NASA/TM-2002-211936

# Meshless Local Petrov-Galerkin Method for Bending Problems

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

*Ivatury S. Raju*
*Langley Research Center, Hampton, Virginia*

September 2002

## 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/TM-2002-211936

# Meshless Local Petrov-Galerkin Method for Bending Problems

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

*Ivatury S. Raju*
*Langley Research Center, Hampton, Virginia*

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

September 2002

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**

The Finite Element Method (FEM) is extensively used as an engineering analysis tool because of its versatility and flexibility. However, the method suffers from drawbacks such as discontinuous secondary variables across inter-element boundaries and the need for remeshing in large deformation problems. Therefore, researchers in recent years have begun to explore the possibility of developing new and innovative analysis tools that do not have these drawbacks, and yet have all the advantages of the FEM.

Recent literature shows extensive research work on meshless or element-free methods. One such method is the Meshless Local Petrov-Galerkin (MLPG) method. This method is based on a local weak form of the governing differential equation and allows for a choice of trial and test functions from different spaces. By a judicious choice of the test functions, the integrations involved in the weak form can be restricted to regular domains. The MLPG method is currently implemented for 2-D potential and elasticity problems.

In this report, the method is further developed for bending of beams – $C^1$ problems. A generalized moving least squares (GMLS) interpolation is used to construct the trial functions, and spline and power weight functions are used as the test functions. The MLPG method for beam problems is applied to problems for which exact solutions are available to evaluate its effectiveness. Additionally, a Petrov-Galerkin implementation of the method is shown to greatly reduce computational time and effort, thus demonstrating that this Petrov-Galerkin approach is preferable over the previously developed Galerkin approach. The MLPG method for beam problems yields continuous

secondary variables without the need for elaborate post-processing techniques, and the accuracy of the method is demonstrated for problems with load discontinuities and continuous beam problems.

This report describes the work that was performed in partial satisfaction of the requirements met by Dawn R. Phillips for the degree of Master of Science from the George Washington University Joint Institute for Advancement of Flight Sciences.

## Table of Contents

Abstract ..... iii

Table of Contents ..... v

List of Figures ..... viii

List of Abbreviations and Symbols ..... xi

List of Tables ..... xvi

Chapter

1. Introduction ..... 1
    - 1.1. Motivation ..... 1
    - 1.2. Background ..... 2
    - 1.3. Objective ..... 5
    - 1.4. Scope ..... 5
    - 1.5. Overview ..... 6
2. MLPG for $C^0$ Problems ..... 8
    - 2.1. Weak Form for 1-D $C^0$ Problems ..... 8
    - 2.2. Moving Least Squares Interpolation ..... 19
    - 2.3. System Equations ..... 27
    - 2.4. Penalty Method for Enforcing Essential Boundary Conditions ..... 31
        - 2.4.1. Penalty Method in the FEM ..... 32
        - 2.4.2. Penalty Method in the MLPG Method ..... 33
    - 2.5. Numerical Examples ..... 36
    - 2.6. Concluding Remarks ..... 47
3. MLPG for $C^1$ Problems ..... 49

3.1. Beam Theory ..... 49

3.2. Local Weak Form for Euler-Bernoulli Beam Problems ..... 52

3.2.1. Boundary Terms in the LWF ..... 59

3.3. Generalized Moving Least Squares Interpolation ..... 70

3.4. Test Functions Used ..... 77

3.5. Development of the MLPG Equations ..... 80

3.6. The Petrov-Galerkin Formulation ..... 85

3.7. Concluding Remarks ..... 87

4. Numerical Examples ..... 89

4.1. Patch Tests ..... 90

4.2. Local Coordinate Approach ..... 94

4.3. Patch Tests Revisited ..... 98

4.4. Problem Parameters ..... 99

4.5. Mixed Boundary Value Problems ..... 101

4.5.1. Cantilever beam with concentrated moment at the free end ..... 102

4.5.2. Cantilever beam with tip load ..... 102

4.5.3. Simply supported beam subjected to uniformly distributed load ..... 103

4.5.4. Simply supported beam subjected to a central concentrated load ..... 106

4.6. Continuous Beams ..... 111

5. Concluding Remarks ..... 115

5.1. Accurate Solutions by the MLPG Method ..... 115

5.2. Continuous Secondary Variables ..... 116

5.3. Local Coordinate Approach ..... 116

5.4. The Petrov-Galerkin Approach ..... 117

5.5. Problem Parameters ..... 118

5.6. Contributions of this Research ..... 118

5.7. Suggestions for Future Work ..... 120

References ..... 121

Appendix A: Computation of Derivatives of Shape Functions ..... 124

Appendix B: Conditioning of Matrices ..... 142

## List of Figures


Figure Page

1.2.1 Modeling in the FEM and MM ..... 3

2.1.1 Finite element trial (shape) functions at node $j$ ..... 11

2.1.2 Trial and test functions and domain of integration ..... 13

2.1.3 Comparison of the domains of the trial and test functions ..... 14

2.1.4 Test functions at various nodes in a 17-node model ..... 17

2.2.1 Moving least squares (MLS) interpolation ..... 20

2.2.2 Comparison of extents of FE and MLPG trial functions ..... 20

2.2.3 A 9-node model of a bar ..... 25

2.2.4 Typical shape functions and their derivatives ..... 26

2.3.1 Test function (of Eq. (2.3.7) with $\beta = 4$) at node 5 of a 9-node model of a bar ..... 29

2.4.1 Test functions near global boundary ..... 35

2.5.1 A 17-node model of a bar of length $l$ ..... 36

2.5.2 Uniform bar of length $l$ with end load of magnitude $\tilde{q}$ ..... 38

2.5.3 $\Omega_s$ definitions for various nodes ..... 40

2.5.4 A 15-node model with unequally spaced nodes ..... 41

2.5.5 Rectangular cooling fin ..... 42

2.5.6 Bar model of rectangular cooling fin ..... 42

2.5.7 Comparison of the MLPG and exact secondary variable distributions for a 17-node model with uniform nodal spacing ..... 46

2.5.8 Comparison of the MLPG and exact secondary variable distributions for a 15-node model with non-uniform nodal spacing ..... 47

2.5.9 Comparison of the MLPG secondary variable distribution before and after model refinement ..... 47

3.1.1 Euler-Bernoulli beam ..... 50

3.1.2 Beam segment subjected to a moment ..... 52

3.2.1 Comparison of the domains of the trial and test functions ..... 56

3.2.2 Test functions at various nodes in a 17-node model ..... 60

3.3.1 Comparison of FEM shape functions for $C^0$ and $C^1$ Problems ..... 71

3.3.2 A 17-node model of a beam of length $4l$ ..... 76

3.3.3 Typical shape functions and their derivatives ..... 76

3.4.1 Typical test functions and their derivatives ..... 80

4.0.1 A 17-node model of the beam ..... 89

4.1.1 Patch tests for beam problems ..... 91

4.2.1 Local coordinate definitions ..... 95

4.4.1 Local sub-domain, $\Omega_s$, definitions for various nodes ..... 100

4.4.2 Rigid body rotation – Comparison of results for different extents of trial functions ..... 101

4.5.1 Cantilever beam with concentrated moment at the free end ..... 102

4.5.2 Cantilever beam with tip load ..... 103

4.5.3 Simply supported beam subjected to a uniformly distributed load ..... 103

4.5.4 A 19-node model with unequally spaced nodes ..... 105

4.5.5 MLPG and exact solutions for a simply supported beam subjected to a uniformly distributed load ..... 105-106

4.5.6 Simply supported beam subjected to a central concentrated load ..... 107

4.5.7 Symmetric representation of a simply supported beam subjected to a central concentrated load ..... 108

4.5.8 MLPG and exact solutions for a simply supported beam with a central concentrated load ..... 110

4.6.1 Continuous beam subjected to a uniformly distributed load ..... 111

4.6.2 MLPG and exact solutions for primary and secondary variables of a continuous beam subjected to a uniformly distributed load ..... 113-114

## List of Abbreviations and Symbols

| | |
|---|---|
| EBC | essential boundary condition |
| FEM | Finite Element Method |
| LWF | local weak form |
| MLPG | Meshless Local Petrov-Galerkin Method |
| MLS | moving least squares |
| MM | Meshless Methods |
| NBC | natural boundary condition |
| 1-D | one dimension |
| 2-D | two dimensions |
| 3-D | three dimensions |
| $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B}_w$, $\mathbf{B}_t$ | matrices of MLS interpolation involving basis functions and weight functions |
| $\mathbf{D}$ | FEM displacement vector |
| $\tilde{D}$ | prescribed FEM displacement |
| $E$ | Young's Modulus |
| $\mathbf{I}$ | identity matrix |
| $I_{yy}$, $I$ | moment of inertia |
| $\mathbf{K}$ | FEM stiffness matrix |
| $L$, $l$ | length |
| $M$ | moment in beam problems |
| $\tilde{M}$ | prescribed moment in beam problems |
| $N$ | FEM trial (shape) functions, or, number of nodes in beam problems |
| $P$ | applied concentrated load |

| | |
|---|---|
| $\mathbf{R}$ | FEM force vector |
| $V$ | shear in beam problems |
| $\tilde{V}$ | prescribed shear in beam problems |
| $\|E\|_1$ | error norm of $\{\mathbf{r}\}$ |
| $\mathbf{K}^{(\text{node})}$ | system "stiffness" matrix |
| $\mathbf{K}^{(\text{bdry})}$ | boundary component of system "stiffness" matrix |
| $L^2, J, H^h$ | weighted discrete error norms |
| $R_o$ | user-defined parameter that controls extent of test functions |
| $R_j$ | user-defined parameter that controls extent of trial functions |
| $cond[\mathbf{D}]$ | conditioning number of matrix $[\mathbf{D}]$ |
| $d_j$ | Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_j$; $d_j = \| \mathbf{x} - \mathbf{x}_j \|$ |
| $\mathbf{d}$ | rigid body displacement vector |
| $\hat{\mathbf{d}}$ | vector of fictitious nodal values |
| $\|e_\Theta\|$, $\|e_H\|$, $\|E_w\|_2$, $\|E_M\|_2$ | error norms |
| $f(x)$ | loading |
| $\mathbf{f}^{(\text{node})}$ | system force vector |
| $\mathbf{f}^{(\text{bdry})}$ | boundary component of system force vector |
| $m$ | order of basis function |
| $n$ | number of nodes in the domain of definition of a node |
| $n_x$ | direction cosine of the unit outward drawn normal to $\Omega$ |
| $\mathbf{p}$ | basis function |
| $\mathbf{p}_x$ | derivative of basis function |

| | |
|---|---|
| $q$ | secondary variable in 1-D $C^0$ problems, or, applied distributed load in beam problems |
| $\tilde{q}$ | prescribed secondary variable in 1-D $C^0$ problems |
| $\hat{\mathbf{s}}$ | nodal displacement vector in beam problems (contains $\{\hat{\mathbf{w}}\}$ and $\{\hat{\mathbf{t}}\}$) |
| $\hat{\mathbf{t}}$ | vector of fictitious nodal values for slope |
| $u$ | displacement in 1-D $C^0$ problems |
| $\tilde{u}$ | prescribed displacement in 1-D $C^0$ problems |
| $\hat{u}_j$ | fictitious nodal values of displacement at node $j$ |
| $\hat{\mathbf{u}}$ | vector of fictitious nodal values |
| $v$, $v_i$ | weight function, test function |
| $w$ | deflection in beam problems |
| $\tilde{w}$ | prescribed deflection in beam problems |
| $\hat{w}_j$ | fictitious nodal values of deflection at node $j$ |
| $\hat{\mathbf{w}}$ | vector of fictitious nodal values for deflection |
| $\mathbf{x}$ | vector of spatial coordinates |
| $\bar{\mathbf{x}}$ | local neighborhood of $\mathbf{x}$ |
| $\Delta x$ | distance between nodes |
| $\Gamma$ | global boundary (of $\Omega$) |
| $\Gamma_M$ | boundary where $M$ is prescribed |
| $\Gamma_V$ | boundary where $V$ is prescribed |
| $\Gamma_q$ | boundary where $q$ is prescribed |
| $\Gamma_s$ | local boundary (of $\Omega_s$) |
| $\Gamma_{sM}$ | local boundary where $M$ prescribed |

| | |
|---|---|
| $\Gamma_{sV}$ | local boundary where $V$ prescribed |
| $\Gamma_{sq}$ | local boundary where $q$ prescribed |
| $\Gamma_{su}$ | local boundary where $u$ prescribed |
| $\Gamma_{sw}$ | local boundary where $w$ prescribed |
| $\Gamma_{s\theta}$ | local boundary where $\theta$ prescribed |
| $\Gamma_u$ | boundary where $u$ is prescribed |
| $\Gamma_w$ | boundary where $w$ is prescribed |
| $\Gamma_\theta$ | boundary where $\theta$ is prescribed |
| $\Omega$ | global domain of a problem |
| $\Omega_d$ | domain of integration (Galerkin formulations) |
| $\Omega_s$ | local sub-domain |
| $\alpha_c$ | penalty parameter for enforcing continuous beam boundary conditions |
| $\alpha_u$ | penalty parameter for enforcing EBCs in 1-D $C^0$ problems |
| $\alpha_w$ | penalty parameter for enforcing deflection in beam problems |
| $\alpha_\theta$ | penalty parameter for enforcing slope in beam problems |
| $\delta_{jk}$ | Kronecker delta |
| $\theta$ | slope in beam problems |
| $\tilde{\theta}$ | prescribed slope in beam problems |
| $\hat{\theta}_j$ | fictitious nodal values of slope at node $j$ |
| $\lambda_j$ | weight functions |
| $\mu_i^{(u)}$ | arbitrary constants for displacement in 1-D $C^0$ problems |

| | |
|---|---|
| $\mu_i^{(w)}$ | arbitrary constants for deflection in beam problems |
| $\mu_i^{(\theta)}$ | arbitrary constants for slope in beam problems |
| $\phi_j$ | shape functions in 1-D $C^0$ problems |
| $\chi_i^{(u)}$ | components of test functions for displacements in 1-D $C^0$ problems |
| $\chi_i^{(w)}$ | components of test functions for deflections in beam problems |
| $\chi_i^{(\theta)}$ | components of test functions for slopes in beam problems |
| $\psi_j^{(w)}$ | shape functions in beam problems |
| $\psi_j^{(\theta)}$ | shape functions in beam problems |

## List of Tables


Table Page

2.5.1 Comparison of the MLPG solution with the exact solution ..... 44

2.5.2 Error norm $\|e\|$ for the 5-, 9-, 17-, and 33-node models ..... 45

4.1.1 Error norm $\|E\|_1$ of the residuals for six models and for two basis functions ..... 93

4.2.1 Comparison of the condition numbers of the **[A]** matrices at various locations on the beam using global and local coordinate methods ..... 97

4.2.2 Error norm $\|E\|_1$ of the residuals computed with the local coordinate approach ..... 97

4.5.1 Error norm $\|E\|_2$ for a simply supported beam subjected to a uniformly distributed load with cubic basis used in the MLPG method. (Trial function using Eq. (3.3.19) with $\alpha$=3 and test function using Eq. (3.4.6) with $\beta$=4.) ..... 104

4.5.2 MLPG values of deflection and slope for models with various nodal arrangements ..... 109

# Chapter 1: Introduction

Aerospace structures are very complex in construction. Structural elements used are usually built up from doubly-curved shells and stiffeners made up of metallic, composite, or sandwich materials. Further, aerospace structures are expected to be durable and damage tolerant, are required to have minimum weight, and are expected to provide superior performance. These structures are also expected to be in service over a wide range of operating conditions and in extreme environments. Satisfying these requirements while maintaining cost effectiveness is a complicated but possible task. The only efficient way to obtain such a system is through very accurate and high fidelity analyses and validation of the resultant design configurations through innovative test techniques.

## 1.1 Motivation

The Finite Element Method, because of its versatility and flexibility, is extensively used as an engineering analysis tool in civil, automotive, marine, off-shore, and aerospace industries. However, the FEM suffers from drawbacks such as discontinuous secondary variables (such as stresses) across inter-element boundaries and the need for remeshing in large deformation problems. As stresses are discontinuous across inter-element boundaries, post-processing techniques are required to achieve smooth stress distributions. Four commonly used smoothing techniques are (Cook *et al.*, 2002) the element smoothing technique, the nodal averaging method, the global averaging method, and patch recovery. These methods involve post-processing the FE output to obtain smooth secondary variables.

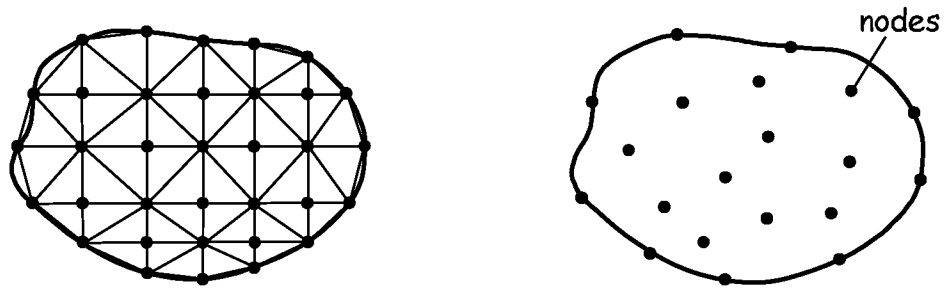
The second disadvantage of the FEM is in geometric or material nonlinear analysis. In nonlinear analysis, severe mesh distortions can occur. These mesh distortions lead to poorly shaped or ill-shaped elements. These ill-shaped elements perform poorly and hence remeshing of the deformed analysis region is needed. The remeshing and the associated interpolation of the current nonlinear solution onto the new mesh is a tedious process. Any method that avoids ill-shaped elements, that provides smooth secondary variable distributions, and that retains the advantages of the FEM is very attractive. Meshless Methods (MM) appear to show promise in these directions. For MM to successfully compete with the FEM, the MM need to be applicable to built-up structures. Meshless Methods so far have been applied to one- and two- dimensional $C^0$ problems. Thus the next step is to apply the MM to $C^1$ problems involving one dimension. In this report, one of the MM, the Meshless Local Petrov-Galerkin (MLPG) method is applied to beam problems.

## 1.2 Background

With the goal of eliminating the disadvantages of the FEM, researchers in recent years have begun to explore the possibility of developing new and innovative analyses tools that do not have the drawbacks, yet retain most of the advantages of the FEM.

Nayroles *et al.* (1992) developed the concept of a diffuse approximation of the finite element method. They proposed replacing the finite element interpolation function with a smooth function that is diffused and using a moving least squares formulation to arrive at the interpolation. A moving least squares (MLS) interpolation uses the local weighted least squares function to evaluate the dependent variable at a point in the

domain of the problem. Coefficients in this least squares function are found by minimizing the sum of the squares of the error between the interpolation and the value of the dependent variable at the nodes. In the FEM, Dirac delta functions are used to perform this minimization. For the current Diffuse Element Method, continuous weighting functions that vanish at a certain distance from the nodes over which they are centered are used. Two very important attributes of the Diffuse Element Method are noted by Nayroles *et al.*(1992): 1) a collection of nodes, without a mesh, and boundary conditions are all that are needed to develop the system matrices, and 2) accurate solutions are obtained for both regular and irregular nodal spacing. Figure 1.2.1 shows the ways in which domains are modeled in the FEM and MM.




**(a) FEM: nodes and elements** **(b) MM: only nodes**

**Figure 1.2.1: Modeling in the FEM and MM**

Belytschko *et al.* (1994) took the ideas of the Diffuse Element Method further and developed the Element Free Galerkin (EFG) Method. In developing their equations, they made the important observation that the coefficients in the MLS interpolation should not be regarded as constants. As a result, when evaluating the derivatives of the shape functions obtained from the MLS interpolation, two very important terms neglected by Nayroles *et al.* (1992) were included. The accuracy of the EFG method thus showed

significant improvement over the accuracy of the Diffuse Element Method. Additionally in the implementation of the EFG method, Lagrange multipliers were used to enforce essential boundary conditions (EBCs), and a "shadow" cell structure was overlaid on the domain to integrate the system matrices. The convergence rate of the EFG method depends on the choice of weight function in the interpolation, but significantly exceeds that of the finite element method. Several observations were also made about the background integration mesh of Belytschko *et al.* (1994). Because the cells are used solely for the purpose of carrying out the numerical integrations, they do not need to satisfy the requirements of finite elements, and they can be easily refined in a local region (unlike in the FEM).

Mukherjee and Mukherjee (1997) made important contributions in the imposition of essential boundary conditions in meshless methods. They recognized that MLS interpolants lack the Kronecker delta property of the usual FEM shape functions. As a result, imposition of EBCs is not straightforward. Mukherjee and Mukherjee proposed that the values of the dependent variable be replaced by fictitious nodal values to accurately satisfy the EBCs at boundary nodes. The resulting system of equations are solved for these fictitious nodal values, which are in turn used in conjunction with the nodal shape functions to arrive at the numerical solution to the problem.

While the overlaid cell structure does not have requirements as stringent as the finite element mesh, the cell structure is still a mesh that is needed for the EFG models. Therefore, one of the advantages of the EFGM is lost. Atluri and Zhu (1998) developed a truly meshless method that does not require the shadow cell structure to perform the numerical integrations. They proposed using a Local Weak Form (LWF), in which

calculations begin from the weak form in a local sub-domain. Essential boundary conditions are imposed by means of a penalty method. The Petrov-Galerkin method is used, as opposed to the Galerkin method used by previous researchers, where the trial and test functions are taken from the same space. By a suitable choice of the test function, the method can be made local. As such, no overlaying cell structure is required to perform the numerical integrations.

### 1.3 Objective

In this report, the MLPG method is first applied to $C^0$ problems to understand various features of this method. The method is further developed for 1-D $C^1$ problems involving Euler-Bernoulli beams. A Petrov-Galerkin formulation for the beam problems is presented. The formulation is applied to several beam problems for which exact solutions are available to evaluate its effectiveness. Various features of the method are studied and the performance of the method to ranges of important parameters are discussed.

### 1.4 Scope

The $C^1$ problems presented in this report are Euler-Bernoulli beams. Thus, the MLPG method is developed using the Euler-Bernoulli beam conventions. These conventions are stated as follows: 1) Euler-Bernoulli beams undergo small deformations, 2) plane sections normal to the neutral axis before deformation remain planar and normal to the neutral axis after deformation, and 3) deflection is a function of the axial

coordinate alone. A more detailed explanation of Euler-Bernoulli beam theory is presented in Chapter 3.

### 1.5 Overview

In the chapters that follow, the phrase "machine accuracy" appears several times. "Machine accuracy" means that the absolute value of the difference between the exact solution and the numerical solution is of the order of $10^{-14}$, using double precision arithmetic.

In Chapter 2, the MLPG method for $C^0$ problems, the problems that are described by a second order ordinary differential equation, are considered. In $C^0$ problems, the dependent variables are continuous, but their derivatives may not be continuous. A local weak form of the governing differential equation is developed. Approximations to the solution known as trial functions are formed using the moving least squares interpolation. The Petrov-Galerkin formulation for these $C^0$ problems is presented. A system of algebraic equations is derived by using the MLS interpolation and the Petrov-Galerkin test functions in the local weak form. Numerical examples, including patch test problems, mixed boundary value problems, and a typical heat transfer problem are worked to evaluate the effectiveness of the method.

In Chapter 3, the MLPG method for $C^1$ problems, specifically for Euler-Bernoulli beams, is presented. These problems are described by fourth order ordinary differential equations. In $C^1$ problems, the dependent variables and their first derivatives are continuous, but higher order derivatives may not be continuous. A local weak form (LWF) of the governing differential equation is developed. The moving least squares

interpolation scheme is generalized to include derivatives of the dependent variables, and is used to construct the trial functions. Test functions are chosen from a different space than the trial functions, making the method a Petrov-Galerkin method. The trial and test functions are then used in the LWF to derive a system of algebraic equations.

Numerical examples of beam problems are presented in Chapter 4. A local coordinate approach is developed, problem parameters are established, and patch tests are performed. Several mixed boundary problems are considered, and the continuity requirements for the Petrov-Galerkin test functions are established. Finally, a continuous beam problem is studied.

In Chapter 5, conclusions drawn from the report are presented and summarized. Several suggestions for future work are also made.

# Chapter 2: MLPG for $C^0$ Problems

A Meshless Local Petrov-Galerkin (MLPG) method has been developed for $C^0$ problems. The method was applied to potential problems by Atluri and Zhu (1998) and to axisymmetric problems by Raju and Chen (2001). Before $C^1$ problems can be discussed, $C^0$ problems must be understood. This chapter presents a description of the method applied to $C^0$ one-dimensional (1-D) problems.

First, a local weak form is developed from the classical weighted-residual form of the governing differential equation. A moving least squares interpolation is used to construct the approximations to the solution known as trial functions. Test functions are chosen from a different space than the trial functions, making the method a Petrov-Galerkin method. Essential boundary conditions are enforced by a penalty method similar to the penalty method employed by the FEM. A system of algebraic equations is derived by substituting the trial and test functions into the local weak form. The method is evaluated by applying it to several patch test and mixed boundary value problems. Finally, a typical example of a heat transfer problem is analyzed using the MLPG method.

## 2.1 Weak Form for 1-D $C^0$ Problems

Consider a 1-D $C^0$ problem (Reddy, 1993) governed by

$$-\frac{d}{dx}\left(b(x)\frac{du}{dx}\right)+c(x)u=f(x) \tag{2.1.1}$$

in domain $\Omega$ $(0 \le x \le l)$ with boundary $\Gamma$, where $b(x)$ and $c(x)$ are problem parameters that may be functions of the coordinate $x$, and $f(x)$ is some "loading," which may also be a function of $x$. The essential and natural boundary conditions are of the form

$$u = \tilde{u} \quad \text{on } \Gamma_u \quad , \quad q = \tilde{q} \quad \text{on } \Gamma_q \tag{2.1.2}$$

where

$$q = b\frac{du}{dx} \tag{2.1.3}$$

and $\Gamma_u$ and $\Gamma_q$ denote the boundary regions where the primary variable, $u$, and the secondary variable, $q$, are prescribed, respectively. In 1-D problems, these boundary regions are the points $x$=0 and $x$=$l$. The variables $u$ and $q$ represent different physical quantities depending on the type of problem considered. For example, in the problem of axial deformation of a bar, the primary variable $u$ is longitudinal displacement, $b$=$EA$ where $E$ is the modulus of elasticity and $A$ is the cross-sectional area, $f$ is the applied body force on the surface of the bar (such as friction, self-weight, etc.), and $b \cdot (du/dx)$, the secondary variable, is the axial force. For a heat transfer problem, $u$ is temperature, $b$ is the thermal conductivity, $f$ is heat generation, and $b \cdot (du/dx)$ is the heat flux (Reddy, 1993).

To obtain an approximate solution to Eq. (2.1.1), a weighted residual technique is employed. As an approximate solution for $u$ is sought, there is an error; that error (residual) is

$$R = -\frac{d}{dx}\left(b\frac{du}{dx}\right) + cu - f \,. \tag{2.1.4}$$

Control of the errors is affected by multiplying the residual by a weight function $v(x)$, integrating over the whole domain, and setting the integral to zero:

$$0 = \int_{\Omega} v\left[-\frac{d}{dx}\left(b\frac{du}{dx}\right) + cu - f\right] dx \,. \tag{2.1.5}$$

Equation (2.1.5) represents the classical weighted residual form of the governing differential equation. An approximate solution for $u$ is chosen such that each term in the approximate solution must be twice differentiable and satisfy all the boundary conditions (Eq. 2.1.2). While these requirements are easy to satisfy in 1-D problems, for higher dimensions, they are difficult to satisfy. Therefore, a formulation that accepts weaker requirements on $u$ is sought. The weak form of the weighted residual equation is set up by transferring the differentiation from the primary variable $u$ to the weight function $v$. This is achieved by integrating by parts in 1-D and by application of the divergence theorem in 2-D and 3-D. Integrating Eq. (2.1.5) by parts yields

$$0 = \int_{\Omega} b \frac{du}{dx}\frac{dv}{dx} dx + \int_{\Omega} cvu\, dx - \int_{\Omega} fv\, dx - \left[ vb\frac{du}{dx} \right]_{\Gamma} . \tag{2.1.6}$$

Integration by parts produces a boundary term $[v \cdot b \cdot (du/dx)]_{\Gamma}$. The prescription of the secondary variable $b \cdot (du/dx)$ on $\Gamma$ is the natural boundary condition (NBC) and is now part of the weak form. The requirements on the approximate solution have thus been weakened, i.e., $u$ must be differentiable once and must satisfy only the essential boundary conditions as the NBCs are included in the weak form. In Eq. (2.1.6), called the weak form of the governing differential equation, the chosen approximating functions for $u$ and $v$ are called the trial and test functions, respectively. (The secondary variables are identified as the coefficients of the weight functions and their derivatives in the boundary expressions of the weak form (Reddy, 1993, p. 31).) This weak form is the starting point of the Finite Element Method (FEM).

In the FEM, $u$ is chosen as a piecewise linear function as shown in Figure 2.1.1.

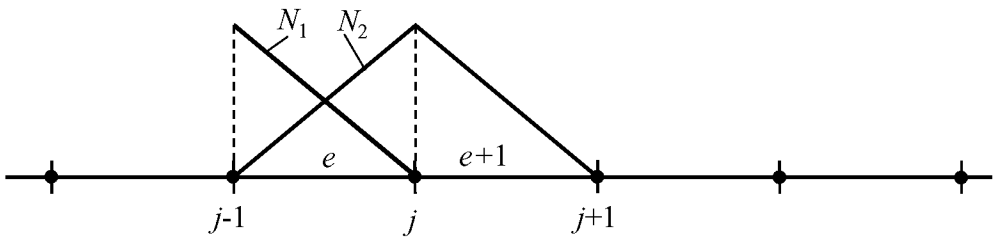



**Figure 2.1.1: Finite element trial (shape) functions at node *j***

The trial functions for element $e$ are chosen as:

$$u^{(e)} = N_1 u_{j-1}^{(e)} + N_2 u_j^{(e)} \tag{2.1.7}$$

where $N_1$ and $N_2$ are shape functions of the $e^{\text{th}}$ element and $u = \sum_{e=1}^{N_{el}} u^{(e)}$ where $N_{el}$ are the number of elements in the model. The test function $v$ is chosen as the variation of $u$:

$$v^{(e)} = \delta u^{(e)} = N_1 \delta u_{j-1}^{(e)} + N_2 \delta u_j^{(e)} . \tag{2.1.8}$$

This choice of $v^{(e)}$ as $\delta u^{(e)}$ makes the FEM a Galerkin method. These choices yield several advantages to the FEM: (1) because the trial functions are piecewise linear, the FEM has a local character, and thus the stiffness matrix is banded, (2) the choice $v=\delta u$ yields a symmetric stiffness matrix, and (3) the stiffness matrix becomes positive definite after the imposition of boundary conditions because the first integrand in Eq. (2.1.6) represents an "energy" quantity.

The secondary variables are usually the quantities sought in an analysis. For the $C^0$ problems considered here, the secondary variable is

$$q = b\frac{du}{dx} . \tag{2.1.9}$$

The secondary variable $q_j$ for the trial function $u_j$ (see Figure 2.1.1) is the slope at node $j$. The slopes at node $j$ for elements $e$ and $e$+1 are obviously unequal. In general, all the secondary variables in the FEM are discontinuous across element boundaries because of the piecewise nature of the approximation for the shape functions. Post processing techniques are required to achieve smooth distributions for the secondary variables. This is considered one of the disadvantages of the FEM.

To overcome the discontinuity problem of the FEM, a diffused element formulation was proposed by Nayroles *et al.* (1992). Later utilizing these concepts, Belytschko *et al.* (1994) developed the Element-Free Galerkin method. In these methods no elements are present, and trial functions $u$ are formed by passing a smooth function through fictitious nodal values (discussed in section 2.2). These trial functions are written as in the EFG methods as (Mukherjee and Mukherjee, 1997)

$$u(x) \cong \sum_{j=1}^{n} \hat{u}_j \phi_j(x), \tag{2.1.10a}$$

where $n$ is the number of nodes in the domain of definition of the trial function, $\hat{u}_j$ are fictitious nodal values of displacement, and $\phi_j(x)$ are shape functions. As the trial functions are smooth, the secondary variables are continuous at every point in the domain of the trial functions. Using the Galerkin methodology, the test functions are chosen as the variation of $u$, $v=\delta u$, and are written in the same manner as the trial functions as

$$v(x) \cong \mu_i^{(u)} \chi_i^{(u)}(x), \tag{2.1.10b}$$

where $\mu_i^{(u)}$ are arbitrary constants for displacement, and $\chi_i^{(u)}$ are components of the test functions. The details of the development of the trial and test functions are discussed in

section 2.2. The trial function for node $j$ and test function for node $i$ in the EFG method for a 1-D problem are shown in Figure 2.1.2. The domain of integration for the $i$-$j$ term in the weak form (Eq. 2.1.6) is the intersection of the trial and test functions and is shown by the shaded region, $\Omega_d$, in Figure 2.1.2.

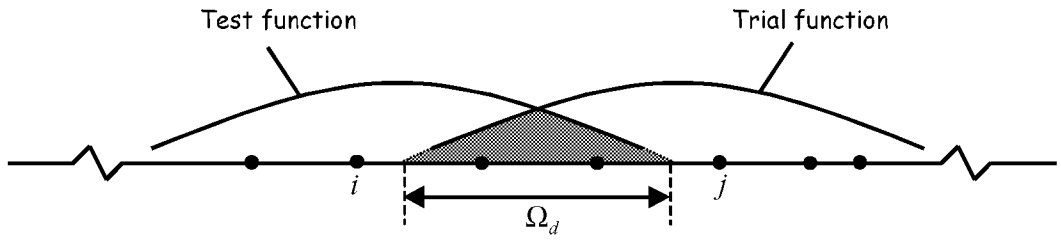



**Figure 2.1.2: Trial and test functions and domain of integration**

This domain can be large, and its shape may be difficult to determine in 2-D and 3-D problems. Because a well-defined shape is desirable for the purpose of integration, a background mesh (also called a shadow mesh) – usually rectangular meshes in 2-D (Belytschko *et al.*, 1994) – is required. As a result, while the formations of the trial and test functions do not require elements, the use of a background mesh to perform integrations negates the advantage of the EFG method and thus the EFG method is not a truly meshless method.

To develop a truly meshless method, Atluri and Zhu (1998) suggested the choice of the test function from a different space, and hence,

$$v \neq \delta u \,, \tag{2.1.11}$$

and, for example, a weight function whose nonzero values define a well-defined shape can be used. Common shapes in 2-D include circles, ellipses, and rectangles. A common test function $v_i$ for node $i$ in 1-D (in comparison with a trial function for node $j$) is presented in Figure 2.1.3. These test functions can be chosen to vanish at a certain

controllable distance, $R_o$, from node $i$. This localized property of the test functions gives the method its local character.

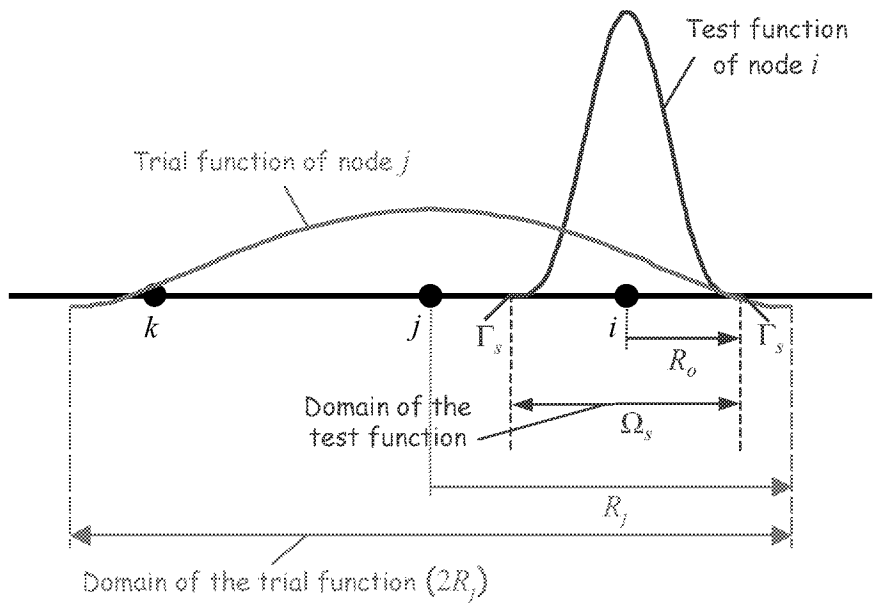



**Figure 2.1.3: Comparison of the domains of the trial and test functions**

Additionally, because the test functions have well-defined shapes and zero value outside the local sub-domain $\Omega_s$, the integrations can be restricted to $\Omega_s$, determined from the extent of the test functions (see Figure 2.1.3). This choice thus eliminates the need for a shadow mesh. The freedom to choose the test function from a different space than the trial function makes this a Petrov-Galerkin method. The proposed method is thus called a Meshless Local Petrov-Galerkin (MLPG) method (Atluri and Zhu, 1998).

The weak form is therefore written for the local sub-domain $\Omega_s$ as

$$0 = \int_{\Omega_s} b \frac{du}{dx}\frac{dv}{dx}\,dx + \int_{\Omega_s} cvu\,dx - \int_{\Omega_s} fv\,dx - \left[ vb\frac{du}{dx} \right]_{\Gamma_s} . \tag{2.1.12}$$

Additionally, the essential boundary conditions are enforced by a penalty method (Zhu and Atluri, 1998). The penalty method is discussed in section 2.4. Thus, the weak form is written as

$$0 = \int_{\Omega_s} b \frac{du}{dx}\frac{dv}{dx} dx + \int_{\Omega_s} cvu\, dx - \int_{\Omega_s} fv\, dx + \alpha_u [(u - \tilde{u})v]_{\Gamma_{su}} - \left[ vb \frac{du}{dx} \right]_{\Gamma_s} \tag{2.1.13}$$

where $\alpha_u$ is the penalty parameter to enforce the EBCs, and $\Gamma_{su}$ is the boundary where $u$ is prescribed on the local boundary ($\Gamma_s \cap \Gamma_u$). Recalling that $q = b \cdot (du/dx)$, Eq. (2.1.13) is written as

$$0 = \int_{\Omega_s} b \frac{du}{dx}\frac{dv}{dx} dx + \int_{\Omega_s} cvu\, dx - \int_{\Omega_s} fv\, dx + \alpha_u [(u - \tilde{u})v]_{\Gamma_{su}} - [vq]_{\Gamma_s} . \tag{2.1.14}$$

Recognizing that the local boundary $\Gamma_s$ could intersect the global boundary $\Gamma$, $\Gamma_s$ is broken into subsets that cover every possibility of boundary condition prescription:

$$\Gamma_s \cap \Gamma_u \quad , \quad \Gamma_s \cap \Gamma_q \quad , \quad \Gamma_s \ (\Omega_s \text{ completely within interior of } \Omega) . \tag{2.1.15}$$

For example, $\Gamma_s \cap \Gamma_u$ means the intersection of $\Gamma_s$ and $\Gamma_u$. Equation (2.1.14) then becomes

$$\begin{aligned} 0 = & \int_{\Omega_s} b \frac{du}{dx}\frac{dv}{dx} dx + \int_{\Omega_s} cvu\, dx - \int_{\Omega_s} fv\, dx + \alpha_u [(u - \tilde{u})v]_{\Gamma_{su}} \\ & - [vq]_{\Gamma_s \cap \Gamma_u} - [v\tilde{q}]_{\Gamma_s \cap \Gamma_q} - [vq]_{\Gamma_s} \quad . \end{aligned} \tag{2.1.16}$$

As mentioned previously, the test function, $v$, can be chosen to vanish on $\Gamma_s$ (see Figure 2.1.3). The term $[vq]_{\Gamma_s}$ is therefore evaluated as zero, and Eq. (2.1.16) is reduced to the local weak form (LWF) for the MLPG method:

$$0 = \int_{\Omega_s} b \frac{du}{dx}\frac{dv}{dx}\,dx + \int_{\Omega_s} cvu\,dx - \int_{\Omega_s} fv\,dx + \alpha_u\left[(u-\tilde{u})v\right]_{\Gamma_{su}} - \left[vq\right]_{\Gamma_{su}} - \left[v\tilde{q}\right]_{\Gamma_{sq}} \tag{2.1.17}$$

where $\Gamma_{su}$ represents $\Gamma_s \cap \Gamma_u$ and $\Gamma_{sq}$ represents $\Gamma_s \cap \Gamma_q$. The weak form of Eq. (2.1.17) is local because the integrations are performed over the local sub-domain $\Omega_s$. If the trial and test functions of Eq. (2.1.17) are chosen from the same space via a Galerkin method, evaluation of the terms of Eq. (2.1.17) yields symmetric stiffness matrices. Thus the weak form could be called a local symmetric weak form. (This is the case in the study of beam problems by Atluri *et al.* (1999).) In this report, a Petrov-Galerkin method is used. The resulting stiffness matrices are not symmetric, and thus the term "symmetric" is omitted from "local symmetric weak form". Substitution of the trial and test functions into Eq. (2.1.17) yields a system of equations of the form

$$\mathbf{K}^{(\text{node})}\hat{\mathbf{u}} + \mathbf{K}^{(\text{bdry})}\hat{\mathbf{u}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = 0 \tag{2.1.18}$$

where the superscript "bdry" denotes boundary, and $\hat{\mathbf{u}}$ are the fictitious nodal values $\hat{u}_j$ (see Eq. 2.1.10a). The formation of the system of equations is presented in detail in section 2.3.

Consider now the last two terms of the LWF,

$$\left[vq\right]_{\Gamma_{su}} \quad \text{and} \quad \left[v\tilde{q}\right]_{\Gamma_{sq}}. \tag{2.1.19}$$

These terms need to be evaluated at the boundary points. The details of these evaluations are explained with the aid of a 1-D domain modeled with 17 equally spaced nodes as shown in Figure 2.1.4. The nodal spacing in this model is $\Delta x = l/16$. The primary variable, $u$, is assumed to be prescribed at node 1, and the secondary variable, $q$, at node

17. In Figure 2.1.4, the test functions are shown at various nodes in the model. These test functions are assumed to have an $(R_o / l)$ of $2\Delta x$.

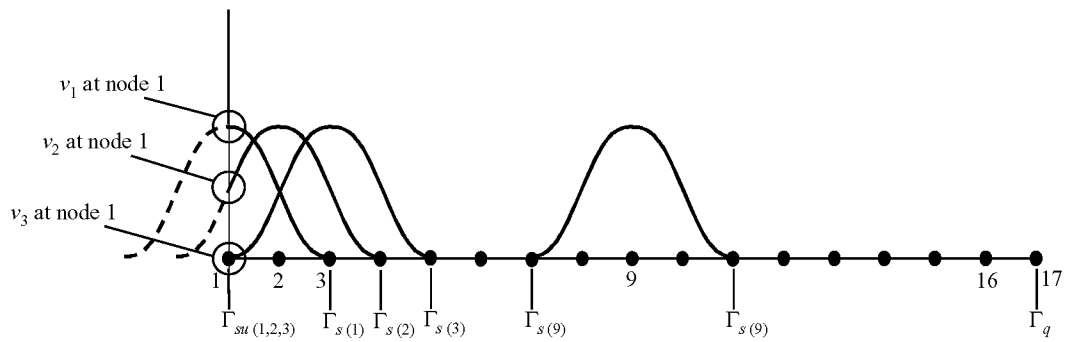



**Figure 2.1.4: Test functions at various nodes in a 17-node model**

Consider the term $[vq]_{\Gamma_{su}}$. This term must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_u$. In the model of Figure 2.1.4, there are three such nodes, nodes 1, 2, and 3. The key to the contribution of each of nodes 1, 2, and 3 to the term $[vq]_{\Gamma_{su}}$ lies in the values of $v_1$, $v_2$, and $v_3$ at node 1, where $x = 0$. First consider node 3:

$$v_3 = 0 \text{ at node 1,} \tag{2.1.20}$$

and therefore

$$[v_3 q]_{\Gamma_{su}} = 0. \tag{2.1.21}$$

Now consider node 1:

$$v_1 = 1 \text{ at node 1,} \tag{2.1.22}$$

and, utilizing Eqs. (2.1.9 and 2.1.10a),

$$[v_1 q]_{\Gamma_{su}} = [q]_{\Gamma_{su}} = \left[ b\frac{du}{dx} \right]_{\Gamma_{su}} = b\left[ \frac{d\phi_1}{dx} \quad \frac{d\phi_2}{dx} \quad \cdots \quad \frac{d\phi_n}{dx} \right]_{x=0} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix} . \tag{2.1.23}$$

Finally, consider node 2:

$$0 < v_2 < 1 \text{ at node 1,} \tag{2.1.24}$$

and therefore

$$[v_2 q]_{\Gamma_{su}} = b\begin{bmatrix} \dfrac{d\phi_1}{dx} & \dfrac{d\phi_2}{dx} & \cdots & \dfrac{d\phi_n}{dx} \end{bmatrix}_{x=0} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix} v_2\big|_{x=0} \quad . \tag{2.1.25}$$

Note that the terms $b[(d\phi_1/dx)\ \ (d\phi_2/dx)\ \ \ldots\ \ (d\phi_n/dx)]_{\Gamma_{su}}$ in Eqs. (2.1.23 and 2.1.25) are evaluated at node 1 and contribute to the $\mathbf{K}^{(\text{bdry})}$ of Eq. (2.1.18) (see Eq. 2.3.11b). The contribution of node 2 to the term $[vq]_{\Gamma_{su}}$, and ultimately to $\mathbf{K}^{(\text{bdry})}$, is of extreme importance and cannot be neglected.

Now consider the term $[v\tilde{q}]_{\Gamma_{sq}}$. This term contributes to the $\mathbf{f}^{(\text{bdry})}$ of Eq. (2.1.18) and must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_q$. For a node whose $v = 0$ at node 17,

$$[v\tilde{q}]_{\Gamma_{sq}} = 0 \,. \tag{2.1.26}$$

For a node whose $v \neq 0$ at node 17, $[v\tilde{q}]_{\Gamma_{sq}}$ is not evaluated as zero unless the prescribed secondary variable is zero. The contribution of such nodes to the term $[v\tilde{q}]_{\Gamma_{sq}}$, and ultimately to the $\mathbf{f}^{(\text{bdry})}$, is of extreme importance and cannot be neglected.

A proper understanding of how the terms of Eq. (2.1.19) are calculated provides users of the MLPG method with considerable freedom in choices of nodal spacing and sizes of test functions. For the case presented in Figure 2.1.4 of a model with equally spaced nodes, a choice of a smaller $(R_o / l)$ for nodes 2 and $N$-1 (for example, here

$(R_o / l) = \Delta x$ for nodes 2 and 16) ensures that $[vq]_{\Gamma_{su}} = 0$ and thus may be preferable. However, note that nodes need not be equally spaced. Likewise, the size of $\Omega_s$ for each $v_i$ need not be uniform. When this is the case, a simple choice of a smaller $(R_o / l)$ for nodes 2 and $N$-1 may not ensure that all the terms of Eq. (2.1.19) are identically zero for additional nodes near the boundaries. In other words, users of the MLPG algorithm cannot assume that a simple reassignment of $(R_o / l)$ will account for the terms of Eq. (2.1.19) as in the example above. In order to exploit the full usefulness of the method, the terms of Eq. (2.1.19) must be evaluated.

## 2.2 Moving Least Squares Interpolation

Several interpolation schemes are available for constructing trial functions at randomly located nodes. The Moving Least Squares (MLS) approximation is one such scheme that boasts high accuracy and ease of extension to multi-dimensional problems (Nayroles *et al.*, 1993, Belytschko *et al.*, 1994, Atluri and Zhu, 1998, Raju and Chen, 2001).

An 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 smooth function and the nodal values is a minimum (see Figure 2.2.1). The MLS interpolations are used to form the trial functions, $u$, in the current implementation of the MLPG method. The trial functions are assumed to be smooth and are nonzero over a controllable distance $R_j$ from node $j$. This distance $R_j$ is usually chosen to extend over a much larger extent than the FE shape functions (see Figure 2.2.2). The extent of the trial functions can be denoted by $\Omega_h$. An MLS approximation

can then be made for $u^h$, the value of $u$ in domain $\Omega_h$. The value for $u^h$ is zero outside of the domain $\Omega_h$.

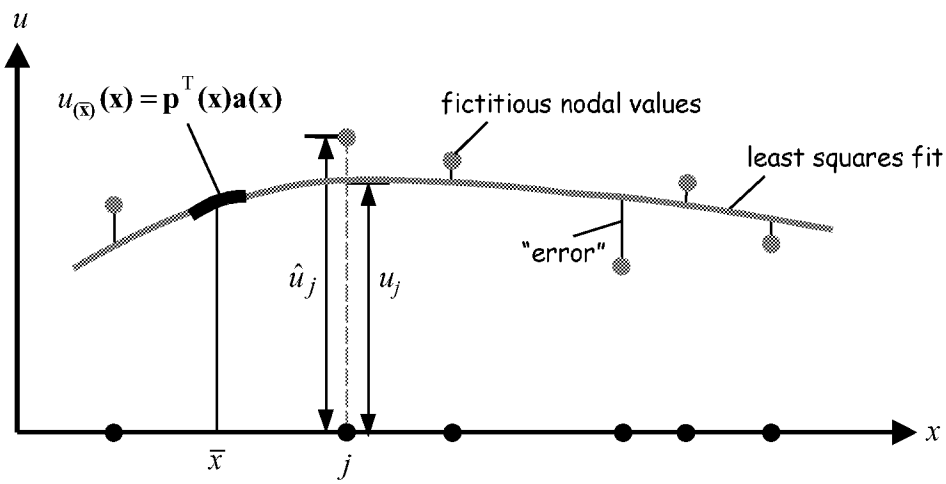



**Figure 2.2.1: Moving least squares (MLS) interpolation**

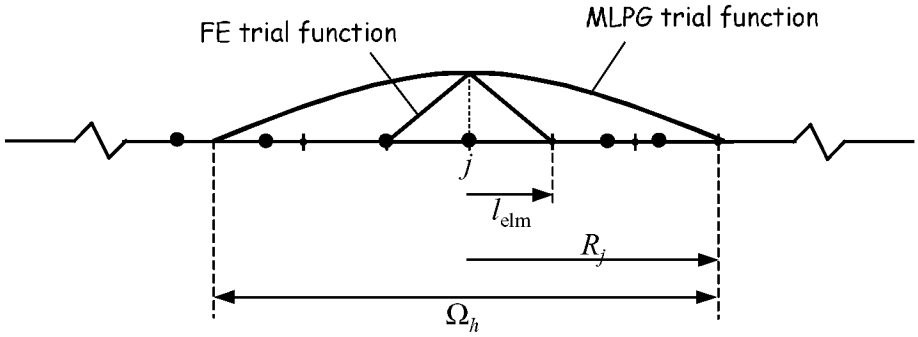



**Figure 2.2.2: Comparison of extents of FE and MLPG trial functions**

The MLS approximation for $u$ in the global domain $\Omega$ may therefore be written as the MLS approximation for $u^h$ in $\Omega_h$ as

$$u(\mathbf{x}) \cong u^h(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{2.2.1}$$

where

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = [p_1(\mathbf{x}), \quad p_2(\mathbf{x}), \quad \ldots, \quad p_m(\mathbf{x})] \tag{2.2.2}$$

is a complete monomial $m^{\text{th}}$ order basis function, and

$$\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), \quad a_2(\mathbf{x}), \quad \ldots, \quad a_m(\mathbf{x})]^{\mathrm{T}} \tag{2.2.3}$$

is a vector of undetermined coefficients. Because the coefficients $\mathbf{a}(\mathbf{x})$ may be functions of the spatial coordinates,

$$\mathbf{x} = [x, \quad y, \quad z]^{\mathrm{T}}, \tag{2.2.4}$$

the values of $\mathbf{a}(\mathbf{x})$ can vary with the position of $x$, $y$, and $z$ in $\Omega$. The global MLS approximation is therefore constructed by superposing local MLS approximations in a local neighborhood, $\overline{\mathbf{x}}$ of $\mathbf{x}$, where $\overline{\mathbf{x}} = \mathbf{x} - \mathbf{x}_j$. The local MLS interpolation is then written as

$$u(\mathbf{x}) \cong u_{\overline{\mathbf{x}}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\overline{\mathbf{x}}). \tag{2.2.5}$$

where $\mathbf{p}(\mathbf{x})$ is the basis function, and $\mathbf{a}(\overline{\mathbf{x}})$ and $u_{\overline{\mathbf{x}}}(\mathbf{x})$ are the vector of undetermined coefficients and the value of $u^h(\mathbf{x})$ in the local neighborhood $\overline{\mathbf{x}}$, respectively. Examples of basis functions for 1-D problems include

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = [1, \quad x], \text{ linear}, m = 2 \text{ and} \tag{2.2.6a}$$

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \left[1, \quad x, \quad x^2\right], \text{ quadratic}, m = 3. \tag{2.2.6b}$$

For 2-D problems, basis functions are obtained from Pascal's triangle (Cook *et al.*, 2002, Zienkiewicz and Taylor, 1989) as

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = [1, \quad x, \quad y], \text{ linear}, m = 3 \text{ and} \tag{2.2.7a}$$

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \left[1, \quad x, \quad y, \quad x^2, \quad xy, \quad y^2\right], \text{ quadratic}, m = 6. \tag{2.2.7b}$$

For 3-D problems, basis functions are obtained from Pascal's tetrahedron as

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = [1, \quad x, \quad y, \quad z], \text{ linear}, m = 4 \text{ and} \tag{2.2.8a}$$

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \left[1, \quad x, \quad y, \quad z, \quad x^2, \quad y^2, \quad z^2, \quad xy, \quad yz, \quad zx\right], \quad \text{quadratic}, m = 10 \ . \tag{2.2.8b}$$

The values of the coefficients $\mathbf{a}(\overline{\mathbf{x}})$ in Eq. (2.2.5) are found by minimizing a weighted discrete $L^2$ error norm defined as (Nayroles *et al.*, 1992)

$$\begin{aligned} J(\overline{\mathbf{x}}) &= \sum_{j=1}^{n} \lambda_j(\overline{\mathbf{x}}) \left[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_j)\mathbf{a}(\overline{\mathbf{x}}) - \hat{u}_j\right]^2 \\ &= \left[\mathbf{P} \cdot \mathbf{a}(\overline{\mathbf{x}}) - \hat{\mathbf{u}}\right]^{\mathrm{T}} \cdot \boldsymbol{\lambda} \cdot \left[\mathbf{P} \cdot \mathbf{a}(\overline{\mathbf{x}}) - \hat{\mathbf{u}}\right] \end{aligned} \tag{2.2.9}$$

where $\lambda_j(\overline{\mathbf{x}})$ are weight functions that vanish at a certain distance from $\mathbf{x}_j$, and $n$ is the number of nodes that fall within the local neighborhood $\overline{\mathbf{x}}$ of $\mathbf{x}_j$ where $\lambda_j(\overline{\mathbf{x}}) > 0$. Also in Eq. (2.2.9), $\mathbf{P}$ is an $(n,m)$ matrix, and $\boldsymbol{\lambda}$ is a diagonal $(n,n)$ matrix defined as

$$[\mathbf{P}] = \left[\ \mathbf{p}^{\mathrm{T}}(\mathbf{x}_1) \quad \mathbf{p}^{\mathrm{T}}(\mathbf{x}_2) \quad \ldots \quad \mathbf{p}^{\mathrm{T}}(\mathbf{x}_n)\ \right]^{\mathrm{T}} \ , \tag{2.2.10}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1(\overline{\mathbf{x}}) & & & \\ & \lambda_2(\overline{\mathbf{x}}) & & \\ & & \ddots & \\ & & & \lambda_n(\overline{\mathbf{x}}) \end{bmatrix} , \tag{2.2.11}$$

and

$$\hat{\mathbf{u}} = [\hat{u}_1, \quad \hat{u}_2, \quad \ldots, \quad \hat{u}_n]^{\mathrm{T}} . \tag{2.2.12}$$

Note that the values $\hat{u}_j$ in Eqs. (2.2.9) and (2.2.12) are fictitious nodal values and, in general, are not equal to the nodal values of the trial function $u^h(\mathbf{x})$ in Eq. (2.2.1) (See Figure 2.2.1).

Equation (2.2.9) can be written as

$$J(\overline{\mathbf{x}}) = \left[\mathbf{a}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}} - \hat{\mathbf{u}}^{\mathrm{T}}\right] \cdot \boldsymbol{\lambda} \cdot \left[\mathbf{P}\mathbf{a} - \hat{\mathbf{u}}\right]$$
$$= \mathbf{a}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\,\boldsymbol{\lambda}\mathbf{P}\mathbf{a} - 2\mathbf{a}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\,\boldsymbol{\lambda}\hat{\mathbf{u}} + \hat{\mathbf{u}}^{\mathrm{T}}\,\boldsymbol{\lambda}\hat{\mathbf{u}} \quad . \tag{2.2.13}$$

The error norm $L^2$ is minimized using

$$\frac{\partial J(\overline{\mathbf{x}})}{\partial a_j} = 0 \quad , \qquad j = 1, 2, \ldots n \, . \tag{2.2.14}$$

Equation (2.2.14) can be rewritten as

$$\frac{\partial J(\overline{\mathbf{x}})}{\partial \mathbf{a}^{\mathrm{T}}} = 0 \, , \tag{2.2.15a}$$

or,

$$\frac{\partial J(\overline{\mathbf{x}})}{\partial \mathbf{a}^{\mathrm{T}}} = 2\mathbf{P}^{\mathrm{T}}\,\boldsymbol{\lambda}\mathbf{P}\mathbf{a} - 2\mathbf{P}^{\mathrm{T}}\,\boldsymbol{\lambda}\hat{\mathbf{u}} = 0 \, . \tag{2.2.15b}$$

This leads to

$$\underset{(m,m)}{[\mathbf{A}]}\ \underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,n)}{[\mathbf{B}]}\ \underset{(n,1)}{\{\hat{\mathbf{u}}\}} \tag{2.2.16}$$

where

$$\underset{(m,m)}{[\mathbf{A}]} = \underset{(m,n)}{\mathbf{P}^{\mathrm{T}}}\ \underset{(n,n)}{\boldsymbol{\lambda}}\ \underset{(n,m)}{\mathbf{P}} = \underset{(m,n)}{[\mathbf{B}]}\ \underset{(n,m)}{\mathbf{P}} = \sum_{j=1}^{n} \lambda_j(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_j)\mathbf{p}^{\mathrm{T}}(\mathbf{x}_j) \tag{2.2.17}$$

and

$$\underset{(m,n)}{[\mathbf{B}]} = \underset{(m,n)}{\mathbf{P}^{\mathrm{T}}}\ \underset{(n,n)}{\boldsymbol{\lambda}} = \left[\lambda_1(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_1), \quad \lambda_2(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_2), \quad \ldots, \quad \lambda_n(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_n)\right] \, . \tag{2.2.18}$$

Solving for $\{\mathbf{a}\}$ in Eq. (2.2.16),

$$\underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,m)}{[\mathbf{A}]^{-1}}\ \underset{(m,n)}{[\mathbf{B}]}\ \underset{(n,1)}{\{\hat{\mathbf{u}}\}} \quad . \tag{2.2.19}$$

Substituting Eq. (2.2.19) into the approximation Eq. (2.2.1) yields

$$u^h(\mathbf{x}) = \underset{(1,m)}{\mathbf{p}^{\mathrm{T}}(\mathbf{x})}\underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,n)}{[\mathbf{B}]}\underset{(n,1)}{\{\hat{\mathbf{u}}\}} \quad . \tag{2.2.20}$$

The MLS trial functions can then be written as

$$u^h(\mathbf{x}) = \mathbf{\Phi}^{\mathrm{T}}(\mathbf{x})\cdot\hat{\mathbf{u}} = \sum_{j=1}^{n}\phi_j(\mathbf{x})\hat{u}_j \tag{2.2.21}$$

where

$$\mathbf{\Phi}^{\mathrm{T}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})[\mathbf{A}]^{-1}[\mathbf{B}] \quad \text{or} \quad \phi_j(\mathbf{x}) = \sum_{g=1}^{m} p_g(\mathbf{x})\left[\mathbf{A}^{-1}\mathbf{B}\right]_{gj} \quad . \tag{2.2.22}$$

In this report, $\mathbf{x} = x$ as 1-D problems are considered. The $\phi_j(x)$ are called the shape functions of the MLS approximation. Also note that $\phi_j(x) = 0$ when $\lambda_j(\bar{x}) = 0$ (See Eqs. 2.2.17 and 2.2.18). Several weight functions, $\lambda_j$, were used to construct the trial functions, $u_j$. These weight functions are power weight functions,

$$\lambda_j(\bar{x}) = \begin{cases} \left[1-\left(d_j^2/R_j^2\right)\right]^{\alpha} & \text{if} \quad 0 \le d_j \le R_j \\ 0 & \text{if} \quad d_j > R_j, \end{cases} \tag{2.2.23}$$

where $d_j$ is the Euclidean distance between $x$ and $x_j$ denoted by $d_j = \|x - x_j\|$, and $\alpha = 1$, 2, 3, and 4, a 3-term spline,

$$\lambda_j(\bar{x}) = \begin{cases} 1-3\left(\dfrac{d_j}{R_j}\right)^2 + 2\left(\dfrac{d_j}{R_j}\right)^3 & \text{if} \quad 0 \le d_j \le R_j \\ 0 & \text{if} \quad d_j > R_j, \end{cases} \tag{2.2.24}$$

and a 4-term spline,

$$\lambda_j(\bar{x}) = \begin{cases} 1 - 6\left(\dfrac{d_j}{R_j}\right)^2 + 8\left(\dfrac{d_j}{R_j}\right)^3 - 3\left(\dfrac{d_j}{R_j}\right)^4 & \text{if} \quad 0 \le d_j \le R_j \\ 0 & \text{if} \quad d_j > R_j, \end{cases} \tag{2.2.25}$$

where $R_j$ is a user-defined parameter that controls the extents of the trial functions (see Figure 2.1.3) and is termed the "support of the node $j$." (In two dimensions, the "supports of the nodal points" are usually chosen as circles of radius $R_j$.)

Consider the $N$-node model presented in Figure 2.2.3, where $N = 9$.

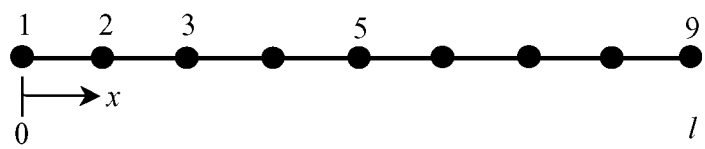
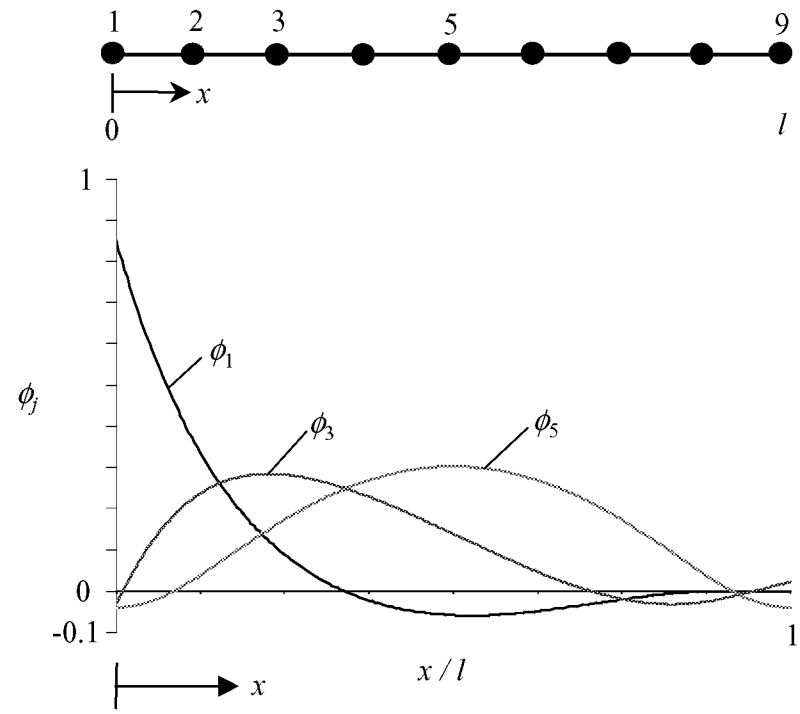
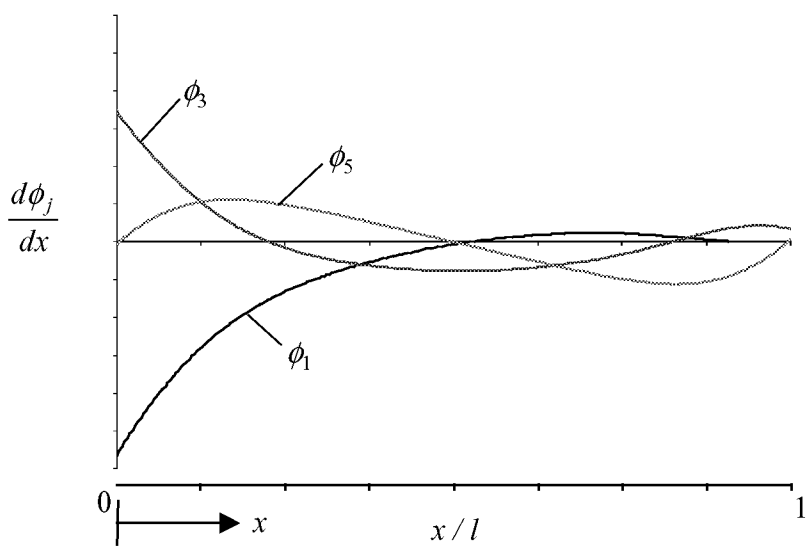



**Figure 2.2.3: A 9-node model of a bar**

Figure 2.2.4a presents typical shape functions $\phi_j$ at nodes $j = 1$, 3, and 5, evaluated using the weight function of Eq. (2.2.23) with $\alpha = 4$, and Figure 2.2.4b presents the derivative $d\phi_j/dx$ for $j = 1$, 3, and 5. These functions were evaluated with a quadratic basis function and with $(R_j/l)$ chosen as $(R_j/l) = 0.6$. Note that shape functions located equal distances on either side of the center nodes of models with uniform nodal spacing are mirror images of each other. For example, for the 9-node model presented above, $\phi_1$ and $\phi_9$, $\phi_2$ and $\phi_8$, etc. are mirror images about the center.




**(a) Shape functions, $\phi_j$**




**(b) Derivative of the shape functions, $d\phi_j/dx$**

**Figure 2.2.4: Typical shape functions and their derivatives**

## 2.3 System Equations

As mentioned previously, the approximations for $u$ are called the trial functions, and $v$ are called the test functions. The assumed trial and test functions (Eqs. 2.1.10) are substituted into the weak form of Eq. (2.1.17),

$$0 = \int_{\Omega_s} b\frac{du}{dx}\frac{dv}{dx}dx + \int_{\Omega_s} cvu\,dx - \int_{\Omega_s} fv\,dx + \alpha[(u-\tilde{u})v]_{\Gamma_{su}} - [vq]_{\Gamma_{su}} - [v\tilde{q}]_{\Gamma_{sq}}\,, \quad (2.3.1)$$

to establish the system matrices. The detailed derivation of this system of equations is presented below.

The primary variable, $u$, is approximated using Eq. (2.2.21):

$$u^h(x) = \sum_{j=1}^{n} \phi_j(x)\hat{u}_j \quad (2.3.2)$$

where $\phi_j$ are the shape functions, and $\hat{u}_j$ are the fictitious nodal values of $u$. Substitution of Eq. (2.3.2) into Eq. (2.3.1) requires the derivative of $u^h(x)$. Since $\hat{u}_j$ is not dependent on $x$, the derivative is carried out over $\phi_j(x)$ as

$$\frac{du^h}{dx} = \sum_{j=1}^{n} \frac{d\phi_j}{dx}\hat{u}_j\,. \quad (2.3.3)$$

The derivative of $\phi_j(x)$ is obtained as (Belytschko *et. al.*, 1994)

$$\phi_{j,x} = \sum_{g=1}^{m}\left[p_{g,x}(\mathbf{A}^{-1}\mathbf{B})_{gj} + p_g(\mathbf{A}^{-1}\mathbf{B}_{,x} - \mathbf{A}^{-1}\mathbf{A}_{,x}\mathbf{A}^{-1}\mathbf{B})_{gj}\right], \quad (2.3.4)$$

where

$$(\ )_{,x} \equiv \frac{d(\ )}{dx}. \quad (2.3.5)$$

The steps involved in the evaluation of the derivatives of the shape functions are presented in Appendix A. As there are $n$ trial functions used to approximate the primary variable, $n$ independent test functions ($v_i$, $i = 1, 2, \ldots n$) need to be chosen to set up the system matrix. Substitution of Eqs. (2.3.3) and (2.3.2) into Eq. (2.3.1) yields

$$\begin{aligned} 0 = & \int_{\Omega_s} b \frac{dv_i}{dx} \sum_{j=1}^{n} \frac{d\phi_j}{dx} \hat{u}_j \, dx + \int_{\Omega_s} c v_i \sum_{j=1}^{n} \phi_j \hat{u}_j \, dx - \int_{\Omega_s} f v_i dx \\ & + \left[ \alpha v_i \sum_{j=1}^{n} \phi_j \hat{u}_j \right]_{\Gamma_{su}} - [\alpha \tilde{u} v_i]_{\Gamma_{su}} - [v_i q]_{\Gamma_{su}} - [v_i \tilde{q}]_{\Gamma_{sq}} \quad (i = 1, 2, \ldots n). \end{aligned} \tag{2.3.6}$$

As discussed in section 2.1, the test functions are chosen as weight functions, similar to those presented in Eqs. (2.2.23 – 2.2.25), whose shapes are well-defined. The various test functions, $v_i$, chosen are power functions,

$$v_i(x) = \begin{cases} \left[ 1 - \left( d_i^2 / R_o^2 \right) \right]^{\beta} & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if} \quad d_i > R_o \end{cases} \tag{2.3.7}$$

with $d_i = \|x - x_i\|$ and $\beta = 1, 2, 3$, and 4, a 3-term spline,

$$v_i(x) = \begin{cases} 1 - 3\left(\frac{d_i}{R_o}\right)^2 + 2\left(\frac{d_i}{R_o}\right)^3 & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if} \quad d_i > R_o, \end{cases} \tag{2.3.8}$$

and a 4-term spline,

$$v_i(x) = \begin{cases} 1 - 6\left(\frac{d_i}{R_o}\right)^2 + 8\left(\frac{d_i}{R_o}\right)^3 - 3\left(\frac{d_i}{R_o}\right)^4 & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if} \quad d_i > R_o. \end{cases} \tag{2.3.9}$$

In Eqs. (2.3.7 – 2.3.9), $R_o$ is a user-defined parameter that controls the extents of the test functions (see Figure 2.1.3). A typical plot of the test function of Eq. (2.3.7) with $\beta = 4$ for node 5 of a 9-node model and $(R_o / l) = 2\Delta x$ of a bar is shown in Figure 2.3.1.

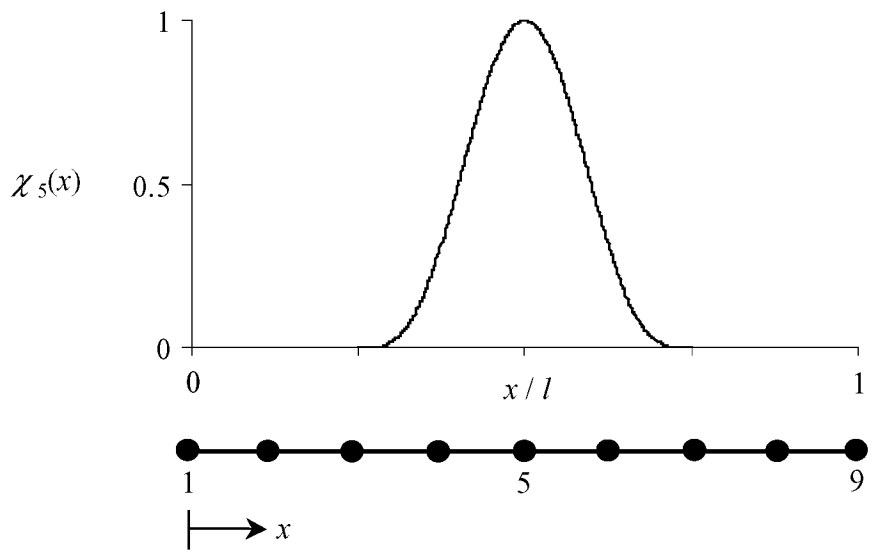



**Figure 2.3.1: Test Function (of Eq. (2.3.7) with $\beta = 4$) at node 5 of a 9-node model of a bar**

Substitution of the trial and test functions into Eq. (2.3.6) leads to the resulting system of equations

$$\mathbf{K}^{(\text{node})}\hat{\mathbf{u}} + \mathbf{K}^{(\text{bdry})}\hat{\mathbf{u}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = 0\,, \tag{2.3.10}$$

where the superscript "bdry" denotes boundary, and $\hat{\mathbf{u}}$ are the fictitious nodal values of the primary variable $u$, and

$$K_{ij}^{(\text{node})} = \int_{\Omega_s^{(i)}} b \frac{dv_i}{dx}\frac{d\phi_j}{dx}dx + \int_{\Omega_s^{(i)}} c v_i \phi_j dx \tag{2.3.11a}$$

and

$$K_{ij}^{(\text{bdry})} = \left[\alpha v_i \phi_j\right]_{\Gamma_{su}^{(i)}} - \left[v_i \frac{d\phi_j}{dx}\right]_{\Gamma_{su}^{(i)}} \tag{2.3.11b}$$

and

$$f_i^{(\text{node})} = \int_{\Omega_s^{(i)}} f v_i dx \tag{2.3.12a}$$

and

$$f_i^{(\text{bdry})} = \left[\alpha \tilde{u} v_i\right]_{\Gamma_{su}^{(i)}} + \left[v \tilde{q}\right]_{\Gamma_{sq}^{(i)}} . \tag{2.3.12b}$$

The "stiffness" matrix **K**, composed of $\mathbf{K}^{(\text{node})}$ and $\mathbf{K}^{(\text{bdry})}$, is clearly not symmetric. Unsymmetric matrices are not necessarily undesirable. Several numerical methods, for example, the boundary element method and the sub-domain collocation method, result in unsymmetric matrices. In this meshless formulation, an unsymmetric **K** is not incorrect because, unlike in the FEM, the **K** matrix in MM is not evaluated from the strain energy of the problem, but is obtained by requiring that the weighted residual is zero in an integral sense.

Numerical integration is used to integrate the system of equations as closed-form integration of the terms in Eqs. (2.3.11a and 2.3.12a) is extremely complicated. In the Gaussian quadrature integration scheme, an $n$-point Gaussian will integrate a $2n$-1 degree polynomial exactly. Equations (2.2.22, 2.3.4, 2.2.17, and 2.2.18) are repeated here for convenience:

$$\phi_j(x) = \sum_{g=1}^{m} p_g(x) \left[\mathbf{A}^{-1}\mathbf{B}\right]_{gj} \qquad \text{(order 2 if quadratic basis is used, i.e., if } p \text{ is quadratic)} \tag{2.3.13}$$

$$\phi_{j,x} = \sum_{g=1}^{m} \left[p_{g,x}(\mathbf{A}^{-1}\mathbf{B})_{gj} + p_g(\mathbf{A}^{-1}\mathbf{B}_{,x} - \mathbf{A}^{-1}\mathbf{A}_{,x}\mathbf{A}^{-1}\mathbf{B})_{gj}\right] \tag{2.3.14}$$

(order 2 if $p$ is quadratic)

$$[\mathbf{A}] = \mathbf{P}^{\mathrm{T}} \boldsymbol{\lambda} \mathbf{P} = [\mathbf{B}]\mathbf{P} = \sum_{j=1}^{n} \lambda_j(\bar{x})\mathbf{p}(x_j)\mathbf{p}^{\mathrm{T}}(x_j) \tag{2.3.15}$$

$$[\mathbf{B}] = \mathbf{P}^{\mathrm{T}} \boldsymbol{\lambda} = [\lambda_1(\bar{x})\mathbf{p}(x_1), \quad \lambda_2(\bar{x})\mathbf{p}(x_2), \quad \ldots, \quad \lambda_n(\bar{x})\mathbf{p}(x_n)] \tag{2.3.16}$$

The order of Gaussian integration required for acceptable results depends on the basis function and weight functions used. The highest order basis function considered is quadratic ($x^2$). The highest order weight function available for use as a test function and for constructing the trial functions is the weight function of Eq. (2.3.7) with $\beta = 4$, and is of the order $x^8$. Using this information in Eqs. (2.3.11a, 2.3.12a, and 2.3.13 – 2.3.16), it is found that the highest order integrand is of the order $x^{10}$. Therefore, a 6-point or higher Gaussian quadrature would successfully integrate the terms of Eqs. (2.3.11a and 2.3.12a). Numerical experimentation showed that an 8-point Gaussian quadrature consistently yielded very good results, and is hence used in the numerical implementation of the problems presented in section 2.5.

## 2.4 Penalty Method for Enforcing Essential Boundary Conditions

Imposition of essential boundary conditions (EBCs) in the EFG and MLPG methods is difficult because the shape functions from the moving least squares approximation (discussed in section 2.2) do not have the Kronecker delta property. Namely, the Moving Least Squares (MLS) shape functions do not pass through the fictitious nodal values used to fit them, and unlike in the FEM,

$$\phi_j(x_k) \neq \delta_{jk} \tag{2.4.1}$$

where $\phi_j(x_k)$ is the shape function for node $j$ evaluated at nodal point $k$, and $\delta_{jk}$ is the Kronecker delta. Because the EBCs cannot be directly enforced, a penalty method is

employed. In the sections that follow, first, the penalty method in the FEM is explained, then, the penalty method used in the MLPG method is presented.

2.4.1 Penalty Method in the FEM

In the FEM, a system set of equations is constructed to solve for unknown nodal displacements and forces.

$$[\mathbf{K}]\{\mathbf{D}\}=\{\mathbf{R}\} \tag{2.4.2}$$

where $[\mathbf{K}]$ is the assembled stiffness matrix, $\{\mathbf{D}\}$ is the nodal displacement vector, and $\{\mathbf{R}\}$ is the vector of nodal forces. EBCs are input as known displacements, and loading and natural boundary conditions (NBCs) are input as known forces. To solve the system of equations, the matrices are reordered as

$$\begin{bmatrix} \mathbf{K}_{\mathrm{UU}} & \mathbf{K}_{\mathrm{UN}} \\ \mathbf{K}_{\mathrm{NU}} & \mathbf{K}_{\mathrm{NN}} \end{bmatrix} \begin{Bmatrix} \mathbf{D}_{\mathrm{U}} \\ \mathbf{D}_{\mathrm{N}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_{\mathrm{N}} \\ \mathbf{R}_{\mathrm{U}} \end{Bmatrix} \tag{2.4.3}$$

where a subscript U denotes values that are unknown, and a subscript N denotes values that are known. The resulting equation

$$\mathbf{K}_{\mathrm{UU}}\mathbf{D}_{\mathrm{U}} + \mathbf{K}_{\mathrm{UN}}\mathbf{D}_{\mathrm{N}} = \mathbf{R}_{\mathrm{N}} \tag{2.4.4}$$

can be solved for $\mathbf{D}_{\mathrm{U}}$, after which

$$\mathbf{K}_{\mathrm{NU}}\mathbf{D}_{\mathrm{U}} + \mathbf{K}_{\mathrm{NN}}\mathbf{D}_{\mathrm{N}} = \mathbf{R}_{\mathrm{U}} \tag{2.4.5}$$

can be used to evaluate the unknown reactions, $\mathbf{R}_{\mathrm{U}}$. This process of reordering works well for small problems and for learning the FEM, but is not used in numerical implementation because the process of reordering the matrices requires large amounts of memory and run time. A penalty method is therefore employed to solve the system of equations.

The penalty method in the FEM involves choosing a penalty parameter, $\alpha$, as a very large number (usually $10^{20}$ or $10^{30}$). The diagonal stiffness term $K_{ii}$, where $i=j$ (corresponding to the known displacement, $D_i$), is multiplied by this penalty parameter. Similarly, the unknown forces $R_i$ are replaced with $\alpha K_{ii}\tilde{D}_i$ where $\tilde{D}_i$ are the EBCs. This inclusion of the EBCs with the force terms rather than with the displacement terms results in a system of equations in which the nodal displacements are the quantities sought. Consider the $i^{\text{th}}$ equation for an $M$-degree of freedom FE model,

$$K_{i1}D_1 + K_{i2}D_2 + \ldots + K_{ii}D_i + \ldots + K_{iM}D_M = R_i \,. \tag{2.4.6}$$

This equation can be modified as

$$K_{i1}D_1 + K_{i2}D_2 + \ldots + \alpha K_{ii}D_i + \ldots + K_{iM}D_M = \alpha K_{ii}\tilde{D}_i \,. \tag{2.4.7}$$

The left hand side of Eq. (2.4.7) can be approximated to $\alpha K_{ii}D_i$ as this term dominates the rest of the terms. Equation (2.4.7) can then be written as

$$\alpha K_{ii}D_i \cong \alpha K_{ii}\tilde{D}_i \tag{2.4.8a}$$

or

$$D_i \cong \tilde{D}_i \,. \tag{2.4.8b}$$

Using this procedure, the prescribed value, $\tilde{D}_i$, for $D_i$ is calculated to an accuracy of the order $(1/\alpha)$.

2.4.2 <u>Penalty Method in the MLPG Method</u>

The penalty method in the MLPG method works in a similar manner to that in the FEM. The "assembled" system of equations is

$$\mathbf{K}\hat{\mathbf{u}} = \mathbf{f} \tag{2.4.9}$$

It is desired that

$$\alpha_u K_{ii} u_i \cong \alpha_u K_{ii} \tilde{u}_i \tag{2.4.10}$$

or

$$\begin{aligned} \alpha_u K_{ii} u_i - \alpha_u K_{ii} \tilde{u}_i &\cong 0 \\ \alpha_u K_{ii} (u_i - \tilde{u}_i) &\cong 0 \\ \alpha_u (u_i - \tilde{u}_i) &\cong 0 \quad . \end{aligned} \tag{2.4.11}$$

As in the weighted residual sense, because $\alpha_u (u_i - \tilde{u}_i)$ is not equal to zero, the term is multiplied by a weight function $v(x)$ (as in section 2.1) and integrated over the boundary:

$$\int_{\Gamma_{su}} \alpha_u (u - \tilde{u}) v \, d\Gamma = 0 \, . \tag{2.4.12}$$

This term for the imposition of the EBCs is included and carried throughout the development of the LWF of the governing equation.

In two- and three-dimensional problems, the boundaries of the domain are 1-D (length) and 2-D (area), respectively, and the integral in Eq. (2.4.12) is evaluated over that local boundary segment. In one-dimensional problems, the boundaries are points. The integral in Eq. (2.4.12) is evaluated with the dirac delta function as

$$\int_{\Gamma_{su}} \alpha_u (u - \tilde{u}) v \delta(x = x_{\Gamma_{su}}) \, d\Gamma = \left[ \alpha_u (u - \tilde{u}) v \right]_{\Gamma_{su}} \, . \tag{2.4.13}$$

Equation (2.4.13) is the form of the penalty method that appears in the development of the weak form in section 2.1. Recall the discussion of the terms of Eq. (2.1.19) in section 2.1. The system of equations is of the form (see Eq. 2.1.18)

$$\mathbf{K}^{(\text{node})} \hat{\mathbf{u}} + \mathbf{K}^{(\text{bdry})} \hat{\mathbf{u}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = 0 \, . \tag{2.4.14}$$

Also recall the 17-node model of a 1-D domain in Figure 2.1.4, repeated in Figure 2.4.1 for convenience.

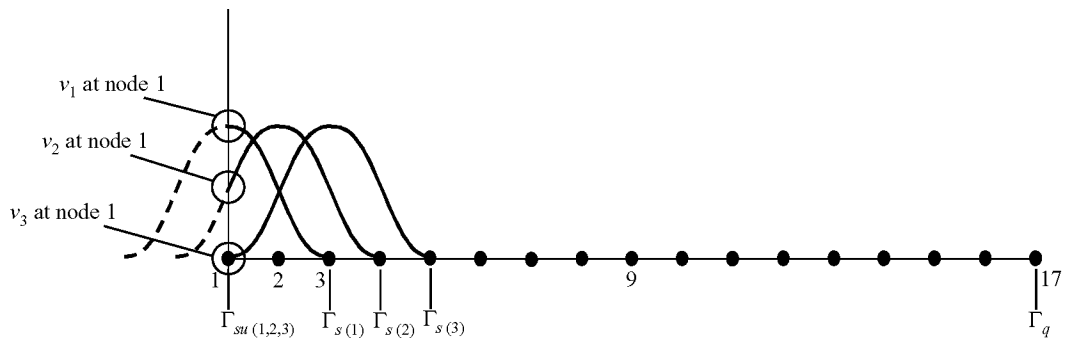



**Figure 2.4.1: Test functions near global boundary**

The primary variable, $u$, is prescribed at node 1, where $x = 0$. Using Eq. (2.1.10a),

$$u(x) = \sum_{j=1}^{n} \phi_j(x)\hat{u}_j \ , \tag{2.4.15}$$

equation (2.4.13) can be rewritten as

$$\left[\alpha_u(u-\tilde{u})v\right]_{\Gamma_{su}} = \alpha_u\left[\phi_1 \quad \phi_2 \quad \dots \quad \phi_n\right]_{\Gamma_{su}} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix} v\big|_{x=0} - \alpha_u\tilde{u}\,v\big|_{x=0} \ . \tag{2.4.16}$$

The term of Eq. (2.4.16) must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_u$. In the model of Figure 2.4.1, these are nodes 1, 2, and 3. Similarly to the terms of Eq. (2.1.19), the key to the contribution of each of nodes 1, 2, and 3 to the term of Eq. (2.4.16) lies in the values of $v_1$, $v_2$, and $v_3$ at node 1. For node 3, $v_3|_{x=0} = 0$. For node 1, $v_1|_{x=0} = 1$. For node 2, $0 < v_2|_{x=0} < 1$. The term of Eq. (2.4.16), evaluated with each successive value of $v_i|_{x=0}$ for nodes $i = 1$, 2, and 3 contributes to both the $\mathbf{K}^{(\text{bdry})}$ and the $\mathbf{f}^{(\text{bdry})}$ of Eq. (2.4.14) (see Eqs. 2.3.11b and 2.3.12b). As previously discussed for the

terms of Eq. (2.1.19), a proper understanding of how the term of Eq. (2.4.16) is calculated provides users of the MLPG method with considerable freedom in choices of nodal spacing and sizes of test functions.

## 2.5 Numerical Examples

In this section, to demonstrate the validity of the MLPG algorithm, the method is applied to examples of 1-D $C^0$ problems. The following exact solutions are considered for "patch tests":

$$\begin{aligned} &\text{I)} \quad u = \text{constant} \\ &\text{II)} \quad u = x \\ &\text{III)} \quad u = x^2. \end{aligned} \tag{2.5.1}$$

To perform a patch test, each exact solution is prescribed as the essential boundary conditions in the problem, and the problem is analyzed with the MLPG algorithm. To pass the patch test, the MLPG algorithm must reproduce the exact solution at all interior nodes of the model to machine accuracy. In addition to the patch test problems, an example problem of heat transfer through a rectangular fin is studied.

### Problem Parameters

A uniform bar of length $l$ is considered. The bar is modeled using 5, 9, 17, and 33 equally spaced nodes. The 17-node model is presented in Figure 2.5.1.

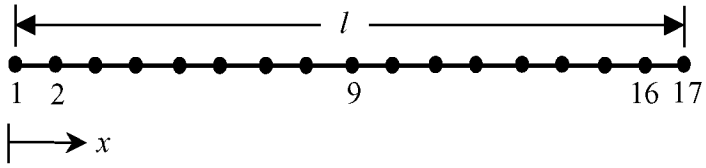



**Figure 2.5.1: A 17-node model of a bar of length *l***

A linear basis function $(1, x)$ should reproduce linear $(x^1)$ and lower order solutions exactly, and is therefore used for problems I and II of Eq. (2.5.1). Similarly, a quadratic basis $(1, x, x^2)$ should reproduce quadratic $(x^2)$ and lower order solutions exactly, and is therefore used for problem III of Eq. (2.5.1). A quadratic basis is also used for the heat transfer problem. The weak form (recall Eq. 2.1.17) requires that the approximating function, $u$, be differentiable at least once. The linear basis function is the lowest order basis function that meets this requirement, and therefore the lowest order basis function that can be used in the MLPG method for $C^0$ problems.

Recall that the governing differential equation is

$$-\frac{d}{dx}\left(b\frac{du}{dx}\right)+cu=f\,. \tag{2.5.2}$$

Here, $b$ and $c$ are user-defined constants. The patch tests are performed for various chosen values of these constants.

**I.** Patch Test – I: $b = 1$; $c = 0$; $u$ = constant = $\beta_1$, where $\beta_1$ is some arbitrary constant. Substitution of these values into Eq. (2.5.2) yields $f = 0$. EBCs are prescribed at nodes 1 $(x = 0)$ and $N$ $(x = l)$ of an $N$-node model as

$$\begin{aligned} u\big|_{x=0} &= \beta_1 \\ u\big|_{x=l} &= \beta_1\,. \end{aligned} \tag{2.5.3}$$

This patch test corresponds to an unstressed rigid body displacement (of magnitude $\beta_1$) of the bar. Values of $(R_o/l)$ and $(R_j/l)$ were chosen as $(R_o/l) = 2\Delta x$ and $(R_j/l) = 1.0$. For the 5- and 9-node models, the algorithm calculated the exact solutions for both the fictitious nodal values and the interpolated primary and secondary variables. For the 17-node model, the algorithm failed to calculate the exact solution for the fictitious nodal

values, but the interpolated values were exact. The value of $(R_j / l)$ was then reduced to $(R_j / l) = 8\Delta x$, and with this value and the 17-node model, the algorithm calculated the fictitious nodal values exactly. Similar results were obtained for the 33-node model. This suggests that the algorithm is capable of reproducing exact interpolated values, but exact fictitious nodal values depend on the parameter $(R_j / l)$. The values of $(R_o / l)$ and $(R_j / l)$ are henceforth chosen as $(R_o / l) = 2\Delta x$ for all models and $(R_j / l)$ = bar length for the 5- and 9-node models and $(R_j / l) = 8\Delta x$ for the 17- and 33-node models.

**II.** Patch Test – II: $b = 1$; $c = 0$; $u = x/l$

Substitution of $b$, $c$, and $u$ into Eq. (2.5.2) yields the loading $f = 0$. EBCs are prescribed at nodes 1 ($x = 0$) and $N$ ($x = l$) of an $N$-node model as

$$\begin{aligned} u\big|_{x=0} &= 0 \\ u\big|_{x=l} &= 1 \quad . \end{aligned} \tag{2.5.4}$$

The 5-, 9-, 17, and 33-node models yielded the exact solution with these boundary conditions at the nodes and every internal point in the domain, thus passing the patch test. The problem can also be worked as the case of a uniform bar with an end load, $\tilde{q}$ (see Figure 2.5.2), i.e., with an EBC prescribed at one end and an NBC prescribed at the other end.

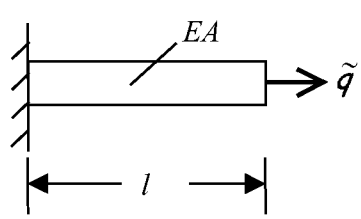



**Figure 2.5.2: Uniform bar of length $l$ with end load of magnitude $\tilde{q}$**

The prescribed boundary conditions and applied loading are

$$u\big|_{x=0} = \tilde{u} = 0$$

$$b\frac{du}{dx}\bigg|_{x=l} = \tilde{q} = 1 \quad \text{where} \quad b = EA \tag{2.5.5}$$

and

$$f = 0 \ .$$

Once again, the MLPG algorithm with each of the four models reproduced the exact solutions for the mixed boundary conditions.

**III.** Patch Test – III: $b = 0$; $c = 1$

The exact solution is

$$u = \left(\frac{x}{l}\right)^2 . \tag{2.5.6}$$

Substitution of $b$, $c$, and Eq. (2.5.6) into Eq. (2.5.2) yields the loading $f = (x/l)^2$. This analysis can be performed using three different sets of boundary conditions.

i) To perform the patch test, EBCs are prescribed at $x = 0$ and $x = l$ as

$$\begin{aligned} u\big|_{x=0} &= 0 \\ u\big|_{x=l} &= 1 \ . \end{aligned} \tag{2.5.7}$$

ii) Alternately, mixed boundary conditions are prescribed as

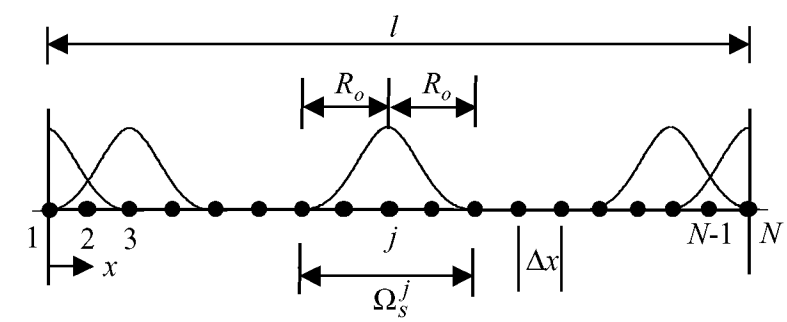
$$\begin{aligned} u\big|_{x=0} &= 0 \\ b\frac{du}{dx}\bigg|_{x=l} &= 2bx/l = 0 \ . \end{aligned} \tag{2.5.8}$$

iii) Thirdly, mixed boundary conditions are prescribed as

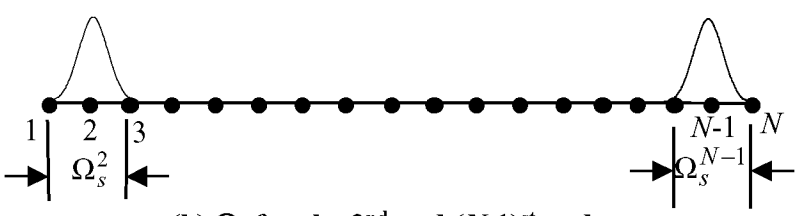
$$\begin{aligned} b\frac{du}{dx}\bigg|_{x=0} &= 0 \\ u\big|_{x=l} &= 1 \ . \end{aligned} \tag{2.5.9}$$

As expected, the MLPG analysis reproduced the exact solutions for all three cases for all nodes of the four models considered.

Recall the discussions of the boundary terms of Eqs. (2.1.19 and 2.4.16). In these discussions, it was noted that the size of $\Omega_s$ for each $v_i$ need not be uniform and that a simple choice of a smaller $(R_o / l)$ for nodes 2 and $N$-1 may be preferable. For example, consider the choice $(R_o / l) = 2\Delta x$ for the 17-node model of Figure 2.5.1. To account for the terms of Eqs. (2.1.19 and 2.4.16), where $0 < v_2 < 1$ and $0 < v_{16} < 1$, the $(R_o / l)$ for nodes 2 and 16 is chosen as $(R_o / l) = \Delta x = 0.0625$ for a bar of length $l = 1$. With this choice, the only nodes that contribute to the terms of Eqs. (2.1.19 and 2.4.16) are nodes 1 and 17. Figure 2.5.3 presents a visualization of the above assignments of $(R_o / l)$.




**(a) $\Omega_s$ for the 1st, 3rd, $j$th, ($n$-2)th, and $n$th nodes**




**(b) $\Omega_s$ for the 2nd and ($N$-1)st nodes**

**Figure 2.5.3: $\Omega_s$ definitions for various nodes**

The patch tests I, II, and III were performed with these new assignments of $(R_o / l)$. As expected, the MLPG analysis reproduced the exact solutions to machine accuracy, thus

passing the patch tests. These results demonstrate the fact that there is no numerical difference between the two choices of $(R_o/l)$, i.e. $(R_o/l) = 2\Delta x$ uniform for all nodes vs. $(R_o/l) = \Delta x = 0.0625$ for nodes 2 and $N$-1, as long as the terms of Eqs. (2.1.19 and 2.4.16) are evaluated correctly.

In the discussions of the boundary terms of Eqs. (2.1.19 and 2.4.16), it was also noted that nodes need not be equally spaced. Consider the 15-node model with unequal nodal spacing shown in Figure 2.5.4.

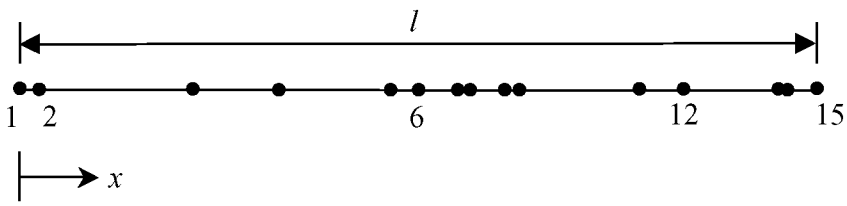



**Figure 2.5.4: A 15-node model with unequally spaced nodes**

This model was generated by randomly placing nodes in the region $0 < x < l$. The $(R_o/l)$ for each node was assigned a different value,

$$\Delta x \le (R_o/l) \le 2\Delta x \tag{2.5.10}$$

where $\Delta x$ is the distance between the nodes of the corresponding 17-node model with equal nodal spacing. For example, for the 17-node model of Figure 2.5.1, $\Delta x = 0.0625$. The $(R_o/l)$ for each node in the model of Figure 2.5.4 was chosen somewhere between $\Delta x = 0.0625$ and $2\Delta x = 0.125$. The patch tests I, II, and III were performed with $(R_o/l)$ for each node assigned as stated above. As expected, the MLPG analysis reproduced the exact solutions at all interior nodes in the model and at all interior points in the bar, thus passing the patch tests.

**Example: Heat transfer through rectangular fin**

Consider the rectangular cooling fin shown in Figure 2.5.5. If the variations along the $y$-direction are negligible, the fin can be modeled as a bar as in Figure 2.5.6, where $A$ is the cross-sectional area, $P$ is the perimeter, $w$ is the width, $l$ is the length, and $t$ is the thickness.

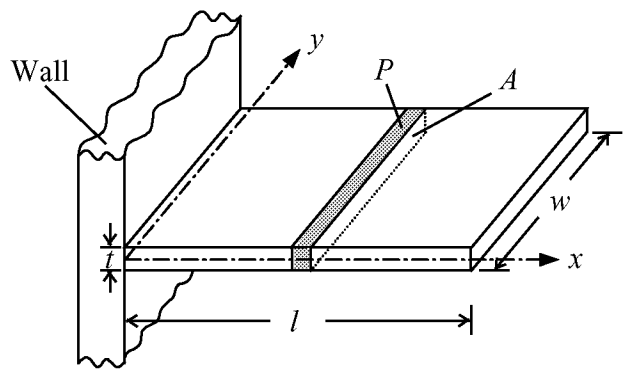



**Figure 2.5.5: Rectangular cooling fin**

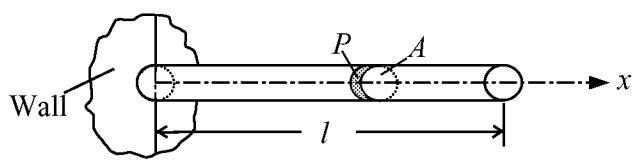



**Figure 2.5.6: Bar model of rectangular cooling fin**

The governing equation is (Reddy, 1993, pp. 133-134)

$$-\frac{d^2T}{dx^2}+\frac{\beta}{kt}(T-T_\infty)=0 \tag{2.5.10}$$

subjected to boundary conditions

$$\begin{aligned} &T(0)=T_{\text{Wall}} \\ &\left(kA\frac{dT}{dx}\right)\bigg|_{x=l}=0 \end{aligned} \tag{2.5.11}$$

where $T$ is temperature, $k$ is thermal conductivity, $\beta$ is the film coefficient $T_\infty$ is the ambient temperature, and $T_{\text{Wall}}$ is the temperature of the wall. The equations are recast via the non-dimensional quantities

$$\Theta = \frac{T - T_\infty}{T_{\text{Wall}} - T_\infty} \quad , \quad \xi = \frac{x}{l} \quad , \quad N = \left( \frac{\beta l^2}{kt} \right)^{1/2} \tag{2.5.12}$$

as

$$-\frac{d^2\Theta}{d\xi^2} + N^2\Theta = 0 \tag{2.5.13}$$

subjected to

$$\begin{aligned} &\Theta(0) = 1 \\ &b\left( \frac{d\Theta}{d\xi} \right)\bigg|_{\xi=1} = 0. \end{aligned} \tag{2.5.14}$$

The exact solution of the problem is

$$\begin{aligned} &\Theta(\xi) = \Theta(0)\frac{\cosh N(l-\xi)}{\cosh N l} \quad , \\ &H = b\frac{d\Theta}{d\xi} = -bN\frac{\sinh N(l-\xi)}{\cosh N l} \quad . \end{aligned} \tag{2.5.15}$$

In the numerical analysis of the problem, the value of $N$ was chosen as $N = 4$. The test function was chosen as Eq. 2.3.7 with $\beta = 4$. The trial function was constructed from the weight function of Eq. 2.2.23 with $\alpha = 4$ and a quadratic basis function. The parameters $(R_o / l)$ and $(R_j / l)$ were chosen as $2\Delta x$ and $8\Delta x$ (not exceeding the bar length), respectively. The integrations were performed using a 10-point Gaussian integration, and the penalty parameter was chosen as $10^6$. The bar was analyzed with four models with 5, 9, 17, and 33 equally spaced nodes. Table 2.5.1 presents the values of the primary and

secondary variables obtained with the 5-, 9-, 17-, and 33-node models at stations $\xi = 0,\ 0.5,$ and $1.0$ along the length of the bar. The values of the exact solutions are also included in this table at these stations. All models yielded very good results and the accuracy of the solutions improved with model refinement.

**Table 2.5.1: Comparison of the MLPG solution with the exact solution**

| | | Exact solution (Eq. 2.5.15) | MLPG model with: | | | |
|---|---|---|---|---|---|---|
| | | | 5 nodes | 9 nodes | 17 nodes | 33 nodes |
| $\Theta(\xi)$ | $\xi = 0$ | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| | $\xi = 0.5$ | 0.1378 | 0.1360 | 0.1377 | 0.1377 | 0.1378 |
| | $\xi = 1.0$ | 0.0366 | 0.0420 | 0.0369 | 0.0360 | 0.0366 |
| $d\Theta/d\xi$ | $\xi = 0$ | -3.9973 | -4.1308 | -4.2705 | -4.0322 | -3.9843 |
| | $\xi = 0.5$ | -0.5312 | -0.5502 | -0.5310 | -0.5309 | -0.5305 |
| | $\xi = 1.0$ | 0 | 0.2737 | 0.0468 | -0.0415 | 0.0024 |

Since the exact solution for this problem is not a simple polynomial, the MLPG method did not reproduce the exact solution. Error norms defined as

$$\|e_\Theta\| = \sqrt{\frac{1}{M}\sum_{g=1}^{M}(\Theta_{MLPG} - \Theta_{Exact})_g^2} \tag{2.5.16a}$$

and

$$\|e_H\| = \sqrt{\frac{1}{M}\sum_{g=1}^{M}(H_{MLPG} - H_{Exact})_g^2} \tag{2.5.16b}$$

were computed at $M$ uniformly spaced points along the bar. These interior points need not be coincident with nodes in the model. A value of $M = 50$ was used. The norms $\|e_\theta\|$ and $\|e_H\|$ are presented in Table 2.5.2. As expected, all models yielded accurate solutions (within 4%), and the error norms improved with model refinement.

**Table 2.5.2: Error norm $\|e\|$ for the 5-, 9-, 17-, and 33-node models**

| Error norm | Number of nodes in the model | | | |
|---|---|---|---|---|
| | 5 | 9 | 17 | 33 |
| $\|e_\Theta\|$ | 0.3127e-2 | 0.6711e-3 | 0.2195e-3 | 0.2154e-4 |
| $\|e_H\|$ | 0.3844e-1 | 0.6590e-2 | 0.5813e-2 | 0.3301e-3 |

Some post-processing is required to evaluate the secondary variables from the fictitious nodal values. To calculate the secondary variables at an interior point, one has a choice of two methods. In the first method, the nearest neighboring node to this interior point in the domain is evaluated. All the nodes in the domain of influence of this node are determined. The nodal shape functions of all these nodes are evaluated at the interior point. These shape functions' values and the fictitious nodal values are then used to find the value of the solution $u$ by direct application of Eq. (2.2.21):

$$u(x) \cong \sum_{j=1}^{n} \hat{u}_j \phi_j(x). \tag{2.5.17}$$

Secondary variables may be found in the same direct manner via Eq. (2.3.3):

$$\frac{du(x)}{dx} \cong \sum_{j=1}^{n} \hat{u}_j \frac{d\phi_j(x)}{dx} \tag{2.5.18}$$

The derivatives of the shape functions are computed at the same time as the shape functions themselves, and hence no additional procedures are required. In the second method for calculating secondary variables, a shape function is formed over the interior point, and all the nodes in the domain of influence of this interior point are determined. The fictitious nodal values of these nodes are then used with the value of the shape function to find the value of the solution $u$ and the secondary variables via Eqs. (2.5.17 and 2.5.18). The MLPG and exact secondary variable distributions for the 17-node model of the heat transfer problem are presented in Figure 2.5.7, and these values agree

with the exact solution at all points along the bar. This example demonstrates that one can obtain a smooth distribution of the secondary variable.

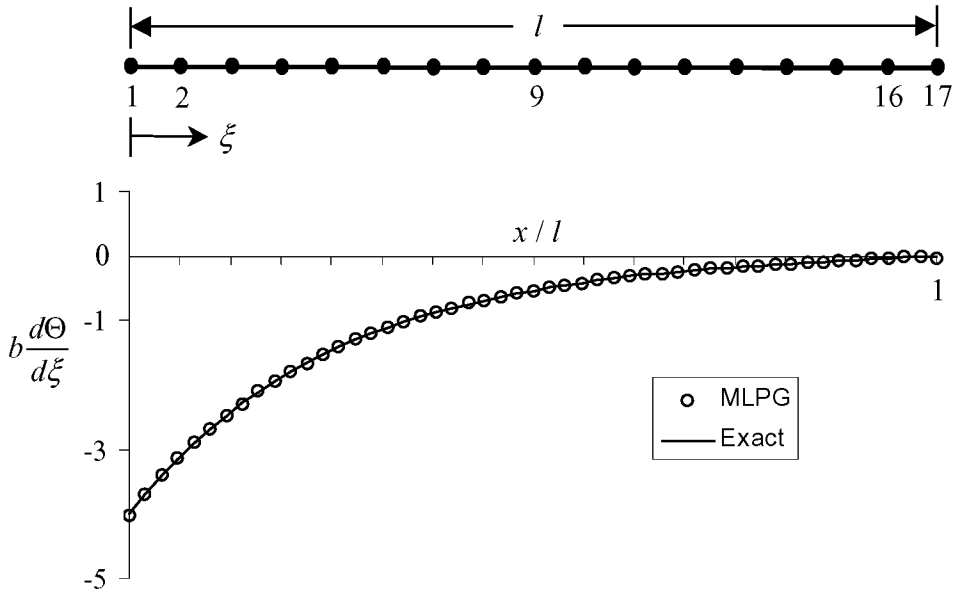



**Figure 2.5.7: Comparison of the MLPG and exact secondary variable distributions for a 17-node model with uniform nodal spacing**

The same heat transfer problem was then worked using the 15-node model of Figure 2.5.4. The ($R_o$ / $l$) and ($R_j$ / $l$) were chosen as in the 17-equally spaced nodal model. The MLPG and exact secondary variable distributions are presented in Figure 2.5.8. From this figure, it is seen that the MLPG solution in the region $0 \le x < l/2$ is not as accurate as the MLPG solution in the region $l/2 \le x < l$. This inaccuracy is due to the large distance between nodes in the region $0 \le x < l/2$. To improve the accuracy in this region, two additional nodes were "sprinkled" into the domain of the problem (see Figure 2.5.9). The MLPG solutions before and after model refinement and the exact solution are compared in Figure 2.5.9. The inclusion of the two additional nodes significantly improves the solution in the region $0 \le x < l/2$.

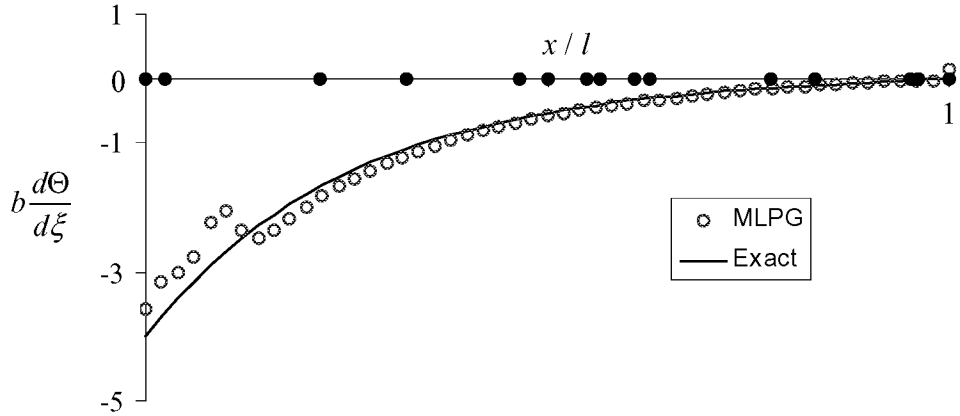



**Figure 2.5.8: Comparison of the MLPG and exact secondary variable distributions for a 15-node model with non-uniform nodal spacing**

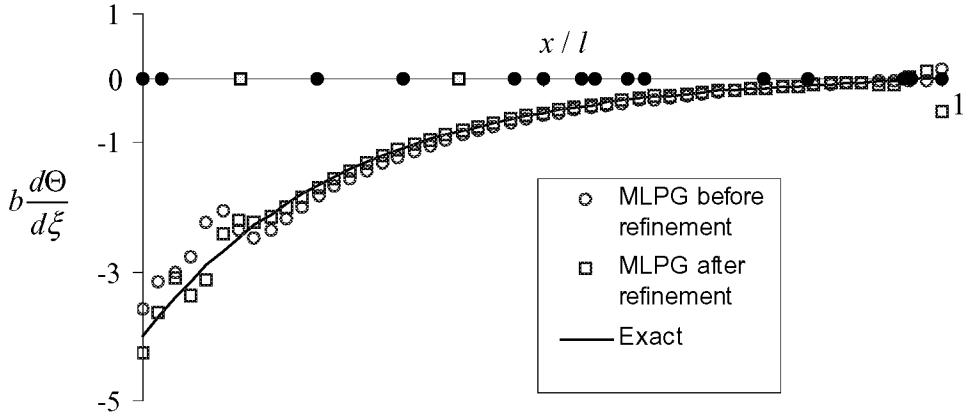



**Figure 2.5.9: Comparison of the MLPG secondary variable distribution before and after model refinement**

## 2.6 Concluding Remarks

This chapter presented the MLPG method applied to $C^0$ one-dimensional (1-D) problems. In the local weak form (LWF) of the governing differential equation, a moving least squares (MLS) interpolation was used to form the approximations to the solution known as trial functions. Test functions, also needed for the LWF were chosen from a different space than the trial functions, making the method a Petrov-Galerkin

method. This choice of test functions led to unsymmetric stiffness matrices. The essential boundary conditions were enforced by a penalty method, and numerical integration was used to evaluate the integrals in the system matrices. The MLPG method was applied to and passed several patch test problems. The method was then applied to a typical heat transfer problem. Very good results for both the primary and secondary variables were obtained. A smooth distribution of the secondary variable was obtained without the use of elaborate post processing techniques.

# Chapter 3: MLPG for $C^1$ Problems

In Chapter 2, the MLPG method was studied for the deformation of bars – $C^0$ problems. In this chapter, the MLPG method is further developed for bending of beams – $C^1$ problems. A local weak form is developed from the classical weighted-residual form of the governing differential equation. A generalized moving least squares interpolation scheme is used to construct the approximations to the solution known as trial functions. Under the Petrov-Galerkin paradigm, the test functions are chosen from a different space than the trial functions as combinations of simple weight functions and their derivatives. System matrices are derived by substituting the trial and test functions into the local weak form.

## 3.1 Beam Theory

The MLPG method for $C^1$ problems presented in this report was developed using the Euler-Bernoulli beam conventions. Consider the beam shown in Figure 3.1.1. Under the Euler-Bernoulli bending assumptions, plane sections normal to the neutral axis before deformation remain planar and normal to the neutral axis after deformation. The deflection $w$ in the $z$-direction is a function of the $x$-coordinate alone, i.e.,

$$w = w(x), \quad u = u(x), \quad \text{and} \quad v = 0 . \tag{3.1.1}$$

In Figure 3.1.1b, consider ΔADC in which

$$\tan\theta = \frac{\mathrm{CD}}{\mathrm{AC}} = \frac{\Delta w}{\Delta x} . \tag{3.1.2}$$

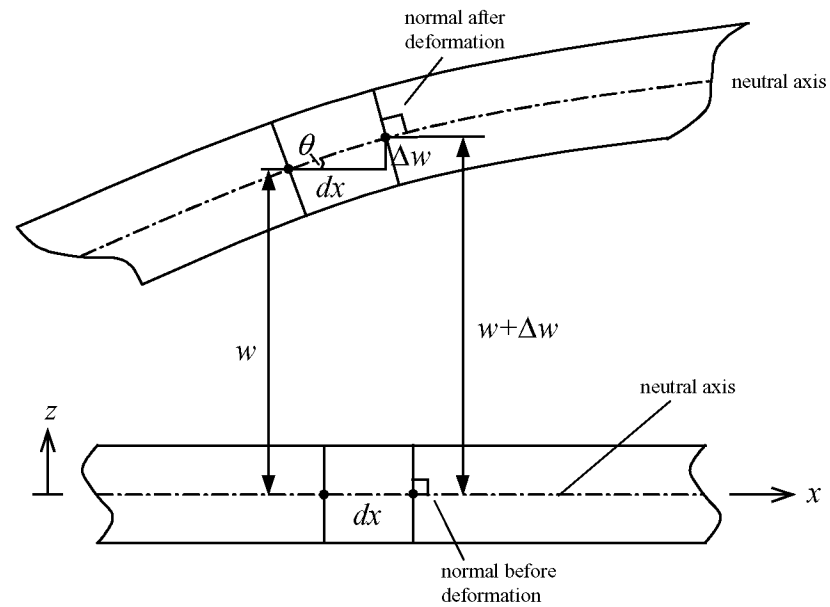
As $\Delta x \rightarrow 0$, and for small angles, $\tan\theta \approx \theta$ gives

$$\theta = \frac{dw}{dx} \tag{3.1.3}$$

where $\theta$ is the slope of the neutral axis. Consider the ΔABC in Figure 3.1.1c.

$$\angle \mathrm{BAC} = \angle \mathrm{CAD} = \theta \tag{3.1.4}$$

because normals before deformation remain normal after deformation.




**(a) Beam configuration before and after deformation**

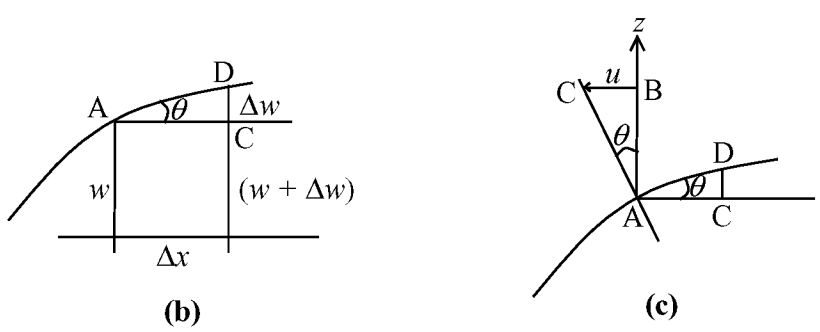



**Figure 3.1.1: Euler-Bernoulli beam**

In ΔABC of Figure 3.1.1c,

$$\frac{\mathrm{BC}}{\mathrm{AB}} = \tan\theta \tag{3.1.5a}$$

or

$$u = -z\tan\theta \approx -z\frac{dw}{dx}. \tag{3.1.5b}$$

The strains that correspond to $u = -z(dw/dx)$, $v = 0$, and $w = w(x)$ can then be evaluated as

$$\begin{aligned}
\varepsilon_x &= \frac{\partial u}{\partial x} = -z\frac{d^2w}{dx} & \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0 \\
\varepsilon_y &= \frac{\partial v}{\partial y} = 0 & \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} = 0 \\
\varepsilon_z &= \frac{\partial w(x)}{\partial z} = 0 & \gamma_{zx} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} = \frac{dw}{dx} - \frac{dw}{dx} = 0
\end{aligned} \tag{3.1.6}$$

Thus all strains except $\varepsilon_x$ are zero. Using the constitutive relationships, the stresses can be evaluated. The stress $\sigma_x$ corresponding to $\varepsilon_x$ can be evaluated as

$$\sigma_x = E\varepsilon_x = -Ez\frac{d^2w}{dx^2}. \tag{3.1.7}$$

Now consider the beam segment subjected to a moment in Figure 3.1.2. The moment, $M$, required to return the beam to its undeformed state is

$$\begin{aligned}
M &= -\int_A (\sigma_x b dz) z \\
&= \int_A Ez^2 b\frac{d^2w}{dx^2} dz \\
&= E\frac{d^2w}{dx^2}\int_A bz^2 dz \quad .
\end{aligned} \tag{3.1.8}$$

The term $\int_A bz^2 dz$ is the second moment of the area about the $y$-axis and is usually termed as the moment of inertia, $I_{yy}$,

$$M = EI_{yy}\frac{d^2w}{dx^2}. \tag{3.1.9}$$

Customarily in this beam theory, the subscripts *yy* are dropped and the moment of inertia is written as *I*. Hence, $M = EI \cdot \left(d^2w/dx^2\right)$. Similarly, the shear, *V*, is

$$V = -EI\frac{d^3w}{dx^3}. \tag{3.1.10}$$

In this report, examples for thin Euler-Bernoulli beams that undergo small displacements are considered.

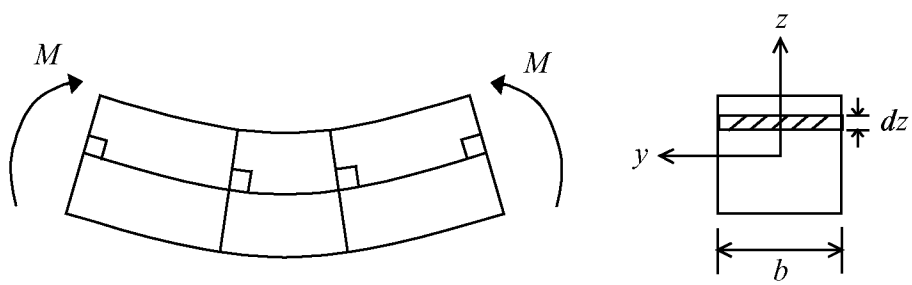



**Figure 3.1.2: Beam segment subjected to a moment**

### 3.2 Local Weak Form for Euler-Bernoulli Beam Problems

The governing equation for an Euler-Bernoulli beam is

$$EI\frac{d^4w}{dx^4} = f \qquad \text{in domain } \Omega\ (0 \le x \le l) \text{ with boundary } \Gamma \tag{3.2.1}$$

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

$$w = \tilde{w} \qquad \text{on } \Gamma_w \quad \text{and}$$
$$\frac{dw}{dx} = \tilde{\theta} \qquad \text{on } \Gamma_\theta \quad , \tag{3.2.2}$$

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

$$V = \tilde{V} \qquad \text{on } \Gamma_V \quad \text{and}$$
$$M = \tilde{M} \qquad \text{on } \Gamma_M \tag{3.2.3}$$

where $V$ and $M$ are the shear force and bending moment, respectively, and are related to the deflection $w$ as (see Eqs. 3.1.9 and 3.1.10)

$$V = -EI\frac{d^3w}{dx^3} \qquad \text{and} \qquad M = EI\frac{d^2w}{dx^2} \tag{3.2.4}$$

and $\Gamma_w$, $\Gamma_\theta$, $\Gamma_V$, and $\Gamma_M$ denote the boundary points where deflection ($w$), slope ($\theta$), shear ($V$), and moment ($M$) are prescribed, respectively. Note that $\tilde{w}$ and $\tilde{V}$ and $\tilde{\theta}$ and $\tilde{M}$ are mutually disjoint (Atluri *et al.*, 1999 and Gu and Liu, 2001), i.e., when $w = \tilde{w}$, the shear force $V$ becomes the corresponding reaction, and when $\theta = \tilde{\theta}$, the moment $M$ becomes the corresponding reaction.

The weak form of the governing differential equation is obtained in a similar manner as for $C^0$ problems. The residual error to be minimized is

$$R = EI\frac{d^4w}{dx^4} - f\,. \tag{3.2.5}$$

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

$$0 = \int_{\Omega} \left( EI \frac{d^4 w}{dx^4} - f \right) v \, dx \, . \tag{3.2.6}$$

An approximate solution for $w$ is chosen such that each term in the approximate solution must be four times differentiable and satisfy all the boundary conditions (Eqs. 3.2.2 and 3.2.3). These requirements are difficult to satisfy. Therefore, a formulation that accepts weaker requirements on $w$ is sought. The weak form of the weighted residual equation is set up by transferring the differentiation from the variable $w$ to the weight function $v$. This is achieved by integrating by parts twice. Integrating by parts once yields

$$0 = -EI \int_{\Omega} \frac{d^3 w}{dx^3} \frac{dv}{dx} dx - \int_{\Omega} f \, v \, dx + n_x \left[ EI \frac{d^3 w}{dx^3} v \right]_{\Gamma} \tag{3.2.7}$$

where $n_x[EI(d^3w/dx^3)v]_{\Gamma}$ is introduced as a boundary term and $n_x$ is the direction cosine of the unit outward drawn normal to $\Omega$ with respect to the $x$-axis. The $n_x$ thus takes values $\pm 1$ in 1-D problems. The prescription of the secondary variable $EI(d^3w/dx^3)$ on $\Gamma$ is a natural boundary condition and is now part of the weak form. Integrating by parts a second time to equalize the derivatives of $w$ and $v$ yields

$$0 = EI \int_{\Omega} \frac{d^2 w}{dx^2} \frac{d^2 v}{dx^2} dx - \int_{\Omega} f \, v \, dx + n_x \left[ EI \frac{d^3 w}{dx^3} v \right]_{\Gamma} - n_x \left[ EI \frac{d^2 w}{dx^2} \frac{dv}{dx} \right]_{\Gamma} \tag{3.2.8}$$

where $n_x[EI(d^2w/dx^2)(dv/dx)]_{\Gamma}$ is introduced as an additional boundary term. The prescription of the secondary variable $EI(d^2v/dx^2)$ on $\Gamma$ is also a natural boundary condition and is now part of the weak form. The requirements on the approximate solution have thus been weakened, i.e., $w$ must now be differentiable twice and must satisfy the essential boundary conditions. Additionally, the essential boundary conditions

are enforced by a penalty method (Atluri *et al.*, 1999). As in section 2.4, the penalty terms are written as

$$\alpha_w[(w-\tilde{w})v]_{\Gamma_w} \tag{3.2.9a}$$

and

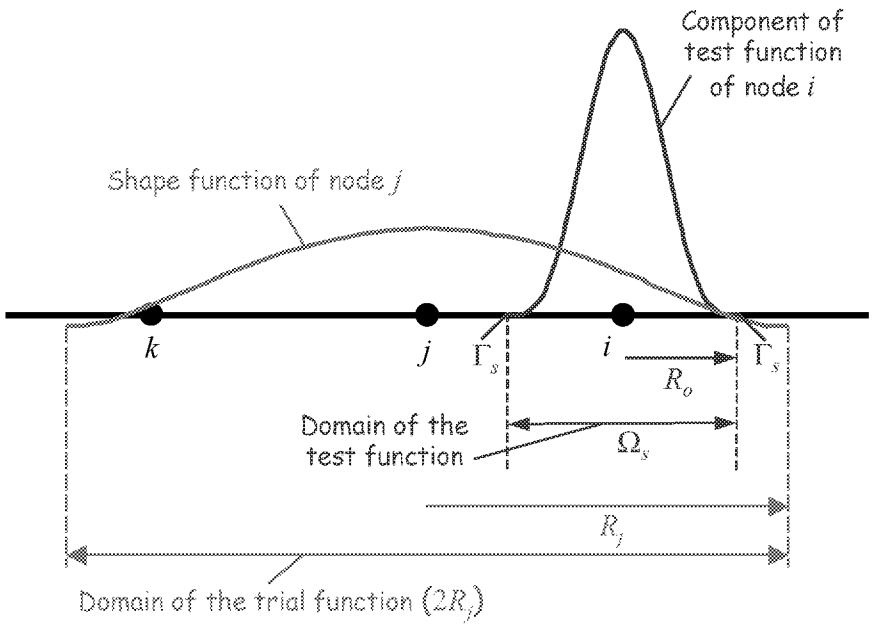
$$\alpha_\theta\left[\left(\frac{dw}{dx}-\tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_\theta} \tag{3.2.9b}$$

where $\alpha_w$ and $\alpha_\theta$ are the penalty parameters to enforce the deflection and slope boundary conditions, respectively. Thus, including the penalty terms, Eq. (3.2.8) is written as

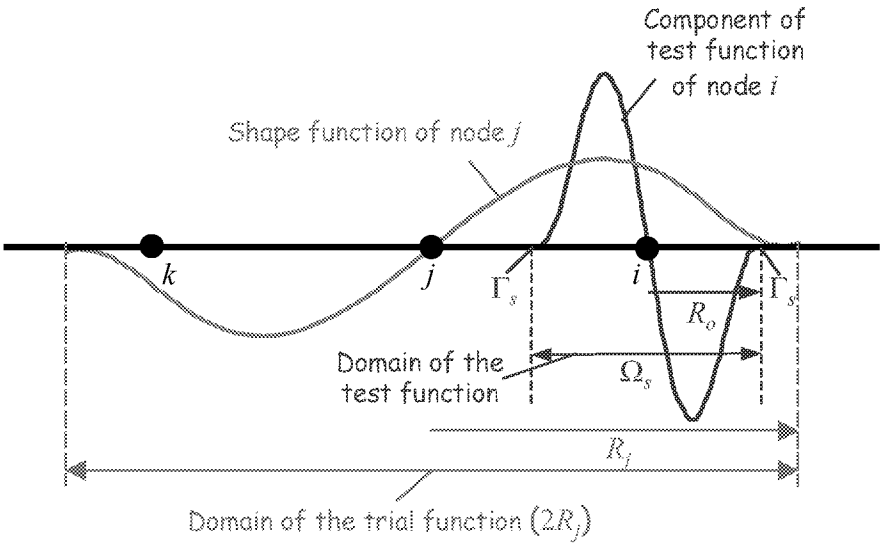
$$\begin{aligned} 0 = EI\int_\Omega \frac{d^2w}{dx^2}\frac{d^2v}{dx^2}dx - \int_\Omega f\,v\,dx + \alpha_w[(w-\tilde{w})v]_{\Gamma_w} + \alpha_\theta\left[\left(\frac{dw}{dx}-\tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_\theta} \\ + n_x\left[EI\frac{d^3w}{dx^3}v\right]_\Gamma - n_x\left[EI\frac{d^2w}{dx^2}\frac{dv}{dx}\right]_\Gamma \end{aligned} \tag{3.2.10}$$

In Eq. (3.2.10), called the weak form of the governing differential equation, the chosen approximations for $w$ are called the trial functions, and $v$ are now called the test functions.

As discussed in Chapter 2, the test functions are chosen independently from the trial functions. Test function components chosen in this report for the primary variable $w$ in 1-D $C^1$ problems are the same as those chosen for $u$ in 1-D $C^0$ problems. Test function components chosen for $\theta$ in 1-D $C^1$ problems are the first derivatives of the components chosen for $w$, as $\theta = dw/dx$ is also a primary variable (see section 3.3). A typical component of the test function $v_i$ for node $i$ in 1-D (in comparison with a trial function component (shape function) for node $j$) is shown in Figure 3.2.1a. As for $C^0$ problems, these components vanish at a certain controllable distance from node $i$.




**(a) Deflection, *w***




**(b) Slope, *θ***

**Figure 3.2.1: Comparison of the domains of the trial and test functions**

The derivatives of these components also vanish at the same distance from node $i$ (see Figure 3.2.1b). This localized property of the test functions preserves the local character of the method. The integrations over $\Omega$ become integrations over a local sub-domain, $\Omega_s$,

and the $\Omega_s$ can be determined from the extent of the test functions (see Figures 3.2.1).

The weak form is therefore written for the local sub-domain $\Omega_s$ as

$$0 = EI \int_{\Omega_s} \frac{d^2 w}{dx^2}\frac{d^2 v}{dx^2} dx - \int_{\Omega_s} f\, v\, dx + \alpha_w \left[ (w - \tilde{w}) v \right]_{\Gamma_{sw}} + \alpha_\theta \left[ \left( \frac{dw}{dx} - \tilde{\theta} \right) \frac{dv}{dx} \right]_{\Gamma_{s\theta}}$$
$$+ n_x \left[ EI \frac{d^3 w}{dx^3} v \right]_{\Gamma_s} - n_x \left[ EI \frac{d^2 w}{dx^2}\frac{dv}{dx} \right]_{\Gamma_s} \tag{3.2.11}$$

where $\Gamma_{sw}$ and $\Gamma_{s\theta}$ are the boundaries where $w$ and $\theta$ are prescribed on the local boundary ($\Gamma_s \cap \Gamma_w$ and $\Gamma_s \cap \Gamma_\theta$). Note that if the local boundary $\Omega_s$ does not intersect $\Gamma_w$ or $\Gamma_\theta$ (i.e. when the $\Omega_s$ is completely within the interior of $\Omega$), the penalty terms are not considered for that local boundary. Recalling Eqs. (3.2.4), Eq. (3.2.11) is written as

$$0 = EI \int_{\Omega_s} \frac{d^2 w}{dx^2}\frac{d^2 v}{dx^2} dx - \int_{\Omega_s} f\, v\, dx + \alpha_w \left[ (w - \tilde{w}) v \right]_{\Gamma_{sw}} + \alpha_\theta \left[ \left( \frac{dw}{dx} - \tilde{\theta} \right) \frac{dv}{dx} \right]_{\Gamma_{s\theta}}$$
$$- n_x \left[ V\, v \right]_{\Gamma_s} - n_x \left[ M \frac{dv}{dx} \right]_{\Gamma_s} \quad . \tag{3.2.12}$$

When the local boundary $\Gamma_s$ intersects the global boundary $\Gamma$, four boundary conditions are possible (Atluri *et al.*, 1999):

$$\Gamma_s \cap \Gamma_w\,, \qquad \Gamma_s \cap \Gamma_\theta\,,$$
$$\Gamma_s \cap \Gamma_V\,, \text{ and } \Gamma_s \cap \Gamma_M\,. \tag{3.2.13}$$

Utilizing these subsets, Eq. (3.2.12) becomes

$$
\begin{aligned}
0 = EI \int_{\Omega_s} \frac{d^2 w}{dx^2}\frac{d^2 v}{dx^2} dx - \int_{\Omega_s} f\, v\, dx + \alpha_w \left[ (w - \tilde{w}) v \right]_{\Gamma_{sw}} + \alpha_\theta \left[ \left( \frac{dw}{dx} - \tilde{\theta} \right) \frac{dv}{dx} \right]_{\Gamma_{s\theta}} \\
- n_x [V\, v]_{\Gamma_s} - n_x \left[ M \frac{dv}{dx} \right]_{\Gamma_s} \\
- n_x \left[ \tilde{V}\, v \right]_{\Gamma_s \cap \Gamma_V} - n_x [V\, v]_{\Gamma_s \cap \Gamma_w} \\
- n_x \left[ \tilde{M} \frac{dv}{dx} \right]_{\Gamma_s \cap \Gamma_M} - n_x \left[ M \frac{dv}{dx} \right]_{\Gamma_s \cap \Gamma_\theta} .
\end{aligned}
\tag{3.2.14}
$$

As mentioned previously, the test function, $v$, and its derivatives can be chosen to vanish on $\Gamma_s$ (see Figures 3.2.1). Equation (3.2.14) then is reduced to the local weak form (LWF) for the MLPG method:

$$
\begin{aligned}
0 = EI \int_{\Omega_s} \frac{d^2 w}{dx^2}\frac{d^2 v}{dx^2} dx - \int_{\Omega_s} f\, v\, dx + \alpha_w \left[ (w - \tilde{w}) v \right]_{\Gamma_{sw}} + \alpha_\theta \left[ \left( \frac{dw}{dx} - \tilde{\theta} \right) \frac{dv}{dx} \right]_{\Gamma_{s\theta}} \\
- n_x \left[ \tilde{V}\, v \right]_{\Gamma_{sV}} - n_x \left[ \tilde{M} \frac{dv}{dx} \right]_{\Gamma_{sM}} + n_x \left[ EI \frac{d^3 w}{dx^3} v \right]_{\Gamma_{sw}} - n_x \left[ EI \frac{d^2 w}{dx^2} \frac{dv}{dx} \right]_{\Gamma_{s\theta}}
\end{aligned}
\tag{3.2.15}
$$

where, as in Eq. (3.2.11), $\Gamma_{sw}$ represents $\Gamma_s \cap \Gamma_w$ and $\Gamma_{s\theta}$ represents $\Gamma_s \cap \Gamma_\theta$, and similarly, $\Gamma_{sV}$ represents $\Gamma_s \cap \Gamma_V$ and $\Gamma_{sM}$ represents $\Gamma_s \cap \Gamma_M$. Now $n_x$ is the direction cosine of the unit outward drawn normal to $\Omega_s$; $n_x = 1$ if the boundary is on the right side of $\Omega_s$, and $n_x = -1$ if the boundary is on the left side of $\Omega_s$. The weak form of Eq. (3.2.15) is local because the integrations are performed over the local sub-domain $\Omega_s$.

The trial functions are written as

$$
w(x) = \sum_{j=1}^{n} \left( \hat{w}_j \psi_j^{(w)}(x) + \hat{\theta}_j \psi_j^{(\theta)}(x) \right), \tag{3.2.16a}
$$

and the test functions are written as

$$v(x) = \mu_i^{(w)} \chi_i^{(w)}(x) + \mu_i^{(\theta)} \chi_i^{(\theta)}(x) \,. \tag{3.2.16b}$$

As discussed in Chapter 2, if the trial and test functions of Eqs. (3.2.16) are chosen from the same space via a Galerkin method, symmetric stiffness matrices are obtained from Eq. (3.2.15). Again, this is the case in the study of beam problems by Atluri *et al.* (1999). In this report, a Petrov-Galerkin method is used, and thus the resulting stiffness matrices are not symmetric. The details of the development of the trial and test functions are presented in sections 3.3 and 3.4. Substitution of the trial and test functions into Eq. (3.2.15) yields a system of equations of the form

$$\mathbf{K}^{(\text{node})}\hat{\mathbf{d}} + \mathbf{K}^{(\text{bdry})}\hat{\mathbf{d}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = \mathbf{0} \tag{3.2.17}$$

where the superscript "bdry" denotes boundary. Note that the locality of the MLPG method (as integrations are performed over $\Omega_s$) makes the stiffness matrices of Eq. (3.2.17) banded. This is one of the advantages of the FEM that is retained by the MLPG method. The detailed formation of the system of equations of Eq. (3.2.17) is presented in section 3.4.

3.2.1 <u>Boundary Terms in the LWF</u>

As in Chapter 2, the boundary terms in the weak form need special attention. The issues related to these boundary terms are discussed below.

Consider the boundary terms of the LWF:

$$n_x \left[ EI \frac{d^3 w}{dx^3} v \right]_{\Gamma_{sw}} \quad , \quad n_x \left[ EI \frac{d^2 w}{dx^2} \frac{dv}{dx} \right]_{\Gamma_{s\theta}} \tag{3.2.18a}$$

$$n_x\left[\tilde{V}\, v\right]_{\Gamma_{sV}} \quad , \quad n_x\left[\tilde{M}\frac{dv}{dx}\right]_{\Gamma_{sM}} \tag{3.2.18b}$$

$$\alpha_w\left[(w-\tilde{w})v\right]_{\Gamma_{sw}} \quad , \quad \alpha_\theta\left[\left(\frac{dw}{dx}-\tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_{s\theta}} \tag{3.2.18c}$$

The terms of Eq. (3.2.18a) resemble the term $[vq]_{\Gamma_{su}}$ of Eq. (2.1.19), and the terms of Eq. (3.2.18b) resemble the term $\left[v\tilde{q}\right]_{\Gamma_{sq}}$ of Eq. (2.1.19). Likewise, the terms of Eq. (3.2.18c) resemble the terms of Eq. (2.4.16). These terms need to be evaluated at the boundary points. The boundary term evaluations are explained with the aid of a typical 17-node model of a beam as shown in Figure 3.2.2.

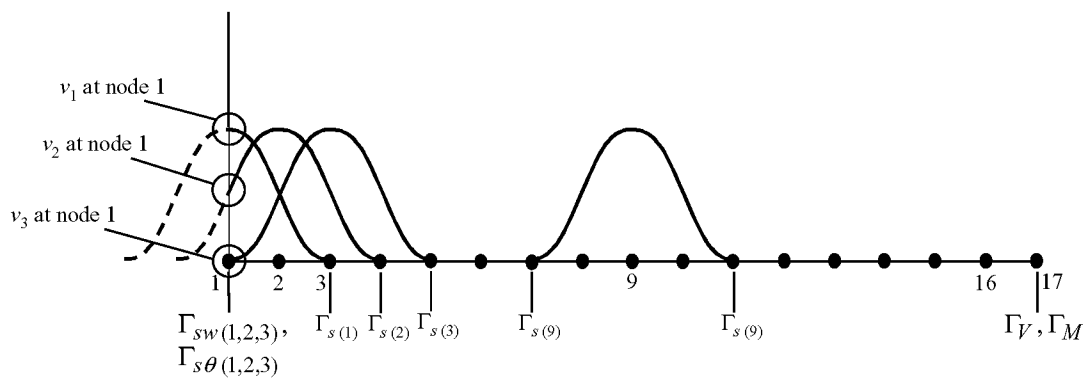



**Figure 3.2.2: Test functions at various nodes in a 17-node model**

The primary variables, $w$ and $\theta$, are assumed to be prescribed at node 1, and the secondary variables, $V$ and $M$, at node 17. Recall that $\tilde{w}$ and $\tilde{V}$ and $\tilde{\theta}$ and $\tilde{M}$ are mutually disjoint, i.e., for example, $w$ and $V$ cannot be prescribed on the same boundary point. In Figure 3.2.2, the test functions with an $(R_o/l)$ of $2\Delta x$ located at nodes 1, 2, 3, and 9 in the model are shown. Consider the term $n_x\left[EI\left(d^3w/dx^3\right)v\right]_{\Gamma_{sw}}$ of Eq. (3.2.18a). This term must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_w$. In the model of Figure 3.2.2, there are three such nodes, nodes 1, 2, and 3. The key to the

contribution of each of nodes 1, 2, and 3 to the term $n_x\left[EI\left(d^3w/dx^3\right)v\right]_{\Gamma_{sw}}$ lies in the values of $v_1$, $v_2$, and $v_3$ at node 1, where $x = 0$ and $n_x = -1$. First consider node 3:

$$v_3 = 0 \text{ at node 1,} \tag{3.2.19}$$

and therefore,

$$n_x\left[EI\frac{d^3w}{dx^3}v_3\right]_{\Gamma_{sw}} = 0\,. \tag{3.2.20}$$

Now consider node 1:

$$v_1 = 1 \text{ at node 1,} \tag{3.2.21}$$

and, substituting Eq. (3.2.16a) into the term $n_x\left[EI\left(d^3w/dx^3\right)v\right]_{\Gamma_{sw}}$,

$$\begin{aligned} n_x\left[EI\frac{d^3w}{dx^3}v_1\right]_{\Gamma_{sw}} &= n_x\left[EI\frac{d^3w}{dx^3}\right]_{\Gamma_{sw}} = \\ &-1\cdot EI\left[\frac{d^3\psi_1^{(w)}}{dx^3}\quad\frac{d^3\psi_2^{(w)}}{dx^3}\quad\cdots\quad\frac{d^3\psi_n^{(w)}}{dx^3}\right]_{x=0}\begin{bmatrix}\hat{w}_1\\ \hat{w}_2\\ \vdots\\ \hat{w}_n\end{bmatrix} \\ &-1\cdot EI\left[\frac{d^3\psi_1^{(\theta)}}{dx^3}\quad\frac{d^3\psi_2^{(\theta)}}{dx^3}\quad\cdots\quad\frac{d^3\psi_n^{(\theta)}}{dx^3}\right]_{x=0}\begin{bmatrix}\hat{\theta}_1\\ \hat{\theta}_2\\ \vdots\\ \hat{\theta}_n\end{bmatrix}\,. \end{aligned} \tag{3.2.22}$$

Finally, consider node 2:

$$0 < v_2 < 1 \text{ at node1,} \tag{3.2.23}$$

and therefore,

$$n_x\left[EI\frac{d^3w}{dx^3}v_2\right]_{\Gamma_{sw}} \tag{3.2.24}$$

$$=-1\cdot EI\left[\frac{d^3\psi_1^{(w)}}{dx^3}\quad\frac{d^3\psi_2^{(w)}}{dx^3}\quad\cdots\quad\frac{d^3\psi_n^{(w)}}{dx^3}\right]_{x=0}\begin{bmatrix}\hat{w}_1\\ \hat{w}_2\\ \vdots\\ \hat{w}_n\end{bmatrix}\left(\mu_2^{(w)}\chi_2^{(w)}\Big|_{x=0}+\mu_2^{(\theta)}\chi_2^{(\theta)}\Big|_{x=0}\right)$$

$$-1\cdot EI\left[\frac{d^3\psi_1^{(\theta)}}{dx^3}\quad\frac{d^3\psi_2^{(\theta)}}{dx^3}\quad\cdots\quad\frac{d^3\psi_n^{(\theta)}}{dx^3}\right]_{x=0}\begin{bmatrix}\hat{\theta}_1\\ \hat{\theta}_2\\ \vdots\\ \hat{\theta}_n\end{bmatrix}\left(\mu_2^{(w)}\chi_2^{(w)}\Big|_{x=0}+\mu_2^{(\theta)}\chi_2^{(\theta)}\Big|_{x=0}\right)\ .$$

Note that the terms

$$-1\cdot EI\left[\frac{d^3\psi_1^{(w)}}{dx^3}\quad\frac{d^3\psi_2^{(w)}}{dx^3}\quad\cdots\quad\frac{d^3\psi_n^{(w)}}{dx^3}\right]_{x=0}$$

and

$$-1\cdot EI\left[\frac{d^3\psi_1^{(\theta)}}{dx^3}\quad\frac{d^3\psi_2^{(\theta)}}{dx^3}\quad\cdots\quad\frac{d^3\psi_n^{(\theta)}}{dx^3}\right]_{x=0}$$

in Eqs. (3.2.22 and 3.2.24) are evaluated at node 1 and contribute to the $\mathbf{K}^{(\mathrm{bdry})}$ of Eq. (3.2.17) (see Eq. 3.5.4e).

The remaining terms of Eqs. (3.2.18) are evaluated in the same manner as the terms of Eqs. (2.1.19 and 2.4.16) and using the trial and test functions of Eqs. (3.2.16) as discussed above. Consider the term $n_x\left[EI\left(d^2w/dx^2\right)\left(dv/dx\right)\right]_{\Gamma_{s\theta}}$ of Eq. (3.2.18a). This term must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_\theta$. In the model

of Figure 3.2.2, these are nodes 1, 2, and 3. The key to the contribution of each of nodes 1, 2, and 3 to the term $n_x\left[EI\left(d^2w/dx^2\right)(dv/dx)\right]_{\Gamma_{s\theta}}$ lies in the values of $(dv_1/dx)$, $(dv_2/dx)$, and $(dv_3/dx)$, at node 1, where $x = 0$ and $n_x = -1$. First consider node 3:

$$(dv_3/dx) = 0 \text{ at node 1,} \tag{3.2.25}$$

and therefore,

$$n_x\left[EI\left(d^2w/dx^2\right)(dv_3/dx)\right]_{\Gamma_{s\theta}} = 0\,. \tag{3.2.26}$$

Now consider node 1:

$$v_1 = 1 \text{ and } (dv_1/dx) = 0 \text{ at node 1,} \tag{3.2.27}$$

and therefore,

$$n_x\left[EI\left(d^2w/dx^2\right)(dv_1/dx)\right]_{\Gamma_{s\theta}} = 0\,. \tag{3.2.28}$$

Finally, consider node 2:

$$(dv_2/dx) \text{ is nonzero at node 1 (in fact, } dv_2/dx < 0 \text{ in Figure 3.2.2),} \tag{3.2.29}$$

and, substituting Eqs. (3.2.16) into the term $n_x\left[EI\left(d^2w/dx^2\right)(dv/dx)\right]_{\Gamma_{s\theta}}$,

$$n_x\left[EI\frac{d^2w}{dx^2}\frac{dv_2}{dx^2}\right]_{\Gamma_{s\theta}} \tag{3.2.30}$$

$$= -1\cdot EI\begin{bmatrix}\dfrac{d^2\psi_1^{(w)}}{dx^2} & \dfrac{d^2\psi_2^{(w)}}{dx^2} & \cdots & \dfrac{d^2\psi_n^{(w)}}{dx^2}\end{bmatrix}_{x=0}\begin{bmatrix}\hat{w}_1\\ \hat{w}_2\\ \vdots\\ \hat{w}_n\end{bmatrix}\left(\mu_2^{(w)}\left.\frac{d\chi_2^{(w)}}{dx}\right|_{x=0}+\mu_2^{(\theta)}\left.\frac{d\chi_2^{(\theta)}}{dx}\right|_{x=0}\right)$$

$$-1\cdot EI\begin{bmatrix}\dfrac{d^2\psi_1^{(\theta)}}{dx^2} & \dfrac{d^2\psi_2^{(\theta)}}{dx^2} & \cdots & \dfrac{d^2\psi_n^{(\theta)}}{dx^2}\end{bmatrix}_{x=0}\begin{bmatrix}\hat{\theta}_1\\ \hat{\theta}_2\\ \vdots\\ \hat{\theta}_n\end{bmatrix}\left(\mu_2^{(w)}\left.\frac{d\chi_2^{(w)}}{dx}\right|_{x=0}+\mu_2^{(\theta)}\left.\frac{d\chi_2^{(\theta)}}{dx}\right|_{x=0}\right).$$

Note that the terms

$$-1\cdot EI\begin{bmatrix}\dfrac{d^2\psi_1^{(w)}}{dx^2} & \dfrac{d^2\psi_2^{(w)}}{dx^2} & \cdots & \dfrac{d^2\psi_n^{(w)}}{dx^2}\end{bmatrix}_{x=0}$$

and

$$-1\cdot EI\begin{bmatrix}\dfrac{d^2\psi_1^{(\theta)}}{dx^2} & \dfrac{d^2\psi_2^{(\theta)}}{dx^2} & \cdots & \dfrac{d^2\psi_n^{(\theta)}}{dx^2}\end{bmatrix}_{x=0}$$

in Eq. (3.2.30) are evaluated at node 1 and contribute to the $\mathbf{K}^{(\mathrm{bdry})}$ of Eq. (3.2.17) (see Eq. 3.5.4e).

Now consider the term $n_x\left[\tilde{\nabla}\, v\right]_{\Gamma_{sV}}$ of Eq. (3.2.18b). This term must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_V$, where $x = l$ and $n_x = 1$. For a node whose $v = 0$ at node 17,

$$n_x\left[\tilde{\nabla}\, v\right]_{\Gamma_{sV}} = 0\,. \tag{3.2.31}$$

For a node whose $v \neq 0$ at node 17, $n_x \left[\tilde{\mathcal{V}}\, v\right]_{\Gamma_{sV}}$ is not evaluated as zero unless the prescribed shear is zero. Substitution of Eq. (3.2.16b) into the term $n_x \left[\tilde{\mathcal{V}}\, v\right]_{\Gamma_{sV}}$ yields

$$n_x \left[\tilde{\mathcal{V}}\, v\right]_{\Gamma_{sV}} = \tilde{\mathcal{V}} \left( \mu_i^{(w)} \chi_i^{(w)} \Big|_{x=l} + \mu_i^{(\theta)} \chi_i^{(\theta)} \Big|_{x=l} \right) \tag{3.2.32}$$

Similarly, the term $n_x \left[\tilde{\mathcal{M}}(dv/dx)\right]_{\Gamma_{sM}}$ of Eq. (3.2.18b) must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_M$, where $x = l$ and $n_x = 1$. For a node whose $(dv/dx) = 0$ at node 17,

$$n_x \left[\tilde{\mathcal{M}}(dv/dx)\right]_{\Gamma_{sM}} = 0 . \tag{3.2.33}$$

For a node whose $(dv/dx) \neq 0$ at node 17, $n_x \left[\tilde{\mathcal{M}}(dv/dx)\right]_{\Gamma_{sM}}$ is not evaluated as zero unless the prescribed moment is zero. Substitution of Eq. (3.2.16b) into the term $n_x \left[\tilde{\mathcal{M}}(dv/dx)\right]_{\Gamma_{sM}}$ yields

$$n_x \left[\tilde{\mathcal{M}}(dv/dx)\right]_{\Gamma_{sM}} = \tilde{\mathcal{M}} \left( \mu_i^{(w)} \frac{d\chi_i^{(w)}}{dx} \Big|_{x=l} + \mu_i^{(\theta)} \frac{d\chi_i^{(\theta)}}{dx} \Big|_{x=l} \right) . \tag{3.2.34}$$

Note that the terms

$$\chi_i^{(w)} \Big|_{x=l} , \quad \chi_i^{(\theta)} \Big|_{x=l} , \quad \frac{d\chi_i^{(w)}}{dx} \Big|_{x=l} , \text{and} \frac{d\chi_i^{(\theta)}}{dx} \Big|_{x=l}$$

in Eqs. (3.2.32 and 3.2.34) are evaluated at node 17 and contribute to the $\mathbf{f}^{(\text{bdry})}$ of Eq. (3.2.17) (see Eq. 3.5.4g).

Now consider the penalty term $\alpha_w \left[(w - \tilde{w})v\right]_{\Gamma_{sw}}$ of Eq. (3.2.18c). This term must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_w$. Again, these are nodes 1, 2, and 3. The key to the contribution of each of nodes 1, 2, and 3 to the term

$\alpha_w[(w-\tilde{\boldsymbol{w}})v]_{\Gamma_{sw}}$ lies in the values of $v_1$, $v_2$, and $v_3$ at node 1, where $x = 0$. Substitution of Eq. (3.2.16a) into the penalty term yields

$$
\begin{aligned}
&\alpha_w[(w-\tilde{\boldsymbol{w}})v]_{\Gamma_{sw}} \\
&= \alpha_w \begin{bmatrix} \psi_1^{(w)} & \psi_2^{(w)} & \dots & \psi_n^{(w)} \end{bmatrix}_{x=0} \begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \\ \vdots \\ \hat{w}_n \end{bmatrix} v_i\big|_{x=0} \\
&+ \alpha_w \begin{bmatrix} \psi_1^{(\theta)} & \psi_2^{(\theta)} & \dots & \psi_n^{(\theta)} \end{bmatrix}_{x=0} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_n \end{bmatrix} v_i\big|_{x=0} \\
&- \alpha_w \tilde{\boldsymbol{w}} \cdot v_i\big|_{x=0} \quad .
\end{aligned}
\tag{3.2.35}
$$

For node 3, $v_3|_{x=0} = 0$. For node 1, $v_1|_{x=0} = 1$. The term of Eq. (3.2.35) is evaluated with each of these values. For node 2, $0 < v_2|_{x=0} < 1$, and substitution of Eq. (3.2.16b) into Eq. (3.2.35) yields

$$\alpha_w[(w-\tilde{w})v]_{\Gamma_{sw}} \tag{3.2.36}$$

$$= \alpha_w \begin{bmatrix} \psi_1^{(w)} & \psi_2^{(w)} & \dots & \psi_n^{(w)} \end{bmatrix}_{x=0} \begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \\ \vdots \\ \hat{w}_n \end{bmatrix} \left( \mu_2^{(w)} \chi_2^{(w)} \Big|_{x=0} + \mu_2^{(\theta)} \chi_2^{(\theta)} \Big|_{x=0} \right)$$

$$+ \alpha_w \begin{bmatrix} \psi_1^{(\theta)} & \psi_2^{(\theta)} & \dots & \psi_n^{(\theta)} \end{bmatrix}_{x=0} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_n \end{bmatrix} \left( \mu_2^{(w)} \chi_2^{(w)} \Big|_{x=0} + \mu_2^{(\theta)} \chi_2^{(\theta)} \Big|_{x=0} \right)$$

$$- \alpha_w \tilde{w} \left( \mu_2^{(w)} \chi_2^{(w)} \Big|_{x=0} + \mu_2^{(\theta)} \chi_2^{(\theta)} \Big|_{x=0} \right) .$$

Finally, consider the penalty term $\alpha_\theta \left[ \left( (dw/dx) - \tilde{\theta} \right) (dv/dx) \right]_{\Gamma_{s\theta}}$ of Eq. (3.2.18c). This term must be evaluated for every node in the model whose $\Omega_s$ intersects $\Gamma_\theta$. Again, these are nodes 1, 2, and 3. The key to the contribution of each of nodes 1, 2, and 3 to the term $\alpha_\theta \left[ \left( (dw/dx) - \tilde{\theta} \right) (dv/dx) \right]_{\Gamma_{s\theta}}$ lies in the values of $(dv_1/dx)$, $(dv_2/dx)$, and $(dv_3/dx)$, at node 1, where $x = 0$. Substitution of Eq. (3.2.16a) into the penalty term yields

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

$$=\alpha_\theta\left[\frac{d\psi_1^{(w)}}{dx}\quad\frac{d\psi_2^{(w)}}{dx}\quad\cdots\quad\frac{d\psi_n^{(w)}}{dx}\right]_{x=0}\begin{bmatrix}\hat{w}_1\\ \hat{w}_2\\ \vdots\\ \hat{w}_n\end{bmatrix}\frac{dv_i}{dx}\bigg|_{x=0}$$

$$+\alpha_\theta\left[\frac{d\psi_1^{(\theta)}}{dx}\quad\frac{d\psi_2^{(\theta)}}{dx}\quad\cdots\quad\frac{d\psi_n^{(\theta)}}{dx}\right]_{x=0}\begin{bmatrix}\hat{\theta}_1\\ \hat{\theta}_2\\ \vdots\\ \hat{\theta}_n\end{bmatrix}\frac{dv_i}{dx}\bigg|_{x=0} \tag{3.2.37}$$

$$-\alpha_\theta\tilde{\theta}\cdot\frac{dv_i}{dx}\bigg|_{x=0}\quad.$$

For nodes 3 and 1, $[dv_i/dx]_{x=0} = 0$ ($i$ = 1, 3), and therefore $\alpha_\theta\left|\left((dw/dx)-\tilde{\theta}\right)(dv/dx)\right|_{\Gamma_{s\theta}} = 0$. For node 2, $(dv_2/dx)$ is nonzero, and substitution of Eq. (3.2.16b) into Eq. (3.2.37) yields

$$\alpha_\theta \left[ \left( \frac{dw}{dx} - \tilde{\theta} \right) \frac{dv}{dx} \right]_{\Gamma_{s\theta}} \tag{3.2.38}$$

$$= \alpha_\theta \begin{bmatrix} \frac{d\psi_1^{(w)}}{dx} & \frac{d\psi_2^{(w)}}{dx} & \cdots & \frac{d\psi_n^{(w)}}{dx} \end{bmatrix}_{x=0} \begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \\ \vdots \\ \hat{w}_n \end{bmatrix} \left( \mu_2^{(w)} \left. \frac{d\chi_2^{(w)}}{dx} \right|_{x=0} + \mu_2^{(\theta)} \left. \frac{d\chi_2^{(\theta)}}{dx} \right|_{x=0} \right)$$

$$+ \alpha_\theta \begin{bmatrix} \frac{d\psi_1^{(\theta)}}{dx} & \frac{d\psi_2^{(\theta)}}{dx} & \cdots & \frac{d\psi_n^{(\theta)}}{dx} \end{bmatrix}_{x=0} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_n \end{bmatrix} \left( \mu_2^{(w)} \left. \frac{d\chi_2^{(w)}}{dx} \right|_{x=0} + \mu_2^{(\theta)} \left. \frac{d\chi_2^{(\theta)}}{dx} \right|_{x=0} \right)$$

$$- \alpha_\theta \tilde{\theta} \cdot \left( \mu_2^{(w)} \left. \frac{d\chi_2^{(w)}}{dx} \right|_{x=0} + \mu_2^{(\theta)} \left. \frac{d\chi_2^{(\theta)}}{dx} \right|_{x=0} \right) .$$

The terms

$$\begin{bmatrix} \frac{d\psi_1^{(w)}}{dx} & \frac{d\psi_2^{(w)}}{dx} & \cdots & \frac{d\psi_n^{(w)}}{dx} \end{bmatrix}_{x=0}$$

and

$$\begin{bmatrix} \frac{d\psi_1^{(\theta)}}{dx} & \frac{d\psi_2^{(\theta)}}{dx} & \cdots & \frac{d\psi_n^{(\theta)}}{dx} \end{bmatrix}_{x=0}$$

of Eqs. (3.2.35 – 3.2.38) contribute to the $\mathbf{K}^{(\mathrm{bdry})}$ of Eq. (3.2.17) (see Eq. 3.5.4e). The terms

$$\left. \chi_i^{(w)} \right|_{x=l}, \ \left. \chi_i^{(\theta)} \right|_{x=l}, \ \left. \frac{d\chi_i^{(w)}}{dx} \right|_{x=l}, \text{and} \left. \frac{d\chi_i^{(\theta)}}{dx} \right|_{x=l}$$

of Eqs. (3.2.35 – 3.2.38) contribute to the $\mathbf{f}^{(\mathrm{bdry})}$ of Eq. (3.2.17) (see Eq. 3.5.4g).

As discussed in Chapter 2, a proper understanding of how the terms of Eqs. (3.2.18) are calculated provides users of the MLPG method with considerable freedom in choices of nodal spacing and sizes of test functions. The value of $(R_o / l)$ may be adjusted in certain cases to account for the terms; however, in order to exploit the full usefulness of the method, the terms of Eq. (3.2.18) must be evaluated.

### 3.3 Generalized Moving Least Squares Interpolation

Recall from section 2.3 the MLS interpolation scheme for constructing trial functions for $C^0$ problems. The local MLS interpolation is written as $u(\mathbf{x}) \cong u_{\overline{\mathbf{x}}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\overline{\mathbf{x}})$ where $\mathbf{p}(\mathbf{x})$ is the basis function, and $\mathbf{a}(\overline{\mathbf{x}})$ is the vector of undetermined coefficients in the local neighborhood $\overline{\mathbf{x}}$. The values of $\mathbf{a}(\overline{\mathbf{x}})$ are found by minimizing a weighted discrete $L^2$ error norm. The 1-D shape functions resulting from this MLS interpolation scheme are $u(\mathbf{x}) \cong \sum_{j=1}^{n} \phi_j(\mathbf{x})\hat{u}_j$ . Note that only the interpolation for the primary variable, displacement, is performed.

In beam problems, both the deflection $w$ and the slope $\theta$ are the primary variables. In the FEM, the Hermite functions are used as interpolation functions for the primary variables. See Figure 3.3.1 for a comparison of the FEM Lagrangian and Hermite shape functions. The additional information (i.e., the slope) used in the Hermite shape functions must also be used in the approximation of the MLPG method. In order to accomplish this, a generalized moving least squares (GMLS) approximation is developed.

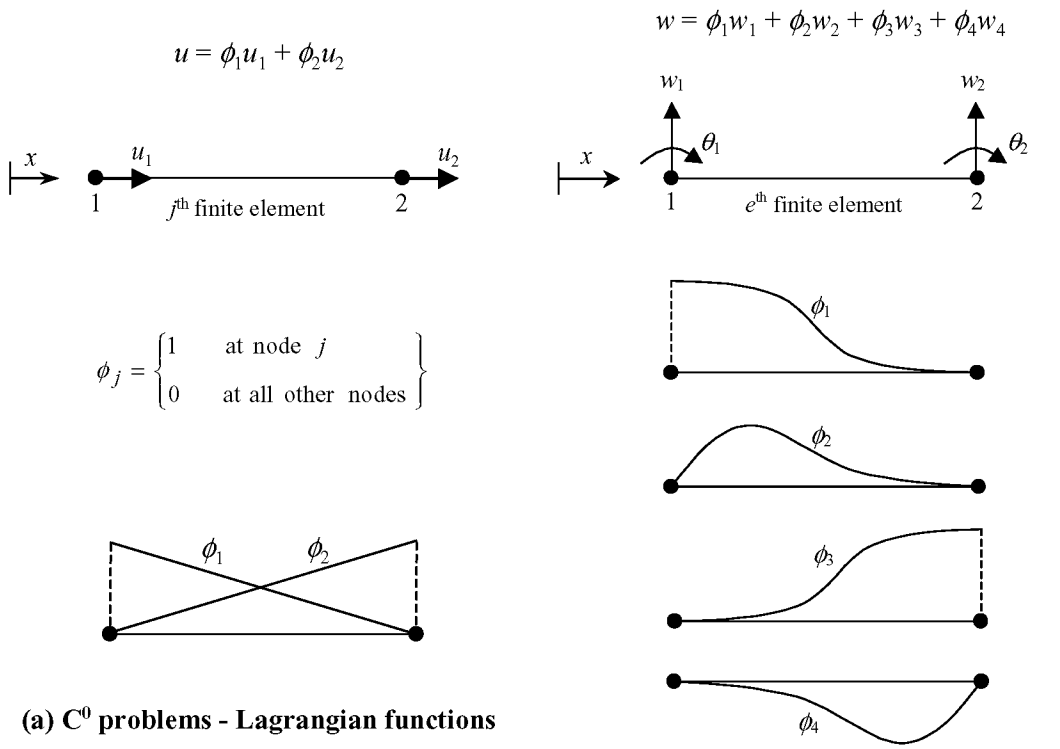



**Figure 3.3.1: Comparison of FEM shape functions for $C^0$ and $C^1$ Problems**

In this report, the spatial coordinates $y$ and $z$ of $\mathbf{x} = [x \quad y \quad z]^T$ are not present as 1-D problems are considered, and therefore $\mathbf{x} = x$. The GMLS approximation for $w$ in a global domain $\Omega$ may be written as in the MLS procedure as

$$w(x) \cong w^h(\mathrm{x}) = \mathbf{p}^T(x)\mathbf{a}(x). \tag{3.3.1a}$$

Likewise, the local GMLS approximation is written as

$$w(x) \cong w_{\bar{x}}(x) = \mathbf{p}^T(x)\mathbf{a}(\bar{x}) \tag{3.3.1b}$$

where $\bar{x} = x - x_j$ , p($x$) is the basis function, and $\mathbf{a}(\bar{x})$, the vector of undetermined coefficients, is found by minimizing a weighted discrete $H^h$ error norm (Nayroles *et al.*, 1992, Atluri *et al.*, 1999):

$$H^h(\mathbf{a}) = \sum_{j=1}^{n} \sum_{|\alpha|\le h} \lambda_j(\bar{x}) \left[ D^\alpha \mathbf{p}^{\mathrm{T}}(x_j)\mathbf{a} - D^\alpha \hat{w}_j \right]^2 \tag{3.3.2}$$

where $\lambda_j$ is a weight function, $D^\alpha$ denotes the $\alpha^{\text{th}}$ derivative, and $\sum_{|\alpha|\le h}$ indicates the summation of all derivatives up to order $h$.

For beam problems, the primary variables are the deflection $w$ and slope $\theta = (dw/dx)$, and hence, the weighted discrete $H^1$ error norm is used:

$$\begin{aligned} H^1(\mathbf{a}) &= \sum_{j=1}^{n} \sum_{|\alpha|\le 1} \lambda_j(\bar{x}) \left[ D^\alpha \mathbf{p}^{\mathrm{T}}(x_j)\mathbf{a} - D^\alpha \hat{w}_j \right]^2 \\ &= \sum_{j=1}^{n} \left\{ \lambda_j^{(w)}(\bar{x}) \left[ \mathbf{p}^{\mathrm{T}}(x_j)\mathbf{a} - \hat{w}_j \right]^2 + \lambda_j^{(\theta)}(\bar{x}) \left[ \frac{d\mathbf{p}^{\mathrm{T}}(x_j)}{dx}\mathbf{a} - \hat{\theta}_j \right]^2 \right\} \end{aligned} \tag{3.3.3}$$

In this report, $\lambda_j^{(w)}(\bar{x})$ and $\lambda_j^{(\theta)}(\bar{x})$ are chosen as the same weight functions, and will hereafter be referred to as $\lambda_j(\bar{x})$. In matrix form, Eq. (3.3.3) is

$$\begin{aligned} H^1(\mathbf{a}) &= [\mathbf{P}\mathbf{a} - \hat{\mathbf{w}}]^{\mathrm{T}} \boldsymbol{\lambda} [\mathbf{P}\mathbf{a} - \hat{\mathbf{w}}] + [\mathbf{P}_x\mathbf{a} - \hat{\mathbf{t}}]^{\mathrm{T}} \boldsymbol{\lambda} [\mathbf{P}_x\mathbf{a} - \hat{\mathbf{t}}] \\ &= \left\{ \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \end{bmatrix} \mathbf{a} - \begin{Bmatrix} \hat{\mathbf{w}} \\ \hat{\mathbf{t}} \end{Bmatrix} \right\}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{\lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \left\{ \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \end{bmatrix} \mathbf{a} - \begin{Bmatrix} \hat{\mathbf{w}} \\ \hat{\mathbf{t}} \end{Bmatrix} \right\} \\ &= [\mathbf{Q}\mathbf{a} - \hat{\mathbf{s}}]^{\mathrm{T}} \boldsymbol{\Lambda} [\mathbf{Q}\mathbf{a} - \hat{\mathbf{s}}] \end{aligned} \tag{3.3.4}$$

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

$$[\mathbf{P}] = \begin{bmatrix} \mathbf{p}^{\mathrm{T}}(x_1) & \mathbf{p}^{\mathrm{T}}(x_2) & \dots & \mathbf{p}^{\mathrm{T}}(x_n) \end{bmatrix}^{\mathrm{T}} , \tag{3.3.5a}$$

$$[\mathbf{P}_x] = \begin{bmatrix} \mathbf{p}_x^{\mathrm{T}}(x_1) & \mathbf{p}_x^{\mathrm{T}}(x_2) & \dots & \mathbf{p}_x^{\mathrm{T}}(x_n) \end{bmatrix}^{\mathrm{T}} , \tag{3.3.5b}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1(\bar{x}) & & & \\ & \lambda_2(\bar{x}) & & \\ & & \ddots & \\ & & & \lambda_n(\bar{x}) \end{bmatrix} \tag{3.3.6}$$

where

$$\mathbf{p}^{\mathrm{T}}(x) = \begin{bmatrix} 1, & x, & x^2, & \ldots & x^{m-1} \end{bmatrix}, \tag{3.3.7a}$$

$$\mathbf{p}_x^{\mathrm{T}}(x) = \frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx} = \begin{bmatrix} 0, & 1, & 2x, & \ldots & (m-1)x^{m-2} \end{bmatrix} \tag{3.3.7b}$$

with ($m$-1) as the order of the 1-D basis function $\mathbf{p}(x)$ used in the MLS approximation. Also,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{P} \\ \mathbf{P}_x \end{bmatrix}, \quad \hat{\mathbf{s}} = \begin{bmatrix} \hat{\mathbf{w}} \\ \hat{\mathbf{t}} \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \tag{3.3.8}$$

are the basis function matrix, the nodal displacement vector, and the weight function matrix, respectively. Further manipulation of Eq. (3.3.4) leads to

$$\begin{aligned} H^1 &= \left[\mathbf{a}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}} - \hat{\mathbf{s}}^{\mathrm{T}}\right]\boldsymbol{\Lambda}\left[\mathbf{Q}\mathbf{a} - \hat{\mathbf{s}}\right] \\ &= \left[\mathbf{a}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Lambda} - \hat{\mathbf{s}}^{\mathrm{T}}\boldsymbol{\Lambda}\right]\left[\mathbf{Q}\mathbf{a} - \hat{\mathbf{s}}\right] \\ &= \mathbf{a}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{Q}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Lambda}\hat{\mathbf{s}} - \hat{\mathbf{s}}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{Q}\mathbf{a} + \hat{\mathbf{s}}^{\mathrm{T}}\boldsymbol{\Lambda}\hat{\mathbf{s}} \\ &= \mathbf{a}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{Q}\mathbf{a} - 2\mathbf{a}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Lambda}\hat{\mathbf{s}} + \hat{\mathbf{s}}^{\mathrm{T}}\boldsymbol{\Lambda}\hat{\mathbf{s}}. \end{aligned} \tag{3.3.9}$$

The norm $H^1$ can be minimized using:

$$\frac{\partial H^1}{\partial a_i} = 0 \quad , \quad i = 1, 2, \ldots n\,. \tag{3.3.10}$$

Equation (3.3.10) can be rewritten as

$$\frac{\partial H^1}{\partial \mathbf{a}^{\mathrm{T}}} = 0 \quad \text{, or,}$$

$$\frac{\partial H^1}{\partial \mathbf{a}^{\mathrm{T}}} = 2\mathbf{Q}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{Q}\mathbf{a} - 2\mathbf{Q}^{\mathrm{T}}\mathbf{\Lambda}\hat{\mathbf{s}} = 0. \tag{3.3.11}$$

Equation (3.3.11) leads to

$$\underset{(m,m)}{[\mathbf{A}]}\ \underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2n,1)}{\{\hat{\mathbf{s}}\}} \tag{3.3.12}$$

where

$$\begin{aligned}\underset{(m,m)}{[\mathbf{A}]} &= \underset{(m,2n)}{\mathbf{Q}^{\mathrm{T}}}\ \underset{(2n,2n)}{\mathbf{\Lambda}}\ \underset{(2n,m)}{\mathbf{Q}} \\ &= \underset{(m,n)}{\mathbf{P}^{\mathrm{T}}}\ \underset{(n,n)}{\boldsymbol{\lambda}}\ \underset{(n,m)}{\mathbf{P}} + \underset{(m,n)}{\mathbf{P}_x^{\mathrm{T}}}\ \underset{(n,n)}{\boldsymbol{\lambda}}\ \underset{(n,m)}{\mathbf{P}_x}\end{aligned} \tag{3.3.13}$$

and

$$\underset{(m,2n)}{[\mathbf{B}]} = \underset{(m,2n)}{\mathbf{Q}^{\mathrm{T}}}\ \underset{(2n,2n)}{\mathbf{\Lambda}} = \left[\underset{(m,n)}{\mathbf{P}^{\mathrm{T}}}\ \underset{(n,n)}{\boldsymbol{\lambda}}\ ,\quad \underset{(m,n)}{\mathbf{P}_x^{\mathrm{T}}}\ \underset{(n,n)}{\boldsymbol{\lambda}}\right]. \tag{3.3.14}$$

Solving for $\{\mathbf{a}\}$ using Eq. (3.3.12) gives

$$\underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,m)}{[\mathbf{A}]^{-1}}\ \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2n,1)}{\{\hat{\mathbf{s}}\}} \tag{3.2.15}$$

Substituting into the approximation Eq. (3.3.1a)

$$w^h(x) = \underset{(1,m)}{\mathbf{p}^{\mathrm{T}}(x)}\ \underset{(m,m)}{[\mathbf{A}]^{-1}}\ \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2\mathrm{n},1)}{\{\hat{\mathbf{s}}\}}\ . \tag{3.3.16}$$

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

$$w(x) = \sum_{j=1}^{n}\left(\hat{w}_j\psi_j^{(w)}(x) + \hat{\theta}_j\psi_j^{(\theta)}(x)\right) \tag{3.3.17}$$

where

$$\psi_j^{(w)}(x) = \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}^{\mathrm{T}}\boldsymbol{\lambda}\right]_{gj}$$
$$\psi_j^{(\theta)}(x) = \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}_x^{\mathrm{T}}\boldsymbol{\lambda}\right]_{gj}. \tag{3.3.18}$$

Note that $\hat{w}_j$ and $\hat{\theta}_j$ in Eq. (3.3.17) are fictitious nodal values of deflection and slope, respectively.

As in Chapter 2, three types of weight functions $\lambda_j(\bar{x})$ are considered for constructing the trial functions: power functions,

$$\lambda_j(\bar{x}) = \begin{cases} \left[1-\left(d_j^2/R_j^2\right)\right]^{\alpha} & \text{if } 0 \le d_j \le R_j \\ 0 & \text{if } d_j > R_j, \end{cases} \tag{3.3.19}$$

with $d_j = \|x - x_j\|$, the Euclidean distance between $x$ and $x_j$, and $\alpha = 1$, 2, 3, and 4,

$$\lambda_j(\bar{x}) = \begin{cases} 1-3\left(\dfrac{d_j}{R_j}\right)^2 + 2\left(\dfrac{d_j}{R_j}\right)^3 & \text{if } 0 \le d_j \le R_j \\ 0 & \text{if } d_j > R_j, \end{cases} \tag{3.3.20}$$

a 3-term spline with $d_j = \| x - x_j \|$, and a 4-term spline,

$$\lambda_j(\bar{x}) = \begin{cases} 1-6\left(\dfrac{d_j}{R_j}\right)^2 + 8\left(\dfrac{d_j}{R_j}\right)^3 - 3\left(\dfrac{d_j}{R_j}\right)^4 & \text{if } 0 \le d_j \le R_j \\ 0 & \text{if } d_j \ge R_j \quad , \end{cases} \tag{3.3.21}$$

where $R_j$ is a user-defined parameter that controls the extents of the trial functions. These weight functions are chosen to demonstrate the robustness of the MLPG method. Consider the 17-node model presented in Figure 3.3.2.

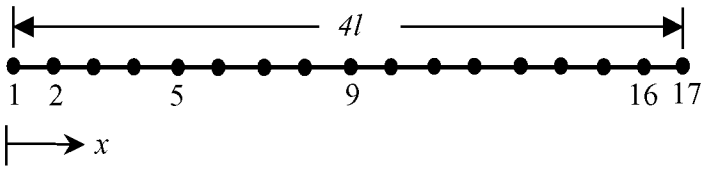
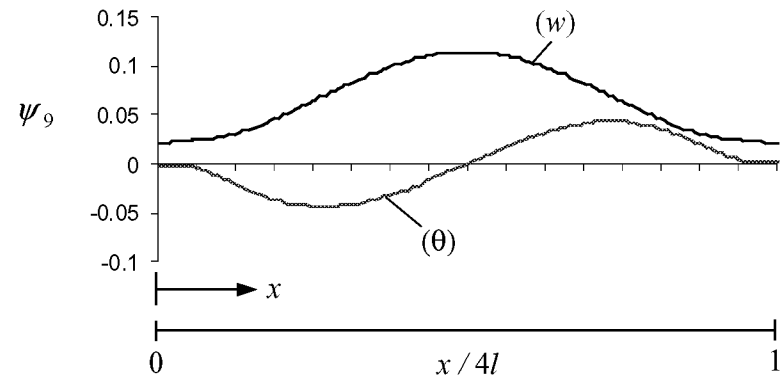
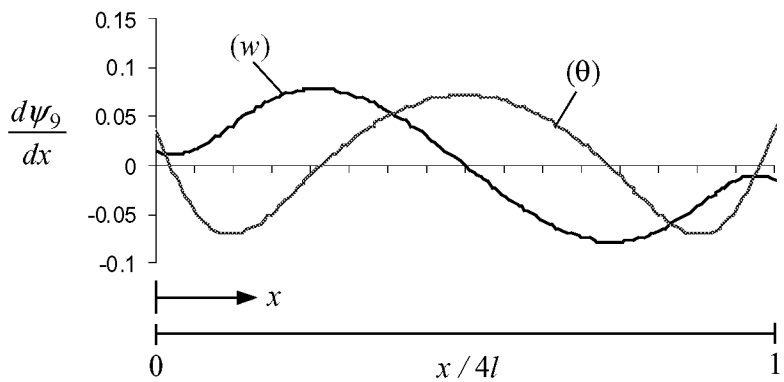



**Figure 3.3.2: A 17-node model of a beam of length 4*l***




**(a) Shape functions, $\psi_9$**




**(b) Derivatives of the shape functions, $d\psi_9/dx$**

**Figure 3.3.3: Typical shape functions and their derivatives**

Figure 3.3.3a presents $\psi_j^{(w)}$ and $\psi_j^{(\theta)}$ at node $j = 9$, typical shape functions evaluated using the weight function of Eq. (3.3.19) with $\alpha = 3$. Figure 3.3.3b presents the $(d\psi_j^{(w)}/dx)$ and the $(d\psi_j^{(\theta)}/dx)$ for node $j = 9$. These functions were evaluated using a quartic basis function. The value of $(R_j/l)$ was chosen as $(R_j/l) = 3.5$. From these plots,

it is seen that while $\theta = dw/dx$, $\psi^{(\theta)} \neq (d\psi^{(w)}/dx)$. This is because the derivative of $\psi^{(w)}$ involves the inverse of the **[A]** matrix. One should note that, while $\psi^{(\theta)} \neq (d\psi^{(w)}/dx)$, the basis function used for $\psi^{(\theta)}$ must be the derivative of the basis function used for $\psi^{(w)}$. For example, if a quadratic basis $(1, x, x^2)$ is used for $\psi^{(w)}$, then $(0, 1, 2x)$ must be used for $\psi^{(\theta)}$. These are important characteristics of the MLPG method for Euler-Bernoulli beam problems.

### 3.4 Test Functions Used

The MLPG equations are derived using the weak form of the governing equation. Recall the weak form from section 3.2:

$$0 = EI \int_{\Omega_s} \frac{d^2 w}{dx^2}\frac{d^2 v}{dx^2} dx - \int_{\Omega_s} f\, v\, dx + \alpha_w \left[(w - \tilde{w})v\right]_{\Gamma_{sw}} + \alpha_\theta \left[\left(\frac{dw}{dx} - \tilde{\theta}\right)\frac{dv}{dx}\right]_{\Gamma_{s\theta}}$$
$$- n_x \left[\tilde{V}\, v\right]_{\Gamma_{sV}} - n_x \left[\tilde{M}\frac{dv}{dx}\right]_{\Gamma_{sM}} + n_x \left[EI\frac{d^3 w}{dx^3} v\right]_{\Gamma_{sw}} - n_x \left[EI\frac{d^2 w}{dx^2}\frac{dv}{dx}\right]_{\Gamma_{s\theta}} . \tag{3.4.1}$$

Also recall Eq. (3.3.17), in which the trial function $w$ is approximated as

$$w(x) = \sum_{j=1}^{n} \left(\hat{w}_j \psi_j^{(w)}(x) + \hat{\theta}_j \psi_j^{(\theta)}(x)\right) \tag{3.4.2}$$

where $\hat{w}_j$ and $\hat{\theta}_j$ are the fictitious nodal deflections and slopes of the trial function, and $\psi^{(w)}$ and $\psi^{(\theta)}$ are their corresponding shape functions, respectively, given by Eqs. (3.3.18). The test function, $v$, is approximated using

$$v(x) = \mu_i^{(w)} \chi_i^{(w)}(x) + \mu_i^{(\theta)} \chi_i^{(\theta)}(x) \tag{3.4.3}$$

where $\mu_i^{(w)}$ and $\mu_i^{(\theta)}$ are the arbitrary constants for deflections and slopes of the test function, and $\chi_i^{(w)}(x)$ and $\chi_i^{(\theta)}(x)$ are components of the the Petrov-Galerkin test functions that are chosen from a different space than $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$. Recall that the expressions for the shape functions are written as in Eq. (3.3.18) as

$$\begin{aligned} \psi_j^{(w)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}^{\mathrm{T}}\lambda\right]_{gj} \\ \psi_j^{(\theta)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}_x^{\mathrm{T}}\lambda\right]_{gj}. \end{aligned} \tag{3.4.4}$$

In a Galerkin approximation, the components of the test functions $\chi_i^{(w)}(x)$ and $\chi_i^{(\theta)}(x)$ would take on the same form as the shape functions:

$$\begin{aligned} \chi_j^{(w)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}^{\mathrm{T}}\,\boldsymbol{\lambda}\right]_{gj} \\ \chi_j^{(\theta)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}_x^{\mathrm{T}}\,\boldsymbol{\lambda}\right]_{gj}. \end{aligned} \tag{3.4.5}$$

As mentioned previously, Atluri *et al.* (1999) used the Galerkin approximation of Eq. (3.4.5). However, in this work, a Petrov-Galerkin approximation is used and the components of the test functions $\chi_i^{(w)}(x)$ and $\chi_i^{(\theta)}(x)$ are chosen from a different space than the shape functions $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$. The Petrov-Galerkin formulation is further discussed in section 3.6. The test function components $\chi_i^{(w)}(x)$ are chosen as simple weight functions similar to those of Eqs. (3.3.19 – 3.3.21) as

$$\chi_i^{(w)}(x) = \begin{cases} \left[ 1 - \left( d_i^2 / R_o^2 \right) \right]^{\beta} & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if} \quad d_i > R_o \end{cases} \tag{3.4.6}$$

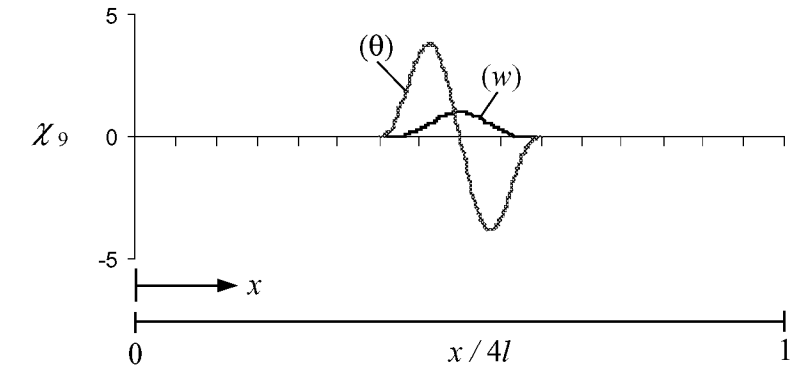
with $d_i = \|x - x_i\|$ and $\beta = 2$, 3, and 4,

$$\chi_i^{(w)}(x) = \begin{cases} 1 - 3\left(\frac{d_i}{R_o}\right)^2 + 2\left(\frac{d_i}{R_o}\right)^3 & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if } d_i > R_o \end{cases} \tag{3.4.7}$$

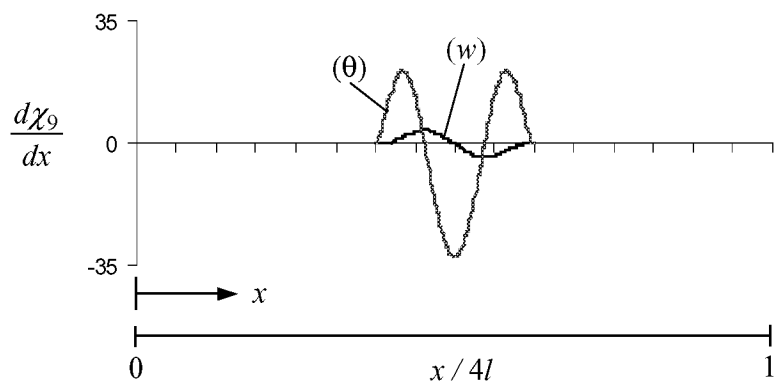
with $d_i = \|x - x_i\|$, and

$$\chi_i^{(w)}(x) = \begin{cases} 1 - 6\left(\frac{d_i}{R_o}\right)^2 + 8\left(\frac{d_i}{R_o}\right)^3 - 3\left(\frac{d_i}{R_o}\right)^4 & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if} \quad d_i > R_o. \end{cases} \tag{3.4.8}$$

where $R_o$ is a user-defined parameter. Plots of the components of the test functions $\chi_i^{(w)}(x)$ and $\chi_i^{(\theta)}(x)$ of Eq. (3.4.6) with $\beta = 4$ for node 9 of a 17-node model of a beam with $(R_o / l) = 2\Delta x$ are shown in Figure 3.4.1a. The corresponding derivatives, $(d\chi_i^{(w)} / dx)$ and $(d\chi_i^{(\theta)} / dx)$ are shown in Figure 3.4.1b. Note that for the test functions, $\chi_i^{(\theta)} = (d\chi_i^{(w)} / dx)$.




**(a) Test functions, $\chi_9$**




**(b) Derivatives of the test functions, $d\chi_9 / dx$**

**Figure 3.4.1: Typical test function components and their derivatives**

### 3.5 Development of the MLPG Equations

To evaluate the integrands and the terms involved in the weak form, the derivatives of $w$ and $v$, the trial and test functions, are needed. Since $\hat{w}_j$, $\hat{\theta}_j$, $\mu_i^{(w)}$, and $\mu_i^{(\theta)}$ are constant values, the derivatives are carried out over $\psi^{(w)}$, $\psi^{(\theta)}$, $\chi^{(w)}$, and $\chi^{(\theta)}$ as

$$\frac{dw}{dx} = \sum_{j=1}^{n} \left( \hat{w}_j \frac{d\psi_j^{(w)}}{dx} + \hat{\theta}_j \frac{d\psi_j^{(\theta)}}{dx} \right)$$

$$\frac{d^2 w}{dx^2} = \sum_{j=1}^{n} \left( \hat{w}_j \frac{d^2\psi_j^{(w)}}{dx^2} + \hat{\theta}_j \frac{d^2\psi_j^{(\theta)}}{dx^2} \right) \tag{3.5.1a}$$

$$\frac{d^3 w}{dx^3} = \sum_{j=1}^{n} \left( \hat{w}_j \frac{d^3\psi_j^{(w)}}{dx^3} + \hat{\theta}_j \frac{d^3\psi_j^{(\theta)}}{dx^3} \right)$$

and

$$\frac{dv}{dx} = \mu_i^{(w)} \frac{d\chi_i^{(w)}}{dx} + \mu_i^{(\theta)} \frac{d\chi_i^{(\theta)}}{dx}$$

$$\frac{d^2 v}{dx^2} = \mu_i^{(w)} \frac{d^2\chi_i^{(w)}}{dx^2} + \mu_i^{(\theta)} \frac{d^2\chi_i^{(\theta)}}{dx^2} \tag{3.5.1b}$$

Appendix A presents explicit expressions for all the derivatives of $\psi^{(w)}$ and $\psi^{(\theta)}$.

Substitution of Eqs. (3.3.17, 3.4.3, and 3.5.1) into the weak form leads to

$$
\begin{aligned}
0 = & \int_{\Omega_s} EI \sum_{j=1}^{n} \left( \hat{w}_j \frac{d^2 \psi_j^{(w)}}{dx^2} + \hat{\theta}_j \frac{d^2 \psi_j^{(\theta)}}{dx^2} \right) \left( \mu_i^{(w)} \frac{d^2 \chi_i^{(w)}}{dx^2} + \mu_i^{(\theta)} \frac{d^2 \chi_i^{(\theta)}}{dx^2} \right) dx \\
& - \int_{\Omega_s} f \left( \mu_i^{(w)} \chi_i^{(w)} + \mu_i^{(\theta)} \chi_i^{(\theta)} \right) dx \\
& + \alpha_w \left[ \sum_{j=1}^{n} \left( \hat{w}_j \psi_j^{(w)} + \hat{\theta}_j \psi_j^{(\theta)} - \tilde{w} \right) \left( \mu_i^{(w)} \chi_i^{(w)} + \mu_i^{(\theta)} \chi_i^{(\theta)} \right) \right]_{\Gamma_{sw}} \\
& + \alpha_\theta \left[ \sum_{j=1}^{n} \left( \hat{w}_j \frac{d\psi_j^{(w)}}{dx} + \hat{\theta}_j \frac{d\psi_j^{(\theta)}}{dx} - \tilde{\theta} \right) \left( \mu_i^{(w)} \frac{d\chi_i^{(w)}}{dx} + \mu_i^{(\theta)} \frac{d\chi_i^{(\theta)}}{dx} \right) \right]_{\Gamma_{s\theta}} \\
& - n_x \left[ \tilde{V} \left( \mu_i^{(w)} \chi_i^{(w)} + \mu_i^{(\theta)} \chi_i^{(\theta)} \right) \right]_{\Gamma_{sV}} \\
& - n_x \left[ \tilde{M} \left( \mu_i^{(w)} \frac{d\chi_i^{(w)}}{dx} + \mu_i^{(\theta)} \frac{d\chi_i^{(\theta)}}{dx} \right) \right]_{\Gamma_{sM}} \\
& + n_x \left[ EI \sum_{j=1}^{n} \left( \hat{w}_j \frac{d^3 \psi_j^{(w)}}{dx^3} + \hat{\theta}_j \frac{d^3 \psi_j^{(\theta)}}{dx^3} \right) \left( \mu_i^{(w)} \chi_i^{(w)} + \mu_i^{(\theta)} \chi_i^{(\theta)} \right) \right]_{\Gamma_{sw}} \qquad (3.5.2) \\
& - n_x \left[ EI \sum_{j=1}^{n} \left( \hat{w}_j \frac{d^2 \psi_j^{(w)}}{dx^2} + \hat{\theta}_j \frac{d^2 \psi_j^{(\theta)}}{dx^2} \right) \left( \mu_i^{(w)} \frac{d\chi_i^{(w)}}{dx} + \mu_i^{(\theta)} \frac{d\chi_i^{(\theta)}}{dx} \right) \right]_{\Gamma_{s\theta}}
\end{aligned}
$$

$$(i = 1, 2, \ldots n).$$

Requiring that Eq. (3.5.2) be valid for arbitrary values of $\mu_i^{(w)}$ and $\mu_i^{(\theta)}$ leads to the MLPG equations as

$$\mathbf{K}^{(\text{node})} \hat{\mathbf{d}} + \mathbf{K}^{(\text{bdry})} \hat{\mathbf{d}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = \mathbf{0} \qquad (3.5.3)$$

where “bdry” denotes boundary and

$$\hat{\mathbf{d}} = \left\{ \hat{w}^1, \quad \hat{\theta}^1, \quad \hat{w}^2, \quad \hat{\theta}^2, \quad \ldots \right\}^{\mathrm{T}} \tag{3.5.4a}$$

are the fictitious nodal values of deflections $w$ and slopes $\theta$, and

$$\mathbf{K}^{(\mathrm{node})} = \left[ \mathbf{k}_{ij}^{(\mathrm{node})} \right] \tag{3.5.4b}$$

$$\mathbf{K}^{(\mathrm{bdry})} = \left[ \mathbf{k}_{ij}^{(\mathrm{bdry})} \right] \tag{3.5.4c}$$

with

$$\mathbf{k}_{ij}^{(\mathrm{node})} = EI \begin{bmatrix} \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2 \chi_i^{(w)}}{dx^2} \frac{d^2 \psi_j^{(w)}}{dx^2} dx & \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2 \chi_i^{(w)}}{dx^2} \frac{d^2 \psi_j^{(\theta)}}{dx^2} dx \\ \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2 \chi_i^{(\theta)}}{dx^2} \frac{d^2 \psi_j^{(w)}}{dx^2} dx & \displaystyle\int_{\Omega_s^{(i)}} \frac{d^2 \chi_i^{(\theta)}}{dx^2} \frac{d^2 \psi_j^{(\theta)}}{dx^2} dx \end{bmatrix} \tag{3.5.4d}$$

$$
\begin{aligned}
\mathbf{k}_{ij}^{(\mathrm{bdry})} &= \alpha_w \begin{bmatrix} \chi_i^{(w)}\psi_j^{(w)} & \chi_i^{(w)}\psi_j^{(\theta)} \\ \chi_i^{(\theta)}\psi_j^{(w)} & \chi_i^{(\theta)}\psi_j^{(\theta)} \end{bmatrix}_{\Gamma_{sw}^{(i)}} \\
&+ n_x EI \begin{bmatrix} \chi_i^{(w)} \dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(w)} \dfrac{d^3\psi_j^{(\theta)}}{dx^3} \\ \chi_i^{(\theta)} \dfrac{d^3\psi_j^{(w)}}{dx^3} & \chi_i^{(\theta)} \dfrac{d^3\psi_j^{(\theta)}}{dx^3} \end{bmatrix}_{\Gamma_{sw}^{(i)}} \\
&+ \alpha_\theta \begin{bmatrix} \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d\psi_j^{(w)}}{dx} & \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d\psi_j^{(\theta)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d\psi_j^{(w)}}{dx} & \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d\psi_j^{(\theta)}}{dx} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}} \\
&- n_x EI \begin{bmatrix} \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(w)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2} \\ \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(w)}}{dx^2} & \dfrac{d\chi_i^{(\theta)}}{dx}\dfrac{d^2\psi_j^{(\theta)}}{dx^2} \end{bmatrix}_{\Gamma_{s\theta}^{(i)}}
\end{aligned}
\tag{3.5.4e}
$$

$$
\mathbf{f}^{(\mathrm{node})} = \begin{Bmatrix} \displaystyle\int_{\Omega_s^{(i)}} \chi_i^{(w)} f \, dx \\ \displaystyle\int_{\Omega_s^{(i)}} \chi_i^{(\theta)} f \, dx \end{Bmatrix}
\tag{3.5.4f}
$$

$$
\begin{aligned}
\mathbf{f}^{(\mathrm{bdry})} &= n_x \tilde{\mathcal{M}} \begin{Bmatrix} \dfrac{d\chi_i^{(w)}}{dx} \\ \dfrac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{sM}^i} + n_x \tilde{\mathcal{V}} \begin{Bmatrix} \chi_i^{(w)} \\ \chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sV}^i} \\
&+ \alpha_w \begin{Bmatrix} \tilde{w}\chi_i^{(w)} \\ \tilde{w}\chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sw}^i} + \alpha_\theta \begin{Bmatrix} \tilde{\theta}\dfrac{d\chi_i^{(w)}}{dx} \\ \tilde{\theta}\dfrac{d\chi_i^{(\theta)}}{dx} \end{Bmatrix}_{\Gamma_{s\theta}^i} ,
\end{aligned}
\tag{3.5.4g}
$$

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

After these equations are solved for the fictitious nodal values $\hat{w}_j$ and $\hat{\theta}_j$, the interpolated primary and secondary variables may be computed. The deflection, $w$, at any point in the beam is calculated from Eq. (3.3.17),

$$w(x) = \sum_{j=1}^{n} \left( \hat{w}_j \psi_j^{(w)}(x) + \hat{\theta}_j \psi_j^{(\theta)}(x) \right) \quad , \tag{3.5.5}$$

and post processing is accomplished by either of the methods discussed in section 2.5. The slope $\theta$, moment $M$, and shear $V$ can just as easily be calculated from Eqs. (3.5.1a).

### 3.6 The Petrov-Galerkin Formulation

As stated in section 3.2, the test functions, $v$, of the LWF are chosen based on the weighted residual (W-R) method being used. Two prominent W-R methods, namely the Galerkin and Petrov-Galerkin methods, and their application to beam problems will now be discussed.

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

upon where the support domains ended within the $\Omega_s$. In each of these subregions, a 10-point Gaussian quadrature was used to integrate the weak from accurately.

In the current work, the Petrov-Galerkin method is used, i.e., the test functions $\chi_i$ are chosen to be distinctly different from the shape functions $\psi_j$ $\left(\chi_j \neq \psi_j\right)$. Recall the weight functions chosen as test function components in Eqs. (3.4.6 – 3.4.8), repeated here for convenience.

$$\chi_i^{(w)}(x) = \begin{cases} \left[1 - \left(d_i^2 / R_o^2\right)\right]^\beta & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if} \quad d_i > R_o \end{cases} \tag{3.6.1}$$

with $d_i = \|x - x_i\|$ and $\beta = 2$, 3, and 4,

$$\chi_i^{(w)}(x) = \begin{cases} 1 - 3\left(\dfrac{d_i}{R_o}\right)^2 + 2\left(\dfrac{d_i}{R_o}\right)^3 & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if } d_i > R_o, \end{cases} \tag{3.6.2}$$

and

$$\chi_i^{(w)}(x) = \begin{cases} 1 - 6\left(\dfrac{d_i}{R_o}\right)^2 + 8\left(\dfrac{d_i}{R_o}\right)^3 - 3\left(\dfrac{d_i}{R_o}\right)^4 & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if } d_i > R_o, \end{cases} \tag{3.6.3}$$

where $R_o$ is a user-defined parameter.

The derivatives of the test functions can be evaluated at the center ($d_i/R_o = 0$) and at the end points ($d_i/R_o = 1$) as

$$\begin{aligned} &\frac{\partial^{m_0}}{\partial x^{m_0}} \chi_i^{(w)}\left(\frac{d_i}{R_o} = 0\right) = 0 \quad ; \quad m_0 \ge 1 \quad \text{and} \\ &\frac{\partial^{m_1}}{\partial x^{m_1}} \chi_i^{(w)}\left(\frac{d_i}{R_o} = 1\right) = 0 \quad ; \quad m_1 \ge 1. \end{aligned} \tag{3.6.4}$$

The test functions are then $C^{\gamma}$ continuous up to the order $\gamma$ where $\gamma = \min(m_0, m_1)$ (see Atluri and Shen, 2002). With these definitions, the test functions from Eq. (3.6.1) with $\beta = 2$, 3, and 4 are $C^1$, $C^1$, and $C^3$ continuous, respectively. Similarly, the spline functions from Eqs. (3.6.2 and 3.6.3) are $C^1$ continuous. As pointed out previously, the lengths $R_j$ and $R_o$ in Eqs. (3.3.19 – 3.3.21 and 3.6.1 – 3.6.3) are user-controlled parameters in the numerical implementation of the MLPG method.

To evaluate the validity of the MLPG method and the usefulness of each of the trial and test functions derived from Eqs. (3.3.19 – 3.3.21 and 3.6.1 – 3.6.3), the MLPG method is applied to various patch tests and mixed boundary value beam problems in Chapter 4.

### 3.7 Concluding Remarks

The MLPG formulation was developed for bending of beams – $C^1$ problems. A local weak form (LWF) was developed from the classical weighted-residual form of the fourth order governing differential equation. The moving least squares interpolation scheme was generalized to include derivatives. These generalized moving least squares (GMLS) approximations were used as the trial functions. The test functions were chosen from a different space than the trial functions as combinations of simple weight functions and their derivatives. This choice of test functions makes the method a Petrov-Galerkin method. Substitution of these trial and test functions into the LWF yielded a system of algebraic equations. Stiffness matrices were found to be unsymmetric and banded. The continuity of the test functions was also discussed. In Chapter 4, several numerical

examples are considered to study the effectiveness of the MLPG method for beam problems.

## Chapter 4: Numerical Examples

Several numerical examples are used to study the effectiveness of the MLPG method for beam problems. For the examples presented, a beam of constant flexural rigidity $EI$ and a length of $4l$ is considered. The length $4l$ was specifically chosen to avoid scaling by unity. Six models with 5, 9, 17, 33, 65, and 129 nodes uniformly distributed along the length of the beam are considered. Figure 4.0.1 shows a typical 17-node model.

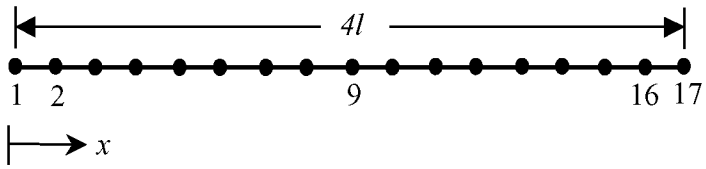



**Figure 4.0.1: A 17- node model of the beam**

The distances between the nodes ($\Delta x/l$) in these models are 1.0, 0.5, 0.25, 0.125, 0.0625, and 0.03125 for the 5-, 9-, 17-, 33-, 65-, and 129-node models, respectively. Three types of basis functions, quadratic basis ($1, x, x^2$), cubic basis ($1, x, x^2, x^3$), and quartic basis ($1, x, x^2, x^3, x^4$) are used. As mentioned in Chapter 3, the system matrices for the MLPG algorithm are of the form (Eq. 3.5.3):

$$\mathbf{K}^{(\text{node})}\hat{\mathbf{d}} + \mathbf{K}^{(\text{bdry})}\hat{\mathbf{d}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = \mathbf{0}\,, \tag{4.0.1}$$

where the superscript "bdry" denotes boundary. These matrices are developed with the previously mentioned parameters. Problems studied in this chapter include both patch test and mixed boundary value problems. First, simple patch test problems are studied wherein a local coordinate approach is developed to improve the accuracy of the method. Error norms of the patch tests for both the global and local methods are compared to demonstrate the validity of the local approach. Next, general rules of thumb for choosing

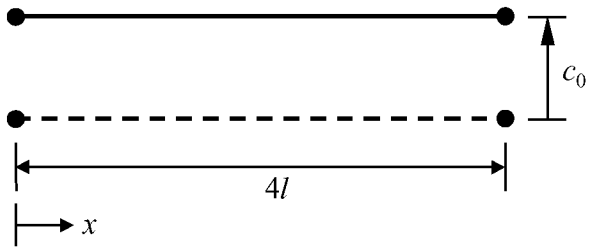
the various user-defined problem parameters are discussed. Then, several mixed boundary value problems are worked. Finally, the method is extended to continuous beams, and an example problem is studied. As will be demonstrated, the MLPG method for beam problems yields excellent results for both primary and secondary variables without the need for elaborate post-processing techniques.

### 4.1 Patch Tests

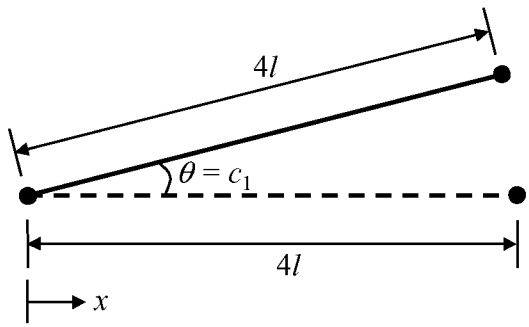
The MLPG formulation for $C^1$ problems was evaluated by applying the formulation to simple patch-test problems. The problems considered were

$$\begin{array}{llll} 1. & w(x) = c_0, & \theta = \dfrac{dw}{dx} = 0; & \text{Rigid body translation} \\ 2. & w(x) = c_1 x, & \theta = c_1; & \text{Rigid body rotation} \\ 3. & w(x) = c_2 x^2/2, & \theta = c_2 x; & \text{Constant - curvature condition} \end{array} \tag{4.1.1}$$

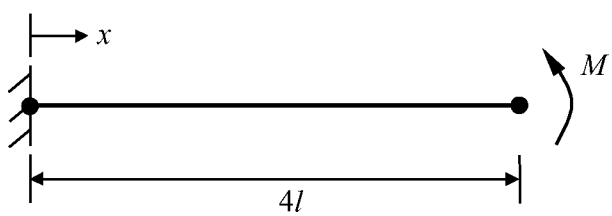
where $c_0$, $c_1$, and $c_2$ are arbitrary constants. The third patch test could be looked upon as the problem of a cantilever beam with a moment, $M=EI(d^2w/dx^2)= EIc_2$, applied at $x=4l$. These three problems are depicted in Figure 4.1.1. All three of these problems satisfy the governing differential equation exactly and as such represent three simple exact solution problems. The deflection $w$ and the slope $\theta$ corresponding to problems 1, 2, and 3 were prescribed as essential boundary conditions (EBCs) at $x=0$ and $x=4l$. With these EBCs, the beam problem was solved using the MLPG method. If the MLPG method recovers the exact solution at all the interior nodes and at every arbitrary point of the beam, then the MLPG method passes the patch test.




**(a) Rigid body translation**




**(b) Rigid body rotation**




**(c) Constant-curvature condition**

**Figure 4.1.1: Patch tests for beam problems**

In preliminary evaluations, the $\chi_i^{(w)}(x)$ test function component in the MLPG weak form was chosen as

$$\chi_i^{(w)}(x) = \begin{cases} \left[ 1 - \left( d_i^2 / R_o^2 \right) \right]^4 & \text{if} \quad 0 \le d_i \le R_o \\ 0 & \text{if} \quad d_i > R_o, \end{cases} \tag{4.1.2}$$

where $d_i = \| x - x_i \|$. The weight functions $\lambda_j(\bar{x})$ used to construct the trial functions were

$$\lambda_j(\bar{x}) = \begin{cases} \left[ 1 - \left( d_j^2 / R_j^2 \right) \right]^4 & \text{if} \quad 0 \le d_j \le R_j \\ 0 & \text{if} \quad d_j > R_j, \end{cases} \tag{4.1.3a}$$

and

$$\lambda_j(\bar{x}) = \begin{cases} 1 - 6\left(\dfrac{d_j}{R_j}\right)^2 + 8\left(\dfrac{d_j}{R_j}\right)^3 - 3\left(\dfrac{d_j}{R_j}\right)^4 & \text{if} \quad 0 \le d_j \le R_j \\ 0 & \text{if } d_j > R_j, \end{cases} \tag{4.1.3b}$$

where $d_j = \| x - x_j \|$. (Recall from section 3.3 that $\bar{x} = x - x_j$ is used in the GMLS approximation to construct the trial functions in the local neighborhood $\bar{x}$ of $x$. Thus, $d_j = \|x - x_j\|$ could also be written as $d_j = \|\bar{x}\|$.) The term $(R_o / l)$ in the test functions (Eq. 4.1.2) in each of these six models was different and chosen equal to $(2\Delta x)$. The $(R_j / l)$ in Eqs. (4.1.3) were chosen to be $(R_j / l = 3.5)$ for the 5-, 9-, and 17- node models and $(R_j / l = 16\Delta x)$ for the 33-, 65-, and 129- node models. The ranges of $(R_o / l)$ and $(R_j / l)$ are discussed later in this chapter, in section 4.4.

The displacement vectors $\{\mathbf{d}\}$ that correspond to each of the conditions in Eq. (4.1.1) (and in the absence of any other loading) when used in Eq. (4.0.1) should result in a null right-hand vector if the $\mathbf{K}^{(\text{node})}$ is evaluated exactly. In general, the product results in a residual $\{\mathbf{r}\}$ vector as

$$\mathbf{K}^{(\text{node})}\{\mathbf{d}\} = \{\mathbf{r}\}. \tag{4.1.4}$$

Each of the components of the vector $\{\mathbf{r}\}$ is nearly equal to machine zero if $\mathbf{K}^{(\text{node})}$ is evaluated accurately. To quantify the residual, an error norm of $\{\mathbf{r}\}$ is computed as

$$\|E\|_1 = \sqrt{\frac{1}{N_d}\sum_{k=1}^{N_d} r_k^2}\ . \tag{4.1.5}$$

where $r_k$ is the $k^{\text{th}}$ component of the vector $\{\mathbf{r}\}$ in Eq. (4.1.4), and $N_d$ is the degrees of freedom in the model.

**Table 4.1.1: Error norm $\|E\|_1$ of the residuals for six models and for two basis functions**

| Number of nodes in the model | $w=c_0$ | | $w=c_1x$ | | $w=c_2x^2/2$ | |
|---|---|---|---|---|---|---|
| | Quadratic Basis | Cubic Basis | Quadratic Basis | Cubic Basis | Quadratic Basis | Cubic Basis |
| 5* | 0.5040e-14 | 0.1278e-12 | 0.2099e-14 | 0.4547e-13 | 0.5733e-14 | 0.9196e-13 |
| 9* | 0.7515e-13 | 0.1496e-11 | 0.2362e-13 | 0.5514e-12 | 0.3321e-13 | 0.9747e-12 |
| 17* | 0.2774e-10 | 0.8211e-10 | 0.1109e-10 | 0.3067e-10 | 0.1582e-10 | 0.5352e-10 |
| 33 | 0.3609e-9 | 0.1062e-5 | 0.1266e-9 | 0.4479e-6 | 0.2587e-10 | 0.9057e-6 |
| 65 | 0.1691e-6 | 0.1435e-2 | 0.7735e-7 | 0.5855e-3 | 0.1726e-6 | 0.1193e-2 |
| 129 | 0.1796e-4 | 0.5599e+0 | 0.8154e-5 | 0.2269e+0 | 0.1794e-4 | 0.4154e+0 |

* $R_j / l = 3.5$

Table 4.1.1 presents the error norm $\|E\|_1$ for the three conditions in Eq. (4.1.1) when the weight function in Eq. (4.1.3b) was used. (Similar results were obtained when the weight function in Eq. (4.1.3a) was used.) As seen from the table, the error norm $\|E\|_1$ deteriorates with model refinement and for higher order basis. Closer examination of the residuals for each of the six models showed that the residuals were of machine accuracy for nodes near the origin while the residuals were largest at nodes farthest from the origin. This observation was confirmed by running different cases with the origin at different locations along the length of the beam. The origin was moved to the center of the beam so that the domain $\Omega$ became $-2l \le x \le 2l$. The computed error norms $\|E\|_1$ of Table 4.1.1 were best at the center of the beam (at $x = 0$) and inferior at the two endpoints $x = -2l$ and $x = 2l$. The origin was then moved to the right end of the beam, i.e., $-4l \le x \le 0$. The error norms $\|E\|_1$ were found to be best at the right end of the beam $(x = 0)$ and inferior at the left end of the beam ($x = -4l$). In fact, the same error norms of

0.1794e-4 and 0.4154e+0 for the 129-node model and $w = c_2x^2/2$ were observed at $(x = -4l)$ when the origin was moved to the right end of the beam. Also, the residuals were largest for the models with the largest number of nodes. This behavior is counter-intuitive.

Closer scrutiny of computations showed that the numerical values of the shape functions for nodes that are systematically located about the center of the beam (for example, nodes 3 and 15, 2 and 16, and 1 and 17 in the 17-node model of Figure 4.0.1) are not exactly identical as expected. These differences increased with model refinement and when a higher order basis was used. The reason for this behavior is explained in section 4.2.

### 4.2 Local Coordinate Approach

In the MLS interpolation, the basis functions are in terms of the global coordinate $x$. The [**A**] matrix thus formed using this basis is generally of the form (see Atluri *et al.*, 1999, Eq. 16)

$$[\mathbf{A}] = \sum_{k=1}^{M} \left\{ \lambda_k(\bar{x})\, \mathbf{p} \cdot \mathbf{p}^{\mathrm{T}} + \lambda_k(\bar{x})\, \mathbf{p}_x \cdot \mathbf{p}_x^{\mathrm{T}} \right\} \tag{4.2.1}$$

where $\bar{x} = x - x_j$, and $M$ are the number of nodes in the domain of definition of node $j$ for which the [**A**] matrix is being computed. (For convenience in presentation, the [**A**] matrices thus formed will be referred to as the global method.) As the order of the polynomial basis increases, the conditioning of the [**A**] matrix deteriorates. For example, the matrix [**A**] will have terms like $1, x^2, x^4, x^6$ on the diagonal for a cubic basis function. The [**A**] matrices for nodes near the origin and the [**A**] matrices for nodes farthest from

the origin will be different. The conditioning is worse for [**A**] matrices for nodes farthest from the origin. This explains the differences in the error norms observed in Table 4.1.1. The error norms in Table 4.1.1 can be improved by using higher precision computations or inversion routines. However, a much simpler alternative to improve the accuracy is discussed.

The conditioning of the [**A**] matrix can be considerably improved if the MLS approximation is defined not in terms of a global basis, but rather in terms of a local basis. Figure 4.2.1 shows two identical shape functions, one centered at node *j*, and the other centered at node *e*.

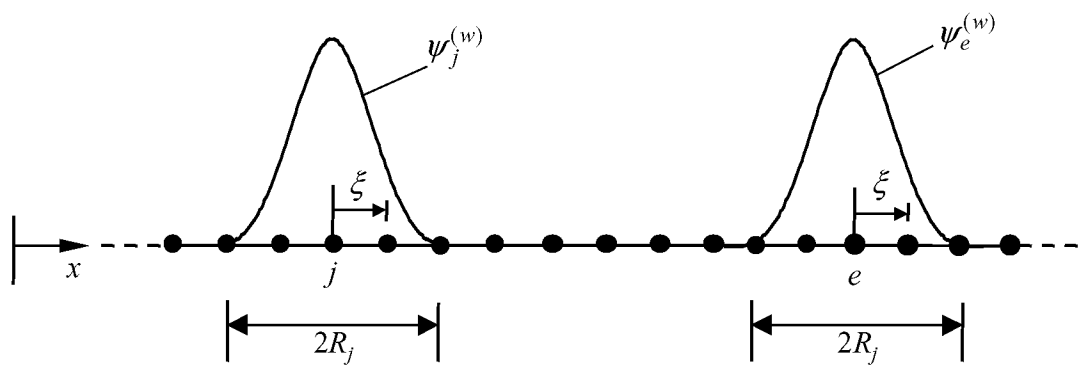



**Figure 4.2.1: Local coordinate definitions**

The global approximation for

$$\begin{aligned} w(x) &= \mathbf{p}^{\mathrm{T}}(x)\mathbf{a}(x) \\ &= a_1 + a_2 x + a_3 x^2 + \ldots + a_m x^{m-1} \end{aligned} \tag{4.2.2}$$

can be rewritten in the neighborhood of node *j*, recognizing that $x = x_j + \xi$ where $\xi$ is a local coordinate measured from node *j,* as

$$
\begin{aligned}
w(x) &= a_1 + a_2\left(x_j + \xi\right) + a_3\left(x_j + \xi\right)^2 + \ldots \\
&= \left(a_1 + a_2 x_j + a_3 x_j^2 + \ldots\right) + \left(a_2 + 2a_3 x_j + \ldots\right)\xi + \left(a_3 + \ldots\right)\xi^2 \\
&= b_1 + b_2\xi + b_3\xi^2 + \ldots + b_m\xi^{m-1}
\end{aligned}
\tag{4.2.3}
$$

where $b_i$, $i = 1, 2, \ldots, m$ are the new undetermined coefficients in the MLS approximation. (A similar local coordinate transformation can be affected for node $e$ in Figure 4.2.1 as $x = x_e + \xi$.) The [**A**] matrix then is computed in a similar manner as in Eq. (4.2.1), but with

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

and $$\mathbf{p}_x^{\mathrm{T}}(\xi) = \left[0, \quad 1, \quad 2\xi, \quad 3\xi^2, \quad \ldots \quad (m-1)\xi^{(m-2)}\right] \tag{4.2.4}$$

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

The local coordinate approach was implemented in the evaluation of the shape functions and their derivatives for all the nodes in the six MLPG models of the beam. Table 4.2.1 compares the condition numbers of the [**A**] matrices at various locations on the beam using global and local coordinate methods. The condition numbers were evaluated using routines available in NAPACK and the procedure outlined in Bathe, 1996 and Chapra and Canale, 1988. A brief review of condition numbers is presented in Appendix B. When the global coordinate method was used, the condition numbers of the [**A**] matrices for nodes farthest from the origin were much larger (suggesting poor conditioning) than the nodes closest to the origin. The conditioning numbers of the [**A**] matrices vastly improved when the local coordinate method was used, clearly demonstrating the advantages of the local coordinate method.

**Table 4.2.1: Comparison of the condition numbers of the [A] matrices at various locations on the beam using global and local coordinate methods**

| Location on the beam ($x/4l$) | Number of nodes in the model | | | | | |
|---|---|---|---|---|---|---|
| | 5* | 9* | 17* | 33 | 65 | 129 |
| | Global Method Conditioning Number | | | | | |
| 0.0 | 0.631e+3 | 0.106e+4 | 0.930e+3 | 0.271e+3 | 0.267e+3 | 0.189e+4 |
| 0.5 | 0.231e+5 | 0.268e+5 | 0.272e+5 | 0.785e+5 | 0.904e+6 | 0.131e+8 |
| 1.0 | 0.914e+6 | 0.771e+6 | 0.127e+7 | 0.422e+8 | 0.153e+10 | 0.365e+11 |
| | Local Method Conditioning Number | | | | | |
| 0.0 | 0.634e+3 | 0.106e+4 | 0.930e+3 | 0.271e+3 | 0.267e+3 | 0.189e+4 |
| 0.5 | 0.478e+3 | 0.496e+2 | 0.411e+2 | 0.111e+2 | 0.153e+2 | 0.141e+3 |
| 1.0 | 0.632e+3 | 0.106e+4 | 0.930e+3 | 0.271e+3 | 0.267e+3 | 0.189e+4 |

* $R_j / l = 3.5$

The error norms shown in Table 4.1.1 were recomputed and the results are presented in Table 4.2.2. As expected, all models and the quadratic and cubic basis functions produced error norms close to machine accuracy, suggesting that the local coordinate approach produces a significant increase in accuracy compared to the global coordinate approach.

**Table 4.2.2: Error norm $\|E\|_1$ of the residuals computed with the local coordinate approach**

| Number of nodes in the model | $w=c_0$ | | $w=c_1x$ | | $w=c_2x^2/2$ | |
|---|---|---|---|---|---|---|
| | Quadratic Basis | Cubic Basis | Quadratic Basis | Cubic Basis | Quadratic Basis | Cubic Basis |
| 5* | 0.1173e-14 | 0.3500e-13 | 0.2342e-15 | 0.1201e-14 | 0.3174e-14 | 0.3853e-13 |
| 9* | 0.2521e-13 | 0.4900e-13 | 0.8357e-14 | 0.1699e-13 | 0.3659e-13 | 0.4146e-13 |
| 17* | 0.1392e-12 | 0.2169e-12 | 0.4764e-13 | 0.1680e-12 | 0.2126e-12 | 0.8124e-12 |
| 33 | 0.4389e-12 | 0.1390e-11 | 0.1876e-12 | 0.5060e-12 | 0.4084e-12 | 0.2183e-11 |
| 65 | 0.4196e-11 | 0.3890e-11 | 0.1142e-11 | 0.1879e-11 | 0.2548e-11 | 0.5930e-11 |
| 129 | 0.4029e-10 | 0.2778e-10 | 0.1240e-10 | 0.8191e-11 | 0.2400e-10 | 0.2166e-10 |

* $R_j / l = 3.5$

In the global MLPG implementation, the [**A**] matrix is calculated and inverted at every node in the model. When using the local coordinate methodology with uniform nodal spacing, the shape functions are exactly identical for nodes whose $R_j$ places the entire shape function in the interior of the domain of the problem. Hence, for those nodes the [**A**] matrices are identical. As such, considerable reduction in computational effort

and cost may be achieved by the proposed local coordinate approach. The local coordinate approach is used to evaluate the shape functions and their derivatives for all numerical examples in the remainder of this report.

### 4.3 Patch Tests Revisited

The three patch test problems,

$$\begin{array}{llll} 1. & w(x) = c_0, & \theta = \dfrac{dw}{dx} = 0; & \text{Rigid body translation} \\ 2. & w(x) = c_1 x, & \theta = c_1; & \text{Rigid body rotation} \\ 3. & w(x) = c_2 x^2/2, & \theta = c_2 x; & \text{Constant - curvature condition} \end{array} \tag{4.3.1}$$

outlined in section 4.1 are now studied.

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

$$\begin{array}{llll} \text{Translation:} & & \text{Rotation:} & \\ w|_{x=0} = c_0 & w/_{x=4l} = c_0 & w/_{x=0} = 0 & w/_{x=4l} = 4c_1 l \\ \theta|_{x=0} = 0 & \theta|_{x=4l} = 0 & \theta|_{x=0} = c_1 & \theta|_{x=4l} = c_1 \end{array} \tag{4.3.2}$$

Since the exact solutions are constant and linear in $x$, respectively, the MLPG method developed with a quadratic or higher order basis function must reproduce the solutions exactly. (A linear basis cannot be used as the LWF requires second derivatives of the trial functions.) As expected, the algorithm reproduces the exact solutions for $w$ and $\theta$ to machine accuracy for both rigid body modes at the nodes and at any arbitrary point in the beam.

For the constant – curvature condition, $w = c_2 x^2/2$, the problem was modeled with EBCs

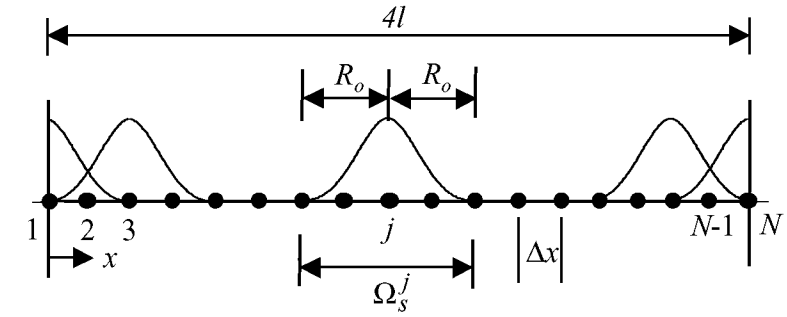
$$w|_{x=0} = 0 \qquad w|_{x=4l} = 8c_2\, l^2 \tag{4.3.3}$$
$$\theta|_{x=0} = 0 \qquad \theta|_{x=4l} = 4c_2\, l$$

Since the exact solution is quadratic in $x$, the MLPG method developed with a quadratic or higher order basis function must reproduce the solution exactly. As expected, the algorithm reproduced the exact solution for the primary variables to machine accuracy at all nodes and at any arbitrary point in the beam.

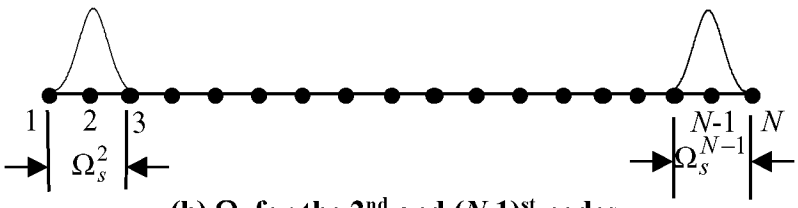
The above analyses were repeated with each of the test function components in Eqs. (3.4.6 – 3.4.8). The MLPG method reproduced exact solutions to machine accuracy, thus passing all the patch tests.

### 4.4 Problem Parameters

As mentioned previously, the parameters $(R_o / l)$ and $(R_j / l)$ in the MLPG method are user-controlled. Ranges of values of these parameters were studied, and a general rule of thumb was established. The previously mentioned lengths $(R_o / l = 2\Delta x)$ and $(R_j / l = 8\Delta x)$ were used at all nodes of an $N$-node model, except at node 2 and node $N$-1 (see Figure 4.4.1). For these nodes, $(R_o / l = \Delta x)$, was used to ensure a symmetric $\Omega_s$ and account for the terms of Eqs. (3.2.18) where, with $(R_o / l = 2\Delta x)$ for nodes 2 and $N$-1, $0 < v_2 < 1$ and $0 < v_{N-1} < 1$. Note that with these assignments of $(R_o / l)$ the test functions for all interior nodes have symmetric $\Omega_s$ configurations. As shown in Figure 4.4.1, no asymmetry is introduced at nodes 1 and $N$ as exactly half of their test functions are used. When these symmetries are violated, the MLPG method requires additional terms as discussed in section 3.2. When these terms are accounted for, the MLPG method passes the patch tests.



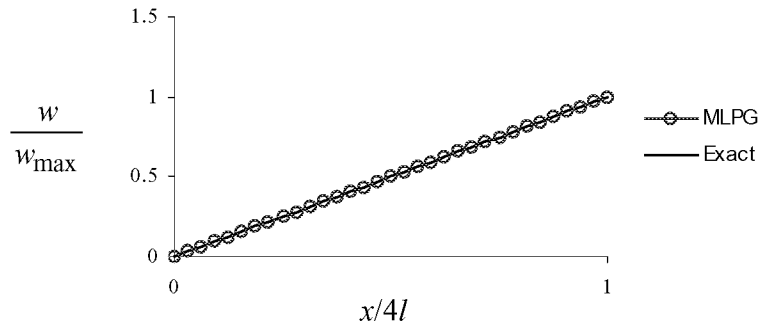

**(a) $\Omega_s$ for the 1st, 3rd, $j$th, ($N$-2)th, and $N$th nodes**



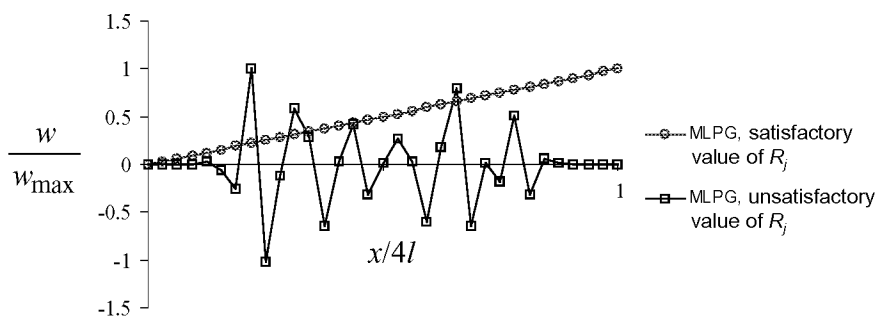

**(b) $\Omega_s$ for the 2nd and ($N$-1)st nodes**

**Figure 4.4.1: Local sub-domain, $\Omega_s$, definitions for various nodes**

As the models are refined, the value of ($\Delta x$ /$l$) decreases and thus the size of $\Omega_s$ and the extent of the trial functions also decrease. For finer models, i.e. for the 33-, 65-, and 129-node models, when $8\Delta x \le R_j / l \le 16\Delta x$ the MLPG method yielded very accurate results. However, when $R_j / l > 16\Delta x$ the MLPG method failed the patch tests for these models. Figure 4.4.2 shows the results of the rigid body rotation problem for these two cases. When $R_j / l > 16\Delta x$ the trial function is too diffused and the size of $\Omega_s$ ($R_o / l = 2\Delta x$) is too small in comparison to ($R_j / l$). The combination of small $\Omega_s$ size and large ($R_j / l$) are apparently incompatible. While the finer models performed well over a large range of ($R_j / l$), the coarser models performed well in a much smaller range of ($R_j / l$). For good performance, ($R_j / l$) needed to be approximately $8\Delta x$ but less than 98% of the total beam length.




**(a) MLPG and exact solutions when $8\Delta x \le R_j/l \le 16\Delta x$**




**(b) MLPG solutions when $R_j/l > 16\Delta x$**

**Figure 4.4.2: Rigid body rotation - Comparison of results for different extents of trial functions**

## 4.5 Mixed Boundary Value Problems

The MLPG method was applied to beam problems with mixed boundary conditions. Recall from section 3.3 that the prescription of displacement and shear and slope and moment are mutually disjoint. When displacement is prescribed, the shear cannot be prescribed. Likewise, when slope is prescribed, the moment cannot be prescribed.

4.5.1 Cantilever beam with concentrated moment at the free end

The first problem considered was a cantilever beam with a concentrated moment at the free end (i.e. $M = M_0$ at $x = 4l$, see Figure 4.5.1).

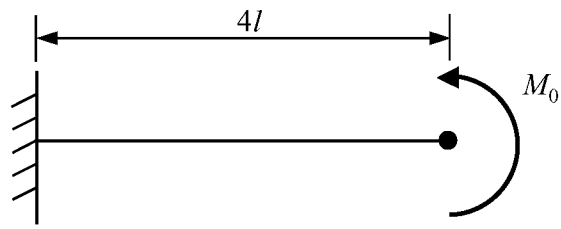



**Figure 4.5.1: Cantilever beam with concentrated moment at the free end**

The exact solution for this problem is $w = M_0x^2 / 2EI$ and $\theta = M_0x / EI$. For all trial functions considered, the MLPG algorithm reproduced the exact solution when the test function components in Eq. (3.4.6) with $\beta = 3$ and 4 and when the 4-term spline function in Eq. (3.4.8) were used. In contrast, the algorithm failed to reproduce the exact solution when the test function component in Eq. (3.4.6) with $\beta = 2$ and the 3-term spline function of Eq. (3.4.7) were used. This example suggests that $\chi_i^{(w)}$ test function components with at least $C^1$ continuity and with $m_1 \geq 2$ (see Eq. 3.6.4) are required for the MLPG algorithm for beam problems.

4.5.2 Cantilever beam with tip load

The second problem considered was a cantilever beam with a tip load (See Figure 4.5.2). Since the exact solution for this problem is cubic in terms of the $x$-coordinate of the beam, all six models with a cubic basis function and a test function with $C^1$ continuity and with $m_1 \geq 2$ reproduced the exact solution to machine accuracy.

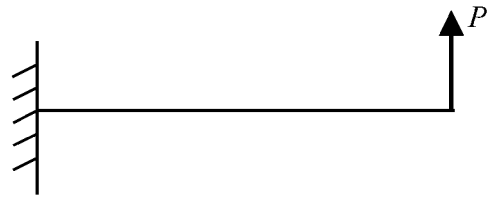



**Figure 4.5.2: Cantilever beam with tip load**

4.5.3 Simply supported beam subjected to uniformly distributed load

The third problem considered was a simply supported beam subjected to a uniformly distributed load (see Figure 4.5.3).

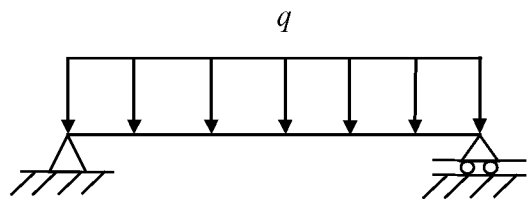



**Figure 4.5.3: Simply supported beam subjected to a uniformly distributed load**

The exact solution for this problem is given by

$$w = \frac{q}{24EI}\left(-2Lx^3 + x^4 + L^3 x\right) \quad , \quad \frac{dw}{dx} = \frac{q}{24EI}\left(-6Lx^2 + 4x^3 + L^3\right) \tag{4.5.1}$$

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

$$\begin{aligned} \|E_w\|_2 &= \sqrt{\frac{1}{g}\sum_{k=1}^{g}\left[\frac{(w_{MLPG} - w_{exact})}{w_{exact}}\right]_k^2} \\ \|E_M\|_2 &= \sqrt{\frac{1}{g}\sum_{k=1}^{g}\left[\frac{(M_{MLPG} - M_{exact})}{M_{exact}}\right]_k^2} \end{aligned} \tag{4.5.2}$$

were computed at $g$ uniformly spaced points along the beam. A value of $g = 200$ was used. The norms $\|E_w\|_2$ and $\|E_M\|_2$ are presented in Table 4.5.1. As expected, all models ($N \geq 9$) yielded accurate solutions (within 4% for $w$ and $M$). As the number of nodes in the models were increased, the $\|E_w\|_2$ norm changed marginally, suggesting the same accuracy in the solutions for the various models. Also, the $\|E_M\|_2$ norm was of the same order as the $\|E_w\|_2$ norm, suggesting the same accuracy for the primary and the secondary variables. To obtain acceptable results using a Galerkin formulation (Atluri *et al.*, 1999), $\Omega_s$ would have to be subdivided into sub-regions within which, for example, a 10-point Gaussian quadrature would be used to perform the integrations (see section 3.6). The numbers in Table 4.5.1 were computed via a Petrov-Galerkin formulation using a 20-point Gaussian integration in each of the single compact support domains $\Omega_s$. When the order of the basis function was increased to quartic, the MLPG method reproduced the exact solutions (for $w$, $\theta$, $M$, and $V$) to machine accuracy.

**Table 4.5.1: Error norm $\|E\|_2$ for a simply supported beam subjected to a uniformly distributed load with cubic basis used in the MLPG method. (Trial function using Eq. (3.3.19) with $\alpha$=3 and test function using Eq.(3.4.6) with $\beta$=4.)**

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

* $R_j / l = 3.5$, † $R_j / l = 8\Delta x$

The previously discussed problem demonstrates an interesting phenomenon. When the order of the basis function equals the order of the exact solution, the previously discussed 8-point Gaussian quadrature in a single $\Omega_s$ is sufficient to integrate the weak form very accurately. However, when the order of the basis function is less than the order of the exact solution, a higher order integration rule (such as a 20-point Gaussian integration) is needed to obtain accurate results. For problems with complicated loading,

where exact solutions are not known, the order of the basis function can easily be increased until convergence of the solution is achieved.

The problem of the simply supported beam subjected to a uniformly distributed load was modeled next using the full beam with non-uniform nodal spacing shown in Figure 4.5.4.

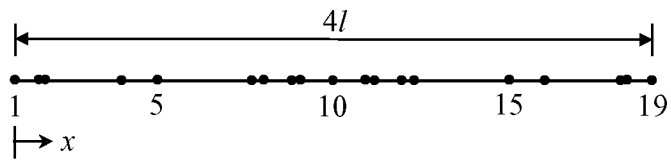
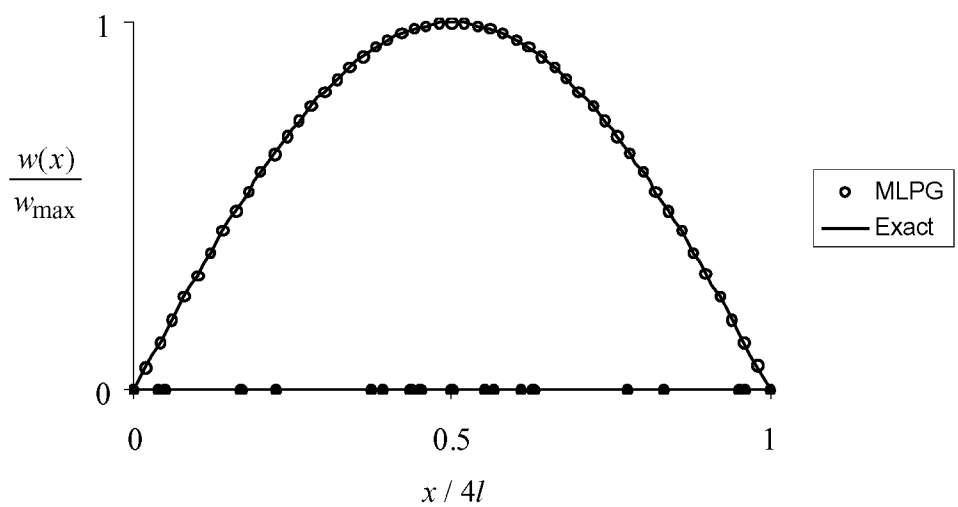


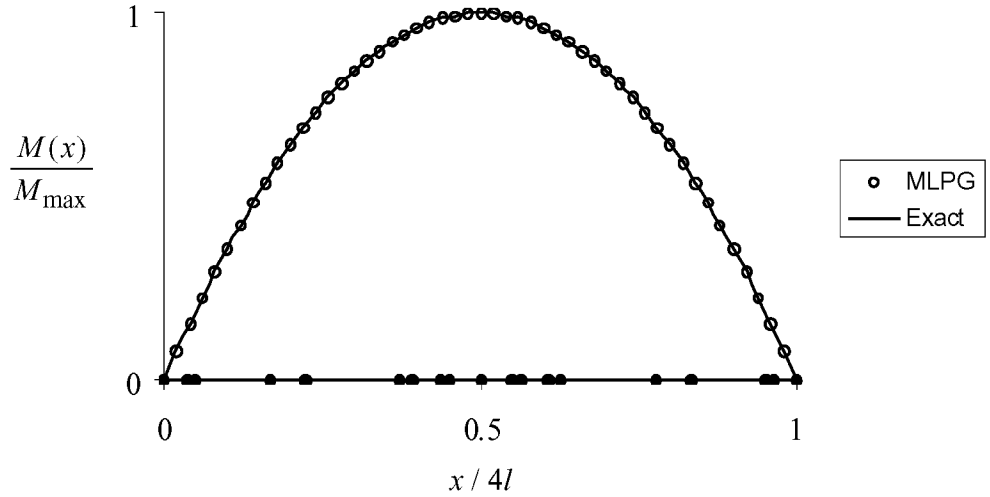

**Figure 4.5.4: A 19-node model with unequally spaced nodes**

This model was generated by randomly placing nodes in the region $0 < x < 2l$ and symmetrically replicating these nodes in the region $2l < x < 4l$. The order of the basis function was increased to quartic. The MLPG and exact solutions for deflection, moment, and shear are presented in Figure 4.5.5.

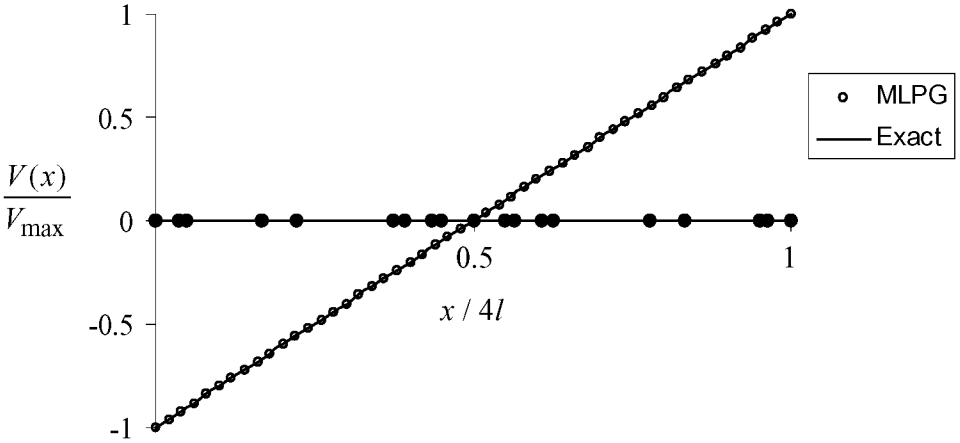



**(a) Deflection**

**Figure 4.5.5: MLPG and exact solutions for a simply supported beam subjected to a uniformly distributed load**




**(b) Moment**




**(c) Shear**

**Figure 4.5.5 Concluded: MLPG and exact solutions for a simply supported beam subjected to a uniformly distributed load**

As expected, the MLPG method reproduced the exact solutions to machine accuracy for both the primary and secondary variables despite the nodal arrangement.

4.5.4 Simply supported beam subjected to a central concentrated load

The fourth problem considered was a simply supported beam subjected to a central concentrated load (see Figure 4.5.6).

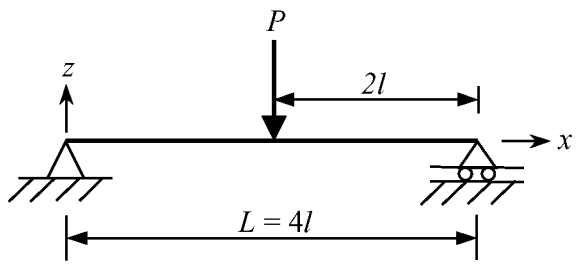



**Figure 4.5.6: Simply supported beam subjected to a central concentrated load**

The exact solution for this problem is

$$EIw = -\frac{P}{12}x^3 + \frac{PL^2}{16}x$$
$$EI\frac{dw}{dx} = EI\theta = -\frac{P}{4}x^2 + \frac{PL^2}{16} \qquad \text{for } 0 \le x \le \frac{L}{2} \qquad (4.5.3)$$

and

$$EIw = \frac{P}{12}x^3 - \frac{PL}{4}x^2 + \frac{3PL^2}{16}x - \frac{PL^3}{48}$$
$$EI\frac{dw}{dx} = EI\theta = \frac{P}{4}x^2 - \frac{PL}{2}x + \frac{3PL^2}{16} \qquad \text{for } \frac{L}{2} \le x \le L \qquad (4.5.4)$$

where $L = 4l$. The problem was analyzed in two different ways. First, symmetry was used, and one-half of the beam was considered. Second, the full beam was modeled without the use of symmetry.

For both representations of the beam, the problem was worked using a quartic basis function, and a 20-point Gaussian integration was used to develop the system of equations. The weight function of Eq. (3.3.19) with $\alpha = 3$ was used to construct the trial functions, and Eq. (3.4.6) with $\beta = 4$ was used for the test functions. The value of $(R_o / l)$ was chosen as $(R_o / l) = 2\Delta x$ for all nodes. For the symmetric representation of the beam,

$(R_j / l)$ was chosen as $(R_j / l) = 8\Delta x$ for the 33-, 65-, and 129-node models, and as $(R_j / l) = 3.5$ for the 5-, 9-, and 17-node models. For the full representation of the beam, the values of $(R_j / l)$ are noted in Table 4.5.2. The exact solution for the deflection under the load is given by

$$w_{\max} = \frac{PL^3}{48EI} = \frac{P(4l)^3}{48EI}. \tag{4.5.5}$$

The exact solutions for the slopes at the end points are given by

$$\theta\big|_{x=0} = \frac{PL^2}{16EI} = \frac{P(4l)^2}{16EI} \quad \text{and} \quad \theta\big|_{x=4l} = -\frac{PL^2}{16EI} = -\frac{P(4l)^2}{16EI}. \tag{4.5.6}$$

For the symmetric representation of the beam, the boundary conditions shown in Figure 4.5.7 were used. As expected, the MLPG method reproduced the exact solutions for all models at all nodes and at all interior points of the beam.

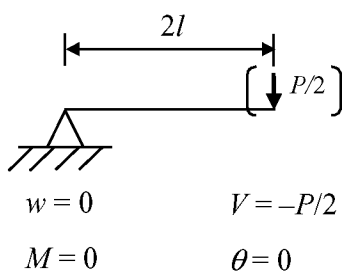



**Figure 4.5.7: Symmetric representation of a simply supported beam subjected to a central concentrated load**

The concentrated load at the center of the beam is expected to cause difficulty when the full beam is considered. As such the full beam is modeled to study the performance of the MLPG method. External loads contribute to the $\mathbf{f}^{(\text{node})}$ of Eq. (4.0.1) (see Eq. 3.5.4f), repeated here:

$$\mathbf{f}^{(\text{node})} = \begin{Bmatrix} \int_{\Omega_s^{(i)}} \chi_i^{(w)} f \, dx \\ \int_{\Omega_s^{(i)}} \chi_i^{(\theta)} f \, dx \end{Bmatrix} . \tag{4.5.7}$$

In numerical implementation, if a concentrated load $P$ is applied at node $p$, the integrals of Eq. (4.5.7) are evaluated with the dirac delta function as

$$\mathbf{f}^{(\text{node})} = \begin{Bmatrix} \int_{\Omega_s^{(i)}} \chi_i^{(w)} f \, dx \\ \int_{\Omega_s^{(i)}} \chi_i^{(\theta)} f \, dx \end{Bmatrix} = \begin{Bmatrix} \int_{\Omega_s^{(i)}} \chi_i^{(w)} f \, \delta(x = x_p) \, dx \\ \int_{\Omega_s^{(i)}} \chi_i^{(\theta)} f \, \delta(x = x_p) \, dx \end{Bmatrix} = \begin{Bmatrix} \chi_i^{(w)}(x_p) P \\ \chi_i^{(\theta)}(x_p) P \end{Bmatrix} . \tag{4.5.8}$$

To evaluate the $\mathbf{f}^{(\text{node})}$, all the nodes in the domain of influence of node $p$ need to be examined. The value of each test function, $v_i$, in the domain of influence of node $p$ is evaluated at node $p$. As the values of the test functions, $v_i$, are computed as (see Eq. 3.4.3)

$$v_i(x_p) = \mu_i^{(w)} \chi_i^{(w)}(x_p) + \mu_i^{(\theta)} \chi_i^{(\theta)}(x_p) , \tag{4.5.9}$$

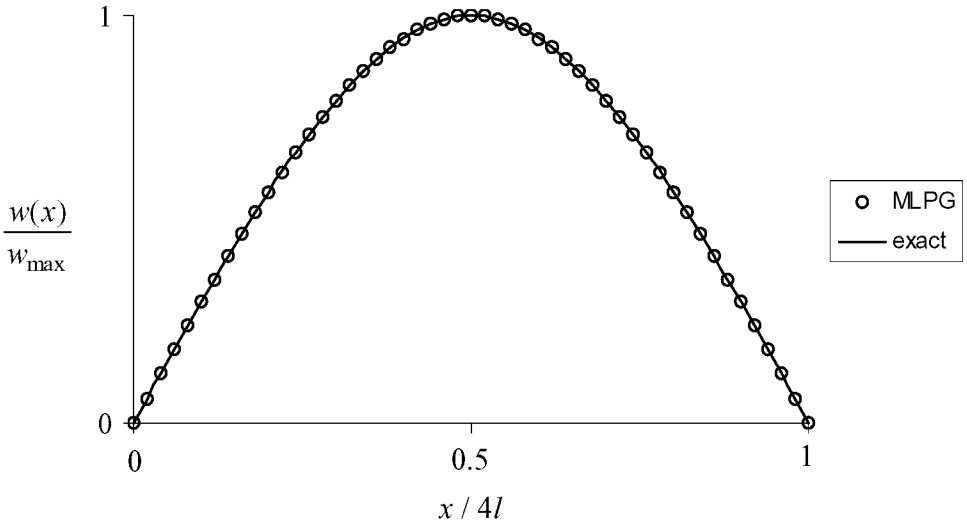
the corresponding $\chi_i^{(w)}$ and $\chi_i^{(\theta)}$ contribute to the $\mathbf{f}^{(\text{node})}$ as in Eq. (4.5.8).

For the full representation of the beam, the MLPG values of $w_{\max}$ and $\theta_{\max}$ for each of the models studied are presented in Table 4.5.2.

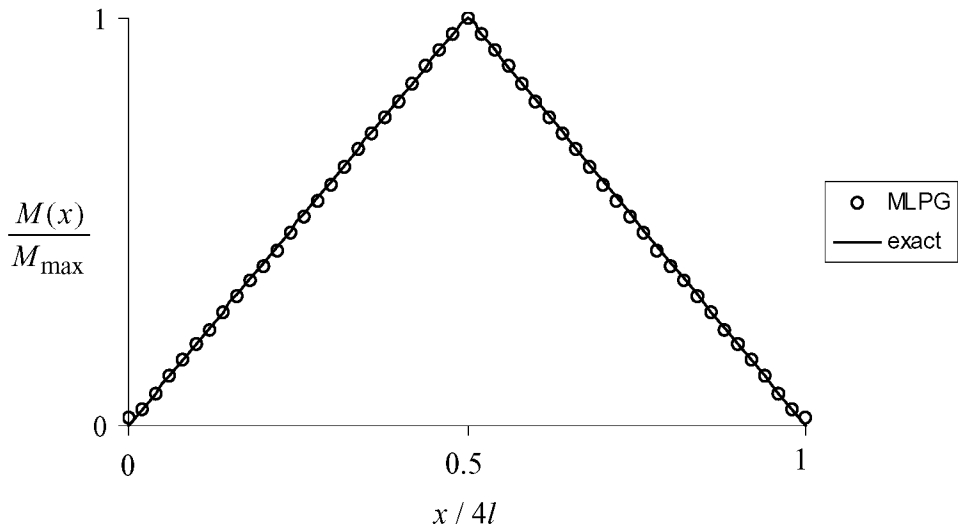
**Table 4.5.2: MLPG values of deflection and slope for models with various nodal arrangements**

| | Number of nodes in the model; ($R_j$ / $l$) | | | | | |
|---|---|---|---|---|---|---|
| | 5; ($3\Delta x$) | 9; ($4\Delta x$) | 17; ($6\Delta x$) | 33; ($8\Delta x$) | 65; ($8\Delta x$) | 129; ($8\Delta x$) |
| $w_{(\max)} / w_{(\max)\text{exact}}$ | 0.9746 | 1.0882 | 1.0368 | 0.9982 | 0.9992 | 1.0120 |
| $\theta_{(\max)} / \theta_{(\max)\text{exact}}$ | 0.3717 | 1.1003 | 1.0380 | 1.0012 | 0.9975 | 1.0126 |

The MLPG and exact solutions for deflection and moment of the 65-node model are compared in Figures 4.5.8.




**(a) Deflection**




**(b) Moment**

**Figure 4.5.8: MLPG and exact solutions for a simply supported beam with a central concentrated load**

These figures and the results presented in Table 4.5.2 demonstrate that the MLPG method yields excellent results for both primary and secondary variables. These results were obtained without the use of elaborate post-processing techniques. As the number of

nodes was increased from 17 to 129, the accuracy of the solutions did not appreciably change, suggesting that a 17-node or 33-node model is sufficient to obtain an accurate solution. The MLPG method apparently handled the discontinuity caused by the central concentrated load well.

### 4.6 Continuous Beams

The MLPG method was then applied to a continuous beam problem to evaluate its effectiveness. A continuous beam with one additional support along the interior of the beam (shown in Figure 4.6.1) is considered.

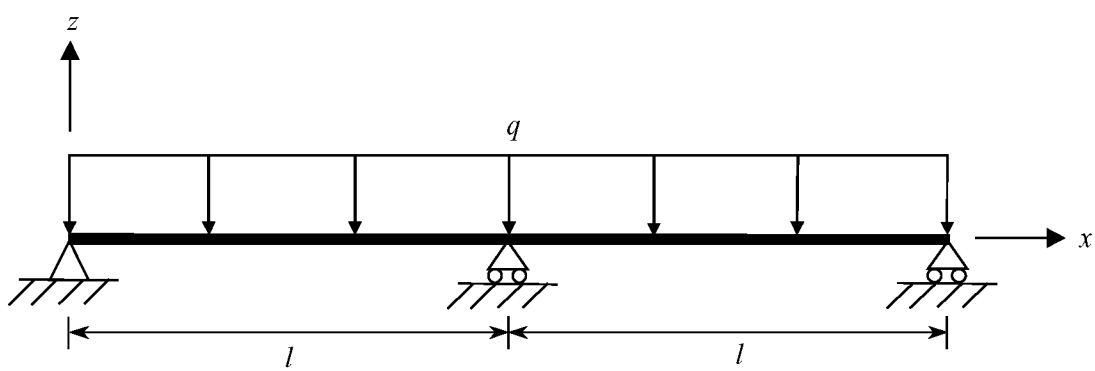



**Figure 4.6.1: Continuous beam subjected to a uniformly distributed load**

In applying the MLPG method to continuous beams, an additional penalty term,

$$\alpha_c[(w-\tilde{w})v]_{\Gamma_c}\,, \tag{4.6.1}$$

where $\alpha_c$ is the penalty parameter to enforce the continuous beam boundary condition, is added to the weak form. The weak form of the governing differential equation then becomes

$$0 = EI \int_{\Omega} \frac{d^2 w}{dx^2} \frac{d^2 v}{dx^2} dx - \int_{\Omega} f\, v\, dx + \alpha_w [(w - \tilde{w}) v]_{\Gamma_w} + \alpha_\theta \left[ \left( \frac{dw}{dx} - \tilde{\theta} \right) \frac{dv}{dx} \right]_{\Gamma_\theta}$$
$$+ \alpha_c [(w - \tilde{w}) v]_{\Gamma_c} + n_x \left[ v \cdot EI \frac{d^3 w}{dx^3} \right]_{\Gamma} - n_x \left[ \frac{dv}{dx} \cdot EI \frac{d^2 w}{dx^2} \right]_{\Gamma} \tag{4.6.2}$$

As in section 3.2, utilizing the boundary condition subsets,

$$\Gamma_s \cap \Gamma_w, \qquad \Gamma_s \cap \Gamma_\theta,$$
$$\Gamma_s \cap \Gamma_V, \text{ and } \Gamma_s \cap \Gamma_M. \tag{4.6.3}$$

leads to the local weak form (LWF) for continuous beam problems:

$$0 = EI \int_{\Omega_s} \frac{d^2 w}{dx^2} \frac{d^2 v}{dx^2} dx - \int_{\Omega_s} f\, v\, dx$$
$$+ \alpha_w [(w - \tilde{w}) v]_{\Gamma_{sw}} + \alpha_\theta \left[ \left( \frac{dw}{dx} - \tilde{\theta} \right) \frac{dv}{dx} \right]_{\Gamma_{s\theta}} + \alpha_c [(w - \tilde{w}) v]_{\Gamma_{sc}} \tag{4.6.4}$$
$$- n_x [\tilde{V}\, v]_{\Gamma_{sV}} - n_x \left[ \tilde{M} \frac{dv}{dx} \right]_{\Gamma_{sM}} + n_x \left[ EI \frac{d^3 w}{dx^3} v \right]_{\Gamma_{sw}} - n_x \left[ EI \frac{d^2 w}{dx^2} \frac{dv}{dx} \right]_{\Gamma_{s\theta}}$$

In comparing Eq. (4.6.4) with the LWF developed in section 3.2 (see Eq. 3.2.15), it is noted that the only difference in the two equations is the term $\alpha_c [(w - \tilde{w}) v]_{\Gamma_c}$.

Therefore, the LWF of Eq. (4.6.4) can be used for all beam problems worked in this report; when no continuous beam boundary conditions are present, $\alpha_c = 0$, and Eq. (4.6.4) becomes Eq. (3.2.15). Following the development of section 3.5, the MLPG equations are

$$\mathbf{K}^{(\text{node})} \hat{\mathbf{d}} + \mathbf{K}^{(\text{bdry})} \hat{\mathbf{d}} - \mathbf{f}^{(\text{node})} - \mathbf{f}^{(\text{bdry})} = \mathbf{0} \tag{4.6.5}$$

where $\mathbf{K}^{(\text{node})}$ and $\mathbf{f}^{(\text{node})}$ remain as Eqs. (3.5.4b, 3.5.4d, and 3.5.4f), and the expressions

$$\alpha_c \begin{bmatrix} \chi_i^{(w)}\psi_j^{(w)} & \chi_i^{(w)}\psi_j^{(\theta)} \\ \chi_i^{(\theta)}\psi_j^{(w)} & \chi_i^{(\theta)}\psi_j^{(\theta)} \end{bmatrix}_{\Gamma_{sc}^{(i)}} \tag{4.6.6a}$$

and

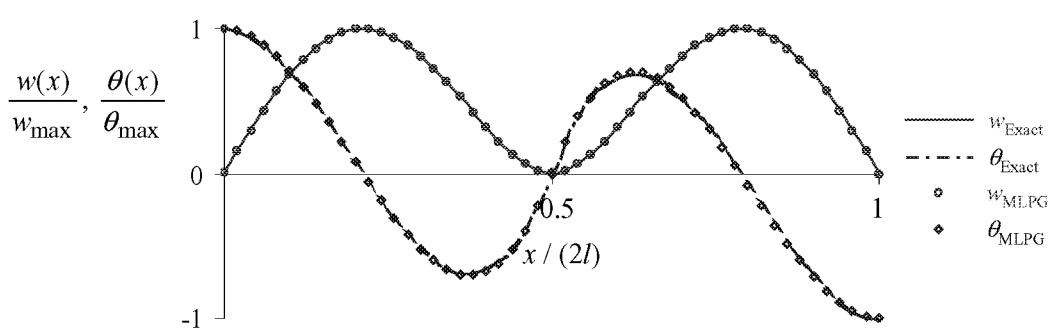
$$\alpha_c \begin{Bmatrix} \tilde{w}\chi_i^{(w)} \\ \tilde{w}\chi_i^{(\theta)} \end{Bmatrix}_{\Gamma_{sc}^{i}} \tag{4.6.6b}$$

are added to the $\mathbf{k}_{ij}^{(\text{bdry})}$ of Eq. (3.5.4e) and the $\mathbf{f}^{(\text{bdry})}$ of Eq. (3.5.4g), respectively.

The exact solution for the problem shown in Figure 4.6.1 is

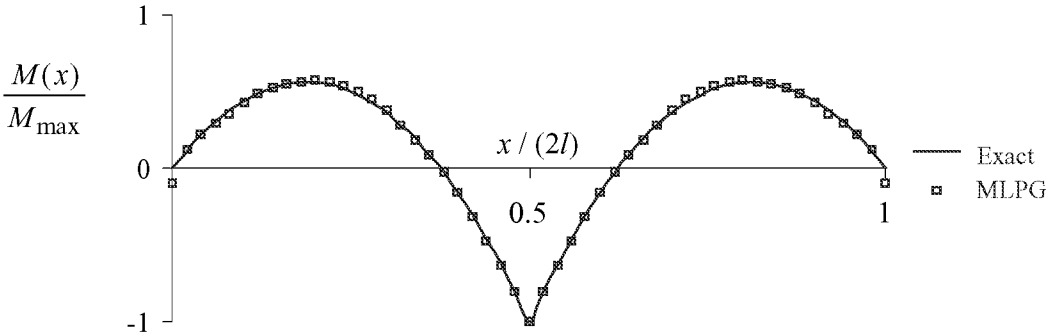
$$\begin{aligned} w(x) &= \frac{q}{48EI}\left[3lx^3 - 2x^4 - l^3x\right] && \text{for} \quad 0 \le x \le l \\ w(x) &= \frac{q}{48EI}\left[13lx^3 - 2x^4 - 30l^2x^2 + 29l^3x - 10l^4\right] && \text{for} \quad l \le x \le 2l \end{aligned} \tag{4.6.7}$$

As for the problem with the central concentrated load, the center support is expected to cause difficulty. The MLPG and exact solutions for deflection, slope, and moment obtained from the 65-node model are shown in Figure 4.6.2. The MLPG method obtained very accurate results for both the primary and secondary variables and handled the discontinuity caused by the center support well.




**(a) Deflection and Slope**

**Figure 4.6.2: MLPG and exact solutions for primary and secondary variables of a continuous beam subjected to a uniformly distributed load**




**(b) Moment**

**Figure 4.6.2 Concluded: MLPG and exact solutions for primary and secondary variables of a continuous beam subjected to a uniformly distributed load**

# Chapter 5: Concluding Remarks

The Meshless Local Petrov-Galerkin (MLPG) method has been implemented for 2-D potential and elasticity problems. In this report, the method was implemented and studied for 1-D $C^0$ problems and further developed for bending of beams – $C^1$ problems. The following conclusions are drawn from the work presented in this report:

- The MLPG method yields accurate solutions for $C^0$ and $C^1$ problems.
- The MLPG method yields continuous secondary variables as demonstrated by the bar and beam problems studied.
- A local coordinate approach is developed and validated for improving the conditioning of the [**A**] matrix that is needed to evaluate the trial functions for the beam problems.
- For beam problems, the Petrov-Galerkin approach is preferable over the Galerkin approach.
- Reasonable ranges of several parameters are required to obtain good results. The ranges of these parameters suggest the robustness of this method.

Each of these conclusions is discussed below.

## 5.1 Accurate Solutions by the MLPG Method

As discussed in Chapter 1, for any new method to compete with the Finite Element Method, the new method must retain the advantages of the FEM. This includes, most importantly, the ability of the method to yield accurate (to machine accuracy) solutions. (As stated in Chapter 1, "machine accuracy" means that the difference between the exact and numerical solutions is of the order of $10^{-14}$ when double precision

arithmetic is used.) For all the $C^0$ and $C^1$ problems presented in this report, the MLPG method yielded accurate solutions for both the primary and secondary variables.

### 5.2 Continuous Secondary Variables

As discussed in Chapter 1, one of the disadvantages of the FEM is the discontinuity of the secondary variables across inter-element boundaries. The discontinuities in the secondary variables arise because of the piecewise linear shape functions that are used to construct the trial functions. Elaborate post-processing techniques are needed to obtain smooth distributions of these secondary variables. In the MLPG method, elements are eliminated, and nodes are utilized in the domain of the problem. A diffused (i.e., not piecewise linear) trial function such as a moving least squares (MLS) interpolation is used. These diffused trial functions are smooth, and hence, smooth distributions of the secondary variables are obtained, thus eliminating the disadvantage of the FEM. These results were confirmed in Chapter 2 by application of the method to a $C^0$ problem and in Chapter 3 for $C^1$ problems.

### 5.3 Local Coordinate Approach

As discussed in Chapters 2 and 3, the trial functions used to approximate the solution are formed from shape functions that are developed by a MLS interpolation. The formation of these shape functions involves the evaluation of the [**A**] matrix. The [**A**] matrix is evaluated using the weight functions and the basis functions. The conditioning of the [**A**] matrix is determined by the order of the basis function used. As the order of the basis function is increased, the conditioning of the [**A**] matrix becomes

poor, especially for nodes far from the origin, resulting in an inaccurate computation of the inverse of [**A**] that leads to poor quality solutions. To improve the conditioning of the [**A**] matrix, the MLS approximation is defined in terms of a local basis rather than a global basis. A comparison of the results of the global and local approaches applied to patch test problems as presented in section 4.2 clearly demonstrates that the local coordinate approach produces very accurate results in comparison to the global coordinate approach.

### 5.4 The Petrov-Galerkin Approach

In the MLPG method, because the trial and test functions are chosen from different spaces, the resulting system stiffness matrices are unsymmetric. This could be perceived as a disadvantage. However, closer examination of the method reveals that this is not a disadvantage. As discussed in section 3.6, when a Galerkin approximation is used, the system matrix is symmetric. However, the Galerkin approach results in discontinuities that arise at the boundaries of the supports of the trial functions in the local sub-domain of the test function. Because of these discontinuities, elaborate numerical integration schemes are needed to integrate the weak form accurately. The local sub-domain, $\Omega_s$, of the test function is divided into sub-regions. The endpoints of these sub-regions are determined by the ends of the support domains of the trial functions that intersect $\Omega_s$. A 10-point Gaussian quadrature is used in each of the sub-regions to accurately integrate the weak form. The sub-region procedure results in large computing effort to integrate the weak form in each sub-domain $\Omega_s$. Alternately, if a Petrov-Galerkin approximation is used, the sub-region integration is not needed. A single higher

order integration rule (for example, a 20-point Gaussian) in a single sub-domain is sufficient to integrate the weak form accurately. This result was confirmed numerically for several examples in Chapter 4. Thus, while the unsymmetry of the stiffness matrices in the Petrov-Galerkin method may be construed as a disadvantage, it is far outweighed by the computational time and effort saved by the weaker requirements for numerical integration.

### 5.5 Problem Parameters

In applications of the MLPG method, several parameters are user-defined. Over certain ranges of these parameters, good performance is obtained. The minimum order of Gaussian integration required depends on the basis functions and weight functions used. Also, extremely high orders of Gaussian integration are unreasonable and unnecessary. For the problems worked in this report, numerical experimentation showed that a 20-point Gaussian, while not necessary for all simpler problems, was found to integrate the weak form accurately. The algorithm performs best when the extents of the test functions are in the range $\Delta x \leq (R_o / l) \leq 2\Delta x$, where $\Delta x$ is the nodal spacing between nodes for a uniformly distributed nodal arrangement. Similarly, the extents of the trial functions are best chosen as $8\Delta x \leq (R_j / l) \leq 16\Delta x$, but no larger than 98% of the domain of the problem (for 1-D problems).

### 5.6 Contributions of this Research

Meshless methods are becoming increasingly popular as evidenced by the large amounts of literature on the subject published in the last five years. However, much of

the research that is being conducted on meshless methods is on $C^0$ problems. In this report, one particular meshless method, the Meshless Local Petrov-Galerkin (MLPG) method, was extensively studied for $C^1$ problems. At the time this research was conducted, the literature available on the MLPG method for beam problems utilized a Galerkin approach. In this report, a Petrov-Galerkin approach was implemented and shown to be far superior to the previously available Galerkin approach (see section 3.6). Additionally, four major contributions of this report to the general field of meshless methods are: (1) a local coordinate approach to be used in the formulation of the trial functions was proposed and validated in section 4.2, (2) the performance of several test functions (presented in section 3.4) was studied, and well-defined continuity requirements for prospective test functions were determined, (3) application of the method to a problem with load discontinuity was demonstrated in section 4.5, and (4) application of the method to continuous beams was also demonstrated (section 4.6). The following publications came out as a result of the work performed in this report:

I. I. S. Raju and D. R. Phillips (2002): "A Local Coordinate Approach in the MLPG Method for Beam Problems," NASA TM-2002-211463, and

II. I. S. Raju and D. R. Phillips (2002): "A Meshless Local Petrov-Galerkin Method for Euler-Bernoulli Beam Problems," Proceedings of the ICES '02 conference, Reno, Nevada, July 31 – August 2, 2002, Paper No. 139.

## 5.7 Suggestions for Future Work

The MLPG method is still in the early stages of its development. More work needs to be performed before the method can reasonably compete with the FEM. The following are the next steps to extend the research conducted in this report.

- The method could be extended to Timoshenko beam problems. (First order shear deformation is accounted for. The assumption of normals before deformation remaining normal after deformation is relaxed to "normals remain straight but need not be normal after deformation.")
- The method could be extended to two dimensions for plate bending.
- The method needs to be modified and applied to study built-up structures.
- The method could be studied for shell analysis.
- The method relies very heavily on the user-defined parameters, namely $R_o$, and $R_j$. More research should be done to determine more robust ranges of these parameters so the method consistently obtains good results.

## References


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

S. N. Atluri, H.-G. Kim, and J. Y. Cho (1999): A critical assessment of the truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods, *Computational Mechanics*, Vol. 24, pp. 348-372.

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

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

S. N. Atluri and T. Zhu (2000): New concepts in meshless methods, *International Journal for Numerical Methods in Engineering*, Vol. 47, pp. 537-556.

K.J. Bathe (1996): *Finite Element Procedures*, Prentice-Hall, Inc., Upper Saddle River, New Jersey.

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

S.C. Chapra and R.P. Canale (1988), *Numerical Methods for Engineers*, Second Edition. McGraw-Hill Book Co., New York (See also http://www.netlib.org/napack/)

R. D. Cook, D. S. Malkus, M. E. Plesha, R. J. Witt (2002): *Concepts and Applications of Finite Element Analysis*, Fourth Edition, John Wiley & Sons, Inc.

S. De and K.-J. Bathe (2001): Towards an efficient meshless computational technique: the method of finite spheres, *Engineering Computations*, Vol. 18, No. 1 and 2, pp. 170-192.

Y. T. Gu and G.R. Liu (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.

Y. Krongauz and T. Belytschko (1996): Enforcement of essential boundary conditions in meshless approximations using finite elements, *Computer Methods in Applied Mechanics and Engineering*, Vol. 131, pp. 133-145.

Y. X. Mukherjee and S. Mukherjee (1997): On boundary conditions in the element-free Galerkin method, *Computational Mechanics*, Vol. 19, pp. 264-270.

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

I. S. Raju and T. Chen (2001): Meshless Local Petrov-Galerkin Method Applied to Axisymmetric Problems, Proceedings of the 42nd AIAA Structures, Structural Dynamics, and Materials Conference, *AIAA-2001-1253.*

J. N. Reddy (1993): *An Introduction to the Finite Element Method*, Second Edition, McGraw-Hill.

T. Zhu and S. N. Atluri (1998): A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method, *Computational Mechanics*, Vol. 21, pp. 211-222.

O. C. Zienkiewicz and R. L. Taylor (1989): *The Finite Element Method*, Fourth Edition, McGraw Hill.

## Appendix A: Computation of Derivatives of Shape Functions

This appendix presents a detailed derivation of the derivatives of the shape functions used in the MLPG method. Section 1 presents the derivatives of the shape functions for $C^0$ problems. Section 2 discusses the **[B]** matrix for $C^1$ problems. Section 3 presents the derivatives of the shape functions for $C^1$ problems.

### A.1 $C^0$ Problems

In this section, the derivatives of the shape functions for $C^0$ problems are derived first in terms of the general spatial coordinates, $x_k$, and then reduced to one dimension. In $C^0$ problems, the approximations for the solution, $u^h(\mathbf{x})$, can be written as

$$u^h(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}), \tag{A.1.1a}$$

where **p** is an $m^{\text{th}}$ order basis functions and **a** is a vector of undetermined coefficients, and

$$u^h(\mathbf{x}) = \sum_{j=1}^{n} \phi_j(\mathbf{x})\hat{u}_j, \tag{A.1.1b}$$

where $\phi_j(x)$ are shape functions, and $\hat{u}_j$ are fictitious nodal values. In Eqs. (A.1.1), **x** represents the spatial coordinates,

$$\mathbf{x} = [x_1 \quad x_2 \quad x_3]^{\mathrm{T}}. \tag{A.1.2}$$

For the local weak form, the derivative of $u^h(\mathbf{x})$ is needed. First consider the statement of $u^h(\mathbf{x})$ in Eq. (A.1.1a). Differentiating,

$$\frac{\partial u^h(\mathbf{x})}{\partial x_k} = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\frac{\partial \mathbf{a}(\mathbf{x})}{\partial x_k} + \frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{x})}{\partial x_k}\mathbf{a}(\mathbf{x}). \tag{A.1.3}$$

In Eq. (A.1.3), the second term is easy to evaluate; however, evaluation of $\partial \mathbf{a}(\mathbf{x})/\partial x_k$ in the first term is not straightforward. The evaluation of this derivative requires closer examination and is discussed below.

Consider the equation

$$\underset{(m,m)}{[\mathbf{A}]}\ \underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,n)}{[\mathbf{B}]}\ \underset{(n,1)}{\{\hat{\mathbf{u}}\}}\ , \tag{A.1.4}$$

in which $\{\hat{\mathbf{u}}\}$ are fictitious nodal values, and $[\mathbf{A}]$ and $[\mathbf{B}]$ are easily evaluated from weight functions and basis functions using Eqs. (2.2.17 and 2.2.18). Eq. (A.1.4) can be differentiated as

$$[\mathbf{A}]\frac{\partial\{\mathbf{a}\}}{\partial x_k} + \frac{\partial[\mathbf{A}]}{\partial x_k}\{\mathbf{a}\} = [\mathbf{B}]\frac{\partial\{\hat{\mathbf{u}}\}}{\partial x_k} + \frac{\partial[\mathbf{B}]}{\partial x_k}\{\hat{\mathbf{u}}\}\ . \tag{A.1.5}$$

In Eq. (A.1.5), the fictitious nodal values $\{\hat{\mathbf{u}}\}$ are not functions of $\mathbf{x}$, and thus the term $[\mathbf{B}]\cdot\left(\partial\{\hat{\mathbf{u}}\}/\partial x_k\right)$ vanishes. Rearranging Eq. (A.1.5) one obtains

$$[\mathbf{A}]\frac{\partial\{\mathbf{a}\}}{\partial x_k} = \frac{\partial[\mathbf{B}]}{\partial x_k}\{\hat{\mathbf{u}}\} - \frac{\partial[\mathbf{A}]}{\partial x_k}\{\mathbf{a}\}\ . \tag{A.1.6}$$

This leads to

$$\frac{\partial\{\mathbf{a}\}}{\partial x_k} = [\mathbf{A}]^{-1}\frac{\partial[\mathbf{B}]}{\partial x_k}\{\hat{\mathbf{u}}\} - [\mathbf{A}]^{-1}\frac{\partial[\mathbf{A}]}{\partial x_k}\{\mathbf{a}\}\ . \tag{A.1.7}$$

The vector $\{\mathbf{a}\}$ can be evaluated using Eq. (A.1.4),

$$\underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,m)}{[\mathbf{A}]^{-1}}\ \underset{(m,n)}{[\mathbf{B}]}\ \underset{(n,1)}{\{\hat{\mathbf{u}}\}}\ . \tag{A.1.8}$$

Substituting Eq. (A.1.8) into Eq. (A.1.7),

$$\frac{\partial\{\mathbf{a}\}}{\partial x_k} = [\mathbf{A}]^{-1}\frac{\partial[\mathbf{B}]}{\partial x_k}\{\hat{\mathbf{u}}\} - [\mathbf{A}]^{-1}\frac{\partial[\mathbf{A}]}{\partial x_k}[\mathbf{A}]^{-1}[\mathbf{B}]\{\hat{\mathbf{u}}\}\ . \tag{A.1.9}$$

So, substituting Eqs. (A.1.9 and A.1.8) into Eq. (A.1.3),

$$\frac{\partial u^h(\mathbf{x})}{\partial x_k} = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\left[[\mathbf{A}]^{-1}\frac{\partial[\mathbf{B}]}{\partial x_k}\{\hat{\mathbf{u}}\} - [\mathbf{A}]^{-1}\frac{\partial[\mathbf{A}]}{\partial x_k}[\mathbf{A}]^{-1}[\mathbf{B}]\{\hat{\mathbf{u}}\}\right] + \frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{x})}{\partial x_k}\left[[\mathbf{A}]^{-1}[\mathbf{B}]\{\hat{\mathbf{u}}\}\right] \ . \tag{A.1.10}$$

Now consider the statement of $u^h(\mathbf{x})$ in Eq. (A.1.1b). Differentiating,

$$\frac{\partial u^h(\mathbf{x})}{\partial x_k} = \sum_{j=1}^{n}\left(\phi_j(\mathbf{x})\frac{\partial \hat{u}_j}{\partial x_k} + \frac{\partial \phi_j(\mathbf{x})}{\partial x_k}\hat{u}_j\right) = \sum_{j=1}^{n}\frac{\partial \phi_j(\mathbf{x})}{\partial x_k}\hat{u}_j \ . \tag{A.1.11}$$

Equating the two expressions for the derivative of $u^h(\mathbf{x})$, i.e.,

$$\left.\frac{\partial u^h(\mathbf{x})}{\partial x_k}\right|_{\text{Eq. A.1.10}} = \left.\frac{\partial u^h(\mathbf{x})}{\partial x_k}\right|_{\text{Eq. A.1.11}} , \tag{A.1.12}$$

leads to

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x})\{\hat{\mathbf{u}}\}\left[[\mathbf{A}]^{-1}\frac{\partial[\mathbf{B}]}{\partial x_k} - [\mathbf{A}]^{-1}\frac{\partial[\mathbf{A}]}{\partial x_k}[\mathbf{A}]^{-1}[\mathbf{B}]\right] + \frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{x})}{\partial x_k}[\mathbf{A}]^{-1}[\mathbf{B}]\{\hat{\mathbf{u}}\} = \sum_{j=1}^{n}\frac{\partial \phi_j(\mathbf{x})}{\partial x_k}\hat{u}_j \ , \tag{A.1.13}$$

or,

$$\frac{\partial \phi_j(\mathbf{x})}{\partial x_k} = \underset{(1,m)}{\mathbf{p}^{\mathrm{T}}_{,k}}\underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,n)}{[\mathbf{B}]} + \underset{(1,m)}{\mathbf{p}^{\mathrm{T}}}\left[\underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,n)}{[\mathbf{B}]_{,k}} - \underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,m)}{[\mathbf{A}]_{,k}}\underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,n)}{[\mathbf{B}]}\right] \tag{A.1.14}$$

where

$$(\ )_{,k} \equiv \frac{\partial(\ )}{\partial x_k} \ , \qquad k = 1, 2, 3 \ . \tag{A.1.15}$$

Finally,

$$\phi_{j,k} = \sum_{g=1}^{m} p_{g,k}\left([\mathbf{A}]^{-1}[\mathbf{B}]\right)_{gj} + p_g\left[[\mathbf{A}]^{-1}[\mathbf{B}]_{,k} - [\mathbf{A}]^{-1}[\mathbf{A}]_{,k}[\mathbf{A}]^{-1}[\mathbf{B}]\right]. \quad \text{(A.1.16)}$$

Note that in this report, there is only one spatial coordinate, $\mathbf{x} = x$, as 1-D problems are considered. As a result, $x_k = x$, and the partial derivatives become full derivatives:

$$\phi_{j,x} = \sum_{g=1}^{m} p_{g,x}\left([\mathbf{A}]^{-1}[\mathbf{B}]\right)_{gj} + p_g\left[[\mathbf{A}]^{-1}[\mathbf{B}]_{,x} - [\mathbf{A}]^{-1}[\mathbf{A}]_{,x}[\mathbf{A}]^{-1}[\mathbf{B}]\right], \quad \text{(A.1.17)}$$

where

$$(\ )_{,x} \equiv \frac{d(\ )}{dx}. \quad \text{(A.1.18)}$$

## A.2 [B] Matrix for $C^1$ Problems

In section A.1, the [B] matrix is an $(m,n)$ matrix given by Eq. (2.2.18) as

$$[\mathbf{B}] = \left[\mathbf{P}^{\mathrm{T}}\boldsymbol{\lambda}\right]. \quad \text{(A.2.1)}$$

In $C^1$ problems, the $[\mathbf{B}]$ matrix is an $(m,2n)$ matrix given by Eq. (3.3.14) as

$$[\mathbf{B}] = \left[\mathbf{P}^{\mathrm{T}}\boldsymbol{\lambda}, \quad \mathbf{P}_x^{\mathrm{T}}\boldsymbol{\lambda}\right]. \quad \text{(A.2.2)}$$

Consider the equation

$$\underset{(m,m)}{[\mathbf{A}]}\ \underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2n,1)}{\{\hat{\mathbf{s}}\}}\ . \quad \text{(A.2.3)}$$

in which, as in Eq. (3.3.8), $\{\hat{\mathbf{s}}\}$ is a vector containing $\{\hat{\mathbf{w}}\}$ and $\{\hat{\mathbf{t}}\}$, the fictitious nodal values of deflection and slope, respectively. Also, $[\mathbf{A}]$ and $[\mathbf{B}]$ are easily evaluated from weight functions, basis functions, and derivatives of basis functions using Eqs. (3.3.13 and 3.3.14). The approximations to the solution, $w^h(x)$, can be written as in Eq. (A.1.2a) as

$$w^h(x) = \mathbf{p}^{\mathrm{T}}(x)\mathbf{a}(x). \tag{A.2.4}$$

Solving for $\{\mathbf{a}\}$ in Eq. (A.2.3) and substituting into Eq. (A.2.4) yields Eq. (3.3.16),

$$w^h(x) = \underset{(1,m)}{\mathbf{p}^{\mathrm{T}}(x)}\underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,2n)}{[\mathbf{B}]}\underset{(2\mathrm{n},1)}{\begin{Bmatrix}\hat{\mathbf{w}}\\ \hat{\mathbf{t}}\end{Bmatrix}}, \tag{A.2.5}$$

or

$$w^h(x) = \underset{(1,m)}{\mathbf{p}^{\mathrm{T}}(x)}\underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,n)}{[\mathbf{B}_w]}\underset{(n,1)}{\{\hat{\mathbf{w}}\}} + \underset{(1,m)}{\mathbf{p}^{\mathrm{T}}(x)}\underset{(m,m)}{[\mathbf{A}]^{-1}}\underset{(m,n)}{[\mathbf{B}_t]}\underset{(n,1)}{\{\hat{\mathbf{t}}\}}, \tag{A.2.6}$$

where

$$[\mathbf{B}_w \quad \mathbf{B}_t] = \left[\mathbf{P}^{\mathrm{T}}\boldsymbol{\lambda} \quad \mathbf{P}_x^{\mathrm{T}}\boldsymbol{\lambda}\right]. \tag{A.2.7}$$

From Eq. (A.2.6), the shape functions are as Eqs. (3.3.18):

$$\begin{aligned}\psi_j^{(w)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}^{\mathrm{T}}\boldsymbol{\lambda}\right]_{gj}\\ \psi_j^{(\theta)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{P}_x^{\mathrm{T}}\boldsymbol{\lambda}\right]_{gj}\end{aligned} \tag{A.2.8}$$

Substitution of $[\mathbf{B}_w]$ and $[\mathbf{B}_t]$ from Eq. (A.2.7) into Eq. (A.2.8) yields

$$\begin{aligned}\psi_j^{(w)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{B}_w\right]_{gj}\\ \psi_j^{(\theta)}(x) &= \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{B}_t\right]_{gj}.\end{aligned} \tag{A.2.9}$$

## A.3 $C^1$ Problems

The first, second, and third derivatives of

$$\psi_j^{(w)}(x) = \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{B}_w\right]_{gj}$$
$$\psi_j^{(\theta)}(x) = \sum_{g=1}^{m} p_g(x_j)\left[\mathbf{A}^{-1}\mathbf{B}_t\right]_{gj} \tag{A.3.1}$$

with respect to $x$ are sought. The derivatives are found via the procedure outlined below.

In $C^1$ problems, the approximations for the solution, $w^h(x)$, can be written as

$$w^h(x) = \mathbf{p}^{\mathrm{T}}(x)\mathbf{a}(x), \tag{A.3.2a}$$

where $\mathbf{p}$ is an $m^{\text{th}}$ order basis function, and $\mathbf{a}$ is a vector of undetermined coefficients, and

$$w^h(x) = \sum_{j=1}^{n}\left(\psi_j^{(w)}(x)\hat{w}_j + \psi_j^{(\theta)}(x)\hat{\theta}_j\right), \tag{A.3.2b}$$

where $\psi_j^{(w)}(x)$ and $\psi_j^{(\theta)}(x)$ are the shape functions (A.2.9), and $\hat{w}_j$ and $\hat{\theta}_j$ are fictitious nodal values. For the local weak form, the first, second, and third derivatives of $w^h(x)$ are needed. To evaluate these derivatives, a general procedure similar to that presented in section A.1 for $C^0$ problems is used.

### A.3.1 <u>First Derivatives</u>

Differentiating Eq. (A.3.2a) with respect to $x$, one obtains

$$\frac{dw^h(x)}{dx} = \mathbf{p}^{\mathrm{T}}(x)\frac{d\mathbf{a}(x)}{dx} + \frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\mathbf{a}(x). \tag{A.3.3}$$

In Eq. (A.3.3), the second term is easy to evaluate; however, evaluation of $d\mathbf{a}(x)/dx$ in the first term is not straightforward. The evaluation of this derivative requires closer examination and is discussed below.

Consider Eq. (A.2.3):

$$\underset{(m,m)}{[\mathbf{A}]}\ \underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,2n)}{[\mathbf{B}]}\ \underset{(2n,1)}{\{\hat{\mathbf{s}}\}}\ . \tag{A.3.4}$$

Eq. (A.3.4) can be rewritten as

$$\underset{(m,m)}{[\mathbf{A}]}\ \underset{(m,1)}{\{\mathbf{a}\}} = \underset{(m,n)}{[\mathbf{B}_w]}\underset{(n,1)}{\{\hat{\mathbf{w}}\}} + \underset{(m,n)}{[\mathbf{B}_t]}\ \underset{(n,1)}{\{\hat{\mathbf{t}}\}}\ , \tag{A.3.5}$$

where $[\mathbf{B}_w]$ and $[\mathbf{B}_t]$ are presented in section A.2. Differentiating Eq. (A.3.5) with respect to $x$, one obtains

$$[\mathbf{A}]\frac{d\{\mathbf{a}\}}{dx} + \frac{d[\mathbf{A}]}{dx}\{\mathbf{a}\} = [\mathbf{B}_w]\frac{d\{\hat{\mathbf{w}}\}}{dx} + \frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{B}_t]\frac{d\{\hat{\mathbf{t}}\}}{dx} + \frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\}\ . \tag{A.3.6}$$

In Eq. (A.3.6), because the fictitious nodal values $\{\hat{\mathbf{w}}\}$ and $\{\hat{\mathbf{t}}\}$ are not functions of $x$, the terms $[\mathbf{B}_w]\cdot\left(d\{\hat{\mathbf{w}}\}/dx\right)$ and $[\mathbf{B}_t]\cdot\left(d\{\hat{\mathbf{t}}\}/dx\right)$ vanish. Rearranging Eq. (A.3.6),

$$[\mathbf{A}]\frac{d\{\mathbf{a}\}}{dx} = \frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + \frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} - \frac{d[\mathbf{A}]}{dx}\{\mathbf{a}\}\ . \tag{A.3.7}$$

This leads to

$$\frac{d\{\mathbf{a}\}}{dx} = [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\{\mathbf{a}\}\ . \tag{A.3.8}$$

The vector $\{\mathbf{a}\}$ can be evaluated using Eq. (A.3.5),

$$\{\mathbf{a}\} = [\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\ . \tag{A.3.9}$$

Substituting Eq. (A.3.9) into Eq. (A.3.8),

$$\begin{aligned}\frac{d\{\mathbf{a}\}}{dx} &= [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} \\ &\quad - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} \\ &\quad - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\ .\end{aligned} \tag{A.3.10}$$

So, substituting Eqs. (A.3.10 and A.3.9) into Eq. (A.3.3),

$$\frac{dw^h(x)}{dx} = \mathbf{p}^{\mathrm{T}}(x)\Bigg[\ [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\}$$

$$-[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\ \Bigg] \qquad \text{(A.3.11)}$$

$$+\frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\Big[[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\Big]\ .$$

Now consider the statement of $w^h(x)$ in Eq. (A.3.2b). Differentiating with respect to $x$,

$$\frac{dw^h(x)}{dx} = \sum_{j=1}^{n}\left(\hat{w}_j\frac{d\psi_j^{(w)}(x)}{dx} + \frac{d\hat{w}_j}{dx}\psi_j^{(w)}(x) + \hat{\theta}_j\frac{d\psi_j^{(\theta)}(x)}{dx} + \frac{d\hat{\theta}_j}{dx}\psi_j^{(\theta)}(x)\right)$$

$$= \sum_{j=1}^{n}\left(\hat{w}_j\frac{d\psi_j^{(w)}(x)}{dx} + \hat{\theta}_j\frac{d\psi_j^{(\theta)}(x)}{dx}\right)\ . \qquad \text{(A.3.12)}$$

Equating the two expressions for the derivative of $w^h(x)$, i.e.,

$$\left.\frac{dw^h(x)}{dx}\right|_{\text{Eq. A.3.11}} = \left.\frac{dw^h(x)}{dx}\right|_{\text{Eq. A.3.12}}\ , \qquad \text{(A.3.13)}$$

leads to

$$\mathbf{p}^{\mathrm{T}}(x)\{\hat{\mathbf{w}}\}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right] + \frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\{\hat{\mathbf{w}}\}[\mathbf{A}]^{-1}[\mathbf{B}_w]$$

$$+\mathbf{p}^{\mathrm{T}}(x)\{\hat{\mathbf{w}}\}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right] + \frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\{\hat{\mathbf{t}}\}[\mathbf{A}]^{-1}[\mathbf{B}_t] \qquad \text{(A.3.14)}$$

$$= \sum_{j=1}^{n}\left(\hat{w}_j\frac{d\psi_j^{(w)}(x)}{dx} + \hat{\theta}_j\frac{d\psi_j^{(\theta)}(x)}{dx}\right)\ .$$

Comparing the coefficients of $\hat{w}_j$ and $\hat{\theta}_j$ on both sides of Eq. A.3.14 gives

$$\frac{d\psi_j^{(w)}}{dx} = \sum_{g=1}^{m} \frac{dp_g}{dx}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]\right)_{gj} + p_g\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right]_{gj} \tag{A.3.15a}$$

and

$$\frac{d\psi_j^{(\theta)}}{dx} = \sum_{g=1}^{m} \frac{dp_g}{dx}\left([\mathbf{A}]^{-1}[\mathbf{B}_t]\right)_{gj} + p_g\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right]_{gj} . \tag{A.3.15b}$$

A.3.2 Second Derivatives

Differentiating Eq. (A.3.3) with respect to $x$, one obtains

$$\frac{d^2 w^h(x)}{dx^2} = \mathbf{p}^{\mathrm{T}}(x)\frac{d^2\mathbf{a}(x)}{dx^2} + 2\frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\frac{d\mathbf{a}(x)}{dx} + \frac{d^2\mathbf{p}^{\mathrm{T}}(x)}{dx^2}\mathbf{a}(x) . \tag{A.3.16}$$

In Eq. (A.3.16), the last term is easy to evaluate. The term $d\mathrm{a}(x)/dx$ was found in section A.3.1. The evaluation of $d^2\mathrm{a}(x)/dx^2$ in the first term requires closer examination and is discussed below.

Consider Eq. (A.3.7):

$$[\mathbf{A}]\frac{d\{\mathbf{a}\}}{dx} = \frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + \frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} - \frac{d[\mathbf{A}]}{dx}\{\mathbf{a}\} . \tag{A.3.17}$$

Differentiating Eq. (A.3.17) with respect to $x$, one obtains

$$[\mathbf{A}]\frac{d^2\{\mathbf{a}\}}{dx^2}+\frac{d[\mathbf{A}]}{dx}\frac{d\{\mathbf{a}\}}{dx}=\frac{d^2[\mathbf{B}_w]}{dx^2}\{\hat{\mathbf{w}}\}+\frac{d^2[\mathbf{B}_t]}{dx^2}\{\hat{\mathbf{t}}\}$$
$$-\frac{d^2[\mathbf{A}]}{dx^2}\{\mathbf{a}\}-\frac{d[\mathbf{A}]}{dx}\frac{d\{\mathbf{a}\}}{dx} \quad . \tag{A.3.18}$$

This leads to

$$\frac{d^2\{\mathbf{a}\}}{dx^2}=[\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_w]}{dx^2}\{\hat{\mathbf{w}}\}+[\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_t]}{dx^2}\{\hat{\mathbf{t}}\}$$
$$-2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\frac{d\{\mathbf{a}\}}{dx}-[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\{\mathbf{a}\} \quad . \tag{A.3.19}$$

Substitution of the expressions for $\{\mathbf{a}\}$ and $d\{\mathbf{a}\}/dx$ from Eqs. (A.3.9 and A.3.10) into Eq. (A.3.19) yields

$$\frac{d^2\{\mathbf{a}\}}{dx^2}=[\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_w]}{dx^2}\{\hat{\mathbf{w}}\}+[\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_t]}{dx^2}\{\hat{\mathbf{t}}\}$$
$$-2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\left\{ [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\}+[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\}\right.$$
$$\left.-[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\}-[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\right\} \tag{A.3.20}$$
$$-[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left\{[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\}+[\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\right\} \quad .$$

So, substituting Eqs. (A.3.20, A.3.10, and A.3.9) into Eq. (A.3.16),

$$
\begin{aligned}
\frac{d^2 w^h(x)}{dx^2} = \mathbf{p}^{\mathrm{T}}(x)\Bigg[ & [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_w]}{dx^2}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_t]}{dx^2}\{\hat{\mathbf{t}}\} \\
& - 2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\Bigg\{ [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} \\
& \qquad - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Bigg\} \\
& - [\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\Big\{[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\Big\} \Bigg] \\
& + 2\frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\Bigg[ [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} \\
& \qquad - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Bigg] \\
& + \frac{d^2\mathbf{p}^{\mathrm{T}}(x)}{dx^2}\Big[[\mathbf{A}]^{-1}[\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}[\mathbf{B}_t]\{\hat{\mathbf{t}}\}\Big] \ .
\end{aligned}
\tag{A.3.21}
$$

Now consider the statement of $dw^h(x)/dx$ in Eq. (A.3.12). Differentiating with respect to $x$,

$$
\frac{d^2 w^h(x)}{dx^2} = \sum_{j=1}^{n}\left( \hat{w}_j \frac{d^2\psi_j^{(w)}(x)}{dx^2} + \hat{\theta}_j \frac{d^2\psi_j^{(\theta)}(x)}{dx^2} \right). \tag{A.3.22}
$$

Equating the two expressions for the second derivative of $w^h(\mathbf{x})$, i.e.,

$$
\left.\frac{d^2 w^h(x)}{dx^2}\right|_{\text{Eq. A.3.21}} = \left.\frac{d^2 w^h(x)}{dx^2}\right|_{\text{Eq. A.3.22}} , \tag{A.3.23}
$$

leads to

$$\mathbf{p}^{\mathrm{T}}(x)\{\hat{\mathbf{w}}\}\left[\ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_w]}{dx^2}-2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\left\{\ [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}-[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\ \right\}\right.$$

$$\left.-[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left\{[\mathbf{A}]^{-1}[\mathbf{B}_w]\right\}\ \right]$$

$$+2\frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\{\hat{\mathbf{w}}\}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx}-[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right]$$

$$+\frac{d^2\mathbf{p}^{\mathrm{T}}(x)}{dx^2}\{\hat{\mathbf{w}}\}\left[[\mathbf{A}]^{-1}[\mathbf{B}_w]\right]$$

$$+\mathbf{p}^{\mathrm{T}}(x)\{\hat{\mathbf{t}}\}\left[\ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_t]}{dx^2}-2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\left\{\ [\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}-[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\ \right\}\right.$$

$$\left.-[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left\{[\mathbf{A}]^{-1}[\mathbf{B}_t]\right\}\right.$$

$$+2\frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\{\hat{\mathbf{t}}\}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx}-[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right]$$

$$+\frac{d^2\mathbf{p}^{\mathrm{T}}(x)}{dx^2}\{\hat{\mathbf{t}}\}\left[[\mathbf{A}]^{-1}[\mathbf{B}_t]\right]$$

$$=\sum_{j=1}^{n}\left(\hat{w}_j\frac{d^2\psi_j^{(w)}(x)}{dx^2}+\hat{\theta}_j\frac{d^2\psi_j^{(\theta)}(x)}{dx^2}\right)\ . \tag{A.3.24}$$

Comparing the coefficients of $\hat{w}_j$ and $\hat{\theta}_j$ on both sides of Eq. (A.3.24) gives

$$
\begin{aligned}
\frac{d^2\psi_j^{(w)}}{dx^2} = \sum_{g=1}^{m} \Bigg\{ & \frac{d^2 p_g}{dx^2}\Big([\mathbf{A}]^{-1}[\mathbf{B}_w]\Big)_{gj} \\
& + 2\frac{dp_g}{dx}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right]_{gj} \\
& + p_g\Bigg[ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_w]}{dx^2} \\
& \quad - 2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\left\{[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right\} \\
& \quad - [\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\Big\{[\mathbf{A}]^{-1}[\mathbf{B}_w]\Big\}\Bigg]_{gj} \Bigg\}
\end{aligned}
\tag{A.3.25a}
$$

and

$$
\begin{aligned}
\frac{d^2\psi_j^{(\theta)}}{dx^2} = \sum_{g=1}^{m} \Bigg\{ & \frac{d^2 p_g}{dx^2}\Big([\mathbf{A}]^{-1}[\mathbf{B}_t]\Big)_{gj} \\
& + 2\frac{dp_g}{dx}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right]_{gj} \\
& + p_g\Bigg[ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_t]}{dx^2} \\
& \quad - 2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\left\{[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right\} \\
& \quad - [\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\Big\{[\mathbf{A}]^{-1}[\mathbf{B}_t]\Big\}\Bigg]_{gj} \Bigg\}
\end{aligned}
\tag{A.3.25b}
$$

A.3.3 Third Derivatives

Differentiating Eq. (A.3.16) with respect to $x$, one obtains

$$\frac{d^3 w^h(x)}{dx^3} = \mathbf{p}^{\mathrm{T}}(x)\frac{d^3\mathbf{a}(x)}{dx^3} + 3\frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx}\frac{d^2\mathbf{a}(x)}{dx^2} + 3\frac{d^2\mathbf{p}^{\mathrm{T}}(x)}{dx^2}\frac{d\mathbf{a}(x)}{dx} + \frac{d^3\mathbf{p}^{\mathrm{T}}(x)}{dx^3}\mathbf{a}(x) \quad . \tag{A.3.26}$$

In Eq. (A.3.26), the last term is easy to evaluate. The term $d\mathbf{a}(x)/dx$ was found in section A.3.1, and the term $d^2\mathbf{a}(x)/dx^2$ was found in section A.3.2. As in sections A.3.1 and A.3.2, the evaluation of $d^3\mathbf{a}(x)/dx^3$ in the requires closer examination and is discussed below.

Consider Eq. (A.3.18). Differentiating with respect to $x$, one obtains

$$\begin{aligned}[\mathbf{A}]\frac{d^3\{\mathbf{a}\}}{dx^3} &+ \frac{d[\mathbf{A}]}{dx}\frac{d^2\{\mathbf{a}\}}{dx^2} + \frac{d[\mathbf{A}]}{dx}\frac{d^2\{\mathbf{a}\}}{dx^2} + \frac{d^2[\mathbf{A}]}{dx^2}\frac{d\{\mathbf{a}\}}{dx} \\ &= \frac{d^3[\mathbf{B}_w]}{dx^3}\{\hat{\mathbf{w}}\} + \frac{d^3[\mathbf{B}_t]}{dx^3}\{\hat{\mathbf{t}}\} \\ &- \frac{d^2[\mathbf{A}]}{dx^2}\frac{d\{\mathbf{a}\}}{dx} - \frac{d^3[\mathbf{A}]}{dx^3}\{\mathbf{a}\} - \frac{d[\mathbf{A}]}{dx}\frac{d^2\{\mathbf{a}\}}{dx^2} - \frac{d^2[\mathbf{A}]}{dx^2}\frac{d\{\mathbf{a}\}}{dx} \quad .\end{aligned} \tag{A.3.27}$$

This leads to

$$\begin{aligned}\frac{d^3\{\mathbf{a}\}}{dx^3} &= [\mathbf{A}]^{-1}\frac{d^3[\mathbf{B}_w]}{dx^3}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1}\frac{d^3[\mathbf{B}_t]}{dx^3}\{\hat{\mathbf{t}}\} \\ &- 3[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\frac{d^2\{\mathbf{a}\}}{dx^2} - 3[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\frac{d\{\mathbf{a}\}}{dx} - \frac{d^3[\mathbf{A}]}{dx^3}\{\mathbf{a}\} \quad .\end{aligned} \tag{A.3.28}$$

Substitution of Eqs. (A.3.9, A.3.10, A.3.20, and A.3.28) into Eq. (A.3.26) yields

$$
\begin{aligned}
\frac{d^3 w^h(x)}{dx^3} = \mathbf{p}^{\mathrm{T}}(x) \Bigg\{ & [\mathbf{A}]^{-1} \frac{d^3[\mathbf{B}_w]}{dx^3}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} \frac{d^3[\mathbf{B}_t]}{dx^3}\{\hat{\mathbf{t}}\} \\
& - 3[\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} \Bigg[ [\mathbf{A}]^{-1} \frac{d^2[\mathbf{B}_w]}{dx^2}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} \frac{d^2[\mathbf{B}_t]}{dx^2}\{\hat{\mathbf{t}}\} \\
& \qquad - 2[\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} \Bigg\{ [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} \\
& \qquad\qquad - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} \\
& \qquad\qquad - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Bigg\} \\
& \qquad - [\mathbf{A}]^{-1} \frac{d^2[\mathbf{A}]}{dx^2} \Big\{ [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Big\} \Bigg] \\
& - 3[\mathbf{A}]^{-1} \frac{d^2[\mathbf{A}]}{dx^2} \Bigg[ [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} \\
& \qquad - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} \\
& \qquad - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Bigg] \\
& - [\mathbf{A}]^{-1} \frac{d^3[\mathbf{A}]}{dx^3} \Big[ [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Big] \Bigg\} \\
& + 3 \frac{d\mathbf{p}^{\mathrm{T}}(x)}{dx} \Bigg\{ [\mathbf{A}]^{-1} \frac{d^2[\mathbf{B}_w]}{dx^2}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} \frac{d^2[\mathbf{B}_t]}{dx^2}\{\hat{\mathbf{t}}\} \\
& \qquad - 2[\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} \Bigg\{ [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} \\
& \qquad\qquad - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} \\
& \qquad\qquad - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Bigg\} \\
& \qquad - [\mathbf{A}]^{-1} \frac{d^2[\mathbf{A}]}{dx^2} \Big\{ [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Big\} \Bigg\} \\
& + 3 \frac{d^2\mathbf{p}^{\mathrm{T}}(x)}{dx^2} \Bigg\{ [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_w]}{dx}\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} \frac{d[\mathbf{B}_t]}{dx}\{\hat{\mathbf{t}}\} \\
& \qquad - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} - [\mathbf{A}]^{-1} \frac{d[\mathbf{A}]}{dx} [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Bigg\} \\
& + \frac{d^3\mathbf{p}^{\mathrm{T}}(x)}{dx^3} \Bigg\{ [\mathbf{A}]^{-1} [\mathbf{B}_w]\{\hat{\mathbf{w}}\} + [\mathbf{A}]^{-1} [\mathbf{B}_t]\{\hat{\mathbf{t}}\} \Bigg\} \ .
\end{aligned}
\tag{A.3.29}
$$

Now consider the statement of $d^3w^h(x)/dx^3$ in Eq. (A.3.22). Differentiating with respect to $x$,

$$\frac{d^3 w^h(x)}{dx^3} = \sum_{j=1}^{n} \left( \hat{w}_j \frac{d^3 \psi_j^{(w)}(x)}{dx^3} + \hat{\theta}_j \frac{d^3 \psi_j^{(\theta)}(x)}{dx^3} \right) . \tag{A.3.30}$$

Equating the two expressions for the third derivative of $w^h(x)$, i.e.,

$$\left. \frac{d^3 w^h(x)}{dx^3} \right|_{\text{Eq. A.3.29}} = \left. \frac{d^3 w^h(x)}{dx^3} \right|_{\text{Eq. A.3.30}} , \tag{A.3.31}$$

and comparing the coefficients of $\hat{w}_j$ and $\hat{\theta}_j$ gives

$$
\begin{aligned}
\frac{d^3\psi_j^{(w)}}{dx^3} = \sum_{g=1}^{m} \Bigg\{ & \frac{d^3 p_g}{dx^3}\left([\mathbf{A}]^{-1}[\mathbf{B}_w]\right)_{gj} \\
& + 3\frac{d^2 p_g}{dx^2}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right]_{gj} \\
& + 3\frac{dp_g}{dx}\Bigg[ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_w]}{dx^2} \\
& \qquad - 2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\left\{[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right\} \\
& \qquad - [\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left\{[\mathbf{A}]^{-1}[\mathbf{B}_w]\right\}\Bigg]_{gj} \\
& + p_g\Bigg[ [\mathbf{A}]^{-1}\frac{d^3[\mathbf{B}_w]}{dx^3} \\
& \quad - 3[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\Bigg\{ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_w]}{dx^2} \\
& \qquad - 2[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left([\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right) \\
& \qquad - [\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}[\mathbf{A}]^{-1}[\mathbf{B}_w] \Bigg\} \\
& \quad - 3[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left\{[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_w]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_w]\right\} \\
& \quad - [\mathbf{A}]^{-1}\frac{d^3[\mathbf{A}]}{dx^3}[\mathbf{A}]^{-1}[\mathbf{B}_w] \Bigg]_{gj} \Bigg\}
\end{aligned}
\tag{A.3.32a}
$$

and

$$
\begin{aligned}
\frac{d^3\psi_j^{(\theta)}}{dx^3} = \sum_{g=1}^{m} \Bigg\{ & \frac{d^3 p_g}{dx^3}\left([\mathbf{A}]^{-1}[\mathbf{B}_t]\right)_{gj} \\
& + 3\frac{d^2 p_g}{dx^2}\left[[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right]_{gj} \\
& + 3\frac{dp_g}{dx}\Bigg[ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_t]}{dx^2} \\
& \qquad - 2[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\left\{[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right\} \\
& \qquad - [\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left\{[\mathbf{A}]^{-1}[\mathbf{B}_t]\right\}\Bigg]_{gj} \\
& + p_g\Bigg[ [\mathbf{A}]^{-1}\frac{d^3[\mathbf{B}_t]}{dx^3} \\
& \qquad - 3[\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}\Bigg\{ [\mathbf{A}]^{-1}\frac{d^2[\mathbf{B}_t]}{dx^2} \\
& \qquad\qquad - 2[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left([\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right) \\
& \qquad\qquad - [\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}[\mathbf{A}]^{-1}[\mathbf{B}_t] \Bigg\} \\
& \qquad - 3[\mathbf{A}]^{-1}\frac{d^2[\mathbf{A}]}{dx^2}\left\{[\mathbf{A}]^{-1}\frac{d[\mathbf{B}_t]}{dx} - [\mathbf{A}]^{-1}\frac{d[\mathbf{A}]}{dx}[\mathbf{A}]^{-1}[\mathbf{B}_t]\right\} \\
& \qquad - [\mathbf{A}]^{-1}\frac{d^3[\mathbf{A}]}{dx^3}[\mathbf{A}]^{-1}[\mathbf{B}_t] \Bigg]_{gj} \Bigg\}
\end{aligned}
\tag{A.3.32b}
$$

## Appendix B: Conditioning of Matrices

In this appendix, the conditioning of matrices is discussed. Properties of ill-conditioned matrices are presented, followed by the definition of the condition number and an application to an example problem.

### B.1 Ill-Conditioning

An ill-conditioned matrix is one that is nearly singular, i.e., the matrix has rows that are almost scalar multiples of each other. Singular matrices cannot be inverted, and thus the inversion of ill-conditioned (nearly singular) matrices yields poor results.

Consider a system of equations,

$$\mathbf{[D]\{R\} = \{P\}}, \tag{B.1.1}$$

for which a solution is sought. The solution is found by inverting the **[D]** matrix,

$$\mathbf{\{R\} = [D]^{-1}\{P\}}. \tag{B.1.2}$$

In numerical applications of equation solving, an accurate computation of $[\mathbf{D}]^{-1}$ depends on the accuracy to which the components of **[D]** are stored, i.e., the number of significant digits maintained for each component of **[D]**. The condition number of **[D]**, *cond*[**D**], provides an estimate of the number of digits lost in computing this inverse. Large condition numbers usually indicate ill conditioning. The question arises, "How large is large in terms of condition numbers?" To answer this question, the method by which condition numbers are calculated and an example is presented below.

## B.2 Conditioning Numbers

Consider the system described by (Cook *et. al.*, 2002)

$$\begin{bmatrix} d_1 & -d_1 \\ -d_1 & d_1 + d_2 \end{bmatrix} \begin{Bmatrix} r_1 \\ r_2 \end{Bmatrix} = \begin{Bmatrix} p_1 \\ 0 \end{Bmatrix} \tag{B.2.1}$$

If $d_1 >> d_2$, the second row of [**D**] is essentially the negative of the first row. Thus the matrix [**D**] is nearly singular and thus ill-conditioned. The conditioning number of a matrix [**D**] may be defined as (Cook *et al.*, 2002)

$$cond[\mathbf{D}] = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{B.2.2}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of [**D**]. The eigenvalues of [**D**] are computed from

$$|\mathbf{D} - \lambda \mathbf{I}| = 0 \tag{B.2.3}$$

where **I** is the identity matrix. In numerical computations, truncation and round-off errors result in the existence of errors $\delta\mathbf{D}$ and $\delta\mathbf{R}$, related to [**D**] and [**R**] by (Bathe, 1996)

$$\delta\,\mathbf{R} = -\mathbf{D}^{-1} \cdot \delta\,\mathbf{D} \cdot \mathbf{R}\,. \tag{B.2.4}$$

Taking norms, Eq. (B.2.4) becomes

$$\frac{\|\delta\,\mathbf{R}\|}{\|\mathbf{R}\|} \le cond[\mathbf{D}] \frac{\|\delta\,\mathbf{D}\|}{\|\mathbf{D}\|}\,. \tag{B.2.5}$$

To evaluate these errors, assume $t$-digit precision in the computer, and $s$-digit precision in the solution. Then,

$$\frac{\|\delta\,\mathbf{D}\|}{\|\mathbf{D}\|} = 10^{-t} \tag{B.2.6a}$$

and

$$\frac{\|\delta \mathbf{R}\|}{\|\mathbf{R}\|} = 10^{-s} \tag{B.2.6b}$$

Substitution of Eqs. (B.2.6) into (B.2.5) yields an estimate of the number of accurate digits maintained in the solution:

$$s \geq t - \log_{10}[cond[\mathbf{D}]]. \tag{B.2.7}$$

In the system of Eq. (B.2.1), consider $d_1 = 2$ and $d_2 = 1$ ($d_1 > d_2$):

$$[D] = \begin{bmatrix} 2 & -2 \\ -2 & 3 \end{bmatrix}. \tag{B.2.8}$$

To compute the eigenvalues,

$$\mathbf{D} - \lambda\mathbf{I} = \begin{bmatrix} 2 & -2 \\ -2 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 2-\lambda & -2 \\ -2 & 3-\lambda \end{bmatrix}, \tag{B.2.9}$$

and, according to Eq. (B.2.3),

$$\begin{vmatrix} 2-\lambda & -2 \\ -2 & 3-\lambda \end{vmatrix} = 0$$
$$(2-\lambda)(3-\lambda) - 4 = 0 \quad . \tag{B.2.10}$$

The eigenvalues are therefore

$$\lambda_1 = 4.56155 \tag{B.2.11}$$

$$\lambda_2 = 0.43845$$

and the condition number is calculated as

$$cond[D] = \frac{4.56155}{0.43845} = 10.4038. \tag{B.2.12}$$

Assuming a double precision computer is used, $t = 14$, and the number of accurate digits maintained in the solution can then be computed as

$$s \geq 14 - \log_{10}[10.4038] \cong 13. \tag{B.2.13}$$

Now consider the case $d_1 >> d_2$, for example, $d_1 = 2$ and $d_2 = 1\times10^{-6}$:

$$[D] = \begin{bmatrix} 2 & -2 \\ -2 & 2.000001 \end{bmatrix}. \tag{B.2.14}$$

From Eq. (B.2.3),

$$\begin{vmatrix} 2-\lambda & -2 \\ -2 & 2.000001-\lambda \end{vmatrix} = 0, \tag{B.2.15}$$

and the eigenvalues are

$$\lambda_1 = 4.0000005 \tag{B.2.16}$$

$$\lambda_2 = 0.0000005 \quad .$$

The condition number is therefore calculated as

$$cond[D] = \frac{4.0000005}{0.0000005} = 8000001. \tag{B.2.17}$$

Using the same double precision computer, t = 14, the number of accurate digits maintained in this solution is computed as

$$s \geq 14 - \log_{10}[8000001] \cong 7. \tag{B.2.18}$$

In the entry 2.000001 of Eq. (B.2.13), the "1" that keeps the matrix from becoming singular is in the seventh significant digit location. Because only seven digits are maintained during subsequent computations (see Eq. B.2.17), the inverse of **[D]** in Eq. (B.2.14) will be very inaccurate. Thus, the conditioning number of a matrix is a good indicator of how well the matrix is conditioned.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| 09-2002 | Technical Memorandum | |

**4. TITLE AND SUBTITLE**
Meshless Local Petrov-Galerkin Method for Bending Problems

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Phillips, Dawn, R. and Raju, Ivatury, S.

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
706-21-21-06

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
NASA Langley Research Center
Hampton, VA 23681-2199

**8. PERFORMING ORGANIZATION REPORT NUMBER**
L-18236

**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/TM -2002-211936

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified - Unlimited
Subject Category 39
Availability: NASA CASI (301) 621-0390 Distribution: Standard

**13. SUPPLEMENTARY NOTES**
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://techreports.larc.nasa.gov/cgi-bin/NTRS
Phillips: Lockheed Martin Space Operations, Langley Research Center, Hampton, Virginia; Raju: Langley Research Center, Hampton, Virginia

**14. ABSTRACT**

Recent literature shows extensive research work on meshless or element-free methods as alternatives to the versatile Finite Element Method. One such meshless method is the Meshless Local Petrov-Galerkin (MLPG) method. In this report, the method is developed for bending of beams – C1 problems. A generalized moving least squares (GMLS) interpolation is used to construct the trial functions, and spline and power weight functions are used as the test functions. The method is applied to problems for which exact solutions are available to evaluate its effectiveness. The accuracy of the method is demonstrated for problems with load discontinuities and continuous beam problems. A Petrov-Galerkin implementation of the method is shown to greatly reduce computational time and effort and is thus preferable over the previously developed Galerkin approach. The MLPG method for beam problems yields very accurate deflections and slopes and continuous moment and shear forces without the need for elaborate post-processing techniques.

**15. SUBJECT TERMS**
Euler-Bernoulli beams, thin beam problems, weight function continuity, moving least squares interpolations, meshless methods, Meshless Local Petrov-Galerkin (MLPG) method

| 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) |
| U | U | U | UU | 164 | 19b. TELEPHONE NUMBER *(Include area code)* (301) 621-0390 |

**Standard Form 298** (Rev. 8-98)
Prescribed by ANSI Std. Z39.18