# A New Implementation of the Meshless Finite Volume Method, Through the MLPG “Mixed” Approach

S. N. Atluri[1], Z. D. Han[1] and A. M. Rajendran[2]

**Abstract:** The Meshless Finite Volume Method (MFVM) is developed for solving elasto-static problems, through a new Meshless Local Petrov-Galerkin (MLPG) “Mixed” approach. In this MLPG mixed approach, both the strains as well as displacements are interpolated, at randomly distributed points in the domain, through local meshless interpolation schemes such as the moving least squares(MLS) or radial basis functions(RBF). The nodal values of strains are expressed in terms of the independently interpolated nodal values of displacements, by simply enforcing the strain-displacement relationships directly by collocation at the nodal points. The MLPG local weak form is then written for the equilibrium equations over the local sub-domains, by using the nodal strains as the independent variables. By taking the Heaviside function as the test function, the local domain integration is avoided; this leads to a Meshless Finite Volume Method, which is a counterpart to the mesh-based finite volume method that is popular in computational fluid dynamics. The present approach eliminates the expensive process of directly differentiating the MLS interpolations for displacements in the entire domain, to find the strains, especially in 3D cases. Numerical examples are included to demonstrate the advantages of the present methods: (i) lower-order polynomial basis can be used in the MLS interpolations; (ii) smaller support sizes can be used in the MLPG approach; and (iii) higher accuracies and computational efficiencies are obtained.

**keyword:** Meshless Local Petrov-Galerkin approach (MLPG), Finite Volume Methods, Mixed Methods, Radial Basis Functions (RBF), and Moving Least Squares (MLS).

## 1 Introduction

The meshless methods have been found to be attractive, mainly due to the possibility of overcoming the drawbacks of mesh-based methods, such as the labor-intensive process of mesh-generation, locking, poor derivative solutions, etc. The meshless methods may also eliminate the mesh distortion problems once the solid/structure undergoes large deformations, in which case, adaptive refinement and adaptive remeshing are required. Several meshless methods have been developed, based on global weak forms, such as the smooth particle hydrodynamics (SPH), and the element-free methods and so on. They require certain meshes or background cells [over the solution domain, for purposes of integration of the weak-form], which may also become distorted during large deformations. In contrast, the meshless local Petrov-Galerkin (MLPG) approach pioneered by Atluri and his colleagues [Atluri (2004)] is based on writing the local weak forms of PDEs, over overlapping local sub-domains. The integration of the weak-form is also performed within the local sub-domains; thus negating any need for any kind of meshes or background cells, making the MLPG approach a truly meshless method. The MLPG approach has been used to solve various problems successfully, including those in elasto-statics [Li, Shen, Han and Atluri (2003); Han and Atluri (2003ab,2004a)], elasto-dynamics [Han and Atluri (2004b)], fracture mechanics [Atluri, Kim and Cho (1999)], and fluid mechanics [Lin and Atluri (2001)] and so on. All these successes demonstrate that the MLPG method is one of the most promising alternative methods for computational mechanics.

As a general framework for meshless methods, the MLPG approach provides the flexibility in choosing the trial and test functions, as well as the sizes and shapes of local sub-domains, and employs various[unsymmetric or symmetric] weak forms of PDEs. As a special case, the Heaviside function can be chosen as the test function,

[1] University of California, Irvine Center for Aerospace Research & Education
5251 California Avenue, Suite 140
Irvine, CA, 92612, USA
[2] Army Research Office, RTP, NC

and applied to the symmetric-weak forms over the local sub-domains; thus eliminating the domain integrals [labeled as MLPG-5Atluri and Shen (2002)]. It has a clear physical meaning that the local weak form represents the momentum balance law of the local sub-domains. This is very similar to the finite volume methods, which have been widely used in computational fluid mechanics. The finite volume methods are usually second-order accurate, based on the integral form of the governing equations, over the non-overlapping sub-domains. The MLPG5 method leads to what may be called as the "Meshless Finite Volume Methods [MFVM]. With the use of iterative solvers, the finite volume methods are more efficient to treat the coupling and non-linearity in an iterative way. In recent years industrial computational fluid dynamics has been dealing with the meshes of the order of up to 100 million cells. The smooth particle hydrodynamics (SPH) uses the so-called particles, interacting with each other through the meshless approximations [Lucy (1977); Gingold and Monaghan (1977)]. Since the standard SPH method has the problem to interpolate exactly when the particles are unevenly spaced and sized, many improvements were made to improve the completeness of the SPH approximation, such as the normalization [Johnson and Beissel (1996)] and the MLS approximation [Dilts (1999)].

The problems of similar scale also exist in structural analysis, especially for those undergoing the large nonlinear deformations under high-speed impact loading as in the case of a projectile penetration problem. It is well known that this has been beyond the capability of the traditional finite element methods, due to the mesh problems and the poor accuracy of the second-order variables. For example, in projectile penetration simulations, Johnson, Stryk, Beissel, and Holmquist (2002) used a SPH based algorithm to convert extremely distorted finite elements into particles to alleviate eroding (dropping) element mass from the calculation and to avoid solution time steps becoming extremely small. The MFVM described in this paper may offer a viable alternative for this class of problems.

In the present work, a new meshless finite volume method [MFVM] is developed for solving elasto-static problems, through an MLPG "mixed" approach. The Heaviside function is used as the test function, in the local-weak-form of the equilibrium equation. Independent meshless approximations are used for both the strains, as well as the displacements. The strain-displacement compatibility is enforced at nodal points by using the collocation method; thus expressing the independent nodal strains in terms of nodal displacements. These strains are then used in the local symmetric weak-form of the equilibrium equations. It retains the same physical meaning as the momentum balance law of the local sub-domains, while the accuracy of the secondary variables has been improved. Theoretically, the present method requires that the trial functions possess only C0 continuity. In contrast, C1 continuities are required for the trial functions if the derivatives of the displacements are directly used in the weak-form. Numerically, the strains are interpolated directly via the meshless approximations, without the calculation of the derivatives of the shape functions. The MLPG mixed method is thus computationally more efficient, because the calculation of the derivatives of the interpolation functions in the meshless approximations, everywhere in the vdomain, is computationally costly. In addition, the second-order polynomial bases are required for the better approximation, to avoid shear-locking if the MLS is used [Han and Atluri (2004a)], in the primal MLPG method (based only on displacement interpolations) Also in the MLPG primal method, a larger support size should be chosen, in order to make the MLS approximation non-singular, which leads to over-smoothed results. The present mixed method requires only a first-order polynomial basis in the MLS approximations of both strains as well as displacements. A smaller support size can be used in the present MLPG mixed method, and the number of nodes is reduced dramatically, especially for 3D cases. The local integrals in the present method contain only the strains, without involving the derivatives of the displacement explicitly. Thus, the present MFVM method, based on the MLPG mixed approach, is more suitable for non-linear problems with large deformations.

The main body of the paper begins with a brief description of the augmented RBF, and MLS approximations in Section 2. The MLPG finite volume method is formulated in Section 3. The numerical implementation with the pseudo codes is presented in Section 4. Numerical examples are given in Section 6, and the conclusions and discussions are given in Section 7.

## 2 Meshless Approximations

This section summarizes, for the sake of completeness, the meshless approximations for the MLPG methods, including the radial basis functions (RBF) and the moving least squares (MLS). More details can be found in [Atluri (2004)].

The MLS method of interpolation is generally considered to be one of the best schemes to interpolate random data with a reasonable accuracy, because of its completeness, robustness and continuity. With the MLS, the distribution of a function $u$ in $\Omega_s$ can be approximated, over a number of scattered local points $\{\mathbf{x}_i\}$, $(i = 1,2,...,n)$, as,

$$u(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega_s \tag{1}$$

where $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x})]$ is a monomial basis of order $m$; and $\mathbf{a}(\mathbf{x})$ is a vector containing coefficients, which are functions of the global Cartesian coordinates $[x_1,x_2,x_3]$, depending on the monomial basis. They are determined by minimizing a weighted discrete $L_2$ norm, defined, as:

$$\begin{aligned} J(\mathbf{x}) &= \sum_{i=1}^{m} w_i(\mathbf{x})[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - \hat{u}_i]^2 \\ &\equiv [\mathbf{P}\cdot\mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}]^T\mathbf{W}[\mathbf{P}\cdot\mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}] \end{aligned} \tag{2}$$

where $w_i(\mathbf{x})$ are the weight functions and $\hat{u}_i$are the fictitious nodal values.

One may obtain the shape function as,

$$u(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\hat{\mathbf{u}} \equiv \mathbf{\Phi}^T(\mathbf{x})\hat{\mathbf{u}} \quad \forall \mathbf{x} \in \partial\Omega_x \tag{3}$$

where matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are defined by

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T\mathbf{W}\mathbf{P} \quad \mathbf{B}(\mathbf{x}) = \mathbf{P}^T\mathbf{W} \quad \forall \mathbf{x} \in \partial\Omega_x \tag{4}$$

The weight function in Eq. (2) defines the range of influence of node $I$. Normally it has a compact support. A fourth order spline weight function is used in the present study.

The radial basis functions (RBF) approximation possesses the Dirac delta function property of the shape functions, and leads to a simplicity of their derivatives. However, they lack completeness, as in the standard SPH. It has been improved by introducing additional polynomials, as the augmented RBF. The RBF approximation is globally continuous only for the case when the approximation is also global. The shape functions are non-continuous if they are used locally [Han and Atluri (2004a)].

Consider a sub-domain $\Omega_s$, the neighborhood of a point $\mathbf{x}$, which is local in the solution domain. To approximate the distribution of function $u$ in $\Omega_s$, over a number of scattered points $\{\mathbf{x}_i\}$, $(i = 1,2,...,n)$, the local augmented RBFs interpolate $u(\mathbf{x})$ of $u$, $\forall \mathbf{x} \in \Omega_s$ can be defined by [Golberg, Chen and Bowman (1999)]

$$u(\mathbf{x}) = \mathbf{R}^T(\mathbf{x})\mathbf{a} + \mathbf{P}^T(\mathbf{x})\mathbf{b} \quad \forall \mathbf{x} \in \Omega_s \tag{5}$$

where $\mathbf{R}^T(\mathbf{x}) = [R_1(\mathbf{x}), R_2(\mathbf{x}), ..., R_n(\mathbf{x})]$ is a set of radial basis functions centered around the $n$ scattered points; $\mathbf{a}^T = [a_1,a_2,...,a_n]$ is a vector of constant coefficients; $\mathbf{P}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x})] = [1,x,y,z,x^2,y^2,z^2,xy,yz,zx,....]$ is a monomial basis of order $m$; $\mathbf{b}^T = [b_1,b_2,...,b_m]$ is a vector of constant coefficients. The radial basis function has the following general form

$$R_i(\mathbf{x}) = R_i(r_i) \quad \text{and} \quad r_i = \|\mathbf{x} - \mathbf{x}_i\| \tag{6}$$

To determine the coefficients $\mathbf{a}$ and $\mathbf{b}$, one may enforce the interpolation to satisfy the given values at the scattered points, as:

$$\begin{aligned} &u(\mathbf{x}_i) = \mathbf{R}^T(\mathbf{x}_i)\mathbf{a} + \mathbf{P}^T(\mathbf{x}_i)\mathbf{b} i = 1,2,...,n \quad \text{or} \\ &\mathbf{u}^e = \mathbf{R}_0\mathbf{a} + \mathbf{P}_0\mathbf{b} \end{aligned} \tag{7a}$$

and

$$\sum_{i=1}^{n} p_j(\mathbf{x}_i)a_i = 0 \quad j = 1,2,...,m \quad \text{or} \quad \mathbf{P}_0^T\mathbf{a} = \mathbf{0} \tag{7b}$$

By solving Eq. (7), and substituting the solution into Eq. (5), one may obtain the shape function as:

$$u(\mathbf{x}) = \left[\mathbf{R}^T(\mathbf{x}), \mathbf{P}^T(\mathbf{x})\right]\mathbf{G}\mathbf{u}^e \equiv \mathbf{\Phi}^T(\mathbf{x})\mathbf{u}^e \quad \forall \mathbf{x} \in \Omega_s \tag{8}$$

where the matrix $\mathbf{G}$ is defined as:

$$\mathbf{G} \equiv \begin{bmatrix} \mathbf{R}_0 & \mathbf{P}_0 \\ \mathbf{P}_0^T & \mathbf{0} \end{bmatrix}^{-1} \tag{9}$$

In the present study, the compactly supported RBFs are used in a local way, and chosen as the weight function used in the MLS.

## 3 MLPG Finite Volume Method and Numerical Discretization

Consider a linear elastic body in a 3D domain $\Omega$, with a boundary $\partial\Omega$, shown in Figure 1. The solid is assumed to undergo infinitesimal deformations. The equations of balance of linear and angular momentum can be written as:

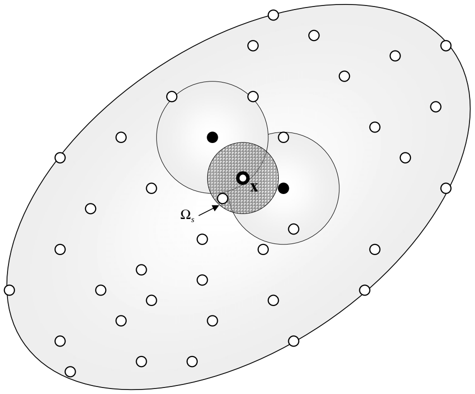


**Figure 1** : A local sub-domain around point **x**

$$\sigma_{ij,j}+f_i=0; \quad \sigma_{ij}=\sigma_{ji}; \quad ()_{,i}\equiv\frac{\partial}{\partial\xi_i} \tag{10}$$

where $\sigma_{ij}$ is the stress tensor, which corresponds to the displacement field $u_i$; $f_i$ is the body force. The corresponding boundary conditions are given as follows,

$$u_i=\overline{u}_i \quad \text{on} \quad \Gamma_u \tag{11a}$$

$$t_i\equiv\sigma_{ij}n_j=\overline{t}_i \quad \text{on} \quad \Gamma_t \tag{11b}$$

where $\overline{u}_i$ and $\overline{t}_i$ are the prescribed displacements and tractions, respectively, on the displacement boundary $\Gamma_u$ and on the traction boundary $\Gamma_t$, and $n_i$ is the unit outward normal to the boundary $\Gamma$.

In the MLPG approaches, one may write a weak form over a local sub-domain $\Omega_s$, which may have an arbitrary shape, and contain the a point **x** in question. A generalized local weak form of the differential equation (10) over a local sub-domain $\Omega_s$, can be written as:

$$\int_{\Omega_s}(\sigma_{ij,j}+f_i)v_i d\Omega=0 \tag{12}$$

where $u_i$ and $v_i$ are the trial and test functions, respectively.

By applying the divergence theorem, Eq. (12) may be rewritten in a symmetric weak form as [Han and Atluri (2004a)]:

$$\int_{\partial\Omega_s}\sigma_{ij}n_jv_id\Gamma-\int_{\Omega_s}(\sigma_{ij}v_{i,j}-f_iv_i)d\Omega=0 \tag{13}$$

Imposing the traction boundary conditions in (11), one obtains

$$\int_{L_s}t_iv_id\Gamma+\int_{\Gamma_{su}}t_iv_id\Gamma+\int_{\Gamma_{st}}\overline{t}_iv_id\Gamma \\ -\int_{\Omega_s}(\sigma_{ij}v_{i,j}-f_iv_i)d\Omega=0 \tag{14}$$

where $\Gamma_{su}$ is a part of the boundary $\partial\Omega_s$ of $\Omega_s$, over which the essential boundary conditions are specified. In general, $\partial\Omega_s=\Gamma_s\cup L_s$, with $\Gamma_s$ being a part of the local boundary located on the global boundary, and $L_s$ is the other part of the local boundary which is inside the solution domain. $\Gamma_{su}=\Gamma_s\cap\Gamma_u$ is the intersection between the local boundary $\partial\Omega_s$ and the global displacement boundary $\Gamma_u$; $\Gamma_{st}=\Gamma_s\cap\Gamma_t$ is a part of the boundary over which the natural boundary conditions are specified.

Therefore, a local symmetric weak form (LSWF) in linear elasticity can be written as:

$$\int_{\Omega_s}\sigma_{ij}v_{i,j}d\Omega-\int_{L_s}t_iv_id\Gamma-\int_{\Gamma_{su}}t_iv_id\Gamma \\ =\int_{\Gamma_{st}}\overline{t}_iv_id\Gamma+\int_{\Omega_s}f_iv_id\Omega \tag{15}$$

One may use the Heaviside function as the test function inj the local symmetric weak form in Eq. (15), and obtain,

$$-\int_{L_s}t_id\Gamma-\int_{\Gamma_{su}}t_id\Gamma=\int_{\Gamma_{st}}\overline{t}_id\Gamma+\int_{\Omega_s}f_id\Omega \tag{16}$$

Eq. (16) has the physical meaning that it represents the balance law of the local sub-domain $\Omega_s$, with the traction boundary conditions being enforced [Atluri (2004); Li, Shen, Han, and Atluri (2003)].

With the constitutive relations of an isotropic linear elastic homogeneous solid, the tractions in Eq. (16) can be written in term of the strains:

$$t_i=\sigma_{ij}n_j=E_{ijkl}\varepsilon_{kl}n_j \tag{17}$$

where,

$$E_{ijkl}=\lambda\delta_{ij}\delta_{kl}+\mu(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}) \tag{18}$$

with λ and $\mu$ being the Lame's constants.

Consider a local sub-domain $\Omega_s$, centered on each nodal point $\mathbf{x}^{(I)}$; then the approximation of traction vectors on the boundary of $\Omega_s$ can be expressed by considering the nodal strains as independent variables. With the use of the shape function in Section 2, the strains are independently interpolated, as,

$$\varepsilon_{kl}(\mathbf{x}) = \sum_{K=1}^{N} \Phi^{(K)}(\mathbf{x})\varepsilon_{kl}^{(K)} \tag{19}$$

With Eqs. (17) and (19), one may discretize the local symmetric weak-form of Eq. (16), as,

$$\begin{aligned} &-\sum_{K=1}^{N} [\int_{L_s} \Phi^{(K)}(\mathbf{x})E_{ijkl}n_j d\Gamma]\ \varepsilon_{kl}^{(K)} \\ &-\sum_{K=1}^{N} [\int_{\Gamma_{su}} \Phi^{(K)}(\mathbf{x})E_{ijkl}n_j d\Gamma]\ \varepsilon_{kl}^{(K)} \\ &= \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega \end{aligned} \tag{20}$$

It clearly shows that no derivatives of the shape functions are involved in the local integrals. Instead, if Eq. (20) is written in terms of the displacement variables, these integrals will involve the derivatives of the shape functions [Atluri (2004)]. It is well known that the meshless approximation is not efficient in calculating such derivatives everywhere in the domain, especially when the MLS approximation is used. Thus, the efficiency of the present method is improved over the traditional MLPG [primal] displacement methods. Secondly, the requirement of the completeness and continuity of the shape functions is reduced by one-order, because the strains, which are the secondary field variables, are approximated independently of the displacements. Thus, lower-order polynomial terms are required in the meshless approximations, and a smaller nodal influence size can be chosen, to speed up the calculation of the shape functions. On the other hand, the number of equations in Eq. (20) is less than the number of the independent strain variables, because the nodal strain variables are more than the displacement ones[ in 3D, there are six nodal-strain variables, but only 3 displacement nodal-variables]. One may reduce the number of the variables by transforming the strain variables back to the displacement variables via the collocation methods, without any changes to Eq. (20). First, the interpolation of displacements can also be accomplished by using the same shape function, for the nodal displacement variables, and written as,

$$u_i(\mathbf{x}) = \sum_{J=1}^{N} \Phi^{(J)}(\mathbf{x})u_i^{(J)} \tag{21}$$

For linear elasto-statics, the strain-displacement relations are:

$$\varepsilon_{kl} = \frac{1}{2}(u_{k,l} + u_{l,k}) \tag{22}$$

The standard collocation method may be applied to enforce Eq. (22) *only at each nodal point* $\mathbf{x}^{(I)}$, instead of the entire solution domain. Thus, the nodal strain variables are expressed in terms of the nodal displacement variables, as

$$\varepsilon_{kl}(\mathbf{x}^{(I)}) = \frac{1}{2}[u_{k,l}(\mathbf{x}^{(I)}) + u_{l,k}(\mathbf{x}^{(I)})] \tag{23}$$

With the displacement approximation in Eq. (21), the two sets of nodal variables can be transformed through a linear algebraic matrix:

$$\varepsilon_{kl}^{(I)} = H_{klm}^{(I)(J)} u_m^{(J)} \tag{24}$$

where the transformation matrix $H$ is banded.

The number of system equations is then reduced to the same number as the nodal-displacement variables, after the transformation. In addition, such a transformation is performed locally, and the system matrix retains its bandedness. For numerical implementation, it is not necessary to calculate and store the matrix $H$ explicitly. The integrals in Eq. (20) are only related to a few nodal points which are near to the point of interestion, $\mathbf{x}^{(I)}$, which means only a very small portion of the transformation matrix $H$ is used. It is possible to calculate this portion from Eq. (24) dynamically, which is less computationally costly with only a few local nodal points involved.

With the meshless approximations, another problem is that the essential boundary conditions can not be imposed directly even with the use of the RBF approximation, because the shape function possesses the Dirac delta property only at the corresponding nodes. Although one may impose the essential boundary conditions at all nodes on the prescribed displacement boundary, these conditions are still not satisfied everywhere on the boundary, except at the nodal points. The reason is that the boundary values everywhere on the boundary, even through the RBF approximation, depend not only on the boundary nodes, but also on the related ones inside the domain. This is quite different from those in the element-based methods, in which the boundary values are interpolated through only the nodes at the boundary nodes. In the present study, the collocation method is also used to impose the displacement boundary conditions. For a nodal point $\mathbf{x}^{(I)}$, if its $i$th displacement DOF belongs to the displacement boundary, i.e., $u_i^{(I)} \in \Gamma_{su}$, the corresponding system equation can be replaced by the one generated from the collocation for this particular DOF, as

$$\alpha u_i(\mathbf{x}^{(I)}) = \alpha \overline{u}_i(\mathbf{x}^{(I)}) \tag{25}$$

This standard collocation still keeps the system equations sparse and banded.

It should be pointed out that the present method is formulated based on the nodal points fully within the local sub-domains, as shown in Eqs. (20), (24) and 4. It makes the present method ready for parallel computation and work with iterative solvers or for solving transient impact problems [Han and Atluri (2004b)].

## 4 Numerical Implementation

Several numerical aspects are discussed in this section, mainly focusing on the issues in numerical implementation and related techniques. The pseudo codes in MATLAB are given in each topic. Even though all algorithms are demonstrated for 2D problems in this section, they are extendable for the 3D cases, quite easily.

### 4.1 Data Preparation

Although only the scattered nodal points are required in the meshless method, it is not efficient to define the boundaries for a complicated domain. The easiest way is to use the robust Delaunay algorithm to triangulate the solution domain into triangles. These triangles are only used to define the solution domain and have no quality requirement, because they are not used for the interpolation or the integration. The steps in Table 1 are used to prepare the data for any general problems.

### 4.2 Quadrature Techniques

In the present method, the tractions are involved in the integrals, which need to be evaluated over the boundary of the local sub-domains. Polynomial termas for the traction can not be expected over the entire local boundary, which may not be covered by all local nodes. In addition, the integrals contain the normal $\mathbf{n}$ to the boundary, and the trigonometric functions are involved. It is well known that the conventional numerical quadrature schemes are designed for polynomials, but are not efficient for trigonometric functions [Han and Atluri (2004)]. For 2D problems, the numerical errors may be controlled by simply increasing the order of the Gauss quadrature scheme, or subdividing the domain of integration into small segments for better accuracy. It has been reported that the subdivision algorithm is much more efficient, than in the case when the integration is performed over the entire domain with a large number of integration points [Atluri, Kim and Cho (1999), Sellountos and Polyzos (2003)]. For those nodes on the global boundary, the intersection between the local and global boundaries also needs some special treatment. In the present implementation, a simple subdivision algorithm is used to simplify the numerical quadrature and improve the accuracy.

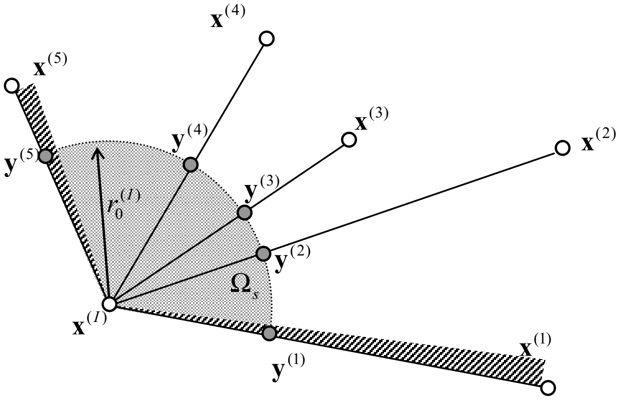


**Figure 2** : a local sub-domain around point $\mathbf{x}$

Consider a local circular sub-domain centered at node $I$, $\mathbf{x}^{(I)}$, with a radius denoted by $r_0^{(I)}$. By drawing a line from node $I$ to its neighbor node $J$, $\{\mathbf{x}^{(J)}\}$, $(J = 1, 2, ..., m)$, a point can be obtained at the intersection between the line and the local circle, denoted by $\{\mathbf{y}^{(J)}\}$, $(J = 1, 2, ..., m)$. A subset of these intersecting points is used to divide the integration domain, i.e. the local circle. It should be pointed out that the intersection points between the local and global boundaries are automatically included in $\mathbf{y}^{(J)}$. Then these special points are kept in the subset and used as the starting and ending points. A set of angles is obtained for performing the numerical integra-

**Table 1** : Sample codes for data preparation

i) read model data

```
load ‘node.dat’ node;               % node contains the node coordinates in rows, as
                                    % [ x1, y1; x2, y2; … ; xn yn]
load ‘element.dat’ element;         % element contains the triangles in rows, as
                                    % […; n1_Ti, n2_Ti, n3_Ti; …]
numnode=size(node,1);               % number of nodes
numelem=size(element,1);            % number of triangles
```

ii) search the global boundary

```
% mlpgedge contains the definition of the global boundary
% with anti-clock-wise node pairs in rows, as
% [… … … ; n1_of_Pi, n2_Pi; … …]

mlpgedge = setdiff( ...
        [element(:,[1 2]);element(:,[2 3]);element(:,[3 1])], ...
        [element(:,[2 1]);element(:,[3 2]);element(:,[1 3])], ...
        'rows');
clear element;              % the triangles are removed after searching the global boundary
```

iii) search neighbor nodes

```
% mlpgnode contains the neighbor node information in rows, as
% [… … ; m, I, n1, n2, … n_m-1;… …]
% where m is the number of the neighbor node of node I, include node I itself.

supportSize = 2.05;         % the radius of the nodal influence domain
testSize = 0.55;            % the radius of the local sub-domain

mlpgnode = zeros(numnode,1);
for i=1:numnode
   d =sqrt((node(:,1)-node(i,1)).^2 + (node(:,2)-node(i,2)).^2);
   neighNode = find(d < supportSize+testSize);
   mlpgnode(i,1:size(neighNode,1)+1) = [size(neighNode,1) i setdiff(neighNode,[i])'];
end
clear d neighNode;
```

tion, as $\{\theta_0, \theta_1, \cdots, \theta_t\}$. The starting angle $\theta_0$ is not equal to the ending angle $\theta_t$ for the nodes on the global boundary. In the present study, the radii of the local sub-domains for the nodes within the solution domain are so chosen that the local sub-domains do not intersect with the global boundary. Thereafter, the subdivided angles cover the entire local circle for these internal nodes. The subdivision is illustrated in Figure 2.

In addition, the subdivision can also be used to divide the local sub-domain in pie slices with node $I$ as the center, if domain integrals are required, such as the body forces. Then, all the integrals over the local sub-domain can be calculated by using the simple Gaussian quadrature.

The codes in Table 2 are used to determine the local subdivision.

### *4.3 Post Processing*

For the MLS approximation, the fictitious nodal values are obtained after solving the system equations. A simple procedure in Table 3 is used to calculate the actual nodal

**Table 2** : Sample codes for quadrature techniques

i) search the starting and ending angles for the nodes on the global boundary

```
% mlpgangle contains the starting and ending angles for all nodes
% [… … … ; starting_angle_of_Node_I, ending_angle_of_Node_I; … …]
% -inf and inf are used, respectively, for the internal nodes

mlpgangle = zeros(numnode,2);
dr = node(mlpgedge(:,2),:) - node(mlpgedge(:,1),:);        % length of all boundary edges
% calculate the angles of the boundary edges
% and used as the starting angle of the first node of the edge
mlpgangle(mlpgedge(:,1),1) = cart2pol(dr(:,1),dr(:,2));
mlpgangle(find(mlpgangle>pi-eps),1)=-pi;           % remove the numerical tolerance

% reverse the angle of the boundary edges,
% and used as the ending angle of the second node of the edge
mlpgangle(mlpgedge(:,2),2) = pi+mlpgangle(mlpgedge(:,1),1);

% regularize the starting and ending angles
da = mlpgangle(:,2)-mlpgangle(:,1);
ii = find(da > 2*pi);
mlpgangle(ii,2) = mlpgangle(ii,2)-2*pi;
ii = find(da < 0);
mlpgangle(ii,1) = mlpgangle(ii,1)-2*pi;

ii = setdiff([1:numnode]', mlpgedge(:,1));             % set -inf & inf for all internal nodes
mlpgangle(ii,1) = -inf;
mlpgangle(ii,2) = inf;
clear dr da ii;
```

ii) sub-divide the local circle

```
% mlpgnode contains the sub-divided angles for integration
% [… … … ; number_of_angles, sita_0, sita_1, …, sita_t; … …]

f0 = linspace(0,1,9)';      % the maximum number of the angles is set to 10
for i=1:numnode
   ii = mlpgnode(i,3:mlpgnode(i,1));        % neighbor nodes excluding itself
   sita = cart2pol(node(ii,1)-node(i,1),node(ii,2)-node(i,2));   % all intersecting  angles

   startangle = max(mlpgangle(i,1),min(sita));
   ii = find(sita <startangle);
   sita(ii) = sita(ii)+2*pi;

   endangle = min(startangle+2*pi,mlpgangle(i,2));
   sita = sita(find(sita <=endangle));
   [f,sita] = ecdf(sita);     % cumulative distribution function is used for subsetting
   sita = sort(unique(sita(dsearchn(f,f0))));
   if(sita(1) > startangle+eps)      % include the starting angle as the first one
      sita = [startangle; sita];
   end
   if(sita(end) < endangle-eps)   % include the ending angle as the last one
      sita = [sita;endangle];
   end
   mlpgintangle(i,1:size(sita,1)+1) = [size(sita,1) sita'];
end
clear f0 ii sita startangle endangle
```

iii) perform the numerical integration

```
for n=1:numnode                % start of node loop
   nn=mlpgnode(n,1);           % the number of the neighbor nodes
   sctr=mlpgnode(n,2:nn+1);    % IDs of all local scattered nodes, include node I

   nseg = mlpgintangle(n,1);   % the number of sub-divided angles
   angle = mlpgintangle(n,2:nseg+1); % the angles
   xn = node(n,:);             % coordinates of the node of interesting
   pts = node(sctr,:);         % coordinates of all local nodes

   for(seg = 1:nseg-1)         % sub-division loop
      angle0 = angle(seg);
      dangle = angle(seg+1)-angle0;
      arclength = dangle*testSize;

      for q=1:size(W,1)           % quadrature loop
         pt=Q(q,1);               % quadrature point
         wt=W(q);                 % quadrature weight
         [nx, ny] = pol2cart(angle0+dangle*pt, 1);        %normal direction
         x0 = xn+ [nx ny]* testSize;                      % integration point
         N=mls_basis(mls_type,x0,pts,supportSize);        %meshless appr.

         %  weight = wt * arclength
         %  integrations come here

      end  % of quadrature loop
   end % of sub-division loop
end    % of node loop
clear n nn sctr nseg angle xn pts angle0 dangle arclength q pt wt x0 N;
```

values.

## 5 Numerical Examples

Several 2-D problems are solved to illustrate the effectiveness of the present method. The numerical results of the present method, as applied to carefully chosen problems in 2D elasto-statics, specifically (i) patch test, (ii) cantilever beam, (iii) curved beam bent by a force at the end, are discussed.

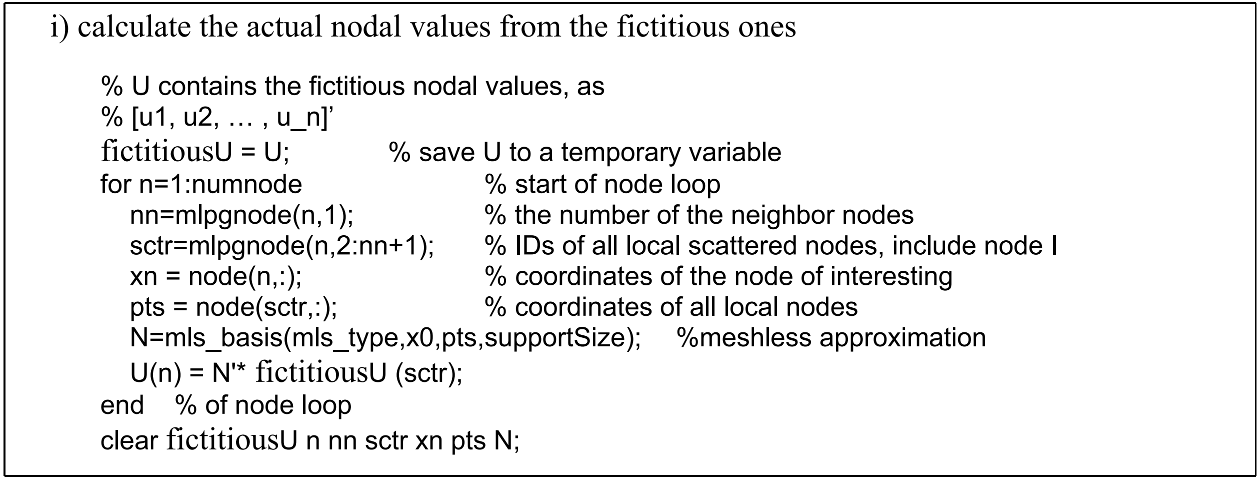
### *5.1 Cube under uniform tension*

The first example is that of a standard patch test, shown in Figure 3.

The material parameters are taken as $E = 1.0$, and $\nu = 0.25$. The nodal configuration contains 9 nodes. Two nodal configurations are used for the testing purpose: one is regular and another is irregular, as shown in Figure 3. In the patch tests, a uniform tensile stress is applied on the upper edge, and proper displacement constraints are applied to the lower edge.

The satisfaction of the patch test requires that the displacements are linear on the lateral faces, and are con-

**Table 3** : Sample codes for post processing



i) calculate the actual nodal values from the fictitious ones

```
% U contains the fictitious nodal values, as
% [u1, u2, … , u_n]'
fictitiousU = U;          % save U to a temporary variable
for n=1:numnode                     % start of node loop
   nn=mlpgnode(n,1);                % the number of the neighbor nodes
   sctr=mlpgnode(n,2:nn+1);         % IDs of all local scattered nodes, include node I
   xn = node(n,:);                  % coordinates of the node of interesting
   pts = node(sctr,:);              % coordinates of all local nodes
   N=mls_basis(mls_type,x0,pts,supportSize);    %meshless approximation
   U(n) = N'* fictitiousU (sctr);
end   % of node loop
clear fictitiousU n nn sctr xn pts N;
```

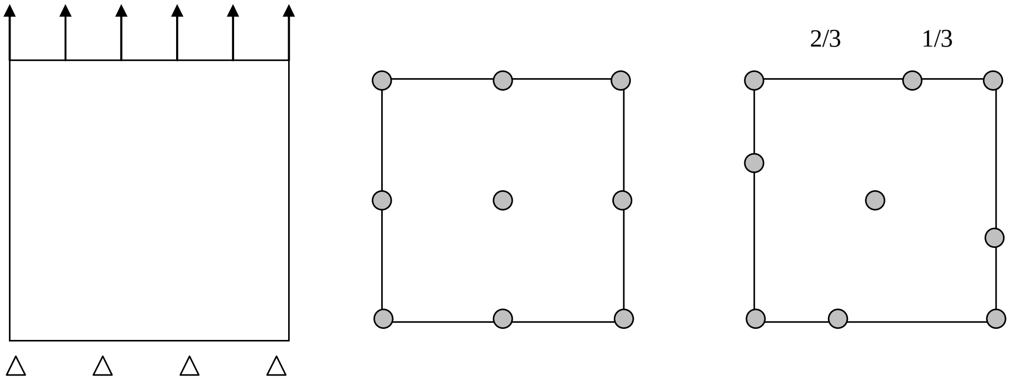


**Figure 3** : A cube under uniform tension, and two nodal configurations

stant on the upper edge; and the stresses are constant in the solution domain. It is found that the present method passes the patch tests with both the MLS and RBF approximations. The maximum numerical errors are limited by the computer for two nodal configurations.

### 5.2 *Cantilever beam*

The performances of the present MLPG formulations are also evaluated, using the problem of a cantilever beam under a transverse load, as shown in Figure 4, for which the following exact solution is given in Timoshenko and Goodier (1970):

$$u_x = -\frac{Py}{6\overline{E}I}\left[3x(2L-x)+(2+\overline{\upsilon})(y^2-c^2)\right]$$
$$u_y = \frac{P}{6\overline{E}I}\left[x^2(3L-x)+3\overline{\upsilon}(L-x)y^2+(4+5\overline{\upsilon})c^2x\right] \tag{26}$$

where the moment of inertia $I$ the beam is given as,

$$I = \frac{c^3}{3} \tag{27}$$

and

$$\overline{E} = \begin{cases} E \\ \frac{E}{(1-\upsilon)^2} \end{cases} \quad \overline{\upsilon} = \begin{cases} \upsilon & \text{for plane stress} \\ \frac{\upsilon}{(1-\upsilon)} & \text{for plane strain} \end{cases} \tag{28}$$

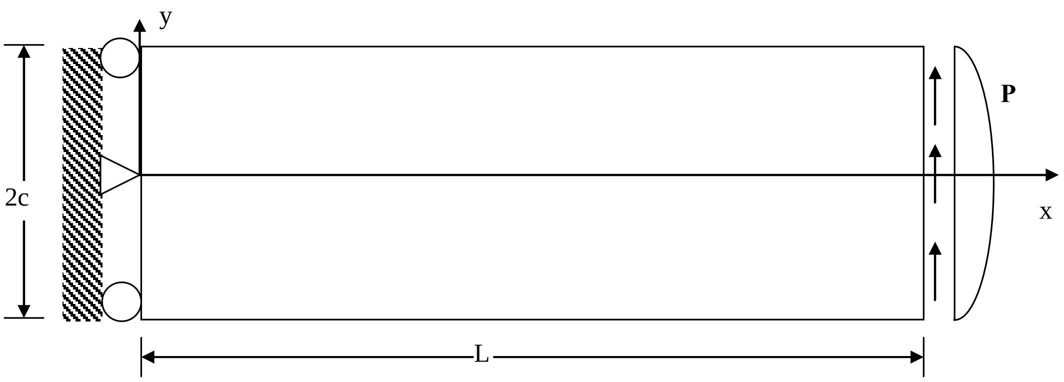


**Figure 4** : A cantilever beam under an end load

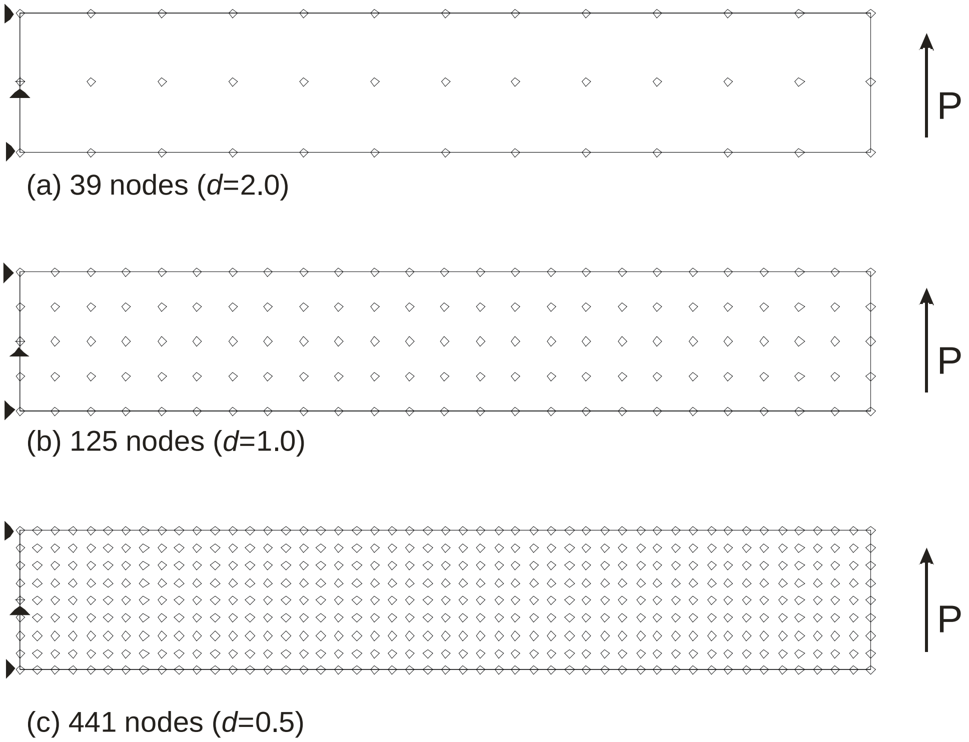


**Figure 5** : three nodal configurations for a cantilever beam

The corresponding stresses are

$$\sigma_x = -\frac{P}{I}(L-x)y$$
$$\sigma_y = 0$$
$$\sigma_{xy} = -\frac{P}{2I}(y^2-c^2) \tag{29}$$

The problem is solved for the plane stress case with $P = 1$, $E = 1$, $c = 2$, $L = 24$ and $\upsilon = 0.25$. Regular uniform nodal configurations with nodal distances, $d$, of 2.0, 1.0, and 0.5 are used, as shown in Figure 5. The numbers of nodes are 39, 125, and 441, respectively.

First, the problem is solved by using the MLS approximation, with a support size of $1.15d$ and a test size of $0.6d$. The vertical displacements are shown in Figure 6a, b, and c, for the three nodal configurations, respectively. They agree with the analytical solution very well. The relative error of the maximum displacement is 0.8% even when a very coarse nodal configuration with only

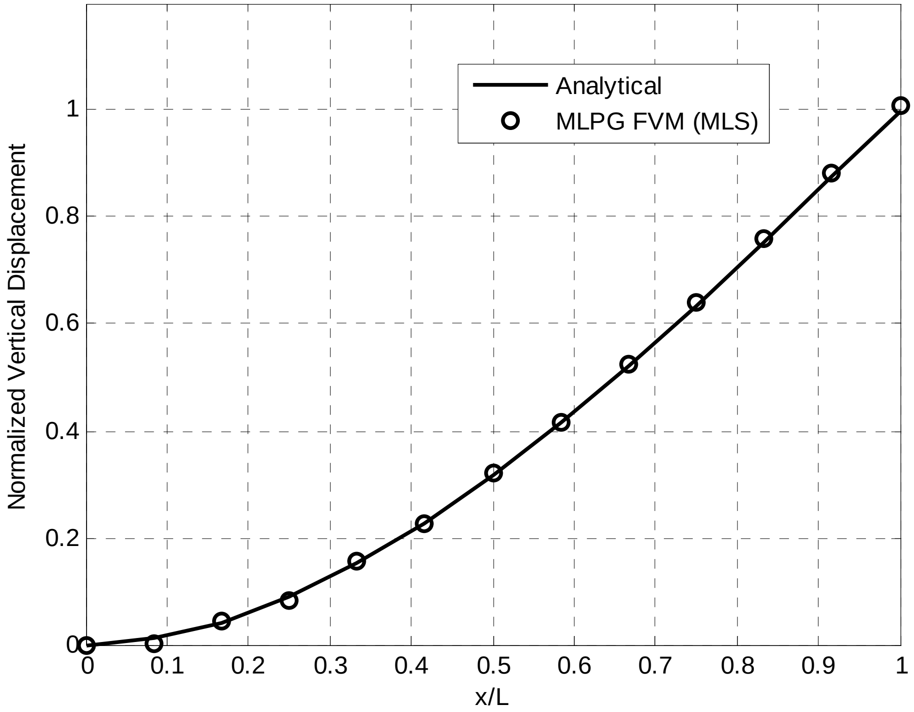


**Figure 6a** : Normalized vertical displacement of a cantilever beam under an end loading (39 nodes with nodal distance d = 2.0)

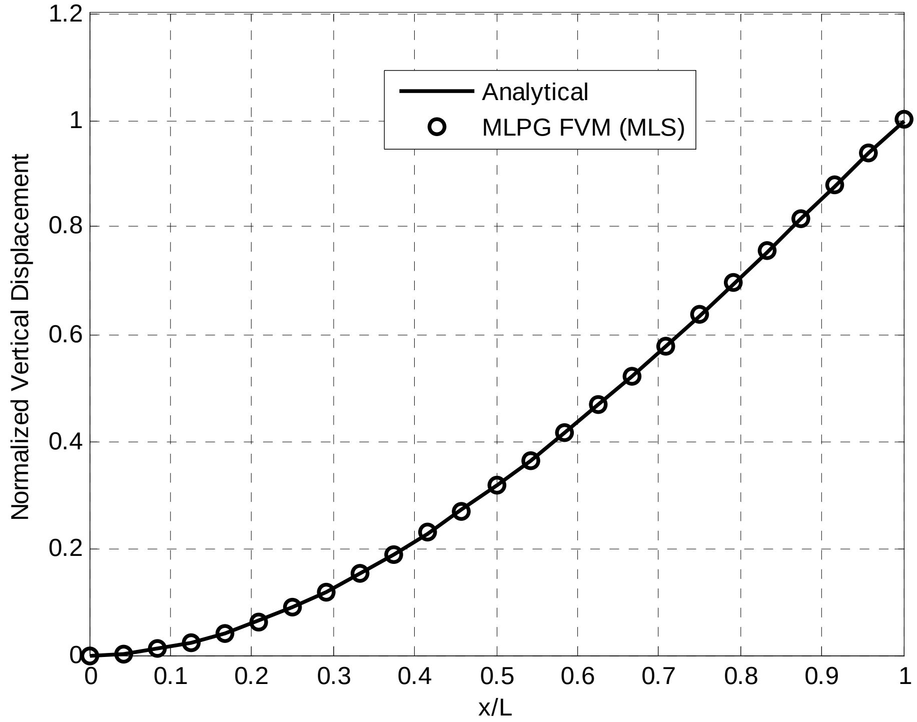


**Figure 6b** : Normalized vertical displacement of a cantilever beam under an end loading (125 nodes with nodal distance d = 1.0)

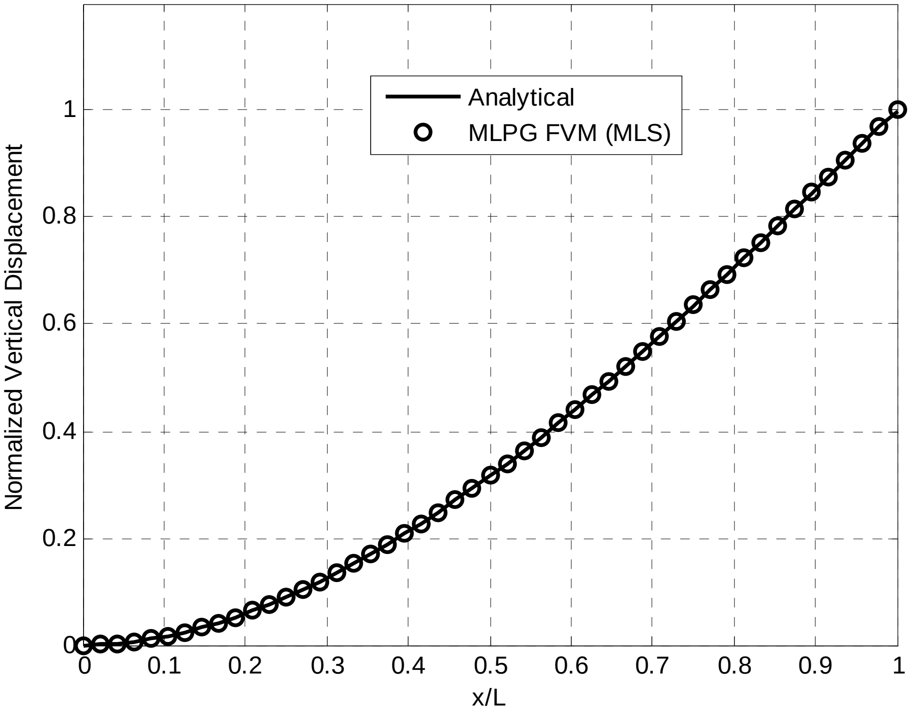


**Figure 6c** : Normalized vertical displacement of a cantilever beam under an end loading (441 nodes with nodal distance d = 0.5)

39 nodes is used, as shown in Figure 6a.

The effects of the approximation methods, the support size, and the test-domain size are studied for the present method (MLPG FVM). The approximations are the MLS with the first order polynomials (labeled as MLS1), the MLS with second order polynomials (labeled as MLS2), and the RBF augmented with the first order polynomials (labeled as RBF), respectively. The support size and the test size are related to the nodal distance, $d$. Normally, the ratio of the support size is greater than 1.0. It should be greater than 2.0 for the MLS2, to make sure that there are enough points to support the nodes on the global boundary. The ratio of the test-domain size is chosen to be less than 1.0 in the present study. They are detailed in the following sub-sections.

### *5.2.1 Effects of the test-domain size*

The local sub-domain is one of the key concepts for the MLPG approach. As over-lapping sub-domains are used, the test-domain size (or the size of the sub-domain) affects the accuracy of the solution and the efficiency of the method. It is very different from the non-over-lapping methods, in which the background cells are required to partition the solution domain. In the present study, the test-domain size is chosen to be proportional to the nodal distance, $d$. Theoretically, the ratio is very flexible. In practice, it is chosen to be less than 1.0 to ensure that the local sub-domains of the internal nodes are entirely within the solution domain, without being intersected by the global boundary. It is chosen to be greater than 0.5 to ensure the sub-domains are over-lapping. In the present study, four ratios are used as 0.5, 0.6, 0.7 and 0.8. The support size is fixed as $1.15d$ for the MLS1 and the RBF, and $2.5d$ for the MLS2.

As all methods give the reasonable results, the relative errors of the maximum displacements are used to examine the effects of the test-domain size. Three nodal configurations are used to examine the displacement errors, as shown Figure 7a,b, and c, respectively.

The MLS1 gives the most accurate results among the three approximations. The maximum error is 0.8% even only 39 nodes are used. It can be seen that good accuracy is obtained when the test-domain size is $0.6d$. It is noticeable that the accuracy is less sensitive to the test-domain size from $0.5 \sim 0.7d$, as the sub-domains are slightly over-lapping. However, the RBF approximation

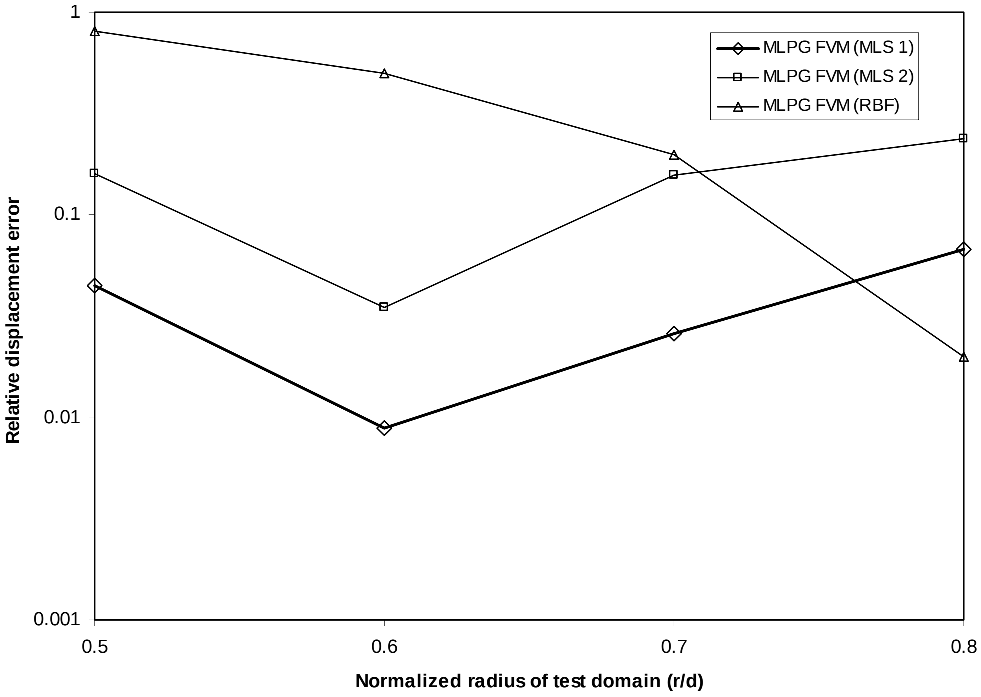


**Figure 7a** : Influence of the test-domain size in a cantilever beam under an end load (39 nodes with nodal distance d = 2.0)

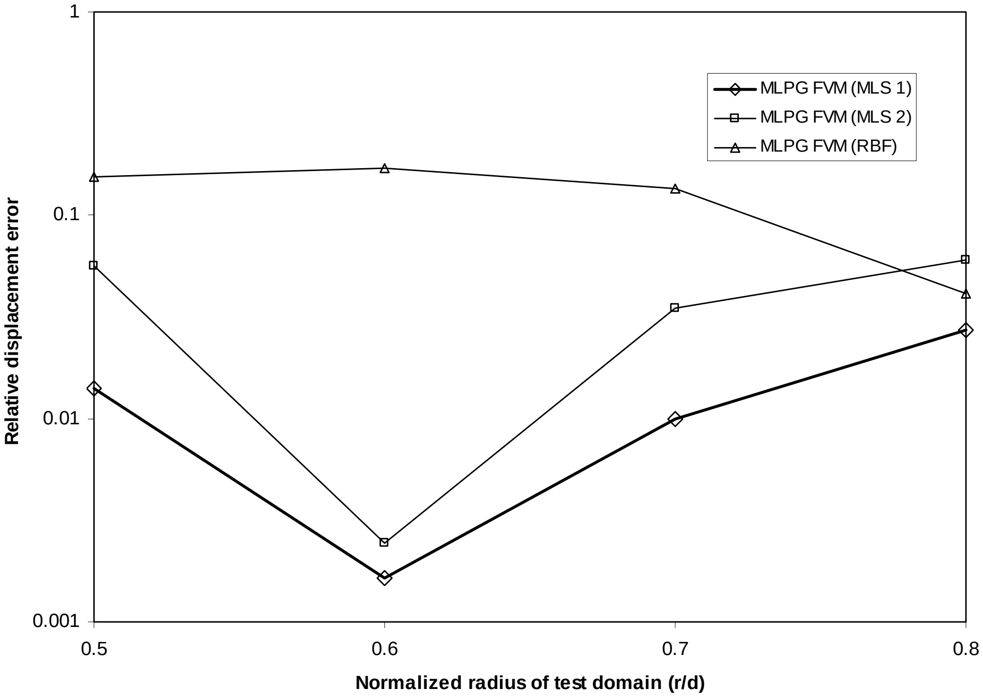


**Figure 7b** : Influence of the test-domain size in a cantilever beam under an end load (125 nodes with nodal distance d = 1.0)

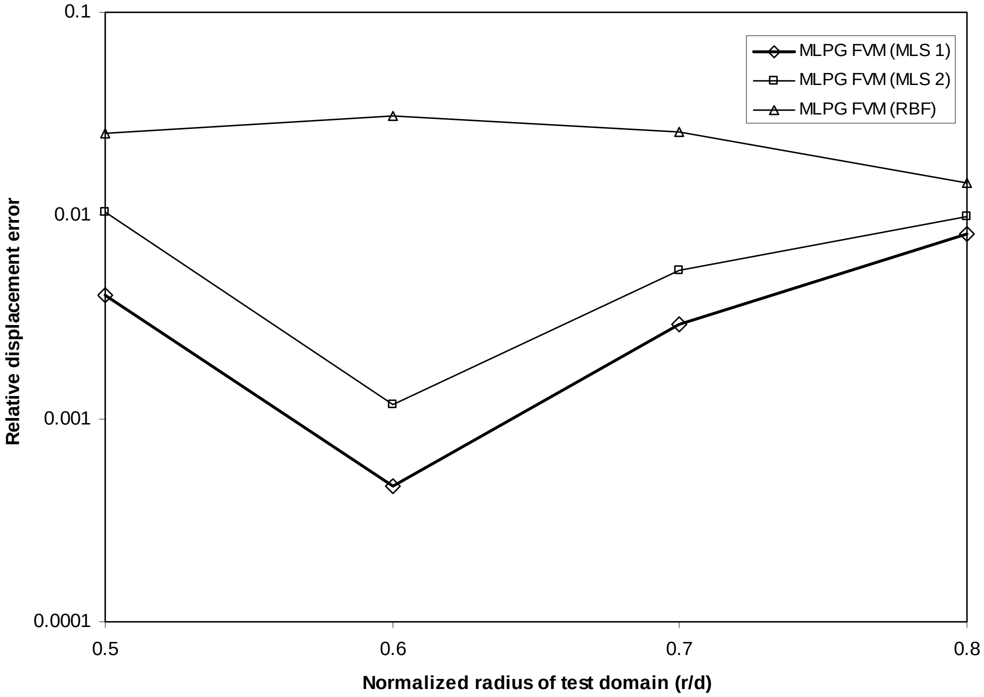


**Figure 7c** : Influence of the test-domain size in a cantilever beam under an end load (441 nodes with nodal distance d =0.5)

becomes more stable when a larger test-domain size is used. It may be due to the non-continuous shape function.

### 5.2.2 Effects of the support size

The support size (or the size of the influence domain) is a very important in meshless methods. It is related to both the accuracy of the solution, as well as the computational efficiency. For a smaller size, the meshless approximation algorithms may be singular and the shape function can not be constructed because of too few nodes. The support size is also chosen to be proportional to the nodal distance. In the present study, four ratios are used for the MLS1 and the RBF, as 1.15, 1.25, 1.5, and 1.8, and three for the MLS2 are 2.05, 2.5 and 2.8. The test size is chosen as $0.6d$.

The relative errors of the maximum displacements are shown in Figure 8.

Again, the MLS1 gives the most accurate results among the three approximations. In addition, the results are less sensitive to the support size when the MLS1 is used with a smaller support size. It makes the present method very efficient by speeding up the MLS approximation. The MLS2 is also less sensitive to the support size. However, the RBF gives better results when the suitable support size is used.

### 5.2.3 Convergence rate

The convergence rate is studied with three nodal configurations. The test-domain size is chosen to be $0.6d$, and the support size is $1.15d$ for the MLS1 and RBF and $2.5d$ for the MLS2. The relative errors of the maximum displacement are used for showing the convergence rate in Figure 9.

The results clearly show that a stable convergence rate is obtained for the present MLPG finite volume method with all three approximations.

## 5.3 Curved beam

As the last example, the problem of a curved beam under an end load is used to evaluate the present method. The problem is shown in Figure 10, for which the following exact solution is given in Timoshenko and Good-

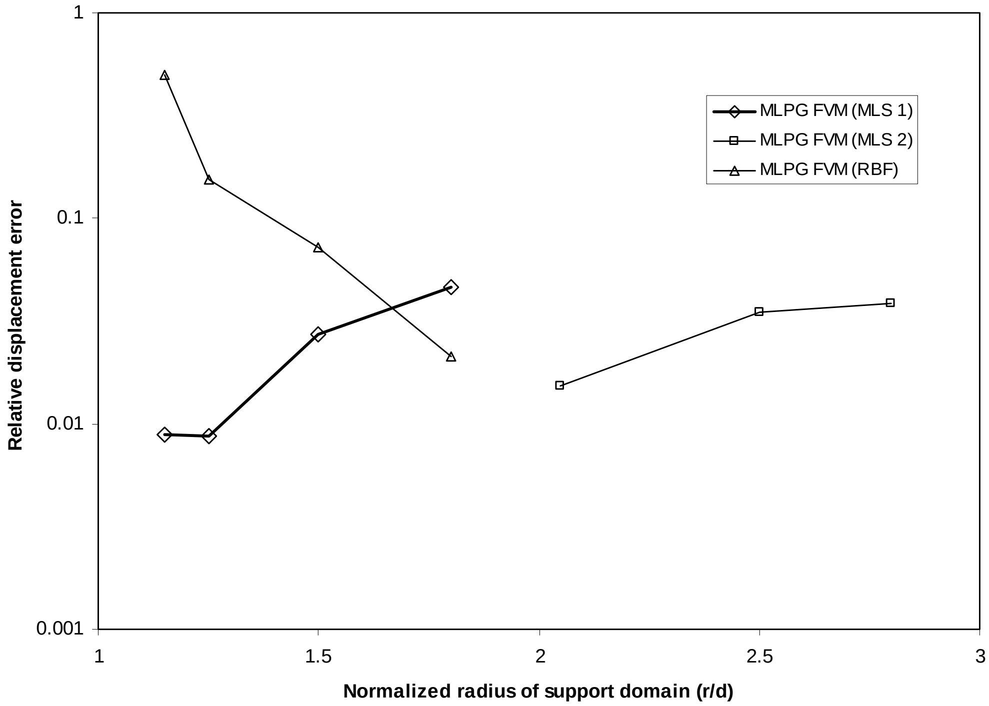


**Figure 8a** : Influence of the support size in a cantilever beam under an end load (39 nodes with nodal distance d = 2.0)

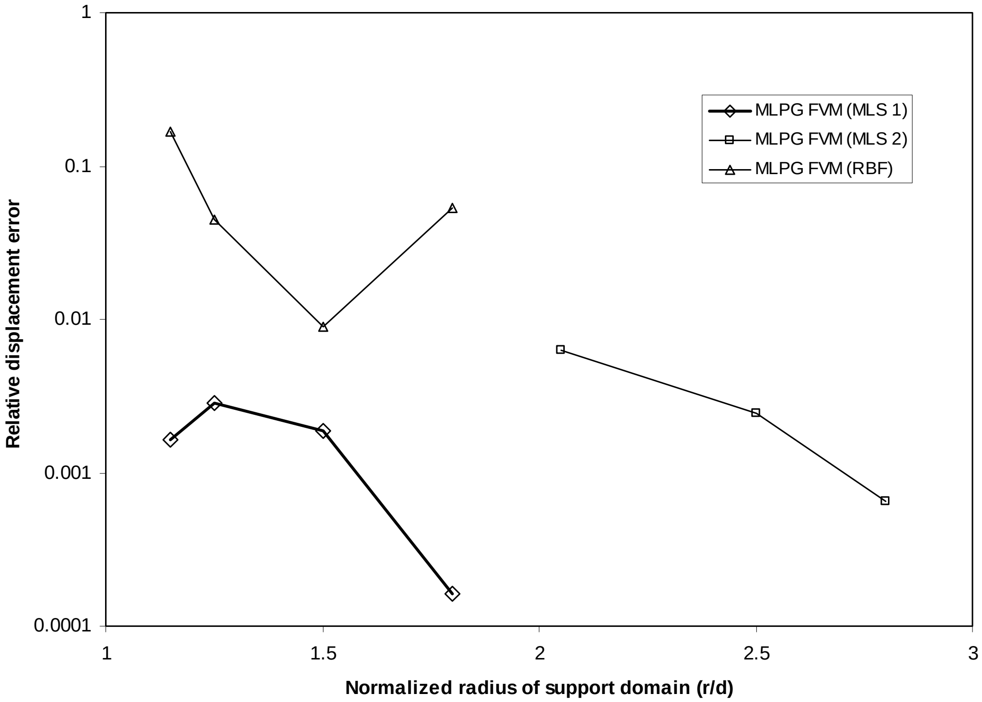


**Figure 8b** : Influence of the support size in a cantilever beam under an end load (125 nodes with nodal distance d = 1.0)

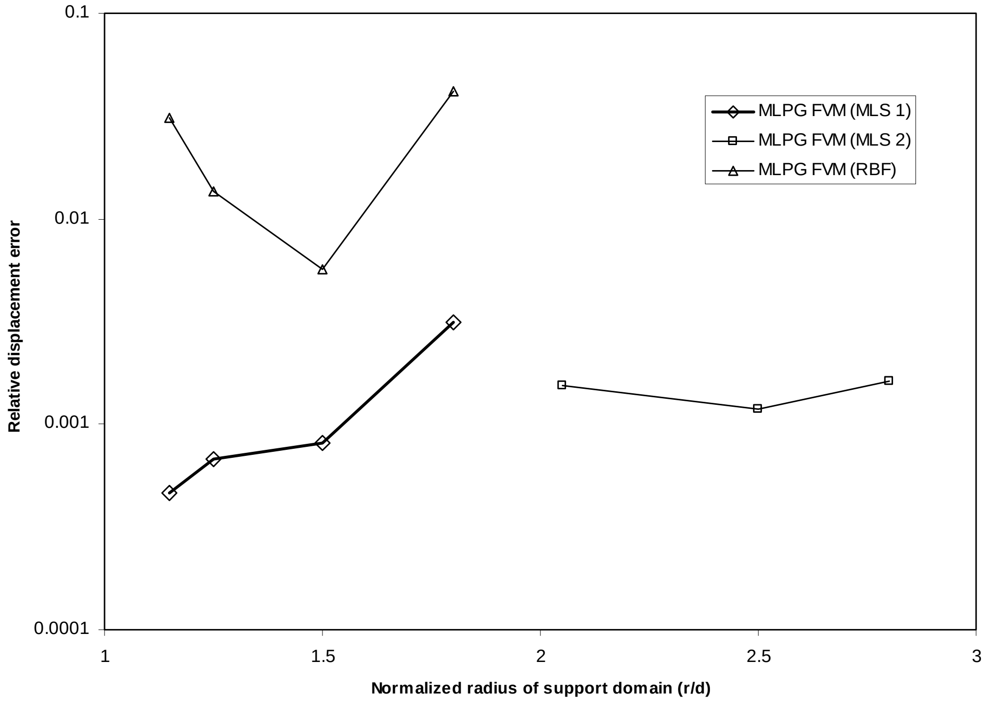


**Figure 8c** : Influence of the support size in a cantilever beam under an end load (441 nodes with nodal distance d =0.5)

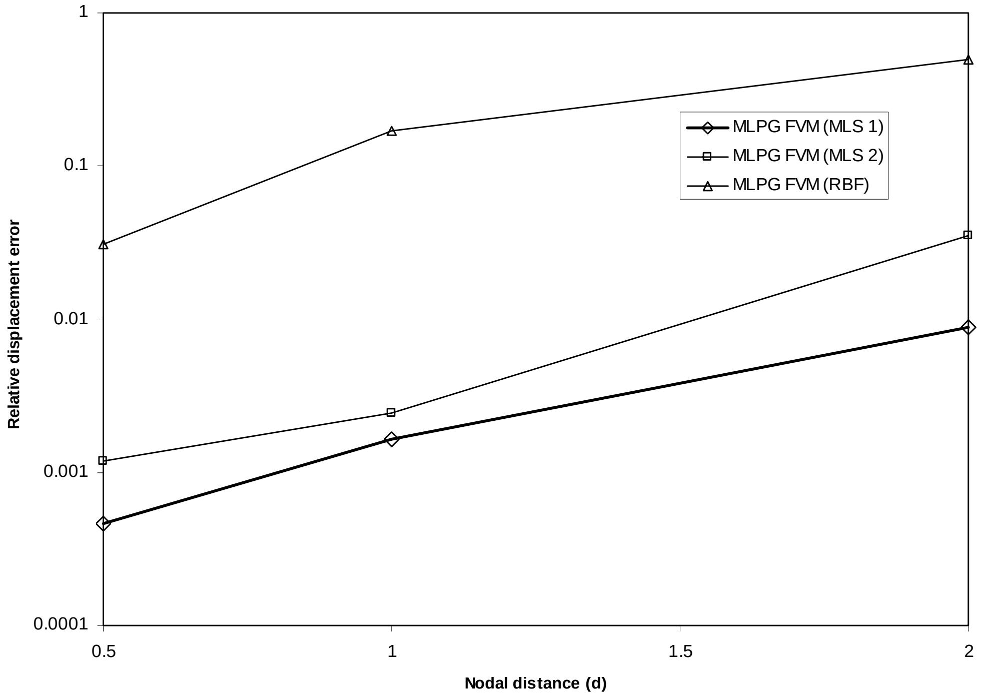


**Figure 9** : Convergence rate in a cantilever beam under an end load

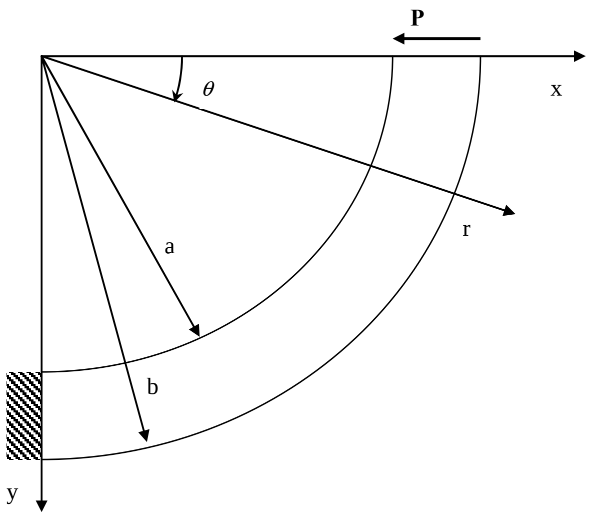


**Figure 10** : A curved bar with an end load

ier (1970):

$$
\begin{aligned}
u_r = & \frac{P}{E}[-2D\theta\cos\theta + \sin\theta(D(1-\bar{\upsilon})\log r \\
& + A(1-3\bar{\upsilon})r^2 + \frac{B(1+\bar{\upsilon})}{r^2}) \\
& + K\sin\theta + L\cos\theta] \\
u_\theta = & \frac{P}{E}[2D\theta\sin\theta - \cos\theta(-D(1-\bar{\upsilon})\log r \\
& + A(5+\bar{\upsilon})r^2 + \frac{B(1+\bar{\upsilon})}{r^2}) \\
& + D(1+\bar{\upsilon})\cos\theta + K\cos\theta + L\sin\theta]
\end{aligned}
\qquad (30)
$$

where the constants are given as,

$$
\begin{aligned}
& N = a^2 - b^2 + (a^2+b^2)\log\frac{b}{a} \\
& A = \frac{1}{2N} \quad B = -\frac{a^2b^2}{2N} \\
& D = -\frac{a^2+b^2}{N} \quad L = D\pi \\
& K = -(D(1-\bar{\upsilon})\log r_0 + A(1-3\bar{\upsilon})r_0^2 + \frac{B(1+\bar{\upsilon})}{r_0^2}) \\
& r_0 = \frac{a+b}{2}
\end{aligned}
\qquad (31)
$$

The corresponding stresses are

$$
\begin{aligned}
\sigma_r &= P(2Ar - \frac{2B}{r^3} + \frac{D}{r})\sin\theta \\
\sigma_\theta &= P(6Ar + \frac{2B}{r^3} + \frac{D}{r})\sin\theta \\
\sigma_{r\theta} &= -P(2Ar - \frac{2B}{r^3} + \frac{D}{r})\cos\theta
\end{aligned}
\qquad (32)
$$

The problem is solved for the plane stress case with $P = 1$, $E = 1$, $a = 13$, $b = 17$ and $\upsilon = 0.25$. Regular uniform nodal configurations with nodal distances, $d$, of 2.0, 1.0, and 0.5 are used, as shown in Figure 11. The numbers of nodes are 39, 125, and 441, respectively.

The displacement and stress fields are more complicated than those in the case of a straight beam, with many non-polynomial terms. However, the first order MLS approximation is still used to solve this problem, with a support size of $1.25d$ and a test-domain size of $0.6d$. The horizontal and vertical displacements are shown in Figure 12a, b, and c, for the three nodal configurations, respectively. They agree with the analytical solution very well.

The relative errors of the maximum displacements are less than 2% even when only 39 nodes are used, as shown in Figure 12a, and they are reduced to less than 0.08% when 441 nodes are used. The influence of the test domain size is also studied in this problem, as shown in Figure 13.

Again, it confirms that the MLS approximation is not

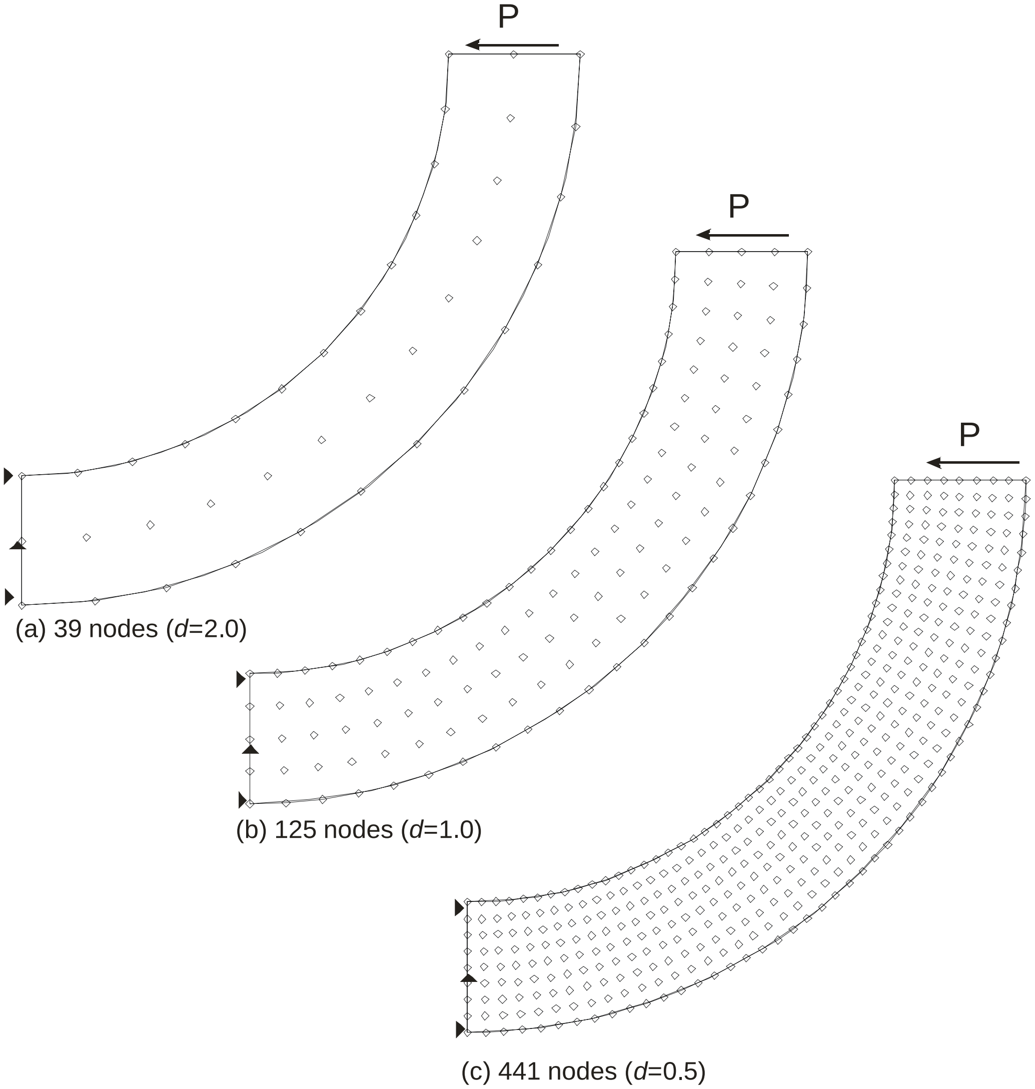


**Figure 11** : three nodal configurations for a curved cantilever beam

sensitive to the test-domain size when the local subdomains are slightly over-lapped. The present method also gives a fast convergence rate by using three nodal configurations, as shown Figure 14.

## 6 Closure

A new Meshless Finite Volume Method (MFVM) is developed through a new MLPG ”Mixed” approach. The differentiation of the shape function is eliminated, by interpolating the strains directly, as independent variables in the local weak form. It reduces the continuity-requirement on the trial function by one-order, and a smaller support size can be used in the meshless approximations with a lower-order polynomial basis. The computational efficiency is improved, due to these two key aspects in the newly developed meshless method. The numerical results demonstrate the accuracy of the present methods for problems whose analytical solutions contain both polynomial and non-polynomial basis. Convergence studies in the numerical examples show that the present method possesses an excellent rate of convergence.

**Acknowledgement:** This work was supported by the Army Research Laboratory, and the ARO, under the cognizance of Drs. R. Namburu, and B. LaMattina. The first two authors gratefully acknowledge this support.

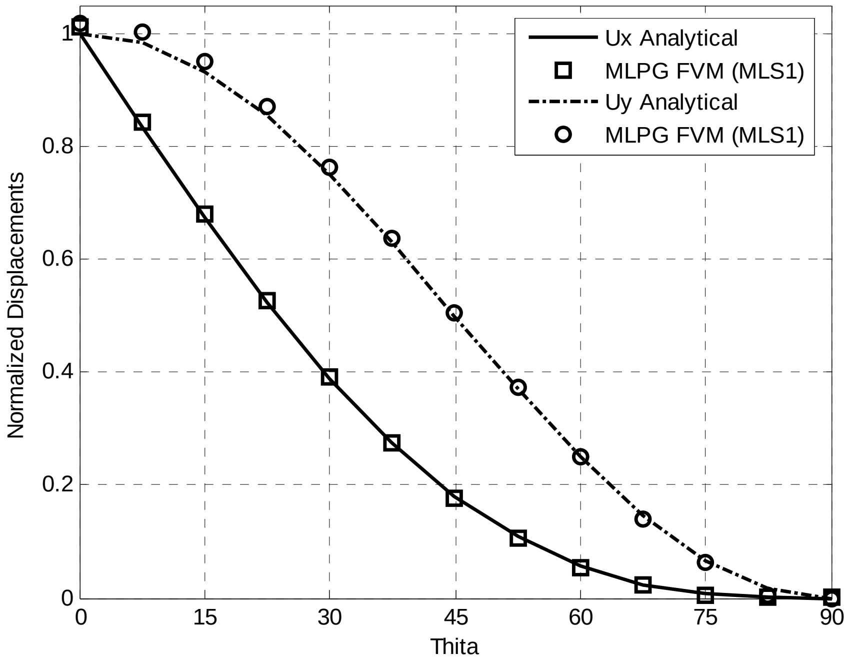


**Figure 12a** : Normalized displacements of a curved cantilever beam under an end loading (39 nodes with nodal distance d = 2.0)

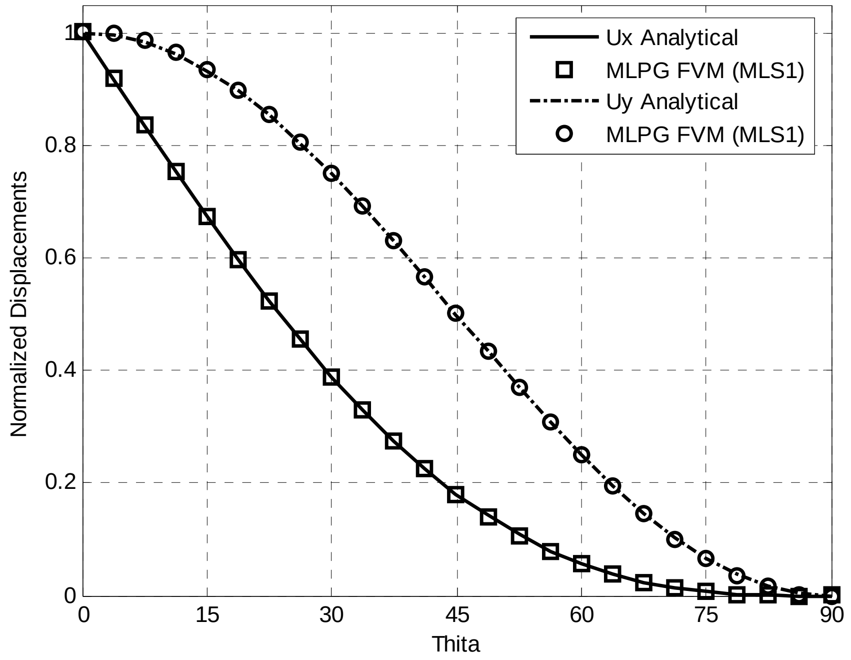


**Figure 12b** : Normalized displacements of a curved cantilever beam under an end loading (125 nodes with nodal distance d = 1.0)

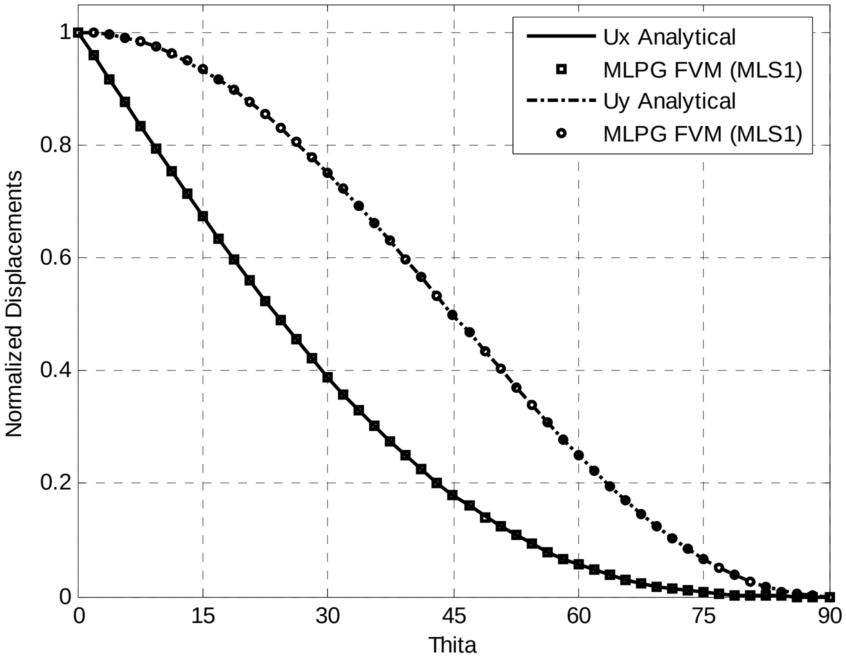


**Figure 12c** : Normalized displacements of a curved cantilever beam under an end loading (441 nodes with nodal distance d = 0.5)

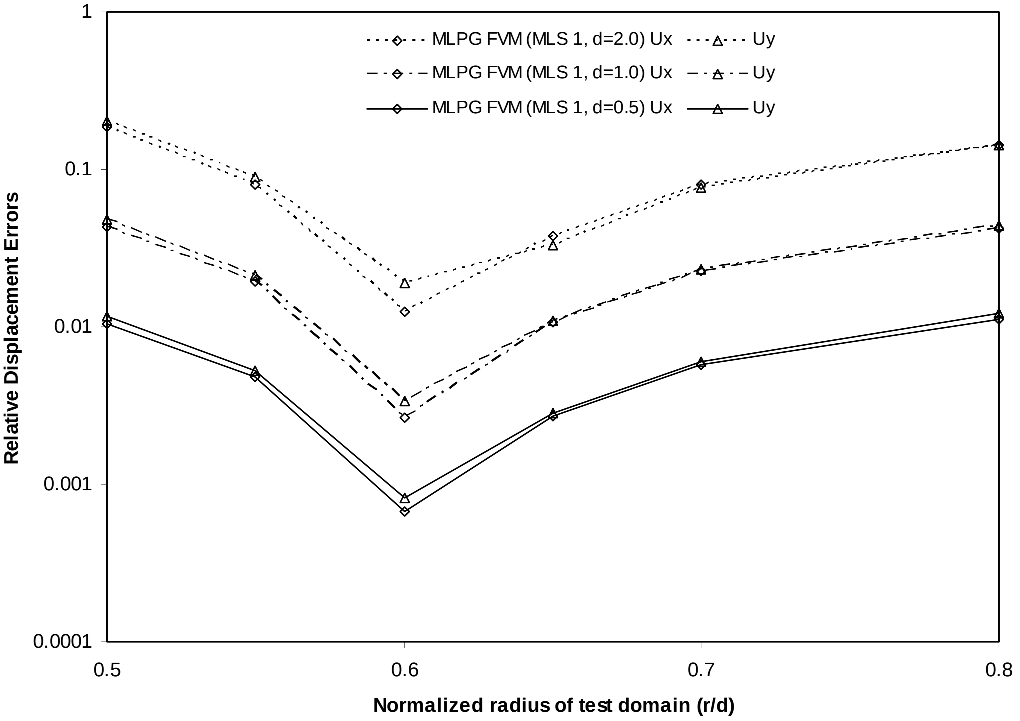


**Figure 13** : Influence of the test-domain size in a curved cantilever beam under an end load

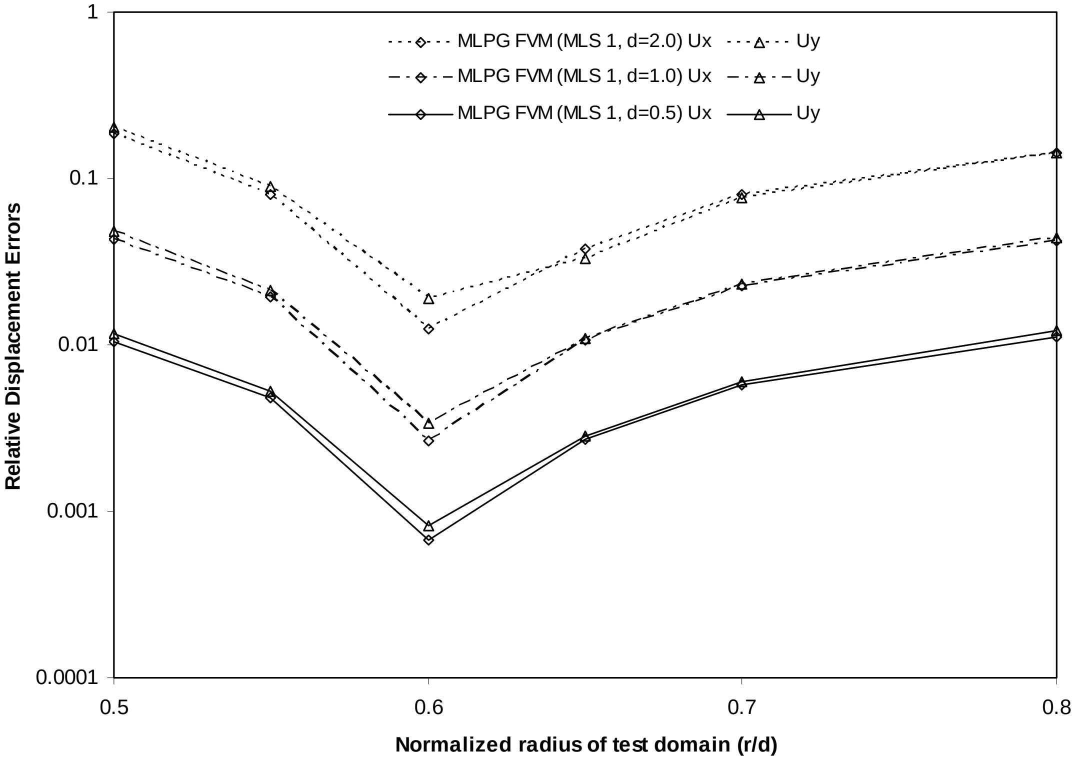


**Figure 14** : Convergence rate in a curved cantilever beam under an end load

## References

**Atluri, S. N.** (2004): *The Meshless Local Petrov-Galerkin ( MLPG) Method for Domain & Boundary Discretizations*, Tech Science Press, 680 pages.

**Atluri, S. N.; Han, Z. D.; Shen, S.** (2003): Meshless Local Patrov-Galerkin (MLPG) approaches for weakly-singular traction & displacement boundary integral equations, CMES: *Computer Modeling in Engineering & Sciences*, vol. 4, no. 5, pp. 507-517.

**Atluri, S. N.; Kim, H. G.; Cho, J. Y.** (1999): A Critical Assessment of the Truly Meshless Local Petrov Galerkin (MLPG) and Local Boundary Integral Equation (LBIE) Methods, *Computational Mechanics*, 24:(5), pp. 348-372.

**Atluri, S. N.; Shen, S.** (2002): The meshless local Petrov-Galerkin (MLPG) method: A simple & less-costly alternative to the finite element and boundary element methods. CMES: *Computer Modeling in Engineering & Sciences*, vol. 3, no. 1, pp. 11-52

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

**Dilts, G. A.** (1999): Moving-Least-Squares-Particle Hydrodynamics - I. Consistency and Stability, *International Journal for Numerical Methods in Engineering*, vol. 44, pp. 1115-1155.

**Gingold, R. A.; Monaghan, J. J.** (1977): Smoothed particle hydrodynamics, theory and application to non-spherical stars, *Mon. Not. Roy. Astr. Soc.*, vol. 181, pp. 375-389.

**Golberg, M. A.; Chen, C. S.; Bowman, H.** (1999): Some recent results and proposals for the use of radial basis functions in the BEM, *Engineering Analysis with Boundary Elements*, vol. 23, pp. 285-296.

**Han. Z. D.; Atluri, S. N.** (2003a): On Simple Formulations of Weakly-Singular Traction & Displacement BIE, and Their Solutions through Petrov-Galerkin Approaches, CMES: *Computer Modeling in Engineering & Sciences*, vol. 4 no. 1, pp. 5-20.

**Han. Z. D.; Atluri, S. N.** (2003b): Truly Meshless Local Petrov-Galerkin (MLPG) Solutions of Traction & Displacement BIEs, CMES: *Computer Modeling in Engineering & Sciences*, vol. 4 no. 6, pp. 665-678.

**Han. Z. D.; Atluri, S. N.** (2004a): Meshless Local Petrov-Galerkin (MLPG) approaches for solving 3D Problems in elasto-statics, CMES: *Computer Modeling in Engineering & Sciences*, vol. 6 no. 2, pp. 169-188.

**Han. Z. D.; Atluri, S. N.** (2004b): A Meshless Local Petrov-Galerkin (MLPG) Approach for 3-Dimensional Elasto-dynamics, CMC: *Computers, Materials & Continua*, vol. 1 no. 2, pp. 129-140.

**Johnson, G. R.; Beissel, S. R.** (1996): Normalized smoothing functions for SPH impact computations, *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 2725-2741.

**Johnson, G.R.; Stryk, R. A.; Beissel, S.R.; Holmquist, T.J.** (2002): An algorithm to automatically convert distorted finite elements into meshless particles during dynamic deformation. *Int. J. Impact Eng.*, vol. 27, pp. 997-1013.

**Li, Q.; Shen, S.; Han, Z. D.; Atluri, S. N.** (2003): Application of Meshless Local Petrov-Galerkin (MLPG) to Problems with Singularities, and Material Discontinuities, in 3-D Elasticity, CMES: *Computer Modeling in Engineering & Sciences*, vol. 4 no. 5, pp. 567-581.

**Lin, H.; Atluri, S. N.** (2001): The Meshless Local Petrov-Galerkin (MLPG) Method for Solving Incompressible Navier-Stokes Equations CMES: *Computer Modeling in Engineering & Sciences*, vol. 2, no. 2, pp. 117-142.

**Lucy** (1977): A numerical approach to the testing of session hypothesis,*Astronomical Journal*, vol. 82, pp. 1013 –1024.

**Sellountos, E. J.; Polyzos, D.** (2003): A MLPG (LBIE) method for solving frequency domain elastic problems, *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, no. 6, pp. 619-636

**Timoshenko, S. P.; Goodier, J. N.** (1976): Theory of Elasticity, $3^{rd}$ edition, McGraw Hill.