CMES, vol.4, no.1, pp.141-159, 2003

Further Developments in the MLPG Method for Beam Problems

I. S. Raju[1] **and D. R. Phillips**[2]

Abstract: An accurate and yet simple Meshless Local Petrov-Galerkin (MLPG) formulation for analyzing beam problems is presented. In the formulation, simple weight functions are chosen as test functions as in the conventional MLPG method. Linear test functions are also chosen, leading to a variation of the MLPG method that is computationally efficient compared to the conventional implementation. The MLPG method is evaluated by applying the formulation to a variety of patch tests, thin beam problems, and problems with load discontinuities. The formulation successfully reproduces exact solutions to machine accuracy when higher order power and spline functions are chosen as test functions or when the linear test function is used, and when constructing the trial functions, the order of the basis function is properly balanced by the order of the weight function. For mixed boundary value problems, deflections, slopes, moments, and shear forces are calculated to the same accuracy by the MLPG method without the use of elaborate post-processing techniques. Problems with load discontinuities require special care – when a reasonable number of nodes are used, the method yields very accurate results.

1 Introduction

Meshless methods are developed to overcome some of the disadvantages of the Finite Element Method (FEM) such as discontinuous secondary variables across inter-element boundaries and the need for remeshing in large deformation problems. Recent literature shows extensive research on meshless methods and in particular the Meshless Local Petrov-Galerkin (MLPG) method. The majority of literature published to date on the MLPG method presents variations of the method for C^0 problems (see the monograph by Atluri and Shen, 2002a). However, a comparatively limited amount of work is reported on the more complicated C^1 problems (Krysl and Belytschko, 1995, Donning and Liu, 1998, Atluri *et al.*, 1999, and Gu and Liu, 2001). Atluri *et al.* (1999) presented an analysis of thin beam problems using a Galerkin implementation of the MLPG method. In their (Atluri *et al.*) paper, a generalized moving least squares (GMLS) approximation was used to construct the trial functions, and the test functions were chosen from the same space. While accurate results for primary and secondary variables were obtained with these choices, special procedures were needed to integrate the weak form accurately.

The purpose of this paper is to explore two MLPG formulations for beam problems – the conventional version of the MLPG method (i.e., a true meshless Petrov-Galerkin implementation, originally presented by Atluri and Zhu in 1998) and a modification to the MLPG method that is computationally less expensive than the conventional version of the method. In the conventional MLPG method, test functions are chosen from a different space than the trial functions and involve simple weight functions. In the computationally efficient and less expensive modification, special test functions are chosen that eliminate the domain integrals. This formulation is evaluated by comparing the results with those obtained using the conventional MLPG method. This paper also presents extensive studies on various parameters of the method for beam problems.

The outline of the paper is as follows. First, the local weak form of the governing differential equation is derived in a general sense, and a system of algebraic equations is developed from this local weak form. Next, the generalized moving least squares interpolation scheme used to construct the trial functions is described. Then, the test functions chosen in the conventional MLPG method are presented. The computationally efficient modification to the MLPG method is described next and

[1] Senior Technologist, Structures and Materials Competency
NASA Langley Research Center
Hampton, Virginia 23681, U.S.A.

[2] Aerospace Engineer, Lockheed Martin Space Operations
NASA Langley Research Center
Hampton, Virginia 23681, U.S.A.

compared to the conventional MLPG method. Several patch test problems are used to validate the method, and specific problem parameters are discussed. Then, various mixed boundary value problems are presented to test the method. Finally, the method is applied to problems with load discontinuities and continuous beam problems.

2 Local Weak Form for Euler-Bernoulli Beam Problems

The governing equation for an Euler-Bernoulli beam is

$$EI\frac{d^4w}{dx^4}=f \text{ in domain } \Omega \quad (0\leq x\leq L) \text{ with boundary } \Gamma, \tag{1}$$

where w is the transverse displacement, L is the length and EI is the flexural rigidity of the beam, and f is the distributed load on the beam. The boundary conditions at $x=0$ and $x=L$ can have several combinations. The essential boundary conditions (EBCs) are of the form

$$w=\tilde{w} \quad \text{on } \Gamma_w \quad \text{and}$$

$$\frac{dw}{dx}=\tilde{\theta} \quad \text{on } \Gamma_\theta, \tag{2}$$

and the natural boundary conditions (NBCs) are of the form

$$V=\tilde{V} \quad \text{on } \Gamma_V \quad \text{and}$$

$$M=\tilde{M} \quad \text{on } \Gamma_M, \tag{3}$$

where V and M are the shear force and bending moment, respectively, and are related to the deflection, w, as

$$V=-EI\frac{d^3w}{dx^3} \quad \text{and} \quad M=EI\frac{d^2w}{dx^2}, \tag{4}$$

and Γ_w, Γ_θ, Γ_V, and Γ_M denote the boundary points where deflection (w), slope (θ), shear (V), and moment (M) are prescribed, respectively. Note that the prescriptions of $\tilde{w}$ and $\tilde{V}$ and $\tilde{\theta}$ and $\tilde{M}$ are mutually disjoint, i.e., when $w=\tilde{w}$ is prescribed, the shear force V becomes the corresponding reaction, and when $\theta=\tilde{\theta}$ is prescribed, the moment M becomes the corresponding reaction.

The classical weighted residual form of the governing differential equation for fourth order problems is obtained by multiplying the residual by a weight function, v, integrating over the whole domain, and setting the integral to zero:

$$0=\int_\Omega\left(EI\frac{d^4w}{dx^4}-f\right)vdx. \tag{5}$$

The weak form of the weighted residual equation is set up by transferring the differentiation from the primary variable, w, to the weight function, v. This is achieved by integrating by parts twice, yielding

$$0=EI\int_\Omega\frac{d^2w}{dx^2}\frac{d^2v}{dx^2}dx-\int_\Omega f\;v\;dx$$
$$+n_x\left[EI\frac{d^3w}{dx^3}v\right]_\Gamma-n_x\left[EI\frac{d^2w}{dx^2}\frac{dv}{dx}\right]_\Gamma, \tag{6}$$

where $n_x[v\cdot EI(d^3w/dx^3)]_\Gamma$ and $n_x[(dv/dx)\cdot EI(d^2w/dx^2)]_\Gamma$ are introduced as boundary terms and n_x is the direction cosine of the unit outward drawn normal to Ω with respect to the x-axis. The n_x thus takes values ±1 in 1-D problems. In this paper, the essential boundary conditions are enforced by a penalty method (Atluri *et al.*, 1999). The penalty terms are written as

$$\alpha_w\left[(w-\tilde{w})\;v\right]_{\Gamma_w} \tag{7}$$

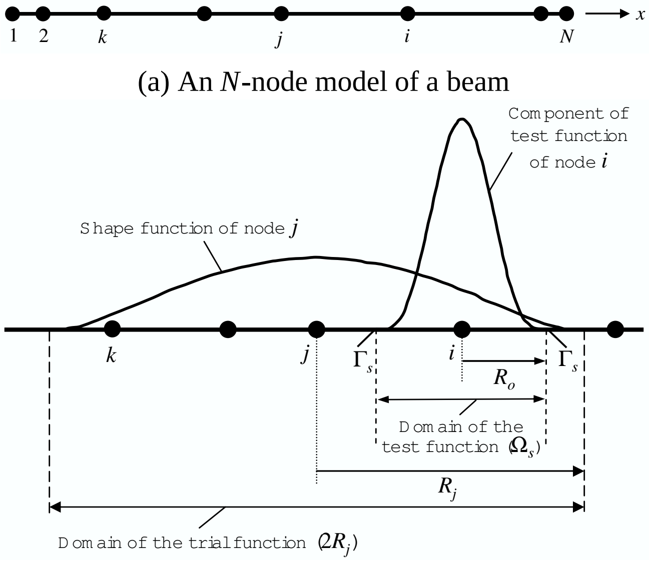
and

$$\alpha_\theta\left[\left(\frac{dw}{dx}-\tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_\theta},$$

where α_w and α_θ are the penalty parameters to enforce the deflection and slope boundary conditions, respectively. Thus, including the penalty terms, Eq. (6) is written as

$$0=EI\int_\Omega\frac{d^2w}{dx^2}\frac{d^2v}{dx^2}dx-\int_\Omega f\;v\;dx+\alpha_w\left[(w-\tilde{w})v\right]_{\Gamma_w}$$
$$+\alpha_\theta\left[\left(\frac{dw}{dx}-\tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_\theta}+n_x\left[EI\frac{d^3w}{dx^3}v\right]_\Gamma$$
$$-n_x\left[EI\frac{d^2w}{dx^2}\frac{dv}{dx}\right]_\Gamma. \tag{8}$$

In Eq. (8), called the weak form of the governing differential equation, the chosen approximations for the variable w are called the trial functions, and the weight functions, v, are now called the test functions. To make the current implementation a Petrov-Galerkin method, the test functions are chosen independently from the trial functions.


x
1
2
k
j
i
N
Component of test function of node i
Shape function of node j
k
j
Γs
i
Γs
Ro
Domain of the test function (Ωs)
Rj
Domain of the trial function (2Rj)

(a) An N-node model of a beam

(b) Components of the trial and test functions

Figure 1 : Comparison of the domains of the trial and test functions

Consider an N-node model of a beam as shown in Figure 1a. The solution to the beam problem for a given loading and set of boundary conditions is sought as w and θ at each of the nodes in the model. To achieve this, first the trial functions are chosen for w and the test functions are chosen for v, and then the weak form of Eq. (8) is applied to each of the nodes in the model to yield a system of algebraic equations in terms of the unknown primary variables, w and θ, at each of the nodes in the model. Figure 1b illustrates the component of the trial function (the shape function) at node j and the component of the test function at node i. Unlike in the finite element method, the trial functions at node j are diffused and extend over a domain $(x_j - R_j)$ to $(x_j + R_j)$. Similarly, the test functions span over the domain $(x_i - R_o)$ to $(x_i + R_o)$. The variables R_o and R_j are user-defined parameters. As the test functions are defined over a local region, $(x_i - R_o)$ to $(x_i + R_o)$, i.e., as they are chosen to be zero for $x < (x_i - R_o)$ and $x > (x_i + R_o)$, the integrations over Ω in Eq. (8) reduce to integrations over a local sub-domain Ω_s, defined by $(x_i - R_o)$ to $(x_i + R_o)$, with boundary Γ_s (see Figure 1b), and the weak form reduces to

$$
\begin{aligned}
0 = EI \int_{\Omega_s} \frac{d^2w}{dx^2}\frac{d^2v}{dx^2}dx - \int_{\Omega_s} f\,vdx \\
+ \alpha_w \left[(w-\tilde{w})\ v\right]_{\Gamma_{sw}} + \alpha_\theta \left[\left(\frac{dw}{dx} - \tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_{s\theta}} \\
+ n_x \left[EI\frac{d^3w}{dx^3}v\right]_{\Gamma_s} - n_x \left[EI\frac{d^2w}{dx^2}\frac{dv}{dx}\right]_{\Gamma_s} \qquad (9)
\end{aligned}
$$

where Γ_{sw} and $\Gamma_{s\theta}$ are the boundaries where w and θ are prescribed on the local boundary ($\Gamma_s \cap \Gamma_w$ and $\Gamma_s \cap \Gamma_\theta$). The penalty terms are considered only if the local sub-domain, Ω_s, intersects Γ_w or Γ_θ . In general, when a local boundary, Γ_s, intersects a global boundary, Γ, four boundary condition possibilities exist. These possibilities are $\Gamma_s \cap \Gamma_w$, $\Gamma_s \cap \Gamma_\theta$, $\Gamma_s \cap \Gamma_V$, and $\Gamma_s \cap \Gamma_M$ and are denoted Γ_{sw}, $\Gamma_{s\theta}$, Γ_{sV}, and Γ_{sM}, respectively. Additionally, when Γ_s coincides with an interior point, that point is denoted Γ_{sI}. Using these boundary condition possibilities, Eq. (9) is rewritten as

$$
\begin{aligned}
0 = EI \int_{\Omega_s} \frac{d^2w}{dx^2}\frac{d^2v}{dx^2}dx - \int_{\Omega_s} f\ v\ dx \qquad (10) \\
+ \alpha_w \left[(w-\tilde{w})\,v\right]_{\Gamma_{sw}} + \alpha_\theta \left[\left(\frac{dw}{dx} - \tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_{s\theta}} \\
+ n_x \left[EI\frac{d^3w}{dx^3}\ v\right]_{\Gamma_{sI}} - n_x \left[EI\frac{d^2w}{dx^2}\frac{dv}{dx}\right]_{\Gamma_{sI}} - n_x \left[\tilde{V}\ v\right]_{\Gamma_{sV}} \\
- n_x \left[\tilde{M}\frac{dv}{dx}\right]_{\Gamma_{sM}} + n_x \left[EI\frac{d^3w}{dx^3}\ v\right]_{\Gamma_{sw}} - n_x \left[EI\frac{d^2w}{dx^2}\frac{dv}{dx}\right]_{\Gamma_{s\theta}} .
\end{aligned}
$$

As mentioned previously, n_x is the direction cosine of the unit outward drawn normal to Ω_s; n_x = 1 if the boundary is on the right side of Ω_s, and n_x = –1 if the boundary is on the left side of Ω_s.

The trial functions are assumed as

$$
w(x) = \sum_{j=1}^{n} \left(\hat{w}_j \psi_j^{(w)}(x) + \hat{\theta}_j \psi_j^{(\theta)}(x)\right), \qquad (11a)
$$

and the test functions are assumed as

$$v(x) = \mu_i^{(w)}\chi_i^{(w)}(x) + \mu_i^{(\theta)}\chi_i^{(\theta)}(x), \tag{11b}$$

where $\hat{w}_j$ and $\hat{\theta}_j$ are the fictitious nodal values of deflection and slope at node j, ψ_j are the shape functions and n is the number of nodes involved in the GMLS interpolation, $\mu_i^{(w)}$ and $\mu_i^{(\theta)}$ are the arbitrary constants for deflections and slopes of the test function, and $\chi_i^{(w)}(x)$ and $\chi_i^{(\theta)}(x)$ are components of the test functions. The trial and test functions will be discussed in detail later.

3 MLPG Equations

Substituting the trial and test functions of Eq. (11) into the weak form of Eq. (10), and requiring that the weak form be valid for arbitrary values of $\mu_i^{(w)}$ and $\mu_i^{(\theta)}$ leads to the MLPG equations as

$$\mathbf{K}^{(\text{node})}\hat{\mathbf{d}} + \mathbf{K}^{(\text{bdry})}\hat{\mathbf{d}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = \mathbf{0} \tag{12}$$

where “bdry” denotes boundary and

$$\hat{\mathbf{d}} = \left\{\hat{w}_1,\, \hat{\theta}_1,\, \hat{w}_2,\, \hat{\theta}_2,\, \ldots,\, \hat{w}_N,\, \hat{\theta}_N\right\}^T \tag{13a}$$

are the fictitious nodal values of deflections, w, and slopes, θ, at all the N nodes of the model used to analyze the problem, and

$$\mathbf{K}^{(\text{node})} = \left[\mathbf{k}_{ij}^{(\text{node})}\right] \tag{13b}$$

$$\mathbf{K}^{(\text{bdry})} = \left[\mathbf{k}_{ij}^{(\text{bdry})}\right] \tag{13c}$$

with

$$\begin{aligned}\mathbf{k}_{ij}^{(\text{node})} &= EI\begin{bmatrix} \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx \\ \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx \end{bmatrix} \\ &+ n_x EI\begin{bmatrix} \chi_i^{(w)}\frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)}\frac{d^3\psi_j^{(\theta)}}{dx^3} \\ \chi_i^{(\theta)}\frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)}\frac{d^3\psi_j^{(\theta)}}{dx^3} \end{bmatrix}_{\Gamma_{sI}^{(i)}} \\ &- n_x EI\begin{bmatrix} \frac{d\chi_i^{(w)}}{dx}\frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(w)}}{dx}\frac{d^2\psi_j^{(\theta)}}{dx^2} \\ \frac{d\chi_i^{(\theta)}}{dx}\frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(\theta)}}{dx}\frac{d^2\psi_j^{(\theta)}}{dx^2} \end{bmatrix}_{\Gamma_{sI}^{(i)}} \end{aligned} \tag{13d}$$

$$\begin{aligned}\mathbf{k}_{ij}^{(\text{bdry})} &= \alpha_w\begin{bmatrix} \chi_i^{(w)}\psi_j^{(w)} & \chi_i^{(w)}\psi_j^{(\theta)} \\ \chi_i^{(\theta)}\psi_j^{(w)} & \chi_i^{(\theta)}\psi_j^{(\theta)} \end{bmatrix}_{\Gamma_{sw}^{(i)}} \\ &+ n_x EI\begin{bmatrix} \chi_i^{(w)}\frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)}\frac{d^3\psi_j^{(\theta)}}{dx^3} \\ \chi_i^{(\theta)}\frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)}\frac{d^3\psi_j^{(\theta)}}{dx^3} \end{bmatrix}_{\Gamma_{sw}^{(i)}} \\ &+ \alpha_\theta\begin{bmatrix} \frac{d\chi_i^{(w)}}{dx}\frac{d\psi_j^{(w)}}{dx} & \frac{d\chi_i^{(w)}}{dx}\frac{d\psi_j^{(\theta)}}{dx} \\ \frac{d\chi_i^{(\theta)}}{dx}\frac{d\psi_j^{(w)}}{dx} & \frac{d\chi_i^{(\theta)}}{dx}\frac{d\psi_j^{(\theta)}}{dx} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}} \\ &- n_x EI\begin{bmatrix} \frac{d\chi_i^{(w)}}{dx}\frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(w)}}{dx}\frac{d^2\psi_j^{(\theta)}}{dx^2} \\ \frac{d\chi_i^{(\theta)}}{dx}\frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(\theta)}}{dx}\frac{d^2\psi_j^{(\theta)}}{dx^2} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}} \end{aligned} \tag{13e}$$

$$\mathbf{f}^{(\text{node})} = \begin{Bmatrix} \int\limits_{\Omega_s^{(i)}} \chi_i^{(w)} f\,dx \\ \int\limits_{\Omega_s^{(i)}} \chi_i^{(\theta)} f\,dx \end{Bmatrix} \tag{13f}$$

and

$$\begin{aligned}\mathbf{f}^{(\text{bdry})} &= n_x\tilde{M}\begin{Bmatrix} \frac{d\chi_i^{(w)}}{dx} \\ \frac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{sM}^i} + n_x\tilde{V}\begin{Bmatrix} \chi_i^{(w)} \\ \chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sV}^i} \\ &+ \alpha_w\begin{Bmatrix} \tilde{w}\chi_i^{(w)} \\ \tilde{w}\chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sw}^i} + \alpha_\theta\begin{Bmatrix} \tilde{\theta}\frac{d\chi_i^{(w)}}{dx} \\ \tilde{\theta}\frac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{s\theta}^i}, \end{aligned} \tag{13g}$$

where i = 1, 2, $\ldots N$ and j = 1, 2, $\ldots n$.

The system of equations presented in Eqs. (12 – 13g) are the general equations valid for any set of trial and test functions. In this paper, a Petrov-Galerkin method is used; the test functions are chosen to be different from the trial functions. The choices for the trial and test functions are discussed next.

4 Trial Functions: The Generalized Moving Least Squares Interpolation

A generalized moving least squares (GMLS) approximation for w and θ is developed (Atluri *et al.*, 1999) following the general principals of the moving least squares approximation used for C^0 problems (Nayroles *et al.*, 1992 and Atluri and Zhu, 1998).

Nayroles *et al.* (1992) suggested an approximation for w in the neighborhood of node j as

$$w(x) = \mathbf{p}^T(x)\mathbf{a}(x) = \sum_{j=1}^{m} p_j(x)a_j(x), \tag{14}$$

or

$$w(x) = a_1 + a_2x + a_3x^2 + \ldots + a_mx^{m-1} \tag{15}$$

where a_1, a_2, $\ldots$, a_m are, in general, functions of x (Belytschko *et al.*, 1994). Figure 2 shows two identical shape functions, one centered at node j, and the other centered at node e. The global approximation for w (Eq. 15) around node j in Figure 2 can be rewritten in terms of a local coordinate, ξ, $\xi = x - x_j$, as (Raju and Phillips, 2002a)

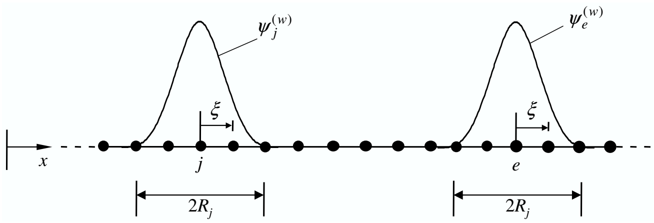


Figure 2 : Local coordinate definitions

$$\begin{aligned} w(x) &= a_1 + a_2\,(x_j+\xi) + a_3\,(x_j+\xi)\ ^2 + \ldots + a_m\,(x_j+\xi)\ ^{m-1} \\ &= \left(a_1 + a_2x_j + a_3x_j^2 + \ldots\right) + \left(a_2 + 2a_3x_j + \ldots\right)\,\xi \\ &+ \left(a_3 + \ldots\right)\,\xi^2 + \ldots + (\,)\,\xi^{m-1}. \end{aligned} \tag{16}$$

Therefore,

$$\begin{aligned} w(x) &= b_1 + b_2\xi + b_3\xi^2 + \ldots + b_m\xi^{m-1} \\ &= \sum_{j=1}^{m} p_j(\xi)b_j(x) = \mathbf{p}^T(\xi)\mathbf{b}(x) \end{aligned} \tag{17}$$

where $x = x_j + \xi$, $\mathbf{p}(\xi)$ is the basis function defined in terms of the local coordinate ξ,

$$\mathbf{p}^T(\xi) = \left[1,\ \xi,\ \xi^2,\ \ldots, \xi^{m-1}\right], \tag{18}$$

with (m-1) as the order of the 1-D basis function, and $\mathbf{b}(x)$ is the new vector of undetermined coefficients, which are functions of x. (A similar local coordinate transformation can be affected for node e in Figure 2 as $x = x_e + \xi$). The $\mathbf{b}(x)$ are evaluated by minimizing a weighted discrete H^h error norm (Nayroles *et al.*, 1992 and Atluri *et al.*, 1999):

$$H^h(\mathbf{b}) = \sum_{j=1}^{n}\sum_{|\alpha|\le h} \lambda_j(x)\left[D^\alpha\{\mathbf{p}^T(\xi)\mathbf{b}\}_j - D^\alpha\hat{w}_j\right]^2 \tag{19}$$

where $\lambda_j(x)$ is a weight function, n is the number of nodes in the domain of definition of the trial function, $D^{\,\alpha}$ denotes the α^{th} derivative, and $\sum\limits_{|\alpha|\le h}$ indicates the summation of all derivatives up to order h. The weight functions, $\lambda_j(x)$, in Eq. (19) play an important role in the MLS approximation. These functions have the following properties. They are continuous, have nonzero values over a range $2R_j$, have unit value at $x = x_j$, and have zero values when $x \le (x_j - R_j)$ and $x \ge (x_j + R_j)$. The extent of the weight function determines the extent of the trial functions (Nayroles *et al.*, 1992, Belytschko *et al.*, 1994, Atluri and Zhu, 1998). Typical weight functions are presented later in the paper.

As the primary variables for beam problems are the deflection, w, and slope, θ = (dw/dx), the weighted discrete H^1 error norm can be written as

$$\begin{aligned} H^1(\mathbf{b}) &= \sum_{j=1}^{n}\sum_{|\alpha|\le 1} \lambda_j(x)\left[D^\alpha\{\mathbf{p}^T(\xi)\mathbf{b}\}_j - D^\alpha\hat{w}_j\right]^{\,2} \\ &= \sum_{j=1}^{n}\left\{\lambda_j^{(w)}(x)\left[\{\mathbf{p}^T(\xi)\mathbf{b}\}_j - \hat{w}_j\right]^2\right. \\ &\left. + \lambda_j^{(\theta)}(x)\left[\left\{\frac{d\mathbf{p}^T(\xi)}{dx}\mathbf{b}\right\}_j - \hat{\theta}_j\right]^2\right\}. \end{aligned} \tag{20}$$

In this paper, $\lambda_j^{(w)}(x)$ and $\lambda_j^{(\theta)}(x)$ are chosen to be identical, and they are hereafter referred to as $\lambda_j(x)$. Equation

(20) can be written in matrix form as

$$H^1(\mathbf{b}) = [\mathbf{P}\mathbf{b}-\hat{\mathbf{w}}]^T \boldsymbol{\lambda} [\mathbf{P}\mathbf{b}-\hat{\mathbf{w}}] + \left[\mathbf{P}_x\mathbf{b}-\hat{\boldsymbol{\theta}}\right]^T \boldsymbol{\lambda} \left[\mathbf{P}_x\mathbf{b}-\hat{\boldsymbol{\theta}}\right]$$

$$= \left\{ \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \end{bmatrix} \{\mathbf{b}\} - \begin{Bmatrix} \hat{\mathbf{w}} \\ \hat{\boldsymbol{\theta}} \end{Bmatrix} \right\}^T \begin{bmatrix} \boldsymbol{\lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \left\{ \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \end{bmatrix} \{\mathbf{b}\} - \begin{Bmatrix} \hat{\mathbf{w}} \\ \hat{\boldsymbol{\theta}} \end{Bmatrix} \right\}$$

$$= [\mathbf{Q}\mathbf{b}-\hat{\mathbf{s}}]^T \boldsymbol{\Lambda} [\mathbf{Q}\mathbf{b}-\hat{\mathbf{s}}] \tag{21}$$

where $\mathbf{P}$ and $\mathbf{P}_x$ are (n,m) matrices and $\boldsymbol{\lambda}$ is a diagonal (n,n) matrix defined as

$$[\mathbf{P}] = \begin{bmatrix} \mathbf{p}^T(\xi_1) & \mathbf{p}^T(\xi_2) & \dots & \mathbf{p}^T(\xi_n) \end{bmatrix}^T, \tag{22a}$$

$$[\mathbf{P}_x] = \begin{bmatrix} \mathbf{p}_x^T(\xi_1) & \mathbf{p}_x^T(\xi_2) & \dots & \mathbf{p}_x^T(\xi_n) \end{bmatrix}^T, \tag{22b}$$

and

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1(x) & & & \\ & \lambda_2(x) & & \\ & & \ddots & \\ & & & \lambda_n(x) \end{bmatrix}, \tag{23}$$

where $\xi_k = x_k - x_j$, k = 1, 2, ..., n,

$$\mathbf{p}^T(\xi) = \begin{bmatrix} 1, & \xi, & \xi^2, & \dots & \xi^{m-1} \end{bmatrix}, \tag{24a}$$

and

$$\mathbf{p}_x^T(\xi) = \frac{d\mathbf{p}^T(\xi)}{dx} = \begin{bmatrix} 0, & 1, & 2\xi, & \dots & (m-1)\xi^{m-2} \end{bmatrix} \tag{24b}$$

as

$$\frac{d}{dx}() = \frac{d}{d\xi}(). \tag{25}$$

Also,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \end{bmatrix}, \hat{\mathbf{s}} = \begin{Bmatrix} \hat{\mathbf{w}} \\ \hat{\boldsymbol{\theta}} \end{Bmatrix}, \text{and } \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \tag{26}$$

are the basis function matrix, the nodal displacement vector, and the weight function matrix, respectively. Minimization of the H^1 norm (i.e., $\partial H^1/\partial\mathbf{b}^T = 0$)leads to

$$\underset{(m,m)}{[\mathbf{A}]}\ \underset{(m,1)}{\{\mathbf{b}\}} = \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2n,1)}{\{\hat{\mathbf{s}}\}}, \tag{27}$$

where

$$\begin{aligned} \underset{(m,m)}{[\mathbf{A}]} &= \underset{(m,2n)}{[\mathbf{Q}]^T}\ \underset{(2n,2n)}{[\boldsymbol{\Lambda}]}\ \underset{(2n,m)}{[\mathbf{Q}]} \\ &= \underset{(m,n)}{[\mathbf{P}]^T}\ \underset{(n,n)}{[\boldsymbol{\lambda}]}\ \underset{(n,m)}{[\mathbf{P}]} + \underset{(m,n)}{[\mathbf{P}_x]^T}\ \underset{(n,n)}{[\boldsymbol{\lambda}]}\ \underset{(n,m)}{[\mathbf{P}_x]} \end{aligned} \tag{28}$$

and

$$\underset{(m,2n)}{[\mathbf{B}]} = \underset{(m,2n)}{[\mathbf{Q}]^T}\ \underset{(2n,2n)}{[\boldsymbol{\Lambda}]} = \begin{bmatrix} \underset{(m,n)}{[\mathbf{P}]^T}\ \underset{(n,n)}{[\boldsymbol{\lambda}]}, & \underset{(m,n)}{[\mathbf{P}_x]^T}\ \underset{(n,n)}{[\boldsymbol{\lambda}]} \end{bmatrix}. \tag{29}$$

Solving for $\{\mathbf{b}\}$ using Eq. (27) gives

$$\underset{(m,1)}{\{\mathbf{b}\}} = \underset{(m,m)}{[\mathbf{A}]^{-1}}\ \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2n,1)}{\{\hat{\mathbf{s}}\}} \tag{30}$$

Substituting into the approximation Eq. (17),

$$w(x) \cong \underset{(1,m)}{\mathbf{p}^T(\xi)}\ \underset{(m,m)}{[\mathbf{A}]^{-1}}\ \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2n,1)}{\{\hat{\mathbf{s}}\}}. \tag{31}$$

The trial functions used for beam problems are finally written as a linear combination of nodal shape functions:

$$w(x) = \sum_{j=1}^{n} \left(\hat{w}_j \psi_j^{(w)}(x) + \hat{\theta}_j \psi_j^{(\theta)}(x) \right), \tag{32}$$

where

$$\begin{aligned} \psi_j^{(w)}(x) &= \sum_{g=1}^{m} p_g(\xi_j) \left[[\mathbf{A}]^{-1} [\mathbf{P}]^T [\boldsymbol{\lambda}] \right]_{gj} \text{ and} \\ \psi_j^{(\theta)}(x) &= \sum_{g=1}^{m} p_g(\xi_j) \left[[\mathbf{A}]^{-1} [\mathbf{P}_x]^T [\boldsymbol{\lambda}] \right]_{gj}. \end{aligned} \tag{33}$$

The derivatives of these shape functions are needed in Eqs. (13). Explicit expressions for these derivatives are presented in Appendix A.

4.1 Weight Functions

Three types of weight functions, $\lambda_j(x)$, are considered for constructing the trial functions – power functions and two different spline functions – to demonstrate the robustness of the MLPG method. The power functions considered are (Atluri *et al.*, 1999)

$$\lambda_j(x) = \begin{cases} \left[1-\left(\frac{d_j}{R_j}\right)^2\right]^\alpha & \text{if } 0 \le d_j \le R_j \\ 0 & \text{if } d_j > R_j, \end{cases} \tag{34}$$

with $d_j = ||x - x_j||$, the Euclidean distance between x and x_j, and α = 2, 3, 4, 5, and 6. Note that higher values of α yield weight functions with higher order continuity (Atluri and Shen, 2002a). The spline functions considered are a 3-term spline,

$$\lambda_j(x) = \begin{cases} 1-3\left(\frac{d_j}{R_j}\right)^2+2\left(\frac{d_j}{R_j}\right)^3 & \text{if } 0 \le d_j \le R_j \\ 0 & \text{if } d_j > R_j, \end{cases} \tag{35}$$

and a 4-term spline (Atluri and Zhu, 1998),

$$\lambda_j(x) = \begin{cases} 1-6\left(\frac{d_j}{R_j}\right)^2+8\left(\frac{d_j}{R_j}\right)^3-3\left(\frac{d_j}{R_j}\right)^4 & \text{if } 0 \le d_j \le R_j \\ 0 & \text{if } d_j > R_j. \end{cases} \tag{36}$$

In Eqs. (34 – 36), R_j is a user-defined parameter that controls the extents of the trial functions.

5 Test Functions

As mentioned previously, the test function, v, is assumed as in Eq. (11b) as

$$v(x) = \mu_i^{(w)}\chi_i^{(w)}(x) + \mu_i^{(\theta)}\chi_i^{(\theta)}(x).$$

In previous literature, a generalized moving least squares (GMLS) interpolation scheme was used to develop a Galerkin formulation for analyzing beam problems. The trial and test functions in the meshless Galerkin formulation for beam problems were chosen to be identical, i.e., $\chi_j \equiv \psi_j$. This formulation showed discontinuities ("scissors") at the boundaries of the supports of the higher order derivatives of the trial functions in the local subdomain of the test function. Due to these scissors, elaborate numerical integration schemes were needed to integrate the weak form accurately (Atluri *et al.*, 1999).

In this paper, a true Petrov-Galerkin method is used, i.e., the test function components, χ_i, are chosen to be distinctly different from the shape functions, $\psi_i \quad (\chi_i \neq \psi_i)$. In the conventional MLPG method, the $\chi_i^{(w)}(x)$components of the test functions are chosen as simple weight functions similar to those of Eqs. (34 – 36) as

$$\chi_i^{(w)}(x) = \begin{cases} \left[1-\left(\frac{d_i}{R_o}\right)^2\right]^\beta & \text{if } 0 \le d_i \le R_o \\ 0 & \text{if } d_i > R_o, \end{cases} \tag{37}$$

power weight functions with $d_i = ||x - x_i||$ and β = 2, 3, and 4, a 3-term spline,

$$\chi_i^{(w)}(x) = \begin{cases} 1-3\left(\frac{d_i}{R_o}\right)^2+2\left(\frac{d_i}{R_o}\right)^3 & \text{if } 0 \le d_i \le R_o \\ 0 & \text{if } d_i > R_o, \end{cases} \tag{38}$$

and a 4-term spline,

$$\chi_i^{(w)}(x) = \begin{cases} 1-6\left(\frac{d_i}{R_o}\right)^2+8\left(\frac{d_i}{R_o}\right)^3-3\left(\frac{d_i}{R_o}\right)^4 & \text{if } 0 \le d_i \le R_o \\ 0 & \text{if } d_i > R_o. \end{cases} \tag{39}$$

In Eqs. (37 – 39), R_o is a user-defined parameter that determines the extent of the test functions (and hence Ω_s). The components of the test functions chosen for θ are the first derivatives of the components of the test functions chosen for the primary variable, w, i.e.,

$$\chi_i^{(\theta)} = \frac{d\chi_i^{(w)}}{dx}, \tag{40}$$

as θ = (dw/dx) is also a primary variable.

For the power functions with β = 3 and 4 and the 4-term spline function, the values of $\chi_i^{(w)}$, $\chi_i^{(\theta)}$, $(d\chi_i^{(w)}/dx)$, and $(d\chi_i^{(\theta)}/dx)$ are zero when $d_i = R_o$. The point $d_i = R_o$ corresponds to the end points of each Ω_s, and when Ω_s does not intersect the global boundary (Γ), its Γ_s are the Γ_{sI} (Eq. 13d). Consequently, when these test functions are used, the $\mathbf{k}^{(\text{node})}$ in Eq. (13d) reduces to

$$\mathbf{k}_{ij}^{(\text{node})} = EI \begin{bmatrix} \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx \\ \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx \end{bmatrix}. \tag{41}$$

For the 3-term spline and the power function with $\beta = 2$, the values of $\chi_i^{(w)}$, $\chi_i^{(\theta)}$, and $(d\chi_i^{(w)}/dx)$ are zero when $d_i = R_o$; however, the derivative $(d\chi_i^{(\theta)}/dx)$ is nonzero when $d_i = R_o$. As such, all the terms involved in Eq. (13d) need to be considered.

6 Computationally Efficient Modification to the MLPG Method

Recently, Atluri and Shen (2002a and 2002b) presented six different variations of the MLPG method (MLPG1, MLPG2, ..., MLPG6) for Poisson problems by choosing different classes of test functions for each of the variations. Of the six, the most attractive variation is the choice of a heaviside function as the test function (called MLPG5 by Atluri and Shen, 2002b). With this choice, the domain (Ω_s) integrals involved in the weak form vanish, and this leads to a very computationally efficient and inexpensive MLPG method.

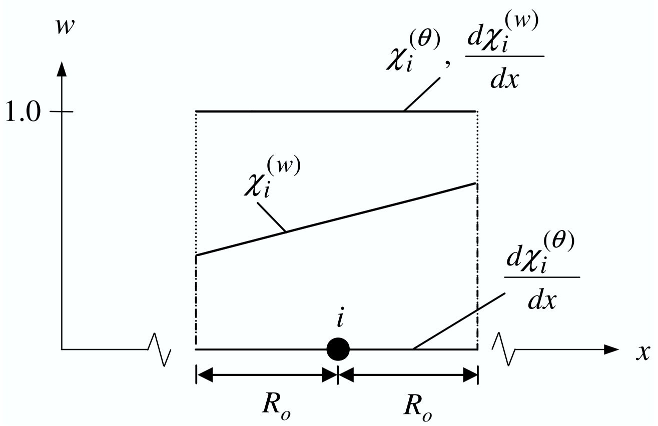

w
$\chi_i^{(\theta)}, \dfrac{d\chi_i^{(w)}}{dx}$
1.0
$\chi_i^{(w)}$
$\dfrac{d\chi_i^{(\theta)}}{dx}$
i
x
R_o
R_o

Figure 3 : Linear test function components and their derivatives

A similar variation of the MLPG method is proposed here for beam problems. The components of the test function of Eq. (11b) in this variation are chosen as (see Figure 3)

$$\begin{aligned} \chi_i^{(w)}(x) &= x \\ \chi_i^{(\theta)}(x) &= 1 \end{aligned} \qquad \text{if } 0 \le d_i \le R_o \tag{42a}$$

and

$$\begin{aligned} \chi_i^{(w)}(x) &= 0 \\ \chi_i^{(\theta)}(x) &= 0 \end{aligned} \qquad \text{if } d_i > R_o \tag{42b}$$

so that

$$\frac{d^2\chi_i^{(w)}}{dx^2} = \frac{d^2\chi_i^{(\theta)}}{dx^2} = 0. \tag{43}$$

Thus the integrand of the integral over Ω_s vanishes in Eq. (13d) and the $\mathbf{k}^{(\text{node})}$ reduces to

$$\mathbf{k}_{ij}^{(\text{node})} = +n_x EI \begin{bmatrix} \chi_i^{(w)} \frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)} \frac{d^3\psi_j^{(\theta)}}{dx^3} \\ \chi_i^{(\theta)} \frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)} \frac{d^3\psi_j^{(\theta)}}{dx^3} \end{bmatrix}_{\Gamma_{sI}} - n_x EI \begin{bmatrix} \frac{d\chi_i^{(w)}}{dx} \frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(w)}}{dx} \frac{d^2\psi_j^{(\theta)}}{dx^2} \\ \frac{d\chi_i^{(\theta)}}{dx} \frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(\theta)}}{dx} \frac{d^2\psi_j^{(\theta)}}{dx^2} \end{bmatrix}_{\Gamma_{sI}}. \tag{44}$$

Note that for the higher order $(\beta > 2)$ power functions and the 4-term spline function considered in Eqs. (37 – 39), the values of the $\chi_i^{(w)}$, $\chi_i^{(\theta)}$, $(d\chi_i^{(w)}/dx)$, and $(d\chi_i^{(\theta)}/dx)$ are zero at Γ_{sI}, while the $(d^2\chi_i^{(w)}/dx^2)$ and $(d^2\chi_i^{(\theta)}/dx^2)$ are nonzero everywhere in Ω_s. In contrast, for the choice of the test function components of Eq. (42), $\chi_i^{(w)}$, $\chi_i^{(\theta)}$, and $(d\chi_i^{(w)}/dx)$ are nonzero at Γ_{sI}, and the $(d^2\chi_i^{(w)}/dx^2)$ and $(d^2\chi_i^{(\theta)}/dx^2)$ are identically zero everywhere in Ω_s. Thus the contributions to the $\mathbf{k}^{(\text{node})}$ for this variation of the MLPG method come only from the ends of Ω_s, the Γ_{sI} points. This leads to a computationally efficient and inexpensive method as the integrations involved in the Ω_s integrals are eliminated.

In the sections that follow, the two variations of the MLPG method are used to analyze several numerical examples. For convenience in presentation, the conventional MLPG method will hereafter be referred to as in Atluri and Shen (2002a and 2002b) as the "MLPG1," and the computationally efficient modification to the MLPG method (with linear test functions) will be referred to as the "MLPG5."

7 Beam Configurations and Models

A beam of constant flexural rigidity EI and a length of $4l$ is considered. The length $4l$ was specifically chosen to avoid scaling by a unit length, l. Six models with 5, 9, 17, 33, 65, and 129 nodes uniformly distributed along the length of the beam are considered. Figure 4 shows a typical 17-node model. The distances between the nodes (Δx/

l) in these models are 1, 0.5, 0.25, 0.125, 0.0625, and 0.03125 for the 5-, 9-, 17-, 33-, 65-, and 129-node models, respectively. Four types of basis function, linear basis (1, ξ), quadratic basis (1, ξ, ξ^2), cubic basis (1, ξ, ξ^2, ξ^3), and quartic basis (1, ξ, ξ^2, ξ^3, ξ^4) are used. Numerical integration is used to integrate the system of equations as closed-form integration of the terms in Eqs. (13d and 13f) is extremely complicated. The order of Gaussian integration required for acceptable results in the MLPG method depends on the basis functions and weight functions used. The system of equations (Eqs. 12 – 13g) is developed with these parameters. For each problem analyzed, the results for the MLPG1 are discussed first, followed by the results for the MLPG5.

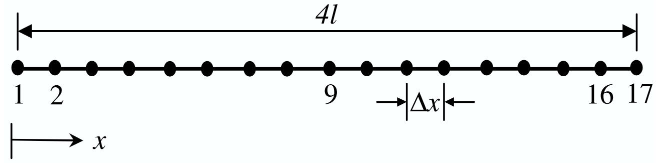


Figure 4 : A 17-node model of the beam

While the MLPG method gives smooth distributions for the primary and secondary variables, in the numerical evaluations that follow, the MLPG solutions are evaluated at 25 points along the length of the beam and shown as symbols in the figures, and the exact solutions are shown as continuous lines. Additionally, certain values of several parameters are specified for the individual problems. For the trial function in each problem, the order of the basis function and the weight function, $\lambda_j(x)$ from Eqs. (34 – 36), are chosen. For the test function in each problem, the component $\chi_i^{(w)}(x)$ is chosen from Eqs. (37, 38, 39, or 42). The component $\chi_i^{(\theta)}(x)$ is the derivative of $\chi_i^{(w)}(x)$. The user-controlled parameters are (R_o/l) – the extent of the test functions, (R_j/l) – the extent of the trial functions, and the order of the Gaussian integration.

8 Numerical Evaluations – Patch Tests

The two current variations of the MLPG formulation were evaluated by applying the formulations to simple patch-test problems. The problems considered were (a) rigid body translation:

$$w(x) = c_0, \theta = \frac{dw}{dx} = 0 \ , \tag{45a}$$

(b) rigid body rotation:

$$w(x) = c_1 x, \theta = c_1, \tag{45b}$$

and (c) constant-curvature condition:

$$w(x) = c_2 x^2/2, \theta = c_2 x, \tag{45c}$$

where c_0, c_1, and c_2 are arbitrary constants. The third patch test could be looked upon as the problem of a cantilever beam with a moment, M=$EI(d^2w/dx^2)$= EIc_2, applied at x=4l. The deflection, w, and the slope, θ, corresponding to problems (a), (b), and (c) were prescribed as essential boundary conditions (EBCs) at x =0 and x =4l. With these EBCs, the beam problems were analyzed using the MLPG method. If the MLPG method recovers the exact solution at all the interior nodes and at every arbitrary point of the beam, then the MLPG method passes the patch test.

In preliminary evaluations, the $\chi_i^{(w)}(x)$component of the test functions in the MLPG weak form was chosen as Eq. (37) with $\beta = 4$. The weight functions $\lambda_j(x)$ used to construct the trial functions were Eq. (34) with α = 3 and α = 4. The term (R_o/l) (Eq. 37) in each of these six models was different and chosen equal to (2Δx). The (R_j/l) in Eq. (34) was chosen to be $(R_j/l = 3.5)$ for the 5- and 9-node models and $(R_j/l = 16\Delta x)$ for the 17-, 33-, 65-, and 129-node models. The usable ranges of (R_o/l) and (R_j/l) are discussed later.

For a displacement of $w(x) = c_0$ and $c_1 x$ units, the rigid body conditions (Eqs. 45a and 45b) were modeled with boundary conditions

(a) $w|_{x=0} = c_0, \quad w|_{x=4l} = c_0,$
$\theta|_{x=0}$= 0, $\quad \theta|_{x=4l}$= 0.

(b) $w|_{x=0}$= 0, $\quad w|_{x=4l}$= $4c_1 l$,
$\theta|_{x=0} = c_1, \quad \theta|_{x=4l} = c_1.$

Since the exact solutions are constant and linear in x, respectively, the MLPG method developed with a linear or higher order basis function must reproduce the solutions exactly. As expected, the algorithm reproduced the exact solutions for w and θ to machine accuracy for both rigid body modes at all the nodes and at any arbitrary point in the beam.

For the constant – curvature condition, $w = c_2x^2/2$, the problem was modeled with EBCs

(c) $w|_{x=0} = 0, \quad w|_{x=4l} = 8c_2l^2,$

$\theta|_{x=0} = 0, \quad \theta|_{x=4l} = 4c_2l.$

Since the exact solution is quadratic in x, the MLPG method developed with a quadratic or higher order basis function must reproduce the solution exactly. As expected, the algorithm reproduced the exact solution for the primary variables to machine accuracy at all nodes and at any arbitrary point in the beam. When the linear basis function was used, the accuracy of the MLPG method increased as the value of α in Eq. (34) increased from 3 to 5. A value of α = 5 gave results that are very close to machine accuracy. Thus, a quintic weight function (α = 5) is preferable when the linear basis function is used.

An 8-point Gaussian quadrature was found to be sufficient to integrate the weak form accurately when the power functions with β = 2, 3, and 4 were used. The MLPG1 passed the patch tests with these test function components. When the spline functions (Eqs. 38 and 39) were used as test function components, elaborate integration like that reported by Atluri *et al.* (1999) was needed. The Ω_s was subdivided into sub-regions in a manner similar to Figure 8 of Atluri *et al.* (1999), and in each subregion, a 20-point Gaussian integration was used. The 4-term spline passed all the patch tests to machine accuracy with this integration scheme. On the other hand, the results obtained with the 3-term spline were numerically unreliable.

The three patch test problems were repeated with the linear test function (Eq. 42). The MLPG5 reproduced exact solutions to machine accuracy, thus passing all the patch tests.

9 Problem Parameters

As mentioned previously, the parameters (R_o/l) and (R_j/l) in the MLPG method are user-controlled. Ranges of values of these parameters were studied, and a general rule of thumb was established. The lengths $(R_o/l = 2\Delta x)$ and $(R_j/l = 8\Delta x)$ were used at all nodes of an N-node model. As the models are refined, the value of $(\Delta x/l)$ decreases, and thus the size of Ω_s and the extent of the trial functions also decrease. For finer models, i.e. for the 33-, 65-, and 129-node models, when $8\Delta x \leq (R_j/l) \leq 16\Delta x$, the MLPG method yielded very accurate results. As the value of (R_j/l) approached $(20\Delta x)$, the accuracy of the MLPG method progressively deteriorated for the patch tests. When (R_j/l) exceeds $(20\Delta x)$, the trial function is too diffused, and the size of $\Omega_s(R_o/l = 2\Delta x)$ is too small in comparison to (R_j/l). The small Ω_s size and large (R_j/l) are apparently incompatible. While the finer models performed well over a large range of (R_j/l), the coarser models performed well in a much smaller range of (R_j/l). For good performance, the (R_j/l) needed to be approximately $(8\Delta x)$ but less than the total beam length. Special cases of (R_o/l) values that should be avoided when using the MLPG5 are presented in Appendix B.

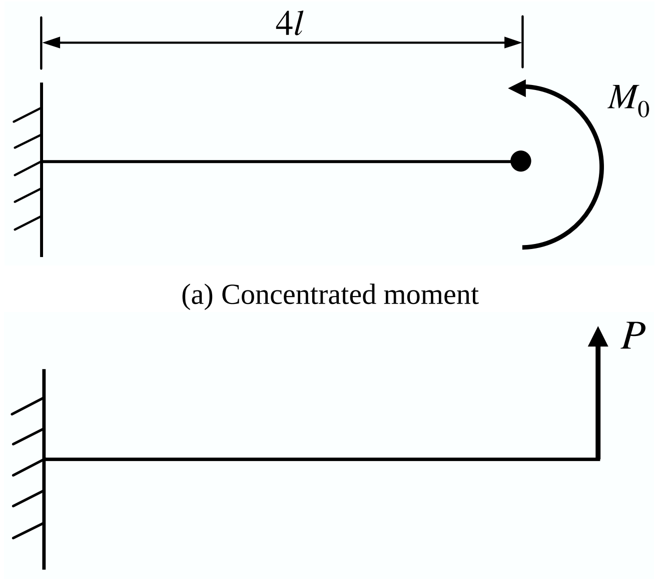
10 Numerical Evaluations – Mixed Boundary Value Problems

The MLPG method was applied to beam problems with mixed boundary conditions. In this section, cantilever beam problems are presented first. Then, the problem of a simply supported beam subjected to a uniformly distributed load is studied in great detail to demonstrate several important features of the MLPG method for beam problems.

10.1 Cantilever beam problems

The first problem considered was a cantilever beam with a concentrated moment at the free end (i.e. $M = M_0$ at x = $4l$, see Figure 5a). The exact solution for this problem is $w = M_0x^2$ / $2EI$ and θ = M_0x / EI. For all trial functions considered (i.e., Eq. (34) with α = 2, 3, and 4, the 3-term spline (Eq. 35), and the 4-term spline (Eq. 36), each combined with a quadratic basis), the MLPG1 reproduced the exact solution when the test function components in Eq. (37) with β = 2, 3, and 4 (and when the 4-term spline function in Eq. 39) were used. Also for all trial functions considered, the MLPG5 reproduced the exact solution. This behavior suggests that the smoothness of the weight functions is less restrictive when the weight functions are used to construct the trial functions than when the weight functions are used as test function components.

The second problem considered was a cantilever beam with a tip load (see Figure 5b). Since the exact solution for this problem is cubic in terms of the x-coordinate of the beam, all six models with cubic basis functions



(a) Concentrated moment

(b) Tip load

Figure 5 : Cantilever beam problems

and power test function components with β = 2, 3, and 4 (MLPG1) or a linear test function (MLPG5) reproduced the exact solution to machine accuracy.

In summary, the power test function components with β = 2, 3, and 4 for the MLPG1 and the linear test function (MLPG5) showed excellent performance in the patch tests and mixed boundary value problems. On the other hand, the spline test function components did not show good performance and needed the special sub-region integration discussed previously. Hence, in the remainder of this paper, the spline test function components are deleted from further consideration.

10.2 Simply supported beam subjected to uniformly distribute load

The third problem considered was a simply supported beam subjected to a uniformly distributed load (see Figure 6). The problem was analyzed with various combinations of parameters to demonstrate interesting features of the MLPG method. The parameters used for each case are summarized in Table 1.

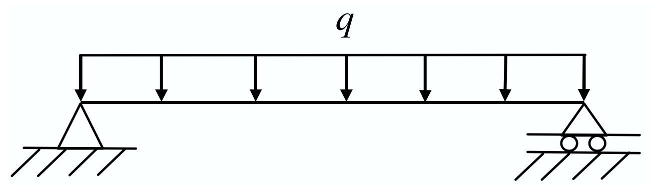


Figure 6 : Simply supported beam subjected to a uniformly distributed load

10.2.1 Basis function

The exact solution for this problem is given by

$$w = \frac{q}{24EI}\left(-2Lx^3 + x^4 + L^3 x\right) ,$$
$$\frac{dw}{dx} = \frac{q}{24EI}\left(-6Lx^2 + 4x^3 + L^3\right) \qquad (46)$$

where L = 4l. Using symmetry, half of the beam was modeled. Since the exact solution for this problem is quartic in terms of the x-coordinate of the beam, the MLPG method with a cubic basis function did not reproduce the exact solution. Error norms defined as

$$\|E_w\|_2 = \sqrt{\frac{1}{g}\sum_{k=1}^{g}\left[\frac{(w_{MLPG} - w_{exact})}{w_{exact}}\right]_k^2}$$
$$\|E_M\|_2 = \sqrt{\frac{1}{g}\sum_{k=1}^{g}\left[\frac{(M_{MLPG} - M_{exact})}{M_{exact}}\right]_k^2} \qquad (47)$$

were computed at g uniformly spaced points along the beam. A value of g = 200 was used. The norms $||E_w||_2$ and $||E_M||_2$ are presented in Table 2.

For the parameters studied, the method did not show monotonic convergence. All models $(N \geq 9)$ yielded accurate solutions (within 4% for w and M). As the number of nodes in the models was increased, the $||E_w||_2$ norm changed marginally, suggesting the same accuracy in the solutions for the various models. Also, the $||E_M||_2$ norm was of the same order as the $||E_w||_2$ norm, suggesting the same accuracy for the primary and the secondary variables. When the order of the basis function was increased to quartic, the MLPG method reproduced the exact solutions (for w, θ, M, and V) to machine accuracy.

When the basis function is not sufficient to recover the exact solution, the accuracy of the MLPG solution depends on the choice of basis and weight function used to

Table 1 : **Problem parameters – simply supported beam subjected to a uniformly distributed load**

Model	Basis used	$\lambda_j(x)$	$N_G^\dagger$	$\chi_i^{(w)}(x)$	(R_o/l)	(R_j/l)
5-, 9-node; uniform spacing; symmetric half	Cubic	α = 3	20	β = 4	2Δx	3.5
	Quartic	α = 3	8	β = 4	2Δx	3.5
17-, 33-, 65-, 129-node; uniform spacing; symmetric half	Cubic	α = 3	20	β = 4	2Δx	8Δx
	Quartic	α = 3	8	β = 4	2Δx	8Δx
65-node; uniform spacing; full beam	Cubic	α = 3	20	β = 4	2Δx	8Δx
	Quadratic	α = 4	20	β = 4	2Δx	8Δx
	Linear	α = 5	20	β = 4	2Δx	8Δx
5-, 9-node; uniform spacing; full beam	Quartic	α = 3	8	β = 4	< Δx	3.5
17-, 33-, 65-, 129-node; uniform spacing; full beam	Quartic	α = 3	8	β = 4	< Δx	8Δx
19-node; non-uniform spacing; full beam	Quartic	α = 3	8	β = 4	0.5	2.0
	Quartic	α = 3	8	Eq. 43	0.5	2.0

$^\dagger N_G$ = order of Gaussian integration

Table 2 : **Error norm** $||\mathbf{E}||_2$ **for a simply supported beam subjected to a uniformly distributed load with cubic basis used in the MLPG1. (Trial function using Eq. (34) with α=3 and test function using Eq.(37) with β=4.)**

Error norm	Number of nodes in the model					
	5*	9*	17†	33†	65†	129†
$\|\|E_w\|\|_2$	0.1662e-1	0.1306e-2	0.4573e-2	0.3829e-1	0.1742e-1	0.2368e-1
$\|\|E_M\|\|_2$	0.2774e+0	0.1057e-1	0.1704e-1	0.3680e-1	0.1763e-1	0.2340e-1

* R_j/l = 3.5, $^\dagger R_j/l$ = 8Δx

construct the trial function. To demonstrate this behavior, the full beam problem was analyzed using three different trial functions. Figure 7 shows the solutions for the three different combinations of basis and weight function. As suggested previously, as the order of the basis function decreased, the order of the corresponding weight function needed to increase to maintain accuracy.

This problem also demonstrates an interesting phenomenon. When the order of the basis function equals the order of the exact solution, an 8-point Gaussian quadrature in a single Ω_s is found to integrate the weak form very accurately. However, when the order of the basis function is less than the order of the exact solution, a higher order integration rule (such as a 20-point Gaussian integration) is needed to obtain accurate results.

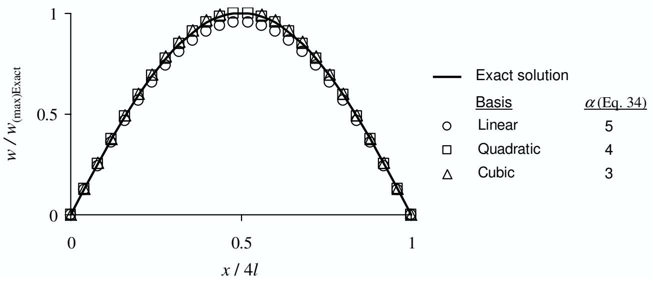


Figure 7 : Comparison of solutions obtained from three different combinations of basis and weight function

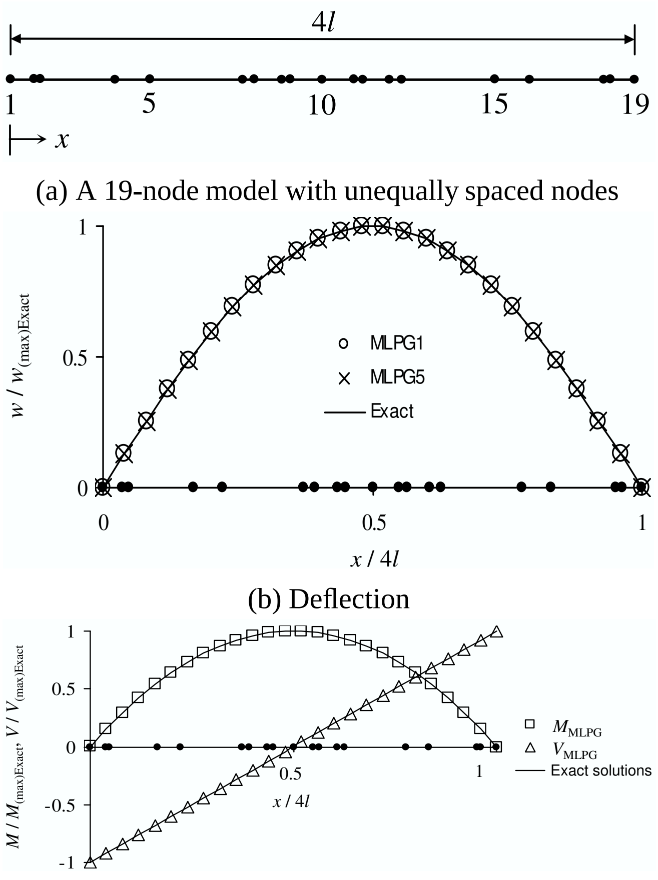
10.2.2 Overlapping Ω_s regions

All the models that have been considered so far had overlapping Ω_s regions in the beam. To demonstrate that overlapping Ω_s regions are not required by the MLPG method, several $(R_o/l < \Delta x/2)$ cases were used with all six (5-, 9-, . . . , 129-node) of the models. For the MLPG1 and MLPG5, all models reproduced the exact solutions to machine accuracy when the quartic basis function was used. These computations suggest that the Ω_s regions need not overlap for the MLPG method. However, for the algorithm to perform accurately, the extents of the trial functions $(2R_j/l)$ must collectively cover the whole analysis domain of the beam – this condition is inherently satisfied by the MLS approximation.

Another numerical experiment was considered. The order of the basis function was set to be less than that required by the problem. (For the simply supported beam problem with the uniformly distributed loading, the order was set to cubic.) Non-overlapping Ω_s regions $(R_o/l < \Delta x/2)$ were used. Both the MLPG1 and MLPG5 failed to yield a solution anywhere near the correct solution. Thus, as a general rule of thumb, non-overlapping Ω_s regions are not recommended.

10.2.3 Non-uniform nodal spacing

The problem of the simply supported beam subjected to a uniformly distributed load was modeled next using the full beam with non-uniform nodal spacing shown in Figure 8a. This model was generated by randomly placing nodes in the region $0 < x < 2l$ and symmetrically replicating these nodes in the region $2l < x < 4l$. A quartic basis function and $(R_o \;/\; l)$ = 0.5 were used. The MLPG1 and exact solutions for deflection, moment, and shear are presented in Figures 8b and 8c. As expected, the MLPG1 reproduced the exact solutions to machine accuracy for both the primary and secondary variables despite the nodal arrangement. The MLPG5 yielded similar results (see Figure 8b).



(a) A 19-node model with unequally spaced nodes

(b) Deflection

(c) Moment and Shear Forces, MLPG1

Figure 8 : MLPG and exact solutions for a simply supported beam subjected to a uniformly distributed load

11 Numerical Evaluations – Load Discontinuities

In all meshless methods, obtaining continuous secondary variables is an objective, and the trial functions are chosen with this in mind. However, across concentrated loads and interior supports, the secondary variables are discontinuous. When meshless methods are applied to problems with load discontinuities, these conflicting conditions are expected to cause difficulties. To evaluate the performance of the MLPG method for problems with concentrated loads, discontinuous loads, and interior supports, three problems were analyzed. These were (1) a simply supported beam with a central concentrated load, (2) a cantilever beam with uniformly distributed loading on a portion of the beam, and (3) a continuous beam subjected to a uniformly distributed load.

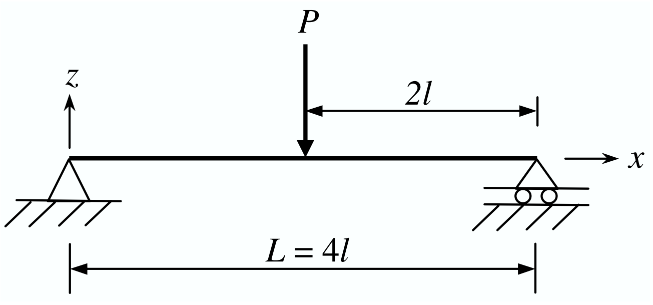


Figure 9 : Simply supported beam subjected to a central concentraded load

11.1 Simply supported beam subjected to a central concentrated load

The first problem considered was a simply supported beam subjected to a central concentrated load (see Figure 9). The exact solution for this problem is

$$\begin{aligned} EIw &= -\tfrac{P}{12}x^3 + \tfrac{PL^2}{16}x \\ EI\tfrac{dw}{dx} &= EI\theta = -\tfrac{P}{4}x^2 + \tfrac{PL^2}{16} \end{aligned} \quad \text{for } 0 \le x \le \frac{L}{2} \tag{48}$$

and

$$\begin{aligned} EIw &= \tfrac{P}{12}x^3 - \tfrac{PL}{4}x^2 + \tfrac{3PL^2}{16}x - \tfrac{PL^3}{48} \\ EI\tfrac{dw}{dx} &= EI\theta = \tfrac{P}{4}x^2 - \tfrac{PL}{2}x + \tfrac{3PL^2}{16} \end{aligned} \quad \text{for } \frac{L}{2} \le x \le L \tag{49}$$

where L = 4l. The problem was analyzed in two different ways. First, utilizing the symmetry in the problem, one-half of the beam was modeled. Next, the full beam was modeled without the use of symmetry.

Various parameters used to analyze the beam are summarized in Table 3. For the symmetric representation of the beam, the boundary conditions used were $w = M = 0$ at $x = 0$, and $V = -P/2$ and $\theta = 0$ at $x = L/2 = 2l$. An 8-point Gaussian integration was used. The MLPG1 and the MLPG5, as expected, reproduced the exact solutions for all models at all nodes and at all interior points of the beam.

Next, the full beam was modeled without utilizing symmetry. The exact solution is cubic in x and is in two parts, and as such the MLPG method is not expected to reproduce the exact solution. To evaluate the performance of the method, cubic and quartic basis functions were considered. As previously mentioned, when the order of the basis function equals the order of the exact solution, an 8-point Gaussian is sufficient to integrate the weak form. However, due to the discontinuity in the applied loading, the 8-point Gaussian was found to be inadequate to accurately integrate the weak form when the full beam was modeled, and a higher order integration rule (such as the 20-point Gaussian) was needed. The normalized MLPG1 values of $w_{(\max)}$ and $\theta_{(\max)}$ for each of the models studied are presented in Table 4.

The MLPG1 solutions with the quartic basis and the exact solutions for deflection and moment of the 65-node model are compared in Figure 10. These plots and the results presented in Table 4 demonstrate that the MLPG1 yields excellent results for both primary and secondary variables. These results were obtained without the use of elaborate post-processing techniques. As the number of nodes was increased from 33 to 129, the accuracy of the solutions did not appreciably change, suggesting that a 33-node model is sufficient to obtain an accurate solution. The solutions obtained with the coarser models (5-, 9-, and 17-node) are not as accurate as those obtained with the finer models. These results suggest that more nodes are needed around a concentrated load to handle the discontinuity caused by the loading condition.

The MLPG5 is considered next with the six (5-, 9-, 17-, 33-, 65-, and 129-node) models for the full beam. This method was unable to yield meaningful results. Apparently, the MLPG5 is unable to handle the load discontinuity. In other words, the linear test function is unable to control the errors when the full beam is modeled and requires further study. Thus, the MLPG5 is not applied to the next two problems.

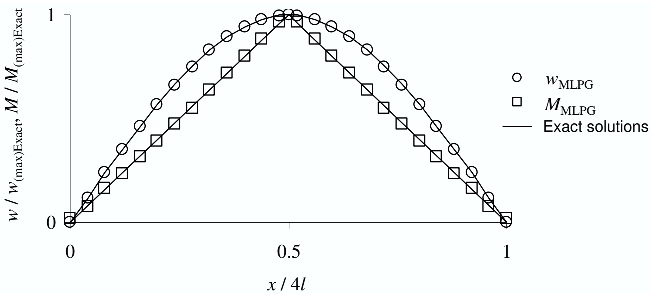


Figure 10 : MLPG1 and exact solutions for a simply supported beam with a central concentrated load

11.2 Cantilever beam with a UDL on a portion of the beam

The second problem considered was the cantilever beam with $L = 2l$ and with a UDL on a portion of the beam depicted in Figure 11. The exact solution for this problem is

$$\left.\begin{aligned} w &= \tfrac{q}{24EI}\left[-(l-x)^4 - 4l^3x + l^4\right] \\ \theta &= \tfrac{q}{6EI}\left[(l-x)^3 - l^3\right] \end{aligned}\right\} \text{ for } 0 \le x \le l \tag{50}$$

and

$$\left.\begin{aligned} w &= \tfrac{q}{24EI}\left[-3l^4 - 4l^3(x-l)\right] \\ \theta &= -\tfrac{ql^3}{6EI} \end{aligned}\right\} \text{ for } l \le x \le 2l. \tag{51}$$

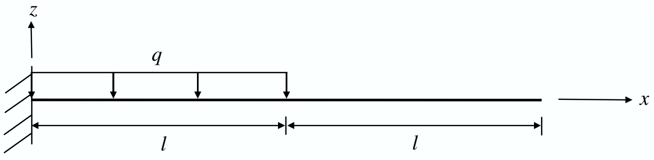


Figure 11 : Cantilever beam subjected to a discontinuous distributed load

As a result of the choice of test function in the algorithm, careful attention is needed when evaluating the contribution of the distributed load to the $\mathbf{f}^{(\text{node})}$ vector in Eq. (13f). Consider the region of the 17-node beam in the area localized around $x = l$ in Figure 12. In this figure, the test functions centered around nodes 9, 10, and 11 are shown along with the distributed load, q. The $\mathbf{f}^{(\text{node})}$ is evaluated over the sub-domain, Ω_s; however, the distributed load, q, does not extend the length of every Ω_s

Table 3 : Problem parameters – simply supported beam subjected to a central concentrated load (weight function $\lambda_j(x)$ with $\alpha = 3$ in Eq. 34 and $(R_o\ /\ l) = 2\Delta x$)

Model	Basis used	$N_G^\dagger$	$\chi_i^{(w)}(x)$	(R_j/l)
5-, 9-node; uniform spacing; symmetric half	Cubic	8	$\beta = 4$	3.5
	Cubic	8	Eq. 42	3.5
17-, 33-, 65-, 129-node; uniform spacing; symmetric half	Cubic	8	$\beta = 4$	$8\Delta x$
	Cubic	8	Eq. 42	$8\Delta x$
5-, 9-node; uniform spacing; full beam	Cubic	20	$\beta = 4$	3.5
	Quartic	20	$\beta = 4$	3.5
17-, 33-, 65-, 129-node; uniform spacing; full beam	Cubic	20	$\beta = 4$	$8\Delta x$
	Quartic	20	$\beta = 4$	$8\Delta x$
	Quartic	20	Eq. 42	$8\Delta x$

$^\dagger N_G$ = order of Gaussian integration

Table 4 : MLPG1 values of deflection and slope for models with various nodal arrangements

Basis function		Number of nodes in the model					
		5*	9*	17	33	65	129
cubic	$w_{(\max)}/w_{(\max)\mathrm{Exact}}$	1.0252	1.0846	0.9946	0.9831	0.9871	0.9830
	$\theta_{(\max)}/\theta_{(\max)\mathrm{Exact}}$	1.0198	1.1986	0.9934	0.9831	0.9871	0.9826
quartic	$w_{(\max)}/w_{(\max)\mathrm{Exact}}$	0.9923	1.2624	1.1818	0.9982	0.9992	1.0120
	$\theta_{(\max)}/\theta_{(\max)\mathrm{Exact}}$	1.0333	1.5309	1.1833	1.0012	0.9975	1.0126

* $(R_j\ /\ l) = 3.5$. $w_{(\max)\mathrm{Exact}} = P(4l)^3/48EI$; $\theta_{(\max)\mathrm{Exact}} = \theta|_{x=0} = \theta|_{x=4l} = P(4l)^2/16EI$.

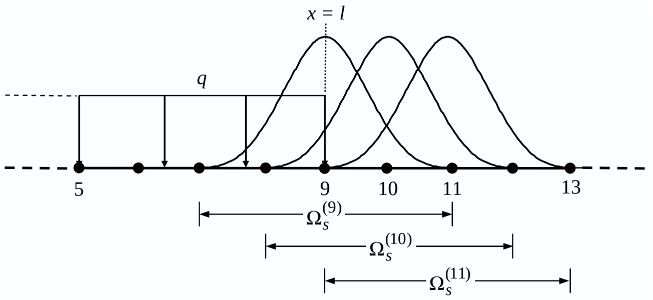


Figure 12 : Localized region of cantilever beam problem

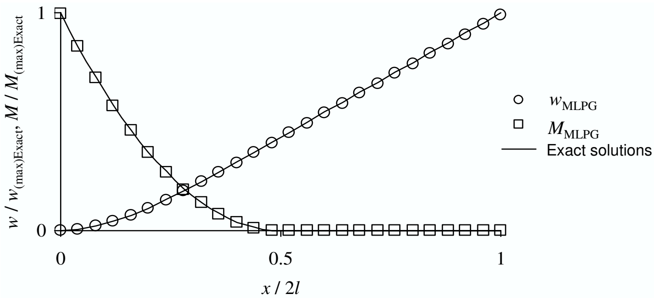


Figure 13 : MLPG1 and exact solutions for the cantilever beam with a UDL on a portion of the beam

in the localized region. For example, for node 9, the distributed load, q, extends only over half of the Ω_s. Similarly, for node 10, the loading spans a small portion of Ω_s, and for node 11, the distributed loading makes no contribution to the $\mathbf{f}^{(\mathrm{node})}$ vector. Thus, to evaluate the $\mathbf{f}^{(\mathrm{node})}$ contributions for each of these nodes, the extent of the region of the integration needs to be calculated carefully. Problem specifics used to analyze this problem are presented in Table 5. The MLPG1 and exact solutions for deflection and moment of the 65-node model are compared in Figure 13. The MLPG1 yielded very accurate results for both the primary and secondary variables.

11.3 Continuous beams

The MLPG1 was then applied to a continuous beam problem to evaluate its effectiveness. A continuous beam with one additional support along the interior of the beam (shown in Figure 14) is considered. The additional support is treated by the penalty terms in the weak form.

The problem specifics used for the continuous beam are presented in Table 6. The MLPG1 and exact solutions for deflection, slope, and moment obtained from the 65-node model are shown in Figure 15. The MLPG1 obtained very accurate results for both the primary and secondary

Table 5 : **Problem parameters – cantilever beam with a UDL on a portion of the beam**

Model	Basis used	$\lambda_j(x)$	$N_G^\dagger$	$\chi_i^{(w)}(x)$	(R_o/l)	(R_j/l)
65-node; uniform spacing; full beam	Quartic	$\alpha = 3$	20	$\beta = 4$	$2\Delta x$	$8\Delta x$

$^\dagger N_G$ = order of Gaussian integration

Table 6 : **Problem parameters – continuous beam**

Model	Basis used	$\lambda_j(x)$	$N_G^\dagger$	$\chi_i^{(w)}(x)$	(R_o/l)	(R_j/l)
65-node; uniform spacing; full beam	Quartic	$\alpha = 3$	20	$\beta = 4$	$2\Delta x$	$8\Delta x$

$^\dagger N_G$ = order of Gaussian integration

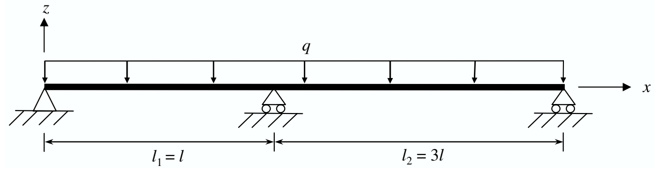
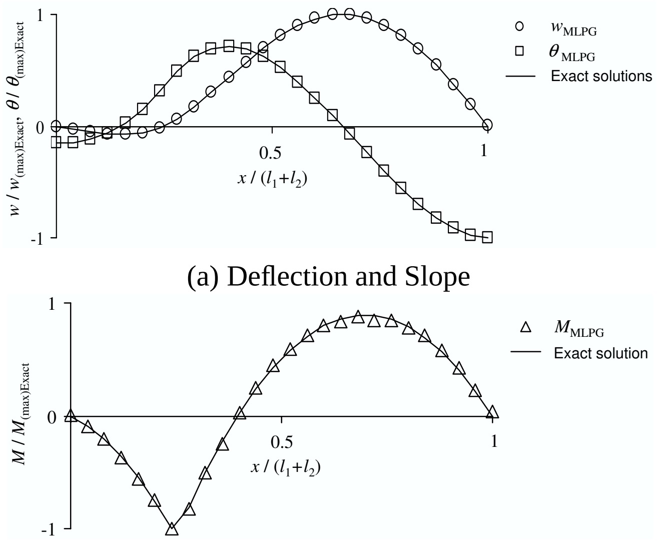


Figure 14 : Continuous beam subjected to a uniformly distributed load



(a) Deflection and Slope

(b) Moment

Figure 15 : MLPG1 and exact solutions for primary and secondary variables of a continuous beam subjected to a uniformly distributed load

variables and handled the discontinuity caused by the additional support well.

12 Concluding Remarks

Two variations of the MLPG formulation were presented for bending of beams – C^1 problems. A local weak form (LWF) was developed from the classical weighted-residual form of the fourth order governing differential equation. The generalized moving least squares interpolation scheme was used to develop the trial functions for the primary variables, the deflections and slopes. The test functions were chosen from a different space than the trial functions, making the method a Petrov-Galerkin method. The Petrov-Galerkin implementation of the method is much more computationally efficient than the previously reported Galerkin implementation for beam problems.

The MLPG method was developed with two classes of test functions – power functions or spline functions (termed as MLPG1) and a linear test function (termed as MLPG5). The MLPG5 is a computationally efficient method as the domain integrals in the weak form are eliminated by the choice of linear test functions. A range of trial functions could be developed using different basis and weight functions. If a proper balance between the basis and weight functions is met, the MLPG method is capable of yielding very accurate results. For example, when a lower order basis function is used, weight functions with higher order continuity are needed to provide smooth trial functions.

The MLPG method was evaluated by applying the formulation to a variety of patch tests. For all trial functions considered, for higher order spline and power test functions, and for a linear test function, the method reproduced exact solutions to machine accuracy, thus passing all the patch tests. Usable ranges of the method's user-controlled parameters were determined. The use of higher order basis and smooth weight functions for the development of the trial functions and higher order power test functions yield accurate results for MLPG1 and hence are recommended. While the trial functions must have a large enough extent to collectively cover the whole domain of the problem, the domains of the test functions need not overlap when the order of the basis function is equal to or greater than that required by the problem. However, overlapping test function domains are recommended for the beam problems with the MLPG method. For mixed boundary value problems, deflections, slopes, moments, and shear forces were calculated to the same accuracy by the MLPG method without the use of elaborate post-processing techniques. Problems with load discontinuities require special care – when a reasonable number of nodes are used around the location of the discontinuity, the MLPG1 yields very accurate results. The method was easily extended to continuous beam problems. The MLPG method with linear test functions (MLPG5) needs further study for problems with load discontinuities.

References

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

Atluri, S. N. and Shen, S. (2002a): *The Meshless Local Petrov-Galerkin (MLPG) Method*, Tech Science Press, Encino, CA.

Atluri, S. N. and Shen, S. (2002b): "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. 11-51.

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

Belytschko, T., Lu, Y. Y., and Gu, L., "Element-free Galerkin methods," *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 229-256.

Donning, B. M. and Liu, W. K. (1998): "Meshless methods for shear-deformable beams and plates," *Computer Methods in Applied Mechanics and Engineering*, Vol. 152, pp. 47-71.

Gu, Y. T. and Liu, G. R. (2001): "A local point interpolation method for static and dynamic analysis of thin beams," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, pp. 5515-5528.

Krysl, P. and Belytschko, T. (1995): "Analysis of thin plates by the element-free Galerkin method," *Computational Mechanics*, Vol. 17, pp. 26-35.

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

Phillips, D. R. and Raju, I. S. (2002): "Meshless local Petrov-Galerkin method for bending problems," NASA TM-2002-211936.

Raju, I. S. and Phillips, D. R. (2002a): "A local coordinate approach in the MLPG method for beam problems," NASA TM-2002-211463.

Raju, I. S. and Phillips, D. R. (2002b): "A meshless local Petrov-Galerkin method for Euler-Bernoulli beam problems," Proceedings of the ICES '02 conference, Reno, Nevada, July 31 – August 2, 2002, Paper No. 139.

Appendix A: Explicit Expressions For the Derivatives of the Shape Functions

This appendix presents the explicit expressions for the first, second, and third derivatives of the shape functions used for 1-D C^1 problems. The detailed derivation of these derivatives may be found in Appendix A of Phillips and Raju (2002). The shape functions (from Eq. 33) are

$$\begin{aligned}\psi_j^{(w)}(x) &= \sum_{g=1}^{m} p_g(\xi_j)\left[[\mathbf{A}]^{-1}[\mathbf{P}]^T[\boldsymbol{\lambda}]\right]_{gj} \text{ and}\\ \psi_j^{(\theta)}(x) &= \sum_{g=1}^{m} p_g(\xi_j)\left[[\mathbf{A}]^{-1}[\mathbf{P}_x]^T[\boldsymbol{\lambda}]\right]_{gj},\end{aligned} \tag{33}$$

or

$$\psi_j^{(w)} = \sum_{g=1}^{m} p_g\left[\,[\mathbf{A}]^{-1}[\mathbf{B}_w]\,\right]_{gj} \tag{52a}$$

and

$$\psi_j^{(\theta)} = \sum_{g=1}^{m} p_g\left[\,[\mathbf{A}]^{-1}[\mathbf{B}_\theta]\,\right]_{gj}, \tag{52b}$$

where

$$\left[\;[\mathbf{B}_w]\quad[\mathbf{B}_\theta]\;\right] = \left[\;[\mathbf{P}]^T[\boldsymbol{\lambda}]\quad[\mathbf{P}_\theta]^T[\boldsymbol{\lambda}]\;\right]. \tag{53}$$

The first derivatives of ψ_j are

$$\begin{aligned}\frac{d\psi_j^{(w)}}{dx} = \psi_{j,x}^{(w)} = \sum_{g=1}^{m}\Big\{ & p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]\right)_{gj} \\ & + p_g\left([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,x} + [\mathbf{A}]^{-1}_{,x}[\mathbf{B}_w]\right)_{gj}\Big\}\end{aligned} \tag{54a}$$

and

$$\begin{aligned}\psi_{j,x}^{(\theta)} = \sum_{g=1}^{m}\Big\{ & p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]\right)_{gj} \\ & + p_g\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,x} + [\mathbf{A}]^{-1}_{,x}[\mathbf{B}_\theta]\right)_{gj}\Big\},\end{aligned} \tag{54b}$$

where

$$[\mathbf{A}]^{-1}_{,x} = -[\mathbf{A}]^{-1}[\mathbf{A}]_{,x}[\mathbf{A}]^{-1}. \tag{55}$$

The second derivatives are

$$\begin{aligned}\psi_{j,,x}^{(w)} = \sum_{g=1}^{m}\Big\{ & p_{g,,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]\right)_{gj} \\ & + 2p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,x} + [\mathbf{A}]^{-1}_{,x}[\mathbf{B}_w]\right)_{gj} \\ & + p_g\left([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,,x} + 2[\mathbf{A}]^{-1}_{,x}[\mathbf{B}_w]_{,x} + [\mathbf{A}]^{-1}_{,,x}[\mathbf{B}_w]\right)_{gj}\Big\}\end{aligned} \tag{56a}$$

and

$$\begin{aligned}\psi_{j,,x}^{(\theta)} = \sum_{g=1}^{m}\Big\{ & p_{g,,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]\right)_{gj} \\ & + 2p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,x} + [\mathbf{A}]^{-1}_{,x}[\mathbf{B}_\theta]\right)_{gj} \\ & + p_g\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,,x} + 2[\mathbf{A}]^{-1}_{,x}[\mathbf{B}_\theta]_{,x} + [\mathbf{A}]^{-1}_{,,x}[\mathbf{B}_\theta]\right)_{gj}\Big\},\end{aligned} \tag{56b}$$

where

$$[\mathbf{A}]^{-1}_{,,x} = -[\mathbf{A}]^{-1}[\mathbf{A}]_{,,x}[\mathbf{A}]^{-1} - 2[\mathbf{A}]^{-1}[\mathbf{A}]_{,x}[\mathbf{A}]^{-1}_{,x}. \tag{57}$$

The third derivatives are

$$\begin{aligned}\psi_{j,,,x}^{(w)} = \sum_{g=1}^{m}\Big\{ & p_{g,,,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]\right)_{gj} \\ & + 3p_{g,,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,x} + [\mathbf{A}]^{-1}_{,x}[\mathbf{B}_w]\right)_{gj} \\ & + 3p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,,x} + 2[\mathbf{A}]^{-1}_{,x}[\mathbf{B}_w]_{,x} + [\mathbf{A}]^{-1}_{,,x}[\mathbf{B}_w]\right)_{gj} \\ & + p_g\big([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,,,x} + 3[\mathbf{A}]^{-1}_{,x}[\mathbf{B}_w]_{,,x} + 3[\mathbf{A}]^{-1}_{,,x}[\mathbf{B}_w]_{,x} \\ & \quad + [\mathbf{A}]^{-1}_{,,,x}[\mathbf{B}_w]\big)_{gj}\Big\}\end{aligned} \tag{58a}$$

and

$$\begin{aligned}\psi_{j,,,x}^{(\theta)} = \sum_{g=1}^{m}\Big\{ & p_{g,,,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]\right)_{gj} \\ & + 3p_{g,,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,x} + [\mathbf{A}]^{-1}_{,x}[\mathbf{B}_\theta]\right)_{gj} \\ & + 3p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,,x} + 2[\mathbf{A}]^{-1}_{,x}[\mathbf{B}_\theta]_{,x} + [\mathbf{A}]^{-1}_{,,x}[\mathbf{B}_\theta]\right)_{gj} \\ & + p_g\big([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,,,x} + 3[\mathbf{A}]^{-1}_{,x}[\mathbf{B}_\theta]_{,,x} + 3[\mathbf{A}]^{-1}_{,,x}[\mathbf{B}_\theta]_{,x} \\ & \quad + [\mathbf{A}]^{-1}_{,,,x}[\mathbf{B}_\theta]\big)_{gj}\Big\},\end{aligned} \tag{58b}$$

where

$$\begin{aligned}[\mathbf{A}]^{-1}_{,,,x} = & -[\mathbf{A}]^{-1}[\mathbf{A}]_{,,,x}[\mathbf{A}]^{-1} - 3[\mathbf{A}]^{-1}[\mathbf{A}]_{,,x}[\mathbf{A}]^{-1}_{,x} \\ & - 3[\mathbf{A}]^{-1}[\mathbf{A}]_{,x}[\mathbf{A}]^{-1}_{,,x}.\end{aligned} \tag{59}$$

Appendix B: Special Cases of (R_o/l)

The purpose of this appendix is to discuss special choices of (R_o/l) that should be avoided when using the MLPG5. While it is convenient to choose the same value of (R_o/l) for all nodes of a model, the MLPG algorithm does not require that the same value of (R_o/l) be used at all nodes. In fact, the algorithm passed all the patch tests when different values of (R_o/l) were chosen for each node. Raju and Phillips (2002b) suggested the use of (Δx) for nodes

2 and (N-1) and $(2\Delta x)$ for all other nodes for the MLPG1. With this choice, the boundary terms in Eq. (13e) for nodes 2 and (N-1) are identically zero, thus simplifying the computations. While this choice (of different (R_o/l) for nodes 2 and (N-1)) is beneficial for the MLPG1, this choice should be avoided for the MLPG5. In the MLPG5, when (R_o/l) is chosen as (Δx) for nodes 2 and (N-1) and as $(2\Delta x)$ for all other nodes, the equations for w and θ developed for nodes 1 and 2 are identical. (As the Γ_s points for these nodes, x = 0 and x = 2Δx, are identical, the values of the test functions are identical, and their derivatives are also identical at these points.) Similarly, the equations for nodes (N-1) and N are identical, leading to a singular "stiffness" matrix. (In the MLPG1, the Ω_s integrals in Eq. (13d) are nonzero, and hence the equations developed for nodes 1 and 2 (and nodes (N-1) and N) are not identical.) Thus in the MLPG5, any choice of (R_o/l) values that leads to identical Γ_s points for any two nodes (for example, $(4\Delta x)$ for node 1 and $(2\Delta x)$ for node 3) should be avoided as such choices lead to singular "stiffness" matrices.

Note that in the MLPG5, other variations of the test function of Eq. (42), such as $v_i = x - x_i$, may avoid the singularity in the w terms, but still lead to singularity in the θ terms of nodes 1 and 2 (and nodes (N-1) and N). Thus, when different (R_o/l) values are used for different nodes, the singularity of the system matrix cannot be avoided for certain choices of (R_o/l). However, the singularity situation can be completely avoided if one assumes the same value of (R_o/l) at all nodes and hence is recommended for both the MLPG1 and the MLPG5.