implies m=3 in $\mathbf{P}(\mathbf{x})$]. As in MLS, the kernels can also be designed to reproduce functions other than the polynomial basis. The equivalence between the MLS and the RKPM is also discussed in Jin, Li & Aluru (2001).

4.5 Radial Basis Functions (RBF) with compact support

Compact support, positive definite Radial Basis Functions have been suggested only recently (Wendland, 1995; Wu, 1995). It was demonstrated that, for a given dimension and smoothness C^{2k}, a positive definite radial basis function in the form of a univariate polynomial of minimal degree always exists, and is unique within a constant factor. The compactly supported positive definite RBFs studied in this paper are (Wu, 1995):

RBF1:

$$R_I(\mathbf{x}) = \begin{cases} \left(1-\frac{d_I}{r_I}\right)^5 \left(8+40\frac{d_I}{r_I}+48\frac{d_I^2}{r_I^2}+25\frac{d_I^3}{r_I^3}+5\frac{d_I^4}{r_I^4}\right), & 0 \le d_I \le r_I \\ 0, & d_I \ge r_I \end{cases} \tag{29}$$

and RBF2:

$$R_I(\mathbf{x}) = \begin{cases} \left(1-\frac{d_I}{r_I}\right)^6 \left(6+36\frac{d_I}{r_I}+82\frac{d_I^2}{r_I^2}+72\frac{d_I^3}{r_I^3}+30\frac{d_I^4}{r_I^4}+5\frac{d_I^5}{r_I^5}\right), & 0 \le d_I \le r_I \\ 0, & d_I \ge r_I \end{cases}$$

(30)

where is the distance from node $\mathbf{x}_I$ to point $\mathbf{x}$, and r_I is the size of the support for the radial basis function at node $\mathbf{x}_I$.

Another kind of radial basis functions can be found in Wendland (1998), where a recursion formula is given to compute the radial basis functions.

We consider the approximation of function $u(\mathbf{x})$ in a local region sub-domain Ωapproximate the distribution of function u in Ω, over a number of randomly located nodes $\{\mathbf{x}_I\}$, I=1, 2, ...N, the radial basis function approximation of $u(\mathbf{x})$, can be defined by:

$$u(\mathbf{x}) = \mathbf{R}^T(\mathbf{x})\mathbf{a} \qquad \forall \mathbf{x} \in \Omega_x \tag{31}$$

where , is the set of radial basis functions centered around $\mathbf{x}_I$; and $\mathbf{a}$ is a vector containing the coefficients of a_I, I=1, 2, ...N.

From interpolation equation (31) for the compact support, positive definite RBFs, the following system of linear equations for the coefficients $\mathbf{a}$ is obtained as:

$$\mathbf{R}_0\mathbf{a} = \mathbf{u} \tag{32}$$

where

$$\mathbf{u}^T = \begin{bmatrix} u^1 & u^2 & \cdots & u^N \end{bmatrix}$$

$$\mathbf{R}_0 = \begin{bmatrix} R_1(\mathbf{x}_1) & R_2(\mathbf{x}_1) & \cdots & R_N(\mathbf{x}_1) \\ R_1(\mathbf{x}_2) & R_2(\mathbf{x}_2) & \cdots & R_N(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ R_1(\mathbf{x}_N) & R_2(\mathbf{x}_N) & \cdots & R_N(\mathbf{x}_N) \end{bmatrix}$$

Here, u^I, I=1, 2, ...N, are the nodal values. The RBFs are positive definite, which ensures that the matrix $\mathbf{R}_0$ is invertible (non-singularity). Hence, we can obtain the vector $\mathbf{a}$ from equation (32),

$$\mathbf{a} = \mathbf{R}_0^{-1}\mathbf{u} \tag{33}$$

So, the approximation $u(\mathbf{x})$ can be expressed as

$$u(\mathbf{x}) = \mathbf{R}^T(\mathbf{x})\mathbf{R}_0^{-1}\mathbf{u} = \sum_{I=1}^{N} \phi^I(\mathbf{x})\, u^I \tag{34}$$

where the nodal shape functions are given by

$$\phi(\mathbf{x}) = \mathbf{R}^T(\mathbf{x})\mathbf{R}_0^{-1} \tag{35}$$

This shape function depends uniquely on the distribution of scattered nodes within the sub-domain, and possesses the Kronecker Delta property, which is same as in the traditional Galerkin FEM; a property not possessed by the shape functions of the MLS approximation.

The partial derivative of the shape function can be obtained as follows,

$$\begin{aligned} \phi(\mathbf{x})_{,k} &= [\mathbf{R}^T(\mathbf{x})]_{,k}\mathbf{R}_0^{-1} \\ &= \begin{bmatrix} R_{1,k}(\mathbf{x}) & R_{2,k}(\mathbf{x}) & \cdots & R_{N,k}(\mathbf{x}) \end{bmatrix}\mathbf{R}_0^{-1} \end{aligned} \tag{36}$$

where, $(\,)_{,i}$denotes , and

$$R_{I,k}(\mathbf{x}) = \frac{\partial R_I(\mathbf{x})}{\partial d_I}\frac{\partial d_I}{\partial x^k} \tag{37}$$

with

$$\frac{\partial d_I}{\partial x^k} = \frac{x^k - x_I^k}{d_I}$$

Thus, it can be very easy to obtain the partial derivative of $\phi^I(\mathbf{x})$. *It is noted that the RBFs, and the corresponding shape functions are of smoothness C^2 and C^4 for RBF1 and RBF2, respectively*. For the linear elasticity problem, the smoothness of the first derivative of shape functions is very important to evaluate the energy correctly.

The nodal shape functions can be written as

$$\phi^I(\mathbf{x}) = \sum_{J=1}^{N} R_J(\mathbf{x})\, r_{JI} \tag{38}$$

where r_{JI} are the elements of matrix R_0^{-1}. And the corresponding derivative of the nodal shape function thus can be expressed as

$$\phi^I_{,k}(\mathbf{x}) = \sum_{J=1}^{N} R_{J,k}(\mathbf{x})\, r_{JI} \tag{39}$$

Equations (38) and (39) show that the shape function and its derivative have polynomial forms of d_I. However, they are still rational forms of $\mathbf{x}$.

The Hermite-Birkhoff type interpolation of scattered multidimensional data, through radial basis functions, can be found in Wu (1992); and this type of interpolation is desirable when the derivative is also used as a degree of freedom at each node.

4.6 Convergence and Summary

In meshless interpolations of trial functions, *the rates of the convergence of the solution may depend upon the nodal distance h, as well as the size of sub-domain defined by* ρ, *where* ρ *is the scaling parameter for the size of the support of the trial function.* Atluri, Kim & Cho (1999) assumed that the interpolation-error-estimate for MLS, Shepard or PU methods has the same *h*-convergence characteristic as in the Galerkin finite elements:

$$\left\|u - u^h\right\|_k \leq G_{\rho,k} h^{t+1-k} |u|_{t+1} \tag{40}$$

where $G_{\rho,k}$ is a function of ρnd k, $\|\cdot\|_k$ is the norm in the Sobolev space, and t indicates the order of the complete polynomial in the interpolation function, and $m = \frac{(t+1)(t+2)}{2}$ in 2 dimension. For examples, for Shepard function (the zeroth order consistency), $m = 1$ and t=0, for linear complete, $m = 3$, $t = 1$.

For radial basis functions, Wendland (1999) proved that the interpolation-error-estimate, in a meshless Galerkin method, also has the *h*-convergence as in equation (40), with

$$t + 1 \leq l + \frac{d+1}{2}$$

where d is the dimension of the radial basis function, and $2l$ is the degree of the smoothness of the radial basis function.

Obviously, *these five trial function approximations, i.e. MLS, Shepard, PU, RKPM, and RBF*, respectively, can be classified into two categories: one in which the corresponding "nodal" shape functions possess the Kronecker Delta properties, and the other in which they do not. Only the radial basis function possesses the Kronecker Delta property, while the rest do not. Unfortunately, the nodal shape functions from all the interpolations are still in a rational form, and are highly complex (although the one from Shepard function is relatively simpler), which means that the integrand in the weak-form cannot be simplified at the present stage, for the Galerkin-equivalent of the MLPG method. A summary of the approximations is given in Table 1.

It is noted that the nodal shape functions (trial functions) from all the interpolations listed here in this paper, possess a high order of continuity: *MLS, Shepard and PU are C^1, RBF1 is C^2, and RBF2 is C^4, respectively*. This high order of continuity provides solutions with smooth derivatives, and is very different to the FEM. Thus, the MLPG methods have the advantage of providing better (smooth) approximation of stresses. Consequently, the postprocessing in MLPG is relatively straightforward, and no additional stress smoothing is required. However, this high order of continuity can also be a disadvantage in problems where the coefficients of the partial differential equation are discontinuous: such as in elastostatics, wherein the coefficients in the partial differential equation are discontinuous along the interface between two different materials, and the derivatives of the displacements may be discontinuous across the material interfaces. Thanks to the MLPG concept, the shape of the support of the test functions can be chosen to be arbitrary, and be different at each nodal point, which provides the key to this difficulty. We can use polygonal local-domains (Kim and Atluri, 2000) along the interface, and use a C^0 trial function (such as the finite element shape function as the weight in the MLS); but the circular local-domains and C^1 trial functions are still used at the remained nodal points. The results based on this idea, to treat the (displacement and derivative) discontinuities will be presented in a forthcoming paper.

5 Numerical implementation: the Meshless Local Petrov-Galerkin (MLPG) Approach

Based on the local weak form, the meshless local Petrov-Galerkin (MLPG) method is a truly meshless method.

Table 1 : Various meshless approximations of the trial function:

Method	Shape function	Completeness	Continuity	Convergence	Remark
MLS or (RKPM)	$\phi^I(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) \left[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\right]_{jI}$ with 4th order spline weight functions w_I	t	C^1	h-convergence Eq. (40)	$\phi^J(\mathbf{x}_I) \neq \delta_{IJ}$
Shepard function	$\phi_0^I(\mathbf{x}) = w_I(\mathbf{x}) \Big/ \sum_{J=1}^{N} w_J(\mathbf{x})$ w_I: 4th order spline	0	C^1	h-convergence Eq. (40)	$\phi^J(\mathbf{x}_I) \neq \delta_{IJ}$
Partitions of unity	$u^h(\mathbf{x}) = \sum_{I=1}^{N}\sum_{j=1}^{m} \phi_0^I(\mathbf{x}) p_j(\mathbf{x}) \beta^{jI}$	t	C^1	h-convergence Eq. (40)	$\phi^J(\mathbf{x}_I) \neq \delta_{IJ}$
RBF	$\phi^I(\mathbf{x}) = \sum_{J=1}^{N} R_J(\mathbf{x}) r_{JI}$	none	RBF1: C^2 RBF2: C^4	h-convergence Eq. (40)	$\phi^J(\mathbf{x}_I) = \delta_{IJ}$

$*$t is the order of the complete polynomial in the interpolation function.

In the conventional Galerkin method, the trial and test functions are chosen from the same space. In general, in MLPG, the nodal trial and test functions can be different, the nodal trial function may correspond to any one of: 1. MLS (or entirely equivalently the RKPM); 2. PU; 3. Shepard function; or 4. RBF interpolations; and the test function may be totally different. Furthermore, the size as well as the shape of the sub-domains over which the nodal trial and test functions are, respectively, non-zero, may be different. Different choices of the basis functions for the trial function and the test functions will lead to different approximation methods. We will use the interpolations introduced in the last section, to generate the trial functions. *Based on the concept of the MLPG, the test functions over each local Ω_s can be chosen through a variety of ways*:

(1) the test function over Ω_s is the same as the weight function in the MLS approximation: The resultant Meshless Local Petrov-Galerkin Method is denoted as MLPG1;

(2) the test function over Ω_s is the collocation Dirac's Delta function (collocation method): The resultant Meshless Local Petrov-Galerkin Method is denoted as MLPG2;

(3) the test function over Ω_s is the same as the error function in the differential equation, using discrete least squares: The resultant Meshless Local Petrov-Galerkin Method is denoted as MLPG3;

(4) the test function over Ω_s is the modified fundamental solution to the differential equation (LBIE): The resultant Meshless Local Petrov-Galerkin Method is denoted as MLPG4; Thus, MLPG4 is synonymous with the Local Boundary Integral Equation (LBIE) method [Zhu, Zhang & Atluri (1998a)].

(5) the test function over Ω_s is the Heaviside step function (constant over each local sub-domain Ω_s): The resultant Meshless Local Petrov-Galerkin Method is denoted as MLPG5;

(6) the test function over Ω_s is identical to the trial function (Galerkin method): The resultant Meshless Local Petrov-Galerkin Method is denoted as MLPG6.

In the following, we will develop the MLPG methods

along these six paths. As a known test function is used in the local weak form (LWF), the use of the LWF for one point (and here for one domain Ω_s) will yield only one algebraic equation. It is noted that the trial function u within the sub-domain Ω_s, in the interpolations without Kronecker Delta properties, is determined by the fictitious nodal values $\hat{u}^I$, within the domain of definition for all points $\mathbf{x}$ falling within Ω_s. One can obtain as many equations as the number of nodes. Hence, we need as many local domains Ω_s as the number of nodes in the global domain, in order to obtain as many equations as the number of unknowns. In the present paper, dealing with 2D problems, the local domain is chosen as a circle, centered at a node $\mathbf{x}_I$.

5.1 MLPG1: Using the MLS weight function as the test function in each Ω_s

To obtain the discrete equations from the LSWF (9b), alternatively the MLS approximation (13), or the Shepard function (23), or the RBF (34) are respectively used to approximate the trial function u. The MLS weight function (23) is taken to be test function in each Ω_s. Substitution of interpolations (13), (23) or (34) into the LSWF (9b) for all the nodes, leads to the following discretized system of linear equations:

$$\mathbf{K}\cdot\hat{\mathbf{u}}=\mathbf{f} \tag{41}$$

where, the entries of the global 'stiffness' matrix $\mathbf{K}$ and the global 'load' vector $\mathbf{f}$, respectively, are defined as:

$$K_{IJ}=\int_{\Omega_s}\phi^J_{,k}(\mathbf{x})\,v_{,k}(\mathbf{x},\mathbf{x}_I)d\Omega-\int_{\Gamma_{su}}\frac{\partial\phi^J(\mathbf{x})}{\partial n}v(\mathbf{x},\mathbf{x}_I)\,d\Gamma +\alpha\int_{\Gamma_{su}}\phi^J(\mathbf{x})\,v(\mathbf{x},\mathbf{x}_I)\,d\Gamma \tag{42}$$

and

$$f_I=\int_{\Gamma_{su}}\overline{q}(\mathbf{x})\,v(\mathbf{x},\mathbf{x}_I)\,d\Gamma-\int_{\Omega_s}p(\mathbf{x})\,v(\mathbf{x},\mathbf{x}_I)d\Omega +\alpha\int_{\Gamma_{su}}\overline{u}(\mathbf{x})\,v(\mathbf{x},\mathbf{x}_I)\,d\Gamma \tag{43}$$

in which ϕ^J is alternatively, the shape function from the MLS, Shepard function or RBF, respectively. In the above, $I=1,\cdots,M$ (where M is the total number of the nodes in the domain), and the only non-zero K_{IJ} coefficient for each I, extend over only $J=1,\cdots,N$, where N is the number of weight functions centered at $\mathbf{x}_J$, which do not vanish at $\mathbf{x}_I$. Futher details of this method using the MLS approximation for the trial function $\mathbf{u}$ can be found in Atluri and Zhu (1998).

It is noted that for the RBF approximation, $\hat{\mathbf{u}}$ should be replaced by the true nodal values, $\mathbf{u}$ in these equations. For the RBF approximation, due to the Kronecker Delta properties, the Dirichlet or essential boundary conditions (2a) can be imposed directly, just as in the conventional FEM.

5.2 MLPG2: The Collocation method: Using the collocation Dirac's Delta function as the test function in each Ω_s

In this method, the discrete equations are obtained from the local unsymmetric weak formulation (LUSWF1) (8); while, alternatively, (i) the MLS approximation (13); (ii) Shepard function (23); and (iii) RBF (34), respectively, are used to approximate the trial function u. *The test function in each Ω_s is taken to be the Kronecker Delta function*. Substitution of interpolations (13), (23) or (34) into the LUSWF1 (8) for the internal nodes, leads to the discretized system of linear equations (41). Now, the entries of the global 'stiffness' matrix $\mathbf{K}$ and the global 'load' vector $\mathbf{f}$ are defined as:

$$K_{IJ}=\begin{cases}\phi^J_{,ii}(\mathbf{x}_I), & \mathbf{x}_I\in\Omega\\ \phi^J_{,n}(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_q\\ \phi^J(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_u\end{cases} \tag{44}$$

and

$$f_I=\begin{cases}p(\mathbf{x}_I), & \mathbf{x}_I\in\Omega\\ \overline{q}(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_q\\ \overline{u}(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_u\end{cases} \tag{45}$$

Here, $I,J=1,2,\cdots,M$. Note that, for the RBF approximation of the trial function, $\phi^J(\mathbf{x}_I)=\delta_{IJ}$. In this method, second derivatives of the shape functions are needed in constructing the global stiffiness matrix for the interior nodes; however, the calculation of the derivatives of shape functions from the MLS approximation is quite costly. No numerical integration is needed in this method.

Thus, the collocation method can be treated simply as a special case of the MLPG approach.

5.3 MLPG3: Using discrete least squares, i.e. Error function is used as the test function in each Ω_s

In this method, the test function is assumed as in the discrete least square method. Once again, the trial function **u**, may be chosen alternatively as (i) MLS (13), or (ii) Shepard (23), or (iii) RBF (34), respectively. The equation (1) can be satisfied by minimizing an L_2 norm, which can be defined as

$$J=\int_{\Omega_s}\left(\sum_{I=1}^{M}\nabla^2[\phi^I(\mathbf{x})\,\hat{u}^I]-p\right)^2 d\Omega \tag{46}$$

To minimize (46), we must have

$$\frac{\partial J}{\partial \hat{u}^I}=0 \tag{47}$$

for every nodal value $\hat{u}^I$. Thus, we can obtain that

$$\int_{\Omega_s}\left(\sum_{J=1}^{M}\nabla^2[\phi^J(\mathbf{x})\,\hat{u}^J]-p\right)\nabla^2\phi^I(\mathbf{x})\,d\Omega=0 \tag{48}$$

Obviously, if we chose the test function as

$$v(\mathbf{x},\mathbf{x}_I)=\phi^I_{,ii}(\mathbf{x}) \tag{49}$$

Then, substituting (49) in to (8), we can also obtain (48).

Substitution of interpolations (13), (23) or (34) into the LUSWF1 (8) for the internal nodes, leads to the discretized system of linear equations (41). Now, the entries of the global 'stiffness' matrix **K** and the global 'load' vector **f** are defined as:

$$K_{IJ}=\begin{cases}\int_{\Omega_s}\phi^J_{,kk}(\mathbf{x})\,\phi^I_{,ii}(\mathbf{x})d\Omega, & \mathbf{x}_I\in\Omega\\ \phi^J_{,n}(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_q\\ \phi^J(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_u\end{cases} \tag{50}$$

and

$$f_I=\begin{cases}\int_{\Omega_s}p(\mathbf{x})\,\phi^I_{,ii}(\mathbf{x})\,d\Omega, & \mathbf{x}_I\in\Omega\\ \overline{q}(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_q\\ \overline{u}(\mathbf{x}_I), & \mathbf{x}_I\in\Gamma_u\end{cases} \tag{51}$$

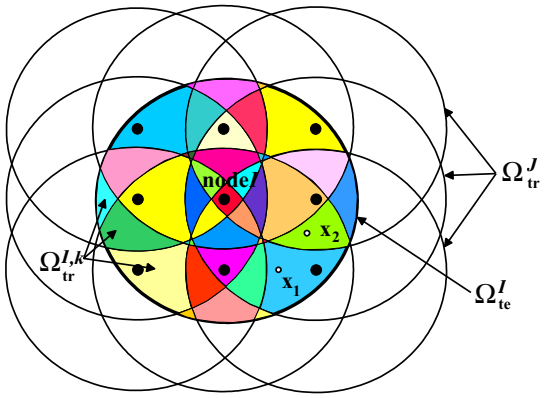


Figure 3 : The nodal shape function has a different form in each small region $\Omega^{I,k}_{tr}$, which is the intersection of Ω^I_{te} and Ω^J_{tr}.

Here, $I,J=1,2,\cdots,M$, and the test function and the trial function come from the same space.

From the detailed analysis in Atluri, Kim & Cho (1999), we can find that the larger the support of the test function is, the more complex the integrand is. This is because of the fact that the nodal shape function consists of a different form of rational function in each small region $\Omega^{I,K}_{tr}$, as shown in Fig. 3. The smaller the size of the support of the test function is, the fewer the small regions are, and the relatively simpler the integrand is. Thus, in this method, the integrand is far more complex, since the test function comes from the same function space as the trial function. Therefore, it may be difficult to perform the domain integration necessary for generating the stiffness matrix. More Gaussian points or the partition method developed in Atluri , Kim & Cho (1999) must be used to get a convergent result. This conclusion is also verified in our numerical examples. As a comparison, we still use equation (49) as the test function, but we reduce the size of its support. The results indicate that the calculation is quite costly in the case when Ω_{te}=Ω_{tr}. This comparison also suggests that it is better to choose a smaller support for the test function (i.e., it is better, in MLPG3, to use: $\Omega_{te}<\Omega_{tr}$).

5.4 MLPG4 or the LBIE: Using the modified fundamental solution to the differential equation as the test function in each Ω

We choose the modified fundamental solution u^* (which vanishes at $\partial\Omega_s$ as long as $\partial\Omega_s$ does not intersect with Γ as the test function, i.e.:

$$\nabla^2 u^*(\mathbf{x},\mathbf{x}_I) + \delta(\mathbf{x},\mathbf{x}_I) = 0 \quad \text{on} \quad \Omega_s \tag{52}$$
$$u^*(\mathbf{x},\mathbf{x}_I) = 0 \quad \text{on} \quad \partial\Omega_s$$

For the 2-D potential operator, the test function u^* can be easily obtained as

$$u^*(\mathbf{x},\mathbf{x}_I) = \frac{1}{2\pi}\ln\frac{r_0}{r} \tag{53}$$

where $r = |\mathbf{x}-\mathbf{x}_I|$ denotes the distance from the source point to the generic point under consideration, and r_0 is the radius of the local sub-domain, Ω_s (i.e. the support of the test function Ω^I_{te}.

Thus, using the test function u^*in the local unsymmetric weak formulation (LUSWF2) (10b), the following local boundary integral equation (LBIE) can be obtained:

$$\alpha(\mathbf{x}_I)u(\mathbf{x}_I) = \int_{\partial\Omega_s} u(\mathbf{x})\frac{\partial u^*(\mathbf{x},\mathbf{x}_I)}{\partial n}d\Gamma + \int_\Gamma \frac{\partial u(\mathbf{x})}{\partial n}u^*(\mathbf{x},\mathbf{x}_I)\,d\Gamma - \int_{\Omega_s} u^*(\mathbf{x},\mathbf{x}_I)\,p(\mathbf{x})\,d\Omega \tag{54}$$

In Eq. (54), the trial function **u**, may be chosen alternatively, as (i) the MLS (13), or (ii) the Shepard function (23), or (iii) the RBF (34), respectively.

Substituting interpolations (13), (23) or (34) into (52), the following linear equations can be obtained

$$\alpha_I u^I = f'_I + \sum_{J=1}^{N} K'_{IJ}\hat{u}^J \text{ for every node } I \tag{55}$$

where

$$\alpha_I = \begin{cases} 1 & \text{internal nodes} \\ 1/2 & \text{nodes on a smooth boundary} \\ \theta/2\pi & \text{nodes on boundary corners} \end{cases} \tag{56}$$

$$K'_{IJ} = \int_{\Gamma_{su}} u^*(\mathbf{x},\mathbf{x}_I)\frac{\partial\phi^J(\mathbf{x})}{\partial n}d\Gamma - \int_{\Gamma_{sq}}\phi^J(\mathbf{x})\frac{\partial u^*(\mathbf{x},\mathbf{x}_I)}{\partial n}d\Gamma - \int_{L_s}\phi^J(\mathbf{x})\frac{\partial u^*(\mathbf{x},\mathbf{x}_I)}{\partial n}d\Gamma \tag{57}$$

$$f'_I = \int_{\Gamma_{sq}} \overline{q}u^*(\mathbf{x},\mathbf{x}_I)\,d\Gamma - \int_{\Gamma_{su}}\overline{u}(\mathbf{x})\frac{\partial u^*(\mathbf{x},\mathbf{x}_I)}{\partial n}d\Gamma - \int_{\Omega_s} u^*(\mathbf{x},\mathbf{x}_I)\,p(\mathbf{x})\,d\Omega \tag{58}$$

Here, $I,J = 1,2,\cdots,M$. Upon imposing the essential boundary condition for u^I in the left hand side of (55) for those nodes where u is specified, we can obtain the linear system (41). Here, for the trial function in interpolations that do not possess the Kronecker Delta properties,

$$K_{IJ} = \begin{cases} -K'_{IJ} & \text{for nodes with } u^I \text{ known,} \\ -K'_{IJ} + \alpha_I\phi^J(\mathbf{x}_I) & \text{for nodes with } u^I \text{ unknown} \end{cases} \tag{59}$$

$$f_I = \begin{cases} f'_I - \alpha_I\overline{u}^I & \text{for nodes with } u^I \text{ known,} \\ f'_I & \text{for nodes } u^I \text{ unknown} \end{cases} \tag{60}$$

while for the interpolations with the Kronecker Delta properties,

$$K_{IJ} = \begin{cases} -K'_{IJ} & \text{for} \quad I \neq J \\ \alpha_I - K'_{IJ} & \text{for} \quad I = J \end{cases} \tag{61}$$

$$f_I = f'_I \tag{62}$$

The details of this method, while using the MLS approximation for the trial function **u**, can be found in Zhu, Zhang & Atluri (1998a). From equation (57) and (58), it is seen that no derivatives of the shape functions are needed, in constructing the stiffness matrix for the interior nodes, as well as for those boundary nodes with no essential boundary condition prescribed sections on their local boundaries. This is attractive in engineering applications, as the calculation of derivatives of shape functions from the MLS approximation is quite costly. However, singular integrals appear in the local boundary integral equation (defined only over a sphere centered at each point in question), to which special attention should be paid (Sladek, Sladek and Atluri (1999)).

5.5 MLPG5: Using the Heaviside step function as the test function in each Ω_s

In this method, the discrete equations are derived from the LSWF (9a); while, alternatively: 1. the MLS approximation (13); 2. Shepard function (23); and 3. RBF (34), respectively, are used to approximate the trial function u. The Heaviside step function is taken to be the test function in each Ω_s; *however, here we use the local symmetric weak form rather than the unsymmetric weak form.* Thus, when the Heaviside step function is used for the test function v in Ω_s, the LSWF (9a) can be rewritten as

$$\int_{L_s}\frac{\partial u}{\partial n}d\Gamma+\int_{\Gamma_{su}}\frac{\partial u}{\partial n}d\Gamma+\int_{\Gamma_{sq}}\overline{q}d\Gamma-\int_{\Omega_s}pd\Omega-\alpha\int_{\Gamma_u}(u-\overline{u})\,d\Gamma=0 \tag{63}$$

In MLPG5, the sub-domains Ω_s are *over-lapping and do not need a mesh, at all*. Substitution of interpolations (13), (23) or (34) into the LSWF (63) leads to the linear system (41), the entries of the global 'stiffness' matrix $\mathbf{K}$ and the global 'load' vector $\mathbf{f}$ are defined by

$$K_{IJ}=-\int_{L_s}\phi^J_{,k}(\mathbf{x})\,n_k d\Gamma-\int_{\Gamma_{su}}\phi^J_{,k}(\mathbf{x})\,n_k d\Gamma+\alpha\int_{\Gamma_{su}}\phi^J(\mathbf{x})\,d\Gamma \qquad I,J=1,\cdots,M \tag{64}$$

and

$$f_I=\int_{\Gamma_{su}}\overline{q}(\mathbf{x})\,d\Gamma-\int_{\Omega_s}p(\mathbf{x})d\Omega+\alpha\int_{\Gamma_{su}}\overline{u}(\mathbf{x})\,d\Gamma \tag{65}$$

It is well-known that the numerical integration plays an important role in the convergence of numerical solutions of meshless methods. The numerical integration in the usual finite element method is not a serious issue, because the domain integration of "energy" [which is of a polynomial form], to evaluate for the stiffness matrix, may be performed exactly, by using Gauss quadrature of the necessary order. However, it may be difficult to perform the domain integration for the stiffness matrix in meshless methods, due to the complexity [rational form] of the integrand. *From equation (64), it can be seen that the domain integral over Ω_s is altogether avoided; Eq. (64) involves only boundary integrals over each circle, which will greatly improve the effectiveness of this method. Hence, this method may be an attractive meshless method, because neither domain integrals nor singular integrals appear in the stiffness matrix, which make the solution stable, fast and accurate. It is thus seen that MLPG5 provides a simple and efficient alternative to the finite element and boundary element methods.*

Futher details of this method are given in Atluri and Shen (2002).

5.6 MLPG6: Meshless Local Galerkin method ["Galerkin-equivalent" of the MLPG]: Using the test function to be the same as the trial function in each Ω_s

In the Galerkin method, the trial function and test function come from the same space. To obtain the discrete equations from the LSWF (9b), alternatively: 1. the MLS approximation (13); 2. Shepard function (23); and 3. RBF (34), respectively, are used to approximate the trial function u, as well as the best function v. Substitution of interpolations (13), (23) or (34) into the LSWF (9b) for all nodes leads to the discretized system of linear equations (41); and the entries of the global 'stiffness' matrix $\mathbf{K}$ and the global 'load' vector $\mathbf{f}$ are defined as:

$$K_{IJ}=\int_{\Omega_s}\phi^J_{,k}(\mathbf{x})\,\phi^I_{,k}(\mathbf{x})\,d\Omega-\int_{\Gamma_{su}}\frac{\partial\phi^J(\mathbf{x})}{\partial n}\phi^I(\mathbf{x})\,d\Gamma+\alpha\int_{\Gamma_{su}}\phi^J(\mathbf{x})\,\phi^I(\mathbf{x})\,d\Gamma \tag{66}$$

and

$$f_I=\int_{\Gamma_{su}}\overline{q}(\mathbf{x})\,\phi^I(\mathbf{x})\,d\Gamma-\int_{\Omega_s}p(\mathbf{x})\,\phi^I(\mathbf{x})d\Omega+\alpha\int_{\Gamma_{su}}\overline{u}(\mathbf{x})\,\phi^I(\mathbf{x})\,d\Gamma \tag{67}$$

Here, $I,J=1,2,\cdots,M$. *In MLPG6, the integrand is far more complex than in the 5 previous MLPGs, due to fact that the test function comes from the same space of the trial function.* In fact, Atluri, Kim & Cho (1999) studied MLPG6 in great detail. Their results showed that the partition method must be used to get a convergent result. Based on their research and the results in Section 7.2, in this paper, we will omit a further study of MLPG6 here. *We now think that the Galerkin method is not a good option for the MLPG method, from the viewpoint of calculation and application; and thus, henceforth, discard MLPG6 from our further considerations.*

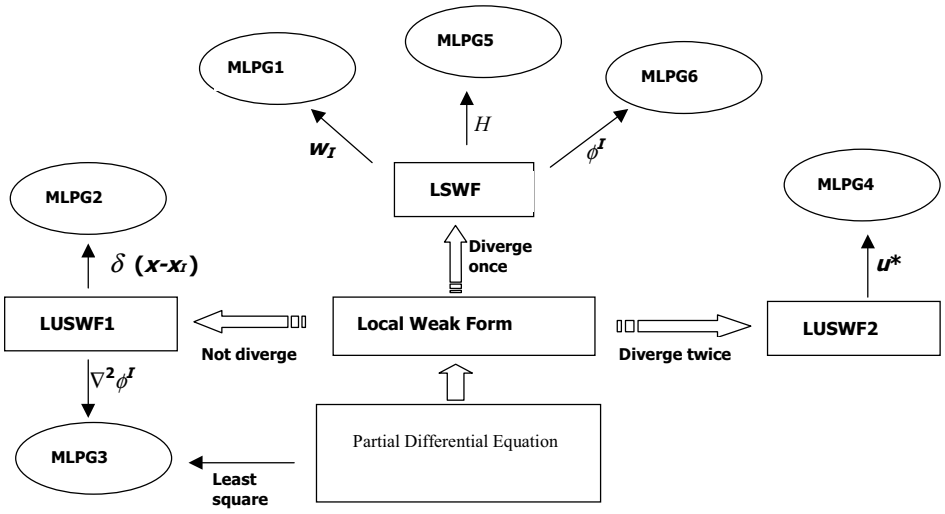


Figure 4 : Interrelationship of meshless methods

5.7 Summary

A summary of the variety of MLPG methods is given in Table 2. In this table, for convenience, we denote the support of the trial function as Ω_{tr}, and the support of the test function as Ω_{te}. *It is noted that in this paper, local sub-domain, Ω_s is the same as the support of the test function, Ω_{te}.* In general, the local sub-domain Ω_s may not be coincident with the support of the test function, Ω_{te}, such as in the case of the MLPG Upwinding scheme (Lin and Atluri, 2000). The interrelationships of these developments can also be illustrated as in Fig. 4. Underlying all these meshless methods is the general concept of the meshless local Petrov-Galerkin method; thus, MLPG provides a rational basis for constructing meshless methods with a greater degree of flexibility. It is noted that in all the MLPG methods in this study, the usual "element assembly" process is not required, to form the global stiffness matrix, unlike in the conventional FEM. Theoretically, as long as the union of all local domains covers the global domain, the equilibrium equation and the boundary conditions will be satisfied in the global domain Ω and its boundary Γ, respectively. However, from our computational experience, the present formulation yields a very satisfactory result even when the union of all local domains does not completely cover the global domain.

In this study, Gaussian quardrature is used to integrate all the weak forms that arise in all the six MLPG methods. A special attention is needed in MLPG3 and MLPG6, because sufficient Gaussian points are needed to guarantee a convergent result. *Once again, we must point out that the PU interpolation is not suitable for MLPG2, MLPG5, MLPG4 and MLPG1.* However, the PU interpolation can be used for MLPG3 and MLPG6, due to the fact that the test function can be taken to be the same as the trial function in MLPG6, and is determined by the least square approach in MLPG3. From the viewpoint that sufficient Gaussian points are needed to guarantee a convergent result in MLPG3 and MLPG6, we do not conduct the numerical analysis for PU interpolation in this paper.

The collocation method (MLPG2) does not involve any numerical integration to generate the global stiffness matrix, thus it is the simplest form of the meshless method. The MLPG4 (LBIE) does not involve the domain integral to generate the stiffness matrix; but does involve a singular integral. The MLPG1, MLPG3 and MLPG6 methods involve domain integrals to generate the stiffness matrix, which is difficult in meshless methods due to complexity of the integrand, especially for the MLPG3 and MLPG6. *The MLPG5 does not involve any domain and singular integrals to generate the global stiffness matrix; it only involves a regular boundary integral. Thus, it is an attractive meshless method.* We are certain that MLPG5 will have a considerable impact on computational mechanics; and may prove to be a simple and efficient alternative to the currently popular finite element and boundary element methods.

It is noted that in MLPG1 MLPG4, and MLPG5, when the size of the support of the test function domain, Ω_s,

Table 2 : Meshless local Petrov-Galerkin (MLPG) methods

Methods	Test function in Ω_{te}	Local weak form over each Ω_s	Relation between Ω_{te} and Ω_{tr}	Integral to evaluate the weak-form
MLPG1	MLS weight function	LSWF	$\Omega_{te} < \Omega_{tr}$	Domain integral
MLPG2	Kronecker Delta $\delta(\mathbf{x},\mathbf{x}_I)$	LUSWF1	Ω_{te} can be arbitrary	None
MLPG3	Least square $\phi^I_{,ii}(\mathbf{x})$	LUSWF1	$\Omega_{te} = \Omega_{tr}$ (for the modified MLPG3: $\Omega_{te} < \Omega_{tr}$)	Domain integral
MLPG4	Fundamental solution u^*	LUSWF2	$\Omega_{te} < \Omega_{tr}$	Singular boundary integral
MLPG5	Heaviside step function	LSWF	$\Omega_{te} < \Omega_{tr}$	Regular boundary integral
MLPG6	Same as the trial function	LSWF	$\Omega_{te} = \Omega_{tr}$	Domain integral

shrinks to zero, the results in these methods approach to those in collocation method. Even in MLPG3, if we still use the test function (49), but shrink its support, the results also approach to those in the collocation method.

6 Essential boundary conditions

For the trial-function-interpolation schemes with the Kronecker Delta properties, such as the RBF, it is trivial to implement the essential boundary conditions. In this study, for the all the methods using RBF, we directly impose the essential boundary conditions.

However, in trial-function interpolation schemes without the Kronecker Delta properties, such as the MLS, Shepard function and PU, the meshless trial-function approximations do not pass through the nodal data, which are the fictitious values at the nodes. It is not easy to implement the essential boundary conditions. In many researches, a Lagrange multiplier technique has been used to impose the essential boundary conditions. However, this technique produces a non-banded and a non-positive definite stiffness matrix. One of the promising methods to enforce the essential boundary conditions in meshless methods is the penalty parameter technique as developed by Zhu and Atluri (1998), which is efficient and does not need any other additional unknown variables. This penalty technique is used in Eq. (42). In this study, in the MLPG1-MLS (Shepard function or PU) and MLPG6-MLS (Shepard function or PU), we use this penalty technique to enforce the essential boundary conditions.

A modified collocation method is also a very important technique to enforce the essential boundary conditions in the meshless method (Zhu and Atluri, 1998) due to its simplicity. In fact, in this study, in MLPG2 and MLPG3, we use this collocation technique to enforce the essential boundary conditions. Our numerical results show that both the penalty parameter and collocation techniques are effective and convenient.

Another technique to exactly impose the essential boundary conditions in meshless methods is developed by Atluri, Kim & Cho (1999). However, it is not employed in the present study.

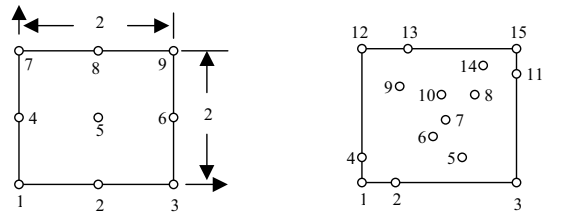
7 Numerical experiments

In this section, some numerical results are presented, in order to compare the efficiency and accuracy of the present MLPG methods, for a variety of meshless trial and test functions. For the purpose of error estimation and convergence studies, the Sobolev norms $\|\cdot\|_k$, k=0,1 are calculated. These norms are defined as:

$$\|u\|_0 = \left(\int_\Omega u^2 d\Omega \right)^{\frac{1}{2}} \tag{68a}$$

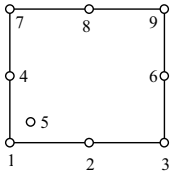
and

$$\|u\|_1 = \left(\int_\Omega u^2 + (|\nabla u|)^2 d\Omega \right)^{\frac{1}{2}} \tag{68b}$$



a Regular nodes

b Irregular nodes



c Irregular node 5

Coordinates for patch (b)

Nodes	
2	0.3,0.0
4	0.0,0.2
5	1.2,0.4
6	0.8,0.9
7	0.9,1.0
8	1.4,1.2
9	0.5,1.3
10	1.1,1.2
11	2.0,1.8
13	0.8,2.0
14	1.6,1.9

Figure 5 : Nodes for the patch test

The relative errors are defined as

$$r_k = \frac{\left\| u^{\text{num}} - u^{\text{exact}} \right\|_k}{\left\| u^{\text{exact}} \right\|_k}, \quad k = 0,1 \tag{69}$$

7.1 Patch test

Consider the standard patch test in a domain of dimension 2×2 as shown in Fig. 5, and for the Dirichlet problem for the Laplace equation, i.e. with p=0, in equation (1). We consider a problem with the exact solution

$$u = x^1 + x^2 \tag{70}$$

where the potential boundary condition Eq. (2a) is prescribed on all boundaries according to Eq. (70). Satisfaction of the patch test requires that the value of u at any interior node be given by the same linear function (70); and that the derivatives of the computed solution be constant in the patch.

In this example, 9 Gauss points are used on each local boundary L_s (a circle for internal nodes and a part of a circle for boundary nodes in this paper), and 6 points are used on each boundary section Γ_s for numerical quadratures in the MLPG4 and MLPG5 methods.

In the MLPG1 method, 6 points are used on each boundary section Γ_s, and 5×8 points are used in the local domain Ω_s for numerical quadratures.

In the MLPG3 method, 15 points are used on each boundary section Γ, and 15×5 points are used in the local domain Ω for numerical quadratures.

The nodal arrangements of all patches are shown in Fig. 5. In the computation, the radius of the local sub-domain r_i=4 are used. In Fig. 5, the coordinates of node 5 for the meshes c1, c2, c3, c4, c5 and c6 are (1.1,1.1), (0.1,0.1), (0.1,1.8), (1.9,1.8), (0.9, 0.9) and (0.3, 0.4) respectively.

The computational results show that all the present MLPG methods, with various test-functions, pass all the patch tests as shown in Fig. 5, for the alternative trial-function interpolations MLS, Shepard functions, and the radial basis functions, respectively, with the given source function p=0.

7.2 Laplace equation

The second example solved here is the Laplace equation in the 2×2 domain shown in Fig. 5, with the exact solution taken to be a cubic polynomial, as:

$$u = -\left(x^1\right)^3 - \left(x^2\right)^3 + 3\left(x^1\right)^2 x^2 + 3x^1\left(x^2\right)^2 \tag{71}$$

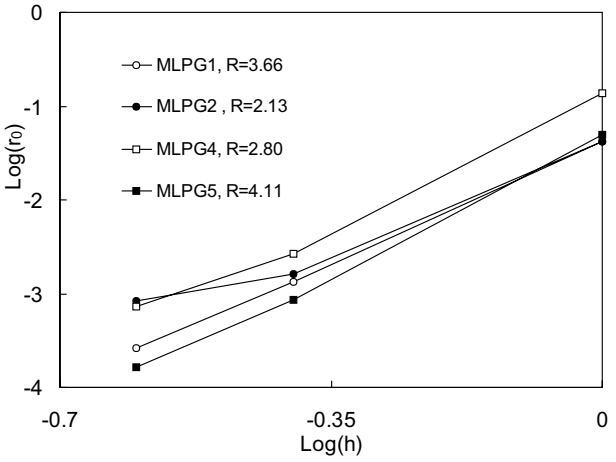
A Dirichlet problem is solved, for which the essential boundary condition is imposed on all sides, and a mixed problem, for which the essential boundary condition is imposed on top and bottom sides and the flux boundary condition is prescribed on left and right sides of the domain.

Regular meshes of 9(3×3), 36(6×6) and 81(9×9) nodes are used to study the convergence of the method. 9 Gauss points are used on each local boundary L_s (a circle for internal nodes and a part of a circle for boundary nodes in this paper), and 6 points are used on each boundary section Γ_s for numerical quadratures in the MLPG4 and MLPG5 methods.

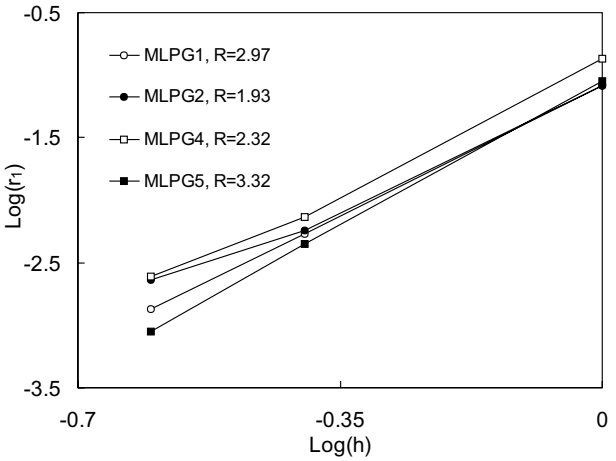
In the MLPG1 method, 6 points are used on each boundary section Γ_s, and 5×8 points are used in the local domain Ω_s for numerical quadratures.

7.2.1 Moving least –square interpolation (MLS)

In this calculation, the trial function is from the MLS, in which the 4th-order spline weight function (22) is used. The size of support for the trial functions is taken to be 5h, with h being distance between nodes; and the size



(a)

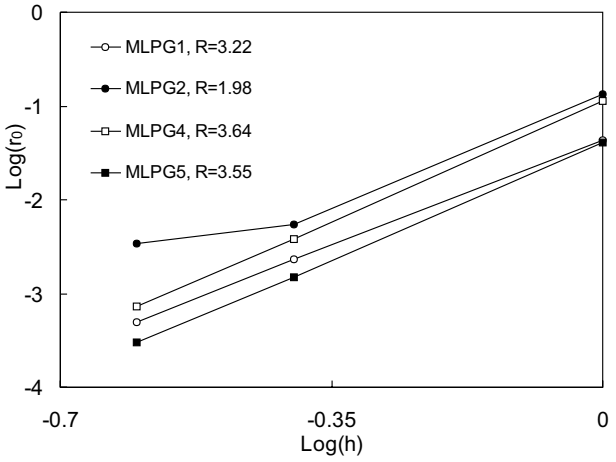


(b)

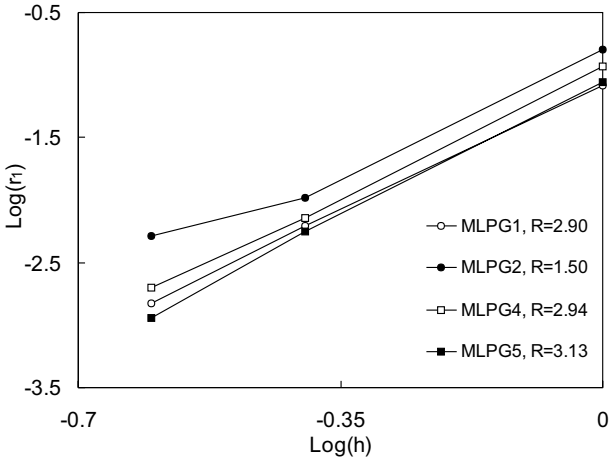
Figure 6 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation for MLS: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$. In this figure and thereafter, *R* is the convergence rate.

(radius) of the local boundary (i.e., support of the test function, Ω) for each node is chosen as $0.9h$ in all the computations, except for the MLPG3, when it is the same as the size of support for trial functions.

The convergence, with the nodal-spacing refinement, of the present methods: MLPG1, MLPG2, MLPG3, MLPG4 and MLPG5 is studied for this problem. For MLPG1, MLPG2, MLPG4 and MLPG5, the results of the relative errors, and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ are shown in Fig. 6 for the Dirichlet problem, and in Fig. 7 for the mixed problem, respectively, where R is the rate of the convergence.

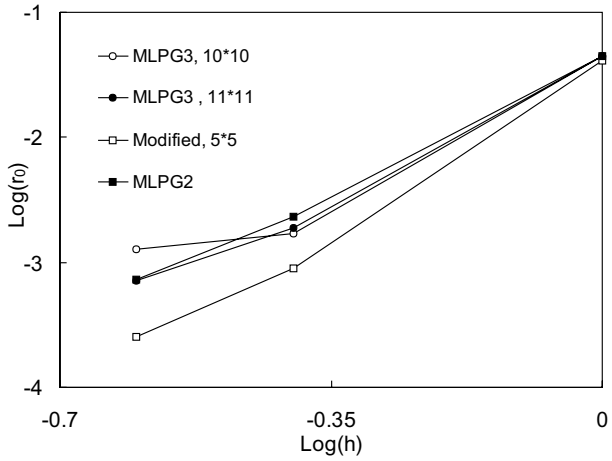


(a)

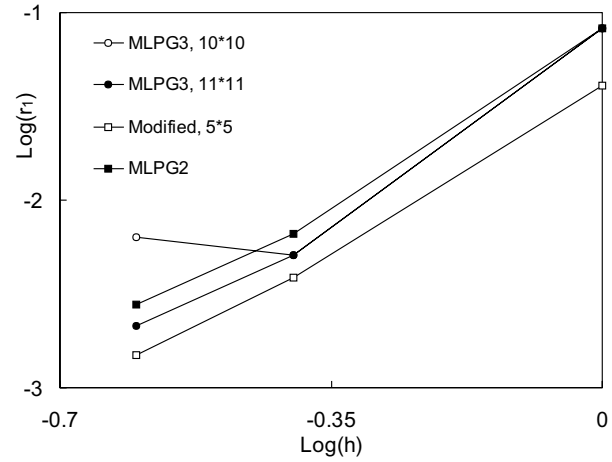


(b)

Figure 7 : Relative errors and convergence rates for the mixed problem of Laplace equation for MLS: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.



(a)



(b)

Figure 8 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation with different Gaussian points for MLS: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

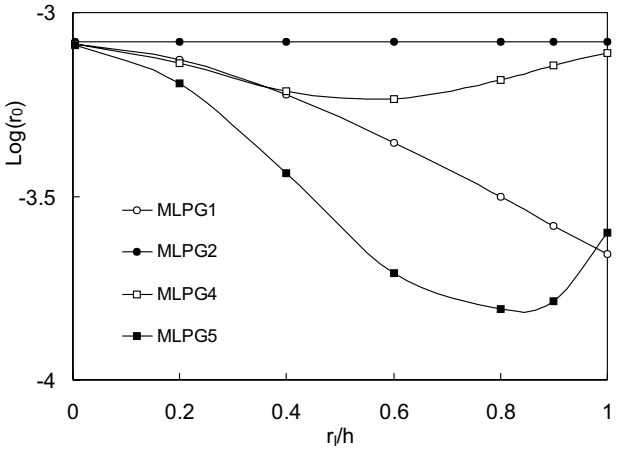
It can be seen that the present MLPG methods have high rates of convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ and give reasonably accurate results for the unknown variable and its derivatives. Among them, MLPG5 appears to be the best (highest rate of convergence).

In MLPG3, the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ are shown in Fig. 8 for the Dirichlet problem, where the term 'modified' means that the equation (49) is still used as the test function, but the size of its support is changed to h (note that the support of the trial function, chosen in this example, is $5h$). It is observed from these figures that a monotonic convergence cannot be obtained if a sufficient numbers of Gaussian points is not used. Particularly, the numerical integration deteriorates as the nodal distance decreases, if the numerical integration is not accurate enough to evaluate the stiffness matrix. In other words, the more nodes are added in the domain, the more integration points are required, in order to achieve the accuracy corresponding to a smaller nodal distance. This kind of difficulty is caused by the complexity of the shape functions in MLPG3, as discussed in Section 5.3 and in Atluri, Kim & Cho (1999). *In MLPG3, the Gauss points must be at least 11×11, almost 5 times that of the 'modified' MLPG3*. Moreover, the accuracy is lower than the corresponding 'modified' MLPG3, or even MLPG2. These results also imply that the size of the support of the test function cannot be too large; a good choice for the size of the support being less than h.

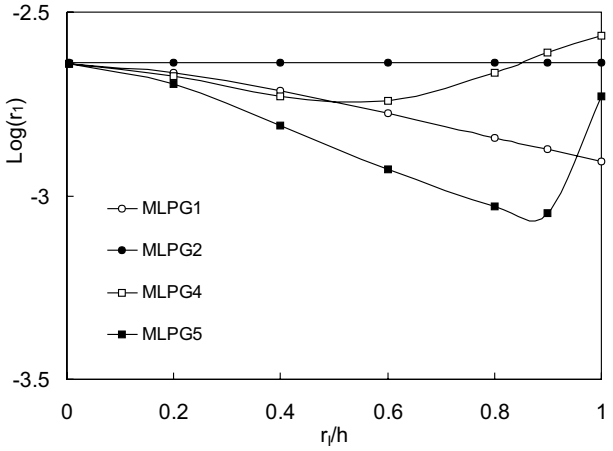
The effect of the size of sub-domain (the support of the test function) is shown in Fig. 9. Here, the size of the support of the trial function is $5h$, and $81(9\times9)$ nodal points are used. It is found that MLPG1, MLPG4 and MLPG5 approach to the MLPG2 as the size of the support of the test function approaches to zero. In almost all of the intervals, MLPG5 is of the highest accuracy. The size of the support of the trial function will affect the trend.

The effect of the size of the support of the trial function is shown (ρ; ρh ρ is changed from 2 to 10; h is the nodal spacing) in Fig. 10. Here, the size of the support of the test function is h, and $81(9\times9)$ nodal points are used. The trend is very complex for MLPG1, MLPG2, MLPG4 and MLPG5. The size of the support of the test function affects this trend.

7.2.2 Shepard function



(a)



(b)

Figure 9 : Relative errors for the Dirichlet problem of Laplace equation vs the size of the support of the test function for MLS: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

In this calculation, the trial function approximation is from the Shepard function. The size of the support for the trial functions is taken to be 6*h* [with *h* being the distance between nodes], and the radius of the local test function domain for each node is chosen as 0.9*h* in the computation; but for the MLPG3, it is the same as the size of support for trial functions.

For MLPG1, MLPG2, MLPG4 and MLPG5, the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ are shown in Fig. 11 for the Dirichlet problem and in Fig. 12 for the mixed problem, respectively. The convergence rate is lower than that in MLS. Once again, the MLPG5 method has the best rate of convergence.

In MLPG3, the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ are shown in Fig. 13 for the Dirichlet problem. *It is indicated that we still need 11×11 Gaussian points to ensure the convergence*. The results for the corresponding 'modified' MLPG3, and MLPG2 are also shown in Fig. 13 for comparison.

The effect of the size of sub-domain (the support of the test function) is shown in Fig. 14. Here, the size of the support of the trial function is 6*h*, and 81(9×9) nodal points are used. The trend is same as that in MLS.

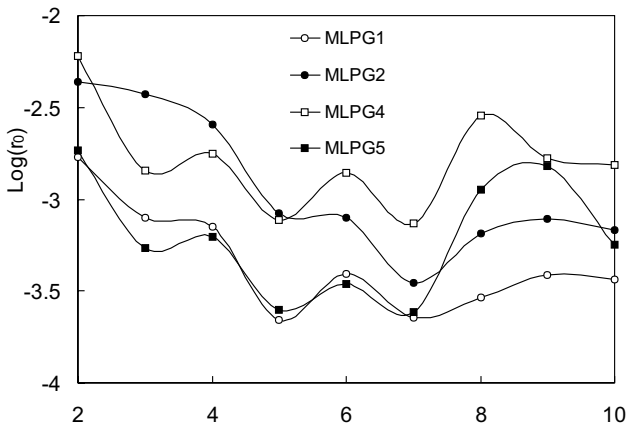
The effect of the size of the support of the trial function is shown in Fig. 15. Here, the size of the support of the test function is *h*, and 81(9×9) nodal points are used. The trend is less complicated than that in MLS.

7.2.3 Radial basis function

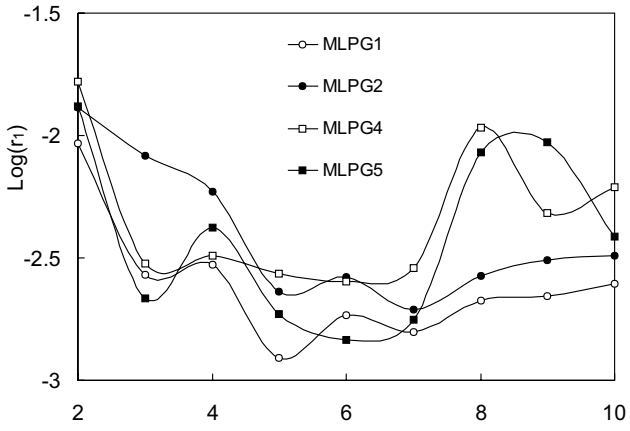
In this calculation, the trial function approximation is from the radial basis function (RBF). Two kinds of radial basis functions, RBF1 Eq. (29) and RBF2 Eq. (30) are considered. The size of support for the trial functions are taken to be 2.5, and the size (radius) of the local domain (the test function support) for each node is chosen as 0.9*h*, with *h* being the distance between nodes. For the MLPG3, the size of the nodal test function domain is same as the size of support for trial functions.

For RBF1, the results of the relative errors, and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ for MLPG1, MLPG2, MLPG4 and MLPG5, are shown in Fig. 16 for the Dirichlet problem, and in Fig. 17 for the mixed problem, respectively. For RBF2, similar results are shown in Fig. 18 and 19, respectively. The convergence rate for RBF1 is lower than that for RBF2.

In MLPG3, for the Dirichlet problem, the results of the

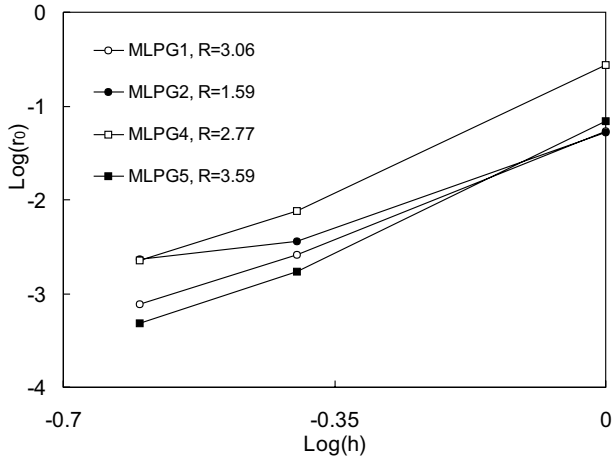


(a)

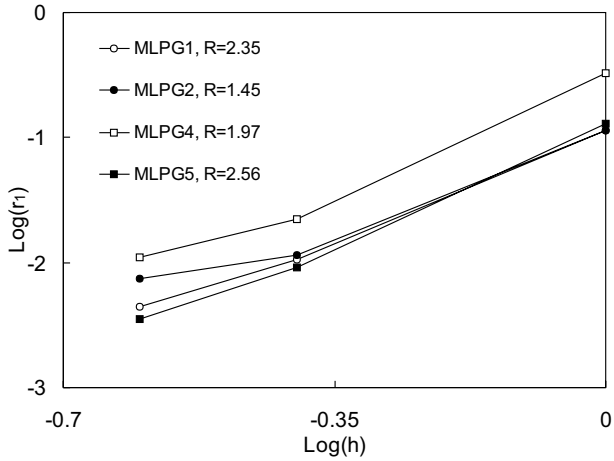


(b)

Figure 10 : Relative errors for the Dirichlet problem of Laplace equation vs the size of the support of the trial function for MLS: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$

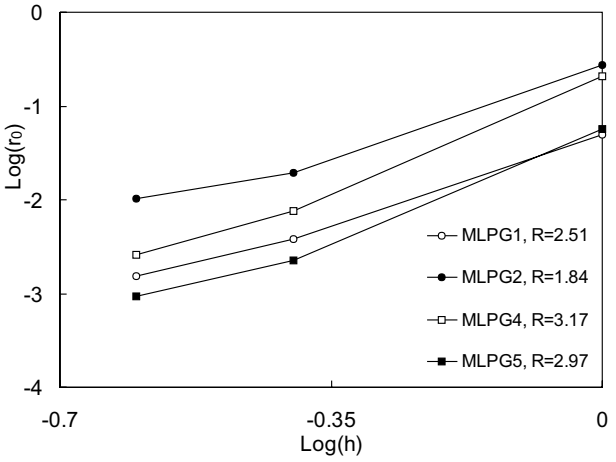


(a)

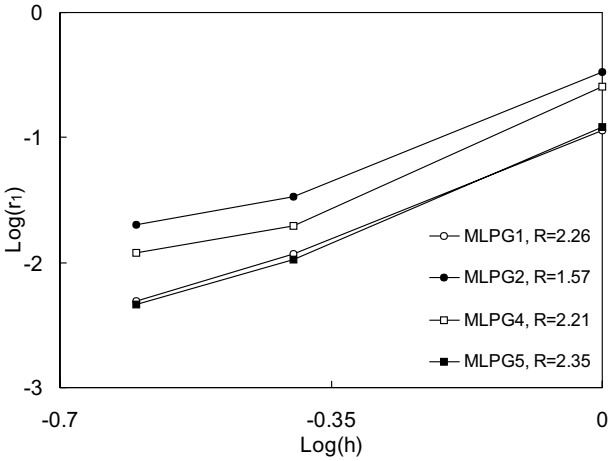


(b)

Figure 11 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation for Shepard: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

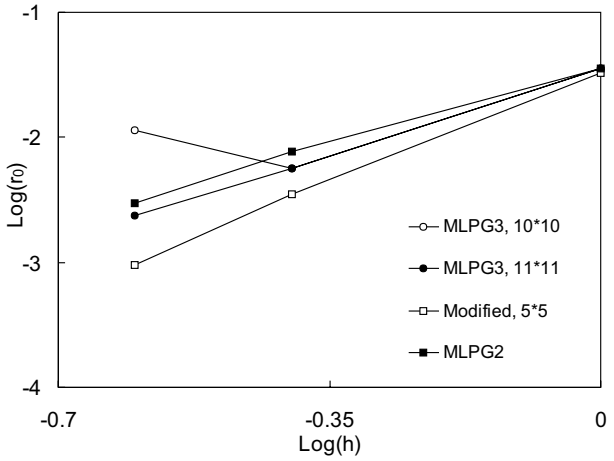


(a)

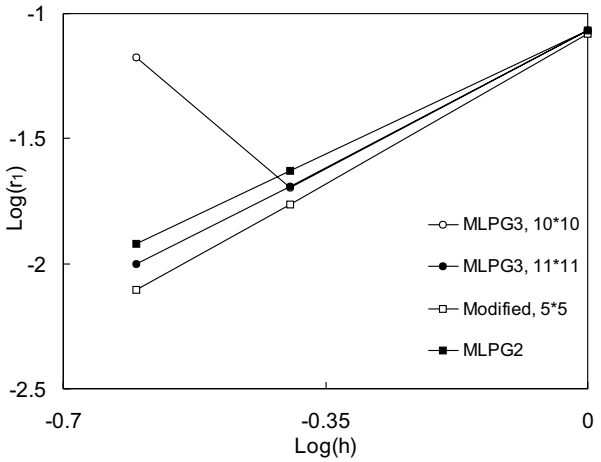


(b)

Figure 12 : Relative errors and convergence rates for the mixed problem of Laplace equation for Shepard: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

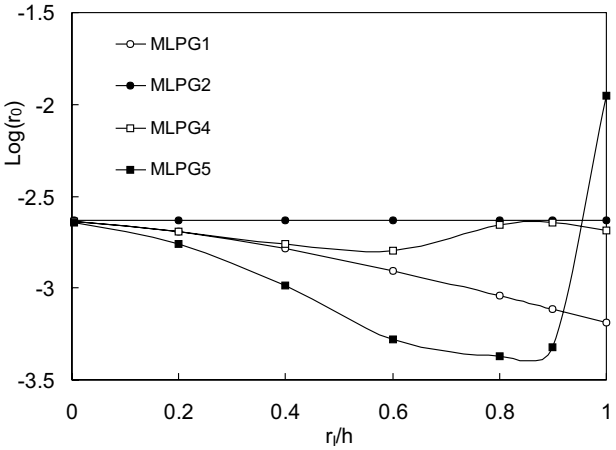


(a)

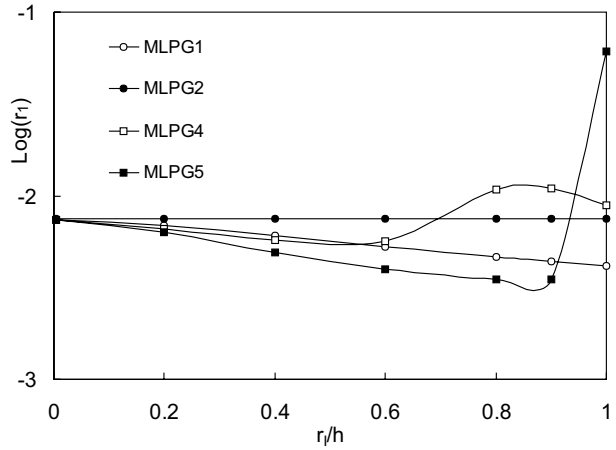


(b)

Figure 13 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation with different Gaussian points for Shepard: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

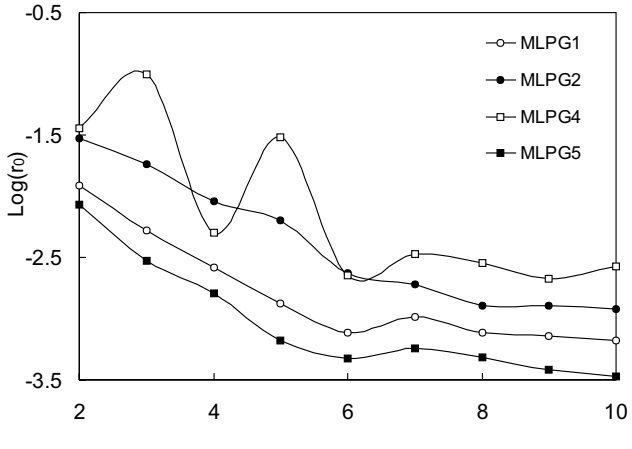


(a)

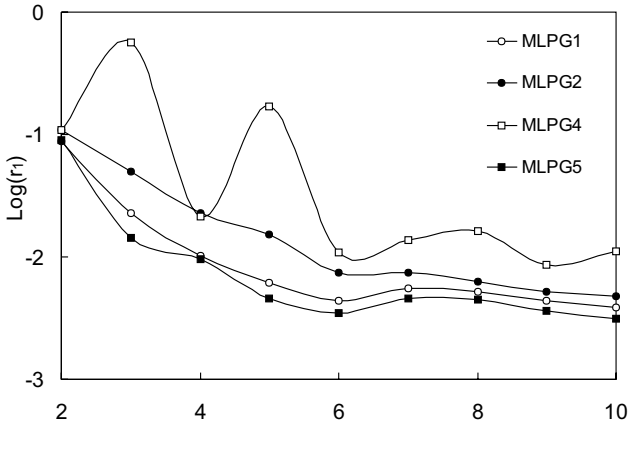


(b)

Figure 14 : Relative errors for the Dirichlet problem of Laplace equation vs the size of the support of the test function for Shepard: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.



(a)



(b)

Figure 15 : Relative errors for the Dirichlet problem of Laplace equation vs the size of the support of the trial function for Shepard: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ are shown in Fig. 20 for RBF1 and Fig. 21 for RBF2, respectively. *It is noted that we still need 11×11 Gaussian points to ensure the convergence for both RBF1 and RBF2*. The results for the corresponding 'modified' MLPG3 (with $\Omega_s \equiv \Omega_{te}$ being of radius h, and Ω_{tr} being of radius 2.5) and MLPG2 are also shown in these figures for comparison.

The effect of the size of the sub-domain (the support of the test function) is shown in Fig. 22. Here, the size of the support of the trial function is 2.5, and 81(9×9) nodal points are used. The trend is more complicated than that in the MLS and the Shepard function approximations.

The effect of the size of the support of the trial function is shown in Fig. 23. Here, the size of the support of the test function is h, and 81(9×9) nodal points are used. The trend is simpler than that in MLS. For MLPG1, MLPG2, MLPG4 and MLPG5, the trend is almost the same.

To compare the efficiency and accuracy of the three trial functions, we also plotted the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ for the Dirichlet problem, as shown in Fig. 24-28 for MLPG1, MLPG2, MLPG4, MLPG5 and MLPG3, respectively. In these figures, the size of support for the trial functions is taken to be 2.5, and the size (radius) of the local boundary for each node is chosen as $0.9h$. For MLPG3, it is same as the size of support for trial functions.

From these figures, we can find that for MLPG1, MLPG2, MLPG4 and MLPG5, the MLS trial-function approximation has the highest accuracy and convergence rate; the second in accuracy is the Shepard function, and lastly the RBF1; while in MLPG3, RBF2 type trial-function approximation has the highest accuracy, the followed by the MLS, and then the RBF1. It is also noted that the size of the support of the trial function in RBF must be large; otherwise we cannot get a high accuracy.

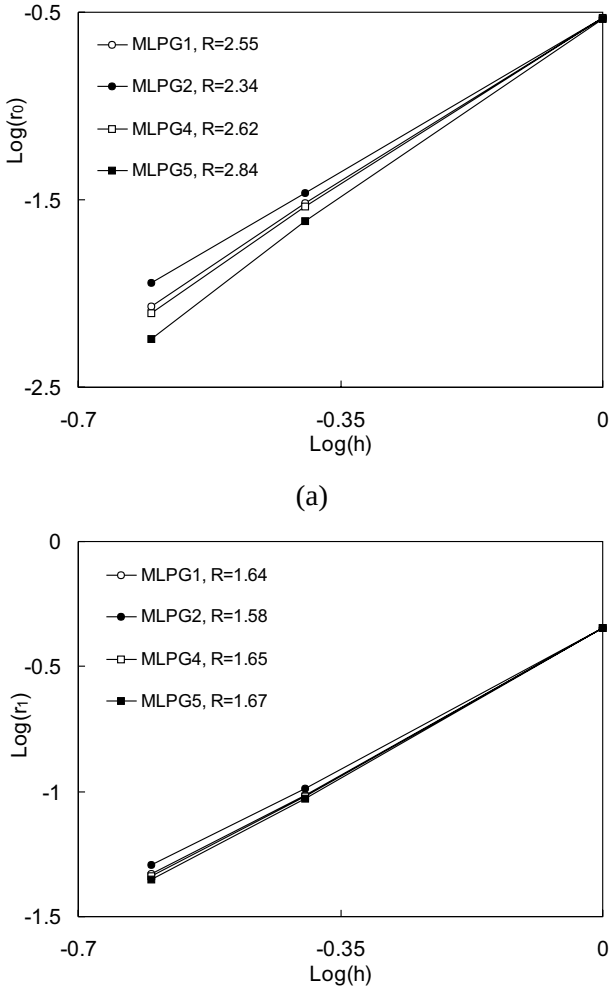
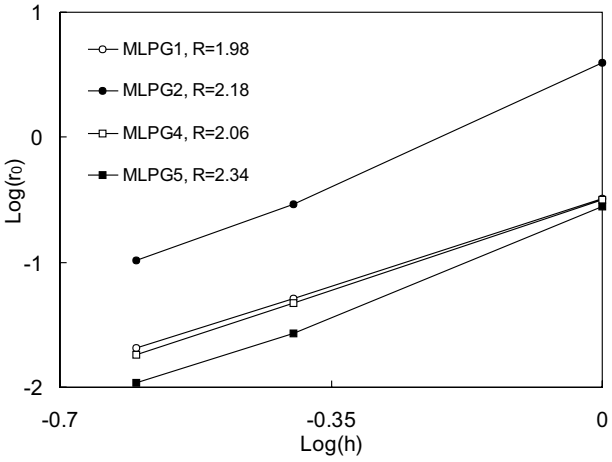


Figure 16 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation for RBF1: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

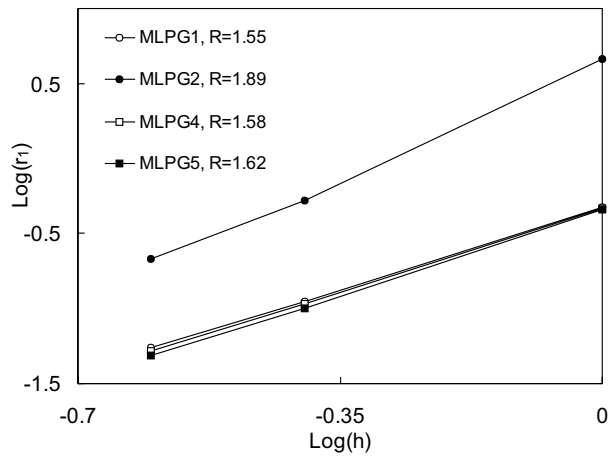
7.3 Poisson's equation

The results from the MLPG1, MLPG2, MLPG4 and MLPG5 methods are also studied for the Poisson's equation, with the source function $p=x^1+x^2$, in a 2×2 domain, for which the exact solution is taken to be

$$u = -\frac{5}{6}\left[\left(x^1\right)^3 + \left(x^2\right)^3\right] + 3\left(x^1\right)^2 x^2 + 3x^1\left(x^2\right)^2 \qquad (72)$$

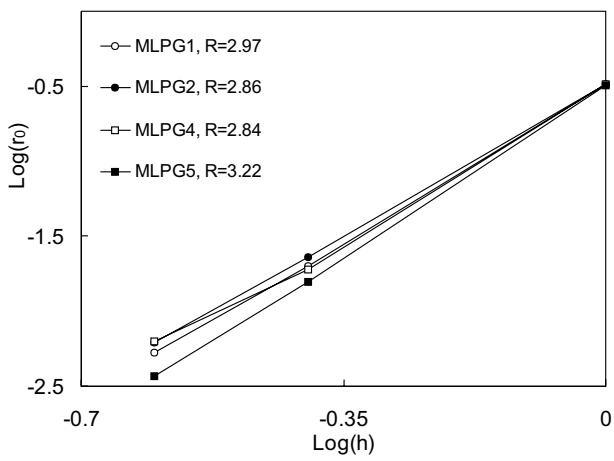


(a)

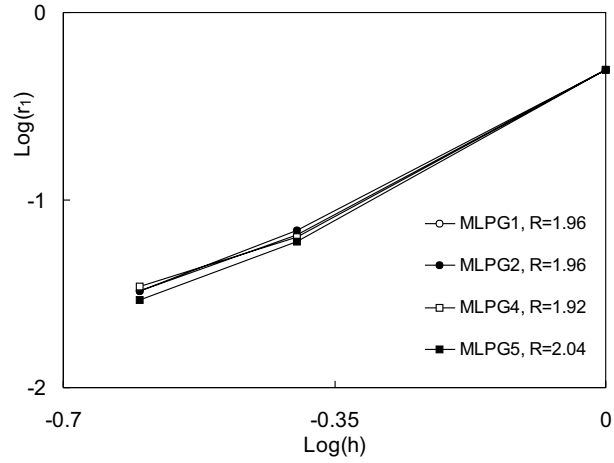


(b)

Figure 17 : Relative errors and convergence rates for the mixed problem of Laplace equation for RBF1: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

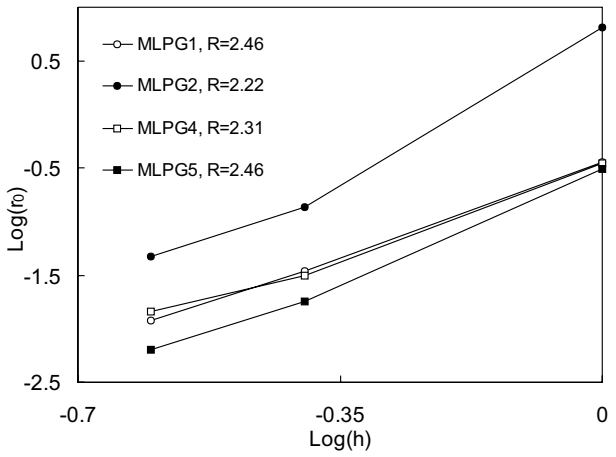


(a)

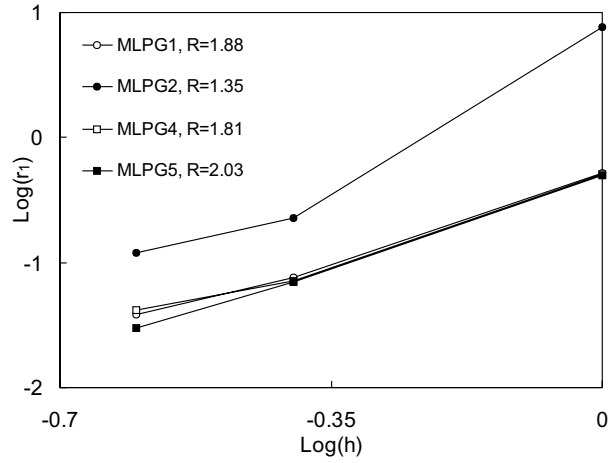


(b)

Figure 18 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation for RBF2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

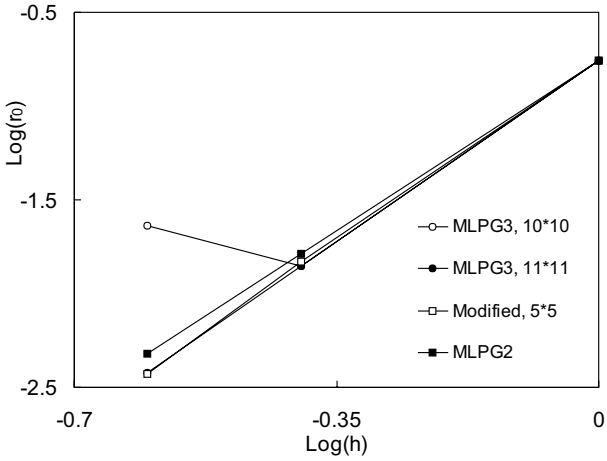


(a)

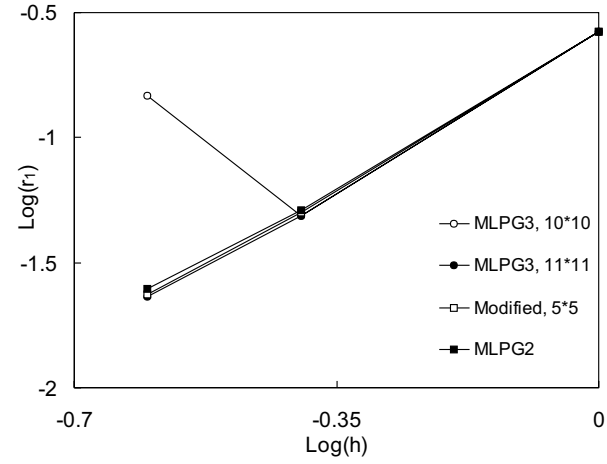


(b)

Figure 19 : Relative errors and convergence rates for the mixed problem of Laplace equation for RBF2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

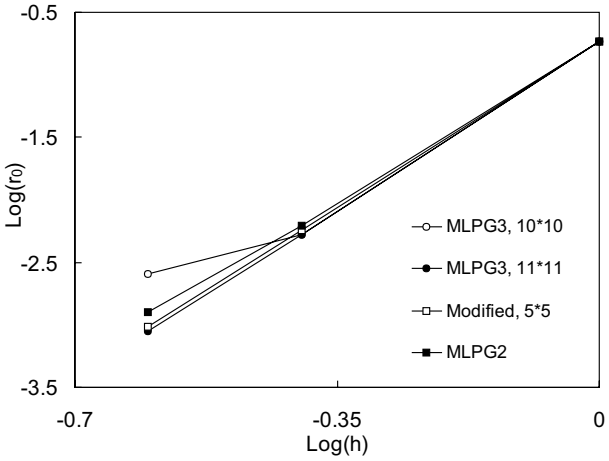


(a)

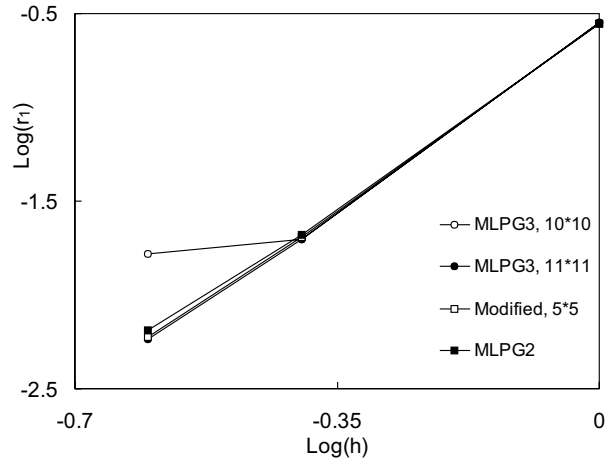


(b)

Figure 20 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation with different Gaussian points for RBF1: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

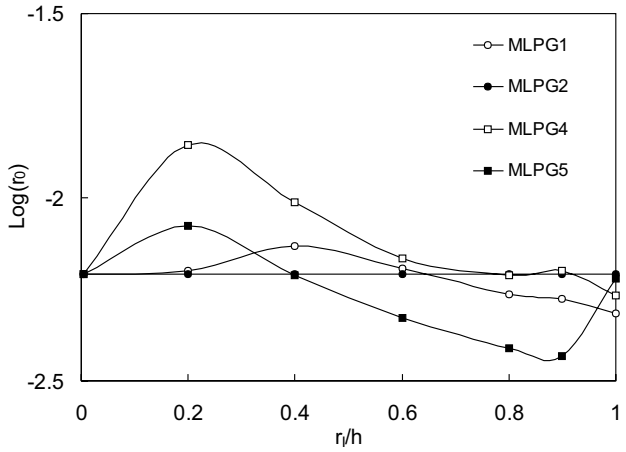


(a)

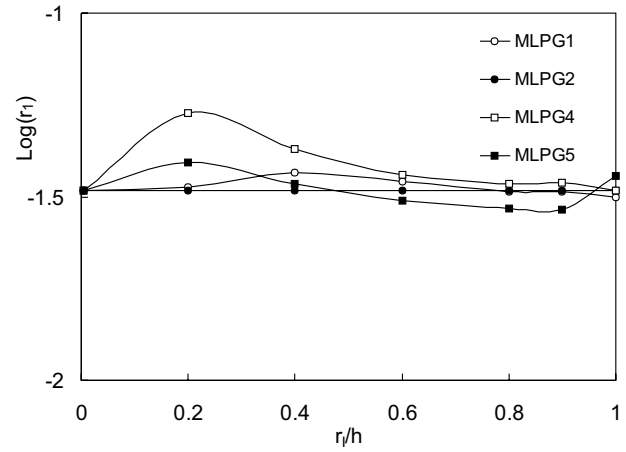


(b)

Figure 21 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation with different Gaussian points for RBF2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

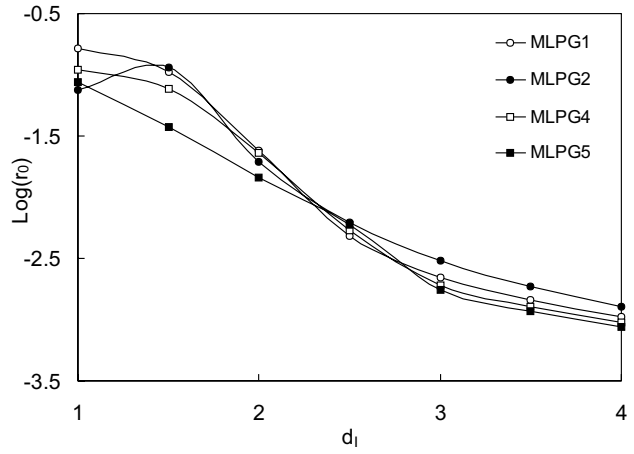


(a)

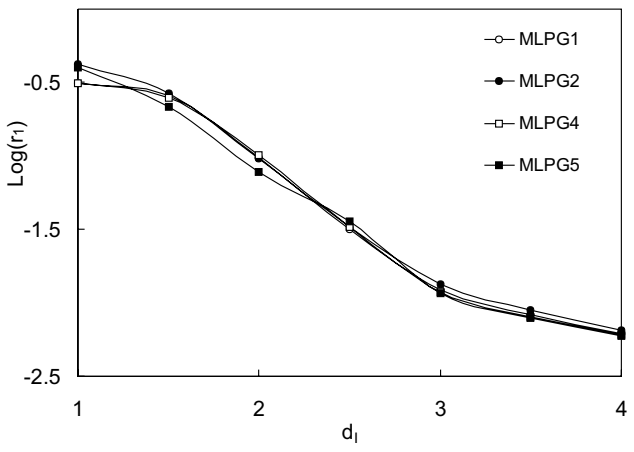


(b)

Figure 22 : Relative errors for the Dirichlet problem of Laplace equation vs the size of the support of the test function for RBF2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

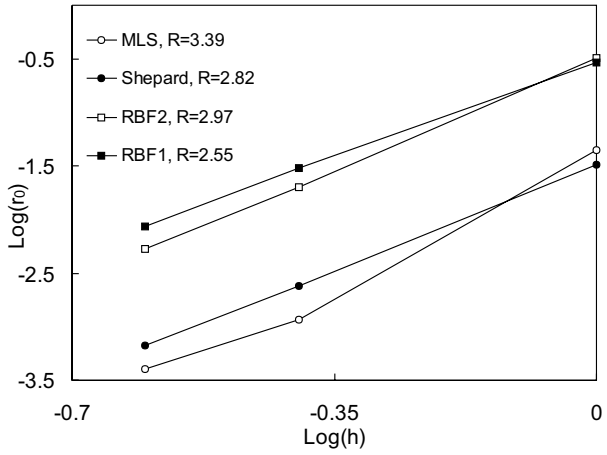


(a)

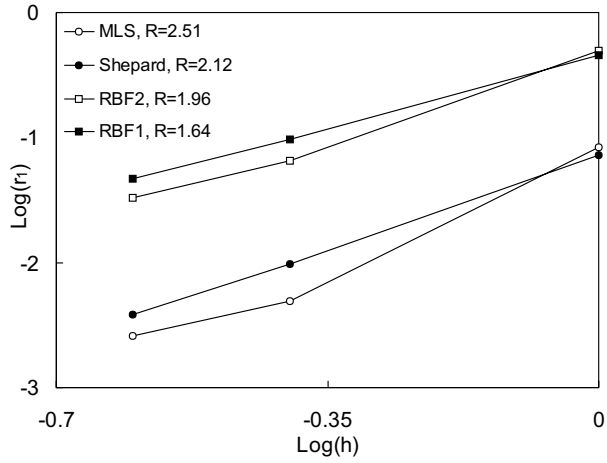


(b)

Figure 23 : Relative errors for the Dirichlet problem of Laplace equation vs the size of the support of the trial function for RBF2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

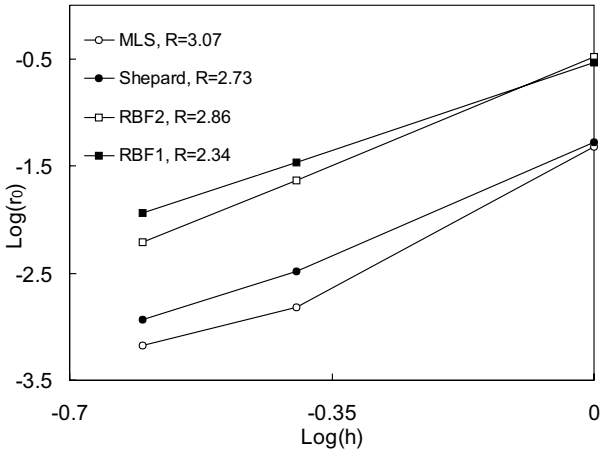


(a)

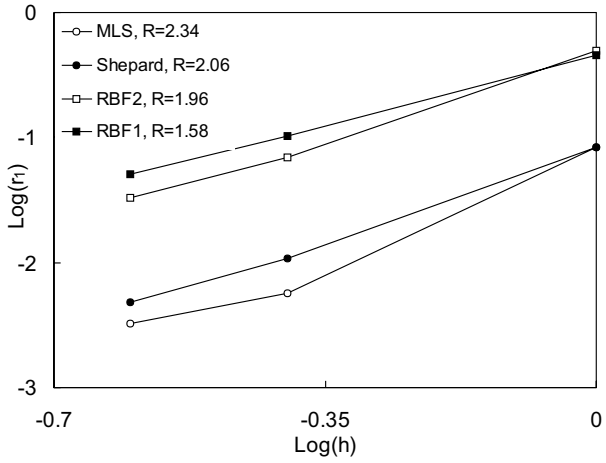


(b)

Figure 24 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation in MLPG1: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

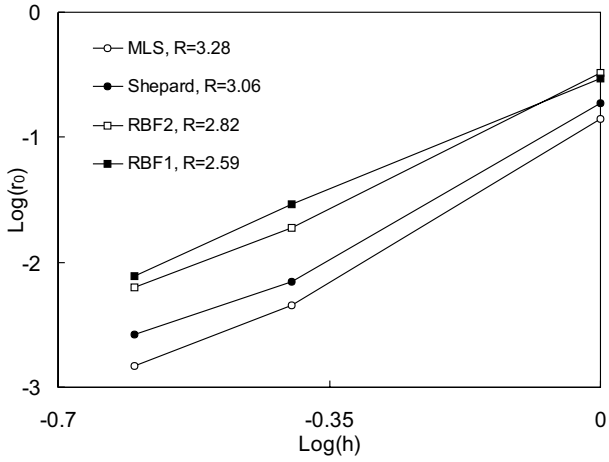


(a)

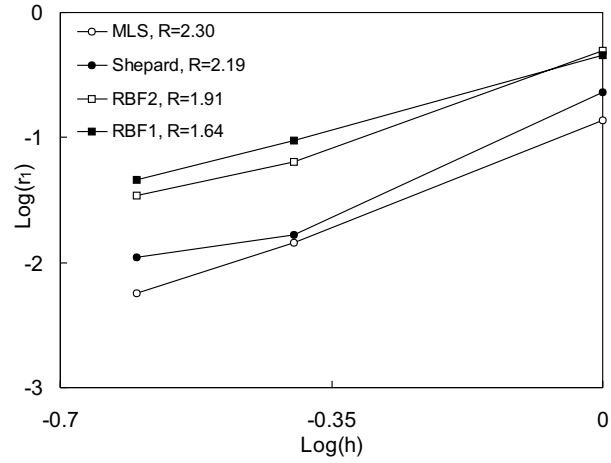


(b)

Figure 25 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation in MLPG2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

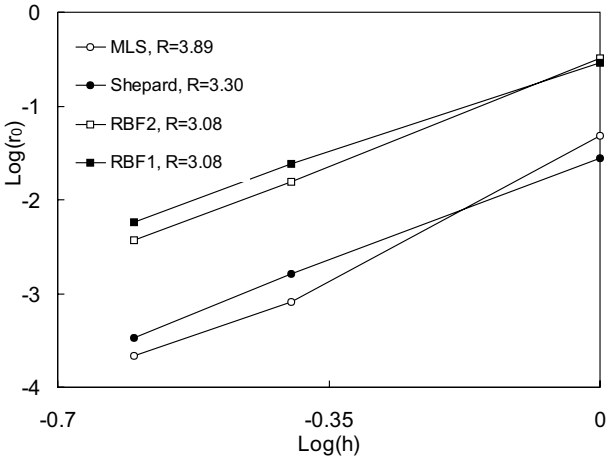


(a)

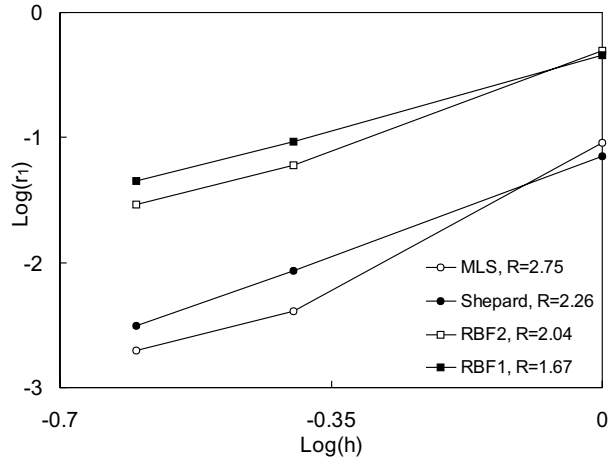


(b)

Figure 26 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation in MLPG4 (LBIE): **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

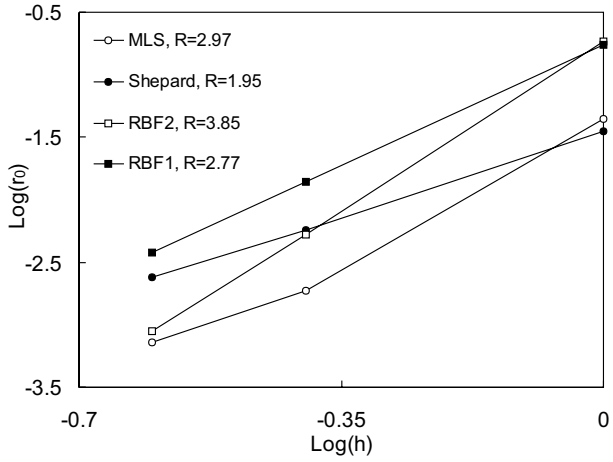


(a)

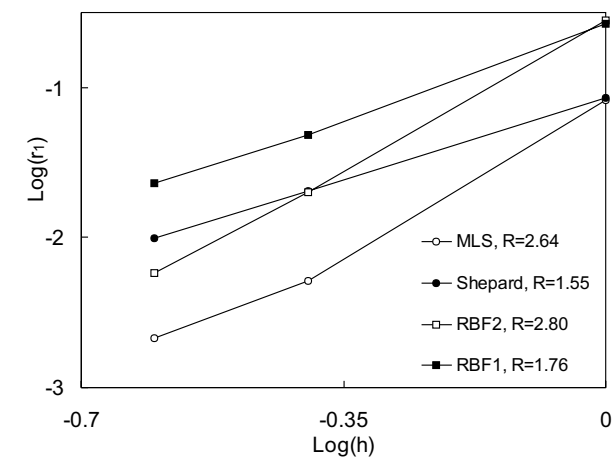


(b)

Figure 27 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation in MLPG5: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.



(a)



(b)

Figure 28 : Relative errors and convergence rates for the Dirichlet problem of Laplace equation in MLPG3: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

The boundary conditions, the size of the local boundary, the size of the support of the trial functions as well as the nodal arrangement are the same as those used in the last example, for every method.

The convergence, with nodal-spacing refinement, of the present methods: MLPG1, MLPG2, MLPG4 and MLPG5 are studied for this problem.

For the trial-function approximation using the MLS, the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ for MLPG1, MLPG2, MLPG4 and MLPG5 are shown in Fig. 29 for the Dirichlet problem, and in Fig. 30 for the mixed problem, respectively.

For the trial-function approximation using the Shepard function, the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ for MLPG1, MLPG2, MLPG4 and MLPG5 are shown in Fig. 31 for the Dirichlet problem, and in Fig. 32 for the mixed problem, respectively.

For the trial-function approximation using the RBF1, the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ for MLPG1, MLPG2, MLPG4 and MLPG5 are shown in Fig. 33 for the Dirichlet problem, and in Fig. 34 for the mixed problem, respectively.

For the trial-function approximation using the RBF2, the results of the relative errors and the convergence for norms $\|\cdot\|_0$ and $\|\cdot\|_1$ for MLPG1, MLPG2, MLPG4 and MLPG5 are shown in Fig. 35 for the Dirichlet problem, and in Fig. 36 for the mixed problem, respectively.

These results show that all the MLPG methods studied in this paper work quite well for the Poisson's equation. *Among them, MLPG5 yields somewhat of a better result than the others, while all the methods possess high accuracy. For the trial-function interpolations, it seems that the MLS yields somewhat of a better result than the Shepard function or the RBF*.

The rates of convergence in all the MLPG methods, especially using the MLS and Shepard functions as the trial function, are higher than that in the FEM.

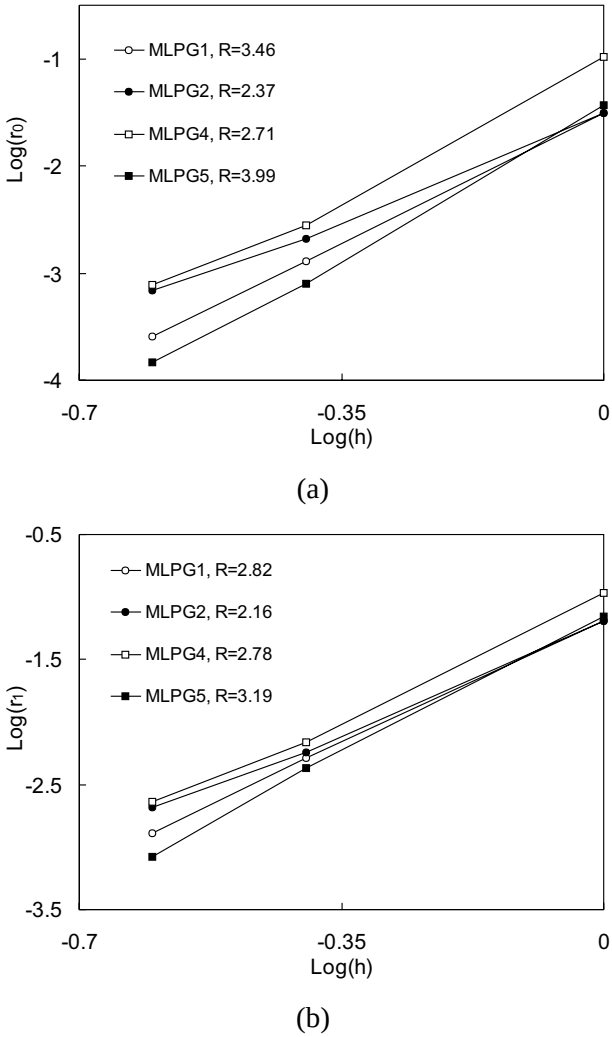
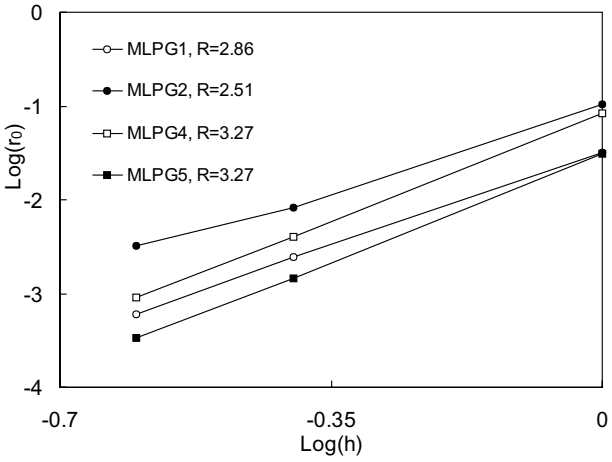


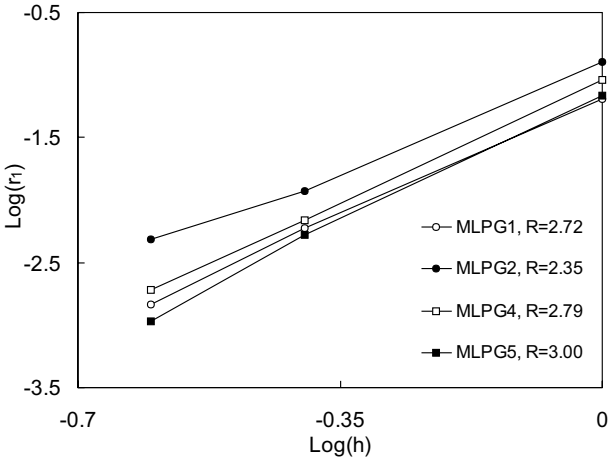
Figure 29 : Relative errors and convergence rates for the Dirichlet problem of Poinson's equation for MLS: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

8 Computational costs

The major factors influencing the success of a methodology are the (human as well as computer) costs versus accuracy considerations. In the MLPG methods, there

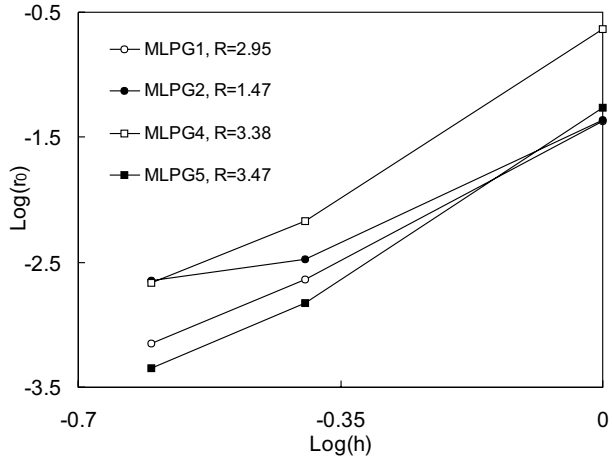


(a)

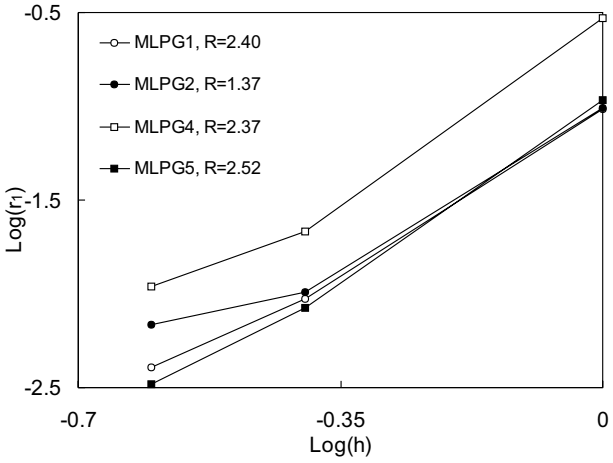


(b)

Figure 30 : Relative errors and convergence rates for the mixed problem of Poinson's equation for MLS: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

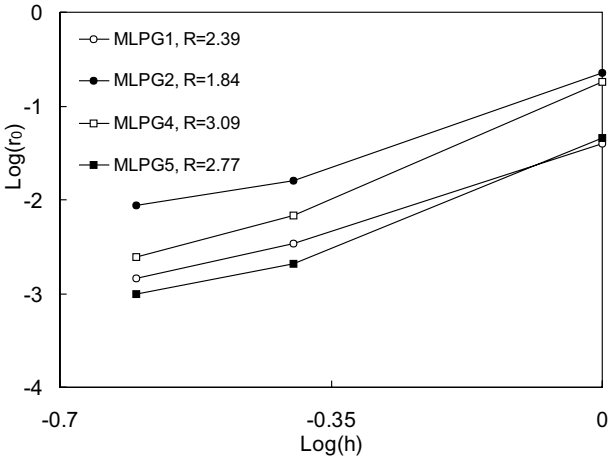


(a)

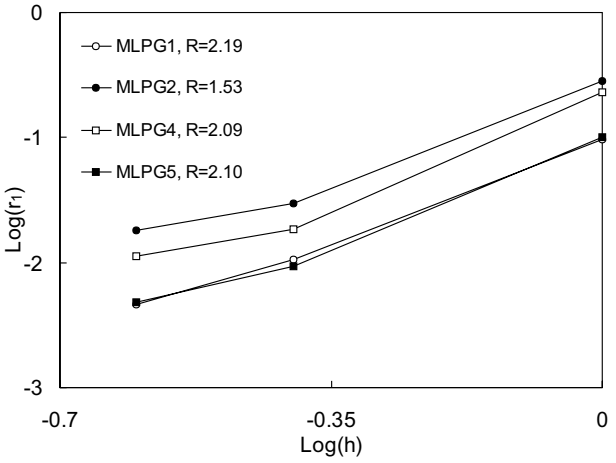


(b)

Figure 31 : Relative errors and convergence rates for the Dirichlet problem of Poinson's equation for Shepard: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

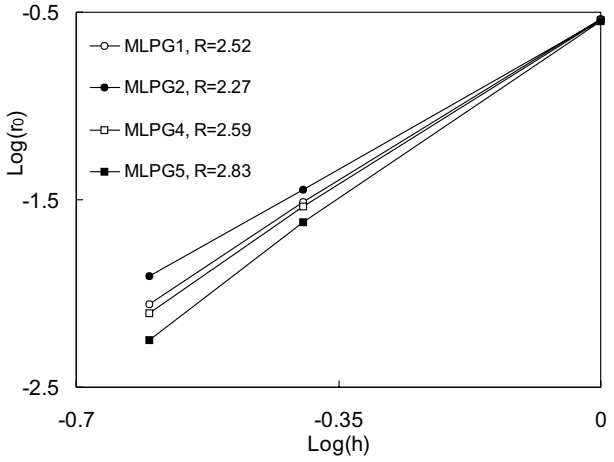


(a)

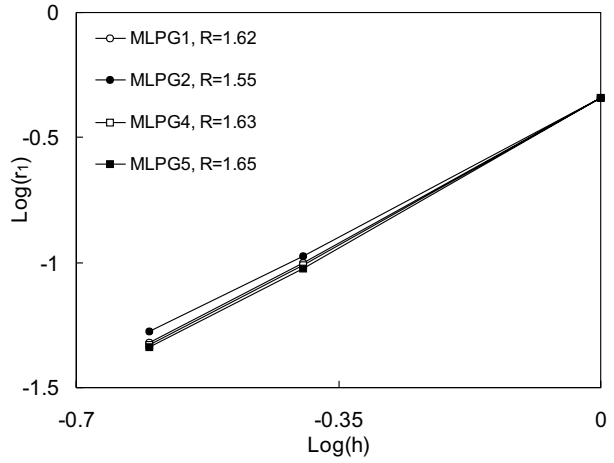


(b)

Figure 32 : Relative errors and convergence rates for the mixed problem of Poinson's equation for Shepard: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

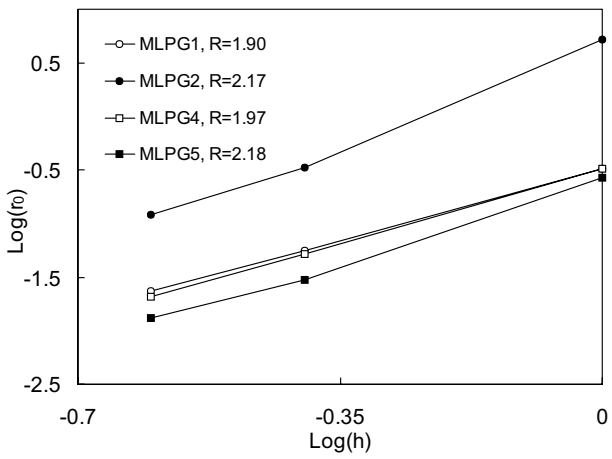


(a)

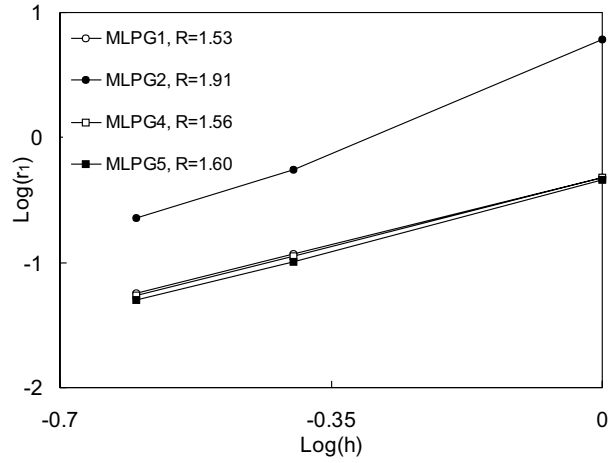


(b)

Figure 33 : Relative errors and convergence rates for the Dirichlet problem of Poinson's equation for RBF1: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

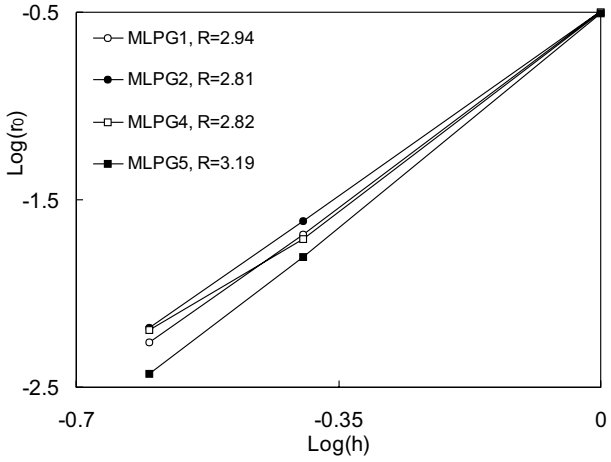


(a)

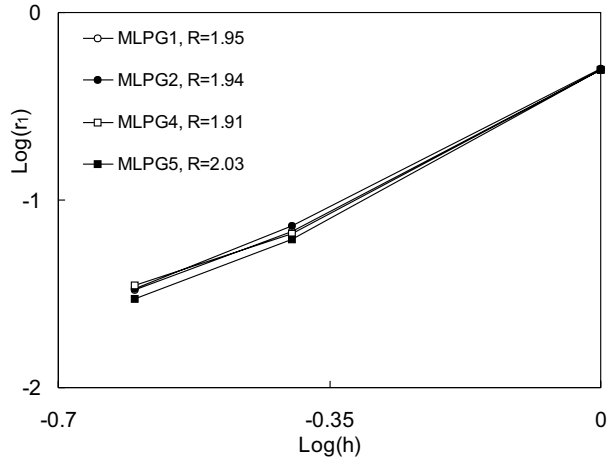


(b)

Figure 34 : Relative errors and convergence rates for the mixed problem of Poinson's equation for RBF1: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

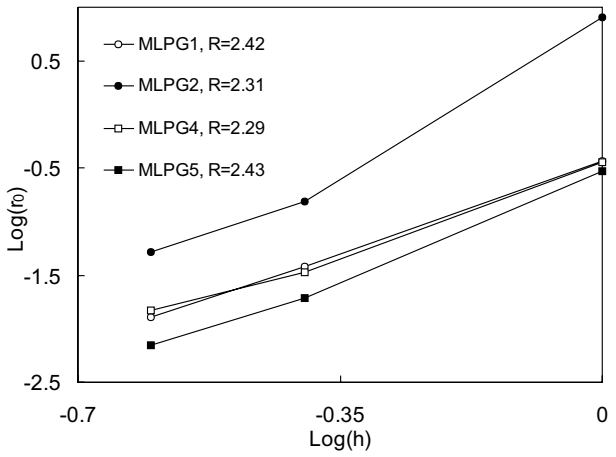


(a)

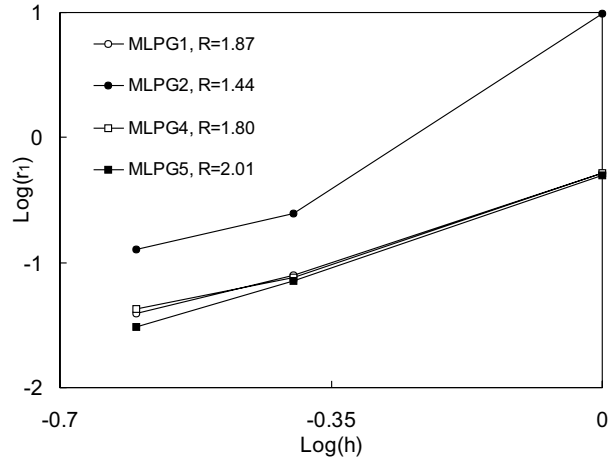


(b)

Figure 35 : Relative errors and convergence rates for the Dirichlet problem of Poinson's equation for RBF2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.



(a)



(b)

Figure 36 : Relative errors and convergence rates for the mixed problem of Poinson's equation for RBF2: **a** for norm $\|\cdot\|_0$, **b** for norm $\|\cdot\|_1$.

are selectable parameters, such as the size of the support of test and trial functions, which affect both the cost and accuracy; however, the effect of the size of the nodal trial-function domain is unclear (see Fig. 10). In this section, we will qualitatively (or roughly) estimate the computational costs in the present MLPG methods, and compare these with that of the classical FEM. We only consider the major *computational costs*: cost of generating the global stiffness matrix and that of solving the resulting algebraic equations. We do not consider the *preprocessing human-labor cost in generating a mesh*, and the postprocessing time, which, of course, are *the main features of attraction and advantage* of the meshless MLPG methods over the FEM. The high order of continuity which MLS inherits from an appropriate choice of the weight function, provides solutions with smooth derivatives, a feature which is very different to the FEM. Hence, as mentioned in Section 4, postprocessing in MLPG is relatively straightforward, and no additional stress smoothing is required.

The estimation here is for a d dimensional problem. We assume a discretization scheme using M nodal points, and assume that the average number of nodal points, whose shape functions are involved at an integration point, is N. Let n_g denote the number of Gaussian integration points per finite element in the FEM, or per Ω_s in the MLPG. In the following, some superscripts will be used to distinguish the same variables for the different MLPG methods presented in this paper.

The computational time for a single term of the element stiffness matrix can be estimated as

$$t_e^{FEM} \sim n_g^{FEM} t_{sh}^{FEM} \tag{73}$$

where t_{sh} is the average computational time for a shape function or its derivative. Hence, the computational time for the global stiffness matrix may be estimated as

$$t_K^{FEM} \sim d^2 M^{FEM} N^{FEM} t_e^{FEM} \sim d^2 n_g^{FEM} M^{FEM} N^{FEM} t_{sh}^{FEM} \tag{74}$$

For the MLPG methods, the computational time for the global stiffness matrix can also be estimated as

$$t_K \sim d^2 M N t_e \sim d^2 n_g M N t_{sh} \tag{75}$$

However, t_{sh} is greater than the t_{sh}^{FEM} if the interpolations are MLS or RBF, which involve the inversion of matrix at each integration point. As an example, here we only consider the MLS with first-order completeness [i.e., t=1, and m=3 in $\mathbf{P}(\mathbf{x})$], which means that the inversion of the matrix can be obtained theoretically, and the cost is lower, in this case, t_{sh}^{MLS} almost is the same as t_{sh}^{SF} in the case of Shepard function, where t_{sh}^{SF} requires the evaluation of N weight functions at each integration point. Hence, for the MLPG methods, the computational time for the global stiffness matrix can be rewritten as

$$t_K \sim d^2 n_g M N t_{sh} \sim d^2 n_g M N^2 t_w \tag{76}$$

where t_w is the computational cost of evaluating the weight function equation (22) or its derivative at a single evaluation point. From the analysis in Section 7, we see that

$$\begin{array}{l} n_g^{MLPG3} > n_g^{MLPG1} > n_g^{MLPG4} > n_g^{MLPG5} \\ \cong n_g^{FEM} > n_g^{MLPG2} = 1 \end{array} \tag{77}$$

Hence, for the present MLPG methods, we have

$$t_K^{MLPG2} < t_K^{MLPG5} < t_K^{MLPG4} < t_K^{MLPG1} < t_K^{MLPG3} \tag{78}$$

It can be seen that, when the cost for the computation in an MLPG method computation is compared to an FE solution with the same number of unknowns (dM), the FEM results are much less expensive. However, comparing the cost based on the approximation accuracy, the conclusion can be quite different. The accuracy of the FEM is less than that of the MLPG5 method, therefore more nodes are needed for the FEM method than that needed for the MLPG5 method to obtain the same accuracy of the results.

Now, we only consider the MLPG5. Assuming that t_{sh}^{FEM} and t_w are of the same order of magnitude, we have

$$\begin{aligned} \frac{t_K^{MLPG5}}{t_K^{FEM}} &\sim \frac{d^2 n_g^{MLPG5} M^{MLPG5} (N^{MLPG5})^2 t_w}{d^2 n_g^{FEM} M^{FEM} N^{FEM} t_{sh}^{FEM}} \\ &\sim \frac{M^{MLPG5} (N^{MLPG5})^2}{M^{FEM} N^{FEM}} \end{aligned} \tag{79}$$

where we assume $n_g^{MLPG5} \approx n_g^{FEM}$. For the same desired accuracy in the obtained numerical solution, $M^{FEM} >> M^{MLPG5}$. Moreover, according to the numerical results in Section 7 (Fig. 6a and 29a), the rate of convergence for $\|\cdot\|_0$ norm, in the present MLPG5 is about 4 with C^1 trial functions, while the rate of convergence in the FEM using linear C^0 elements is 2. Thus, it is reasonable to say that

$$M^{FEM} \sim (M^{MLPG5})^2 \tag{80}$$

In the 2-D FEM, for 3-noded triangular element (see Fig. 37), we have

$$N^{FEM} = 7 \tag{81}$$

(for 8-noded element in 2-D, N^{FEM}=21, see Fig. 38). For the MLPG5, if we take, for instance, the supports of the trial function and test function to be $3h$ and $0.9h$, respectively, we can obtain that

$$N^{MLPG5} \sim 25 \tag{82}$$

Obviously, $N^{MLPG5} > N^{FEM}$. Now, it can be derived, in this example, that

$$\frac{t_K^{MLPG5}}{t_K^{FEM}} \sim \frac{(N^{MLPG5})^2}{M^{MLPG5} N^{FEM}} \sim \frac{625}{(7) \cdot M^{MLPG5}} \approx \frac{89}{M^{MLPG5}} \tag{83}$$

In the present example, $M^{MLPG5} \sim$100. Thus, we find that:

$$\frac{t_K^{MLPG5}}{t_K^{FEM}} \sim 1.0 \tag{84}$$

As for the cost of solving the resulting algebraic equations, due to fact that $M^{FEM} >> M^{MLPG5}$ for the same accuracy, it is reasonable to estimate that this cost in MLPG5 is also less than that in FEM.

In general, in view of Eq. (80), we find from Eq. (79):

$$\frac{t_K^{MLPG5}}{t_K^{FEM}} \sim \frac{(N^{MLPG5})^2}{M^{MLPG5} N^{FEM}} \tag{85}$$

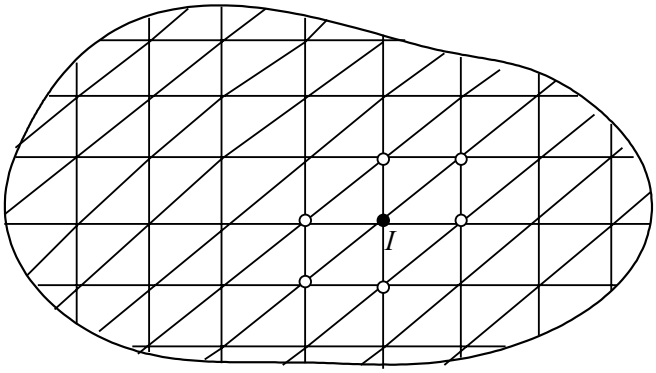


Figure 37 : 2-D 3-noded triangular mesh.

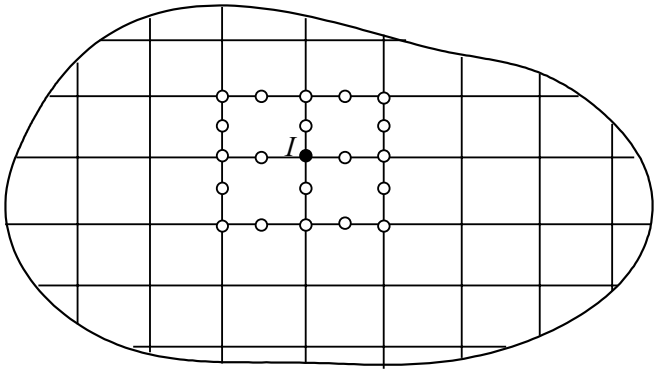


Figure 38 : 2-D 8-noded quadrilateral mesh.

As seen before, N^{FEM} for a 2-D, 3-noded triangular mesh, N^{FEM}=7; and for a 2-D, 8-noded quadrilateral mesh, N^{FEM}=21. Thus, the "band-width" of the FEM increases rapidly, as "higher-order" C^0 elements are used. However, the MLPG5 method, with MLS trial function, is inherently a "higher-order, but C^1 interpolation". Thus, N^{FEM}, and N^{MLPG5} are of the same order of magnitude, especially when higher-order FEM are used.

Thus, from Eq. (85), it can be derived that

$$\frac{t_K^{MLPG5}}{t_K^{FEM}} \sim \frac{N^{MLPG5}}{M^{MLPG5}} \tag{86}$$

which, for most realistic computations, leads to the conclusion that:

$$\frac{t_K^{MLPG5}}{t_K^{FEM}} < 1.0 \tag{87}$$

It may be concluded that:

1. MLPG5 has little or no preprocessing and postprocessing costs, thus the human-labor cost in MLPG5 is negligible compared to that in FEM;

2. For the same accuracy, the computational cost in MLPG5 is also lower than that in FEM, or at least comparable to that in FEM.

Thus, the MLPG5 method has the potential to replace the FEM as a method of choice for engineering analysis.

9 Conclusion

We have presented a comparison study of the costs, efficiency and accuracy of a variety of meshless trial and test functions in this paper, based on the general concept of the meshless local Petrov-Galerkin (MLPG) method. 5 types of trial functions and 6 types of test functions are examined. *Different test functions result in different MLPG methods, and 6 such MLPG methods are presented in this paper*. In all these six MLPG methods, absolutely no meshes are needed either for interpolation of the trial and test functions, or for the integration of the weak forms, while other meshless methods (such as the EFG) require background cells. The nodal shape functions, for the trial function, from all the 5 interpolations (MLS, RKPM, PU, Shepard, and RBF) are of a rational form, and are highly complex, leading to the difficulty in the numerical integration of the weak-forms, especially in the Galerkin equivalent of the MLPG (viz., the MLPG6). A large support of the test function also makes the shape function for the trial functions more complicated. Hence, it is suggested that the size of the support of the test function should be smaller (between 0 and *h*, *h* being the nodal spacing). Because the complicated shape functions for the trial-function are inevitable at the present stage, to develop a fast and robust meshless method, we try to avoid the use of the domain integral [involved in the weak-form] by choosing a proper test function. *The MLPG5 method avoids the need for both a domain integral as well as a singular integral, and thus shows a great promise in engineering applications*.

Convergence studies in the numerical examples show that all of the six MLPG methods presented here possess excellent rates of convergence for both the unknown variables and their derivatives. The analysis of computational costs shows that the *MLPG5 result is less expensive in computational costs, and definitely less expensive in human-labor costs than the FEM, or BEM*. Particu-

larly, due to its speed, accuracy and robustness, MLPG5 may be expected to replace the FEM or BEM in the near future.

Acknowledgement: The authors are pleased to acknowledge the support of NASA, and ONR (and the respective program officials Drs. I. S. Raju, and Y. D. S. Rajapakse) of their research efforts. Special thanks are also expressed to Dr. Er-Ping (Tony) Chen of Sandia National Labs, Livemore, CA, for his support of, and interest in, the ideas presented herein.

References

Atluri SN (2002): *Methods of computer modeling in engineering and sciences*. Tech. Science Press, 1400 pages.

Atluri SN, Cho JY, Kim HG (1999): Analysis of the beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations. *Comput. Mech.* 24: 334-347

Atluri SN, Shen S (2002): *The meshless local Petrov-Galerkin (MLPG) method*. Tech. Science Press.

Atluri S.N., Sladek J, Sladek V, Zhu T (2000): The local boundary integral equation (LBIE) and it's meshless implementation for linear elasticity. *Comput. Mech.* 25: 180-198

Atluri SN, Zhu T (1998a): A new meshless local Petrov-Galerkin (MLPG) approach to nolinear problems in computational modeling and simulation. *Comput. Modeling Simulation in Engrg.* 3: 187-196

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

Atluri SN, Zhu T (2000): The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics, *Comput. Mech.* 25: 169-179

Atluri SN, Kim HG, Cho JY (1999): A critical assessment of the truly meshless local Petrov-Galerkin (MLPG) and local boundary integral equation (LBIE) methods. *Comput. Mech.* 24: 348-372

Atluri SN, Zhu T (2000): New concepts in meshless methods. *Int. J. Numer. Mech. Engrg.* 47: 537-556

Babuska I, Melenk JM (1997): The partition of unity method. *Int. J. Num. Meth. Engrg.* 40: 727-758

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

Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P (1996): Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech. Engrg.* 139:3-47

Ching HK, Batra RC (2001): Determination of Crack Tip Fields in Linear Elastostatics by the Meshless Local Petrov-Galerkin (MLPG) Method. *CMES: Computer Modeling in Engineering & Sciences* 2 (2): 273-290

Duarte C, Oden JT (1996): Hp-cloud – a meshless method to solve boundary-value problems. *Comput. Meth. Appl. Mech. Eng.* 139: 237-262

Gingold RA, Monaghan JJ (1977): Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. Roy. Astron. Soc.* 181:375-389

Gu YT, Liu GR (2001): A meshless local Petrov-Galerkin (MLPG) formulation for static and free vibration analysis of thin plates. *CMES: Computer Modeling in Engineering & Sciences* 2 (4): 463-476

Jin X, Li G, Aluru NR (2001): On the equivalence between Least-squares and kernel approximations in meshless methods. *CMES: Computer Modeling in Engineering & Sciences* 2 (4): 463-476

Kim HG, Atluri SN (2000): Arbitrary placement of secondary nodes, and error control, in the meshless local Petrov-Galerkin (MPLG) method. *CMES: Computer Modeling in Engineering & Sciences* 1(3): 11-32

Lin H, Atluri SN (2000): Meshless local Petrov-Galerkin (MPLG) method for convection-diffusion problems. *CMES: Computer Modeling in Engineering & Sciences* 1 (2): 45-60

Lin H, Atluri SN (2001): The meshless local Petrov-Galerkin (MPLG) method for solving incompressible Navier-stokes equations. *CMES: Computer Modeling in Engineering & Sciences* 2 (2): 117-142

Liu WK, Chen Y, Uras RA, Chang CT (1996): Generalized multiple scale reproducing kernel particle methods. *Comput. Methods Appl. Mech. Engrg.* 139:91-157

Long S, Atluri SN (2002): A meshless local Petrov-Galerkin (MLPG) method for solving the bending problem of a thin plate. *CMES: Computer Modeling in Engineering & Sciences* 3 (1)

Nayroles B, Touzot G, Villon P (1992): Generalizing

the finite element method: diffuse approximation and diffuse elements. *Comput. Mech.* 10: 307-318

Onate E, Idelsohn S, Zienkiewicz OC, Taylor RL (1996): A finite point method in computational mechanics. Applications to convective transport and fluid flow. *Int. J. Numer. Methods Engrg.* 39:3839-3866

Shepard D (1968): A two-dimensional function for irregularly spaced points. *Proc. Of ACM Nat'l Conf.*: 517-524

Sladek V, Sladek J, Atluri SN (2000): Numerical integration of singularities in meshless implementation of the LBIE. *Comput. Mech.* 25: 394-403

Sladek J, Sladek V, Atluri SN (2001): A pure contour formulation for the meshless local boundary integral equation method in thermoelasticity. *CMES: Computer Modeling in Engineering & Sciences* 2 (4): 423-434

Sukumar N, Moran B, Belytschko T (1998): The natural element method in solid mechanics. *Int. Num. Meth. Eng.* 43: 839-887

Wu Z (1992): Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory Appl.* 8: 1-10

Wu Z (1995): Compactly supported positive definite radial functions. Adv. *Comput. Math.* 4:283-292

Wendland H (1995): Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Adv. Comput. Math.* 4:389-396

Wendland H (1998): Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *J. Approx. Theory* 93: 258-272

Wendland H (1999): Meshless Galerkin methods using radial basis function. *Math. Comput.* 68(228): 1521-1531

Zhu T, Atluri SN (1998): A modified collocation and a penalty formulation for enforceing the essential boundary conditions in the element free Galerkin method. *Comput. 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. *Comput. Mech.* 21: 223-235

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