MESHLESS LOCAL PETROV-GALERKIN EULER-BERNOULLI BEAM PROBLEMS: A RADIAL BASIS FUNCTION APPROACH

I. S. Raju[*], D. R. Phillips[†], T. Krishnamurthy[‡]
NASA Langley Research Center, Hampton, Virginia 23681, U.S.A.

Abstract

A radial basis function implementation of the meshless local Petrov-Galerkin (MLPG) method is presented to study Euler-Bernoulli beam problems. Radial basis functions, rather than generalized moving least squares (GMLS) interpolations, are used to develop the trial functions. This choice yields a computationally simpler method as fewer matrix inversions and multiplications are required than when GMLS interpolations are used. Test functions are chosen as simple weight functions as in the conventional MLPG method. Compactly and non-compactly supported radial basis functions are considered. The non-compactly supported cubic radial basis function is found to perform very well. Results obtained from the radial basis MLPG method are comparable to those obtained using the conventional MLPG method for mixed boundary value problems and problems with discontinuous loading conditions.

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.[1-4] 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.[5, 6] However, a comparatively limited amount of work [4, 7-10] is reported on the more complicated C^1 problems. Atluri *et al.* [4] present an analysis of thin beam problems using a Galerkin implementation of the MLPG method. In reference 4, a generalized moving least squares (GMLS) approximation is used to construct the trial functions, and the test functions are chosen from the same space. In references 11-14, a meshless Petrov-Galerkin implementation of the MLPG method is presented; the GMLS approximation is used to construct the trial functions, and the test functions are chosen from a different space. Closer scrutiny of these formulations shows that a large number of calculations are required to compute the first and second order derivatives of the moving least squares (MLS) trial functions. Hence, a computationally efficient alternative to the MLS trial functions is preferred.

This paper demonstrates the use of radial basis interpolation functions in the meshless local Petrov-Galerkin formulation for beam problems. The radial basis functions are simple, and the evaluation of the derivatives is simpler than for the traditional MLS approximations. In the present radial basis MLPG (RPG) formulation, simple weight functions are chosen as test functions, and Gaussian quadrature is used to integrate the weak form. The effectiveness of the RPG method is evaluated by applying the formulation to a variety of patch test and mixed boundary value problems.

The outline of the paper is as follows: First, the moving least squares interpolation used in the conventional MLPG method is discussed as motivation for finding a more computationally efficient alternative. Next, an overview of radial basis functions (RBF) for C^0 problems is presented; the shape functions obtained from radial basis interpolation are derived, and the shape functions obtained when polynomial basis functions are included in the interpolation are derived. The development of these radial basis shape functions is then expanded and repeated for beam problems. The system of algebraic equations developed from the local weak form of the governing differential equation and the chosen trial and test functions is presented. Patch test problems are used to validate the RPG method for different choices of radial basis function. Then, the RPG method is applied to mixed boundary value problems. Finally, the method is applied to a problem with discontinuous loading conditions.

Interpolation Schemes

In this section, the moving least squares interpolation scheme used in the conventional MLPG

[*] Structures and Materials Competency, Senior Technologist, Fellow AIAA
[†] Lockheed Martin Space Operations
[‡] Analytical and Computational Methods Branch, Member AIAA

method is discussed first. Then, two interpolation schemes involving radial basis functions (RBF) are presented. In the first scheme, radial basis functions alone are used to construct the shape functions. The second scheme is a hybrid that uses both radial basis functions and polynomial basis functions to construct the shape functions.

The Moving Least Squares Interpolation

A moving least squares (MLS) interpolation is a scheme that passes a smooth function through an assumed set of fictitious nodal values. The interpolation is performed such that the least squares error between the function and the nodal values is a minimum.[1, 2] A schematic of the MLS interpolation is presented in Figure 1.

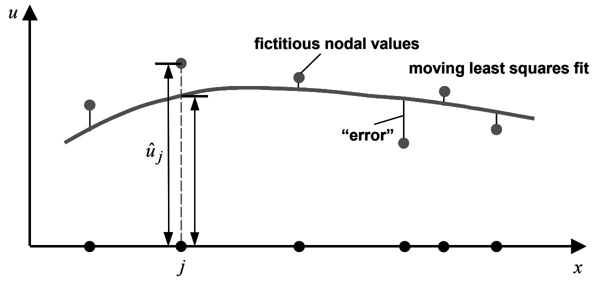


Figure 1: Moving Least Squares (MLS) interpolation

C^0 MLS Shape Functions: In one-dimensional problems, the C^0 MLS shape functions are given by

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

where $p(x)$ is a polynomial basis function, and

$$\begin{aligned}[\mathbf{A}] &= [\mathbf{P}]^{\mathrm{T}}[\boldsymbol{\lambda}][\mathbf{P}] \\ [\mathbf{B}] &= [\mathbf{P}]^{\mathrm{T}}[\boldsymbol{\lambda}].\end{aligned} \tag{2}$$

In Equation (2), $[\mathbf{P}]$ is an (n, m) matrix, and $[\boldsymbol{\lambda}]$ is a diagonal (n, n) matrix defined as

$$[\mathbf{P}] = \left[\mathbf{p}^{\mathrm{T}}(x_1) \quad \mathbf{p}^{\mathrm{T}}(x_2) \quad \ldots \quad \mathbf{p}^{\mathrm{T}}(x_n)\right]^{\mathrm{T}} \tag{3}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1(x) & & & \\ & \lambda_2(x) & & \\ & & \ddots & \\ & & & \lambda_n(x) \end{bmatrix}, \tag{4}$$

where

$$\begin{aligned}[\mathbf{p}^{\mathrm{T}}(x)] &= \left[p_1(x) \quad p_2(x) \quad p_3(x) \quad \ldots \quad p_m(x)\right] \\ &= \left[1 \quad x \quad x^2 \quad \ldots \quad x^{m-1}\right],\end{aligned} \tag{5}$$

and $\lambda_j(x)$, $j = 1\ldots n$, is a weight function. The first derivative of these shape functions is all that is required by the MLPG method and is given by [13]

$$\begin{aligned}\phi_{j,x} = \sum_{g=1}^{m} \Big\{ & p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}]\right)_{gj} \\ & + p_g\left[[\mathbf{A}]^{-1}[\mathbf{B}]_{,x} + [\mathbf{A}]^{-1}_{,x}[\mathbf{B}]\right]_{gj} \Big\},\end{aligned} \tag{6}$$

where $(\ )_{,x} \equiv d(\ )/dx$, and

$$[\mathbf{A}]^{-1}_{,x} = -[\mathbf{A}]^{-1}[\mathbf{A}]_{,x}[\mathbf{A}]^{-1}. \tag{7}$$

C^1 MLS Shape Functions: The C^1 GMLS shape functions for deflection, w, and slope, θ, in 1-D are given, using the local coordinate approach of references 12 and 14, by

$$\begin{aligned}\psi_j^{(w)}(x) &= \sum_{g=1}^{m} p_g(\xi_j)\left[[\mathbf{A}]^{-1}[\mathbf{P}]^{\mathrm{T}}[\boldsymbol{\lambda}]\right]_{gj} \quad \text{and} \\ \psi_j^{(\theta)}(x) &= \sum_{g=1}^{m} p_g(\xi_j)\left[[\mathbf{A}]^{-1}[\mathbf{P}_x]^{\mathrm{T}}[\boldsymbol{\lambda}]\right]_{gj},\end{aligned} \tag{8}$$

or

$$\psi_j^{(w)} = \sum_{g=1}^{m} p_g\left[[\mathbf{A}]^{-1}[\mathbf{B}_w]\right]_{gj} \tag{9a}$$

and

$$\psi_j^{(\theta)} = \sum_{g=1}^{m} p_g\left[[\mathbf{A}]^{-1}[\mathbf{B}_\theta]\right]_{gj}, \tag{9b}$$

where

$$\left[[\mathbf{B}_w] \quad [\mathbf{B}_\theta]\right] = \left[[\mathbf{P}]^{\mathrm{T}}[\boldsymbol{\lambda}] \quad [\mathbf{P}_x]^{\mathrm{T}}[\boldsymbol{\lambda}]\right]. \tag{10}$$

In Equations (8) – (10), $p(x)$ is a polynomial basis function, and

$$[\mathbf{A}] = [\mathbf{P}]^{\mathrm{T}}[\boldsymbol{\lambda}][\mathbf{P}] + [\mathbf{P}_x]^{\mathrm{T}}[\boldsymbol{\lambda}][\mathbf{P}_x]. \tag{11}$$

In Equation (11), [**P**] and [$\mathbf{P}_x$] are (n, m) matrices, and [**λ**] is a diagonal (n, n) matrix defined as

$$[\mathbf{P}] = \left[\mathbf{p}^{\mathrm{T}}(\xi_1) \;\; \mathbf{p}^{\mathrm{T}}(\xi_2) \;\; \ldots \;\; \mathbf{p}^{\mathrm{T}}(\xi_n)\right]^{\mathrm{T}} \tag{12a}$$

$$[\mathbf{P}_x] = \left[\mathbf{p}_x^{\mathrm{T}}(\xi_1) \;\; \mathbf{p}_x^{\mathrm{T}}(\xi_2) \;\; \ldots \;\; \mathbf{p}_x^{\mathrm{T}}(\xi_n)\right]^{\mathrm{T}}, \tag{12b}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1(x) & & & \\ & \lambda_2(x) & & \\ & & \ddots & \\ & & & \lambda_n(x) \end{bmatrix}, \tag{13}$$

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

$$\mathbf{p}^{\mathrm{T}}(\xi) = \left\lfloor 1, \;\; \xi, \;\; \xi^2, \;\; \ldots \;\; \xi^{m-1} \right\rfloor, \text{ and} \tag{14a}$$

$$\mathbf{p}_x^{\mathrm{T}}(\xi) = \frac{d\mathbf{p}^{\mathrm{T}}(\xi)}{dx} = \left[0, \;\; 1, \;\; 2\xi, \;\; \ldots \;\; (m-1)\xi^{m-2}\right] \tag{14b}$$

as

$$\frac{d}{dx}(\;) = \frac{d}{d\xi}(\;). \tag{15}$$

In C[1] problems, the first, second, and third derivatives are required by the MLPG method. The first derivatives of ψ_j are

$$\frac{d\psi_j^{(w)}}{dx} = \psi_{j,x}^{(w)} = \sum_{g=1}^{m} \Big\{ p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]\right)_{gj} + p_g\left([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,x} + [\mathbf{A}]_{,x}^{-1}[\mathbf{B}_w]\right)_{gj} \Big\} \tag{16a}$$

and

$$\psi_{j,x}^{(\theta)} = \sum_{g=1}^{m} \Big\{ p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]\right)_{gj} + p_g\left([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,x} + [\mathbf{A}]_{,x}^{-1}[\mathbf{B}_\theta]\right)_{gj} \Big\}, \tag{16b}$$

where

$$[\mathbf{A}]_{,x}^{-1} = -[\mathbf{A}]^{-1}[\mathbf{A}]_{,x}[\mathbf{A}]^{-1}. \tag{7}$$

The second derivatives and third derivatives involve considerably more complex expressions containing $[\mathbf{A}]^{-1}$, $[\mathbf{A}]_{,x}^{-1}$, $[\mathbf{A}]_{,xx}^{-1}$, etc., and the detailed derivations are given in reference 13 and are not repeated here. Note how the additional degree of freedom (θ) and the need for the higher order derivatives yield very complicated expressions for these derivatives. For thin plate problems (2-D C[1] problems), these derivatives become even more complicated. (Expressions for the partial derivatives for the 2-D shape function may be found in reference 10.) Therefore, a more computationally efficient method for approximating the trial functions in the MLPG method is sought. Radial basis functions appear to be a good candidate for achieving such a purpose because the shape functions obtained from radial basis interpolation are simpler than the shape functions presented above for the MLS. More importantly, the derivatives of the radial basis shape functions are simple and involve considerably fewer matrix inverse and multiplication operations than the derivatives of the MLS shape functions. The radial basis functions (RBF) are discussed next.

Radial Basis Function

The radial basis formulation provides a continuous interpolating function for $u(\mathbf{x})$ as a linear combination of radial functions.[15] The interpolating function is given by

$$u(\mathbf{x}) = \sum_{j=1}^{N} R_j(\mathbf{x})\, a_j\,, \tag{17}$$

where $R_j(\mathbf{x})$, the radial basis functions (RBF), are functions at each of the N scattered points, and a_j are the unknown coefficients, j = 1, 2, …, N. The RBF, $R_j(\mathbf{x})$, are functions of distance r_j and are defined as

$$R_j(\mathbf{x}) = R_j(r_j)\,. \tag{18}$$

The radial distance, r_j, in Cartesian coordinates can be expressed as

$$r_j(\mathbf{x}) = \sqrt{\left(x - x_j\right)^2 + \left(y - y_j\right)^2}\,, \tag{19}$$

where x_j and y_j are the coordinates of node j. Equation (17) can be written in matrix form for C[0] problems as

$$u(\mathbf{x}) = \mathbf{R}^{\mathrm{T}}(\mathbf{x})\,\mathbf{a}\,, \tag{20}$$

where

$$\begin{aligned} \mathbf{R}^{\mathrm{T}}(\mathbf{x}) &= \left[R_1(\mathbf{x}), \;\; R_2(\mathbf{x}), \;\; R_3(\mathbf{x}), \;\; \ldots, \;\; R_N(\mathbf{x})\right] \\ \mathbf{a} &= \left\{a_1, \;\; a_2, \;\; a_3, \;\; \ldots, \;\; a_N\right\}^{\mathrm{T}}. \end{aligned} \tag{21}$$

Forcing the interpolation of Equation (17) to pass through the N scattered points, a set of equations to determine the coefficients a_j can be written as

$$\mathbf{R}_\mathrm{B}\,\mathbf{a} = \mathbf{u}\,, \tag{22}$$

where

$$\mathbf{u}^\mathrm{T} = [u_1,\ u_2,\ u_3,\ \cdots,\ u_N] \tag{23}$$

$$\mathbf{R}_\mathrm{B} = \begin{bmatrix} R_1(\mathbf{x}_1) & R_2(\mathbf{x}_1) & \cdots & R_N(\mathbf{x}_1) \\ R_1(\mathbf{x}_2) & R_2(\mathbf{x}_2) & \cdots & R_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(\mathbf{x}_N) & R_2(\mathbf{x}_N) & \cdots & R_N(\mathbf{x}_N) \end{bmatrix}. \tag{24}$$

Note that $\mathbf{R}_\mathrm{B}$ is an (N, N) matrix. Here, $\mathbf{u}$ $(u_j,\ j = 1, 2, \ldots, N)$ are the nodal values of u at the N scattered points. The unknown coefficients in Equation (22) can be obtained as

$$\mathbf{a} = \mathbf{R}_\mathrm{B}^{-1}\,\mathbf{u}\,. \tag{25}$$

The interpolating function for $u(\mathbf{x})$ in Equation (20) can now be rewritten as

$$u(\mathbf{x}) = \mathbf{R}^\mathrm{T}(\mathbf{x})\mathbf{R}_\mathrm{B}^{-1}\mathbf{u} = \sum_{j=1}^{N} \varphi_j(\mathbf{x}) u_j\,. \tag{26}$$

The nodal shape functions are then

$$\begin{aligned} \varphi(\mathbf{x}) &= \mathbf{R}^\mathrm{T}(\mathbf{x})\mathbf{R}_\mathrm{B}^{-1} \\ &= [\varphi_1(\mathbf{x}),\ \varphi_2(\mathbf{x}),\ \varphi_3(\mathbf{x}),\ \ldots,\ \varphi_N(\mathbf{x})] \end{aligned} \tag{27}$$

or

$$\varphi_l(\mathbf{x}) = \sum_{k=1}^{N} R_k(\mathbf{x}) \xi_{kl}\,, \tag{28}$$

where ξ_{kl} are the elements of the matrix $\mathbf{R}_\mathrm{B}^{-1}$. The shape function $\varphi_j(\mathbf{x})$ obtained through the above procedure satisfies the Kronecker Delta property *only at the nodes* [5], i.e.,

$$\begin{aligned} &\varphi_j(\mathbf{x}_j) \equiv 1, \text{ and} \\ &\varphi_j(\mathbf{x}_k) \equiv 0\,. \end{aligned} \tag{29}$$

Note that the shape functions in Equation (28) also satisfy the property

$$\sum_{j=1}^{N} \varphi_j(\mathbf{x}) = 1 \tag{30}$$

at the nodes exactly. As the number of non-nodal interpolation points, M, is increased, the shape functions in Equation (28) satisfy

$$\lim_{M\to\infty} \sum_{l=1}^{M} \varphi_l(\mathbf{x}) = 1\,. \tag{31}$$

Unlike in the moving least squares (MLS) method, the derivatives of the shape functions are easy to evaluate using Equation (28) as

$$\begin{aligned} \frac{\partial \varphi_l(\mathbf{x})}{\partial x} &= \sum_{k=1}^{N} \frac{\partial R_k}{\partial x} \cdot \xi_{kl} \\ \frac{\partial \varphi_l(\mathbf{x})}{\partial y} &= \sum_{k=1}^{N} \frac{\partial R_k}{\partial y} \cdot \xi_{kl}\,, \end{aligned} \tag{32}$$

where

$$\frac{\partial R_k}{\partial x} = \frac{\partial R_k}{\partial r_k}\frac{\partial r_k}{\partial x} \quad ; \quad \frac{\partial R_k}{\partial y} = \frac{\partial R_k}{\partial r_k}\frac{\partial r_k}{\partial y} \tag{33}$$

with

$$\frac{\partial r_k}{\partial x} = \frac{x - x_k}{r_k} \quad ; \quad \frac{\partial r_k}{\partial y} = \frac{y - y_k}{r_k} \tag{34}$$

and

$$r_k(\mathbf{x}) = \sqrt{(x - x_k)^2 + (y - y_k)^2}\,. \tag{35}$$

Some of the classical radial functions used in multivariate interpolation are presented in Table 1. Note that the shape parameter c in the radial basis functions in Table 1 is user-defined and can be adjusted

Table 1: Classical radial basis functions [16]

Classical RBF	Equation
Linear	$R(r) = cr$
Cubic	$R(r) = (r + c)^3$
Thin plate spline	$R(r) = r^2 \log(cr^2)$
Gaussian	$R(r) = e^{-cr^2}$
Multiquadric	$R(r) = \sqrt{r^2 + c^2}$

to fit the required data. The classical radial functions have two limitations; (1) the matrix $\mathbf{R}_\mathrm{B}$ may not be positive definite, and (2) the functions do not possess local support, i.e, changing the location of the center (x_j, y_j) in Equation (19) affects the entire interpolation. to overcome these limitations, compactly supported positive definite radial functions were proposed.[17] These functions were derived using a constraint to guarantee positive definiteness of the interpolation matrix, $\mathbf{R}_\mathrm{B}$. The compact support of these functions guarantees that every point in a compact radial basis interpolation domain does not necessarily have an affect on every other point in the domain. The compact RBF adapted and used here are

Compact-I:

$$R_j(\mathbf{x}) = \begin{cases} (1-t)^5(8+40t+48t^2 \\ \qquad +25t^3+5t^4), \ 0 \le t \le 1 \\ 0 \qquad\qquad\qquad\qquad t > 1 \end{cases} \tag{36}$$

Compact-II:

$$R_j(\mathbf{x}) = \begin{cases} (1-t)^6(6+36t+82t^2 \\ \qquad +72t^3+30t^4+5t^5), \ 0 \le t \le 1 \\ 0 \qquad\qquad\qquad\qquad t > 1 \end{cases} \tag{37}$$

where $(t = r / s_j)$, and s_j is the radius of the domain of compact support. The shape functions (Equation (28)) obtained from the Compact-I and Compact-II functions possess all the properties in Equations (29) – (31). Several other forms of compact support functions can be found in references 16 and 17.

Hybrid Radial Basis Function

The classical radial basis functions shown in Table 1 and the compactly supported functions in Equations (36) and (37) cannot represent polynomial solutions exactly [18, 19]; they can represent the polynomial values only at the N scattered points. Figures 2 and 3 show radial basis interpolations obtained from the compact RBF in Equation (36) with $s_j = 0.6$ using 5 nodes in the interval $x_1 \le x \le x_2$, where x_1 = -1 and x_2 = +1. Figures 2a and 3a show the RBF values that correspond to a constant polynomial,

$$f = 1, \tag{38}$$

and a linear polynomial,

$$f = 1 + x, \tag{39}$$

respectively. The function values are evaluated at the 5 nodes and are prescribed as u_j's in Equation (26). The values of u are evaluated using Equation (26) at 200 points in the interval $x_1 \le x \le x_2$ and are plotted in Figures 2b and 3b. As seen from these figures, the compact RBF recovers the polynomial values (Equations (38) and (39)) exactly only at the 5 nodal points, and elsewhere the values of the polynomials in Equations (38) and (39) are not recovered.

In order to improve the polynomial accuracy of the solutions, Powell [15] suggested adding polynomial basis functions to the radial basis functions as

$$u(\mathbf{x}) = \sum_{j=1}^{N} R_j(\mathbf{x})\, a_j + \sum_{k=1}^{m} p_k(\mathbf{x})\, z_k , \tag{40}$$

where R_j, a_j, and N are as in Equation (17), $p(\mathbf{x})$ is the polynomial basis function, z_k are the unkown coefficients associated with the k^th polynomial term, and m is the order of the polynomial basis function. Equation (40) is written in matrix form as [18]

$$\begin{aligned} u(\mathbf{x}) &= \mathbf{R}^\mathrm{T}(\mathbf{x})\,\mathbf{a} + \mathbf{p}^\mathrm{T}(\mathbf{x})\,\mathbf{z} \\ &= \begin{bmatrix} \mathbf{R}^\mathrm{T}(\mathbf{x}) & \mathbf{p}^\mathrm{T}(\mathbf{x}) \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix}, \end{aligned} \tag{41}$$

where $\mathbf{a}$ and $\mathbf{R}^\mathrm{T}(\mathbf{x})$ are as in Equation (21), and

$$\begin{aligned} \mathbf{z} &= \{z_1,\ z_2,\ z_3,\ \ldots,\ z_m\}^\mathrm{T} \\ \mathbf{p}^\mathrm{T}(\mathbf{x}) &= [p_1(\mathbf{x}),\ p_2(\mathbf{x}),\ p_3(\mathbf{x}),\ \ldots,\ p_m(\mathbf{x})]. \end{aligned} \tag{42}$$

The interpolation is forced to pass through the N points, with the constraint,

$$\sum_{j=1}^{N} p_k(\mathbf{x}_j) a_j = 0, \quad k = 1, 2, \ldots, m, \tag{43}$$

imposed to guarantee unique approximation.[18] The set of equations to determine the coefficients a_j and z_k is thus written as

$$\begin{aligned} \begin{bmatrix} \mathbf{R}_\mathrm{B} & \mathbf{P}_\mathrm{B} \\ \mathbf{P}_\mathrm{B}^\mathrm{T} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix} &= \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix}, \text{ or} \\ [\mathbf{G}] \begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix} &= \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix}, \end{aligned} \tag{44}$$

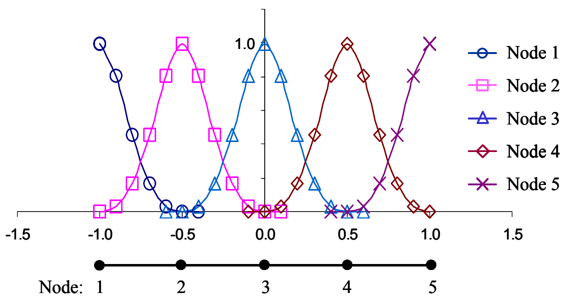


Figure 2a: Radial basis function values that correspond to a constant polynomial

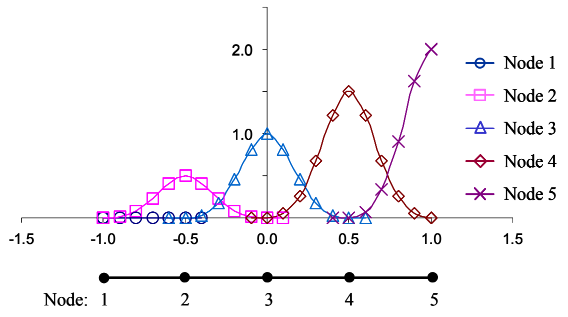


Figure 3a: Radial basis function values that correspond to a linear polynomial

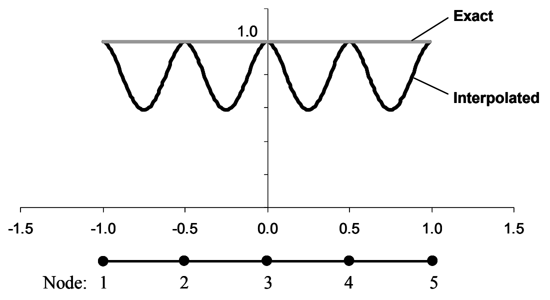


Figure 2b: Interpolated and exact values of a constant polynomial

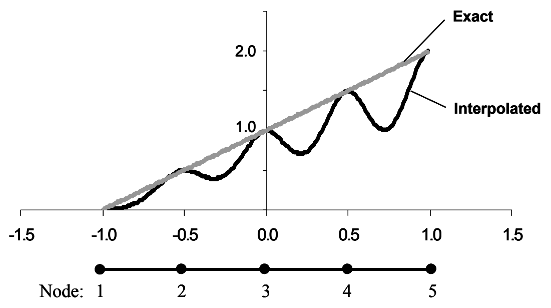


Figure 3b: Interpolated and exact values of a linear polynomial

where $\mathbf{u}^{\mathrm{T}}$ and $\mathbf{R}_{\mathrm{B}}$ are defined in Equations (23) and (24), and

$$\mathbf{P}_{\mathrm{B}} = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_N) & p_2(\mathbf{x}_N) & \cdots & p_m(\mathbf{x}_N) \end{bmatrix}. \tag{45}$$

The unknown coefficients in Equation (44) are obtained as

$$\begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix} = [\mathbf{G}]^{-1} \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix}. \tag{46}$$

The interpolating functions $u(\mathbf{x})$ in Equation (41) can now be rewritten as

$$\begin{aligned} u(\mathbf{x}) &= \begin{bmatrix} \mathbf{R}^{\mathrm{T}}(\mathbf{x}) & \mathbf{p}^{\mathrm{T}}(\mathbf{x}) \end{bmatrix} [\mathbf{G}]^{-1} \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix} \\ &= \sum_{j=1}^{N} \varphi_j(\mathbf{x}) u_j . \end{aligned} \tag{47}$$

The nodal shape functions are then

$$\varphi_l(\mathbf{x}) = \sum_{j=1}^{N} R_j(\mathbf{x}) \gamma_{jl} + \sum_{k=1}^{m} p_k(\mathbf{x}) \gamma_{(N+k)(l)} , \tag{48}$$

where γ are the elements of the matrix $[\mathbf{G}]^{-1}$. The shape functions in Equation (48) possesss the Kronecker Delta and the unity partition properties of Equations (29) and (30). The derivatives of the shape functions are easy to evaluate using Equation (48) as

$$\begin{aligned} \frac{\partial \varphi_l(\mathbf{x})}{\partial x} &= \sum_{j=1}^{N} \frac{\partial R_j(\mathbf{x})}{\partial x} \gamma_{jl} + \sum_{k=1}^{m} \frac{\partial p_k(\mathbf{x})}{\partial x} \gamma_{(N+k)(l)} \\ \frac{\partial \varphi_l(\mathbf{x})}{\partial y} &= \sum_{j=1}^{N} \frac{\partial R_j(\mathbf{x})}{\partial y} \gamma_{jl} + \sum_{k=1}^{m} \frac{\partial p_k(\mathbf{x})}{\partial y} \gamma_{(N+k)(l)} , \end{aligned} \tag{49}$$

where $(\partial R_j / \partial x)$ and $(\partial R_j / \partial y)$ are as in Equations (33) – (35).

Beam Problem Interpolation Schemes

This section presents the interpolation schemes used in the RPG method for beam problems. The shape functions for both the radial basis and hybrid interpolations are derived. These shape functions will be used in the next section in the system of algebraic equations developed from the local weak form of the governing differential equation.

Radial Basis Function

The radial basis functions, $R_j(x)$, are functions of r_j, where in 1-D

$$r_j(x) = x - x_j. \tag{50}$$

In C¹ problems, the deflection, w, and the slope, $\theta = dw/dx$, are both primary variables and degrees of freedom whose continuity need to be satisfied. The interpolating function for $w(x)$ is assumed to be of the form

$$\begin{aligned} w(x) &= R_1(x)a_1 + S_1(x)b_1 + R_2(x)a_2 + S_2(x)b_2 \\ &\quad + \ldots + R_N(x)a_N + S_N(x)b_N \end{aligned}, \tag{51}$$

where a_j and b_j, j=1, 2, …, N, are unknown coefficients, $R_j(x)$ are the radial basis functions, and $S_j(x) = dR_j(x)\,/\,dx$. Because of the direct relationship between the slope and the deflection, the approximating functions for θ cannot be chosen independently from the functions for w, and as in Equation (51), the approximations for θ are written as

$$\begin{aligned} \theta = \frac{dw(x)}{dx} &= \frac{dR_1(x)}{dx}a_1 + \frac{dS_1(x)}{dx}b_1 \\ &\quad + \frac{dR_2(x)}{dx}a_2 + \frac{dS_2(x)}{dx}b_2 + \ldots \\ &\quad + \frac{dR_N(x)}{dx}a_N + \frac{dS_N(x)}{dx}b_N. \end{aligned} \tag{52}$$

In matrix form, Equation (51) is written as

$$w(x) = \mathbf{Q}^{\mathrm{T}}(x)\{\mathbf{c}\}, \tag{53}$$

where

$$\begin{aligned} \mathbf{Q}^{\mathrm{T}}(x) &= \begin{bmatrix} R_1(x) & S_1(x) & R_2(x) & S_2(x) & \ldots & R_N(x) & S_N(x) \end{bmatrix} \\ \{\mathbf{c}\} &= \{a_1 \quad b_1 \quad a_2 \quad b_2 \quad \ldots \quad a_N \quad b_N\}^{\mathrm{T}}. \end{aligned} \tag{54}$$

Similarly, Equation (52) is written as

$$\theta = \frac{d\mathbf{Q}^{\mathrm{T}}(x)}{dx}\{\mathbf{c}\}, \tag{55}$$

where

$$\frac{d\mathbf{Q}^{\mathrm{T}}(x)}{dx} = \begin{bmatrix} \frac{dR_1(x)}{dx} & \frac{dS_1(x)}{dx} & \frac{dR_2(x)}{dx} & \frac{dS_2(x)}{dx} & \ldots & \frac{dR_N(x)}{dx} & \frac{dS_N(x)}{dx} \end{bmatrix}. \tag{56}$$

Forcing the interpolations to pass through N nodal values, the set of equations to estimate the coefficients a_j and b_j is written as

$$\underset{(2N,2N)}{[\mathbf{Q}_{\mathrm{B}}]}\ \underset{(2N,1)}{\{\mathbf{c}\}} = \underset{(2N,1)}{\{\mathbf{d}\}}, \tag{57}$$

where

$$\{\mathbf{d}\}^{\mathrm{T}} = \{w_1 \quad \theta_1 \quad w_2 \quad \theta_2 \quad \ldots \quad w_N \quad \theta_N\} \tag{58}$$

is the vector of nodal values of w and θ at the N nodes, and

$$[\mathbf{Q}_{\mathrm{B}}] = \begin{bmatrix} R_1(x_1) & S_1(x_1) & R_2(x_1) & S_2(x_1) & \ldots & R_N(x_1) & S_N(x_1) \\ \frac{dR_1(x_1)}{dx} & \frac{dS_1(x_1)}{dx} & \frac{dR_2(x_1)}{dx} & \frac{dS_2(x_1)}{dx} & \ldots & \frac{dR_N(x_1)}{dx} & \frac{dS_N(x_1)}{dx} \\ R_1(x_2) & S_1(x_2) & R_2(x_2) & S_2(x_2) & \ldots & R_N(x_2) & S_N(x_2) \\ \frac{dR_1(x_2)}{dx} & \frac{dS_1(x_2)}{dx} & \frac{dR_2(x_2)}{dx} & \frac{dS_2(x_2)}{dx} & \ldots & \frac{dR_N(x_2)}{dx} & \frac{dS_N(x_2)}{dx} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ R_1(x_N) & S_1(x_N) & R_2(x_N) & S_2(x_N) & \ldots & R_N(x_N) & S_N(x_N) \\ \frac{dR_1(x_N)}{dx} & \frac{dS_1(x_N)}{dx} & \frac{dR_2(x_N)}{dx} & \frac{dS_2(x_N)}{dx} & \ldots & \frac{dR_N(x_N)}{dx} & \frac{dS_N(x_N)}{dx} \end{bmatrix}. \tag{59}$$

The unknown coefficients in Equation (57) are obtained as

$$\{\mathbf{c}\} = [\mathbf{Q}_{\mathrm{B}}]^{-1}\{\mathbf{d}\}. \tag{60}$$

The interpolation for w in Equation (53) can be written as

$$\begin{aligned} w(x) &= \mathbf{Q}^{\mathrm{T}}(x)[\mathbf{Q}_{\mathrm{B}}]^{-1}\{\mathbf{d}\} \\ &= \sum_{j=1}^{2N} \varphi_j(x) d_j, \end{aligned} \tag{61}$$

where φ are the nodal shape functions,

$$\begin{aligned}\varphi(x) &= \mathbf{Q}^{\mathrm{T}}(x)[\mathbf{Q}_{\mathrm{B}}]^{-1} \\ &= \Big[\psi_1^{(w)}(x) \;\; \psi_1^{(\theta)}(x) \;\; \psi_2^{(w)}(x) \;\; \psi_2^{(\theta)}(x) \;\; \ldots \\ &\qquad \psi_N^{(w)}(x) \;\; \psi_N^{(\theta)}(x)\Big].\end{aligned} \tag{62}$$

From Equation (62), the individual shape functions for deflection and slope, $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$, are

$$\begin{aligned}\psi_l^{(w)}(x) &= \sum_{k=1}^{N}\left(R_k(x)\cdot\eta_{(2k-1)(2l-1)} + S_k(x)\cdot\eta_{(2k)(2l-1)}\right) \\ \psi_l^{(\theta)}(x) &= \sum_{k=1}^{N}\left(R_k(x)\cdot\eta_{(2k-1)(2l)} + S_k(x)\cdot\eta_{(2k)(2l)}\right),\end{aligned} \tag{63}$$

where η_{kl} are the elements of the matrix $[\mathbf{Q}_{\mathrm{B}}]^{-1}$. The derivatives of these shape functions are easily evaluated as

$$\begin{aligned}\frac{d\psi_l^{(w)}(x)}{dx} &= \sum_{k=1}^{N}\left(\frac{dR_k(x)}{dx}\cdot\eta_{(2k-1)(2l-1)} + \frac{dS_k(x)}{dx}\cdot\eta_{(2k)(2l-1)}\right) \\ \frac{d\psi_l^{(\theta)}(x)}{dx} &= \sum_{k=1}^{N}\left(\frac{dR_k(x)}{dx}\cdot\eta_{(2k-1)(2l)} + \frac{dS_k(x)}{dx}\cdot\eta_{(2k)(2l)}\right),\end{aligned} \tag{64}$$

$$\begin{aligned}\frac{d^2\psi_l^{(w)}(x)}{dx^2} &= \sum_{k=1}^{N}\left(\frac{d^2R_k(x)}{dx^2}\cdot\eta_{(2k-1)(2l-1)} + \frac{d^2S_k(x)}{dx^2}\cdot\eta_{(2k)(2l-1)}\right) \\ \frac{d^2\psi_l^{(\theta)}(x)}{dx^2} &= \sum_{k=1}^{N}\left(\frac{d^2R_k(x)}{dx^2}\cdot\eta_{(2k-1)(2l)} + \frac{d^2S_k(x)}{dx^2}\cdot\eta_{(2k)(2l)}\right),\end{aligned} \tag{65}$$

$$\begin{aligned}\frac{d^3\psi_l^{(w)}(x)}{dx^3} &= \sum_{k=1}^{N}\left(\frac{d^3R_k(x)}{dx^3}\cdot\eta_{(2k-1)(2l-1)} + \frac{d^3S_k(x)}{dx^3}\cdot\eta_{(2k)(2l-1)}\right) \\ \frac{d^3\psi_l^{(\theta)}(x)}{dx^3} &= \sum_{k=1}^{N}\left(\frac{d^3R_k(x)}{dx^3}\cdot\eta_{(2k-1)(2l)} + \frac{d^3S_k(x)}{dx^3}\cdot\eta_{(2k)(2l)}\right).\end{aligned} \tag{66}$$

Hybrid Radial Basis Function

As discussed for C^0 problems, in order to improve the polynomial accuracy of the solutions, interpolations involving both radial basis functions and polynomial basis functions are considered as

$$\begin{aligned}w(\mathbf{x}) &= \mathbf{Q}^{\mathrm{T}}(\mathbf{x})\{\mathbf{c}\} + \mathbf{p}^{\mathrm{T}}(\mathbf{x})\{\mathbf{z}\} \\ &= \left[\mathbf{Q}^{\mathrm{T}}(\mathbf{x}) \;\; \mathbf{p}^{\mathrm{T}}(\mathbf{x})\right]\begin{Bmatrix}\mathbf{c}\\ \mathbf{z}\end{Bmatrix},\end{aligned} \tag{67}$$

where $\{\mathbf{c}\}$ and $\mathbf{Q}^{\mathrm{T}}(\mathbf{x})$ are as in Equation (54), and $\mathbf{p}^{\mathrm{T}}(x)$ are the polynomial basis functions,

$$\begin{aligned}\mathbf{p}^{\mathrm{T}}(x) &= \left[p_1(x),\;\; p_2(x),\;\; p_3(x),\;\; \ldots,\;\; p_m(x)\right] \\ &= \left[1 \;\; x \;\; x^2 \;\; \ldots \;\; x^{m-1}\right],\end{aligned} \tag{68}$$

and $\{\mathbf{z}\}$ are the unknown coefficients associated with $\mathbf{p}^{\mathrm{T}}(x)$,

$$\{\mathbf{z}\} = \{z_1,\;\; z_2,\;\; z_3,\;\; \ldots,\;\; z_m\}^{\mathrm{T}}. \tag{69}$$

The interpolation of Equation (67) is required to pass through the N points with constraints,

$$\begin{aligned}&\sum_{j=1}^{N} p_k(x_j)a_j = 0, \\ &\sum_{j=1}^{N} \frac{dp_k(x_j)}{dx}b_j = 0,\end{aligned} \tag{70}$$

where k = 1, 2, …, m, and a_j and b_j are the unknown coefficients in Equation (54). Equation (70) is imposed to guarantee a unique approximation. The set of equations to determine the coefficients $\{\mathbf{c}\}$ and $\{\mathbf{z}\}$ is thus written as

$$\begin{aligned}&\begin{bmatrix}\mathbf{Q}_{\mathrm{B}} & \mathbf{T}_{\mathrm{B}}\\ \mathbf{T}_{\mathrm{B}}^{\mathrm{T}} & \mathbf{0}\end{bmatrix}\begin{Bmatrix}\mathbf{c}\\ \mathbf{z}\end{Bmatrix} = \begin{Bmatrix}\mathbf{d}\\ \mathbf{0}\end{Bmatrix}, \text{ or} \\ &[G]\begin{Bmatrix}\mathbf{c}\\ \mathbf{z}\end{Bmatrix} = \begin{Bmatrix}\mathbf{d}\\ \mathbf{0}\end{Bmatrix},\end{aligned} \tag{71}$$

where $\{\mathbf{d}\}^{\mathrm{T}}$ and $[\mathbf{Q}_{\mathrm{B}}]$ are defined in Equations (58) and (59), and $[\mathbf{T}_{\mathrm{B}}]$ is a $(2N, m)$ matrix,

$$\mathbf{T}_{\mathrm{B}} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ \frac{dp_1(x_1)}{dx} & \frac{dp_2(x_1)}{dx} & \cdots & \frac{dp_m(x_1)}{dx} \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \frac{dp_1(x_2)}{dx} & \frac{dp_2(x_2)}{dx} & \cdots & \frac{dp_m(x_2)}{dx} \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_N) & p_2(x_N) & \cdots & p_m(x_N) \\ \frac{dp_1(x_N)}{dx} & \frac{dp_2(x_N)}{dx} & \cdots & \frac{dp_m(x_N)}{dx} \end{bmatrix}. \tag{72}$$

The unknown coefficients in Equation (71) are obtained as

$$\begin{Bmatrix}\mathbf{c}\\\mathbf{z}\end{Bmatrix}=[G]^{-1}\begin{Bmatrix}\mathbf{d}\\\mathbf{0}\end{Bmatrix}. \tag{73}$$

The interpolating function for w in Equation (67) is now written as

$$\begin{aligned}w(x)&=\left[\mathbf{Q}^{\mathrm{T}}(x)\quad \mathbf{p}^{\mathrm{T}}(x)\right][G]^{-1}\begin{Bmatrix}\mathbf{d}\\\mathbf{0}\end{Bmatrix}\\&=\sum_{j=1}^{2N}\varphi_j(x)d_j,\end{aligned}\tag{74}$$

where φ are the nodal shape functions

$$\begin{aligned}\varphi(x)&=\left[\mathbf{Q}^{\mathrm{T}}(x)\quad \mathbf{p}^{\mathrm{T}}(x)\right][G]^{-1}\\&=\left[\psi_1^{(w)}(x)\quad \psi_1^{(\theta)}(x)\quad \psi_2^{(w)}(x)\quad \psi_2^{(\theta)}(x)\quad \dots\right.\\&\qquad\left.\psi_N^{(w)}(x)\quad \psi_N^{(\theta)}(x)\right].\end{aligned}\tag{75}$$

From Equation (75), the shape functions for the deflection and slope, $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$, may be written as

$$\begin{aligned}\psi_l^{(w)}(x)&=\sum_{j=1}^{N}\left(R_j(x)\cdot\zeta_{(2j-1)(2l-1)}+S_j(x)\cdot\zeta_{(2j)(2l-1)}\right)\\&\quad+\sum_{k=1}^{m}p_k(x)\cdot\zeta_{(2N+k)(2l-1)}\\\psi_l^{(\theta)}(x)&=\sum_{j=1}^{N}\left(R_j(x)\cdot\zeta_{(2j-1)(2l)}+S_j(x)\cdot\zeta_{(2j)(2l)}\right)\\&\quad+\sum_{k=1}^{m}p_k(x)\cdot\zeta_{(2N+k)(2l)}\end{aligned}\tag{76}$$

where ζ_{kl} are the elements of the matrix $[G]^{-1}$. The derivatives of these shape functions are easy to evaluate as in Equations (64) – (66).

MLPG Equations for Beam Problems

In this paper, the radial basis function is used in the MLPG method for beam problems that are governed by the fourth-order equation

$$EI\frac{d^4w}{dx^4}=f \qquad \text{in}\quad 0\le x\le L \tag{77}$$

subjected to four boundary conditions, two at each end ($x = 0$ and $x = L$). The boundary conditions are on w, θ, V, and M, where

$$\theta=\frac{dw}{dx},\quad V=-EI\frac{d^3w}{dx^3},\quad \text{and}\quad M=EI\frac{d^2w}{dx^2} \tag{78}$$

are the slope, shear force, and moment, respectively. The essential boundary conditions are on w and θ, while the natural boundary conditions are on V and M. The boundary condition sets on w and V and θ and M are disjoint, i.e., if w is prescribed then V cannot be prescribed, and vice versa.

The MLPG equations are derived using a weighted residual weak form of the governing equation (Equation (77)). The MLPG equations are [4, 11, 13, 14]

$$\mathbf{K}^{(\mathrm{node})}\mathbf{d}+\mathbf{K}^{(\mathrm{bdry})}\mathbf{d}-\mathbf{f}^{(\mathrm{node})}-\mathbf{f}^{(\mathrm{bdry})}=\mathbf{0}, \tag{79}$$

where the superscript "bdry" denotes boundary,

$$\mathbf{d}^{\mathrm{T}}=\{w_1\quad \theta_1\quad w_2\quad \theta_2\quad \dots\quad w_N\quad \theta_N\} \tag{80a}$$

are the nodal values of deflections, w, and slopes, θ, at all the N nodes of the model used to analyze the problem (Equation (58)), and

$$\mathbf{K}^{(\mathrm{node})}=\left[\mathbf{k}_{ij}^{(\mathrm{node})}\right] \tag{80b}$$

$$\mathbf{K}^{(\mathrm{bdry})}=\left[\mathbf{k}_{ij}^{(\mathrm{bdry})}\right] \tag{80c}$$

with

$$\begin{aligned}\mathbf{k}_{ij}^{(\mathrm{node})}&=EI\begin{bmatrix}\displaystyle\int_{\Omega_s^{(i)}}\frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \displaystyle\int_{\Omega_s^{(i)}}\frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx\\ \displaystyle\int_{\Omega_s^{(i)}}\frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \displaystyle\int_{\Omega_s^{(i)}}\frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx\end{bmatrix}\\&\quad+n_xEI\begin{bmatrix}\chi_i^{(w)}\dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)}\dfrac{d^3\psi_j^{(\theta)}}{dx^3}\\ \chi_i^{(\theta)}\dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)}\dfrac{d^3\psi_j^{(\theta)}}{dx^3}\end{bmatrix}_{\Gamma_{sI}^{(i)}}\\&\quad-n_xEI\begin{bmatrix}\dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2}\\ \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2}\end{bmatrix}_{\Gamma_{sI}^{(i)}}\end{aligned}\tag{80d}$$

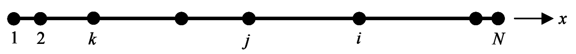
$$
\mathbf{k}_{ij}^{(\mathrm{bdry})} = \alpha_w \begin{bmatrix} \chi_i^{(w)}\psi_j^{(w)} & \chi_i^{(w)}\psi_j^{(\theta)} \\ \chi_i^{(\theta)}\psi_j^{(w)} & \chi_i^{(\theta)}\psi_j^{(\theta)} \end{bmatrix}_{\Gamma_{sw}^{(i)}}
+ n_x EI \begin{bmatrix} \chi_i^{(w)}\dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)}\dfrac{d^3\psi_j^{(\theta)}}{dx^3} \\ \chi_i^{(\theta)}\dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)}\dfrac{d^3\psi_j^{(\theta)}}{dx^3} \end{bmatrix}_{\Gamma_{sw}^{(i)}}
$$

$$
+ \alpha_\theta \begin{bmatrix} \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d\psi_j^{(w)}}{dx} & \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d\psi_j^{(\theta)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d\psi_j^{(w)}}{dx} & \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d\psi_j^{(\theta)}}{dx} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}}
- n_x EI \begin{bmatrix} \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2} \\ \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}} \tag{80e}
$$

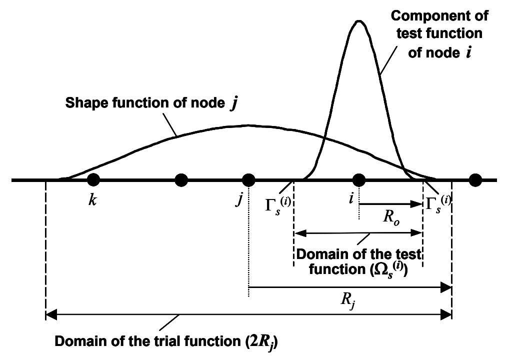
$$
\mathbf{f}^{(\mathrm{node})} = \begin{Bmatrix} \displaystyle\int_{\Omega_s^{(i)}} \chi_i^{(w)} f\, dx \\ \displaystyle\int_{\Omega_s^{(i)}} \chi_i^{(\theta)} f\, dx \end{Bmatrix} \quad \text{and} \tag{80f}
$$

$$
\mathbf{f}^{(\mathrm{bdry})} = n_x\widetilde{M} \begin{Bmatrix} \dfrac{d\chi_i^{(w)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{sM}^{(i)}} + n_x\widetilde{V} \begin{Bmatrix} \chi_i^{(w)} \\ \chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sV}^{(i)}}
+ \alpha_w\widetilde{w} \begin{Bmatrix} \chi_i^{(w)} \\ \chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sw}^{(i)}} + \alpha_\theta\widetilde{\theta} \begin{Bmatrix} \dfrac{d\chi_i^{(w)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{s\theta}^{(i)}} , \tag{80g}
$$

where i = 1, 2, …, N and j = 1, 2, …, n, and n is the number of nodes in the domain of definition of the trial function. In these equations, $\chi_i^{(w)}$ and $\chi_i^{(\theta)}$ are components of the test functions, $\psi_j^{(w)}$ and $\psi_j^{(\theta)}$ are the shape functions, $\Omega_s^{(i)}$ (see Figure 4b) is the local sub-domain of the test function at node i, n_x is the unit outward normal to $\Omega_s^{(i)}$, and $\Gamma_s^{(i)}$ are the boundary points of $\Omega_s^{(i)}$ (see Figure 4b). When $\Gamma_s^{(i)}$ coincides with an interior point, that point is denoted $\Gamma_{sI}^{(i)}$, and $\Gamma_{sw}^{(i)}$, $\Gamma_{s\theta}^{(i)}$, $\Gamma_{sM}^{(i)}$, and $\Gamma_{sV}^{(i)}$ denote the boundary points where $\Gamma_s^{(i)}$ intersects the boundary when w, θ, M, and V are prescribed, respectively. Also in these equations, α_w and α_θ are penalty parameters to enforce the essential boundary conditions, and $\widetilde{w}$, $\widetilde{\theta}$, $\widetilde{M}$, and $\widetilde{V}$ are prescribed values of the deflection, slope, moment, and shear, respectively. See reference 14 for a more detailed explanation of these terms.



(a) An N-node model of a beam


Component of test function of node i
Shape function of node j
k
j
i
$\Gamma_s^{(i)}$
$\Gamma_s^{(i)}$
R_o
Domain of the test function ($\Omega_s^{(i)}$)
R_j
Domain of the trial function ($2R_j$)

(b) Components of the trial and test functions

Figure 4: Comparison of the domains of the trial and test functions

The system of equations presented in Equations (79) – (80g) is the general set of 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 now briefly discussed.

<ins>Trial Functions</ins>

The trial functions are chosen as

$$
w(x) = \sum_{j=1}^{N} \left(\psi_j^{(w)}(x) w_j + \psi_j^{(\theta)}(x) \theta_j \right), \tag{81}
$$

where $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$ are the radial basis shape functions of Equation (63), and w_j and θ_j are the nodal values of w and θ at the N nodes (Equation (58)). Note that in MLPG algorithms employing the moving least squares interpolation scheme for the trial functions, the values $\mathbf{d}$ (Equation (80a)) are fictitious nodal values, $\hat{\mathbf{d}}$. In this paper, radial basis functions are fit to the actual nodal values, $\mathbf{d}$.

<ins>Test Functions</ins>

The test function, v, is assumed as in reference 14 as

$$
v(x) = \mu_i^{(w)} \chi_i^{(w)}(x) + \mu_i^{(\theta)} \chi_i^{(\theta)}(x) . \tag{82}
$$

In this paper, the test function components, χ_i, are chosen as in the conventional MLPG method. The $\chi_i^{(w)}(x)$ components of the test functions are chosen as power weight functions [14],

$$\chi_i^{(w)}(x) = \begin{cases} \left[1 - \left(\dfrac{d_i}{s_o} \right)^2 \right]^4 & \text{if} \quad 0 \le d_i \le s_o \\ 0 & \text{if} \quad d_i > s_o , \end{cases} \tag{83}$$

with $d_i = \|x - x_i\|$. In Equation (83), s_o is a user-defined parameter that determines the extent of the test functions (and hence Ω_s – see Figure 4). 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.,

$$\chi_i^{(\theta)} = \frac{d\chi_i^{(w)}}{dx} , \tag{84}$$

as $\theta = (dw/dx)$ is also a primary variable.

For this power function, the values of $\chi_i^{(w)}$, $\chi_i^{(\theta)}$, $(d\chi_i^{(w)}/dx)$, and $(d\chi_i^{(\theta)}/dx)$ are zero when $d_i = s_o$. As discussed in reference 14, when this test function is used, the $\mathbf{k}^{(\text{node})}$ in Equation (80d) reduces to

$$\mathbf{k}_{ij}^{(\text{node})} = EI \begin{bmatrix} \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2} dx & \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2} dx \\ \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2} dx & \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2} dx \end{bmatrix} . \tag{85}$$

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. Five models with 5, 9, 17, 33, and 65 nodes uniformly distributed along the length of the beam are considered. Figure 5 shows a typical 17-node model. The distances between the nodes (Δx / l) in these models are 1, 0.5, 0.25, 0.125, and 0.0625 for the 5-, 9-, 17-, 33-, and 65-node models, respectively. Numerical integration is used to integrate the system of equations as closed-form integration of the terms in Equations (80d) and (80f) is extremely complicated.

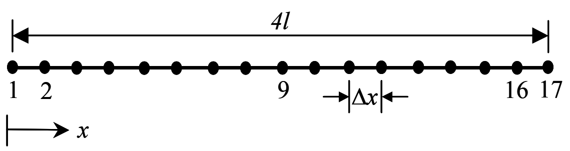


Figure 5: A 17-node model of the beam

Numerical Evaluations

The radial basis MLPG (RPG) method was evaluated by applying the method to simple patch-test problems. The problems considered were (a) rigid body translation:

$$w(x) = \beta_0, \qquad \theta = \frac{dw}{dx} = 0 , \tag{86a}$$

(b) rigid body rotation:

$$w(x) = \beta_1 x, \qquad \theta = \beta_1 , \tag{86b}$$

and (c) constant-curvature condition:

$$w(x) = \beta_2 x^2/2, \;\; \theta = \beta_2 x , \tag{86c}$$

where β_0, β_1, and β_2 are arbitrary constants. The third patch test is equivalent to the problem of a cantilever beam with a moment, $M=EI(d^2w/dx^2)= EI\beta_2$, 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 RPG method with no polynomial basis. If the RPG method recovers the exact solution at all the interior nodes and at every arbitrary point of the beam, then the method passes the patch test. Note that in this work, near recovery of the exact solution is sufficient to pass the patch tests as the radial basis functions alone cannot represent polynomial solutions exactly.

Compact RBF

The compact radial functions described by Equations (36) and (37) were considered first. When using the compactly supported functions, the Equations (59) – (66) are evaluated with $N = n$, the number of nodes in the influence domain of the point x under consideration.[18, 19] This use of the compact functions forces the [$\mathbf{Q}_B$] of Equation (59) to become a ($2n$, $2n$) matrix that must be evaluated once for every node in the model.

The RPG method with no polynomial terms was *unable* to reproduce the exact solutions in Equation (86), and thus *failed the patch tests*. A quadratic polynomial basis ($m = 3$; p_w: $(1, x, x^2)$, p_θ: $(0, 1, 2x)$) was used in Equation (67), increasing the size of the [$\mathbf{Q}_B$] matrix to ($2n+m$, $2n+m$). The RPG method with polynomial basis (the hybrid RPG method) reproduced the solutions in Equation (86) to machine accuracy, thus passing the patch tests.

Next, mixed boundary value problems were considered. The first problem considered was a cantilever beam with a tip load (Figure 6). Because the exact solution for this problem is cubic in x, the hybrid RPG method with a cubic polynomial basis function reproduced the exact solution. A simply supported beam subjected to a uniformly distributed load (Figure 7) was considered next. Because the exact solution for this problem is quartic in x, the hybrid RPG method with quartic basis yielded the solution exactly.

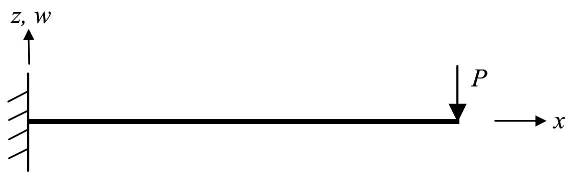


Figure 6: Cantilever beam with a tip load

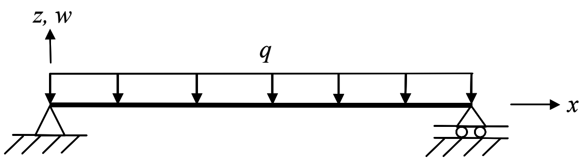


Figure 7: Simply supported beam subjected to a uniformly distributed load

As with the finite element method and the conventional MLPG method [13, 14], the hybrid RPG algorithm should be robust enough to yield good solutions when a low order polynomial basis function is used. A convergence test was conducted to study the performance of the hybrid method in solving the problems in Figures 6 and 7. A quadratic polynomial basis function was used. For all models (5, 9, 17, 33, and 65 nodes), the method did not yield meaningful results. Thus, it was concluded that as long as the order of the polynomial basis was sufficient to reproduce the solution exactly, the polynomial terms overpowered the radial basis functions. This condition is too restrictive, and hence compact radial functions are dropped from further consideration.

Non-compactly Supported RBF

Because the compactly supported radial basis functions are incapable of producing meaningful results for beam problems, the non-compact functions of Table 1 are considered. In these functions,

$$r = \frac{d_j}{s_j}, \tag{87}$$

where $d_j = \|x - x_j\|$, and s_j is some normalizing distance, usually chosen to be the entire problem domain, Ω (in this work, $0 \le x \le L$). As s_j covers the entire problem domain, [$\mathbf{Q}_B$] is an (N, N) matrix that is evaluated and inverted once.

Upon implementation of the functions in Table 1, the cubic function,

$$R(r) = r^3, \tag{88}$$

worked very well for the current C^1 problems. The RPG method with no polynomial basis and using Equation (88) was applied to the patch tests represented by Equations (86). The method successfully reproduced the exact solutions to machine accuracy, thus passing all the patch tests. Additionally, all functions of the form

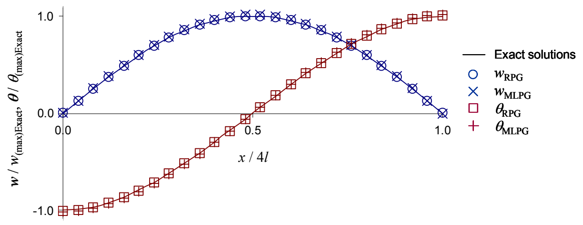
$$R(r) = r^{(2z-1)}, \tag{89}$$

where $z > 1$, performed successfully, though r^3 gave the best results.

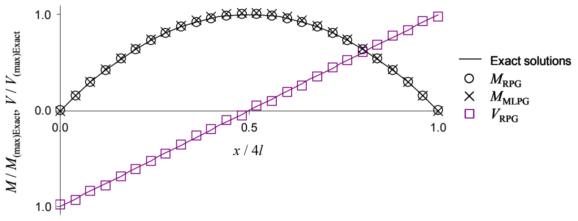
Next, the RPG method with the RBF in Equation (88) was used to solve mixed boundary value problems. In the method, a 12-point Gaussian integration was used, the value of (s_o / l), which defines the extent of the test functions (see Equation (83)), was set as ($s_o / l = 2\Delta x$), and the value of (s_j / l), which defines the extent of the trial functions (Equation (87)), was set as ($s_j / l = L$). For the cantilever beam with a tip load in Figure 6, the RPG method yielded excellent results. The simply supported beam problem with a uniformly distributed load (Figure 7) was analyzed using 17-, 33-, and 65-node models. The maximum deflection values, i.e., the deflection at ($x = L / 2$), for these three models obtained using the RPG method and using the conventional MLPG method with a quadratic polynomial basis function are compared in Table 2. In the MLPG method, a 20-point Gaussian integration was used, the value of (s_o / l) was set as ($s_o / l = 2\Delta x$), and the value of (s_j / l) was set as ($s_j / l = 8\Delta x$). From this table, it is seen that the RPG method performs as accurately as the conventional MLPG method.

Table 2: Maximum deflection values for three nodal models obtained using the RPG and MLPG methods compared to the exact solution.

Model	Maximum deflection (at $x = L / 2$)		
	Exact	RPG	MLPG
17-node	-3.3333e-7	-3.2739e-7	-3.3106e-7
33-node	-3.3333e-7	-3.3407e-7	-3.3735e-7
65-node	-3.3333e-7	-3.3420e-7	-3.3848e-7



(a) Primary variables



(b) Secondary variables (NOTE: V_{MLPG} not shown)

Figure 8: RPG, MLPG, and Exact solutions obtained using the 65-node model for the simply supported beam subjected to a uniformly distributed load

The results obtained for deflection, slope, moment, and shear using the 65-node model are presented in Figure 8. In this figure, the RPG results are compared to the exact solution and to the solution obtained using the conventional MLPG method with a quadratic polynomial basis function. For each of the nodal models (17, 33, and 65 nodes), the RPG values for deflection, slope, and moment were as accurate as the MLPG values and were in excellent agreement with the exact values. In addition, the RPG values for shear converged with model refinement. The MLPG solution for the shear was erratic, and is not shown in Figure 8. The quadratic basis function is insufficient to accurately calculate the third derivatives for this problem, and the method could not recover the values with model refinement; the solution for the shear converged only as the order of the basis function was increased to quartic.[13] The results discussed for this problem verify the perceived advantages of the RPG method over the MLPG method.

The RPG method with the RBF in Equation (88) was then applied to a problem with load discontinuity. The problem considered was the cantilever beam with a uniformly distributed load on a portion of the beam shown in Figure 9. The RPG solution (with $(s_o / l = 4\Delta x)$) for the cantilever beam problem exhibited convergence with model refinement. These results are consistent with those reported in reference 14, where this problem was studied using the conventional MLPG method. The exact, MLPG, and RPG values for deflection and moment for this problem obtained using a 65-node model are compared in Figure 10. The parameters used for the MLPG method are the same as those reported above for the simply supported beam problem. The RPG method handled the load discontinuity well and yielded results in overall agreement with the exact solutions.

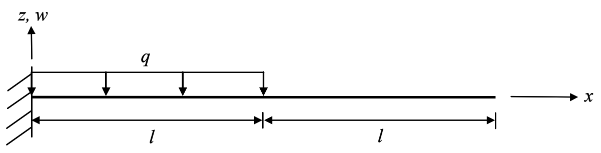


Figure 9: Cantilever beam with a uniformly distributed load on a portion of the beam

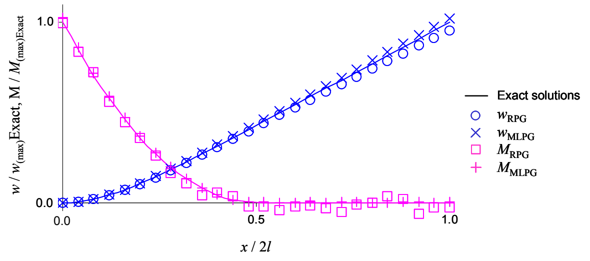


Figure 10: RPG, MLPG, and Exact solutions obtained using the 65-node model for the cantilever beam with a uniformly distributed load on a portion of the beam

Concluding Remarks

A radial basis function implementation of the MLPG method was presented to study Euler-Bernoulli beam problems. Like the conventional MLPG method, this radial basis variation (RPG) is based on the local weak form developed from the classical weighted residual form of the fourth-order governing differential equation. In this method, radial basis functions, rather than generalized moving least squares (GMLS) interpolations, were used to develop the trial functions, and test functions were chosen as simple weight functions as in the conventional MLPG method. RPG equations were developed with and without including polynomial basis function terms.

The compactly supported radial basis functions did not perform well without polynomial terms in the computations. When polynomial terms were included, the compactly supported RPG method passed the patch tests. However, the method did not yield meaningful results for mixed boundary value problems unless the order of the polynomial basis function was of the same order as the exact solution of the problem. This result restricts the use of the method. The use of compactly supported radial basis functions is not recommended.

The non-compactly supported cubic radial basis function performed very well when no polynomial terms were included in the computations. The RPG method with the cubic radial basis function passed all the patch tests and yielded results for mixed boundary value problems that are comparable to those obtained using the conventional MLPG method. The RPG method with a cubic radial basis also yielded very good results for a problem with discontinuous loading conditions. The accuracy of solutions obtained by the RPG method, combined with the computational efficiency of using the radial basis functions rather than the GMLS interpolations to approximate the trial functions, makes the RPG method a very attractive variation of the MLPG method.

References

[1]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.

[2]Belytschko, T., Lu, Y. Y., and Gu, L. (1994): "Element-free Galerkin methods," *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 229-256.

[3]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.

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

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

[6]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," *Computer Modeling in Engineering & Sciences*, Vol. 3, No. 1, pp. 11-51.

[7]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.

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

[9]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.

[10]Long, S. and Atluri, S. N. (2002): "A meshless local Petrov-Galerkin method for solving the bending problem of a thin plate," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 3, No. 1, pp. 53-63.

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

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

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

[14]Raju, I. S. and Phillips, D. R. (2003): "Further developments in the MLPG method for beam problems," *CMES: Computer Modeling in Engineering & Sciences*, in press.

[15]Powell, M.J.D., "The theory of radial basis function approximation in 1990," in *Advances in Numerical Analysis*, Vol. 2, Clarendon Press, Oxford 1992, pp. 105-210.

[16]Wendland, H. (1999): "On the smoothness of positive definite and radial functions," *Journal of Computational and Applied Mathematics*, pp. 177-188.

[17]Wu, Z. (1995): "Compactly supported positive definite radial function," *Advances in Computational Mathematics*, pp. 283-292.

[18]Wang, J. G. and Liu, G. R. (2002): "A point interpolation meshless method based on radial basis functions," *International Journal for Numerical Methods in Engineering*, Vol. 54, pp. 1623-1648.

[19]Wang, J. G. and Liu, G. R. (2002): "On the optimal shape parameters of radial basis functions used for 2-D meshless methods," *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, pp. 2611-2630.