MESHLESS LOCAL PETROV-GALERKIN METHOD FOR PLANE ELASTICITY PROBLEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENIZ CAN ERDAYI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

FEBRUARY 2014

Approval of the thesis:

MESHLESS LOCAL PETROV-GALERKIN METHOD FOR PLANE ELASTICITY PROBLEMS

submitted by **DENIZ CAN ERDAYI** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Süha Oral
Head of Department, **Mechanical Engineering** _____

Prof. Dr. Süha Oral
Supervisor, **Mechanical Engineering Department, METU** _____

Examining Committee Members:

Prof. Dr. Bülent Doyum
Mechanical Engineering Department, METU _____

Prof. Dr. Süha Oral
Mechanical Engineering Department, METU _____

Prof. Dr. Haluk Darendeliler
Mechanical Engineering Department, METU _____

Prof Dr. Serkan Dağ
Mechanical Engineering Department, METU _____

Assoc. Prof. Dr. Mustafa Uğur Polat
Civil Engineering Department, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: DENIZ CAN ERDAYI

Signature :

ABSTRACT

MESHLESS LOCAL PETROV-GALERKIN METHOD FOR PLANE ELASTICITY PROBLEMS

Erdayı, Deniz Can

M.S., Department of Mechanical Engineering

Supervisor : Prof. Dr. Süha Oral

February 2014, 79 pages

In this research, Meshless Local Petrov-Galerkin Method (MLPG) has been used in order to solve problems of elasto-statics. Moving least squares approximation (MLS) has been used to construct trial function. MLS weight function has been selected as test function. Thus, method is so-called MLPG1. Cantilever beam problem has been solved with MLPG. Effect of integration and influence domain sizes have been investigated for infinite plate with circular hole problem. Optimal parameters have been determined. Results have been compared with exact solution.

Keywords: Mesh-free methods, meshless local Petrov-Galerkin method, solid mechanics, elasticity, moving least squares methods

ÖZ

DÜZLEM ELASTİSİTE PROBLEMLERİ İÇİN AĞSIZ YEREL PETROV-GALERKİN YÖNTEMİ

Erdayı, Deniz Can
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Süha Oral

Şubat 2014 , 79 sayfa

Bu çalışmada elastostatik problemleri çözmek için ağsız yerel Petrov-Galerkin yöntemi kullanılmıştır. Deneme fonksiyonun oluşturulmasında hareketli en küçük kareler yöntemi kullanılmıştır. Test fonksiyonu, hareketli en küçük kareler yönteminin ağırlıklandırma fonksiyonu ile aynı seçilmiştir. Bu metoda MLPG1 denilmektedir. Ankastre çubuk analizi gerçekleştirilmiştir. Integral ve etki tanım kümelerinin çözüm üzerine etkileri delikli sonsuz plakalar için araştırılmış ve bu parametlerin en uygun değerleri tespit edilmiştir. Sonuçlar kesin çözüm ile karşılaştırılmak sureti ile değerlendirilmiştir.

Anahtar Kelimeler: Ağsız yöntemler, ağsız yerel Petrov-Galerkin yöntemi, katı mekaniği, elastisite, hareketli en küçük kareler yöntemi

to my wife, Laçin

ACKNOWLEDGMENTS

This thesis is made under the sincere guidance of Prof. Dr. Süha Oral. I would like to express my gratitude to my supervisor Prof. Dr. Süha Oral for the useful comments, remarks and engagement through the learning process of this master thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CPU	Central processing unit
DEM	Diffuse element method
EFG	Element-free Galerkin method
FEM	Finite element method
MFree	Mesh-free
MLPG	Meshless local Petrov-Galerkin method
MLPG1	Meshless local Petrov-Galerkin method 1
MLS	Moving least squares
PIM	Point interpolation method
RKPM	Reproducing kernel particle method
SPH	Smoothed particle hydrodynamics
WLS	Weighted least squares

LIST OF SYMBOLS

x	Coordinate
u	Displacement
ϵ	Strain
σ	Stress
b	Body force
Ω	Problem domain
Γ	Boundary of problem domain
$\bar{u}$	Prescribed displacement
$\bar{t}$	Prescribed traction
n	Surface normal
Γ_u	Essential boundary
Γ_t	Natural Boundary
E	Young's modulus
ν	Poisson's ratio
δ	Kronecker delta function
$\mathbf{D}$	Constitutive matrix
$\bar{E}$	Equivalent Young's modulus
$\bar{v}$	Equivalent Poisson's ratio
a_x, a_y	Integration domain size in x and y direction
b_I	Radius of influence domain
d_s	Nodal spacing
α_q	Integration domain size multiplier
α_I	Influence domain size multiplier
$\overline{W}$	Weighting function for MLS
u^h	Approximated displacement function
$\mathbf{p}$	vector of basis functions
$\mathbf{a}$	vector of coefficients for MLS
d	problem dimension

c	order of completeness
$\mathbf{A}$	Moment matrix of MLS
$\mathbf{C}$	C matrix of MLS
ϕ	Shape function
n	Number of support nodes
m	Number of monomials
W	Weight function for local symmetric weak form
α	Penalty parameter
Ω_I	Local integration domain for node I
Γ_I	Boundary of integration domain
Γ_{Ii}	Internal boundary of integration domain
Γ_{It}	Traction boundary of integration domain
Γ_{Iu}	Essential boundary of integration domain
$\mathbf{B}$	Strain matrix
$\mathbf{W}_I$	Matrix of shape functions for node I
$\mathbf{V}_I$	Matrix of derivatives of shape functions for node I
$\mathbf{K}_I$	Stiffness matrix of node I
$\mathbf{F}_I$	Force vector of node I
$\mathbf{K}$	Global stiffness matrix
$\mathbf{F}$	Global force vector
$\mathbf{E}$	Error

CHAPTER 1

INTRODUCTION

Methodology of solving solid mechanics problems begins with constructing a physical model. Firstly, environment and problem domain are defined, and idealizations are considered. Then, a convenient mathematical model is constructed. Mathematical modeling offers governing equations with boundary conditions for physical model. These governing equations can be solved either approximately by numerical methods or exactly by analytical methods.

Compared to the past, the problems of engineering are more complex. However, analytical methods offer solutions to relatively simple problems. On the other hand, numerical methods offer a practical way to solve these problems approximately. In numerical methods, more computational power is required in order to handle more complex problems and get more accurate results. Thanks to the recent developments on computer technologies, more complex problems can be solved more accurately in reasonable time.

Numerical methods have been developed for centuries in solid mechanics field. One of the well-known example is finite element method which is widely used in many engineering problems. It is classified as mesh-based method because a predefined element mesh is required in order to represent and discretize domain.

In addition to the mesh-based methods, another class of numerical methods is so-called mesh-free methods, which have been developed over past 40 years.

One of the noticeable properties of mesh-free methods is that predefined mesh is not required to discretize domain. Instead, points scattered to problem domain are used in order to represent the problem. Using scattered points instead of predefined mesh to represent problem domain enables to overcome some limitations and shortcomings of mesh-based methods which are discussed in following sections.

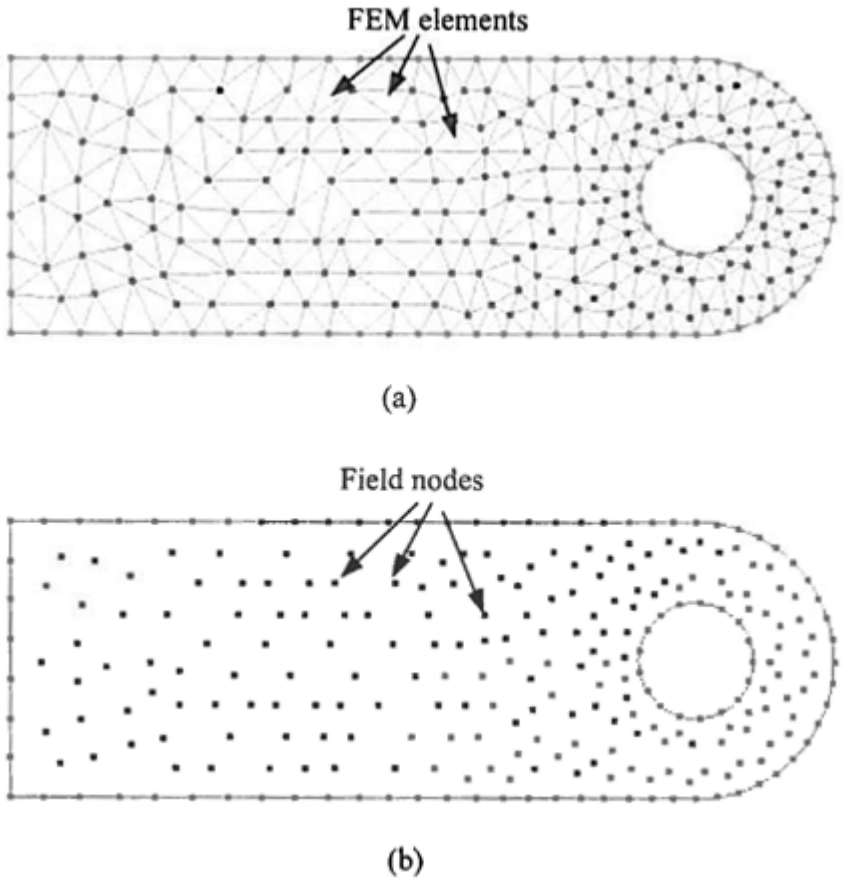

Figure 1.1: Domain representation: (a) FEM (b) Mesh-free methods [29]

1.1 Mesh-free Methods

Liu [29] defines mesh-free methods as following:

> "A MFree method is a method used to establish system algebraic equation for the whole problem domain without the use of a predefined mesh for the discretization."

Mesh-free methods use scattered nodes in problem domain and on its boundaries to represent the problem. Then, a complex problem is transformed to a system of algebraic equations. Two important steps of this transformation are function approximation and problem formulation.

Firstly, shape functions are constructed in function approximation step. In contrast to mesh-based methods, shape functions of mesh-free methods are constructed during the solution step. In other words, there is no predefined shape function. Field variables at any point within the problem domain is approximated using the values at field nodes within the support domain.

Support domain for a point includes the field nodes which are used for approximating the value at that point. Support domains may differ from one point to another and be in various shapes and sizes. The value of shape function is zero outside the support domain. Hence, mesh-free shape functions are locally supported. Some methods used in approximation are Moving Least Squares method (MLS), Weighted Least Squares method (WLS), Reproducing Kernel Particle Method (RKPM), and Point Interpolation Method (PIM).

Secondly, in formulation step results of mathematical model, namely governing differential equations and boundary conditions, are transformed in a system of linear algebraic equations. Two categories of formulation step are strong form formulation and weak form formulation. In strong form formulation, governing differential equations are satisfied at each field node. On the other hand, in weak formulation, integral functional of governing differential equations are satisfied. In other words, governing equations are satisfied averagely in problem domain.

1.2 Advantages of Mesh-free Methods

Mesh-free methods have certain advantages over mesh-based counterparts. Comparison between mesh-free and mesh-based methods in this section is based

on the studies presented by Nguyen et.all [40] and Li at all. [22]. A number of advantages are discussed at following paragraphs.

Firstly, mesh-based methods require mesh to discretize problem domain. Meshing of problems with complex geometry is not performed automatically by computers at all times. In other words, constructing a quality mesh is a semi-automatic process and requires human intervention. Conversely, in mesh-free methods point cloud is constructed by computers only, regardless of its complexity. [16]

Next, FEM requires re-meshing steps to handle some problems, such as large deformation and moving discontinuities problems. Although adaptive mesh generators are developed for this purpose, they have certain drawbacks. At first, re-meshing at each step extends CPU time significantly in three dimensional problems. Furthermore, mapping of field variables between meshes worsens accuracy of the problem. On the other hand mesh-free methods handle these problems in an efficient way, because connectivity between nodes is generated during the solution process and can be regenerated if required.

Moreover, derivatives of field functions, such as strain and stress, are not smooth in problem domain and discontinuous at element interfaces in FEM. Hence, post-processing and special treatments are required in order to obtain smooth derivatives. Conversely, required order of continuity should be determined easily at the beginning of solution process in mesh-free methods. As a result, smooth derivatives of field variables can be obtained.[23]

Furthermore, convergence rate of mesh-free methods are generally better than the mesh-based methods.[21]

As well as, h-adaptivity is implemented quite easily to mesh-free methods. Since connectivity is constructed in run-time, accuracy of the solution is in-

creased by adding nodes into required regions.

Finally, mesh alignment affects solution accuracy for moving boundary problems, such as crack growth or shear bands. Therefore, mesh alignment sensitivity analysis is required for mesh-based methods. On the other hand, it is not an issue for mesh-free methods because of absence of mesh.

1.3 Disadvantages of Mesh-free Methods

Compared to mesh-based methods, one of the disadvantages of mesh-free methods is high computational cost, of which main reasons are listed as follows:

- Higher integration schema is required in order to achieve required accuracy.
- For each integration point (gauss point) a number of additional processes are required, such as neighborhood searching and computation of shape function derivatives.
- Band width of mesh-free spare matrices is generally larger than the mesh-based counterparts.
- Stiffness matrix of some mesh-free methods, such as MLPG, is antisymmetric.

Another disadvantage of mesh-free methods is that imposing essential boundary conditions requires extra efforts due to lack of Kronecker delta property.

1.4 Objective of Study

Main objective of this study is to solve problems of elasto-statics with meshless Local-Petrov Galerkin (MLPG) method and investigate effect of parameters on solution accuracy. In order to achieve this task, following steps have been performed.

- In chapter 3, mathematical model for 3 dimensional and plane elastostatics are presented.
- In chapter 4, MLPG method is examined. Moving least squares approximation schema and local weak form formulation are presented detailedly. In addition, weighting functions and error norms are pointed out.
- In chapter 5, firstly a cantilever beam problem is solved in order to check correctness of formulation and computer implementation. Secondly, effect of integration and influence domain size for infinite plate with circular hole is investigated. Thirdly, solution of infinite plate with circular hole is presented for selected integration and influence domain size.
- In chapter 6, conclusions are presented.

CHAPTER 2

LITERATURE REVIEW

Studies on mesh-free methods started in early 1970s. First mesh-free method was smoothed particle hydrodynamics which is developed by Lucy [36] and Gingold, Monaghan [14] in 1977. SPH has been based on strong form formulation in Lagrangian description. Although SPH was originally used for simulation of astrophysics problems, it became a widely used method to solve fluid mechanics problems. Some of noteworthy researches about implementation of SPH to fluid dynamics problems were conducted by Monaghan [37] [38].

First implementation of SPH to solid mechanics problem is about dynamics of elastoplastic solids by Libersky,et all [24]. However, Libersky noted that there were two major issues about SPH. These issues are inaccurate results near boundaries and tension instability. Tension instability of SPH was first investigated by Swegle [42]. Other notable research about stability of SPH was perfomed by Belytschko, et all [7] and Xiao, et all [44]. In addition, improvements to SPH method were performed to solve impact problems by Johnson, et all [17] [18], Libersky,et all [25],Bonet et all. [9].

Another class of mesh-free methods is based on weak formulation. In strong forms, differential equations are satisfied at each field node, whereas in weak forms, integral equations are satisfied. These integral equations are integral functionals of variational principle, such as weighted least square functional.

After beginning of 1990s, research on mesh-free weak formulation are significantly increased after publication of Diffuse Element Method.

Diffuse element method, which is introduced by Nayroles et al. [39] is the first example of this class. Nayroles et.al modeled thermodynamics problem by DEM. DEM is the first method which use moving least squares approximation in galerkin weak form. It was originally offered as an improvement to FEM. However, DEM failed to pass patch tests because of simplification in formulation.

Afterwards, Belytschko et all. discovered that if approximation schema is moving least square approximation, than accurate computation of derivatives and integrals in Galerkin formulation is critical for solution quality . Accordingly, they modified and refined DEM and named it as element-free Galerkin method (EFG). Study was published in 1994 [8].

Belytschko et al. applied EFG to solve elasticity and crack growth problems. They argue that EFG is highly applicable and gives accurate results for crack grow problems.[35][34].

Reproducing Kernel Particle Methods is yet another mesh-free method based on global weak formulation and proposed by Liu et all in 1995 [33]. Liu et all. used correction functions in SPH approximation in order to assure required consistency.

RKPM has been applied to various problems. Solution to large deformation problems proposed by Chen at all.[10] [11] and Liu [31]. As well as fluid dynamics application of RKPM is also established by Liu et.all [32].

In global weak forms, such as EFG and RKPM, mesh is not required to approximate field variable. However in order to integrate energy functional, such

as Galerkin weak form for EFG, a mesh network is required in global problem domain which is generally called shadow elements.

In order to eliminate background meshing process in global weak formulation, local weak formulation was evolved. Difference between global weak form and local weak form is that for global weak form, energy integral is computed over global domain whereas in local weak form, energy integral is computed over sub-domains which may be overlapping and union of them represents the global problem domain.

Atluri et all. proposed MLPG method based on local weak formulation[4]. In MLPG, energy integration can be evaluated over local domain around each points instead of global domain. These local sub-domains are regularly shaped for internal ones (depending on problem dimensions, they can be circle, rectangle, sphere) and there is no compatibility requirement between these internal sub-domains. In other words it is not important that whether these domains overlap or not. Only local sub-domains which intersect with global boundary are locally meshed in order to represents boundary.

Atluri claims that MLPG method is truly meshless method because background mesh is not required for either function approximation or energy integration, however global weak forms need a background mesh for integration purpose.[4]

Liu states that MLPG is one of the widely used mesh-free method[28]. Many researchers, especially Atluri et al., have done improvements and implementations to different problems since first introduction of MLPG. A number of notable studies are listed as follows:

- Solution to linear Poisson's equation by Atluri et all. in 1998[4]
- Critical evaluation of basic framework of MLPG with numerical examples by Atluri et al. in 1999 [2]

- Solution to thin beams, 4^{th} order boundary value problems by Atluri et al. in 1999 [1]
- Solution to problems of elasto-statics by Atluri et all. in 2000 [5][6]
- Solution to elastodynamics problems by Gu,and Liu in 2001[15]
- Solution to shear deformable beams by Cho et all. in 2001 [13]
- Solution to thin plate bending by Long and Atluri in 2002 [41]
- Solution to linear crack problem by Ching and Batra in 2001 [12]
- Error analysis by Kim and Atluri in 2000 [19]
- Solution to fluid mechanics problem by Lin and Atluri in 2001 [26] [27],Liu et all [30]

MLPG can use any test function and mesh-free function approximation procedure in solution process. Atluri and Shen defined 6 different MLPG schema in [3] based on variety of trial and test function . One of these is MLPG1 whose test function is same as the weight function of moving least squares approximation.

CHAPTER 3

MATHEMATICAL MODEL OF PROBLEM

3.1 Problems of Linear Elasto-statics

For linear elasto-statics problems it is assumed that loading is sufficiently slow. Hence, there is no dynamic effect and accelerations are negligible. It is known as quasi-static process. Another consideration is that material is isotropic and homogeneous.

In addition, it is assumed that displacements and displacements gradients are sufficiently small and stress is a linear function of strain. Therefore, infinitesimal strain tensor is used to describe deformation in linear elasto-statics.

Three dimensional problem of linear elasticity shown in figure 3.1 are formulated in following sub-sections.

3.1.1 Strain - Displacement Relations

Strain tensor of linear elasto-static problem is infinitesimal strain tensor and defined in terms of displacements as

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{3.1}$$

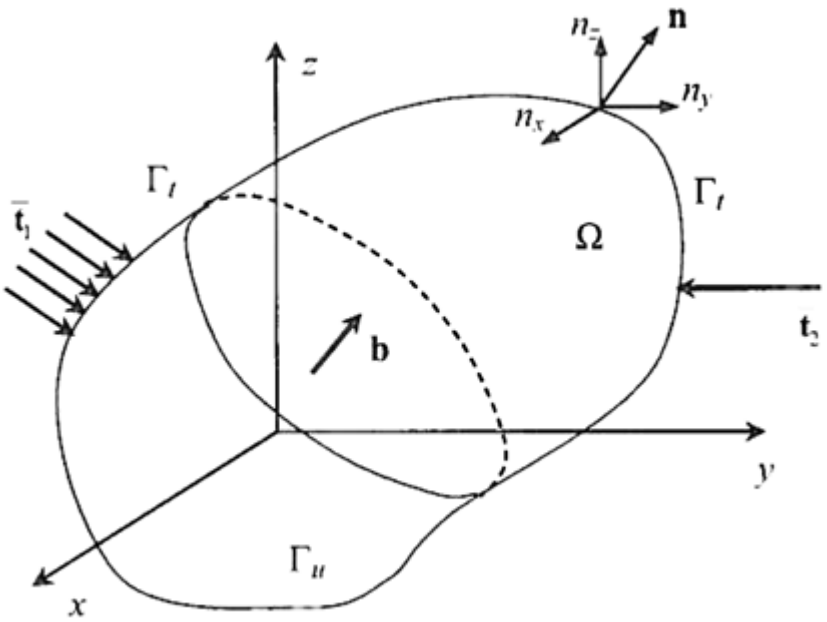

Figure 3.1: Three Dimensional Elasticity Problem

3.1.2 Equilibrium Equations

For quasi-static process, equilibrium equations in domain Ω bounded by the boundary Γ are

$$\sigma_{ij,j} + b_j = 0 \quad \text{in} \quad \Omega \tag{3.2}$$

Corresponding boundary conditions are

$$u_i = \bar{u}_i \quad \text{on} \quad \Gamma_u \tag{3.3a}$$

$$\sigma_{ij} n_j = t_i = \bar{t}_i \quad \text{on} \quad \Gamma_t \tag{3.3b}$$

where $\bar{u}_i$ is prescribed essential boundary condition, and $\bar{t}_i$ is prescribed natural boundary condition as shown in figure 3.1.

3.1.3 Constitutive Law

Hooke's law for isotropic bodies in terms of elastic constants for isothermal process is

$$\sigma_{ij} = \frac{E}{1+\nu}\left(\epsilon_{ij} + \frac{\nu}{1-2\nu}\delta_{ij}\epsilon_{kk}\right) \tag{3.4}$$

3.2 Two Dimensional Formulation of Elasto-statics

In this section, foregoing equations of general elasto-static problem is simplified for two dimensional problems. Two different assumptions can be considered for two dimensional problems,namely plane stress and plane strain assumptions. For plane stress assumption, it is assumed that through thickness stresses σ_{zz}, σ_{xz}, σ_{yz} are zero. For plane strain assumption, it is assumed that through thickness displacement u_z are zero or constant.

Following equations are transformed from indicial form to matrix form in order to construct matrix form of system algebraic equations.

For two dimensional problems, stress vector is

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} \tag{3.5}$$

For two dimensional problems strain vector is

$$\boldsymbol{\epsilon} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{Bmatrix} \tag{3.6}$$

Strain displacement relation is

$$\boldsymbol{\epsilon} = \mathbf{L}\mathbf{u} \tag{3.7}$$

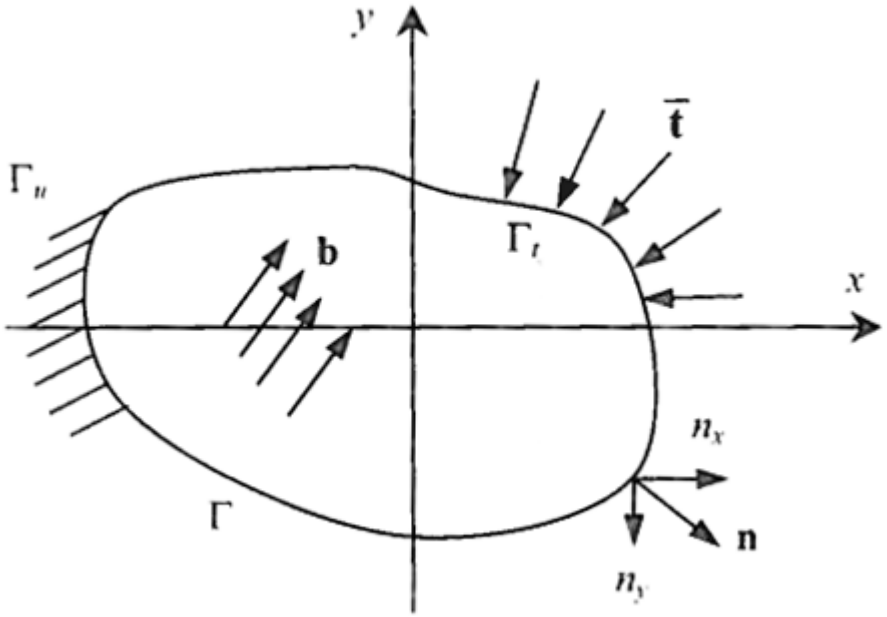

Figure 3.2: Two Dimensional Elasticity Problem

where **u**, displacement vector, is

$$\mathbf{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{3.8}$$

and **L**, differential operator, is

$$\mathbf{L} = \begin{Bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{Bmatrix} \tag{3.9}$$

Constitutive matrix for isotropic materials in terms of equivalent elastic constants is

$$\mathbf{D} = \frac{\bar{E}}{1-\bar{v}^2} \begin{Bmatrix} 1 & \bar{v} & 0 \\ \bar{v} & 1 & 0 \\ 0 & 0 & (1-\bar{v})/2 \end{Bmatrix} \tag{3.10}$$

where

$$\bar{E} = \begin{cases} E, & \text{for plane stress} \\ \dfrac{E}{1-v^2}, & \text{for plane strain} \end{cases} \tag{3.11}$$

and

$$\bar{v} = \begin{cases} v, & \text{for plane stress} \\ \dfrac{v}{1-v}, & \text{for plane strain} \end{cases} \tag{3.12}$$

Stress vector is written in terms of displacements as follows:

$$\boldsymbol{\sigma} = \mathbf{D}\,\mathbf{L}\,\mathbf{u} \tag{3.13}$$

For two dimensional linear elasto-static problems shown in figure 3.2, equilibrium equations in matrix form are

$$\mathbf{L}\boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{on} \quad \Omega \tag{3.14}$$

where $\mathbf{b}$, body force vector, is

$$\mathbf{b} = \begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \tag{3.15}$$

Boundary conditions of foregoing equilibrium equation are

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on} \quad \Gamma_u \tag{3.16a}$$

$$\mathbf{t} = \mathbf{n}\boldsymbol{\sigma} = \bar{\mathbf{t}} \quad \text{on} \quad \Gamma_t \tag{3.16b}$$

where $\mathbf{n}$, matrix of unit outward normals, is

$$\mathbf{n}_{[2\times 3]} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \tag{3.17}$$

Prescribed displacements in vector form, $\bar{\mathbf{u}}$, is

$$\bar{\mathbf{u}} = \begin{Bmatrix} \bar{u}_x \\ \bar{u}_y \end{Bmatrix} \tag{3.18}$$

Prescribed tractions in vector form, $\bar{\mathbf{t}}$, is

$$\bar{\mathbf{t}} = \begin{Bmatrix} \bar{t}_x \\ \bar{t}_y \end{Bmatrix} \tag{3.19}$$

CHAPTER 4

MESHLESS LOCAL PETROV-GALERKIN METHOD (MLPG)

Various form of MLPG are presented by Atluri. Both of them are based on Petrov-Galerkin formulation in local sub-domains. In this paper, MLPG1 is used in order to solve problems of elasto-statics. Properties of MLPG1, which is distinguished from other MLPGs, are presented in succeeding paragraphs.

Four main steps of MLPG are listed as

1. Domain representaion
2. Function approximation
3. Formation of system equation
4. Solution to system of linear algebraic equations

Firstly, problem domain and boundaries are represented by set of scattered nodes in MLPG. Density of nodes effects accuracy of solution. If higher accuracy is required for a particular region, then nodal density should be increased for that region. Nodal density does not have to be uniform in MLPG.

Secondly, in function approximation step, value of field variable at any point in local sub-domain is represented by values of finite set of field nodes in local support domain of the point. Thus, field variables in local sub-domain is continuously defined in terms of values of field nodes. This continuous function

is known as trial function. Moving least square method is used for function approximation in MLPG1.

Thirdly, local symmetric weak formulation is used in order to find system of linear algebraic equations. Local symmetric weak form is a variant of weighted residual method. Idea of MLPG is that weighted residual integral is evaluated at a small local sub-domain of field nodes. In other words, weighted residual integral are satisfied in each field node. Then algebraic equation sets of each field node are assembled in order to find system of equations. In Petrov-Galerkin formulation, test and trial function domains do not have to be same. On the other hand, MLPG1 use same form of weight function for test and trial function but with different domains (domain size and shapes).

Fourthly, system of linear algebraic equations are solved. For solid mechanics problems, outputs of solution are displacements. However, due to lack of Kronecker delta property of MLS , these displacements are fictitious. Thus, displacement value of a point is approximated by MLS with respect to fictitious displacement values of field nodes in support domain of the point.

4.1 Domain Definitions

There are 4 domain definitions for MLPG method in this study. These are listed as follows:

- Problem domain
- Integration domain
- Influence Domain
- Support Domain

All domains except problem domain are defined with respect to nodal spacing of field nodes.

4.1.1 Nodal Spacing

Nodal spacing may differ in problem domain due to non-uniform node distribution. Nodal space, d_s, for a field node is minimum distance between the node and its neighborhoods.

4.1.2 Problem Domain

It is the problem domain, Ω, bounded by Γ in equation (3.2). Domain and its boundary are illustrated in 4.1.

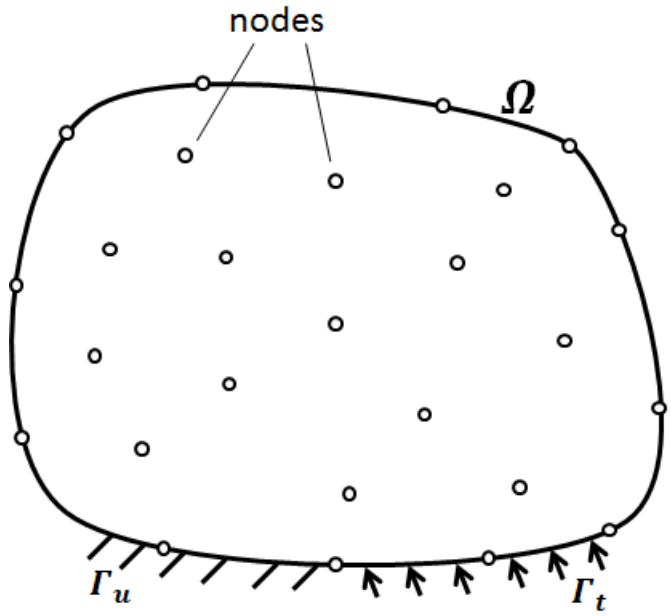

Figure 4.1: Problem domain Ω and nodes

4.1.3 Integration Domain

It is the local sub-domain, Ω_I, bounded by Γ_I for node I. For each node, this domain is costructed in order to evaluate integral in equation (4.23). Integration domain for node K and I are shown in figure 4.2. Integration domain of node K is fully inside the problem domain. On the other hand, integration domain of node I intersects with global boundary.

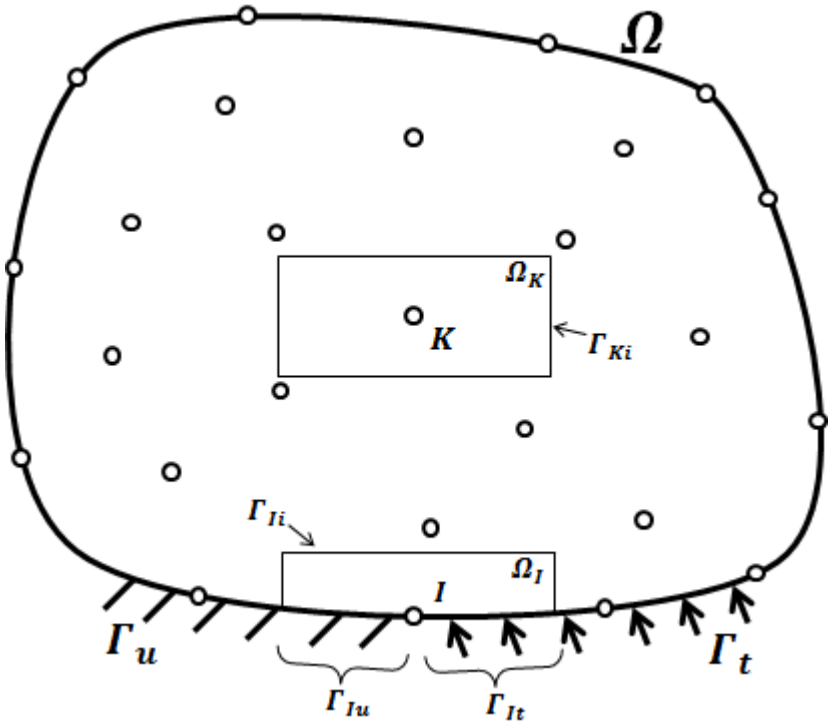

Figure 4.2: Integration domain for node I and K

Integration domain sizes (fig. 4.3) for node I are

$$a_x = \alpha_q \times d_s \tag{4.1a}$$

$$a_y = \alpha_q \times d_s \tag{4.1b}$$

where d_s is nodal spacing of the field node and α_q is the integration domain size multiplier. α_q is a analysis parameter and determined by user.

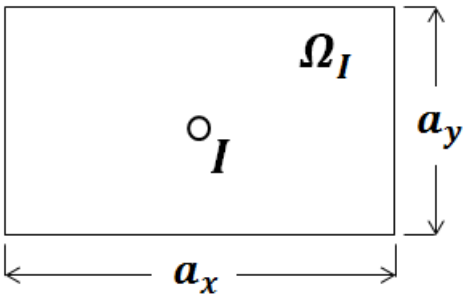

Figure 4.3: Integration domain size for node I

For each integration domain, there are gauss quadrature points in order to evaluate integral in equation (4.23) numerically. Gauss points distribution in integral domain of node I is illustrated in figure 4.4.

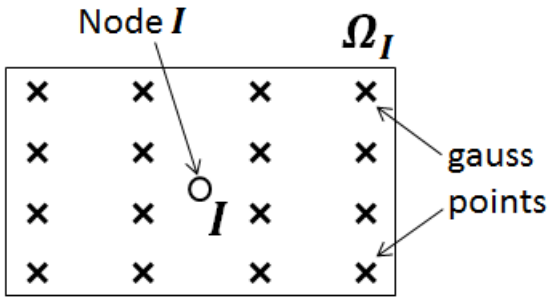

Figure 4.4: Gauss points of integration domain for node I

4.1.4 Influence and Support Domain

For each node, influence domain is constructed. Influence domain of a node defines a region in which the node contributes to field variable (displacement) approximation.

For instance, in figure 4.5 influence domains of nodes **I**, **K** and **L** is plotted. Nodes **I**, **K** and **L** contribute to field variable (displacement) approximation at gauss point **Q**, because gauss point is in the influence domain of these nodes. On the other hand, nodes **I** and **K** contribute to field variable (displacement) approximation at gauss point **P**, because gauss point is not in the influence domain of node **P**.

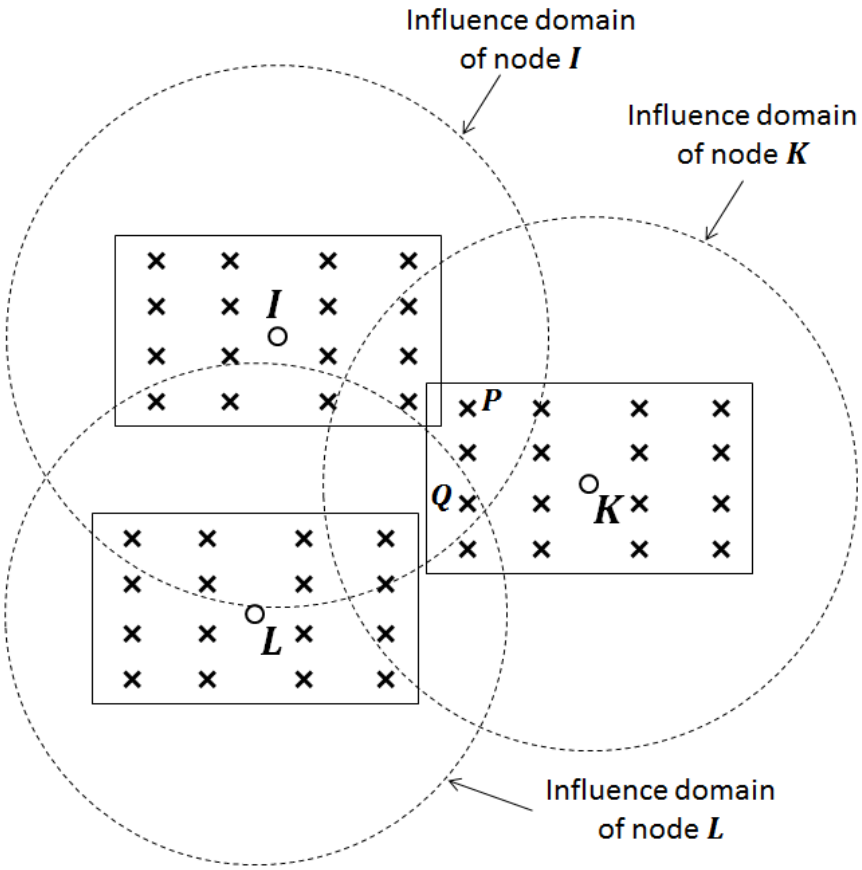

Figure 4.5: Influence domains

Influence domain is for node, and the corresponding domain for gauss point is support domain. Support domain of a gauss point includes nodes which contributes field variable approximation at that gauss point. For instance, in figure 4.5 support domain of gauss point **Q** includes nodes **I**, **K** and **L**. On the other hand, support domain of gauss point **P** includes nodes **I** and **K**.

Radius of influence domain for a field node is

$$b_I = \alpha_I \times d_s \tag{4.2}$$

where d_s is nodal spacing of the field node and α_I is the influence domain size multiplier. α_I is a analysis parameter and determined by user.

4.2 Moving Least Squares (MLS)

Moving least square approximation is used to construct shape functions of MLPG. Method was originally used for smoothing and interpolating scattered data by Lancaster at [20].

Objective of MLS is to construct a continuous function, $u^h(\mathbf{x})$, of field variable with respect to fictitious scalar values $\mathbf{u_i}$ at points $\mathbf{x}_i$ in a reasonable error as shown in figure 4.6 .

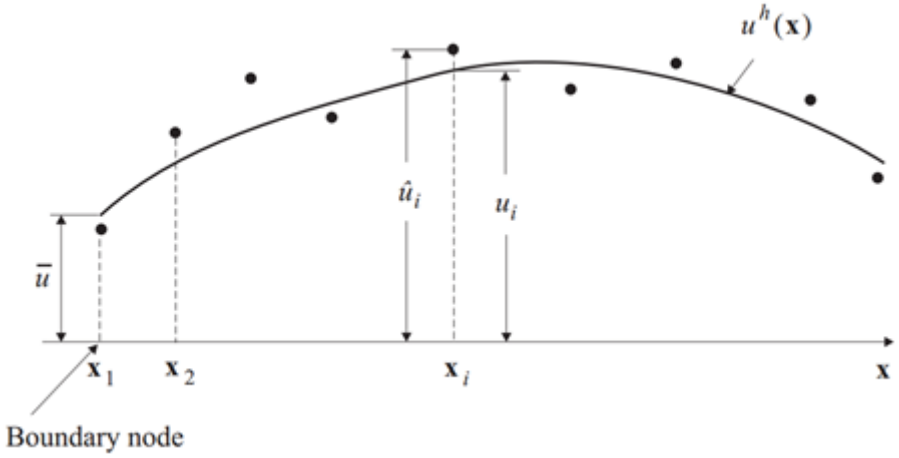

Figure 4.6: Function approximation by moving least square

Thus, weighted least-square error function is constructed as follows:

$$J = \sum_{i}^{n} \overline{W}(\|\mathbf{x} - \mathbf{x}_i\|) \, \|u(x_i) - u_i\|^2 \tag{4.3}$$

where $\overline{W}$ is weighting function, x_i is position of support node, and u_i is value of support node .

Minimizing error function, given in (4.3), proposes solution to approximation problem.

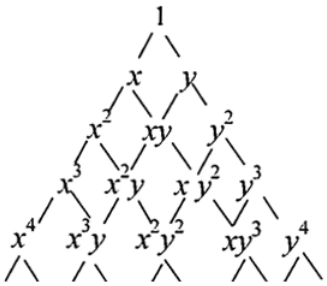

Figure 4.7: Pascal Triangle for 2 parameters; namely x and y

Unknown function, $u(\mathbf{x})$, is approximated as follows:

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{4.4}$$

where $\mathbf{p}^T(\mathbf{x})$ is vector of basis functions, $\mathbf{a}(\mathbf{x})$ is vector of unknown coefficients, m is number of basis function.

For two dimensional problem, spatial coordinates are

$$\mathbf{x}^T = \begin{Bmatrix} x & y \end{Bmatrix} \tag{4.5}$$

In order to ensure minimum completeness requirement, complete set of monomials from Pascal triangle (figure 4.7) are used as basis functions.

There are 6 monomial for two dimensional problems to provide 2^{nd} order completeness. Vector form of these basis functions is

$$\mathbf{p}^T(\mathbf{x}) = \begin{Bmatrix} 1 & x & y & x^2 & xy & y^2 \end{Bmatrix}_{[1\times 6]} \tag{4.6}$$

Note that number of monomials can be found by

$$\text{number of monomials} = \frac{(d+c)!}{d!\,c!} \tag{4.7}$$

where d is dimension of space and c is order of completeness.

Inserting approximated function in equation 4.4 into equation 4.3 gives

$$J = \sum_{i}^{n} \overline{W}(\|\mathbf{x} - \mathbf{x}_i\|) \, \|\mathbf{p}^T(\mathbf{x_i})\mathbf{a}(\mathbf{x}) - u_i\|^2 \tag{4.8}$$

where n is the number of data points.

Weighted, discrete, L_2 norm given in equation (4.8) is minimized if partial derivatives of function with respect to unknown coefficients is set to zero. This leads the following linear relation set.

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{C}(\mathbf{x})\mathbf{U_s} \tag{4.9}$$

where $\mathbf{U}_s$ is vector of fictitious field variables in support domain.

$$\mathbf{U}_s = \begin{Bmatrix} u_1 & u_2 & \cdots & u_n \end{Bmatrix}^T \tag{4.10}$$

Symmetric matrix $\mathbf{A}(\mathbf{x})$, weighted moment matrix, is

$$\mathbf{A}(\mathbf{x}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \overline{\mathbf{W}}_{\mathbf{i}}(\mathbf{x} - \mathbf{x_i})\, \mathbf{p}(\mathbf{x_i})\, \mathbf{p^T}(\mathbf{x_i}) \tag{4.11}$$

For two dimensional problem and 2nd order completeness with n support points, $\mathbf{A}(\mathbf{x})$ is

$$\begin{aligned}
\mathbf{A}(\mathbf{x})_{[6\times6]} = \overline{W}_1(x - x_1) & \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ x_1 & x_1^2 & x_1y_1 & x_1^3 & x_1^2y_1 & x_1y_1^2 \\ y_1 & x_1y_1 & y_1^2 & x_1^2y_1 & x_1y_1^2 & y_1^3 \\ x_1^2 & x_1^3 & x_1^2y_1 & x_1^4 & x_1^3y_1 & x_1^2y_1^2 \\ x_1y_1 & x_1^2y_1 & x_1y_1^2 & x_1^3y_1 & x_1^2y_1^2 & x_1y_1^3 \\ y_1^2 & x_1y_1^2 & y_1^3 & x_1^2y_1^2 & x_1y_1^3 & y_1^4 \end{bmatrix}_{[6\times6]} \\
+ \overline{W}_2(x - x_2) & \begin{bmatrix} 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \\ x_2 & x_2^2 & x_2y_2 & x_2^3 & x_2^2y_2 & x_2y_2^2 \\ y_2 & x_2y_2 & y_2^2 & x_2^2y_2 & x_2y_2^2 & y_2^3 \\ x_2^2 & x_2^3 & x_2^2y_2 & x_2^4 & x_2^3y_2 & x_2^2y_2^2 \\ x_2y_2 & x_2^2y_2 & x_2y_2^2 & x_2^3y_2 & x_2^2y_2^2 & x_2y_2^3 \\ y_2^2 & x_2y_2^2 & y_2^3 & x_2^2y_2^2 & x_2y_2^3 & y_2^4 \end{bmatrix}_{[6\times6]} \\
+ \quad \cdots \quad + \overline{W}_n(x - x_n) & \begin{bmatrix} 1 & x_n & y_n & x_n^2 & x_ny_n & y_n^2 \\ x_n & x_n^2 & x_ny_n & x_n^3 & x_n^2y_n & x_ny_n^2 \\ y_n & x_ny_n & y_n^2 & x_n^2y_n & x_ny_n^2 & y_n^3 \\ x_n^2 & x_n^3 & x_n^2y_n & x_n^4 & x_n^3y_n & x_n^2y_n^2 \\ x_ny_n & x_n^2y_n & x_ny_n^2 & x_n^3y_n & x_n^2y_n^2 & x_ny_n^3 \\ y_n^2 & x_ny_n^2 & y_n^3 & x_n^2y_n^2 & x_ny_n^3 & y_n^4 \end{bmatrix}_{[6\times6]}
\end{aligned} \tag{4.12}$$

$\mathbf{C}(\mathbf{x})$ is a antisymmetric matrix and defined as follows:

$$\mathbf{C}(\mathbf{x}) = \left[\overline{W}_1(\mathbf{x})\mathbf{p}(\mathbf{x_1}) \quad \overline{W}_2(\mathbf{x})\mathbf{p}(\mathbf{x_2}) \quad \cdots \quad \overline{W}_n(\mathbf{x})\mathbf{p}(\mathbf{x_n})\right]_{[6\times n]} \tag{4.13}$$

For two dimensional problem and 2^{nd} order completeness with n support points, $\mathbf{C}(\mathbf{x})$ is

$$\mathbf{C}(\mathbf{x})_{[6\times n]} = \begin{bmatrix} \overline{W}_1\,1 & \overline{W}_2\,x_2 & \cdots & \overline{W}_n\,x_n \\ \overline{W}_1\,x_1 & \overline{W}_2\,x_2^2 & \cdots & \overline{W}_n\,x_n^2 \\ \overline{W}_1\,y_1 & \overline{W}_2\,x_2y_2 & \cdots & \overline{W}_n\,x_ny_n \\ \overline{W}_1\,x_1^2 & \overline{W}_2\,x_2^3 & \cdots & \overline{W}_n\,x_n^3 \\ \overline{W}_1\,x_1y_1 & \overline{W}_2\,x_2^2y_2 & \cdots & \overline{W}_n\,x_n^2y_n \\ \overline{W}_1\,y_1^2 & \overline{W}_2\,x_2y_2^2 & \cdots & \overline{W}_n\,x_ny_n^2 \end{bmatrix}_{[6\times n]} \tag{4.14}$$

In order to find unknown coefficients $\mathbf{a}(\mathbf{x})$, each side of equation (4.9) is multiply by inverse of $\mathbf{A}(\mathbf{x})$.

Then, $\mathbf{a}(\mathbf{x})$ is

$$\mathbf{a}(\mathbf{x})_{[6\times1]} = \mathbf{A}(\mathbf{x})^{-1}_{[6\times6]}\mathbf{C}(\mathbf{x})_{[6\times n]}\mathbf{U}_{\mathbf{s}[n\times1]} \tag{4.15}$$

Substituting coefficients vector, $\mathbf{a}(\mathbf{x})$, into equation (4.4) gives

$$u^h(\mathbf{x})_{[1\times1]} = \mathbf{p}^T(\mathbf{x})_{[1\times6]}\mathbf{A}(\mathbf{x})^{-1}_{[6\times6]}\mathbf{C}(\mathbf{x})_{[6\times n]}\mathbf{U}_{\mathbf{s}[n\times1]} \tag{4.16}$$

which is the solution of approximation problem.

Equation (4.16) is rearranged as follows:

$$u^h(\mathbf{x}) = \mathbf{\Phi}_s(\mathbf{x})\mathbf{U_s} \tag{4.17}$$

where $\mathbf{\Phi}_s(\mathbf{x})$, vector of shape functions, is

$$\mathbf{\Phi}_s(\mathbf{x}) = \left\{\phi_1(\mathbf{x}) \quad \phi_2(\mathbf{x}) \quad \cdots \quad \phi_n(\mathbf{x})\right\} = \mathbf{p}^T(\mathbf{x})\mathbf{A}(\mathbf{x})^{-1}\mathbf{C}(\mathbf{x}) \tag{4.18}$$

Shape function, $\phi_i(\mathbf{x})$, corresponding to the i^{th} node is

$$\phi_i = P_k{}^T(\mathbf{x}) A_{kj}{}^{-1}(\mathbf{x}) C_{ji}(\mathbf{x}) \qquad \begin{matrix} i = 1..n \\ j = 1..m \\ k = 1..m \end{matrix} \tag{4.19}$$

Field variable vector, $\mathbf{U_s}$, in equation (4.17) contains only one field variable,namely u. One should extend formulation to 2 variables, namely u and v for 2 dimensional problems. Then, equation (4.17) is rearranged as follows:

$$\mathbf{u}^h(\mathbf{x}) = \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \cdots & \phi_n & 0 \\ 0 & \phi_1 & 0 & \phi_2 & \cdots & 0 & \phi_n \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{Bmatrix} = \underset{[2\times 2n]}{\mathbf{\Phi}} \; \underset{[2n\times 1]}{\mathbf{u}} \tag{4.20}$$

where $\mathbf{u}$ is the ordered vector of two displacement components of field nodes in support domain.

Note that matrix $\mathbf{A}(\mathbf{x})$ must have an inverse. Thus, condition of weighted moment matrix, $\mathbf{A}(\mathbf{x})$, is crucial for solution. Some reasons of ill condition are

- Insufficient number of nodes in support domain of any gauss point. $n < m$
- Special nodal configurations, such as linear arrangement of all support nodes

4.2.1 Weight Function

Cubic spline function is used as weighting function in MLS. The cubic spline weight function is

$$\overline{W}_i(\mathbf{x}) = \begin{cases} 2/3 - 4r_i^2 + 4r_i^3, & r_i \leq 0.5 \\ 4/3 - r_i + 4r_i^2 - 4/3r_i^3, & 0.5 < r_i \leq 1 \\ 0, & r_i > 1 \end{cases} \tag{4.21}$$

and

$$r_i = \frac{\sqrt{(x - x_i)^2 + (y - y_i)^2}}{b_I} \tag{4.22}$$

where b_I is the radius of influence domain, x and y are coordinates of gauss point, x_i and y_i are coordinates of node.

In figure 4.8, cubic spline weight function with respect to r_i is plotted. Weight function should have smooth derivatives up to n^{th} order for $2n^{\text{th}}$ order governing differential equation.

Governing differential equations of elasto-statics are 2^{nd} order. Therefore, requirement for weight function is that it has smooth derivatives up to 1^{st}. In figure 4.9, 4.10 , first and second order derivatives of cubic spline weight function with respect to r_i are plotted. Although second derivative is not smooth, first derivative of function is smooth and this is sufficient condition for elasto-statics.

Cubic spline weight function fulfill following requirements.[29]

- Weight function is positively defined within the support domain.
- Weight function equals to zero outside the support domain.
- Weight function decreases when move away from point of interest.
- Weight function is sufficiently smooth for derivative operation in required order.

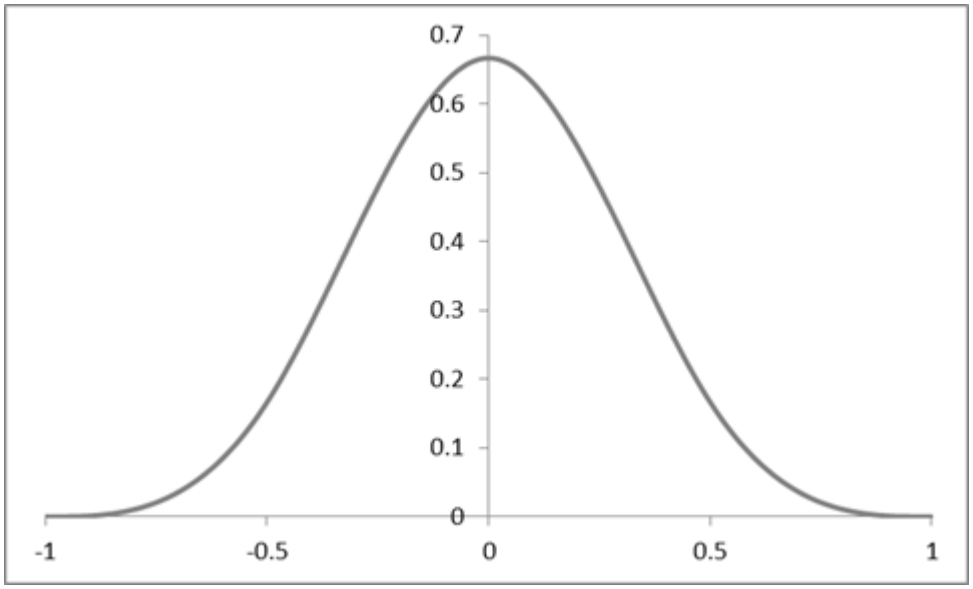

Figure 4.8: Cubic spline weight function with respect to r_i

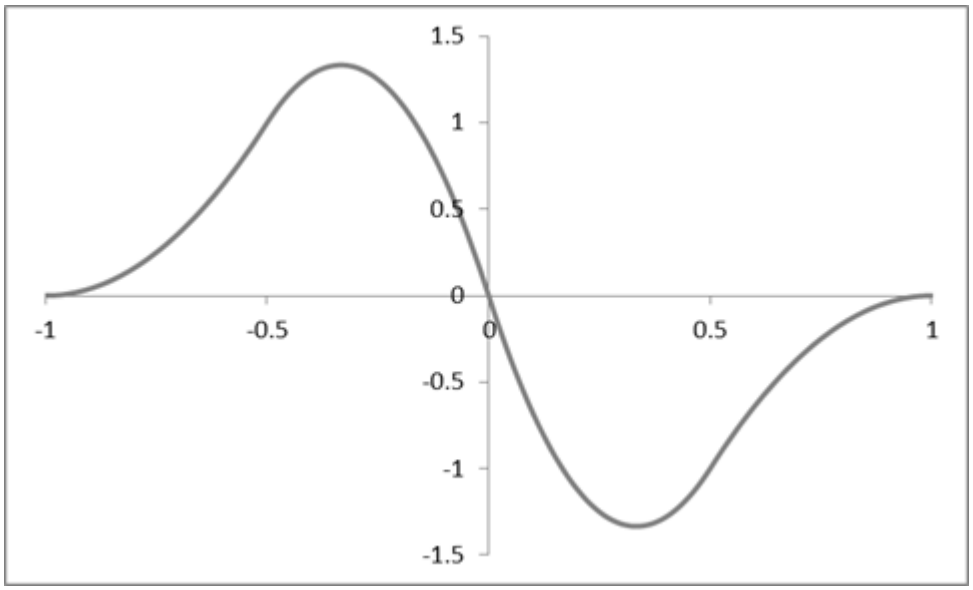

Figure 4.9: First derivative of cubic spline weight function with respect to r_i

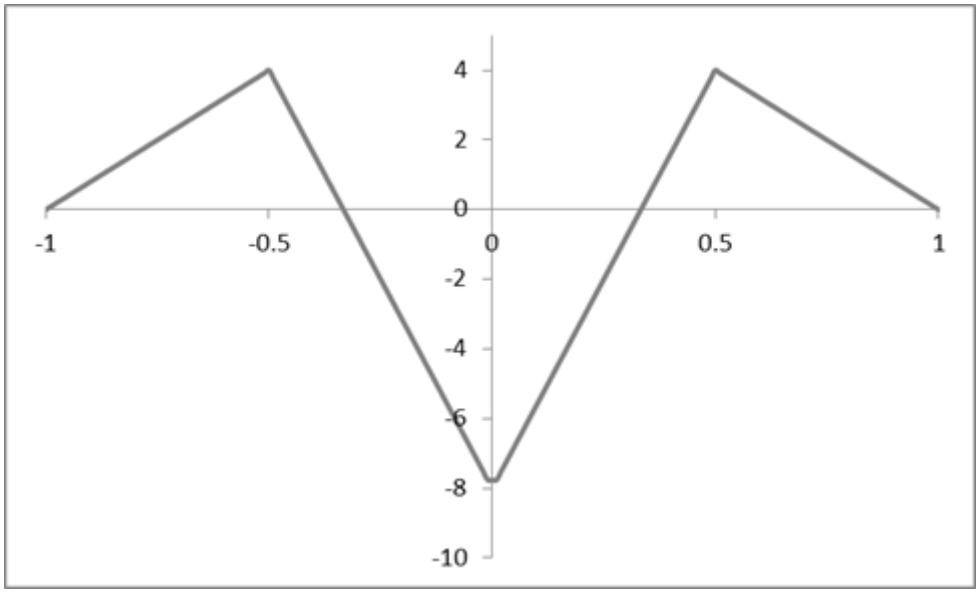

Figure 4.10: Second derivative of cubic spline weight function with respect to r_i

4.3 Local Symmetric Weak Form

Atluri [5] presented local symmetric weak form for node **I** over local sub domain Ω_I for problem of linear elasto-statics as follows:

$$\int_{\Omega_I} W_I(\sigma_{ij,j} + b_i)d\Omega - \alpha \int_{\Gamma_I} W_I(u_i - \bar{u}_i)d\Gamma = 0 \tag{4.23}$$

where Ω_I is integration domain for I^{th} node, Γ_I is boundary of Ω_I, W is test function, and $\alpha \gg 1$ is penalty parameter in order to impose essential boundary conditions.

Because of lack of knocker delta property of MLS shape function, penalty parameter is used to impose BCs.

By divergence theorem, equation (4.23) is

$$\int_{\Gamma_I} W_I n_j \sigma_{ij} d\Gamma - \int_{\Omega_I} W_{I,j}\sigma_{ij}d\Omega + \int_{\Omega_I} W_I b_i d\Omega - \alpha \int_{\Gamma_{Iu}} W_I(u_i - \bar{u}_i)d\Gamma = 0 \tag{4.24}$$

where $\mathbf{n}$ is unit outward normal of boundry and n_j is jth component of this

vector.

There are three possibilities for boundary of the local integration domain Γ_q, namely:

1. Internal boundary Γ_{Ii} : There is no intersection between local integration boundary and global boundary
2. Global boundary
   (a) Traction boundary Γ_{It} : Local integration domain intersect with global traction boundary
   (b) Essential boundary Γ_{Iu} : Local integration domain intersect with global essential boundary

Then, equation (4.24) can be rewritten as follows:

$$\int_{\Gamma_{Ii}} W_I n_j \sigma_{ij} d\Gamma + \int_{\Gamma_{Iu}} W_I n_j \sigma_{ij} d\Gamma + \int_{\Gamma_{It}} W_I n_j \sigma_{ij} d\Gamma - \int_{\Omega_I} W_{I,j} \sigma_{ij} d\Omega + \int_{\Omega_I} W_I b_i d\Omega - \alpha \int_{\Gamma_{Iu}} W_I (u_i - \bar{u}_i) d\Gamma = 0 \quad (4.25)$$

Traction vector is

$$\sigma_{ij} n_j = t_i \quad (4.26)$$

Substituting equation (4.26) and boundary condition defined in (3.16b) into (4.25) gives

$$\int_{\Omega_I} W_{I,j} \sigma_{ij} \, d\Omega - \int_{\Gamma_{Ii}} W_I n_j \sigma_{ij} \, d\Gamma - \int_{\Gamma_{Iu}} W_I n_j \sigma_{ij} \, d\Gamma = + \int_{\Omega_I} W_I b_i \, d\Omega + \int_{\Gamma_{It}} W_I \bar{t}_i \, d\Gamma - \alpha \int_{\Gamma_{Iu}} W_I (u_i - \bar{u}_i) \, d\Gamma \quad (4.27)$$

Equations (4.27) is in indicial form and for two dimensional problems, two equations are obtained for a field node. Equation (4.27) is tranformed to matrix form as follows:

$$\int_{\Omega_I} \mathbf{V}_I \boldsymbol{\sigma}\, d\Omega - \int_{\Gamma_{Ii}} \mathbf{W}_I \mathbf{n}\boldsymbol{\sigma}\, d\Gamma - \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{S}\mathbf{n}\boldsymbol{\sigma}\, d\Gamma + \alpha \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{S}\mathbf{u}^h\, d\Gamma$$
$$= + \int_{\Omega_I} \mathbf{W}_I \mathbf{b}\, d\Omega + \int_{\Gamma_{It}} \mathbf{W}_I \bar{\mathbf{t}}\, d\Gamma + \alpha \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{S}\bar{\mathbf{u}}\, d\Gamma \quad (4.28)$$

where matrix of weight functions is

$$\mathbf{W}_{I\,[2\times 2]} = \begin{bmatrix} W(\mathbf{x}, \mathbf{x}_I) & 0 \\ 0 & W(\mathbf{x}, \mathbf{x}_I) \end{bmatrix} \quad (4.29)$$

and matrix of derivatives of weight functions is

$$\mathbf{V}_{I\,[3\times 2]} = \begin{bmatrix} W_{,x}(\mathbf{x}, \mathbf{x}_I) & 0 & W_{,y}(\mathbf{x}, \mathbf{x}_I) \\ 0 & W_{,y}(\mathbf{x}, \mathbf{x}_I) & W_{,x}(\mathbf{x}, \mathbf{x}_I) \end{bmatrix} \quad (4.30)$$

and traction vector on boundary is

$$\bar{\mathbf{t}}_{[2\times 1]} = \mathbf{n}\bar{\boldsymbol{\sigma}} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \begin{Bmatrix} \bar{\sigma}_{xx} \\ \bar{\sigma}_{yy} \\ \bar{\sigma}_{xy} \end{Bmatrix} \quad (4.31)$$

Inserting trial function in equation (4.20) to stress displacement relation defined in equation (3.13) gives,

$$\boldsymbol{\sigma}_{[3\times 1]} = \mathbf{D}\,\mathbf{B}\,\mathbf{u} \quad (4.32)$$

where $\mathbf{B}$ is named as strain matrix and defined as

$$\mathbf{B} = \mathbf{L}\boldsymbol{\Phi} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} & 0 & \frac{\partial \phi_2}{\partial x} & 0 & \cdots & \frac{\partial \phi_n}{\partial x} & 0 \\ 0 & \frac{\partial \phi_1}{\partial y} & 0 & \frac{\partial \phi_2}{\partial y} & \cdots & 0 & \frac{\partial \phi_n}{\partial y} \\ \frac{\partial \phi_1}{\partial y} & \frac{\partial \phi_1}{\partial x} & \frac{\partial \phi_2}{\partial y} & \frac{\partial \phi_2}{\partial x} & \cdots & \frac{\partial \phi_n}{\partial y} & \frac{\partial \phi_n}{\partial x} \end{bmatrix} \quad (4.33)$$

$\mathbf{S}$ is the degree of freedom selector for essential boundary condition and

$$\mathbf{S} = \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \quad (4.34)$$

where

$$s_i = \begin{cases} 1, & \text{if } u_i \text{ is prescribed on} \Gamma_q \\ 0, & \text{else} \end{cases} \tag{4.35}$$

Substitute equations (4.32),(4.20) into (4.28) gives two discrete equations for the I$^{\text{th}}$ field node as

$$\int_{\Omega_I} \mathbf{V}_I \mathbf{D}\,\mathbf{B}\,\mathbf{u}\,d\Omega - \int_{\Gamma_{Ii}} \mathbf{W}_I \mathbf{n}\mathbf{D}\,\mathbf{B}\,\mathbf{u}\,d\Gamma - \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{n}\mathbf{S}\mathbf{D}\,\mathbf{B}\,\mathbf{u}\,d\Gamma + \alpha \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{S}\mathbf{\Phi}\mathbf{u}\,d\Gamma = + \int_{\Omega_I} \mathbf{W}_I \mathbf{b}\,d\Omega + \int_{\Gamma_{It}} \mathbf{W}_I \bar{\mathbf{t}}\,d\Gamma + \alpha \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{S}\bar{\mathbf{u}}\,d\Gamma \tag{4.36}$$

Equation (4.36) is rewritten as

$$\mathbf{K}_{I[2\times 2n]}\mathbf{u}_{[2n\times 1]} = \mathbf{f}_{I[2\times 1]} \tag{4.37}$$

where $\mathbf{K}_I$ is matrix called nodal stiffness matrix associated with I$^{\text{th}}$ field node and defined as follows:

$$\mathbf{K}_I = \int_{\Omega_I} \mathbf{V}_I \mathbf{D}\,\mathbf{B}\,d\Omega - \int_{\Gamma_{Ii}} \mathbf{W}_I \mathbf{n}\mathbf{D}\,\mathbf{B}\,d\Gamma - \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{n}\mathbf{S}\mathbf{D}\,\mathbf{B}\,d\Gamma + \alpha \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{S}\mathbf{\Phi}\,d\Gamma \tag{4.38}$$

and $\mathbf{F}_I$ is vector called force vector associated with I$^{\text{th}}$ field node and defined as follows:

$$\mathbf{F}_I = \int_{\Omega_I} \mathbf{W}_I \mathbf{b}\,d\Omega + \int_{\Gamma_{It}} \mathbf{W}_I \bar{\mathbf{t}}\,d\Gamma + \alpha \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{S}\bar{\mathbf{u}}\,d\Gamma \tag{4.39}$$

Note that equations (4.36), (4.38), and (4.39) are discrete equations for one node (I$^{\text{th}}$) and its local domain. For two dimensional problem each node has 2 equation. For all N field node, these two equations are assembled to obtain global system of equation.

System of linear algebraic equations based on global numbering is

$$\underbrace{\begin{bmatrix} K_{(1)(1)} & K_{(1)(2)} & \cdots & K_{(1)(2N-1)} & K_{(1)(2N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{(2I-1)(1)} & K_{(2I-1)(2)} & \cdots & K_{(2I-1)(2N-1)} & K_{(2I-1)(2N)} \\ K_{(2I)(1)} & K_{(2I)(2)} & \cdots & K_{(2I)(2N-1)} & K_{(2I)(2N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{(2N)(1)} & K_{(2N)(2)} & \cdots & K_{(2N)(2N-1)} & K_{(2N)(2N)} \end{bmatrix}}_{\mathbf{K}_{2N\times 2N}} \underbrace{\begin{Bmatrix} u_1 \\ v_1 \\ \vdots \\ u_I \\ v_I \\ \vdots \\ u_N \\ v_N \end{Bmatrix}}_{\mathbf{U}_{2N\times 1}} = \underbrace{\begin{Bmatrix} f_{(1)x} \\ f_{(1)y} \\ \vdots \\ f_{(I)x} \\ f_{(I)y} \\ \vdots \\ f_{(N)x} \\ f_{(N)y} \end{Bmatrix}}_{\mathbf{F}_{2N\times 1}} \tag{4.40}$$

If one satisfies equilibrium equation and boundary conditions for all sub domains whose union covers the global domain then global solution is obtained. On the other hand, Atluri [4] states that even sub domains do not cover the global domain, MLPG gives sufficiently good results.

4.3.1 Numerical Integration

For numerical integration gauss quadrature schema is used. Gauss quadrature implemented form of stiffness matrix is

$$\begin{aligned} \mathbf{K}_I = &+ \sum_{k=1}^{g} w_k \, \mathbf{V}_I(\mathbf{x}_{Gk}) \mathbf{D} \, \mathbf{B}(\mathbf{x}_{Gk}) \, \mathbf{J}_{\Omega_I} \\ &+ \sum_{k=1}^{gi} w_k \, \mathbf{W}_I(\mathbf{x}_{Gk}) \mathbf{n} \mathbf{D} \, \mathbf{B}(\mathbf{x}_{Gk}) \, \mathbf{J}_{\Gamma_{Ii}} \\ &+ \sum_{k=1}^{gu} w_k \, \mathbf{W}_I(\mathbf{x}_{Gk}) \mathbf{n} \mathbf{D} \, \mathbf{B}(\mathbf{x}_{Gk}) \, \mathbf{J}_{\Gamma_{Iu}} \\ &+ \alpha \int_{\Gamma_{Iu}} \mathbf{W}_I \mathbf{\Phi} \, d\Gamma \end{aligned} \tag{4.41}$$

Gauss quadrature implemented form of force vector is

$$\begin{aligned}\mathbf{F}_I = &+ \sum_{k=1}^{g} w_k \, \mathbf{W}_I(\mathbf{x}_{Gk}) \, \mathbf{b} \, \mathbf{J}_{\Omega_I} \\ &+ \sum_{k=1}^{gt} w_k \, \mathbf{W}_I(\mathbf{x}_{Gk}) \, \bar{\mathbf{t}} \, \mathbf{J}_{\Gamma_{It}} \\ &+ \alpha \int_{\Gamma_{qu}} \mathbf{W}_I \bar{\mathbf{u}} \, d\Gamma\end{aligned} \tag{4.42}$$

where g, gi, gu, gt, is number of gauss points.

4.3.2 Weight Function

Cubic spline function is used as weighting function in local symmetric weak form (Eq. (4.23)). The cubic spline weight function is

$$W_i(\mathbf{r}_i) = \begin{cases} 2/3 - 4r_i^2 + 4r_i^3, & r_i \leq 0.5 \\ 4/3 - r_i + 4r_i^2 - 4/3r_i^3, & 0.5 < r_i \leq 1 \\ 0, & r_i > 1 \end{cases} \tag{4.43}$$

where r_i is unit distance between gauss point and node I.
For rectangular integration domain in 2 dimension, r_x and r_y are as follows.

$$r_x = \frac{|x - x_i|}{a_x} \tag{4.44a}$$

$$r_y = \frac{|y - y_i|}{a_y} \tag{4.44b}$$

and a_x, a_y are integration domain size.

Weight function in 2 dimension for rectangular integration domain is as follows

$$W_I = W(r_x) \times W(r_y) \tag{4.45}$$

4.4 Enforcement of Essential Boundary Condition

Instead of penalty method, direct interpolation method is used in order to enforce essential boundary conditions [29]. In MLPG each node have two rows in

the global stiffness matrix and global force vector for two dimensional problems. Then, for a node on essential boundary, its two equation in system of equations are replaced with two essential boundary equation given in (4.46). Consequently, essential boundary condition equations are also satisfied.

Interpolation of essential boundary condition for a node is

$$\mathbf{u}_I^h = \begin{Bmatrix} u_I^h \\ v_I^h \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \cdots & \phi_n & 0 \\ 0 & \phi_1 & 0 & \phi_2 & \cdots & 0 & \phi_n \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{Bmatrix} = \mathbf{\Phi}\ \mathbf{u} = \begin{Bmatrix} \bar{u}_I \\ \bar{v}_I \end{Bmatrix} \tag{4.46}$$

For a field node on essential boundary, essential boundary conditions are imposed in two steps by direct interpolation method. Firstly, two rows of global stiffness matrix, which belong to the field node, are replaced with shape functions matrix in equation (4.46). Secondly, two rows of global force vector, which belong to the field node, are replaced with prescribed displacements in equation (4.46)

After replacement stiffness matrix, $\mathbf{K}$, is

$$\mathbf{K} = \begin{bmatrix} K_{(1)(1)} & K_{(1)(2)} & \cdots & K_{(1)(2N-1)} & K_{(1)(2N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi_1 & 0 & \cdots & \phi_n & 0 \\ 0 & \phi_1 & \cdots & 0 & \phi_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{(2N)(1)} & K_{(2N)(2)} & \cdots & K_{(2N)(2N-1)} & K_{(2N)(2N)} \end{bmatrix} \tag{4.47}$$

After replacement force vector, $\mathbf{F}$, is

$$\mathbf{F} = \begin{Bmatrix} f_{(1)x} \\ f_{(1)y} \\ \vdots \\ \bar{u}_I \\ \bar{v}_I \\ \vdots \\ f_{(N)x} \\ f_{(N)y} \end{Bmatrix} \tag{4.48}$$

4.5 Displacements

After solving linear systems of equations, $\mathbf{u}$ vector is obtained. However due to lack of Kronecker delta property, these displacement values is fictitious and cannot be used directly. Following equation is used in order to find displacements of point at $\mathbf{x}$.

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \cdots & \phi_n & 0 \\ 0 & \phi_1 & 0 & \phi_2 & \cdots & 0 & \phi_n \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{Bmatrix} \tag{4.49}$$

where $\phi_i(\mathbf{x})$ is shape function corresponding to the i^{th} node in support domain of point at $\mathbf{x}$ and defined in equation (4.19).

4.6 Stresses

After solving linear systems of equations $\mathbf{u}$ vector is obtained. In order to find stress of a position, one should construct MLS shape functions for that point. Then, following equation is used to get stress of a point

$$\boldsymbol{\sigma} = \mathbf{D}\,\mathbf{B}\,\mathbf{u} \tag{4.50}$$

where $\mathbf{D}$ is defined in (3.10) and $\mathbf{B}$ is defined in (4.33).

4.7 Error Estimation

In addition to direct comparison of stresses and displacements with exact solution, following norms and errors are also considered.

Displacement L^2 norm and relative error are

$$\|\mathbf{u}\| = \left(\int_{\Omega} \mathbf{u}^T\,\mathbf{u}\,d\Omega \right)^{\frac{1}{2}} \tag{4.51a}$$

$$r_u = \frac{\|\mathbf{u}^{num} - \mathbf{u}^{exact}\|}{\|\mathbf{u}^{exact}\|} \tag{4.51b}$$

Energy norm and relative error are

$$\|\mathbf{E}\| = \left(\frac{1}{2} \int_{\Omega} \varepsilon^T\,\mathbf{D}\,\varepsilon\,d\Omega \right)^{\frac{1}{2}} \tag{4.52a}$$

$$r_E = \frac{\|\mathbf{E}^{num} - \mathbf{E}^{exact}\|}{\|\mathbf{E}^{exact}\|} \tag{4.52b}$$

CHAPTER 5

NUMERICAL EXAMPLES

5.1 Cantilever Beam

A Cantilever beam subjected to a parabolic traction at the free end is considered as seen in figure 5.1. Beam has unit thickness and parabolic transverse loading, given in equation (5.1), is applied at free end.

Properties of problem are tabulated in table 5.1 and problem is solved for plane stress case. MLPG solution is compared with analytical solution given in §5.1.1.

Traction of right edge given as follows

$$t(y) = \frac{P}{2I}\left[\frac{D^2}{4} - y^2\right] \tag{5.1}$$

where I is moment of inertia and defined as follows:

$$I = \frac{D^3}{12} \tag{5.2}$$

5.1.1 Exact Solution

Exact solution of problem is given by Timoshenko and Goodier [43]

Table 5.1: Properties of Cantilever Beam Problem

Parameters	Values
Height, D	12 mm
Length, L	48 mm
Thickness, t	1 mm
Load, P	-100 N
Modulus of Elasticity, E	70000 MPa
Poisson's Ratio, ν	0.33

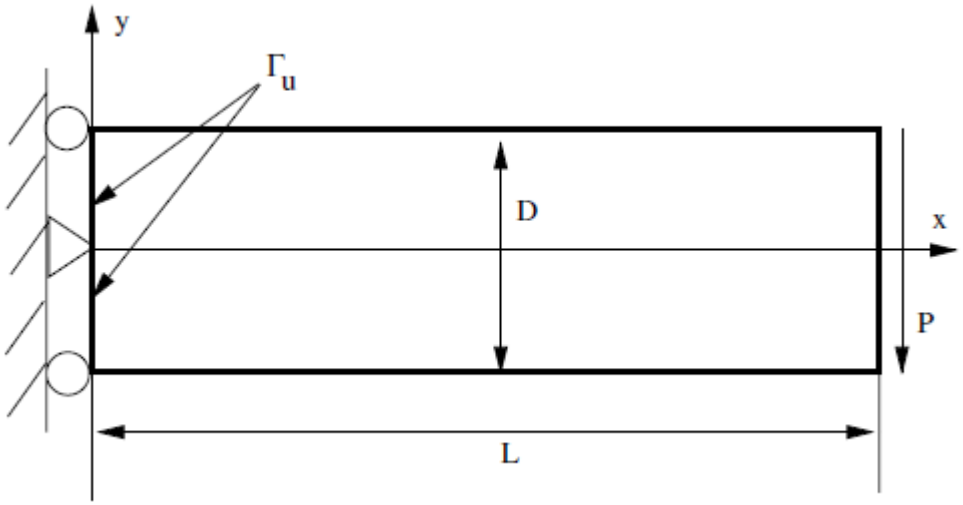

Figure 5.1: Cantilever beam

Displacement in x-direction is

$$u(x,y) = -\frac{Py}{6EI}\left[(6L-3x)x + (2+v)\left(y^2 - \frac{D^2}{4}\right)\right] \tag{5.3}$$

Displacement in y-direction is

$$v(x,y) = \frac{Py}{6EI}\left[3vy^2(L-x) + (4+5v)\frac{D^2x}{4} + (3L-x)x^2\right] \tag{5.4}$$

Stress in x-direction is

$$\sigma_{xx}(x,y) = -\frac{P(L-x)y}{I} \tag{5.5}$$

Stress in y-direction is

$$\sigma_{yy}(x,y) = 0 \tag{5.6}$$

Shear stress is

$$\tau_{xy}(x,y) = \frac{P}{2I}\left[\frac{D^2}{4} - y^2\right] \tag{5.7}$$

5.1.2 MLPG Solution

Node distribution and gauss integration domain for an internal node is shown in figure 5.2. In addition, in figure 5.3, integration domain for an boundary node is illustrated.

For trial function construction 6 monomials are used. Influence domain size is 3.5 times nodal spacing.

Integration domain size is 2.5 times nodal spacing. For integration, integration domain is divided into 4 sub-domains. 4 gauss points are used for each sub-domain.

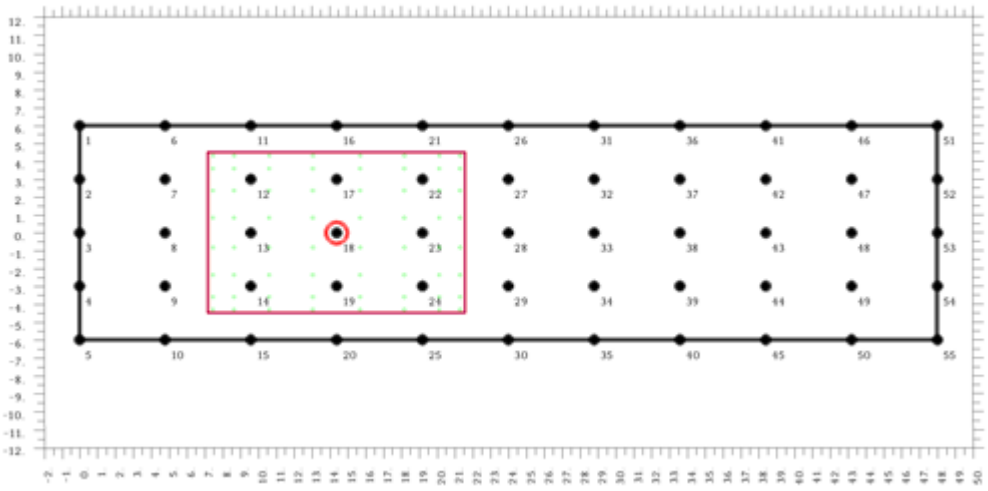

Figure 5.2: Cantilever beam node distribution and internal integration domain

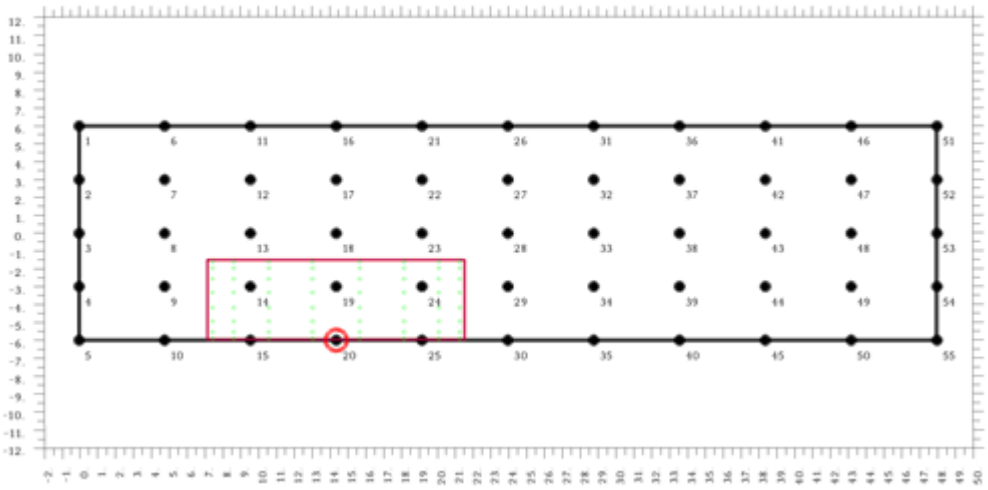

Figure 5.3: Cantilever beam node distribution and edge integration domain

5.1.3 Results

Results of MLPG solution is directly compared with exact solution. In addition, energy norm error is also calculated and given in table 5.2

Tip displacement of solution and exact errors are tabulated in table 5.3 and 5.4.

Normal and shear stresses at mid-length of beam and associated exact errors are tabulated in table 5.5 and 5.6.

Color plots of normal and shear stress of MLPG and exact solution for whole domain are given in figure 5.4 and 5.5.

Color plots of displacements of MLPG and exact solution for whole domain are given in figure 5.6 and 5.7.

Table 5.2: Global energy error of cantilever beam problem

Energy Error
0.05

Table 5.3: Tip displacement u_x of cantilever beam ($x = 48\ mm$)

y [mm]	u_x exact [mm]	u_x MLPG [mm]	% Error
6	0.068571	0.067895	0.99
3	0.033974	0.033649	0.96
0	0.000000	0.000000	N/A
-3	-0.033974	-0.033649	0.96
-6	-0.068571	-0.067895	0.99

Table 5.4: Tip displacement u_y of cantilever beam ($x = 48\ mm$)

y [mm]	u_y exact [mm]	u_y MLPG [mm]	% Error
6	-0.381857	-0.379470	0.63
3	-0.381857	-0.379520	0.61
0	-0.381857	-0.379510	0.61
-3	-0.381857	-0.379520	0.61
-6	-0.381857	-0.379470	0.63

Table 5.5: Mid-length stress σ_x of cantilever beam ($x = 24\ mm$)

y [mm]	σ_x exact [MPa]	σ_x MLPG [MPa]	% Error
6	100.0	97.8	2.17
3	50.0	48.9	2.16
0	0.0	0.0	N/A
-3	-50.0	-48.9	2.16
-6	-100.0	-97.8	2.17

Table 5.6: Mid-length stress σ_{xy} of cantilever beam ($x = 24\ mm$)

y [mm]	$\sigma_x y$ exact [MPa]	$\sigma_x y$ MLPG [MPa]	% Error
6	0.0	-0.4	N/A
3	-9.4	-9.4	0.43
0	-12.5	-12.1	2.98
-3	-9.4	-9.4	0.43
-6	0.0	-0.4	N/A

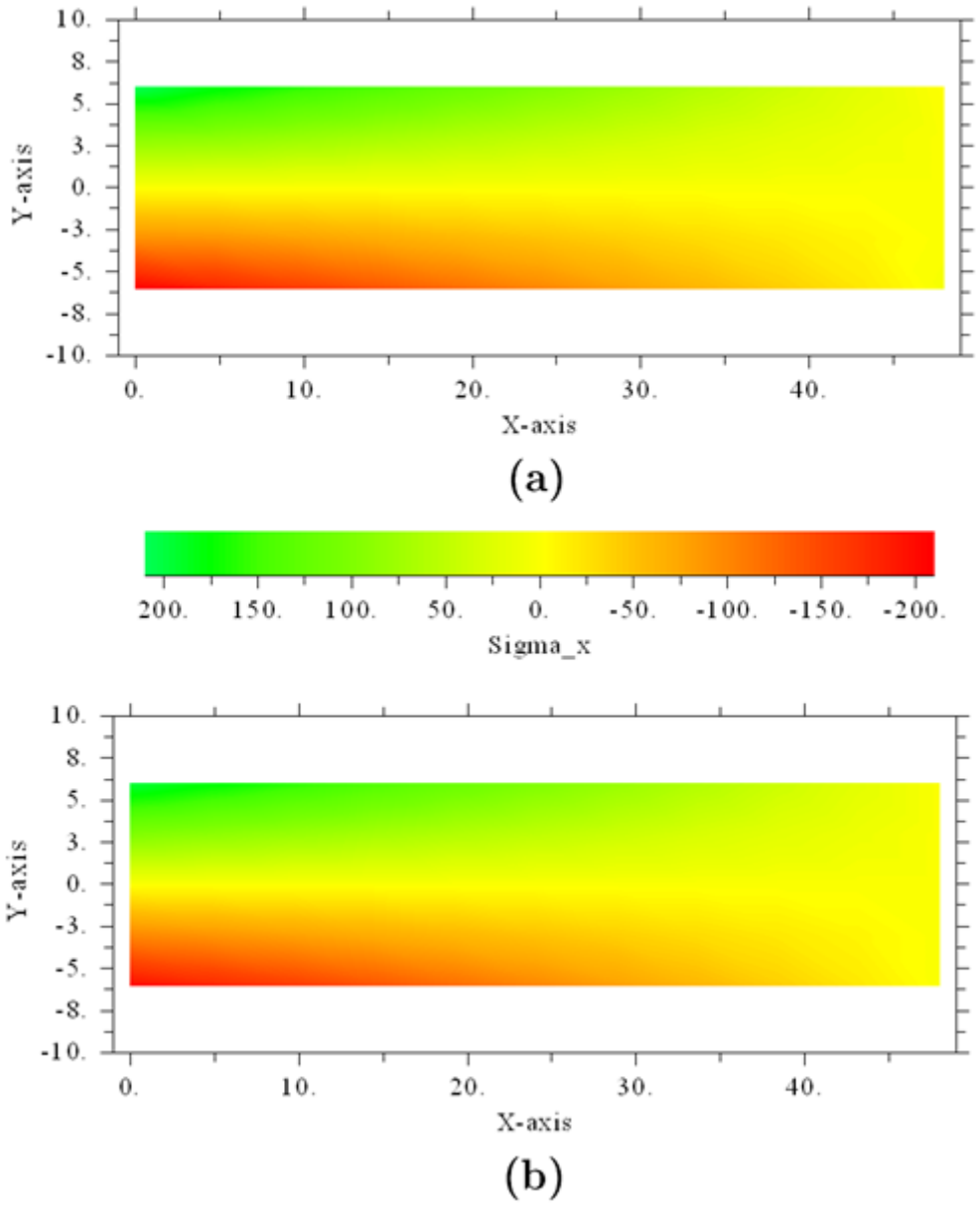

Figure 5.4: σ_x stress of cantilever beam (a) MLPG (b) Exact

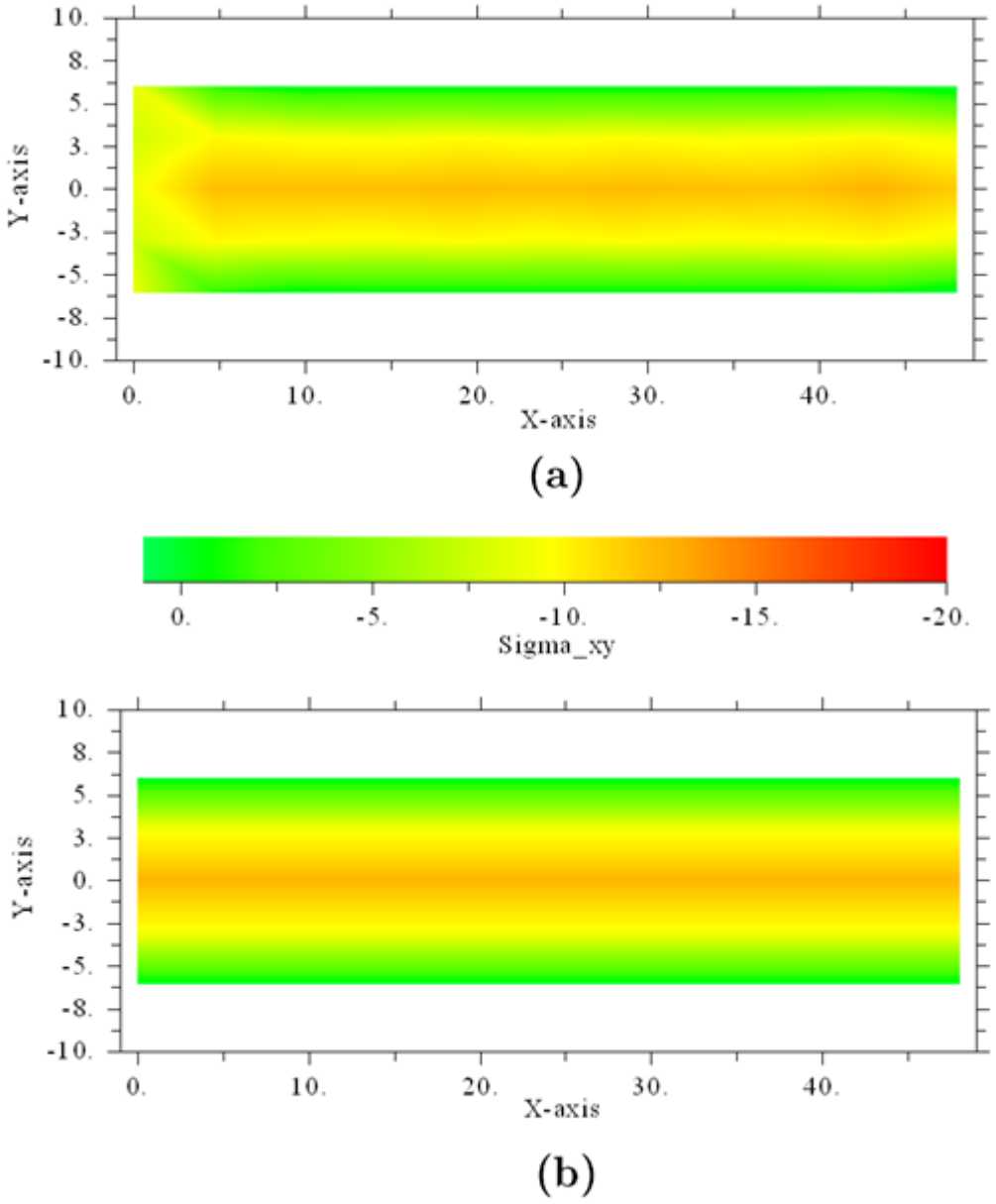

Figure 5.5: σ_{xy} stress of cantilever beam (a) MLPG (b) Exact

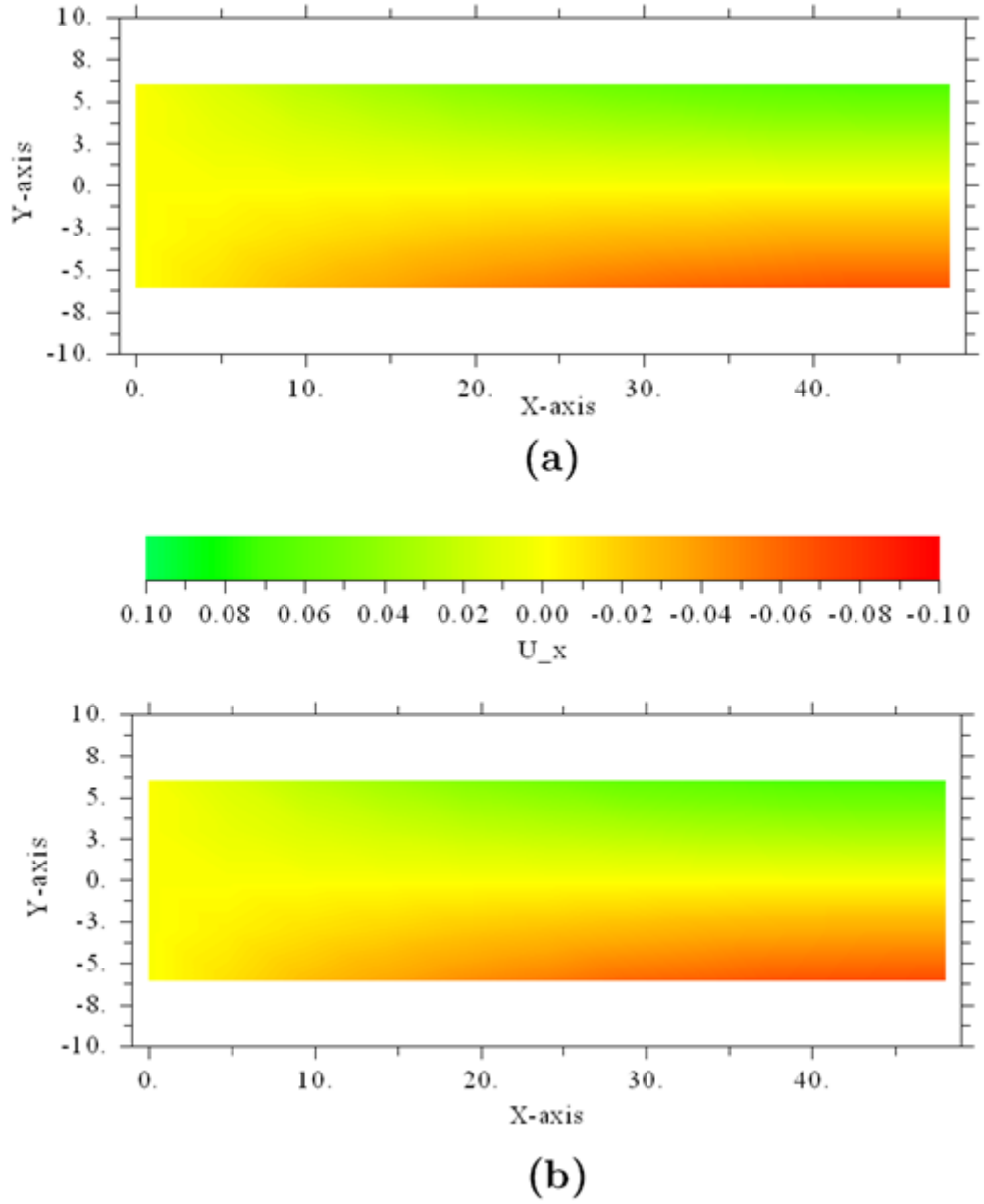

Figure 5.6: u_x displacement of cantilever beam (a) MLPG (b) Exact

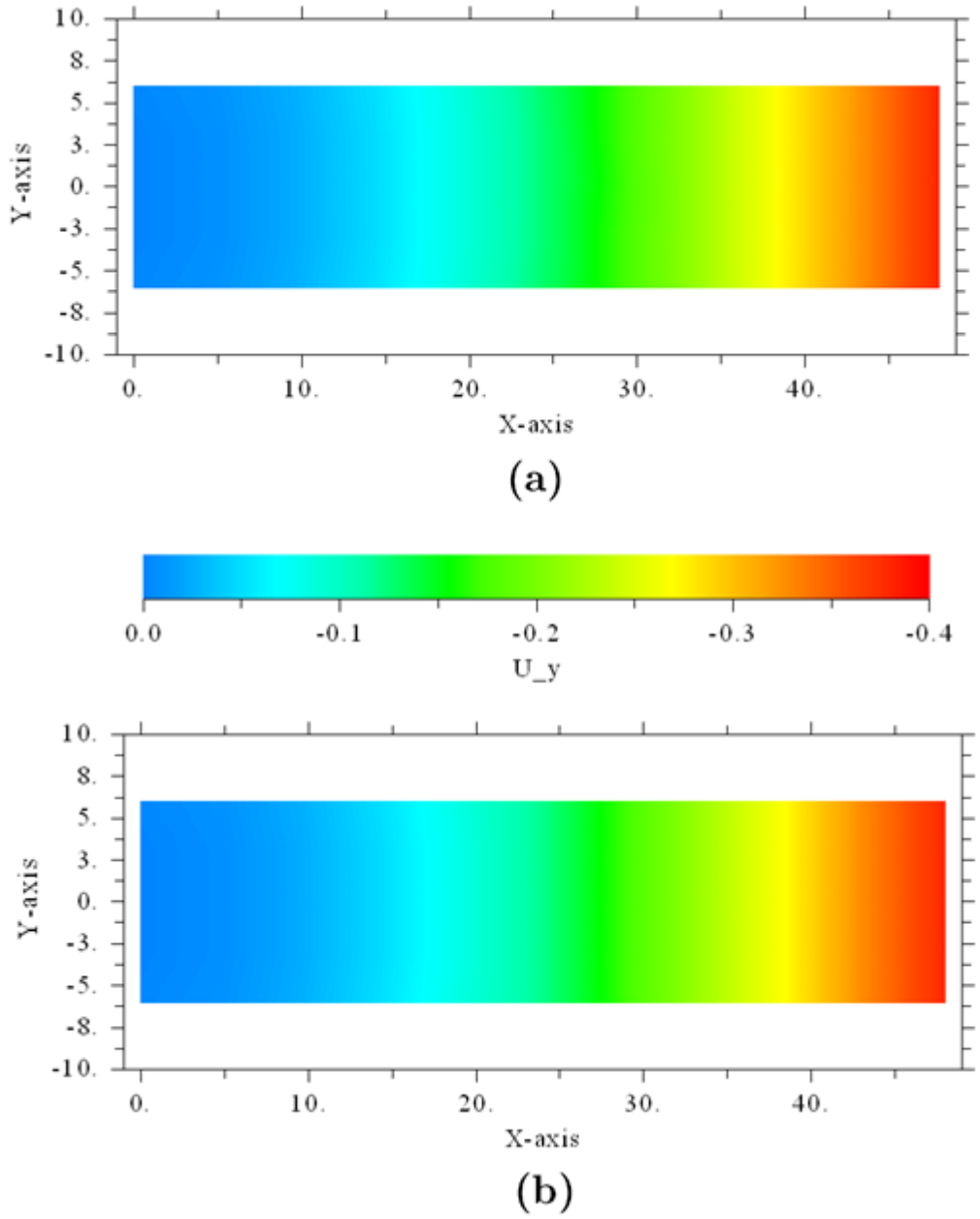

Figure 5.7: u_y displacement of cantilever beam (a) MLPG (b) Exact

5.2 Infinite Plate with Circular Hole

Infinite plate with circular hole under uniaxial tension , shown in figure 5.8, is considered for effects of parameter study. Properties of problem are tabulated in table 5.7 and problem is solved for plane stress case.

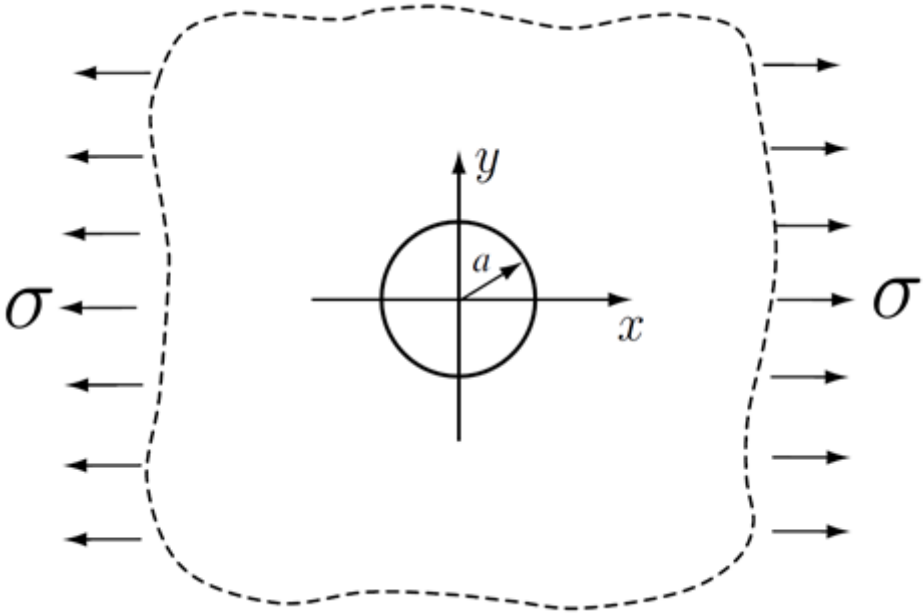

Figure 5.8: Infinite plate with circular hole

Table 5.7: Properties of Plate Problem

Parameters	Values
Hole Radius, a	9 mm
Height, D	36 mm
Length, L	36 mm
Thickness, t	1 mm
Load, σ	100 MPa
Modulus of Elasticity, E	30000 MPa
Poisson's Ratio, ν	0.3

Due to symmetry, upper right quadrant of the plate is modeled as shown in figure 5.9. Symmetric boundary conditions are imposed on the bottom and left edges. Traction stresses of exact solution, given in equation (5.15), (5.16),and (5.17) , are applied to the right and top edge. Circular boundary is traction free.

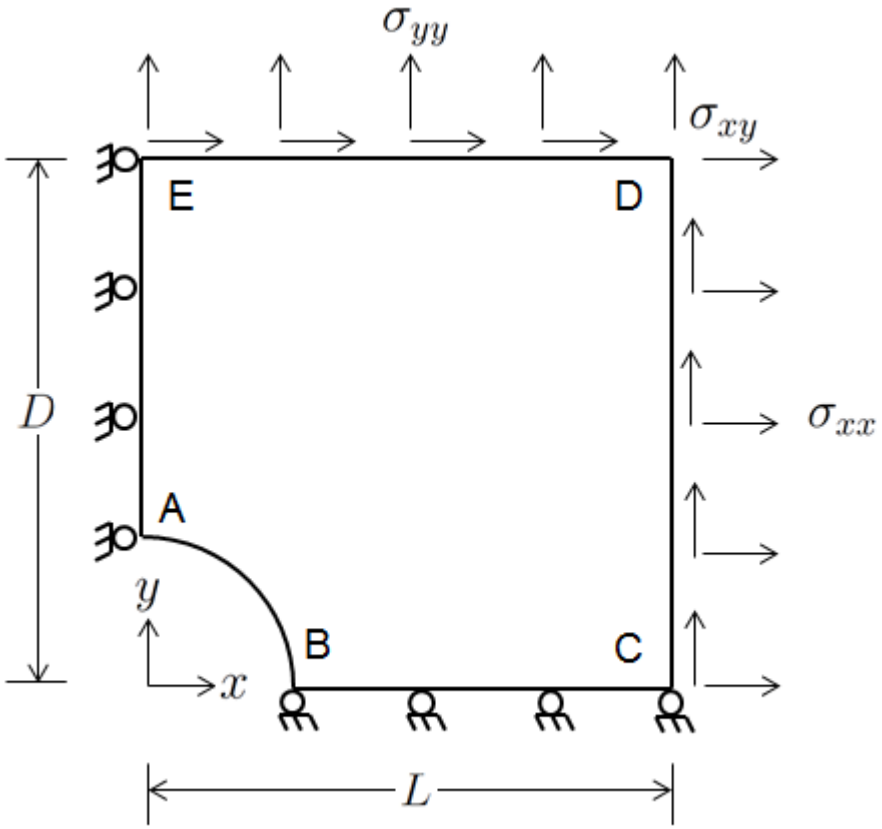

Figure 5.9: One quarter of infinite plate with circular hole

- Segment AB: prescribed natural boundary

$$\bar{t}_1 = 0, \qquad \bar{t}_2 = 0 \tag{5.8}$$

- Segment BC: prescribed essential and natural boundary

$$\bar{u}_2 = 0, \qquad \bar{t}_1 = 0 \tag{5.9}$$

- Segment CD: prescribed natural boundary

$$\bar{t}_1 = \sigma_{xx}, \qquad \bar{t}_2 = \sigma_{xy} \tag{5.10}$$

- Segment DE: prescribed natural boundary

$$\bar{t}_1 = \sigma_{xy}, \qquad \bar{t}_2 = \sigma_{yy} \tag{5.11}$$

- Segment EA: prescribed essential and natural boundary

$$\bar{u}_1 = 0, \qquad \bar{t}_2 = 0 \tag{5.12}$$

5.2.1 Exact Solution

Exact solution of problem is presented in [3].

Displacement in x-direction with respect to radial coordinates is

$$u(r,\theta) = \frac{1+\bar{v}}{\bar{E}}\sigma\left(\frac{1}{1+\bar{v}}r\cos\theta + \frac{2}{1+\bar{v}}\frac{a^2}{r}\cos\theta + \frac{1}{2}\frac{a^2}{r}\cos 3\theta - \frac{1}{2}\frac{a^4}{r^3}\cos 3\theta\right) \tag{5.13}$$

Displacement in y-direction with respect to radial coordinates is

$$u(r,\theta) = \frac{1+\bar{v}}{\bar{E}}\sigma\left(\frac{-\bar{v}}{1+\bar{v}}r\sin\theta - \frac{1-\bar{v}}{1+\bar{v}}\frac{a^2}{r}\sin\theta + \frac{1}{2}\frac{a^2}{r}\sin 3\theta - \frac{1}{2}\frac{a^4}{r^3}\sin 3\theta\right) \tag{5.14}$$

Stress in x-direction with respect to radial coordinates is

$$\sigma_{xx}(r,\theta) = \sigma\left[1 - \frac{a^2}{r^2}\left(\frac{3}{2}\cos 2\theta + \cos 4\theta\right) + \frac{3a^4}{2r^4}\cos 4\theta\right] \tag{5.15}$$

Stress in y-direction with respect to radial coordinates is

$$\sigma_{yy}(r,\theta) = \sigma\left[-\frac{a^2}{r^2}\left(\frac{1}{2}\cos 2\theta - \cos 4\theta\right) - \frac{3a^4}{2r^4}\cos 4\theta\right] \tag{5.16}$$

Shear stress with respect to radial coordinates is

$$\sigma_{xy}(r,\theta) = \sigma\left[-\frac{a^2}{r^2}\left(\frac{1}{2}\sin 2\theta + \sin 4\theta\right) + \frac{3a^4}{2r^4}\sin 4\theta\right] \tag{5.17}$$

5.2.2 MLPG Solution

Node distribution is shown in figure 5.10. There are 8 field nodes distributed uniformly in tangential direction. On the other hand, there are 13 field nodes distributed non-uniformly in radial direction. A total of 104 field nodes is used to represent problem domain. In radial direction, ratio of adjacent spaces between nodes is 1.15.

For trial function construction 6 monomials are used. For integration, integration domain is divided into 4 sub-domains. 4 gauss points are used for each sub-domain.

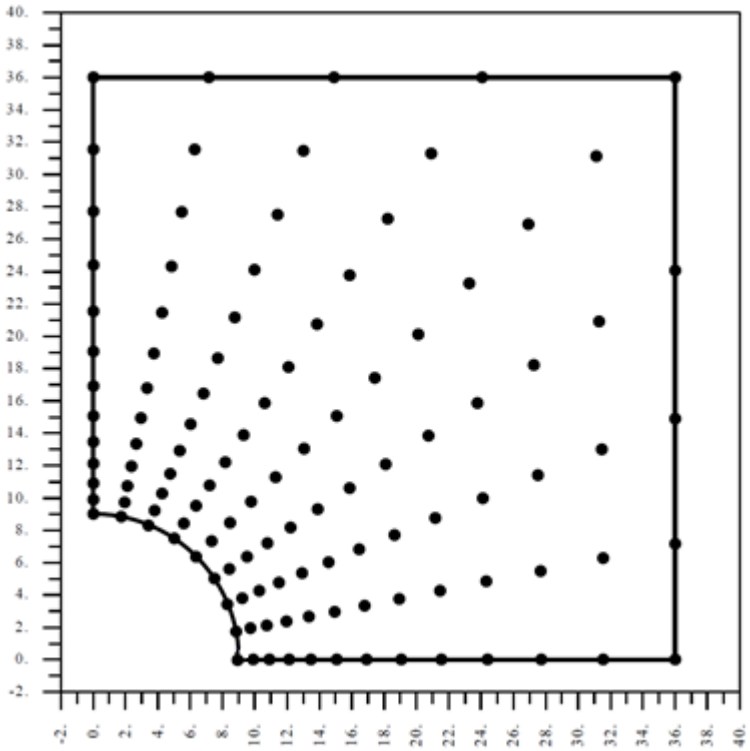

Figure 5.10: Nodal arrangement for infinite plate

5.2.3 Results

4 sets of run are performed in order to investigate effects of integration and influence domain size. Results of exact solution is used to calculate exact error norms.

In first set, influence domain size multiplier is 3.5 and analysis is performed for different values of integration domain size multiplier which vary from 0.5 to 4.0 with 0.1 increments. Results are shown in figure 5.11.

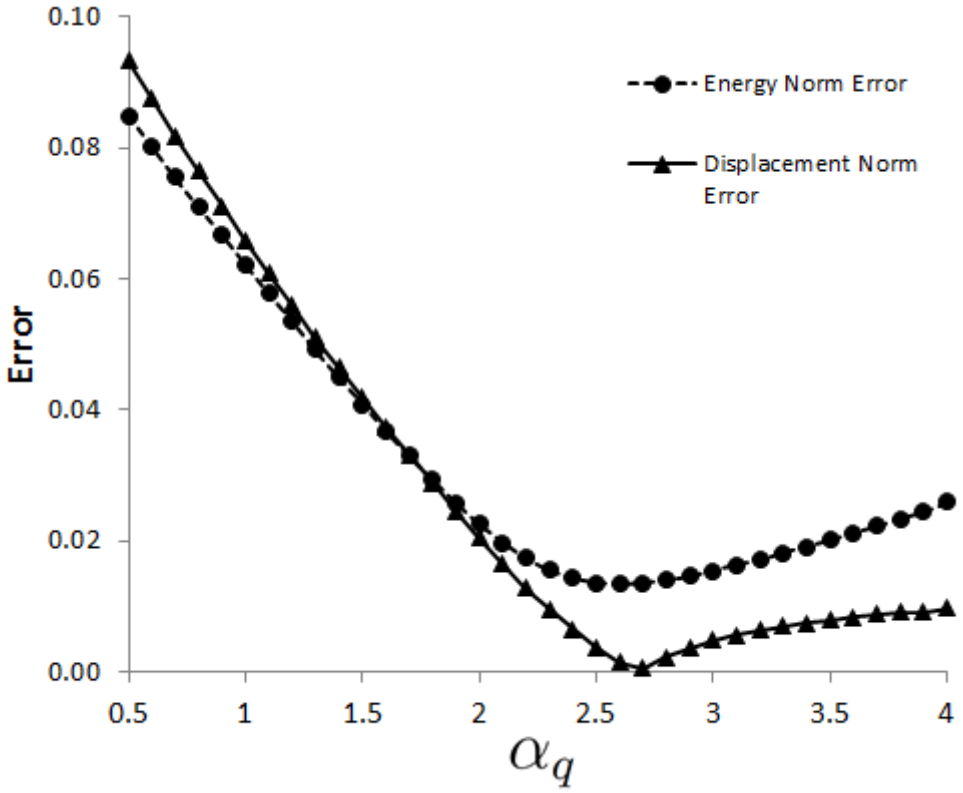

Figure 5.11: Energy and displacement L^2-norm exact error with respect to integration domain size α_q for influence domain size $\alpha_I = 3.5$

In second set, influence domain size multiplier is 4.0 and analysis is performed for different values of integration domain size multiplier which vary from 0.5 to 4.0 with 0.1 increments. Results are shown in figure 5.12.

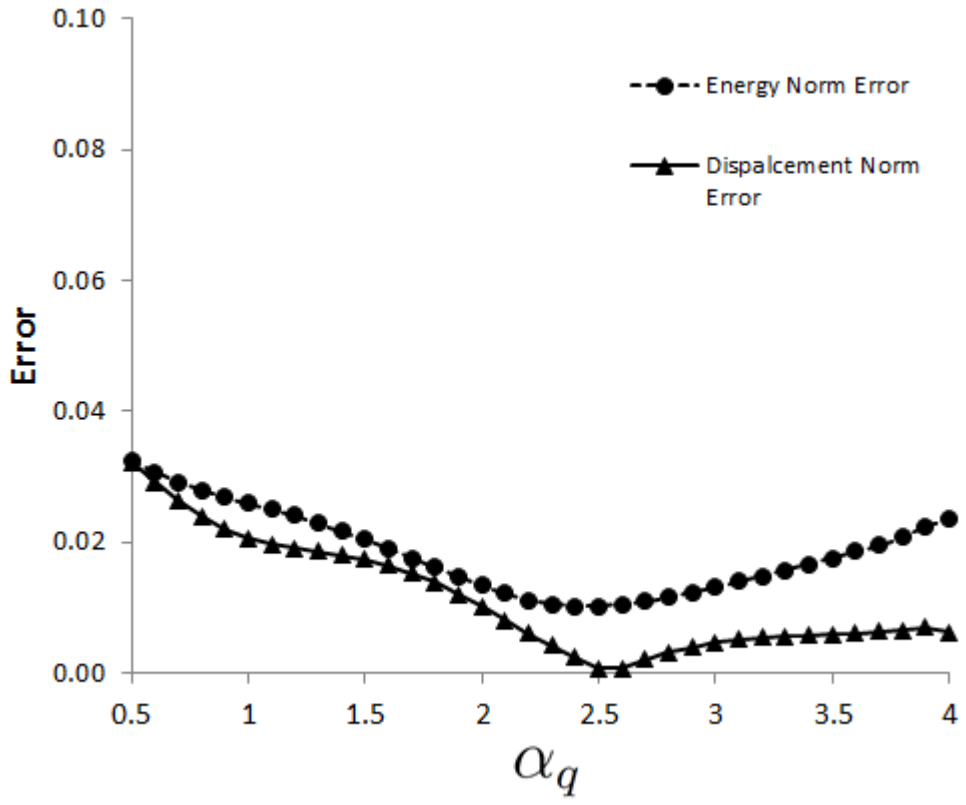

Figure 5.12: Energy and displacement L^2-norm exact error with respect to integration domain size α_q for influence domain size $\alpha_I = 4.0$

In third set, influence domain size multiplier is 4.5 and analysis is performed for different values of integration domain size multiplier which vary from 0.5 to 4.0 with 0.1 increments. Results are shown in figure 5.13.

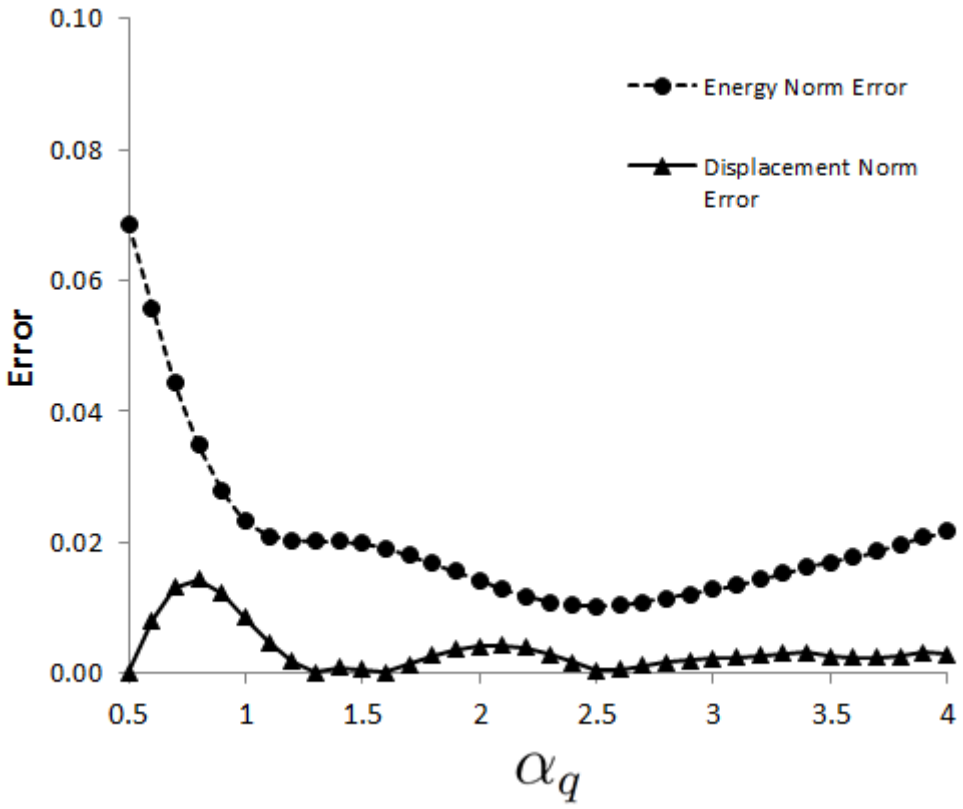

Figure 5.13: Energy and displacement L^2-norm exact error with respect to integration domain size α_q for influence domain size $\alpha_I = 4.5$

In fourth set, integration domain size multiplier is 2.5 and analysis is performed for different values of influence domain size multiplier which vary from 3.5 to 4.5 with 0.1 increments. Results are shown in figure 5.14.

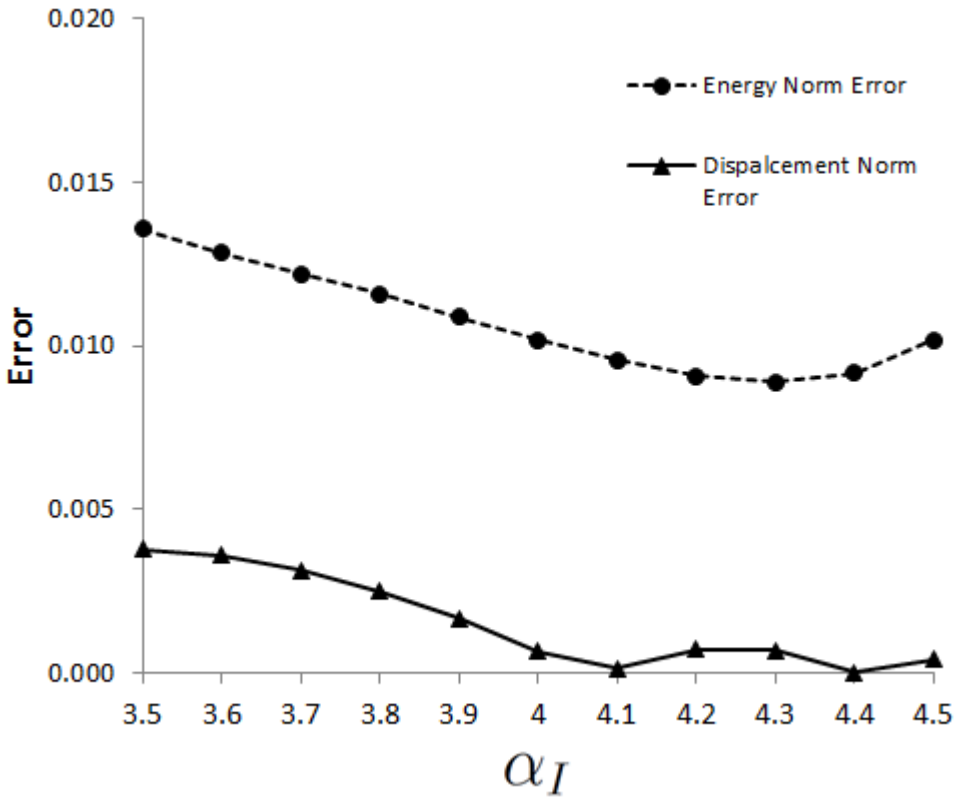

Figure 5.14: Energy and displacement L^2-norm exact error with respect to influence domain size α_I for integration domain size $\alpha_q = 2.5$

Finally, results of plate with circular hole problem is evaluated with selected parameters. These parameters are as follows:

- Integration domain size multiplier is set to 2.5
- Influence domain size is set to 4.0

Following results are based on foregoing parameters.

Tangential stresses in cylindrical coordinates for circular section where $r = 9$ mm are shown in figure 5.15

Normal stresses, and shear stress for bottom edge where $\theta = 0°$ are shown in figure 5.16, 5.17, and 5.18.

Normal stresses, and shear stress for diagonal where $\theta = 45°$ are shown in figure 5.19, 5.20, and 5.21.

Normal stresses, and shear stress for left edge where $\theta = 90°$ are shown in figure 5.22, 5.23, and 5.24.

Color plots of normal stresses and shear stress of MLPG and exact solution for problem domain are given in figure 5.25, 5.26 and 5.5.

Color plots of displacements of MLPG and exact solution for problem domain is given in figure 5.28 and 5.29.

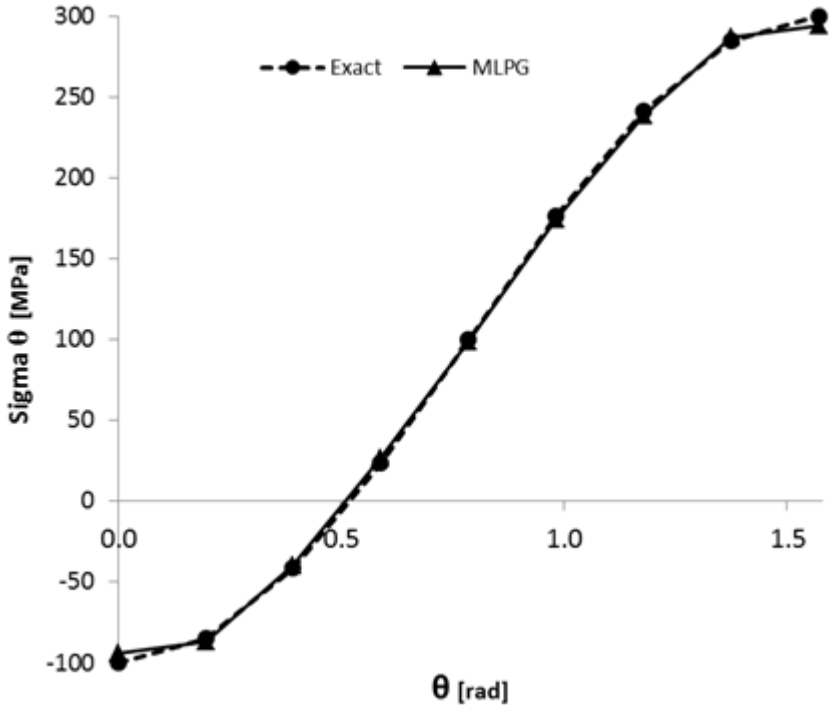

Figure 5.15: σ_θ stress of plate problem at $r = 9$ mm for MLPG and Exact Solution

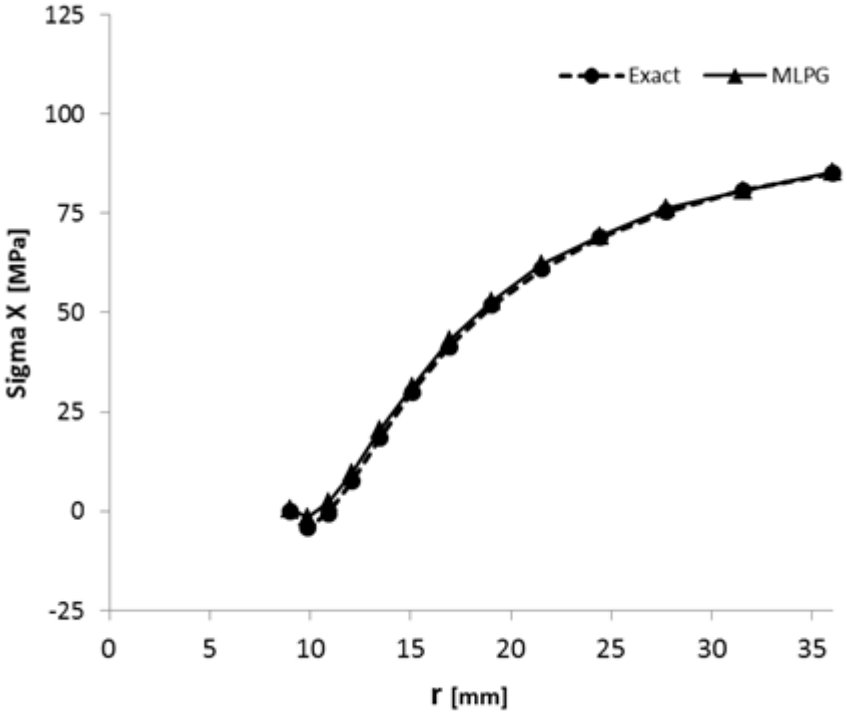

Figure 5.16: σ_x stress of plate problem at $\theta = 0°$ for MLPG and Exact Solution

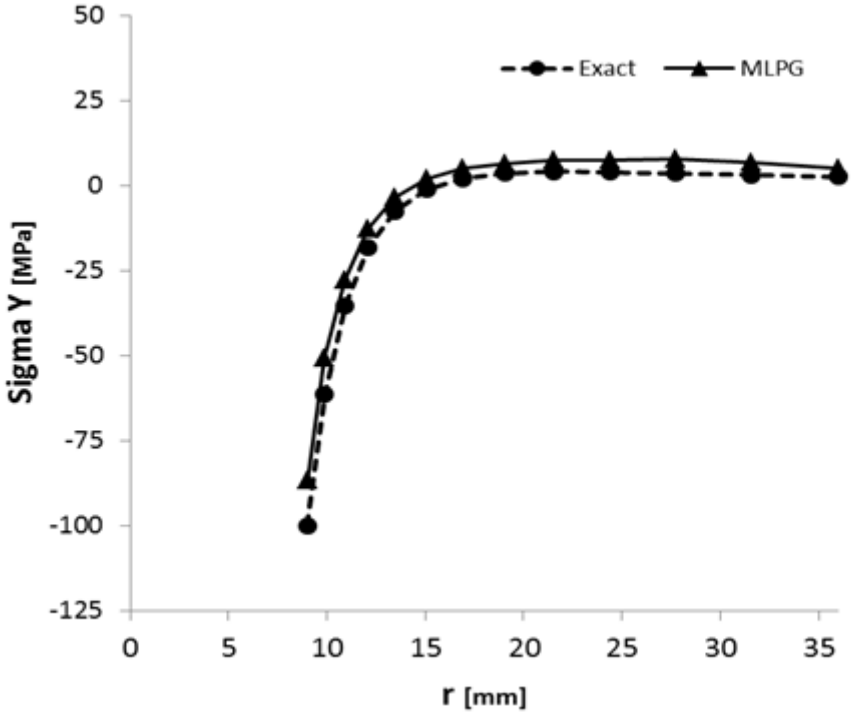

Figure 5.17: σ_y stress of plate problem at $\theta = 0°$ for MLPG and Exact Solution

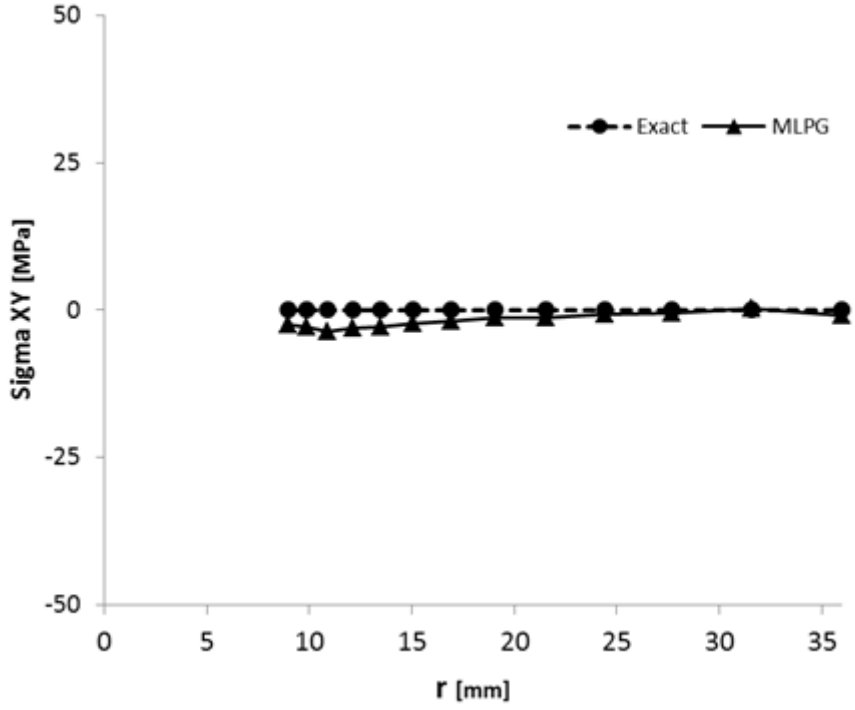

Figure 5.18: σ_{xy} stress of plate problem at $\theta = 0°$ for MLPG and Exact Solution

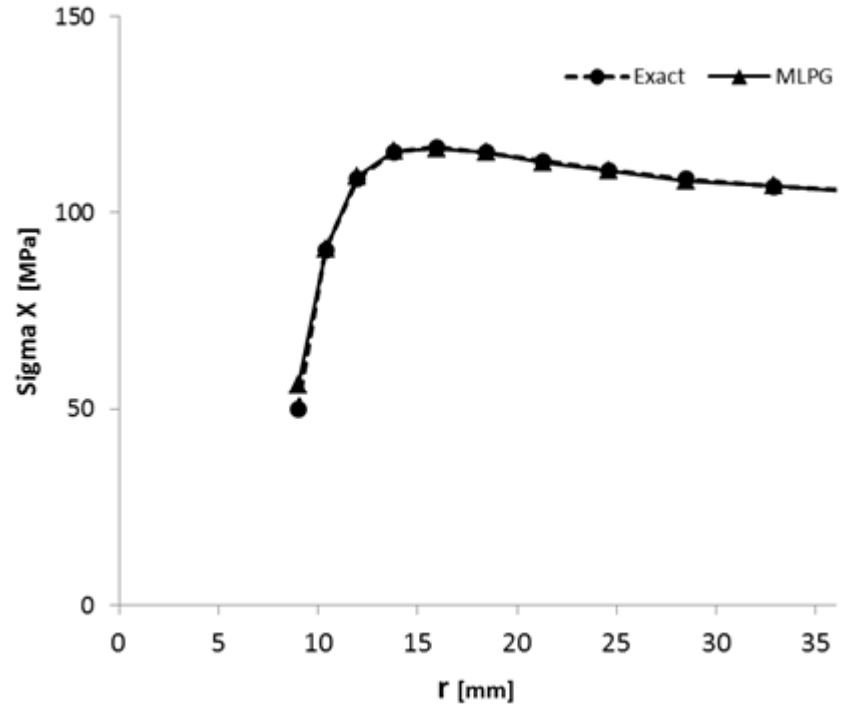

Figure 5.19: σ_x stress of plate problem at $\theta = 45°$ for MLPG and Exact Solution

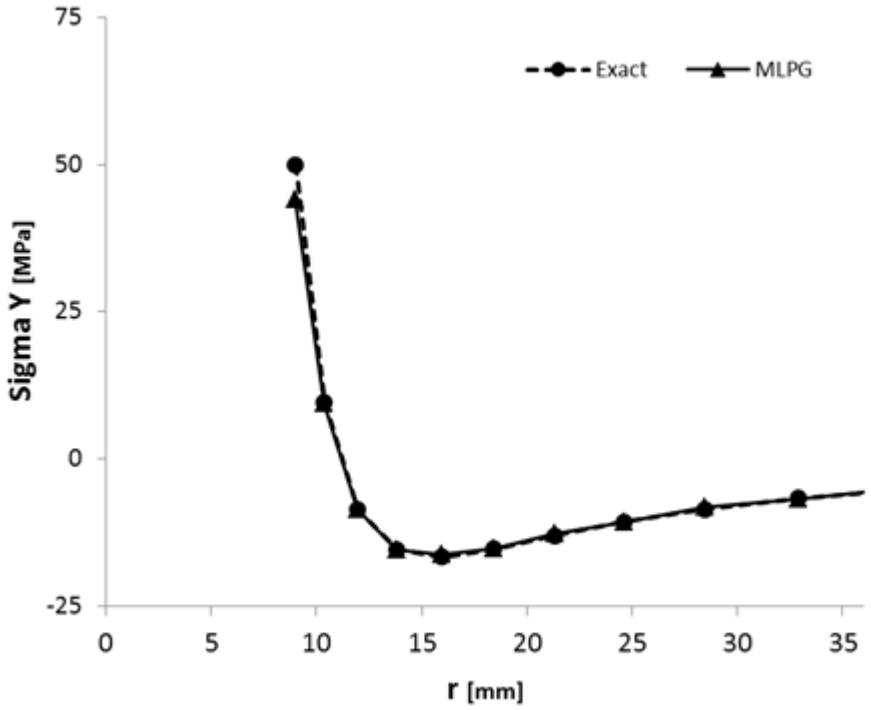

Figure 5.20: σ_y stress of plate problem at $\theta = 45°$ for MLPG and Exact Solution

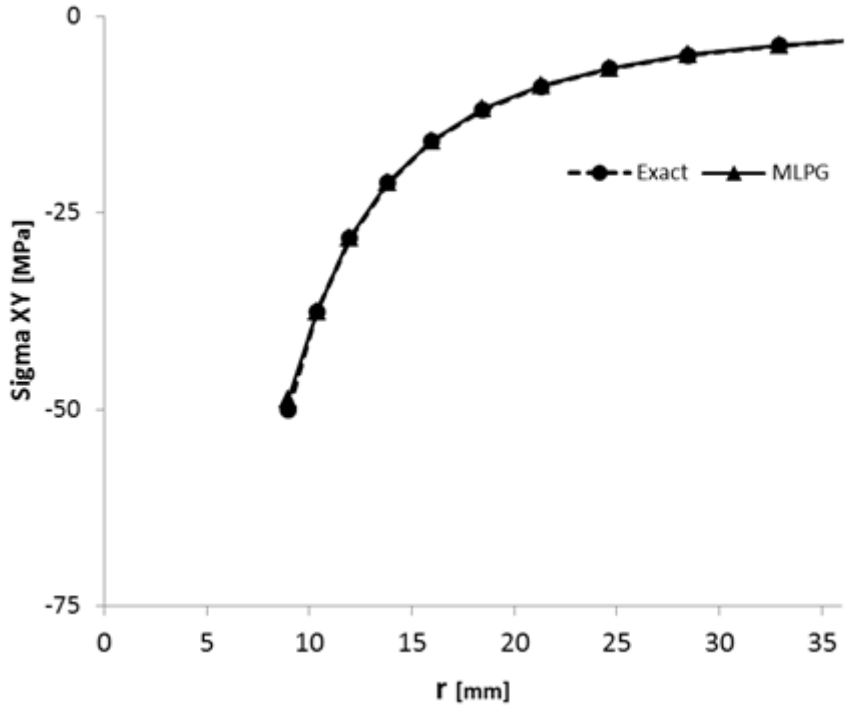

Figure 5.21: σ_{xy} stress of plate problem at $\theta = 45°$ for MLPG and Exact Solution

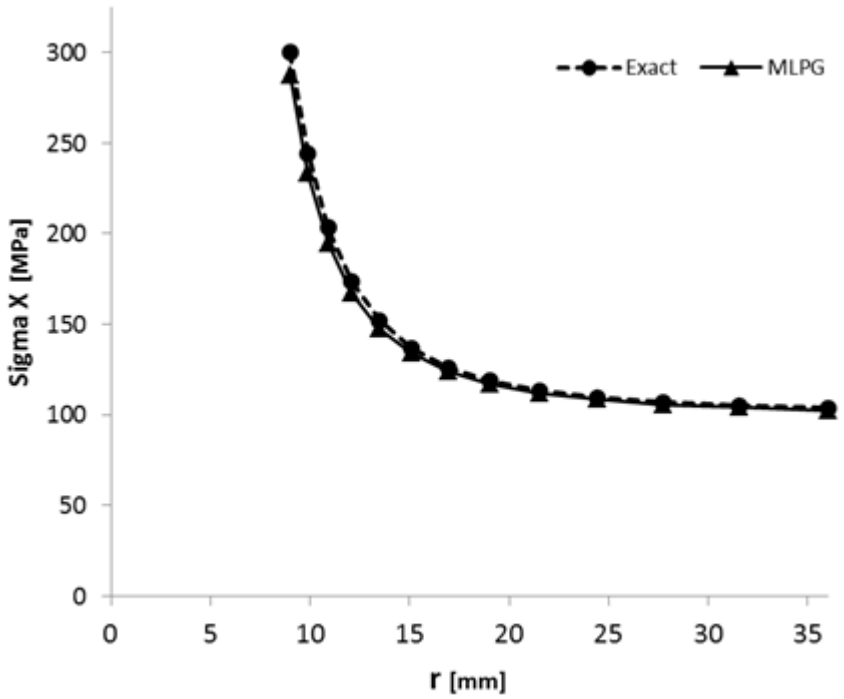

Figure 5.22: σ_x stress of plate problem at $\theta = 90°$ for MLPG and Exact Solution

Energy and displacement errors are presented in table 5.8 for selected parameters.

Table 5.8: Global energy and displacement error of infinite plate with circular hole

Energy Error	Displacement Error
0.01	0.0006

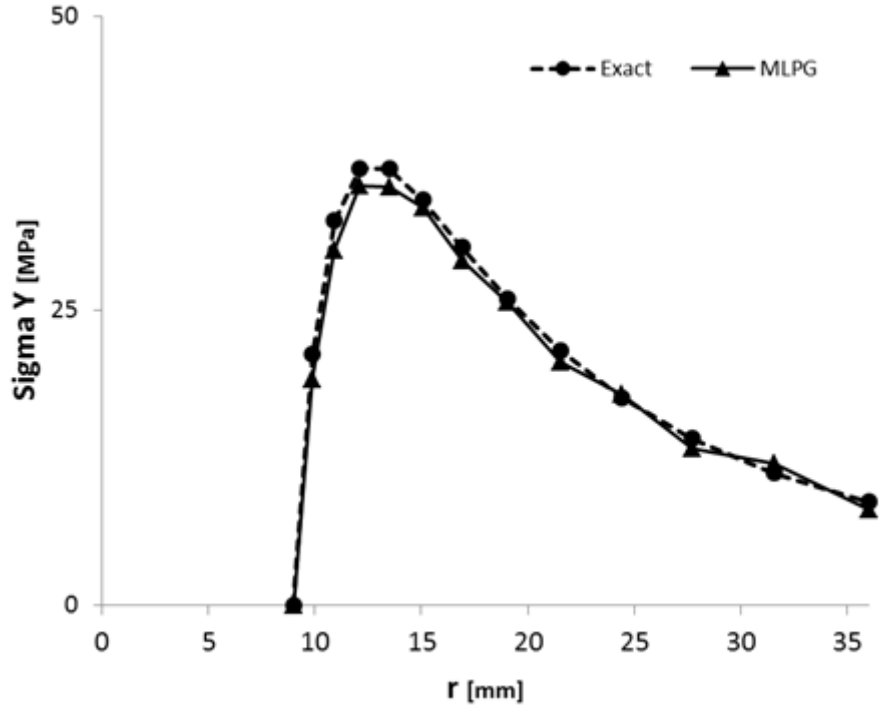

Figure 5.23: σ_y stress of plate problem at $\theta = 90°$ for MLPG and Exact Solution

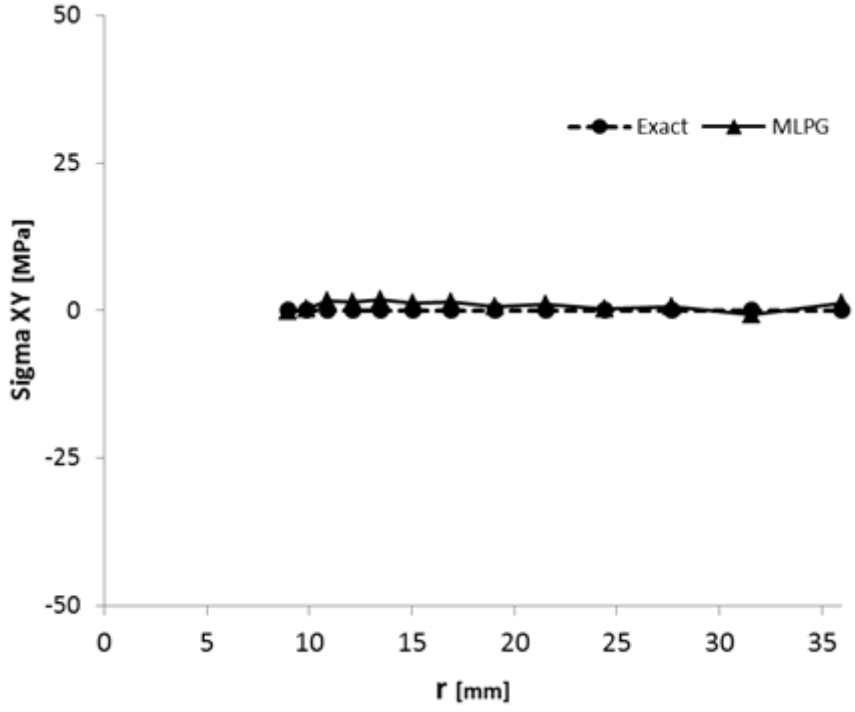

Figure 5.24: σ_{xy} stress of plate problem at $\theta = 90°$ for MLPG and Exact Solution

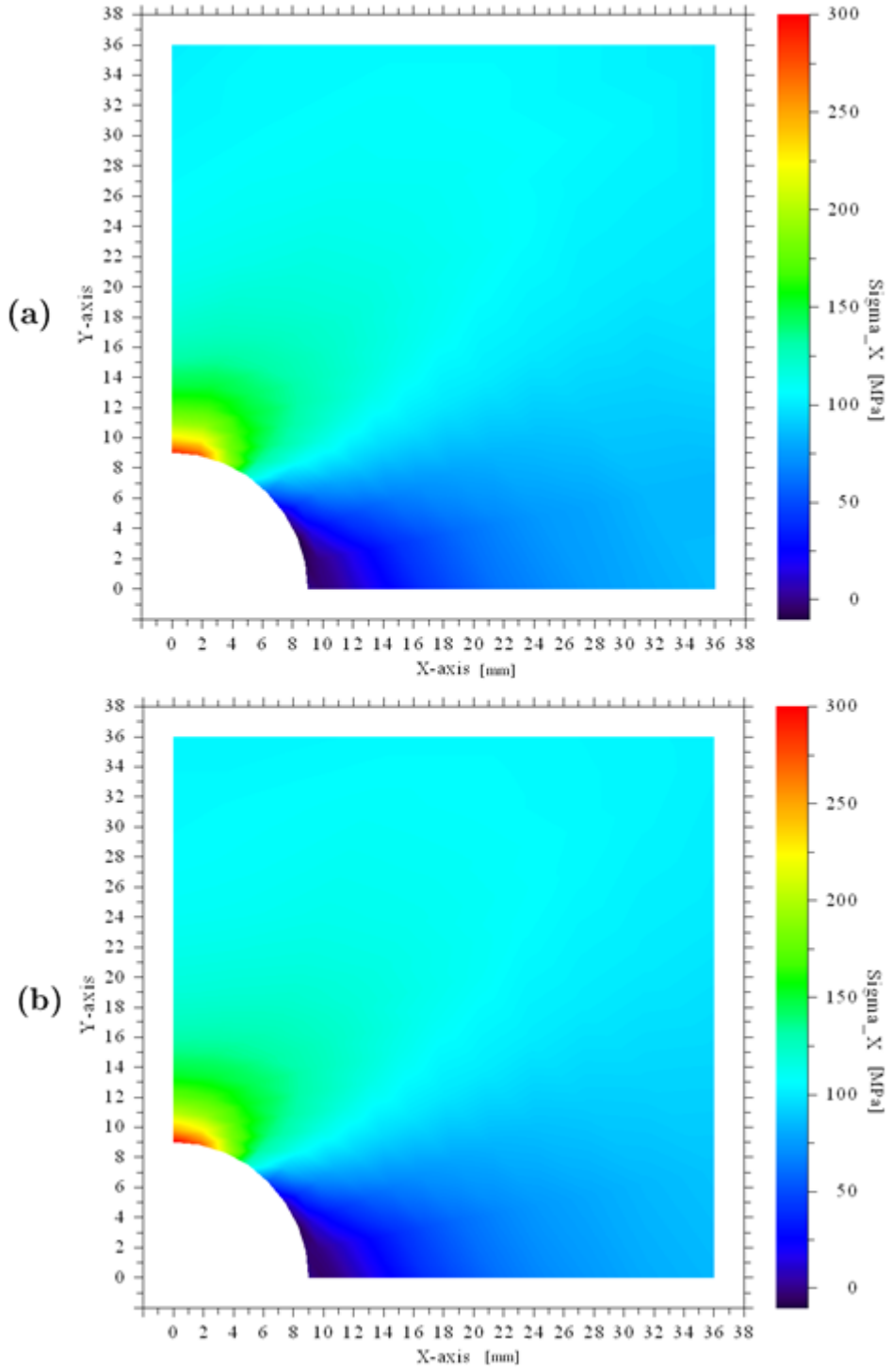

Figure 5.25: σ_x stress of plate problem (a) MLPG (b) Exact

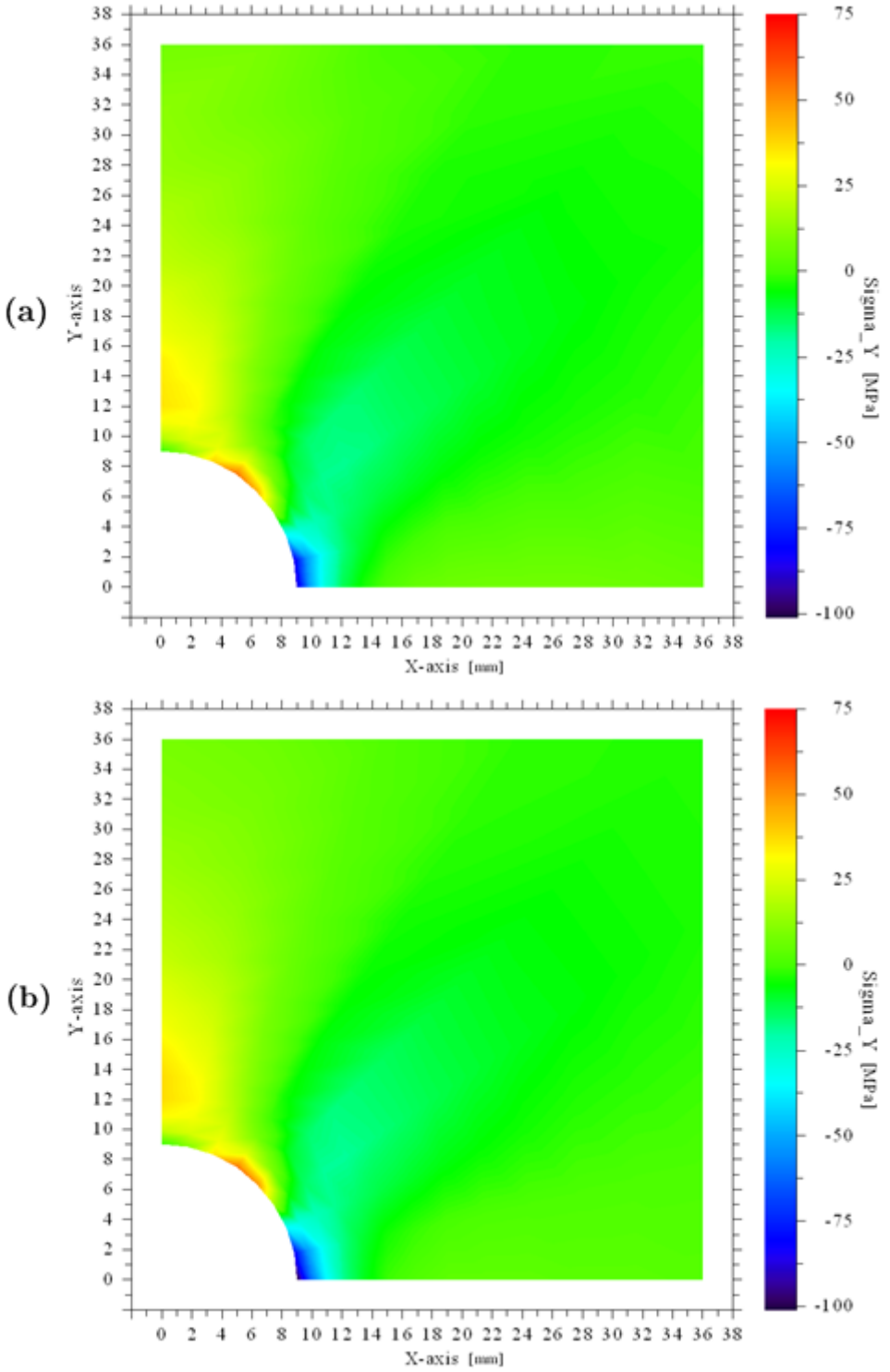

Figure 5.26: σ_y stress of plate problem (a) MLPG (b) Exact

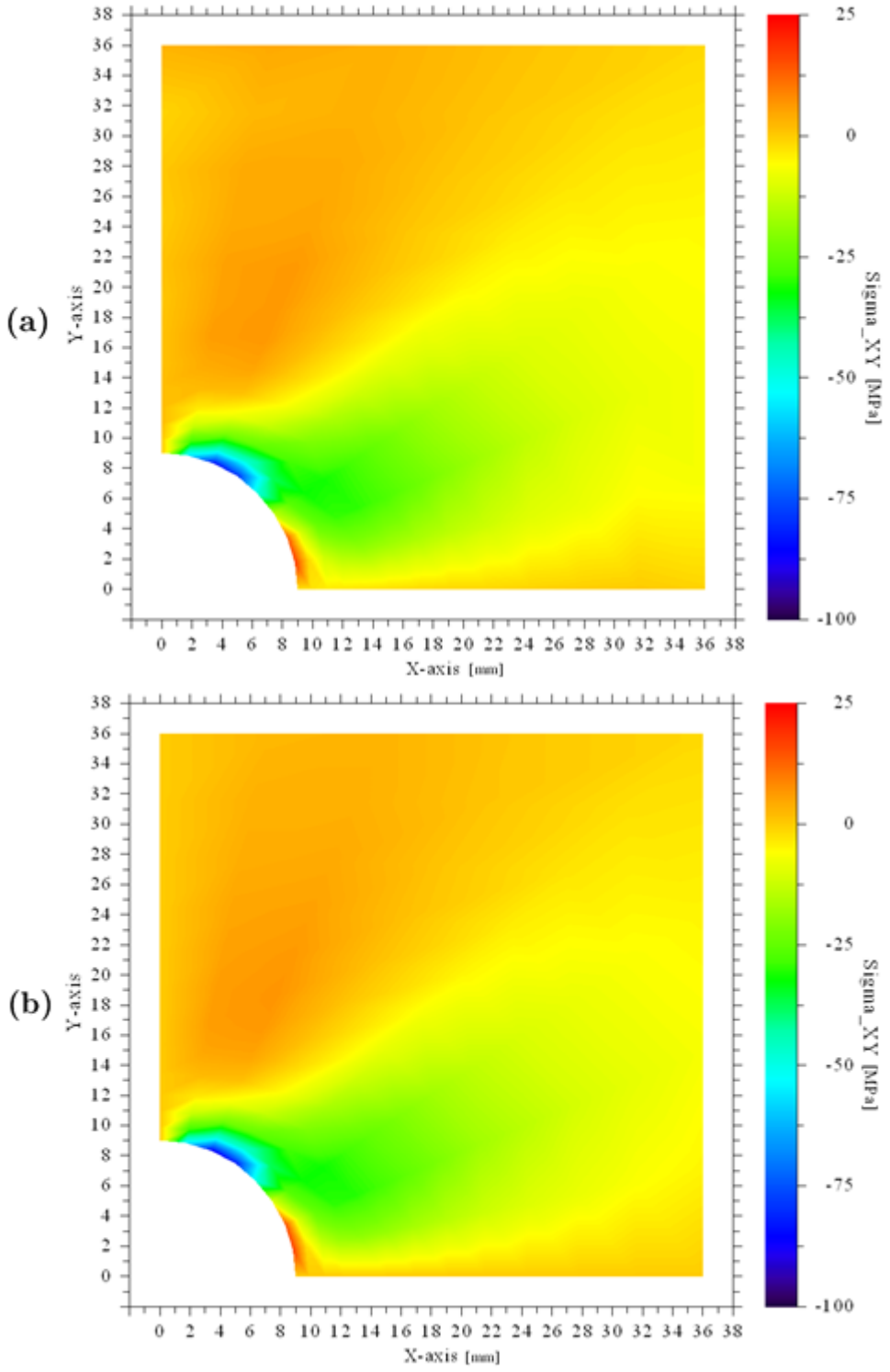

Figure 5.27: σ_{xy} stress of plate problem (a) MLPG (b) Exact

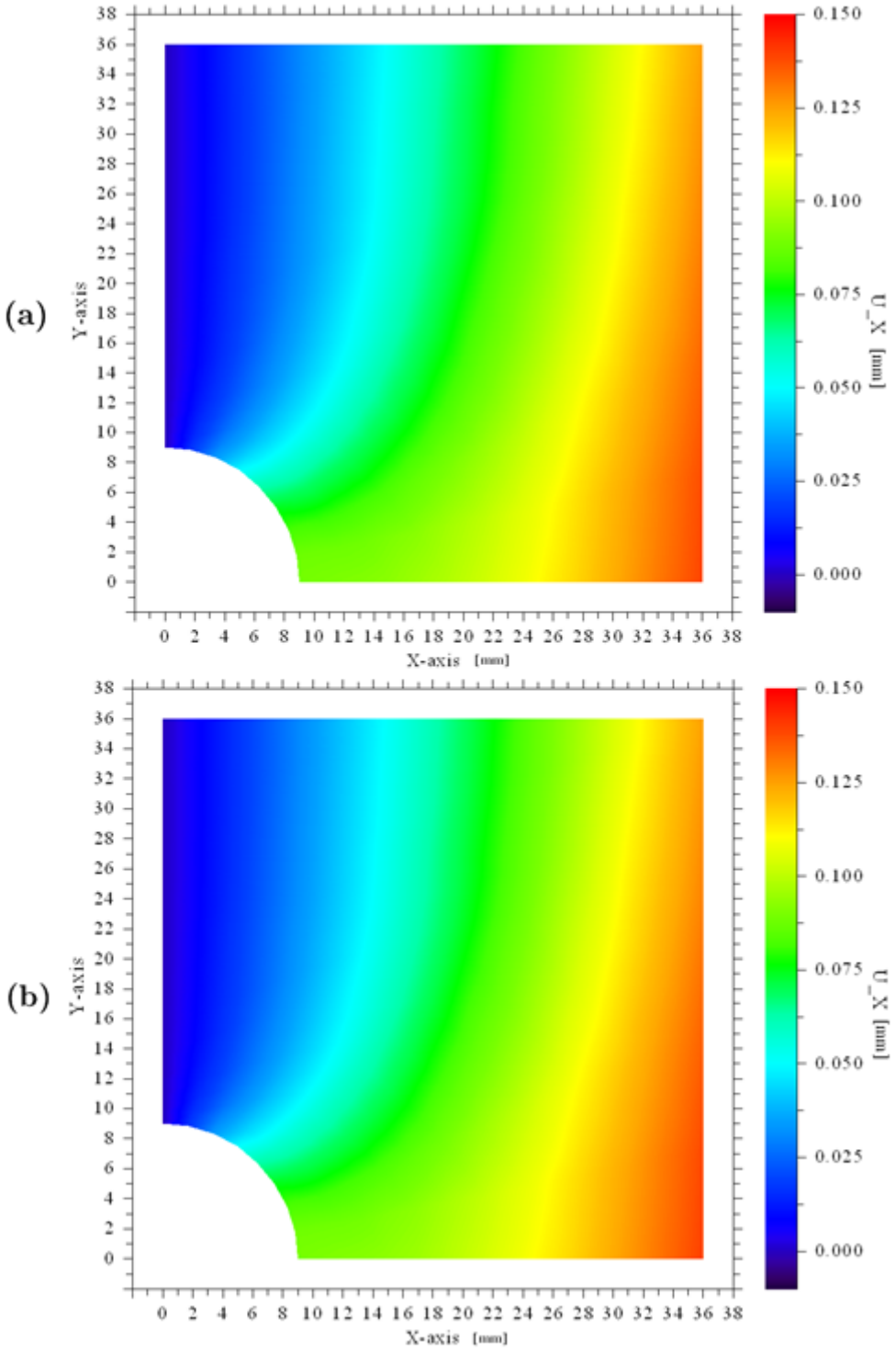

Figure 5.28: u_x displacement of plate problem (a) MLPG (b) Exact

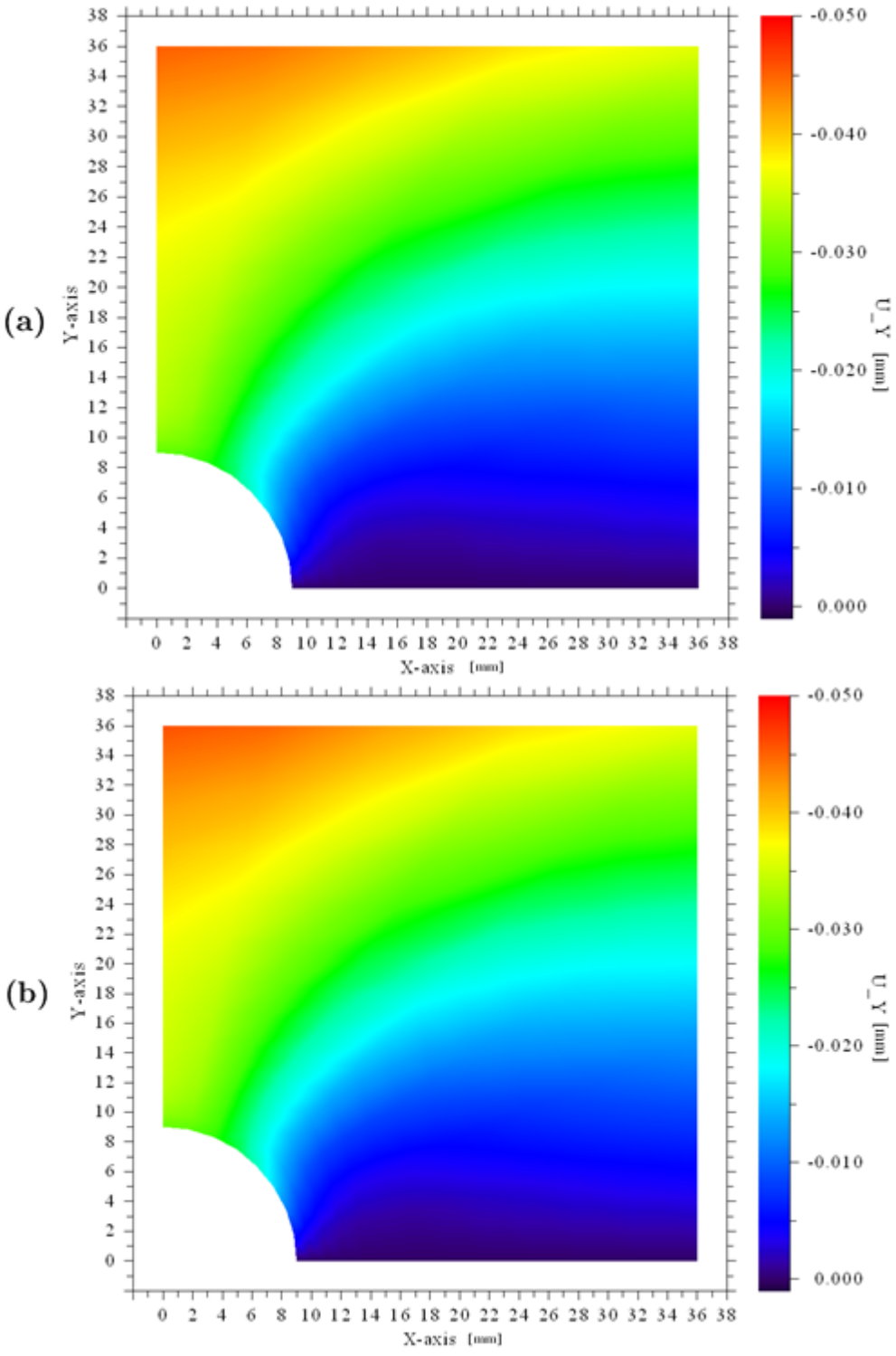

Figure 5.29: u_y displacement of plate problem (a) MLPG (b) Exact

CHAPTER 6

CONCLUSION

In this study, MLPG method, based on local symmetric weak form and moving least squares approximation, is implemented to problems of elasto-statics.

MLPG1 sub-class of MLPG is considered. In MLPG1, MLS weight function is used as test function for weighting local weak form. However, their spaces are chosen differently. Cubic spline function is selected for weighting purpose. 6 monomials are utilized for shape function construction.

For MLPG, accuracy of integration is very important. Thus, total of 64 gauss point is constructed for each sub-domain in order to evaluate integration accurately.

Although MLPG has not Kronecker delta property, essential boundary conditions implemented easily by direct interpolation technique.

Derivatives of field variable, namely stresses, is smooth enough due to 2^{nd} order consistency. Special techniques, such as stress extrapolation or stress recovery, are not required to achieve smooth stresses.

Cantilever beam problem is solved by using uniform nodal distribution. Stresses and displacements are compared with exact solution. Although nodal density is not high, results are satisfactory and exact global energy error is 0.05.

Firstly, effect of integration domain size is investigated for infinite plate with circular hole problem. For three different influence domain sizes, most accurate results can be achieved when integration domain size multiplier is around 2.5.

Secondly, effect of influence domain size is investigated for infinite plate with circular hole problem. Integration domain size multiplier is set to 2.5. Most accurate results can be achieved when influence domain size multiplier is around 4.3.

Finally, problem is solved and results are presented for integration domain size multiplier 2.5 and influence domain size multiplier 4.0. Stresses and displacements are compared with exact solution. Stresses and displacements are accurate enough and exact global energy error is 0.01.

REFERENCES

[1] S N Atluri, J Y Cho, and H.-G. Kim. Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations. *Computational Mechanics*, 24(5):334–347, November 1999.

[2] S N Atluri, H G Kim, and J Y Cho. A critical assessment of the truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods. *Computational Mechanics*, 24(5):348–372, November 1999.

[3] S N Atluri and S Shen. *The meshless local Petrov-Galerkin (MLPG) method*. Tech Science Press, Encino, CA, 2002.

[4] S N Atluri and T Zhu. A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22(2):117–127, August 1998.

[5] S N Atluri and T Zhu. The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics. *Computational Mechanics*, 25(2-3):169–179, March 2000.

[6] S. N. Atluri and Tulong Zhu. New concepts in meshless methods. *International Journal for Numerical Methods in Engineering*, 47(1-3):537–556, January 2000.

[7] T. Belytschko, Y. Guo, W.K. Liu, and S.P. Xiao. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering*, 48(9):1359–1400, 2000.

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

[9] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, 47(6):1189–1214, 2000.

[10] Jiun-Shyan Chen, Chunhui Pan, Cheng-Tang Wu, and Wing Kam Liu. Reproducing Kernel Particle Methods for large deformation analysis of non-

linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):195–227, December 1996.

[11] Jiun-Shyan Chen, Cristina Maria Oliveira Lima Roque, Chunhui Pan, and Sérgio Tonini Button. Analysis of metal forming process based on meshless method. *Journal of Materials Processing Technology*, 80:642–646, 1998.

[12] H.-K. Ching and R. C. Batra. Determination of Crack Tip Fields in Linear Elastostatics by the Meshless Local Petrov- Galerkin (MLPG) Method. *CMES: Computer Modeling in Engineering & Sciences*, 2(2):273–289, 2001.

[13] J.Y. Cho and S.N. Atluri. Analysis of shear flexible beams, using the meshless local Petrov-Galerkin method, based on a locking-free formulation. *Engineering Computations*, 18(1/2):215–240, January 2001.

[14] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *MNRAS*, 181:375–389, November 1977.

[15] Y. T. Gu and G. R. Liu. A meshless local Petrov-Galerkin (MLPG) method for free and forced vibration analyses for solids. *Computational Mechanics*, 27(3):188–198, March 2001.

[16] S R Idelshon, E Onate, N Calvo, and F D Pin. The Meshless Finite Element Method. *International Journal for Numerical Methods in Engineering*, 58(4):893–912, 2003.

[17] G.R. Johnson and S.R. Beissel. Normalized smoothing functions for sph impact computations. *International Journal for Numerical Methods in Engineering*, 39(16):2725–2741, 1996.

[18] G.R. Johnson, S.R. Beissel, and R.A. Stryk. Generalized particle algorithm for high velocity impact computations. *Computational Mechanics*, 25(2):245–256, 2000.

[19] H.-G. Kim and S. N. Atluri. Arbitrary Placement of Secondary Nodes, and Error Control, in the Meshless Local Petrov-Galerkin (MLPG) Method. *CMES: Computer Modeling in Engineering & Sciences*, 1(3):11–32, 2000.

[20] P Lancaster and K Salkauskas. Surfaces Generated by Moving Least Squares Methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[21] Shaofan Li and Wing Kam Liu. Moving least-square reproducing kernel method Part II: Fourier analysis. *Computer Methods in Applied Mechanics and Engineering*, 139(1):159–193, 1996.

[22] Shaofan Li and Wing Kam Liu. Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1):1, 2002.

[23] Shaofan Li, Hongsheng Lu, Weimin Han, Wing Kam Liu, and Daniel C. Simkins. Reproducing kernel element method Part II: Globally conforming Im/Cn hierarchies. *Computer Methods in Applied Mechanics and Engineering*, 193(12):953–987, 2004.

[24] Larry D. Libersky, Albert G. Petschek, Theodore C. Carney, Jim R. Hipp, and Firooz A. Allahdadi. High Strain Lagrangian Hydrodynamics. *Journal of Computational Physics*, 109(1):67–75, 1993.

[25] Larry D. Libersky, Phil W. Randles, Ted C. Carney, and David L. Dickinson. Recent improvements in SPH modeling of hypervelocity impact. *International Journal of Impact Engineering*, 20(6):525–532, 1997.

[26] H Lin and S N Atluri. Meshless Local Petrov-Galerkin (MLPG) method for connection-diffusion problems. *CMES: Computer Modeling in Engineering & Sciences*, 1(2):45–60, 2000.

[27] H Lin and S N Atluri. Analysis of incompressible Navier-Stokes flows by the meshless MLPG method. *CMES: Computer Modeling in Engineering & Sciences*, 2(2):117–142, 2001.

[28] G R Liu. *Meshfree methods : moving beyond the finite element method*, volume 2nd ed. CRC Press, Boca Raton, 2010.

[29] G R Liu and Y T Gu. *An Introduction to Meshfree Methods and Their Programming*. Springer, Dordrecht, 2005.

[30] G R Liu, Y L Wu, and Y T Gu. Application of Meshless Local Petrov-Galerkin (MLPG) Approach to Fluid Flow Problems. In *First Asian-Pasicfic Congress on Computational Mechanics*, pages 20–23, 2001.

[31] Wing Kam Liu and Sukky Jun. Multiple-scale reproducing kernel particle methods for large deformation problems. *International Journal for Numerical Methods in Engineering*, 41(7):1339–1362, April 1998.

[32] Wing Kam Liu, Sukky Jun, Dirk Thomas Sihling, Yijung Chen, and Wei Hao. Multiresolution reproducing kernel particle method for computational fluid dynamics. *International Journal for Numerical Methods in Fluids*, 24(12):1391–1415, June 1997.

[33] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, April 1995.

[34] T Belytschko Lu, L Gu, and Y Y. Fracture and crack growth by element free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering*, 2(3A):519, 1994.

[35] Y.Y. Y Lu, T. Belytschko, and L. Gu. A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 113(3–4):397–414, March 1994.

[36] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013, December 1977.

[37] J. J. Monaghan. Why Particle Methods Work. *SIAM Journal on Scientific and Statistical Computing*, 3(4):422–433, December 1982.

[38] J.J. Monaghan. An introduction to SPH. *Computer Physics Communications*, 48(1):89–96, 1988.

[39] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5):307–318, 1992.

[40] V P Nguyen, T Rabczuk, S Bordas, and M Duflot. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3):763–813, December 2008.

[41] S. N. Atluri Shuyao Long. A Meshless Local Petrov-Galerkin Method for Solving the Bending Problem of a Thin Plate. *CMES: Computer Modeling in Engineering & Sciences*, 3(1):53–64, 2002.

[42] J W Swegle, D L Hicks, and S W Attaway. Smoothed Particle Hydrodynamics Stability Analysis. *Journal of Computational Physics*, 116(1):123–134, January 1995.

[43] S P Timoshenko and Goodier J N. *Theory of Elasticity*. McGraw-Hill, New York, 3rd edition, 1970.

[44] S. P. Xiao and T. Belytschko. Material stability analysis of particle methods. *Advances in Computational Mathematics*, 23(1-2):171–190, July 2005.

APPENDIX A

GAUSS QUADRATURE INTEGRAL

A.1 Isoparametric formulation of Quadrilateral Element

Shape functions for one to one isoparametric mapping for quadrilateral elements are

$$\begin{aligned} N_1 &= \frac{1}{4}(1-\xi)(1-\eta) \\ N_2 &= \frac{1}{4}(1+\xi)(1-\eta) \\ N_3 &= \frac{1}{4}(1+\xi)(1+\eta) \\ N_1 &= \frac{1}{4}(1-\xi)(1+\eta) \end{aligned} \tag{A.1}$$

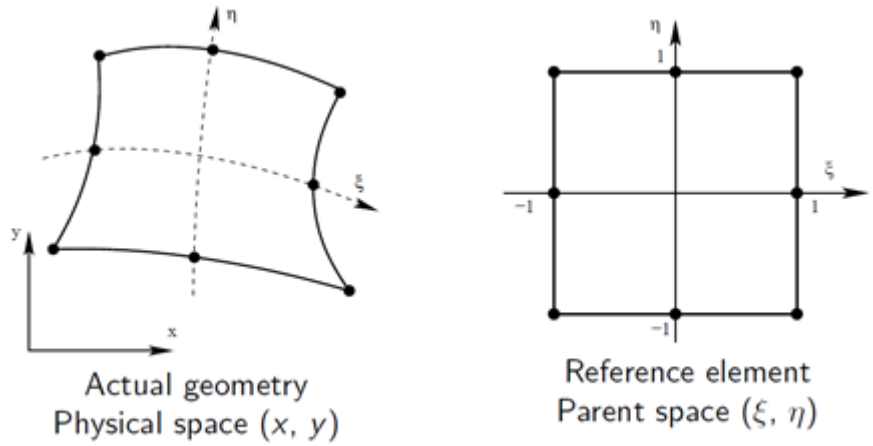

Figure A.1: Isoparametric quadrilateral

Jacobian for transformation is

$$|J| = \begin{vmatrix} \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi_i} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi_i} y_i \\ \sum_{i=1}^{4} \frac{\partial N_i}{\partial \eta_i} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial \eta_i} y_i \end{vmatrix} \tag{A.2}$$

where x_i and y_i are the coordinates of vertices.

Transformation of isoparametric coordinates to global can be done by

$$\begin{aligned} x &= \sum_{i=1}^{4} N_i x_i \\ y &= \sum_{i=1}^{4} N_i y_i \end{aligned} \tag{A.3}$$

A.2 Isoparametric formulation of Line Element

Jacobian for transformation is

$$|J| = \frac{1}{2}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{A.4}$$

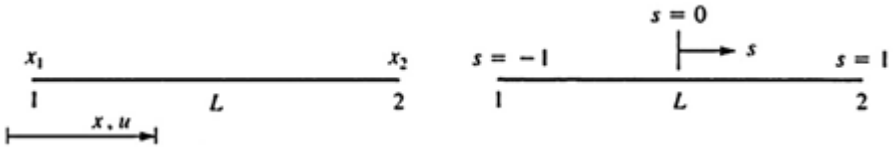

Figure A.2: Isoparametric line

Transformation of isoparametric coordinates to global can be done by

$$\begin{aligned} x &= \frac{1}{2}(x_2 - x_1)\, s + \frac{1}{2}(x_2 + x_1) \\ y &= \frac{1}{2}(y_2 - y_1)\, s + \frac{1}{2}(y_2 + y_1) \end{aligned} \tag{A.5}$$

A.3 Gauss Points

Abscissa and weight for 4 gauss points are given in table A.1

Table A.1: Abscissa and weight for 4 gauss points

i	x_i	w_i
1	-0.8611363115940526	0.3478548451374538
2	-0.3399810435848563	0.6521451548625461
3	0.3399810435848563	0.6521451548625461
4	0.8611363115940526	0.3478548451374538