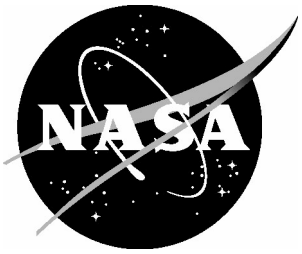
NASA/TP-2004-212996



A Meshless Method Using Radial Basis Functions for Beam Bending Problems

I. S. Raju
Langley Research Center, Hampton, Virginia

D. R. Phillips
Lockheed Martin Space Operations
Langley Research Center, Hampton, Virginia

T. Krishnamurthy
Langley Research Center, Hampton, Virginia

November 2004

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

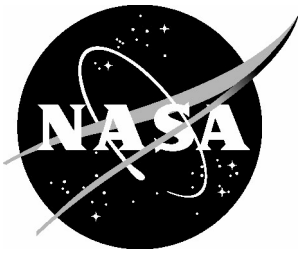
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TP-2004-212996



A Meshless Method Using Radial Basis Functions for Beam Bending Problems

I. S. Raju
Langley Research Center, Hampton, Virginia

D. R. Phillips
Lockheed Martin Space Operations
Langley Research Center, Hampton, Virginia

T. Krishnamurthy
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

November 2004

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Abstract

A meshless local Petrov-Galerkin (MLPG) method that uses radial basis functions (RBFs) as trial functions in the study of Euler-Bernoulli beam problems is presented. RBFs, 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. Test functions are chosen as simple weight functions as they are in the conventional MLPG method. Compactly and noncompactly supported RBFs are considered. Noncompactly supported cubic RBFs are found to be preferable. Patch tests, mixed boundary value problems, and problems with complex loading conditions are considered. Results obtained from the radial basis MLPG method are either of comparable or better accuracy than those obtained when using the conventional MLPG method.

Introduction

Meshless methods are developed to overcome some of the disadvantages of the finite element method (FEM), such as discontinuous secondary variables across interelement boundaries and the need for remeshing in large deformation problems (Atluri, Cho, and Kim, 1999; Atluri and Shen, 2002b; Atluri and Zhu, 1998; Belytschko, Liu, and Gu, 1994; and Nayroles, Touzot, and Villon, 1992). Recent literature shows extensive research on meshless methods and, in particular, the meshless local Petrov-Galerkin (MLPG) method. Two recent monographs (Atluri and Shen, 2002a; and Liu, 2002) summarize the status to date. Most literature published to date on the MLPG method presents variations of the method for C^0 problems. A comparatively limited amount of work (Atluri, Cho, and Kim, 1999; Donning and Liu, 1998; Gu and Liu, 2001a and 2001b; and Krysl and Belytschko, 1995) is reported on C^1 problems. The additional degrees of freedom involved in C^1 problems introduce levels of complexity that are difficult to implement in meshless methods. Gu and Liu present a local point interpolation method (LPIM) for static and dynamic analysis of thin beams (Gu and Liu, 2001a). Atluri, Cho, and Kim (1999) present an analysis of thin-beam problems by using a Galerkin implementation of the MLPG method, in which 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 Phillips and Raju (2002) and Raju and Phillips (2002, 2003a, and 2003c), an MLPG method is presented in which a GMLS approximation is used to construct the trial functions; 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 simpler alternative to the MLS trial functions that provides the same accuracy as the MLS functions is preferred.

In this paper, the use of radial basis interpolation functions (see Powell, 1992; Wendland, 1999; and Wu, 1995) as trial functions is explored in the MLPG formulation for beam problems. The radial basis functions (RBFs) and their implementation are simple, and the evaluation of the derivatives is much simpler than for the traditional MLS approximations (Raju, Phillips, and Krishnamurthy, 2003b). In the present radial basis MLPG formulation (abbreviated hereafter as RPG), 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 intent of this paper is to present an alternative to the MLS interpolation that may be more computationally efficient in the MLPG method for beam problems. Currently, the MLPG method is more computationally expensive compared to the FEM. The objective of this work is not to improve the computational efficiency of the MLPG method but rather to explore the feasibility of using RBFs as interpolations in the method.

The outline of the paper is as follows: First, the MLS interpolation used in the conventional MLPG method is discussed as motivation for finding a simpler alternative. Next, an overview of RBFs for C^0 problems is presented, the shape functions obtained from radial basis interpolation are derived, and the shape functions that are obtained when polynomial basis functions are included in the interpolation are derived. The MLS and radial basis shape function discussions are then expanded and repeated for beam (C^1) 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 RBF, and the RPG method is applied to mixed boundary value problems. Finally, the method is applied to problems with complex loading and boundary conditions.

C^0 Interpolation Schemes

In this section, the MLS interpolation scheme used in the conventional MLPG method is discussed first; then, two interpolation schemes involving RBFs are presented. In the first scheme, RBFs alone are used to construct the shape functions. The second scheme is a hybrid that uses both RBFs and polynomial basis functions to construct the shape functions.

The MLS Interpolation

An MLS interpolation is a scheme that fits a smooth function through an assumed set of fictitious nodal values. The interpolation is performed such that the sum of the least squares error between the interpolated function and the fictitious nodal values is a minimum (Belytschko, Lu, and Gu, 1994; and Nayroles, Touzot, and Villon, 1992). A schematic of the MLS interpolation is presented in figure 1.

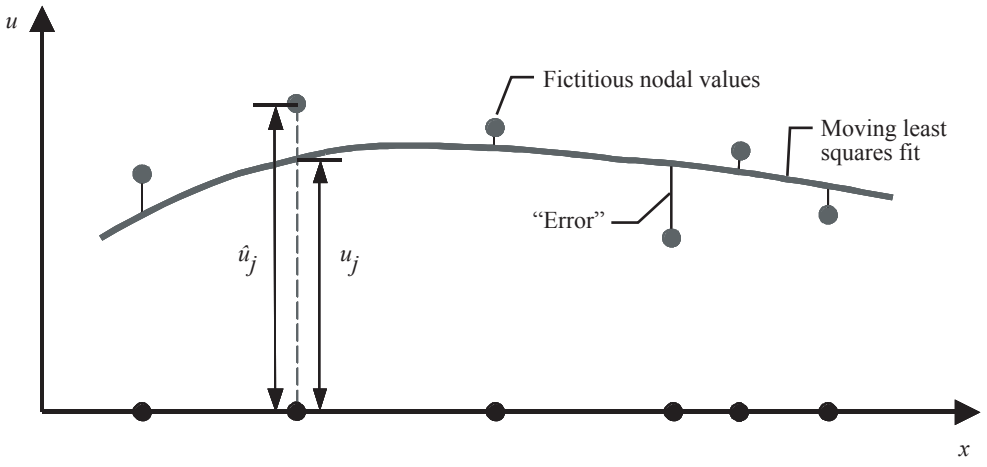


Figure 1. Moving least squares (MLS) interpolation.

In one-dimensional problems, the C^0 MLS shape function for node j can be constructed as follows. In the equations that follow, n is the number of nodes in the domain of definition of node j, and m is the order of the polynomial basis function (Nayroles, Touzot, and Villon, 1992). The shape function ϕ_j at node j can be written as

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

where $\mathbf{p}^{\mathrm{T}}(x)$ is a polynomial basis function

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

and

$$\begin{aligned}[\mathbf{A}] &= [\mathbf{P}]^{\mathrm{T}}[\lambda][\mathbf{P}] \\ [\mathbf{B}] &= [\mathbf{P}]^{\mathrm{T}}[\lambda]\end{aligned} \tag{3}$$

In equations (3), [**P**] is an (n, m) matrix, and [$\boldsymbol{\lambda}$] is a diagonal (n, n) matrix defined as

$$[\mathbf{P}] = \left[\mathbf{p}^{\mathrm{T}}(x_1) \quad \mathbf{p}^{\mathrm{T}}(x_2) \quad \ldots \quad \mathbf{p}^{\mathrm{T}}(x_n)\right]^{\mathrm{T}} \tag{4}$$

$$\lambda = \begin{bmatrix} \lambda_1(x) & & & \\ & \lambda_2(x) & & \\ & & \ddots & \\ & & & \lambda_n(x) \end{bmatrix} \tag{5}$$

where $\lambda_j(x),\ j = 1, \ldots, n,$ is a weight function. The first derivative of these shape functions is required by the MLPG method and is given by (Atluri and Zhu, 1998; and Belytschko, Liu, and Gu, 1994)

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

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

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

RBF Interpolation

The radial basis formulation for C^0 problems is presented first. The RBF provides a continuous interpolating function for $u(\mathbf{x})$ as a linear combination of radial functions (Powell, 1992). The interpolating

function is

$$u(\mathbf{x}) = \sum_{j=1}^{N} R_j(\mathbf{x})\, a_j \tag{8}$$

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

$$R_j(\mathbf{x}) = R_j(r_j) \tag{9}$$

In two dimensions, the radial distance r_j can be expressed in terms of the Cartesian coordinates as

$$r_j(\mathbf{x}) = \sqrt{\left(x - x_j\right)^2 + \left(y - y_j\right)^2} \tag{10}$$

where x_j and y_j are the coordinates of node j. Equation (8) can be written in matrix form for C^0 problems as

$$u(\mathbf{x}) = \mathbf{R}^{\mathrm{T}}(\mathbf{x})\, \mathbf{a} \tag{11}$$

where

$$\begin{aligned} \mathbf{R}^{\mathrm{T}}(\mathbf{x}) &= \left[R_1(\mathbf{x}), \quad R_2(\mathbf{x}), \quad R_3(\mathbf{x}), \quad \ldots, \quad R_N(\mathbf{x})\right] \\ \mathbf{a} &= \left\{a_1, \quad a_2, \quad a_3, \quad \ldots, \quad a_N\right\}^{\mathrm{T}} \end{aligned} \tag{12}$$

Forcing the interpolation of equation (8) to pass through the N scattered points, a set of equations to determine the coefficients a_j in terms of nodal values u_j can be written as

$$\mathbf{R}_{\mathrm{B}}\, \mathbf{a} = \mathbf{u} \tag{13}$$

where

$$\mathbf{u}^{\mathrm{T}} = \left(u_1, \quad u_2, \quad u_3, \quad \cdots, \quad u_N\right) \tag{14}$$

$$\mathbf{R}_{\mathrm{B}} = \begin{bmatrix} R_1(\mathbf{x}_1) & R_2(\mathbf{x}_1) & \cdots & R_N(\mathbf{x}_1) \\ R_1(\mathbf{x}_2) & R_2(\mathbf{x}_2) & \cdots & R_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(\mathbf{x}_N) & R_2(\mathbf{x}_N) & \cdots & R_N(\mathbf{x}_N) \end{bmatrix} \tag{15}$$

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 (13) can be obtained as

$$\mathbf{a} = \mathbf{R}_{\mathrm{B}}^{-1}\, \mathbf{u} \tag{16}$$

The interpolating function for $u(\mathbf{x})$ in equation (11) can now be rewritten as

$$u(\mathbf{x}) = \mathbf{R}^{\mathrm{T}}(\mathbf{x})\mathbf{R}_{\mathrm{B}}^{-1}\mathbf{u} = \sum_{j=1}^{N} \varphi_j(\mathbf{x}) u_j \tag{17}$$

The nodal shape functions are then

$$\begin{aligned} \varphi(\mathbf{x}) &= \mathbf{R}^{\mathrm{T}}(\mathbf{x})\mathbf{R}_{\mathrm{B}}^{-1} \\ &= [\varphi_1(\mathbf{x}), \quad \varphi_2(\mathbf{x}), \quad \varphi_3(\mathbf{x}), \quad \ldots, \quad \varphi_N(\mathbf{x})] \end{aligned} \tag{18}$$

or

$$\varphi_l(\mathbf{x}) = \sum_{k=1}^{N} R_k(\mathbf{x}) \xi_{kl} \tag{19}$$

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 at the nodes (Atluri and Shen, 2002a), that is,

$$\varphi_j(\mathbf{x}_k) = \delta_{jk} \tag{20}$$

Note that the shape functions in equation (19) also satisfy the property

$$\sum_{j=1}^{N} \varphi_j(\mathbf{x}) = 1 \tag{21}$$

only at the nodes. As the number of nodes is increased, however, the shape functions in equation (19) satisfy (Raju, Phillips, and Krishnamurthy, 2003b)

$$\lim_{N \to \infty} \sum_{j=1}^{N} \varphi_j(\mathbf{x}) = 1 \tag{22}$$

everywhere in the domain.

In contrast to the MLS method, the derivatives of the shape functions are easy to evaluate by using equation (19) as

$$\begin{aligned} \frac{\partial \varphi_l(\mathbf{x})}{\partial x} &= \sum_{k=1}^{N} \frac{\partial R_k}{\partial x} \cdot \xi_{kl} \\ \frac{\partial \varphi_l(\mathbf{x})}{\partial y} &= \sum_{k=1}^{N} \frac{\partial R_k}{\partial y} \cdot \xi_{kl} \end{aligned} \tag{23}$$

where

$$\frac{\partial R_k}{\partial x} = \frac{\partial R_k}{\partial r_k}\frac{\partial r_k}{\partial x} \quad ; \quad \frac{\partial R_k}{\partial y} = \frac{\partial R_k}{\partial r_k}\frac{\partial r_k}{\partial y} \tag{24}$$

with

$$\frac{\partial r_k}{\partial x} = \frac{x - x_k}{r_k} \quad ; \quad \frac{\partial r_k}{\partial y} = \frac{y - y_k}{r_k} \tag{25}$$

Some of the well-known classical radial functions are presented in table 1, in which ($t_j = r_j/s_j$) and s_j is some normalizing distance. Note that the shape parameter c in the RBFs (table 1) is user defined. The classical radial functions have two limitations: (1) the matrix $\mathbf{R}_B$ in equation (15) may not be positive definite, and (2) the functions do not possess local support, that is, the changes in a single location (x_j, y_j) affect the entire interpolation. To overcome these limitations, compactly supported positive definite radial functions were proposed (Wu, 1995). These functions were derived by using a constraint to guarantee positive definiteness of the interpolation matrix $\mathbf{R}_B$.

Table 1. Classical Radial Basis Functions (RBFs)

Classical RBF	Equation
Linear	$R_j(\mathbf{x}) = ct_j$
Cubic	$R_j(\mathbf{x}) = (t_j + c)^3$
Thin plate spline	$R_j(\mathbf{x}) = t_j^2 \log(ct_j^2)$
Gaussian	$R_j(\mathbf{x}) = e^{-ct_j^2}$
Multiquadric	$R_j(\mathbf{x}) = \sqrt{t_j^2 + c^2}$

*See also tables 5.2 and 5.3 (Liu, 2002).

The compactly supported functions assign zero weight to all points not belonging to the compactly supported region, thus leading to a sparse interpolation matrix. Adapted compact RBFs are used here:

Compact-I:

$$R_j(\mathbf{x}) = \begin{cases} (1-t_j)^5(8 + 40t_j + 48t_j^2 + 25t_j^3 + 5t_j^4), & 0 \le t_j \le 1 \\ 0 & t_j > 1 \end{cases} \tag{26}$$

Compact-II:

$$R_j(\mathbf{x}) = \begin{cases} (1-t_j)^6(6 + 36t_j + 82t_j^2 + 72t_j^3 + 30t_j^4 + 5t_j^5), & 0 \le t_j \le 1 \\ 0 & t_j > 1 \end{cases} \tag{27}$$

where ($t_j = r_j/s_j$) and s_j, the normalizing distance, is the radius of the domain of compact support. The shape functions (eq. (19)) obtained from the Compact-I and Compact-II functions possess all the properties in equations (20)–(22). Several other forms of compact support functions can be found in Wendland (1999) and Wu (1995). The functions in equations (26) and (27) were specifically chosen because they were previously adapted and used as test functions in the MLPG method (Atluri and Shen, 2002a).

Hybrid RBF Interpolation

Most of the classical RBFs shown in table 1 and the compactly supported functions in equations (26) and (27) cannot represent simple polynomials exactly at every point in the domain of the interpolation (Wang and Liu, 2002a and 2002b); they can represent the polynomial values exactly only at the N scattered points where polynomial values are prescribed. Figures 2 and 3 show RBFs (the thin curves) obtained from the compact RBF in equation (26) with $s_j = 0.6$ using 5 nodes in the interval $-1 \le x \le 1$ for a constant polynomial of value unity and a linear polynomial, respectively. The function values are evaluated at the five nodes and are prescribed as the u_j values in equation (17). The values of u are evaluated by using equation (17) at 200 points in the interval $-1 \le x \le 1$ and are plotted in these figures (the thick black curves). The exact functions are also shown (thick gray lines). The compact RBF interpolation recovers the polynomial values exactly only at the five nodal points, and elsewhere the values of the polynomials deviate from the exact values somewhat significantly. The deviation from the exact values can, however, be reduced by increasing the number of nodes in the domain $-1 \le x \le 1$.

To improve the polynomial accuracy of the solutions, Powell (1992) suggested augmenting the radial basis interpolation by adding polynomial basis functions as

$$u(\mathbf{x}) = \sum_{j=1}^{N} R_j(\mathbf{x})\, a_j + \sum_{k=1}^{m} p_k(\mathbf{x})\, z_k \tag{28}$$

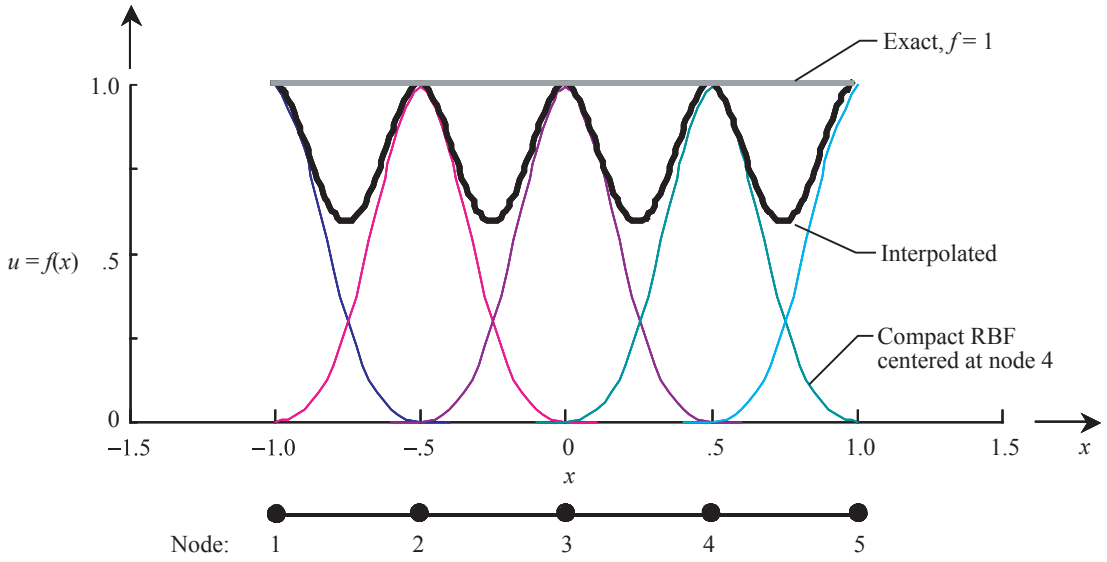


Figure 2. RBF values that correspond to interpolated and exact values of a constant polynomial.

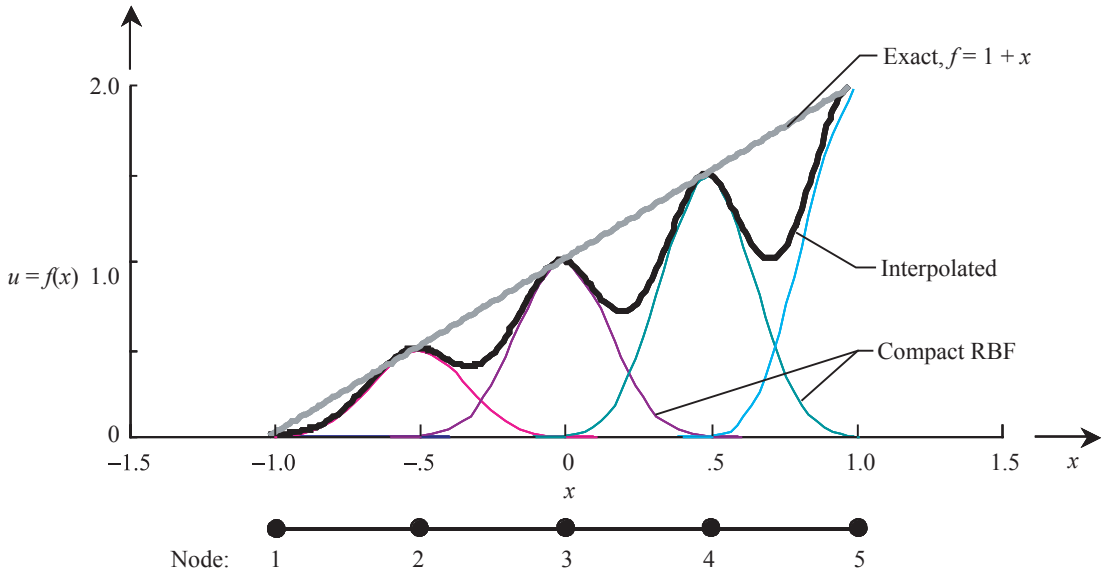


Figure 3. RBF values that correspond to interpolated and exact values of a linear polynomial.

where R_j, a_j, and N are as shown in equation (8), $p(\mathbf{x})$ are polynomial basis functions, z_k are the unkown coefficients associated with the kth polynomial term, and m is the order of the polynomial basis function. Equation (28) is written in matrix form as (Wang and Liu, 2002a)

$$\begin{aligned} u(\mathbf{x}) &= \mathbf{R}^{\mathrm{T}}(\mathbf{x})\ \mathbf{a} + \mathbf{p}^{\mathrm{T}}(\mathbf{x})\ \mathbf{z} \\ &= \begin{bmatrix} \mathbf{R}^{\mathrm{T}}(\mathbf{x}) & \mathbf{p}^{\mathrm{T}}(\mathbf{x}) \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix} \end{aligned} \tag{29}$$

where $\mathbf{a}$ and $\mathbf{R}^{\mathrm{T}}(\mathbf{x})$ are as in equation (12), and

$$\begin{aligned} \mathbf{z} &= \{ z_1, \quad z_2, \quad z_3, \quad \ldots, \quad z_m \}^{\mathrm{T}} \\ \mathbf{p}^{\mathrm{T}}(\mathbf{x}) &= \begin{bmatrix} p_1(\mathbf{x}), & p_2(\mathbf{x}), & p_3(\mathbf{x}), & \ldots, & p_m(\mathbf{x}) \end{bmatrix} \end{aligned} \tag{30}$$

In 1-D problems,

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \begin{bmatrix} 1, & x, & x^2, & \ldots, & x^{m-1} \end{bmatrix}$$

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

$$\sum_{j=1}^{N} p_k(\mathbf{x}_j) a_j = 0, \quad k = 1, 2, \ldots, m \tag{31}$$

imposed to guarantee unique approximation (Wang and Liu, 2002a). The set of equations to determine the coefficients $\{\mathbf{a}\}$ and $\{\mathbf{z}\}$ can then be written as

$$\begin{bmatrix} \mathbf{R}_\mathrm{B} & \mathbf{P}_\mathrm{B} \\ \mathbf{P}_\mathrm{B}^\mathrm{T} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix} = \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix}, \text{ or}$$
$$[\mathbf{G}] \begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix} = \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix} \tag{32}$$

where $\mathbf{u}^\mathrm{T}$ and $\mathbf{R}_\mathrm{B}$ are defined in equations (14) and (15), and

$$\mathbf{P}_\mathrm{B} = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_N) & p_2(\mathbf{x}_N) & \cdots & p_m(\mathbf{x}_N) \end{bmatrix} \tag{33}$$

The unknown coefficients in equation (32) are obtained as

$$\begin{Bmatrix} \mathbf{a} \\ \mathbf{z} \end{Bmatrix} = [\mathbf{G}]^{-1} \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix} \tag{34}$$

The interpolating functions $u(\mathbf{x})$ in equation (29) can now be rewritten as

$$\begin{aligned} u(\mathbf{x}) &= \begin{bmatrix} \mathbf{R}^\mathrm{T}(\mathbf{x}) & \mathbf{p}^\mathrm{T}(\mathbf{x}) \end{bmatrix} [\mathbf{G}]^{-1} \begin{Bmatrix} \mathbf{u} \\ \mathbf{0} \end{Bmatrix} \\ &= \sum_{j=1}^{N} \varphi_j(\mathbf{x}) u_j \end{aligned} \tag{35}$$

The nodal shape functions are then

$$\varphi_I(\mathbf{x}) = \sum_{j=1}^{N} R_j(\mathbf{x}) \gamma_{jI} + \sum_{k=1}^{m} p_k(\mathbf{x}) \gamma_{(N+k)(I)} \tag{36}$$

where γ_{pq} are the elements of the matrix $[\mathbf{G}]^{-1}$. The shape functions in equation (36) possess the properties of equations (20) and (21). The derivatives of the shape functions are easy to evaluate using equation (36) as

$$\begin{aligned} \frac{\partial \varphi_I(\mathbf{x})}{\partial x} &= \sum_{j=1}^{N} \frac{\partial R_j(\mathbf{x})}{\partial x} \gamma_{jI} + \sum_{k=1}^{m} \frac{\partial p_k(\mathbf{x})}{\partial x} \gamma_{(N+k)(I)} \\ \frac{\partial \varphi_I(\mathbf{x})}{\partial y} &= \sum_{j=1}^{N} \frac{\partial R_j(\mathbf{x})}{\partial y} \gamma_{jI} + \sum_{k=1}^{m} \frac{\partial p_k(\mathbf{x})}{\partial y} \gamma_{(N+k)(I)} \end{aligned} \tag{37}$$

where $(\partial R_j / \partial x)$ and $(\partial R_j / \partial y)$ are as defined in equations (24) and (25).

Consider again the constant and linear polynomials depicted in figures 2 and 3. When a constant polynomial basis ($\mathbf{p}^T(x) = [1]$) is included in the interpolation, as described by equations (28)–(37), the constant polynomial of value unity (see fig. 2) is recovered exactly at all locations in the domain (the solution coincides with the exact solution in the figure). Similarly, when a linear polynomial basis ($\mathbf{p}^T(x) = [1, x]$) is included in the interpolation, the values of the linear polynomial (fig. 3) are recovered exactly at all locations in the domain.

Interpolation Schemes—Summary

In the previous section, two interpolation schemes were presented for C^0 problems: the MLS interpolation and the RBF interpolation. In this section, the FEM, MLS, and RBF interpolations are compared in terms of the continuity of the secondary variables. For C^0 problems, the secondary variable is, for example, ($q = du/dx$). Examples of trial functions formed for each of the interpolations are presented schematically in figure 4.

In the FEM, u is chosen as a piecewise linear function, as is shown in figure 4(a). (For simplicity, the nodal shape functions for element e are shown, and the shape functions for the remaining elements are omitted from the figure.) The slopes at node j, evaluated from elements e and $e + 1$, are obviously unequal. In general, all secondary variables in the FEM are discontinuous across element boundaries because of the piecewise nature of the approximation for the trial function. Postprocessing techniques are required to achieve smooth distributions for the secondary variables.

As discussed previously, an MLS interpolation is a scheme that fits a smooth function through an assumed set of fictitious nodal values. Because the trial functions formed by the MLS interpolation are smooth (see fig. 4(b), in which shape functions are omitted entirely from the figure), the secondary variables are continuous at every point in the domain of the trial functions.

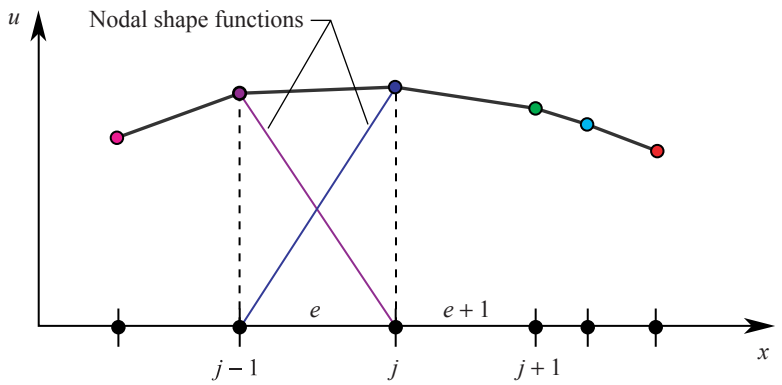
In the RBF interpolation scheme, a smooth function passes through the set of actual nodal values obtained from the radial basis shape functions. Again, because these trial functions are smooth (see fig. 4(c), only shape function for node j is shown), the secondary variables are continuous.

Beam Problem (C^1) Interpolation Schemes

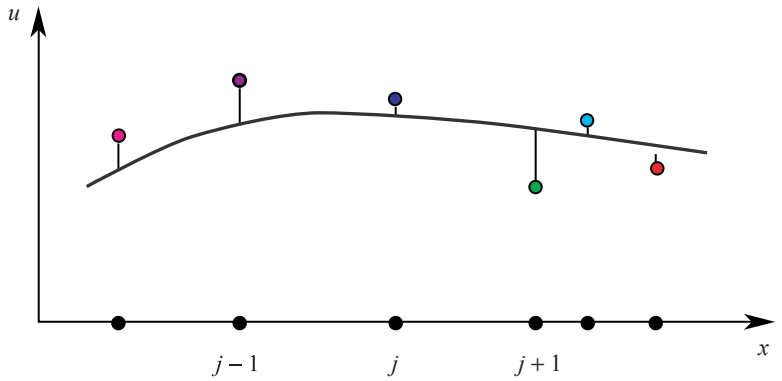
This section presents the interpolation schemes used in the RPG method for beam (C^1) problems. The GMLS interpolation is discussed first; then, the shape functions for both the radial basis and hybrid interpolations are presented. These shape functions will be used in the beam problem implementation of the RPG method.

Generalized MLS (GMLS) Interpolation

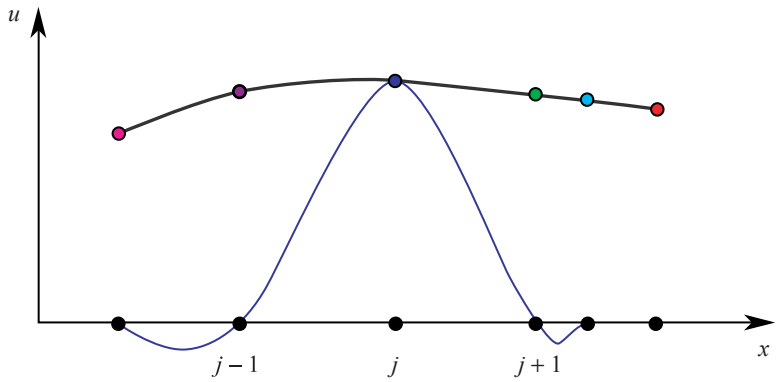
The GMLS approximations are used for beam problems in the traditional MLPG method (Atluri, Cho, and Kim, 1999; Phillips and Raju, 2002; Raju and Phillips, 2002, 2003a, and 2003c). The primary



(a) FEM.



(b) MLS.



(c) RBF.

Figure 4. Comparison of FEM, MLS, and RBF trial functions for C^0 problems.

variables for these problems are the deflection w and the slope θ ($\theta = dw/dx$). The shape functions for w and θ, using a local coordinate approach (Raju and Phillips, 2003a and 2003c), are

$$\psi_j^{(w)}(x) = \sum_{g=1}^{m} p_g(\xi_j)\left[[\mathbf{A}]^{-1}[\mathbf{P}]^{\mathrm{T}}[\lambda]\right]_{gj} \quad \text{and}$$
$$\psi_j^{(\theta)}(x) = \sum_{g=1}^{m} p_g(\xi_j)\left[[\mathbf{A}]^{-1}[\mathbf{P}_x]^{\mathrm{T}}\ [\lambda]\right]_{gj} \tag{38}$$

or

$$\psi_j^{(w)} = \sum_{g=1}^{m} p_g\left[[\mathbf{A}]^{-1}[\mathbf{B}_w]\right]_{gj} \tag{39a}$$

and

$$\psi_j^{(\theta)} = \sum_{g=1}^{m} p_g\left[[\mathbf{A}]^{-1}[\mathbf{B}_\theta]\right]_{gj} \tag{39b}$$

where

$$\left[[\mathbf{B}_w] \quad [\mathbf{B}_\theta]\right] = \left[[\mathbf{P}]^{\mathrm{T}}[\lambda] \quad [\mathbf{P}_x]^{\mathrm{T}}[\lambda]\right] \tag{40}$$

In equations (38)–(40), $p(\xi)$ is a polynomial basis function, and

$$[\mathbf{A}] = [\mathbf{P}]^{\mathrm{T}}[\lambda][\mathbf{P}] + [\mathbf{P}_x]^{\mathrm{T}}[\lambda][\mathbf{P}_x] \tag{41}$$

In equation (41), $[\mathbf{P}]$ and $[\mathbf{P}_x]$ are (n, m) matrices, and $[\boldsymbol{\lambda}]$ is a diagonal (n, n) matrix defined as

$$[\mathbf{P}] = \left[\mathbf{p}^{\mathrm{T}}(\xi_1) \quad \mathbf{p}^{\mathrm{T}}(\xi_2) \quad \ldots \quad \mathbf{p}^{\mathrm{T}}(\xi_n)\right]^{\mathrm{T}} \tag{42a}$$

$$[\mathbf{P}_x] = \left[\mathbf{p}_x^{\mathrm{T}}(\xi_1) \quad \mathbf{p}_x^{\mathrm{T}}(\xi_2) \quad \ldots \quad \mathbf{p}_x^{\mathrm{T}}(\xi_n)\right]^{\mathrm{T}} \tag{42b}$$

$$\lambda = \begin{bmatrix} \lambda_1(x) & & & \\ & \lambda_2(x) & & \\ & & \ddots & \\ & & & \lambda_n(x) \end{bmatrix} \tag{43}$$

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

$$\mathbf{p}^{\mathrm{T}}(\xi) = \begin{bmatrix} 1, & \xi, & \xi^2, & \ldots & \xi^{m-1} \end{bmatrix} \tag{44a}$$

and

$$\mathbf{p}_x^{\mathrm{T}}(\xi) = \frac{d\mathbf{p}^{\mathrm{T}}(\xi)}{dx} = \begin{bmatrix} 0, & 1, & 2\xi, & \ldots & (m-1)\xi^{m-2} \end{bmatrix} \tag{44b}$$

as

$$\frac{d}{dx}(\ ) = \frac{d}{d\xi}(\ ) \tag{45}$$

In C^1 problems, the MLPG method requires the first, second, and third derivatives (Atluri, Cho, and Kim, 1999; Gu and Liu, 2001a; Phillips and Raju, 2002; and Raju and Phillips, 2003a). The first derivatives of ψ_j are

$$\begin{aligned} \frac{d\psi_j^{(w)}}{dx} = \psi_{j,x}^{(w)} = \sum_{g=1}^{m} \Big\{ & p_{g,x} \Big([\mathbf{A}]^{-1}[\mathbf{B}_w] \Big)_{gj} \\ & + p_g \Big([\mathbf{A}]^{-1}[\mathbf{B}_w]_{,x} + [\mathbf{A}]_{,x}^{-1}[\mathbf{B}_w] \Big)_{gj} \Big\} \end{aligned} \tag{46a}$$

and

$$\begin{aligned} \psi_{j,x}^{(\theta)} = \sum_{g=1}^{m} \Big\{ & p_{g,x} \Big([\mathbf{A}]^{-1}[\mathbf{B}_\theta] \Big)_{gj} \\ & + p_g \Big([\mathbf{A}]^{-1}[\mathbf{B}_\theta]_{,x} + [\mathbf{A}]_{,x}^{-1}[\mathbf{B}_\theta] \Big)_{gj} \Big\} \end{aligned} \tag{46b}$$

where $[\mathbf{A}]_{,x}^{-1}$ is defined in equation (7). The second and third derivatives involve considerably more complex expressions containing $[\mathbf{A}]^{-1}$, $[\mathbf{A}]_{,x}^{-1}$, $[\mathbf{A}]_{,xx}^{-1}$, and so on, and the detailed derivations are given in Phillips and Raju (2002) and are not repeated here. For thin plate problems (2-D C^1 problems), these derivatives become even more complicated (see Long and Atluri, 2002 for details). Evaluation of the derivatives is computationally expensive, which appears to be a disadvantage of using the GMLS for the trial functions. Thus, an alternate formulation for trial functions that is computationally simple and yet as accurate as the GMLS approximations is sought. The RBFs appear to be good candidates for achieving such a purpose because the shape functions obtained from radial basis interpolation are simpler than the shape functions presented previously for the GMLS. More importantly, the derivatives of the radial basis shape functions are simple and involve considerably fewer matrix inversion and multiplication operations than the derivatives of the GMLS shape functions. The RBFs are discussed next.

RBF Interpolation

In the beam problems considered in this paper, the deflection w and the slope $\theta = dw/dx$ are the primary variables. The interpolating function for $w(x)$ is assumed to be of the form

$$w(x) = R_1(x)a_1 + S_1(x)b_1 + R_2(x)a_2 + S_2(x)b_2 + \ldots + R_N(x)a_N + S_N(x)b_N \tag{47}$$

where a_j and b_j, $j = 1, 2, \ldots, N$, are unknown coefficients, $R_j(x)$ are the RBFs, and $S_j(x) = dR_j(x)/dx$ at node j. 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 (eq. (47)); thus, the approximations for θ are written as

$$\begin{aligned}\theta(x) = \frac{dw(x)}{dx} &= \frac{dR_1(x)}{dx}a_1 + \frac{dS_1(x)}{dx}b_1 \\ &+ \frac{dR_2(x)}{dx}a_2 + \frac{dS_2(x)}{dx}b_2 + \ldots + \frac{dR_N(x)}{dx}a_N + \frac{dS_N(x)}{dx}b_N\end{aligned} \tag{48}$$

In matrix form, equation (47) is written as

$$w(x) = \mathbf{Q}^{\mathrm{T}}(x)\{\mathbf{c}\} \tag{49}$$

where

$$\begin{aligned}\mathbf{Q}^{\mathrm{T}}(x) &= \begin{bmatrix} R_1(x) & S_1(x) & R_2(x) & S_2(x) & \ldots & R_N(x) & S_N(x) \end{bmatrix} \\ \{\mathbf{c}\} &= \begin{Bmatrix} a_1 & b_1 & a_2 & b_2 & \ldots & a_N & b_N \end{Bmatrix}^{\mathrm{T}}\end{aligned} \tag{50}$$

Similarly, equation (48) is written as

$$\theta(x) = \frac{d\mathbf{Q}^{\mathrm{T}}(x)}{dx}\{\mathbf{c}\} \tag{51}$$

where

$$\frac{d\mathbf{Q}^{\mathrm{T}}(x)}{dx} = \begin{bmatrix} \frac{dR_1(x)}{dx} & \frac{dS_1(x)}{dx} & \frac{dR_2(x)}{dx} & \frac{dS_2(x)}{dx} & \ldots & \frac{dR_N(x)}{dx} & \frac{dS_N(x)}{dx} \end{bmatrix} \tag{52}$$

The interpolations in equations (50) and (52) are required to pass through N nodal values, which yields

$$\underset{(2N,\ 2N)}{[\mathbf{Q}_{\mathrm{B}}]} \quad \underset{(2N,\ 1)}{\{\mathbf{c}\}} \quad = \quad \underset{(2N,\ 1)}{\{\mathbf{d}\}} \tag{53}$$

where

$$\{\mathbf{d}\}^{\mathrm{T}} = \begin{Bmatrix} w_1 & \theta_1 & w_2 & \theta_2 & \ldots & w_N & \theta_N \end{Bmatrix} \tag{54}$$

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

$$[\mathbf{Q}_{\mathrm{B}}] = \begin{bmatrix} R_1(x_1) & S_1(x_1) & R_2(x_1) & S_2(x_1) & \cdots & R_N(x_1) & S_N(x_1) \\ \dfrac{dR_1(x_1)}{dx} & \dfrac{dS_1(x_1)}{dx} & \dfrac{dR_2(x_1)}{dx} & \dfrac{dS_2(x_1)}{dx} & \cdots & \dfrac{dR_N(x_1)}{dx} & \dfrac{dS_N(x_1)}{dx} \\ R_1(x_2) & S_1(x_2) & R_2(x_2) & S_2(x_2) & \cdots & R_N(x_2) & S_N(x_2) \\ \dfrac{dR_1(x_2)}{dx} & \dfrac{dS_1(x_2)}{dx} & \dfrac{dR_2(x_2)}{dx} & \dfrac{dS_2(x_2)}{dx} & \cdots & \dfrac{dR_N(x_2)}{dx} & \dfrac{dS_N(x_2)}{dx} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ R_1(x_N) & S_1(x_N) & R_2(x_N) & S_2(x_N) & \cdots & R_N(x_N) & S_N(x_N) \\ \dfrac{dR_1(x_N)}{dx} & \dfrac{dS_1(x_N)}{dx} & \dfrac{dR_2(x_N)}{dx} & \dfrac{dS_2(x_N)}{dx} & \cdots & \dfrac{dR_N(x_N)}{dx} & \dfrac{dS_N(x_N)}{dx} \end{bmatrix} \tag{55}$$

The unknown coefficients in equation (53) are obtained as

$$\{\mathbf{c}\} = [\mathbf{Q}_{\mathrm{B}}]^{-1}\{\mathbf{d}\} \tag{56}$$

The interpolation for w in equation (49) can be written as

$$\begin{aligned} w(x) &= \mathbf{Q}^{\mathrm{T}}(x)[\mathbf{Q}_{\mathrm{B}}]^{-1}\{\mathbf{d}\} \\ &= \sum_{j=1}^{2N} \varphi_j(x) d_j \end{aligned} \tag{57}$$

where φ are the nodal shape functions

$$\begin{aligned} \varphi(x) &= \mathbf{Q}^{\mathrm{T}}(x)[\mathbf{Q}_{\mathrm{B}}]^{-1} \\ &= \begin{bmatrix} \psi_1^{(w)}(x) & \psi_1^{(\theta)}(x) & \psi_2^{(w)}(x) & \psi_2^{(\theta)}(x) & \cdots & \psi_N^{(w)}(x) & \psi_N^{(\theta)}(x) \end{bmatrix} \end{aligned} \tag{58}$$

From equation (58), the individual shape functions for deflection and slope, $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$, are

$$\begin{aligned} \psi_l^{(w)}(x) &= \sum_{k=1}^{N} \left(R_k(x) \cdot \eta_{(2k-1)(2l-1)} + S_k(x) \cdot \eta_{(2k)(2l-1)} \right) \\ \psi_l^{(\theta)}(x) &= \sum_{k=1}^{N} \left(R_k(x) \cdot \eta_{(2k-1)(2l)} + S_k(x) \cdot \eta_{(2k)(2l)} \right) \end{aligned} \tag{59}$$

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

$$\frac{d\psi_l^{(w)}(x)}{dx} = \sum_{k=1}^{N}\left(\frac{dR_k(x)}{dx}\cdot\eta_{(2k-1)(2l-1)} + \frac{dS_k(x)}{dx}\cdot\eta_{(2k)(2l-1)}\right)$$
$$\frac{d\psi_l^{(\theta)}(x)}{dx} = \sum_{k=1}^{N}\left(\frac{dR_k(x)}{dx}\cdot\eta_{(2k-1)(2l)} + \frac{dS_k(x)}{dx}\cdot\eta_{(2k)(2l)}\right) \tag{60}$$

$$\frac{d^2\psi_l^{(w)}(x)}{dx^2} = \sum_{k=1}^{N}\left(\frac{d^2R_k(x)}{dx^2}\cdot\eta_{(2k-1)(2l-1)} + \frac{d^2S_k(x)}{dx^2}\cdot\eta_{(2k)(2l-1)}\right)$$
$$\frac{d^2\psi_l^{(\theta)}(x)}{dx^2} = \sum_{k=1}^{N}\left(\frac{d^2R_k(x)}{dx^2}\cdot\eta_{(2k-1)(2l)} + \frac{d^2S_k(x)}{dx^2}\cdot\eta_{(2k)(2l)}\right) \tag{61}$$

$$\frac{d^3\psi_l^{(w)}(x)}{dx^3} = \sum_{k=1}^{N}\left(\frac{d^3R_k(x)}{dx^3}\cdot\eta_{(2k-1)(2l-1)} + \frac{d^3S_k(x)}{dx^3}\cdot\eta_{(2k)(2l-1)}\right)$$
$$\frac{d^3\psi_l^{(\theta)}(x)}{dx^3} = \sum_{k=1}^{N}\left(\frac{d^3R_k(x)}{dx^3}\cdot\eta_{(2k-1)(2l)} + \frac{d^3S_k(x)}{dx^3}\cdot\eta_{(2k)(2l)}\right) \tag{62}$$

Hybrid RBF Interpolation

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

$$\begin{aligned} w(\mathbf{x}) &= \mathbf{Q}^{\mathrm{T}}(\mathbf{x})\ \{\mathbf{c}\} + \mathbf{p}^{\mathrm{T}}(\mathbf{x})\ \{\mathbf{z}\} \\ &= \begin{bmatrix}\mathbf{Q}^{\mathrm{T}}(\mathbf{x}) & \mathbf{p}^{\mathrm{T}}(\mathbf{x})\end{bmatrix}\begin{Bmatrix}\mathbf{c}\\ \mathbf{z}\end{Bmatrix} \end{aligned} \tag{63}$$

where $\{\mathbf{c}\}$ and $\mathbf{Q}^{\mathrm{T}}(\mathbf{x})$ are as in equations (50), and $\mathbf{p}^{\mathrm{T}}(x)$ are the polynomial basis functions

$$\begin{aligned} \mathbf{p}^{\mathrm{T}}(x) &= \begin{bmatrix} p_1(x), & p_2(x), & p_3(x), & \ldots, & p_m(x)\end{bmatrix} \\ &= \begin{bmatrix} 1, & x, & x^2, & \ldots, & x^{m-1}\end{bmatrix} \end{aligned}$$

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

$$\{\mathbf{z}\} = \begin{Bmatrix} z_1, & z_2, & z_3, & \ldots, & z_m\end{Bmatrix}^{\mathrm{T}} \tag{64}$$

The interpolation of equation (63) is required to pass through the N points with constraints

$$\sum_{j=1}^{N}\left(p_k(x_j)c_{2j-1} + \frac{dp_k(x_j)}{dx}c_{2j} \right) = 0 \tag{65}$$

where $k = 1, 2, \ldots, m$, and $\{\mathbf{c}\}$ are the unknown coefficients in equations (50) (Wang and Liu, 2002a and 2002b). Equation (65) is imposed to guarantee a unique approximation. The set of equations to determine the coefficients $\{\mathbf{c}\}$ and $\{\mathbf{z}\}$ is thus written as

$$\begin{bmatrix} \mathbf{Q}_\mathrm{B} & \mathbf{T}_\mathrm{B} \\ \mathbf{T}_\mathrm{B}^\mathrm{T} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{c} \\ \mathbf{z} \end{Bmatrix} = \begin{Bmatrix} \mathbf{d} \\ \mathbf{0} \end{Bmatrix} \quad \text{or}$$

$$[\boldsymbol{G}] \begin{Bmatrix} \mathbf{c} \\ \mathbf{z} \end{Bmatrix} = \begin{Bmatrix} \mathbf{d} \\ \mathbf{0} \end{Bmatrix} \tag{66}$$

where $\{\mathbf{d}\}^\mathrm{T}$ and $[\mathbf{Q}_\mathrm{B}]$ are defined in equations (54) and (55), and $[\mathbf{T}_\mathrm{B}]$ is a $(2N, m)$ matrix

$$\mathbf{T}_\mathrm{B} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ \dfrac{dp_1(x_1)}{dx} & \dfrac{dp_2(x_1)}{dx} & \cdots & \dfrac{dp_m(x_1)}{dx} \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \dfrac{dp_1(x_2)}{dx} & \dfrac{dp_2(x_2)}{dx} & \cdots & \dfrac{dp_m(x_2)}{dx} \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_N) & p_2(x_N) & \cdots & p_m(x_N) \\ \dfrac{dp_1(x_N)}{dx} & \dfrac{dp_2(x_N)}{dx} & \cdots & \dfrac{dp_m(x_N)}{dx} \end{bmatrix} \tag{67}$$

The unknown coefficients in equation (66) are obtained as

$$\begin{Bmatrix} \mathbf{c} \\ \mathbf{z} \end{Bmatrix} = [\boldsymbol{G}]^{-1} \begin{Bmatrix} \mathbf{d} \\ \mathbf{0} \end{Bmatrix} \tag{68}$$

The interpolating function for w in equation (63) is now written as

$$\begin{aligned} w(x) &= \begin{bmatrix} \mathbf{Q}^\mathrm{T}(x) & \mathbf{p}^\mathrm{T}(x) \end{bmatrix} [\boldsymbol{G}]^{-1} \begin{Bmatrix} \mathbf{d} \\ \mathbf{0} \end{Bmatrix} \\ &= \sum_{j=1}^{2N} \varphi_j(x) d_j \end{aligned} \tag{69}$$

where φ are the nodal shape functions

$$\varphi(x) = \left\lfloor \mathbf{Q}^{\mathrm{T}}(x) \quad \mathbf{p}^{\mathrm{T}}(x) \right\rfloor [\boldsymbol{G}]^{-1}$$

$$= \left[\psi_1^{(w)}(x) \quad \psi_1^{(\theta)}(x) \quad \psi_2^{(w)}(x) \quad \psi_2^{(\theta)}(x) \quad \ldots \quad \psi_N^{(w)}(x) \quad \psi_N^{(\theta)}(x) \right] \tag{70}$$

From equation (70), the shape functions for the deflection and slope $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$ may be written as

$$\begin{aligned} \psi_l^{(w)}(x) &= \sum_{j=1}^{N} \left(R_j(x) \cdot \zeta_{(2j-1)(2l-1)} + S_j(x) \cdot \zeta_{(2j)(2l-1)} \right) \\ &\quad + \sum_{k=1}^{m} p_k(x) \cdot \zeta_{(2N+k)(2l-1)} \\ \psi_l^{(\theta)}(x) &= \sum_{j=1}^{N} \left(R_j(x) \cdot \zeta_{(2j-1)(2l)} + S_j(x) \cdot \zeta_{(2j)(2l)} \right) \\ &\quad + \sum_{k=1}^{m} p_k(x) \cdot \zeta_{(2N+k)(2l)} \end{aligned} \tag{71}$$

where ζ_{kl} are the elements of the matrix $[\boldsymbol{G}]^{-1}$. The derivatives of these shape functions are easy to evaluate, as in equations (60)–(62).

MLPG Equations for Beam (C^1) Problems

Beam problems are governed by the fourth-order equation

$$EI \frac{d^4 w}{dx^4} = f \quad \text{in} \quad 0 \le x \le L \tag{72}$$

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

$$\left.\begin{aligned} \theta &= \frac{dw}{dx} \\ M &= EI \frac{d^2 w}{dx^2} \\ V &= -EI \frac{d^3 w}{dx^3} \end{aligned}\right\} \tag{73}$$

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, that is, if w is prescribed, V cannot be prescribed, and vice versa.

The MLPG equations are derived using a weighted residual weak form of the governing equation (eq. (72)). The MLPG equations are (Atluri, Cho, and Kim, 1999; Phillips and Raju, 2002; and Raju and Phillips, 2002 and 2003a)

$$\mathbf{K}^{(\text{node})}\mathbf{d} + \mathbf{K}^{(\text{bdry})}\mathbf{d} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = \mathbf{0} \tag{74}$$

where the superscript "bdry" denotes boundary,

$$\mathbf{d}^{\mathrm{T}} = \left\{ w_1 \quad \theta_1 \quad w_2 \quad \theta_2 \quad \ldots \quad w_N \quad \theta_N \right\} \tag{75a}$$

are the nodal values of deflections w and slopes θ at the N nodes in the model of the beam, and

$$\mathbf{K}^{(\text{node})} = \left\lfloor \mathbf{k}_{ij}^{(\text{node})} \right\rfloor \tag{75b}$$

$$\mathbf{K}^{(\text{bdry})} = \left\lfloor \mathbf{k}_{ij}^{(\text{bdry})} \right\rfloor \tag{75c}$$

with

$$\begin{aligned}
\mathbf{k}_{ij}^{(\text{node})} = EI & \begin{bmatrix} \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx \\ \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(w)}}{dx^2}dx & \int\limits_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2}\frac{d^2\psi_j^{(\theta)}}{dx^2}dx \end{bmatrix} \\
& + n_x EI \begin{bmatrix} \chi_i^{(w)}\frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)}\frac{d^3\psi_j^{(\theta)}}{dx^3} \\ \chi_i^{(\theta)}\frac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)}\frac{d^3\psi_j^{(\theta)}}{dx^3} \end{bmatrix}_{\Gamma_{sI}^{(i)}} \\
& - n_x EI \begin{bmatrix} \frac{d\chi_i^{(w)}}{dx}\frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(w)}}{dx}\frac{d^2\psi_j^{(\theta)}}{dx^2} \\ \frac{d\chi_i^{(\theta)}}{dx}\frac{d^2\psi_j^{(w)}}{dx^2} & \frac{d\chi_i^{(\theta)}}{dx}\frac{d^2\psi_j^{(\theta)}}{dx^2} \end{bmatrix}_{\Gamma_{sI}^{(i)}}
\end{aligned} \tag{75d}$$

$$\mathbf{k}_{ij}^{(\mathrm{bdry})} = \alpha_w \begin{bmatrix} \chi_i^{(w)}\psi_j^{(w)} & \chi_i^{(w)}\psi_j^{(\theta)} \\ \chi_i^{(\theta)}\psi_j^{(w)} & \chi_i^{(\theta)}\psi_j^{(\theta)} \end{bmatrix}_{\Gamma_{sw}^{(i)}}$$

$$+ n_x EI \begin{bmatrix} \chi_i^{(w)} \dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)} \dfrac{d^3\psi_j^{(\theta)}}{dx^3} \\ \chi_i^{(\theta)} \dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)} \dfrac{d^3\psi_j^{(\theta)}}{dx^3} \end{bmatrix}_{\Gamma_{sw}^{(i)}}$$

$$+ \alpha_\theta \begin{bmatrix} \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d\psi_j^{(w)}}{dx} & \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d\psi_j^{(\theta)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d\psi_j^{(w)}}{dx} & \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d\psi_j^{(\theta)}}{dx} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}}$$

$$- n_x EI \begin{bmatrix} \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2} \\ \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}} \tag{75e}$$

$$\mathbf{f}^{(\mathrm{node})} = \begin{Bmatrix} \displaystyle\int_{\Omega_s^{(i)}} \chi_i^{(w)} f \, dx \\ \displaystyle\int_{\Omega_s^{(i)}} \chi_i^{(\theta)} f \, dx \end{Bmatrix} \tag{75f}$$

and

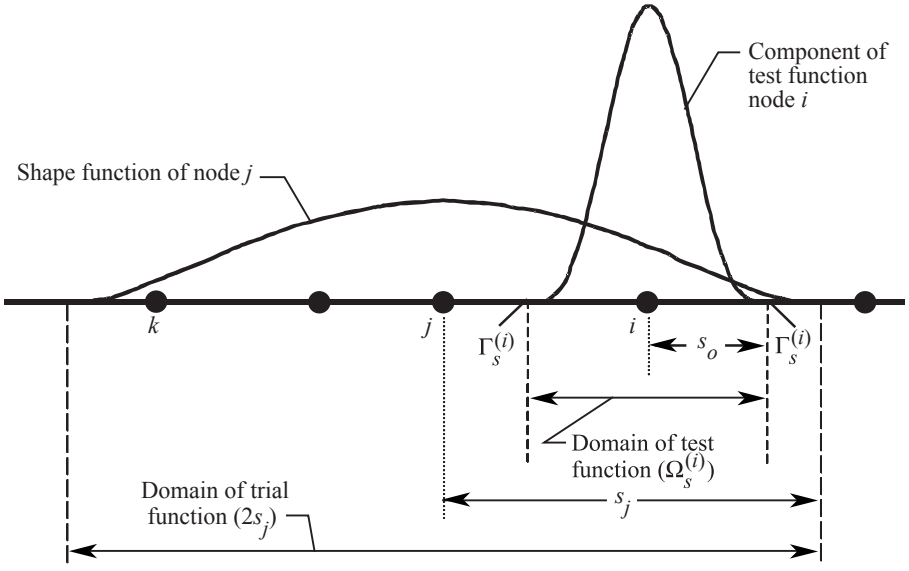
$$\mathbf{f}^{(\mathrm{bdry})} = n_x \tilde{M} \begin{Bmatrix} \dfrac{d\chi_i^{(w)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{sM}^{(i)}} + n_x \tilde{V} \begin{Bmatrix} \chi_i^{(w)} \\ \chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sV}^{(i)}}$$

$$+ \alpha_w \tilde{w} \begin{Bmatrix} \chi_i^{(w)} \\ \chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sw}^{(i)}} + \alpha_\theta \tilde{\theta} \begin{Bmatrix} \dfrac{d\chi_i^{(w)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{s\theta}^{(i)}} \tag{75g}$$

where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, n$, 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 fig. 5(b)) is the local subdomain of the test function at node i (see fig. 5(a)), n_x is the unit outward normal to $\Omega_s^{(i)}$ ($n_x = -1$ at the left boundary point, and $n_x = +1$ at the right boundary point), and $\Gamma_s^{(i)}$ are the boundary points of $\Omega_s^{(i)}$ (see fig. 5(b)). 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 $\tilde{w}$, $\tilde{\theta}$, $\tilde{M}$, and $\tilde{V}$ are prescribed values of the deflection, slope, moment, and shear, respectively. See Raju and Phillips (2003a) for a detailed explanation of these terms.

The system of equations presented in equations (74)–(75g) 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.

(a) N-node model of beam.



(b) Components of trial and test functions.

Figure 5. Comparison of domains of trial and test functions.

Trial Functions

The trial functions are chosen as

$$w(x) = \sum_{j=1}^{N} \left(\psi_j^{(w)}(x) w_j + \psi_j^{(\theta)}(x) \theta_j \right) \tag{76}$$

where $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$ are the radial basis shape functions of equations (59), and w_j and θ_j are the nodal values of w and θ at the N nodes (eq. (54)). Note that in MLPG algorithms that employ the moving least squares interpolation scheme for the trial functions, the values $\mathbf{d}$ (eq. (75a)) are fictitious nodal values $\hat{\mathbf{d}}$. In this paper, RBFs pass through the actual nodal values $\mathbf{d}$.

Test Functions

The test function v is assumed as in Raju and Phillips (2003a) as

$$v(x) = \mu_i^{(w)} \chi_i^{(w)}(x) + \mu_i^{(\theta)} \chi_i^{(\theta)}(x) \tag{77}$$

where $\mu_i^{(w)}$ and $\mu_i^{(\theta)}$ are arbitrary constants. In this paper, the $\chi_i^{(w)}(x)$ components of the test functions are chosen as power weight functions (Raju and Phillips, 2003a)

$$\chi_i^{(w)}(x) = \begin{cases} \left[1 - \left(\dfrac{d_i}{s_o} \right)^2 \right]^4 & \text{if} \quad 0 \le d_i \le s_o \\ 0 & \text{if} \quad d_i > s_o \end{cases} \tag{78}$$

with $d_i = \|x - x_i\|$. In equation (78), s_o is a user-defined parameter that determines the extent of the test functions (and hence Ω_s; see fig. 5). 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, that is,

$$\chi_i^{(\theta)} = \frac{d\chi_i^{(w)}}{dx} \tag{79}$$

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 Raju and Phillips (2003a), when this test function is used, the $\mathbf{k}^{(\text{node})}$ in equation (75d) reduces to

$$\mathbf{k}_{ij}^{(\text{node})} = EI \begin{bmatrix} \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2} \frac{d^2\psi_j^{(w)}}{dx^2} dx & \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(w)}}{dx^2} \frac{d^2\psi_j^{(\theta)}}{dx^2} dx \\ \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2} \frac{d^2\psi_j^{(w)}}{dx^2} dx & \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2\chi_i^{(\theta)}}{dx^2} \frac{d^2\psi_j^{(\theta)}}{dx^2} dx \end{bmatrix} \tag{80}$$

Numerical Evaluations

In this section and the sections that follow, several example problems are presented to validate the RPG method for beam (C^1) problems. First, the beam configurations and models used in the numerical evaluations are described, and then three patch test problems are described. Next, the compact RBF interpolations and their hybrid form are used in the RPG method to study the patch tests and simple mixed boundary value problems. The performance of noncompactly supported RBFs is presented next. Finally, the RPG method with a noncompact cubic function is used to study problems with complex loading and boundary conditions.

Beam Configurations and Models

A beam of constant flexural rigidity EI and a length of $4l$ is considered. The length $4l$ is 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 6 shows a typical 17-node model. The distances between the nodes ($\Delta x/l$) 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 (a 12-point Gaussian integration) is used to integrate the terms in the system of equations shown in equations (75d) and (75f).

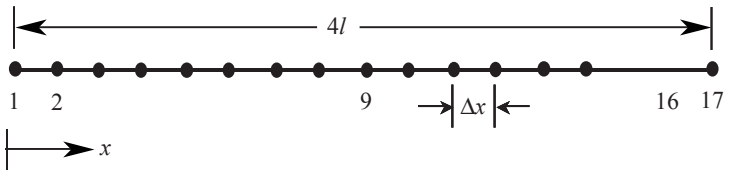


Figure 6. A 17-node model of beam.

Patch Tests

The radial basis MLPG (RPG) method is evaluated by applying the method to simple patch-test problems. The problems considered are

(a) rigid body translation:

$$w(x) = \beta_0 \qquad \theta = \frac{dw}{dx} = 0 \tag{81a}$$

(b) rigid body rotation:

$$w(x) = \beta_1 x \qquad \theta = \beta_1 \tag{81b}$$

(c) constant-curvature condition:

$$w(x) = \beta_2 x^2/2 \qquad \theta = \beta_2 x \tag{81c}$$

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 that correspond to problems (a), (b), and (c) are prescribed as essential boundary conditions

(EBCs) at $x = 0$ and $x = 4l$. With these EBCs, the beam problems are analyzed by 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.

Compact RBF

The compact radial functions described by equations (26) and (27) were considered first. When using the compactly supported functions, equations (55)–(62) are evaluated with $N = n$, the number of nodes in the influence domain of the point x under consideration (Wang and Lu, 2002a and 2002b). This use of the compact functions forces the $[\mathbf{Q}_B]$ of equation (55) to become a $(2n, 2n)$ matrix that must be evaluated once for every node in the model. For these functions, the value of s_o (eq. (78)) is chosen as $(s_o/l = 2\Delta x/l)$, and the value of s_j (eqs. (26) and (27)) is chosen as $(s_j/l = 8\Delta x/l)$.

The RPG method with no polynomial terms was *unable* to reproduce the exact solutions in equation (81) and thus *failed the patch tests*. A quadratic polynomial basis ($m = 3$; $\mathbf{p}^T(x) = [1, x, x^2]$, $\mathbf{p}_x^T(x) = [0, 1, 2x]$) was used in equation (63), increasing the size of the $[\mathbf{Q}_B]$ matrix to $(2n + m, 2n + m)$. The RPG method with a polynomial basis (the hybrid RPG method) reproduced the solutions in equation (81) 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 (fig. 7). 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 (fig. 8) 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.

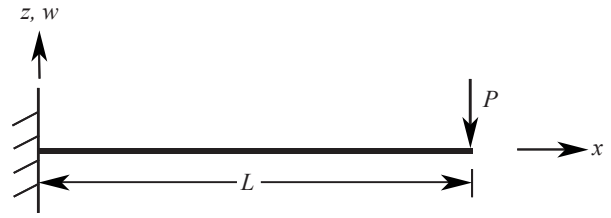


Figure 7. Cantilever beam with tip load.

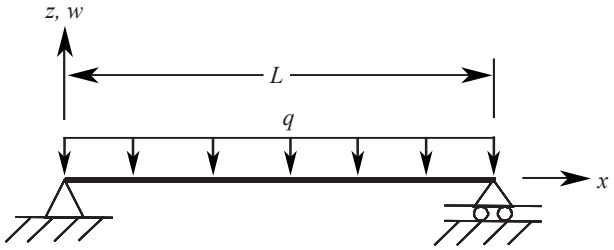


Figure 8. Simply supported beam subjected to uniformly distributed load.

As with the finite element method and the conventional MLPG method (Phillips and Raju, 2002; and Raju and Phillips, 2003a), 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 7 and 8. 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 RBFs. This condition is too restrictive, and hence compact radial functions are dropped from further consideration.

Numerical Evaluations With Noncompactly Supported RBF

The noncompact functions of table 1 are considered next. As stated previously, in these functions,

$$t_j = \frac{r_j}{s_j} \tag{82}$$

where $r_j = \|x - x_j\|$, and s_j is some normalizing distance, usually chosen to be the entire problem domain, Ω (here, Ω is $0 \le x \le L$). Because s_j covers the entire problem domain, $[\mathbf{Q}_B]$ is an $(2N, 2N)$ matrix that is evaluated and inverted once. As a result, the current RPG method involves one inversion of a large matrix and one solution of a large system of equations ($[\mathbf{Q}_B]^{-1}$ to obtain the shape functions and a solution involving $[\mathbf{K}]$). In contrast, the conventional MLPG method involves many inversions of smaller matrices to obtain the shape functions and their derivatives and the solution of a large system of equations that involves the $[\mathbf{K}]$ matrix. Because of this difference, the RPG method *may not be* more computationally efficient than the MLPG method for large models.

Upon examination of all the functions in table 1, the cubic function (with $c = 0$)

$$R_j(x) = t_j^3 \tag{83}$$

was chosen because it is simple and gives nonzero second and third derivatives, as required by the weak form. (Note that $R_j(x) = t_j$ cannot be chosen because the second and higher order derivatives are zero.) The RPG method using equation (83) was applied to the patch tests represented by equations (81). The method successfully reproduced the exact solutions to machine accuracy, thus *passing all the patch tests*. Additionally, all functions of the form

$$R_j(x) = t_j^{(2z-1)} \tag{84}$$

where $z > 1$, performed successfully, although t_j^3 yielded the best results (Raju, Phillips, and Krishnamurthy, 2003b; and Raju and Phillips, 2003d). Note that when this function is chosen, the polynomial terms *are not* included.

Next, the RPG method with the RBF in equation (83) was used to analyze mixed boundary value problems. In the method, a 12-point Gaussian integration was used to integrate the weak form, and the value of (s_o/l), which defines the extent of the test functions (see eq. (78)), was set as $(s_o/l = 2\Delta x/l)$. Because a noncompact function was chosen for the trial functions, the value of (s_j/l) (eq. (82)) was set as $(s_j/l = L/l)$. For the cantilever beam with a tip load in figure 7, the RPG method yielded the exact

solution. The simply supported beam problem with a uniformly distributed load (fig. 8) was analyzed by using 17-, 33-, and 65-node models. The maximum deflection values (i.e., the deflection at ($x = L/2$)), for these three models, obtained by using the RPG method and 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 to integrate the weak form (Raju and Phillips, 2003a), the value of (s_o/l) was set as ($s_o/l = 2\Delta x/l$), and the value of (s_j/l) was set as ($s_j/l = 8\Delta x/l$). From table 2, it is seen that the RPG method yields solutions that are as accurate as the solutions obtained from the conventional MLPG method.

Table 2. Comparison of Maximum Deflection Values Obtained Using RPG and MLPG Methods ($w_{\text{Exact}} = 5qL^4/384EI$)

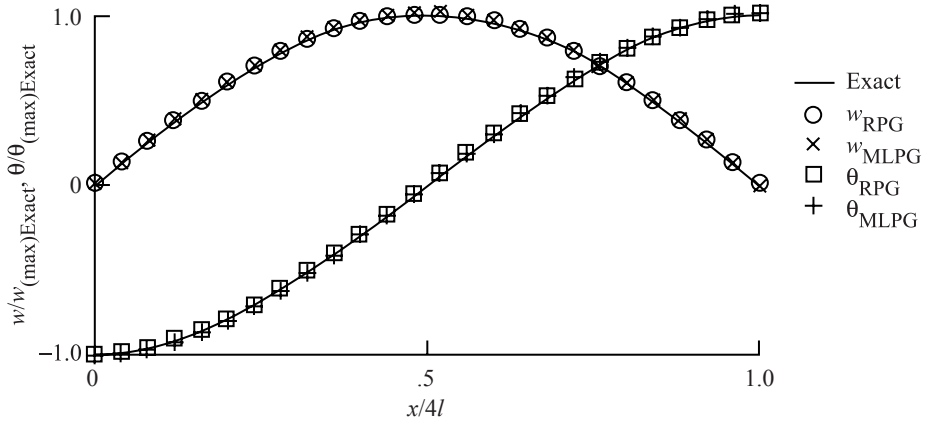
Model	(w/w_{Exact}) at $x = L/2$	
	RPG	MLPG
17-node	0.982	0.993
33-node	1.002	1.012
65-node	1.003	1.015

The results obtained for deflection, slope, moment, and shear by using the 65-node model are presented in figure 9. In this figure, the RPG results are compared to the exact solution and to the solution obtained when using the conventional MLPG method with a quadratic polynomial basis function. The RPG values for deflection, slope, and moment are as accurate as the MLPG values and are in excellent agreement with the exact values. In addition, the RPG values for shear converge with model refinement. The MLPG solution for the shear is erratic and is not shown in figure 9. In the MLPG solution, the quadratic basis function is insufficient to accurately calculate the third derivatives for this problem, and the method cannot recover the values with model refinement; the solution for the shear converges only as the order of the basis function is increased to quartic (Phillips and Raju, 2002). Similar results were obtained by using the 17- and 33-node models; these results are not shown. The results discussed for this problem verify a perceived advantage of the RPG method over the MLPG method; namely, accurate solutions are obtained with simpler evaluation of the shape function derivatives.

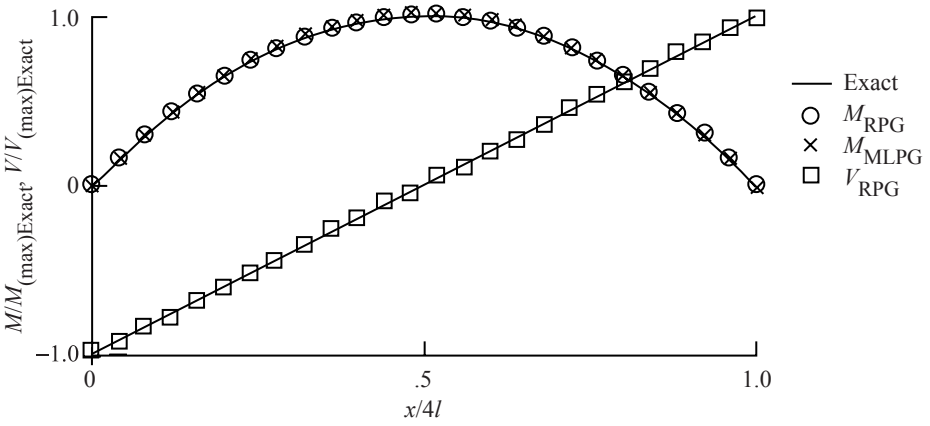
Problems With Complex Loading and Boundary Conditions

The RPG method with the RBF in equation (83) was next applied to the complex beam problems with the loading and boundary conditions shown in figure 10. The problems considered were (a) a cantilever beam ($L = 2l$) long with a uniformly distributed load on a portion of the beam shown in figure 10(a); (b) a continuous beam ($L = 4l$) long with the additional support in the center of the beam (fig. 10(b)); and (c) a fixed continuous beam ($L = 3l$) long with a hinge, as shown in figure 10(c).

The RPG solution (with ($s_o/l = 4\Delta x/l$)) for the cantilever beam problem (problem (a)) exhibited convergence with model refinement. These results are consistent with those reported in Raju and Phillips (2003a), in which this problem was studied by using the conventional MLPG method. The exact MLPG and RPG values for deflection and moment for this problem were obtained by using a 65-node model and are compared in figure 11 (with parameters as reported previously for the simply supported beam problem). The RPG method handled the load discontinuity well and yielded results in overall agreement with the exact solutions.



(a) Primary variables.

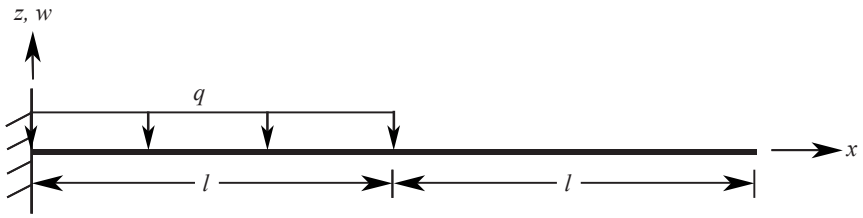


(b) Secondary variables (Note: V_{MLPG} not shown).

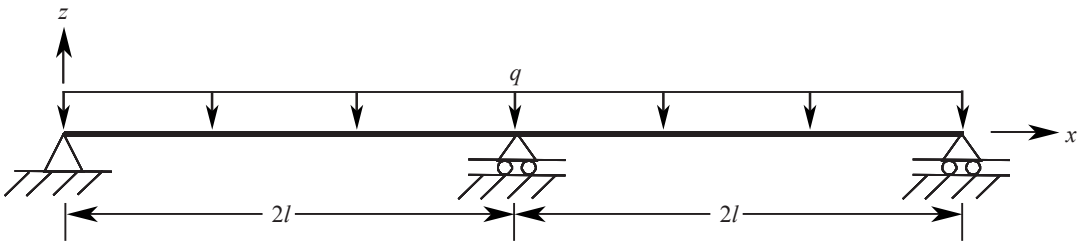
Figure 9. RPG, MLPG, and exact solutions obtained using 65-node model for simply supported beam subjected to uniformly distributed load.

For the continuous beam problem (fig. 10(b)), the RPG method yielded very accurate results for both the primary and secondary variables and handled the discontinuity caused by the additional support well. The MLPG, RPG, and exact moment distributions are shown in figure 12. For this problem, the MLPG method required a large number of nodes to obtain an accurate solution (Raju and Phillips, 2003a), but the RPG method yielded accurate results when the smaller nodal (17-node and 33-node) models were used. These results again verify perceived advantages of the RPG method over the MLPG method; accurate results can be obtained by the RPG method for comparable computing efforts.

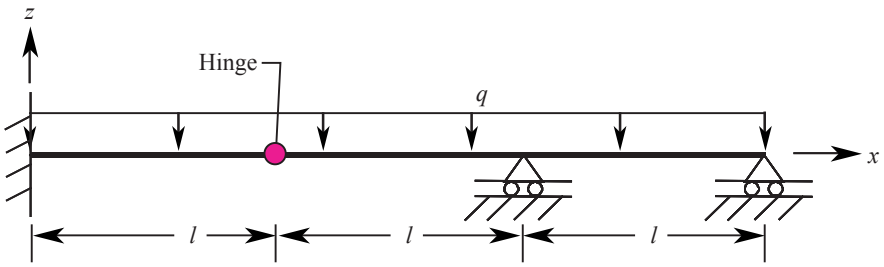
Next, the hinge problem shown in figure 10(c) was considered. Because the hinge at ($x = l$) (see fig. 10(c)) cannot admit the moment, there is a slope discontinuity at ($x = l$). Thus, there are two separate slopes, θ^{L} and θ^{R}, just to the left ($x = l^{-}$) and just to the right ($x = l^{+}$) of the hinge, respectively, while the deflections, w^{L} and w^{R}, are equal. Three models—14-node, 26-node, and 50-node—were considered. The 14-node model consisted of 5 nodes in the left segment ($0 \le x \le l$) of the beam and 9 nodes in the right segment ($l \le x \le 3l$) of the beam. Similarly, the 26-node and 50-node models were composed of 9 nodes in the left and 17 nodes in the right, and 17 nodes in the left and 33 nodes in the right,



(a) Cantilever beam with uniformly distributed load on portion of beam.



(b) Continuous beam subjected to uniformly distributed load.



(c) Fixed continuous beam with hinge.

Figure 10. Cantilever and continuous beams with complex loading and boundary conditions.

respectively. The problem was analyzed by using the MLPG and RPG methods. For both methods, the value of (s_o/l) was set as $(s_o/l = 2\Delta x/l)$. For the MLPG method, the value of (s_j/l) was set as $(s_j/l = 8\Delta x/l)$, and for the RPG method, the value of (s_j/l) was set as $(s_j/l = 1)$ for the left segment and $(s_j/l = 2)$ for the right segment.

The RPG and the exact solutions for the deflection and slope at $x = l$, $2l$, and $3l$ are compared in table 3. In this table, the RPG solutions that were obtained with the three nodal refinements are presented. Accurate solutions are obtained with the coarse model, and the solutions improve with nodal refinement.

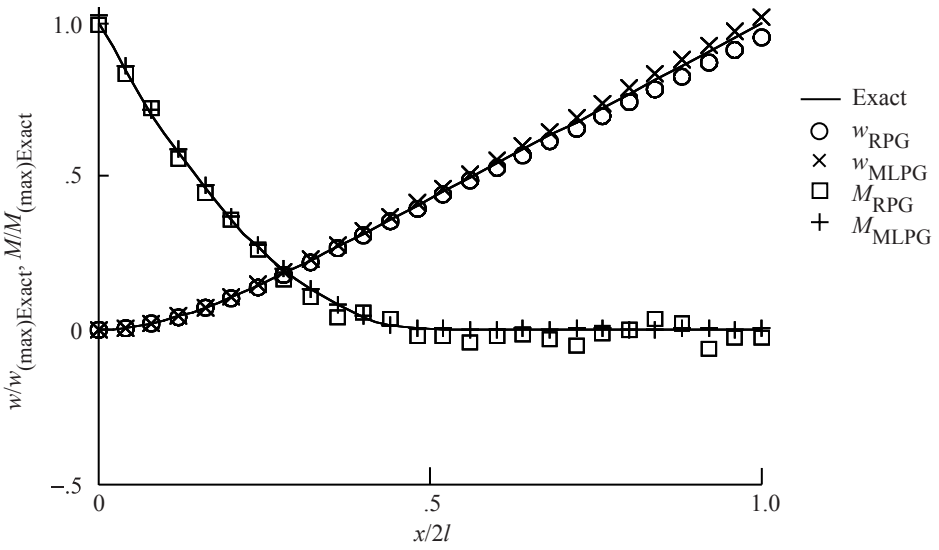


Figure 11. RPG, MLPG, and exact solutions obtained using 65-node model for cantilever beam with uniformly distributed load on portion of beam.

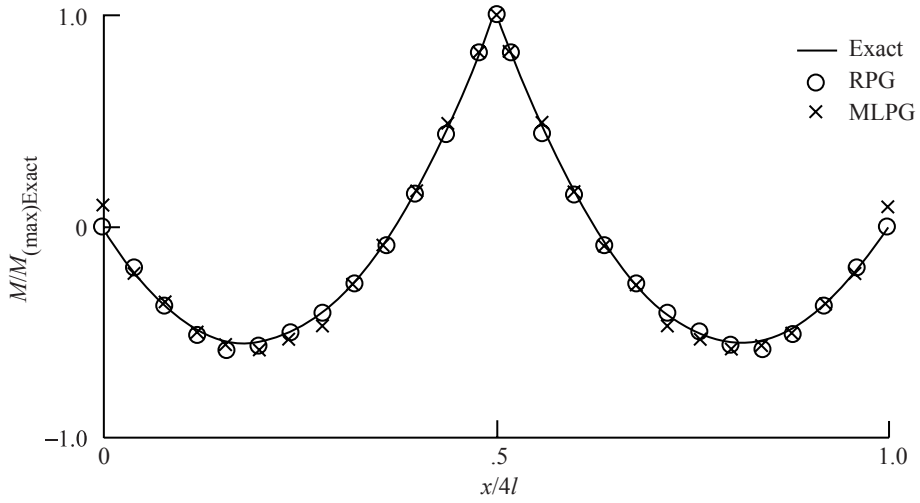


Figure 12. Comparison of RPG (33-node model), MLPG (65-node model), and exact solutions for moment of continuous beam subjected to uniformly distributed load.

Table 3. Comparison of RPG Results and Exact Solutions at Several Locations Along Beam

Model	$w/w_{\text{Exact}}(x=l)$	$\theta^L/\theta^L_{\text{Exact}}(x=l)$	$\theta^R/\theta^R_{\text{Exact}}(x=l)$	$\theta/\theta_{\text{Exact}}(x=2l)$	$\theta/\theta_{\text{Exact}}(x=3l)$
14-node	0.9769	0.9748	0.9679	0.9774	0.8944
26-node	0.9814	0.9789	0.9798	0.9817	0.9656
50-node	0.9880	0.9867	0.9868	0.9882	0.9755

Exact values are $w(x=l)=\dfrac{ql^4}{6EI}$, $\theta^{\text{L}}=\dfrac{11ql^3}{48EI}$, $\theta^{\text{R}}=\dfrac{-3ql^3}{16EI}$, $\theta(x=2l)=\dfrac{-ql^3}{12EI}$, $\theta(x=3l)=\dfrac{ql^3}{48EI}$.

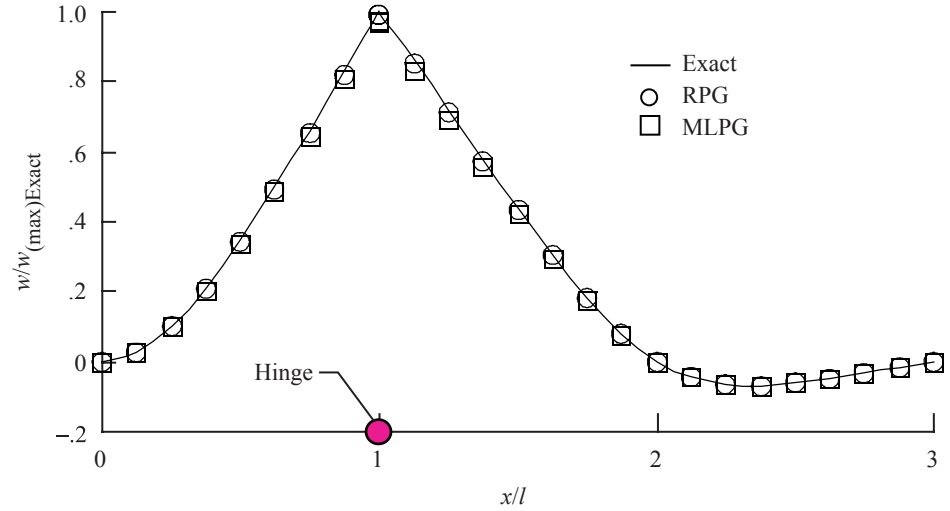


Figure 13. RPG, MLPG, and exact solutions for deflection of fixed continuous, hinged beam subjected to uniformly distributed load.

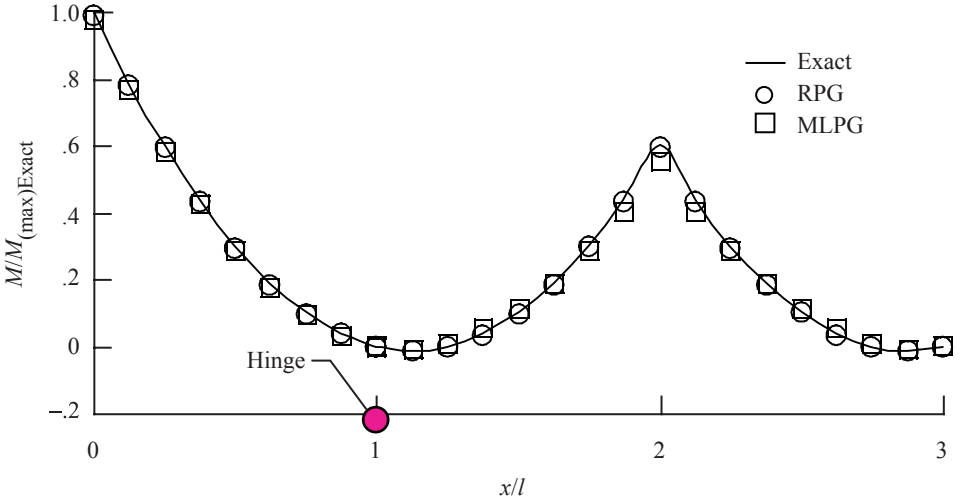


Figure 14. RPG, MLPG, and exact solutions for moment distribution of fixed continuous, hinged beam subjected to uniformly distributed load.

The RPG and MLPG solutions obtained with the 50-node model are compared in figures 13, 14, and 15 for deflection, moment, and shear force, respectively. For the deflection and moment distributions, both the RPG and MLPG solutions agree very well with the exact solutions. The MLPG method is unable to accurately represent the shear force; hence, this solution is not included in the figure 15 comparison. The RPG solution, on the other hand, agreed very well with the exact solution.

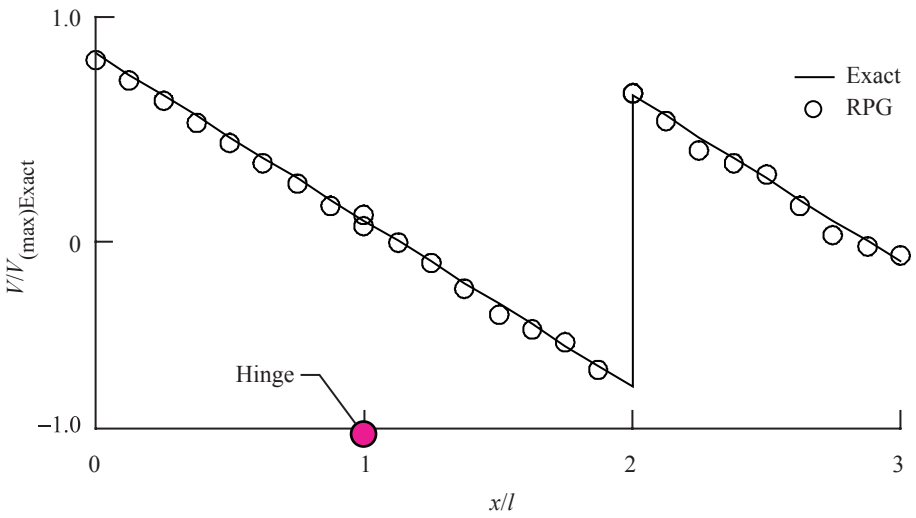


Figure 15. RPG and exact solutions for shear force distribution of fixed continuous, hinged beam subjected to uniformly distributed load (Note: MLPG not shown).

Concluding Remarks

A radial basis function (RBF) implementation of the meshless local Petrov-Galerkin (MLPG) method in the study of Euler-Bernoulli beam problems was presented. 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, rather than generalized moving least squares (GMLS) interpolations, RBFs were used to develop the trial functions, and test functions were chosen as simple weight functions, as in the conventional MLPG method. Two types of RBFs, compact and noncompact, were considered. Compact RBF interpolations were developed with and without including polynomial basis function terms, while noncompact RBF interpolations without polynomial terms were considered.

The compactly supported RBFs 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. Also, polynomial terms tended to overpower the RBFs; hence, the use of compactly supported RBFs presented in this paper is not recommended in the RPG method.

The noncompactly supported cubic RBF performed very well when no polynomial terms were included. The use of RBFs rather than the traditional moving least squares interpolations reduced the computing effort required to solve problems; substantially fewer matrix inversion and multiplication operations were required by the RPG to evaluate the derivatives of the shape functions. When noncompactly supported RBFs were used, the RPG method involved one inversion of a large matrix to evaluate the shape functions. The RPG method with the cubic RBF yielded accurate solutions for the problems studied; the method passed all the patch tests, and for mixed boundary value problems, achieved the same accuracy in results as the conventional MLPG method, with comparable computing effort. The RPG method with a cubic radial basis also yielded very good results in contrast to the conventional MLPG method, which could not yield consistently good results for problems with complex loading and boundary conditions. The accuracy of solutions obtained by the RPG method, combined with the computational simplicity of using the RBFs rather than the GMLS interpolations to approximate the trial functions, makes the RPG method a very attractive variation of the MLPG method.

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 Mech.*, 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. *Computer Modeling in Eng. & Sci.*, vol. 3.1, pp. 11–51.

Atluri, S. N.; and Zhu, T. (1998): A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics. *Computational Mech.*, vol. 22, pp. 117–127.

Belytschko, T.; Lu, Y. Y.; and Gu, L. (1994): Element-Free Galerkin Methods. *International J. for Numerical Methods in Eng.*, vol. 37, pp. 229–256.

Donning, B. M.; and Liu, W. K. (1998): Meshless Methods for Shear-Deformable Beams and Plates. *Computer Methods in Appl. Mech. & Eng.*, vol. 152, pp. 47–71.

Gu, Y. T.; and Liu, G. R. (2001a): A Local Point Interpolation Method for Static and Dynamic Analysis of Thin Beams. *Computer Methods in Appl. Mech. & Eng.*, vol. 190, pp. 5515–5528.

Gu, Y. T.; and Liu, G. R. (2001b): A Meshless Local Petrov-Galerkin (MLPG) Formulation for Static and Free Vibration Analyses of Thin Plates. *CMES: Computer Modeling in Eng. & Sci.*, vol. 2.4, pp. 463–476.

Krysl, P.; and Belytschko, T. (1995): Analysis of Thin Plates by the Element-Free Galerkin Method. *Computational Mech.*, vol. 17, pp. 26–35.

Liu, G. R. (2002): *Mesh Free Methods: Moving Beyond the Finite Element Method*. CRC Press, Boca Raton, USA.

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 Eng. & Sci.*, vol. 3.1, pp. 53–63.

Nayroles, B.; Touzot, G.; and Villon, P. (1992): Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements. *Computational Mech.*, 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.

Powell, M. J. D. (1992): The Theory of Radial Basis Function Approximation in 1990. Advances in Numerical Analysis, 2, Clarendon Press, Oxford, pp. 105–210.

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.

Raju, I. S.; and Phillips, D. R. (2003a): Further Developments in the MLPG Method for Beam Problems. *CMES: Computer Modeling in Eng. & Sci.*, vol. 4.1, pp. 141–159.

Raju, I. S.; Phillips, D. R.; and Krishnamurthy, T. (2003b): Meshless Local Petrov-Galerkin Euler-Bernoulli Beam Problems: a Radial Basis Function Approach. *Proceedings of the 44th SDM Conference*, Norfolk, Virginia, April 7–10, 2003.

Raju, I. S.; and Phillips, D. R. (2003c): Local Coordinate Approach in Meshless Local Petrov-Galerkin Method for Beam Problems. *AIAA J.*, 41.5, pp. 975–978.

Raju, I. S.; and Phillips, D. R. (2003d): A Cubic Radial Basis Function in the MLPG Method for Beam Problems. *Proceedings of the ICES '03 Conference*, Corfu, Greece, July 24–29, 2003, Paper No. 170.

Wang, J. G.; and Liu, G. R. (2002a): A Point Interpolation Meshless Method Based on Radial Basis Functions. *International J. for Numerical Methods in Eng.*, vol. 54, pp. 1623–1648.

Wang, J. G.; and Liu, G. R. (2002b): On the Optimal Shape Parameters of Radial Basis Functions Used for 2-D Meshless Methods. *Computer Methods in Appl. Mech. & Eng.*, vol. 191, pp. 2611–2630.

Wendland, H. (1999): On the Smoothness of Positive Definite and Radial Functions. *J. of Computational and Appl. Math.*, pp. 177–188.

Wu, Z. (1995): Compactly Supported Positive Definite Radial Function. *Advances in Computational Mathematics*, pp. 283–292.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 11 - 2004	Technical Paper	

4. TITLE AND SUBTITLE

A Meshless Method Using Radial Basis Functions for Beam Bending Problems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Raju, I. S.; Phillips, D. R.; and Krishnamurthy, T.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

23-762-55-TH

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-19056

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TP-2004-212996

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 39
Availability: NASA CASI (301) 621-0390

13. SUPPLEMENTARY NOTES

Raju and Krishnamurthy, Langley Research Center. Phillips, Lockheed Martin Space Operations.
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://ntrs.nasa.gov
This TP was upgraded from NASA TM-2004-212996 in October 2004.

14. ABSTRACT

A meshless local Petrov-Galerkin (MLPG) method that uses radial basis functions (RBFs) as trial functions in the study of Euler-Bernoulli beam problems is presented. RBFs, 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 they are in the conventional MLPG method. Compactly and noncompactly supported RBFs are considered. Noncompactly supported cubic RBFs are found to be preferable. Patch tests, mixed boundary value problems, and problems with complex loading conditions are considered. Results obtained from the radial basis MLPG method are either of comparable or better accuracy than those obtained when using the conventional MLPG method.

15. SUBJECT TERMS

Meshless Local Petrov-Galerkin Method (MLPG); Radial basis functions (RBFs); Moving least squares (MLS); Beam problems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	38	(301) 621-0390