

THE DEVELOPMENT OF MESHLESS LOCAL PETROV GALERKIN (MLPG) METHOD FOR COMPLEX FLUID FLOW SIMULATION

A THESIS

*Submitted in partial fulfilment of the
requirements for the award of the degree
of*

DOCTOR OF PHILOSOPHY

in

CHEMICAL ENGINEERING

by

VIJAY KUMAR VERMA



DEPARTMENT OF CHEMICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247667 (INDIA)

November, 2017



Acknowledgments

It was fantastic to have the opportunity to work regarding of my research in Indian Institute of technology Roorkee(Uttarakhand). First of all I would like to express my sincere gratitude to my supervisor Dr. Ram Prakash Bharti for his invaluable support, guidance, freedom and encouragement during my thesis work. At many stages in the course of this research, I benefited from his advice, particularly so when exploring new ideas. His positive outlook and confidence in my research inspired me and gave me confidence to present myself. I cannot forget their kind help and cooperation in bringing me to this position. His careful editing contributed enormously to the production of this thesis. I could not have imagined having a better adviser and mentor for my PhD duration. At last I would like to say about him the attitude and nature will help me to become a better mentor in future as well.

I am very grateful to the all chemical engineering staff for their unfailing support and assistance, and the MHRD for funding the PhD research work. I am also grateful to the members of my committee (Dr Surendra Kumar, Dr. Prateek Jha, Dr. Akhilesh Gupta(MIED), and SRC chairman Dr. Shishir Sinha) for their patience and support throughout my research.

I would like to thank my fellow doctoral student Rituraj Singh for their feedback in numerical method, cooperation and of course friendly motivation.

My gratitude to my lab-mates, Prateek Kakkar, Vipin Sachan, Jyoti Toamr, Venky, Sarad Tiwari, and Abhisekh lal for friendly working environment in lab and making my experience something, which I will cherish forever. Thanks to my room met Satya Narayn Murthi for the fun we had in the spare times.

Special Thanks to all my friends for helping me through difficult times , and for all their emotional support, entertainments and caring they provided. I feel fortunate to have support of them with me and making my stay at IIT Roorkee, memorable. I would never

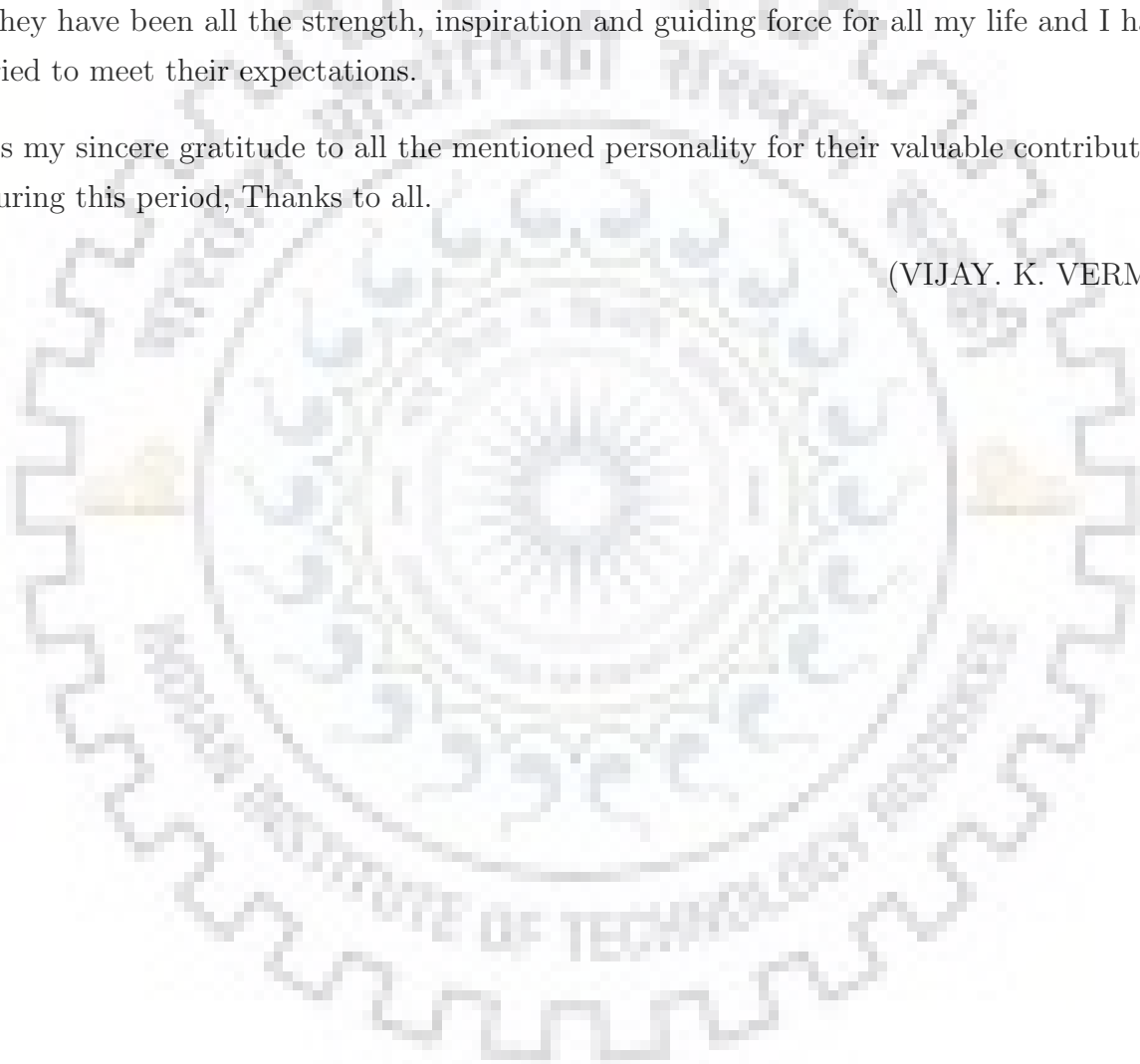
forget the company of you all. Thanks to all my PhD group (Krunal, Kundu, Ravikant, Amit rai, Soumitra, Anil, Dharmendra, Vinit, Nilamber, Ram kishore, Nitin, Lokendra thakur, Ravi shankar, and Deepak Sahu).

I also would like to adored to my lovely juniors Gajendra, Vishal, Vikrant and Umesh.

Finally, I would like to express my reverence and great admiration for my parents, my siblings(Ajai and Kanchan), my sister-in-law, and my wife Priya, to whom I owe everything. They have been all the strength, inspiration and guiding force for all my life and I have tried to meet their expectations.

Its my sincere gratitude to all the mentioned personality for their valuable contribution during this period, Thanks to all.

(VIJAY. K. VERMA)



ABSTRACT

Over the past decades, mesh reducing techniques (meshfree or meshless) have considered as effective numerical techniques for solving engineering problems. Meshing difficulties and consumption of more time for specific engineering problems, in the existing numerical methods (FDM,FEM,FVM), which produces serious numerical errors, were the initial thrust for the development of the meshfree methods.

The predefined numerical methods(FDM, FEM, FVM) is not well suited to those problems which having complex geometries, large deformation of the materials, encountering moving discontinuities such as crack propagation along arbitrary and complex paths, involving meshing or remeshing in structural optimization problems and moving material interfaces etc ([Liu, 2003a](#)). Therefor it is impossible to completely overcome to these difficulties by a well known mesh-based method.

Due to overcome the above mentioned difficulties by the mesh reducing techniques, the meshfree or meshless methods achieved great attentions from past few decade. There are several meshfree techniques have been developed by the previous researchers, in which main attentions achieved by [Lucy \(1977\)](#) and [Gingold and Monaghan \(1977\)](#) by his first developed meshfree technique to solve the astrophysical phenomena, which is well known as SPH method now these days.

Due to inaccurate or having large numerical errors at the boundary in the SPH method, further various types of meshfree techniques have been developed such as diffuse element method(DEM), element free Galerkin(EGG), hp Cloud method, reproducing kernal particle method, finite point method (FPM), partition of unity method (PUM), boundary node method (BNM), local boundary integral equation (LBIE) method, meshless local Petrov-Galerkin method (MLPG) etc.

Between these meshfree techniques having there advantages and disadvantages, not considered as complete meshfree technique because to need a background mesh for integration

purpose of the computational domain except LBIE and MLPG method. The MLPG and LBIE is the only complete meshfree methods which acts as a locally and does not need any background mesh for integration. [Atluri and Zhu \(1998a\)](#) were the first who developed the MLPG method for solid mechanics problems. But for the time being MLPG achieved great attentions to solve structural mechanics, fluid flow (convection-diffusion) phenomena and other engineering field problems successfully. Hence Meshless Local-Petrov Galerkin (MLPG) is selected as a numerical tool for our objectives because of its truly meshfree characteristics for the various types of bench mark engineering problems having one dimensional and two dimensional space, including diffusion problems and fluid flow examples to check its efficiency and accuracy. Indeed non-Newtonian fluid flow behavior is encountered in almost all the chemical as well as allied processing industries. Therefore this method is extended for the one dimensional and two dimensional complex fluid flow simulations, because of all the natural rheology follows the non-Newtonian behaviors and are very useful such as sewage industries, paint industry, water treatment plants and other process industries etc. Due to very limited or negligible research has been done in the complex fluid flow simulations by using this method. Therefore In this study, a numerical investigation is done to observe the complex fluid flow behavior and characterize the non-Newtonian fluid properties.

Based on considering available literature in chapter 2, the essential features and its numerical formulation of MLPG method, basic numerical techniques to impose the boundary conditions are explained in chapter 3. According to study the MLPG needs three basic steps to formulate on a local sub-domain Ω_s in to entire computational domain Ω , as (1) a weight function $w(x)$, to convert the PDE in to weaker consistency, (2) a basis function $P(x)$ and (3) a shape function, which is formulated with the help of basis function $P(x)$ by using non-element interpolation technique such as moving least square (MLS) approximations ([Lancaster and Salkauskas, 1986](#)). It is clearly distinguished between support domain Ω_{tr}^J and test domain Ω_{te}^I , and to select the suitable domain sizes for multidimensional spaces. The types of MLPG and their applications are explained for different field problems as given in [Atluri and Shen \(2005\)](#). Due to the complex nature of MLPG integrational technique described in [Atluri et al. \(1999b\)](#), the Gauss integration technique for quadrature domain Ω_q are considered, which is well explained in the book "Meshfree Methods, moving beyond the finite element method" ([Liu, 2003a](#)).

A computational tool (C++) for MLPG has been developed to validate the basic bench mark CFD problems for 1D space diffusion problems given in the text book [Versteeg and Malalasekera \(2007\)](#) and 2D diffusion problems. For one dimensional problem the

numerical solutions are validated with exact solution, and for 2D space the numerical solution are validated with commercial CFD package FLUENT 6.3. The results shows very close agreement, thus it is expected that MLPG method (which is a truly meshless) is very promising in solving engineering heat conduction problems in a great extent. Afterwords 1D convection-diffusion problems (Lin and Atluri, 2000) at high Peclet number by using suitable upwind scheme to avoid its wiggle nature at high Pe , is validated, and found very good agreement for convection dominated problems, which is discussed in detailed as in chapter 4.

In the first attempts to validate and verify the use of proposed in-house MLPG based solver for complex fluid flow simulations for multidimensional space. In order to fulfill this objective, the one-dimensional incompressible steady flow of non-Newtonian power-law fluids through rectangular channel flow problem has been considered. It is governed by the mass continuity and Navier-Stokes equations. The fluid rheology is governed by the non-Newtonian power-law fluid viscosity model. The computational solutions using in-house solver has been compared and validated with the exact solution. The influence of computational nodes, $N=21, 41$ and 51 is also analyzed. In particular, the velocity profile, shear-stress profile and the values of pressure drop have been obtained a compared for a range of power-law index ($0.4 \leq n \leq 1.8$). Since the shear-dependent viscosity, special attempts was required to handle the oscillations in numerical solutions due to shear-dependent nature of viscosity and shear-stress. The present results are corresponding well with the exact solution over the ranges of conditions considered herein. Hence, the present work builds up a new numerical algorithm for their reliable use in the understanding of complex fluid flow phenomena and lastly summarizes the present work and opens up the scope of future work.

Subsequently, the two dimensional incompressible steady flow of non-Newtonian power-law fluids through square channel flow problem has been considered at different Reynolds number(Re). In the non-Newtonian fluid rheology is also governed by the power-law fluid viscosity model which is governed by continuity and momentum equations in both the directions. The space domain for $N=11 \times 11$ and 21×21 is chosen. The numerical solutions has been compared with exact solution. The power law index between $0.6 \leq n \leq 1.4$ is varied at different Re . Since due to shear-dependent viscosity, special attempts was required to handle the oscillations in numerical solutions. A good agreement has been observed for different power-law index (n) at $Re = 1$ and 20 . But at high Reynolds number $Re \leq 1$, the well established upwind scheme for Newtonian flow is not enough to avoid its wiggle nature due to shear-dependent nature of viscosity and shear-stress. Therefore, more

investigation is needed for non-Newtonian fluid rheology at higher Reynolds number flow of shear-dependent viscous flow.

Finally, the last chapter summarizes the findings of the present thesis and provides the possibilities of future scope of the research.



Contents

Acknowledgments	i
ABSTRACT	iii
List of Figures	xi
List of Tables	xv
Nomenclature	xvii
List of publications	xxi
1 Introduction	1
1.1 Classical CFD approaches	2
1.1.1 Finite difference method (FDM)	3
1.1.2 Finite element method (FEM)	3
1.1.3 Finite volume method (FVM)	4
1.2 Meshfree method	6
1.3 Organization of thesis	9
2 Literature review	11
2.1 Need of Mesh Free Methods	11
2.2 Meshless methods	12
2.2.1 Strong form based meshless method	13
2.2.2 Weak-strong form based meshless method	14
2.2.3 Weak form based meshless method	14
2.3 MLPG application in diffusion problems	20
2.4 MLPG application in convection-diffusion problems	21
2.5 Conventional approaches in analysis of complex fluid flow	24
2.6 Summary of literature review	25
2.7 Objectives	26
3 Mathematical formulation and numerical methodology	29
3.1 Meshless local Petrov Galerkin (MLPG) method	29
3.2 Concept of support and influence domains	30

3.3	Determination of dimension of a support domain	31
3.4	MLPG formulations	33
3.4.1	Basis function approximation	34
3.4.2	Choice of weight functions	35
3.4.3	Moving least square (MLS) approximation	39
3.4.3.1	Formulation of MLS shape functions	39
3.4.3.2	Properties of MLS shape functions	46
3.4.4	Flowchart of the shape function calculation	47
3.4.5	Imposition of boundary conditions	48
3.4.6	Direct interpolation method	52
3.5	Integration techniques in MLPG	53
3.5.1	Gaussian quadrature	55
3.6	MLPG algorithm	58
3.6.1	Solution of convection-diffusion problems	59
3.6.2	Solution of Navier-Stokes equations	60
3.7	Dimensionless parameters used	62
3.7.1	Reynolds Number (Re):	62
3.7.2	Prandtl Number (Pr):	62
3.7.3	Peclet Number (Pe):	63
4	Application of MLPG solver in diffusion and convection-diffusion problems	65
4.1	Solution of diffusion equation	66
4.1.1	Steady state heat conduction	66
4.1.2	Steady state heat conduction with uniform heat generation	71
4.1.3	Convective cooling of a circular fin	75
4.1.4	Heat conduction in a square slab	77
4.1.5	Heat conduction in rectangular slab	80
4.2	Solution of convection-diffusion equation	82
4.2.1	Meshless upwind schemes	84
4.2.1.1	MLPG Upwinding Scheme I (US-I)	85
4.2.1.2	MLPG Upwinding Scheme II (US-II)	86
4.2.2	Steady state convection-diffusion	87
4.2.3	Steady state convection-diffusion without source	90
5	Application of MLPG solver in Navier-Stokes problems with Newtonian and non-Newtonian fluids	95
5.1	Introduction	95
5.2	1-D fully developed flow of Newtonian and non-Newtonian fluids through a channel	98
5.3	2-D flow of Newtonian and non-Newtonian fluids through a channel	107
6	Conclusions and future directions	129
6.1	Recapitulation of purpose and findings	129
6.2	Recommendations for future research directions	132

Bibliography

133





List of Figures

3.1	Types of support domain used	30
3.2	Influence domain of a point x	31
3.3	Pascal triangle for general selection of basis function (Liu and Gu, 2005).	34
3.4	Weightage of influencing points in subdomain.	35
3.5	Weight function and its 1 st derivatives	38
3.6	Weight function and their derivatives (W1: cubic spline; W2: quartic spline; W3: exponential)(Liu and Gu, 2005).	38
3.7	The appropriate function $u^h(x)$ and the nodal parameter u_i in the MLS approximation (Atluri and Zhu, 1998a).	41
3.8	Shape function and its first derivative at point of interest $x = 11$ and $x_I = (1, 2, 3, \dots, 21)$	45
3.9	Shape function and its first derivative in 3-D space by using cubic spline weight function (Liu and Gu, 2005).	46
3.10	Flowchart for computation of MLS shape functions.	47
3.11	u_I^h is the fititious nodal data at node I and u_I is the exact nodal value at node I for interpolation	49
3.12	Schematics of the MLPG method and the boundary intersection domain Atluri et al. (1999b)	51
3.13	An integration domain modeled using the MFree local weak-form. Ω_q and Ω_w for field nodes, and the support domain Ω_s for a Gauss quadrature point x_q (Liu and Gu, 2005).	55
3.14	Transformation of first quarter of a circle into a standard square (Liu, 2003a)	56
3.15	The sample of Gauss points in circular domain presented for $i = 2$ and $i = 120^{th}$ nodes in 156 uniformly distributed nodes in rectangle domain.	58
3.16	MLPG flow chart for any convection-diffusion problems.	60
3.17	MLPG flow chart for any Navier-Stokes equations.	61
4.1	Heat conduction in a bar of length L	66
4.2	Comparison of numerical results with analytical solution for $n = 11$ and 22 nodes.	68
4.3	Heat conduction in large plate.	71
4.4	Comparison of numerical MLPG results with analytical solution for temperature distribution in a plate.	73
4.5	Convective cooling of a fin.	75
4.6	Temperature distribution along the length of fin.	76
4.7	Square slab exposed to isothermal conditions.	78

4.8	Rectangular slab exposed to both isothermal and insulated conditions. . .	80
4.9	Temperature distribution in a rectangular plate	82
4.10	Illustration of the crosswind diffusion problem with $Pe = 100$ ((Leonard, 1979)). Exact (left) and optimal upwinding (right) solutions.	83
4.11	MLPG method without upwinding(Lin and Atluri, 2000).	85
4.12	MLPG upwinding scheme I (US-I) (Lin and Atluri, 2000).	85
4.13	US-I illustration (Lin and Atluri, 2000).	86
4.14	MLPG upwinding scheme II (US-II) (Lin and Atluri, 2000).	86
4.15	US-II illustration (Lin and Atluri, 2000).	87
4.16	1-D convection-diffusion flow domain of length L	88
4.17	Results of Case 1 using MLPG, MLPG-US-I and MLPG-US-II for $1 \leq Pe \leq 1000$	89
4.18	Results of Case 2 using MLPG, MLPG-US-I and MLPG-US-II for $1 \leq Pe \leq 1000$	90
4.19	Distribution of $\phi(x)$ for case A.	91
4.20	Distribution of $\phi(x)$ for case B.	91
5.1	Velocity profiles for $n = 0.8$ and 0.4 by using 21 uniformly distributed computational nodes.	100
5.2	Velocity profiles for shear-thinning ($0.2 \leq n \leq 1$) fluids by using $N = 21$ computational nodes	102
5.3	Velocity profiles for shear-thickening ($1 \leq n \leq 1.8$) fluids by using $N = 21$ computational nodes	103
5.4	Velocity profiles for power-law fluids ($0.4 \leq n \leq 1.6$) fluids by using $N = 41$ computational nodes	104
5.5	Velocity profiles for power-law fluids ($0.4 \leq n \leq 1.6$) fluids by using $N = 51$ computational nodes	106
5.6	Pressure gradient as a function of power-law index ($0.2 \leq n \leq 1.8$) and computational nodes ($N = 41$ and 51)	107
5.7	Flow through a channel.	108
5.8	Fully developed velocity profile at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N = 121$ nodes.	113
5.9	Viscosity profiles at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N = 121$ nodes.	114
5.10	Strain tensor profiles at the outlet of channel for different n at $Re = 1$. Comparison of exact with numerical solutions for $N = 121$ nodes.	115
5.11	Fully developed velocity profile at the outlet of channel for different n at $Re = 1$. Comparison of exact with numerical solutions for $N = (441)$ nodes.	116
5.12	Viscosity profiles at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N = (21 \times 21)$ nodes.	117
5.13	Strain tensor profiles at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N = (21 \times 21)$ nodes.	118

5.14 Fully developed velocity profile at the outlet of channel for different n at fixed $Re = 20$. Comparison of exact with MLPG solutions. 120

5.15 Viscosity profiles at the outlet of channel for different n at fixed $Re = 20$. Comparison of exact solution with MLPG solutions. 121

5.16 Viscosity profiles at the outlet of channel for different n at fixed $Re = 20$ with 21×21 nodes. Comparison of exact solution with MLPG solutions. . . 122

5.17 Pressure contour at different n by using MLPG with $N = (21 \times 21)$ nodes at $Re = 20$ 123






List of Tables

2.1	MLPG methods	18
3.1	MLPG types and their characteristics	52
4.1	Temperature distribution in a rod calculated for $n = 11$ nodes by using linear (Lin) and Quadratic (Quad) basis functions.	69
4.2	Temperature distribution in a rod calculated for $n = 21$ nodes by using linear (Lin) basis function.	70
4.3	Temperature distribution in a plate calculated by using $n = 11$ nodes and linear (Lin) basis function.	74
4.4	Temperature distribution in a plate calculated by using $n = 21$ nodes and linear (Lin) basis function.	74
4.5	Temperature distribution along the length of fin calculated by using $n = 11$ nodes and linear (Lin) basis function.	77
4.6	Temperature distribution in square slab calculated by using $n = 25$ nodes.	79
4.7	Temperature distribution in square slab calculated by using $n = 121$ nodes.	79
4.8	Distribution of $\phi(x)$ for case A of convection-diffusion process using $n = 11$ computational nodes.	92
4.9	Distribution of $\phi(x)$ for case B of convection-diffusion process using $n = 11$ computational nodes.	92
5.1	Optimized values of β for a range of power-law index (n) and computational nodes (N).	105
5.2	Velocity, viscosity and stress fields for shear-thinning ($n = 0.4$) fluid by using $N = 41$ nodes.	105
5.3	Optimized value of β and radius of support domains at $Re = 1$ for a range of power-law index ($0.6 \leq n \leq 1.4$). Here Δ is the shifting of test domain.	115
5.4	Optimized value of β , radius of support domains (r_s) and correction factor (f) for different n at $Re = 20$ by using MLPG with $N = 441$ computational nodes.	119
5.5	Comparison of local velocity and viscosity values with exact solution nodes at $Re = 1$ by using for 11×11 computational nodes. (Here A, B and C represents for exact, MLPG and MLPG-US-II solutions).	124
5.6	Comparison of local velocity and viscosity values with exact solution nodes at $Re = 1$ by using for 21×21 computational nodes. (Here A, B and C represents for exact, MLPG and MLPG-US-II solutions).	125

5.7 Comparison of local velocity and viscosity values with exact solution nodes at $Re = 20$ by using for 21×21 computational nodes. (Here A, and B represents for exact, and MLPG solutions). 126



Nomenclature



$\mathbf{a}(\mathbf{x})$	coefficient of basis function and its vector form
BNM	boundary node method
CFD	computational fluid dynamics
CV	control volume
dc	distance between two neighbouring nodes
D_s	diameter of support domain
DEM	diffuse element method
ds	radius of support domain
DSD	deforming spatial domain
EFG	element free Galerkin
FDM	finite difference method
FEM	finite element method
F_I	global vector/load vector
f_i	body force, N
FPM	finite point method
FVM	finite volume Method
Gr	Grashof number, dimensionless
H	height of domain, m
IDE	integral differential equation
I_2	second invariant of the rate of strain tensor
K_{IJ}	stiffness matrix
L	length of domain, m
LBIE	local boundary integral equation

MLPG	meshless local Petrov-Galerkin
n	power law index
n_j	unit normal vector (outward to the boundary)
ODE	ordinary differential equation
$\mathbf{P}(\mathbf{x})$	basis function and its vector form
P	pressure, dimensionless
PDE	partial differential equation
Pe	Peclet number, dimensionless
Pr	Prandtl number, dimensionless
PUM	particle of unity method
Ra	Rayleigh number, dimensionless
RBF	radial basis function
Re	Reynolds number, dimensionless
RKPM	reproducing kernel particle method
Re_L	local Reynolds number, dimensionless
r_s	radius of support domain, m
r_q	radius of quadrature domain, m
SPH	smooth particle hydrodynamics
SST	stabilized space time
T	temperature, K
u_I^h	fictitious nodal value
U_{avg}	average velocity at inlet, m/s
U_{max}	maximum velocity at inlet, m/s
w_i	weight function in fixed direction
$w(x)$	weight function
<i>Greek symbols</i>	
α_t	thermal diffusivity, m^2/s
α	dimensionless parameter
Ω_s	local support domain
ϕ	shape function

Γ_t	boundary for Neumann condition
Γ_u	boundary for essential condition
Γ	boundary of computational domain
μ	dynamic viscosity, kg/m.s
ν	kinematic viscosity, m ² /s
Ω	computational domain
β	dimensionless parameter
Ω_{te}^I	local test domain
Ω_{tr}^J	local support domain
ϕ_{te}^I	test function
τ	perterbation term
ϕ_{tr}^J	trial function
ε_{ij}	strain rate
δ_{ij}	kronecker delta function
λ	lagrange multiplier
τ_{xy}	shear stress, N/m ²
Ω_q	quadrature domain
<i>Subscripts</i>	
<i>min, max</i>	minimum, maximum
<i>x, y</i>	direction in x and y coordinates
<i>Superscripts</i>	
<i>a</i>	analytical
<i>num</i>	numerical



List of publications

Refereed International Journals

1. **Vijay K. Verma** and Ram P. Bharti. Development of meshless local Petrov Galerkin (MLPG) method for computation of incompressible complex fluid flow. *Journal of Computational Physics*, Communicated (2017).







Chapter 1

Introduction

General transport processes occurring on natural and industrial scale levels such as biological, mechanical, geological, electrical, chemical, electronic, etc., are widely referred to under the umbrella of transport phenomena. Due to the simultaneous occurrence of these processes, such problems are generally quite complex in nature. These transport phenomena problems are often governed by the set of differential or integral equations in terms of the rate of change of fluxes of transport quantities. The governing field equations are coupled together due to simultaneous occurrence of the processes. Additional complexities arise from the initial and boundary conditions of practical problems. It, in turn, yields inherent complex dependencies of the solution variables on each other, in conjunction to the processing parameters. Unfortunately, exact solutions of such problems can only be obtained for the limiting processing conditions. The numerical procedures are, therefore, adopted to obtain approximate solutions to the general practical problems which strongly depend on advances in numerical analysis and mathematical modeling.

The birth certificate of numerical analysis is given by [Neumann and Goldstine \(1947\)](#) through his first paper on “*numerical inverting of matrices of high order*” for modern algebraic problems. Their work was focused on the rounding-off errors in processing of higher order matrices, which is considered as the basis the modern scientific computing. The present thesis aims toward the enrichment of the scientific computing of fluid flow

and heat transfer in complex fluids, which is commonly categorized under the umbrella of computational fluid dynamics (CFD).

CFD problems are physically governed by the conservation principles of mass, momentum and thermal energy. These principles are generally expressed in the form of ordinary differential equations (ODE), partial differential equations (PDE) and integro-differential equations (IDE). They are further subjected to a set of physically realistic initial and boundary conditions. Due to the inherent complexities associated with the governing physics, a numerical analysis of such problems, first, requires a suitable technique to transform ODE/PDE/IDE into a set of algebraic equations. Then, a solution of the set of algebraic equations can be obtained by using a direct or iterative numerical techniques. A mathematical process of transforming ODE/PDE/IDE into a set of algebraic equations is termed as discretization, which forms one of the major steps of CFD cycle.

Over the decades, numerous CFD techniques have evolved in the scientific literature for the efficient discretization of ODE/PDE/IDE. These techniques may broadly be categorized into two major groups, namely, conventional or traditional and advanced discretization approaches. Traditionally used methods are finite difference method (FDM), finite element method (FEM) and finite volume method (FVM). These well known classical methods are completely mesh-based methods which have been employed as very powerful numerical tools for solving various engineering problems. Advanced techniques amongst others include lattice Boltzmann method (LBM), boundary element method (BEM), spectral element method (SEM), direct numerical simulations (DNS) and meshless methods, etc. The framework of some of these advanced methods is widely different from that of the classical techniques. While the present thesis is focused on one of the advanced methods viz meshless methods, classical approaches are also briefly discussed subsequently.

1.1 Classical CFD approaches

Classical CFD approaches viz finite difference, finite element and finite volume methods are appraised herein.

1.1.1 Finite difference method (FDM)

The advent of most commonly and useful method to solve the differential equations (ODE/PDE) in a bounded area, i.e., finite difference method (FDM), began in the early 1950's. It was probably brought into the numerical picture by Euler in the year of 1968 (Blazek, 2001). The basic idea of FDM is transformation of continuous derivatives in differential equations into the discrete functions of dependent variables. The region, over which the independent variables in PDE's are defined, is replaced by a grid (or mesh) of finite points. PDE's at each grid point are approximated from their neighboring values by using an expansion of Taylor's theorem. This process yields a set of $n \times m$ algebraic equations for n grid points and m dependent variables.

FDM has been widely used in the past decades to characterize the fluid dynamics and heat transfer of both Newtonian and non-Newtonian flow behavior for wide ranging flow geometries. For instance, oscillating flow past a cylinder (Wang and Dalton, 1991), shear flow over a circular cylinder (Lei et al., 2000), steady flow of a power-law fluid past a square cylinder (Gupta et al., 2003; Paliwal et al., 2003), simulation of moving particles (Perrin and Hu, 2006), natural convection around a square, horizontal, heated cylinder placed in an enclosure (De and Dalal, 2006), fluid structure interaction (FSI) modeling with the deforming-spatial domain(DSD)/stabilized-space-time(SST) method for the fluid and finite difference method for the structure (Tain, 2014), etc. In addition to simplicity, the advantage of FDM lies in the possibility to achieve higher-order accuracy of the spatial discretization and easy to implement. However, the range of application is clearly restricted (Anderson, 1995; Blazek, 2001; Sharma, 2016) due to its requirement of structure grid.

1.1.2 Finite element method (FEM)

Finite element method or analysis (FEM or FEA) was originally introduced and employed for structural analysis by Turner et al. (1956). In this method, an arbitrary computational domain is divided into smaller, simpler and regular parts called 'elements', and the equations formulated for every element are then assembled into a large system of equations

that model the entire domain/problem. Depending on the element type and required accuracy, a certain number of Gauss points at the boundaries and/or inside an element are specified, where the solution of flow problem has to be found. Furthermore, the approximated shape functions for the solution variables have to be defined for interpolation of dependent variables inside the element. This process yields a set of algebraic equations, which are numerically solved to obtain the field variables. In the early 1990's, FEM gained its popularity in the solution of the Euler's and the Navier-Stokes equations. [Price et al. \(1968\)](#) reported that FEM requires fewer nodes and less computational time than FDM to achieve comparable accuracy in solving one-dimensional convection-diffusion equation with a trapezoidal-rule scheme. However, advantages of computational time in FEM are usually lost with higher dimensional problems due to the relatively more complex matrices generated by FEM than those in FDM or FVM as the number of dimensions increases. Nevertheless, FEM is attractive because of its integral formulation and the use of unstructured grids, which are preferable for complex flow geometries. Over the time, it has been widely used in the computational analysis of complex fluid flow such as vortex-induced cross-flow and in-line oscillations of a circular cylinder at low Reynolds numbers ([Mittal and Kumar, 1999](#)), lift on a slightly deformable and freely rotating and translating cylinder in two-dimensional (2-D) channel flow ([Zhao and Sharp, 1999](#)), stability of elliptical cylinders in 2-D channel flow ([Zhao and Sharp, 2000](#)), flow simulation of injection pultrusion ([Liu, 2003b](#)), solid-liquid two phase flows ([Wan and Turek, 2007](#)), simulation of a pulsatile non-Newtonian flow past a stenosed 2-D artery with atherosclerosis ([Tian et al., 2013](#)), and deforming-spatial-domain/stabilized-space-time (DSD/SST) method in computation of non-Newtonian fluid flow and heat transfer with moving boundaries ([Tian et al., 2014b](#)), etc.

1.1.3 Finite volume method (FVM)

It was first employed by [McDonald \(1971\)](#) for the simulation of 2-D inviscid flows. In FVM, computational domain is discretized into small control volumes (CV). The flow variables may be defined either on the cell center or node center of CV. Accordingly, the fluxes of

variable are defined either on the cell centroid or face centroid ([Anderson, 1995](#); [Blazek, 2001](#); [Sharma, 2016](#)). The volume integrals appearing in a IDE that contain a divergence term are converted into the surface integrals by divergence theorem. It allows for the strict conservation of physical quantities in local CV as well as in global computational domain. The main advantages of FVM is that the spatial discretization is carried out directly in the computational space. FVM takes the advantages of both FDM and FEM to become the robust method in computational flow simulations. Nowadays, it is frequently used in fluid dynamics and other related potential problems. FVM is efficiently implements on both structured as well as unstructured mesh using Navier-Stokes equation ([Dalal et al., 2008](#); [Sharma, 2016](#)). It has been used in wide ranging applications including inclined complicated cavity with spatially variable wall temperature ([Dalal and Das, 2005](#)), fluid flow and heat transfer from a triangular cylinder placed in a channel ([De and Dalal, 2007](#)), fluid flow and heat transfer around a square cylinder at incidence using unstructured grid ([Ranjan et al., 2008](#)), flow around a row of square cylinders ([Kumar et al., 2008](#)), complex flows involving moving boundaries and irregular geometries ([Wang and Tian, 2017](#)), and periodically fully developed heat and fluid flow characteristics in a furrowed wavy channel ([Sarkar et al., 2017](#)), etc. Its application in computations of non-Newtonian fluid flow such as power law fluids across a circular and square cylinders ([Bharti et al., 2006](#); [Sahu et al., 2010](#)), heat transfer from a sphere to non-Newtonian power law fluids ([Dhole et al., 2006](#)), immersed boundary model for simulations of complex flow [Wang et al. \(2017\)](#) suggest wide applications of FVM in process industries.

The above discussed conventional numerical methods (FDM, FEM, FVM) are based on pre-defined grid or mess structure to discretize the differential equations. In such a mesh, each data point has a fixed number of predefined neighbors. The connectivity between these neighbors is defined by some mathematical operators to construct the system equations and to know the domain properties. These grid points are used to evaluate the discrete quantities of physics such as velocity, pressure and viscosity, etc. The accuracy of the classical CFD approaches depends on the number of grid points. Therefore, large number of grid points are required to find the best possible numerical results. This improved accuracy,

in turn, enhances the computational efforts. Large deformation of the material or sharper changes in the gradients are generally observed in the flow simulations, the connectivity of the mesh can be difficult to maintain without introducing an error in the simulation. Therefore, these numerical approaches faces difficulty in application to the complex flow problems such as large element deformation, projectile impact, fluid interaction, complex free surface flows, where the mesh generation imposes great challenges and time consuming phenomena. Significant efforts are being made, over the years, to reduce the computational efforts in obtaining the most accurate results. One alternate outcome is meshless methods. A brief discussion of these approaches is made in the subsequent section.

1.2 Meshfree method

Meshfree (or meshless) methods, as an alternative numerical techniques to eliminate the drawbacks of conventional mesh-based methods (FDM, FVM & FEM), proposed initially as smooth particle hydrodynamics (SPH) method for modeling in an astrophysical phenomena (Lucy, 1977; Gingold and Monaghan, 1977). SPH, one of the oldest meshfree method, treats data points as physical particles having finite mass and density. These particles can move around over the time and transport required information with them. SPH was first applied in solid mechanics by Libersky et al. (1993). The main drawbacks of SPH are inaccurate results near the boundaries and tension instability (Sweble et al., 1995).

Over the decades, many meshfree method have been developed based on global/local and strong/weak forms of discretization of PDEs, such as diffuse element method (DEM) (Nayroles et al., 1992), element free Galerkin (EFG) method (Belytschko et al., 1994), reproducing kernel particle method (RKPM) (Liu and Jun, 1995; Liu et al., 1995b), hp-cloud method (Duarte and Oden, 1996a,b), finite point method (FPM) (Onate et al., 1996a), partition of unity method (PUM) (Babuska and Melenk, 1997), boundary node method (BNM) (Mukherjee and Mukherjee, 1997a), local boundary integral equation (LBIE) method (Zhu et al., 1998a,b), meshless local Petrov Galerkin method (MLPG) (Atluri and Zhu, 1998a,b), etc. A progress of meshfree methods over the last two decades

have been thoroughly reviewed by [Chen et al. \(2017a\)](#). Meshless methods are based on either global weak forms or local weak forms of conservation equations, namely, PDE. The major challenge with meshless methods based on the global weak forms revolves around the effective evaluation of the integrals in the weak form. Most of the above noted meshfree methods, except LBIE and MLPG, needs background mesh for numerical integration of the weak form. The background cell integration does not lead to be truly meshless. Therefore, these methods are not truly meshless method. The construction of background mesh is also associated with additional cost. In order to develop a truly meshless method, the local weak forms must be used to avoid the need of background meshing. Use of local weak forms is the main point of the departure of LBIE and MLPG methods.

Both MLPG and LBIE use a weak form of PDE for a local sub-domain, and use numerical integration in thier local sub-domain of arbitrary shapes such as circle, ellipse, rectangle and parallelopiped for 2-D space, and sphere and cube for 3-D space. Due to this natural approach, these methods are considered as truly meshless methods.

The main objective of truly meshless methods is to get rid of, or at least alleviate the difficulty of meshing and re-meshing the entire structure by only adding or deleting nodes in the entire computational domain/structure. Moreover, as in the FEM, the construction of even a C^1 trial function approximation is difficult, and has been unsatisfactory so far. However, the trial functions commonly employed in the MLPG method, can achieve high-order continuous approximations (even C^o approximations) in a very straightforward manner. This is because MLPG methods are expected to excel over FEM ([Atluri and Shen, 2005](#)).

The MLPG concept provides a rational basis for constructing a variety of meshless methods, i.e., MLPG- n for $1 \leq n \leq 6$ ([Atluri and Shen, 2005](#)). In MLPG category, MLPG1, MLPG5 and MLPG6 are based on local symmetric weak form (LSWF) whereas MLPG2, MLPG3 and MLPG4 are based on local unsymmetric weak form (LUSWF). A comprehensive summary of all types of MLPG methods is given by [Atluri and Shen \(2002a\)](#). Various researchers have analyzed various numerical problems using MLPG, and found that both

MLPG1 and MLPG5 methods are highly effective and accurate (Gu and Liu, 2001b). Due to the distinct advantages of truly meshless MLPG methods, amongst other they have been efficiently utilized to solve elasto-static problems (Han and Atluri, 2004b; Atluri and Zhu, 2000), energy problems (Wang et al., 2005), elasto-dynamics (Han and Atluri, 2004a), thermoelasticity (Atluri et al., 2000), and plate bending problems (Long and Atluri, 2002), and fluid mechanics using the upwinding concept to simulate the convective-diffusive transport and the incompressible Navier-Stokes equations (Lin and Atluri, 2000, 2001)etc. In the category of MLPG, numerical results show that both MLPG1 and MLPG5 methods are very much effective and accurate. Although there are still many challenges remaining, such as an adequate numerical integration scheme, and its applications in complex fluid flow simulations etc. But a fast, accurate, cheap, and robust MLPG method is still on the way Atluri and Shen (2005). Prior efforts suggest that the meshless methods, in particular, MLPG method, have been successful in obtaining an accurate solutions of solid mechanics problems. Fluid mechanics problems have also been solved within the greater ease and accuracy. However, these methods have not been used in obtaining a solution of complex fluid flow problems.

The present work has aimed to explore the effectiveness and ease of MLPG method for complex fluid flow problems. In particular, non-Newtonian fluid flow simulation using MLPG methods has been the focus of this work. This objective has been achieved by developing a MLPG complex fluid flow solver using C++ programming language. The detailed algorithm has also been developed herein, due to the lack of information for such applications. The solver, first, has been validated and verified by using varieties of standard benchmark problems of CFD. Non-Newtonian fluids are being processed in many chemical and process industries associated with varieties of transport phenomena. Therefore, the approach developed in MLPG complex fluid flow solver will be very useful in analyzing and designing such applications at greater ease.

1.3 Organization of thesis

This dissertation has been divided into 6 chapters as follows. Chapter 2 presents the critical review of literature related to the Meshless Local Petrov Galerkin (MLPG) methods and its applications. In chapter 3, the mathematical formulation and numerical methodology has been discussed. Non-element interpolation tools such as moving least square (MLS) is used to form the shape functions approximations. The suitable weight function $w(\mathbf{x})$ are considered for present work. The numerical integration technique using Gauss points have also been discussed in detailed. Suitable techniques are presented in detail to impose the boundary conditions such as essential boundary condition. The MLPG methods are also comprehended in this chapter. The detailed algorithm has been developed herein. It is used to develop an in-house MLPG complex fluid flow solver using C++ programming language,

The validation of the in-house MLPG solver with various diffusion problems is discussed in chapter 4 using different boundary conditions. Further, the Navier-Stokes flow problems such as convection dominant flow problem, convection-diffusion flow problems have been used to validate the solver. In chapter 5, complex fluid flow solver is validated on 1-D fully developed channel flow problem by using non-Newtonian power-law fluids. Further it has extended the implementation and validation of complex fluid flow solver for 2-D space using Navier-Stokes equations. 2-D channel flow problems by using non-Newtonian power-law fluids have been validated herein. Last chapter concludes the work presented in this dissertation and discussed the challenges with MLPG for complex fluid flow simulations. The limitation of the present dissertation have also been pointed out herein. Finally, the dissertation has been closed up by presenting the future scope of work.



Chapter 2

Literature review

A detailed review of the literature on meshless methods is presented herein this chapter. Particular focus has been on MLPG methods. Prior to the review of meshless methods, a need of them is also highlighted in subsequent section.

2.1 Need of Mesh Free Methods

The following limitations of the traditional CFD methods such as FDM, FEM and FVM are evident in particular book ([Liu, 2003a](#)).

- Meshing of complex geometries is very tedious and expensive in terms of times and manpower, efficient current generation computing facilities may reduces the time and manpower in the meshing process.
- Stress calculations are less accurate.
- Accuracy is lost due to the mesh distortion in large deformation problems.
- Difficult to simulate crack growth with arbitrary and complex paths and face transformations due to the discontinuities that do not coincide with the original nodal lines.

- Re-meshing approaches needed for some moving materials can be difficult to maintain accuracy without introducing an error into simulation.

The meshless methods, as an alternative numerical techniques to eliminate the above drawbacks, have attracted much attention and achieved remarkable applications. *Meshless methods are techniques to form a system of algebraic equation for the complete computational domain without the use of a predefined meshing in the field variable interpolation or approximation.* Over the years, a number of meshless methods have been developed by different researchers. Detailed information on a variety of meshless methods can be found elsewhere (Belytschko et al., 1996b; Nguyen et al., 2008; Fries and Matthies, 2004). A progress of meshfree methods over the last two decades have been thoroughly reviewed elsewhere (Chen et al., 2017a). Subsequent sections present a literature review of the meshless methods.

2.2 Meshless methods

The meshless or meshfree methods are usually categorized as follows: (i) strong form based (ii) weak form based and (iii) combination of weak and strong form based method. Strong form methods are ideal in obtaining the exact solution of system equations, but unfortunately it is quite difficult for practical engineering problems which are usually complex in nature. In comparison to the weak-form methods, strong-form methods have the following attractiveness:

1. Simple algorithms.
2. Computationally efficient.
3. Truly meshless methods for both field variable approximation and integration schemes.

Due to these advantages, strong-form methods have been successfully used in variety of fluid mechanics and other engineering phenomena. They are, however, often unstable and less accurate for problems governed by PDEs subjected to Neumann boundary conditions (BCs), especially with stress (natural) BCs. In the direct meshfree collocation methods, Neumann BCs are implemented using a series of separate equations, which are different

from the governing equations. The error induced from the boundaries, therefore, cannot be efficiently controlled (Zahiri, 2011).

Weak forms, in contrast to strong forms, require a weaker consistency on the approximate function. It provides a number of ways to formulate methods for approximate solutions for complex systems. Formulation based on weak form can usually produce a very stable set of discretized system equations which produces more accurate results. An excellent comparison between local weak and strong forms methods is noted elsewhere (Trobec et al., 2012). There are two major categories for constructing weak form of PDE, namely, variational method and weighted residual methods. The Galerkin and Petrov-Galerkin weak forms are the most widely used approaches for constructing system equations. The weighted residual method, a powerful mathematical tool, is used widely for constructing system equations for many engineering fields problems.

2.2.1 Strong form based meshless method

In the strong form meshfree methods, the strong form of governing equations and boundary conditions is directly discretized at the field nodes using simple collocation techniques (Slater, 1934) to obtain a set of algebraic system of equations. The strong form meshfree method such as smooth particle hydrodynamics (SPH) was used for modeling in astrophysical phenomena (Lucy, 1977; Gingold and Monaghan, 1977) and in solid mechanics (Libersky et al., 1993). SPH is reported to produce inaccurate results near boundaries and tension instability (Sweple et al., 1995). A new meshfree technique, finite point method (FPM) was developed by Onate et al. (1996a,b) to solve the convection-diffusion and fluid flow type problems. It is based on the use of a weighted least-square interpolation procedure together with point collocation for evaluating the approximation integrals. Special attention is given on the stabilization of the convective terms and the Neumann BC which are essential to obtain accurate results. A residual stabilization technique which seems very adequate for FPM has also been proposed and results show that the fixed least-square (FLS) interpolation combined with a point collocation technique is a promising

FPM for the solution of fluid mechanics problems. Furthermore, the meshless collocation method (Kansa, 1990a,b) is a global collocation method that uses all the grids in the problem domain, which leads to fully populated system matrix. A more stable symmetric formulation has also been proposed (Wu, 1992) by using Radial basis function (RBF) to avoid the singularity of moment matrix. Overall, limited research efforts are devoted on the strong form based meshless methods because of less robust than the method based on weak form.

2.2.2 Weak-strong form based meshless method

The key idea of meshless weak-strong (MWS) form method developed by (Liu and Gu, 2002b) is that in establishing the discretized system of equations, the local weak form is used for all boundary nodes (and their nearby nodes) imposed with Neumann BCs and strong form is used for rest of the nodes, including nodes imposed with Dirichlet BCs, in the problem domain. Due to the use of least background cells (quadratic domain) for integration, it is considered as closely ideal meshless method that can provide stable and accurate solutions for many types of PDEs. This method has been successfully used in time-dependent problems of both solid and fluid mechanics (Liu and Gu, 2003b; Liu et al., 2004).

2.2.3 Weak form based meshless method

In weak form based meshfree methods, the governing PDEs with Neumann BCs are first transformed into a set of so called weak form integral equations using various interpolation techniques. These equations are then used to derive a set of algebraic equations through a numerical integration process using sets of background cells that may be constructed globally or locally in the problem domain. Meshfree weak form method based on the global and local weak forms are called *Mfree global weak-form methods*, and *Mfree locally weak-form methods*, respectively.

Many methods have been developed under the category of *Mfree global weak-form methods*. For instance, [Nayroles et al. \(1992\)](#) proposed a diffuse element method (DEM). In this method, only boundary description and mesh of nodes are needed to solve PDE using non-element interpolation technique such as moving least square (MLS) interpolants in curve and surface fitting. [Lancaster and Salkauskas \(1981, 1986\)](#) used all available data points for the particular case of a global approximation. The concept was quite similar to SPH method. The advantages over FEM are that DEM does not rely on a grid, and is more precise in the evaluation of the derivatives of the reconstructed functions. Element free Galerkin (EFG) method is another important method introduced by [Belytschko et al. \(1994\)](#), in which they have included certain terms in the derivatives of the interpolants and used a large number of quadrature points arranged in a cell structure, which were absent in DEM ([Nayroles et al., 1992](#)), for better accuracy. Local weak form is constructed for Galerkin equations by using MLS approximation for the shape function construction based on nodal positions only in the local sub-domain and found that the performance of this method was dependent on the weight function. To preserve the continuity and consistency on the interface elements, a coupling procedure is also developed by combining EFG and FEM to solve the problems of elasticity and crack propagation ([Belytschko et al., 1995](#)). Afterwards, a new procedure has been developed for shape functions in the EFG method so that the functions are continuous in domains with concave corners ([Belytschko et al., 1996a](#)). They found that the major disadvantages of the meshless methods, and EFG method in particular, is the increased computational cost.

Liu and co-researchers ([Liu and Jun, 1995](#); [Liu et al., 1995b](#)) developed a new class of meshless method based on wavelet analysis and reproducing kernel. It is termed as reproducing kernel particle method (RKPM). It allows us to develop a new class of shape function approximation using an integral window transformation. This window function can be translated and diluted around the domain, and thus, replacing the need to define elements and providing refinement ([Onate et al., 1996a](#)).

Similarly, [Duarte and Oden \(1996a,b\)](#) have shown that the moving least squares functions (MLSF) ([Lancaster and Salkauskas, 1981](#)) constitutes a partition of unity and have

developed a new meshless method called $h-p$ cloud method. The basic idea of the method is to multiply a partition of unity, i.e., MLSF, by polynomials or other appropriate class of functions. The resulting functions, called $h-p$ clouds, retain good properties of MLSF, such as high regularity and compactness, and linear combination of these functions can represent polynomials of any degree. This property allows the implementation of p and $h-p$ adaptivity in the $h-p$ clouds with the same striking features of $h-p$ FEM, which is a general form of FEM. Here, $h-p$ indicates that the mesh is refined using a suitable combination of h refinements (sequential reduction of element size) and p refinements (increasing the accuracy of approximation by raising the degree of polynomial).

Subsequently, point interpolation methods (PIM), an another global weak form based meshfree method, was proposed (Liu and Gu, 1999, 2001a) based on the Galerkin weak form. PIM constructs the shape functions based on only a group of arbitrarily distributed nodes in a local domain. Galerkin weak-forms requires a global background cell structure for the evaluation of integrals. The main feature of PIM is Kronecker delta property of their shape functions. Since the moment matrix can be singular in polynomial PIM, a two-stage matrix triangularization algorithm (MTA) is proposed to overcome this problem. It is a novel approach to solve the problem of the singular moment matrix in the construction of PIM shape functions. MTA automatically excludes the nodes and the terms of the polynomial basis used in the formation of the moment matrix (Liu and Gu, 2003a).

The radial point interpolation method (RPIM) (Wang and Liu, 2000, 2002), a typical global weak form based meshfree method, uses radial basis functions (RBF) to overcome the matrix singularity issues. Due to its stability and robustness for arbitrary nodal distributions, RPIM is much widely used than the polynomial PIM. The partition of unity Methods (PUM) and partition of unity FEM (PUFEM) are considered to be essentially identical methods developed by Melenk and Babuska (1996). PUFEM offers an easy way to include analytical information about the problems being solved in FE space. The PUM allows an easier construction of ansatz spaces of any desired regularity, therefore, trial spaces for the use in variational formulations of higher-order differential equations (Babuska and Melenk, 1997). A boundary type meshless method such as boundary node

method (BNM) (Mukherjee and Mukherjee, 1997a) is also a class of meshfree method. It is a combination of MLS approximation scheme and the standard boundary integral equation (BIE) method. Detailed information on a variety of meshless methods based on both strong and weak forms based can be found elsewhere (Belytschko et al., 1996b; Fries and Matthies, 2004; Nguyen et al., 2008).

All the above explored methods requires a background mesh for the numerical integration. Global weak forms (GWF) involve integrals over the global domain including boundary, hence they are not purely meshless methods. Counterpart of global weak form, i.e., the local weak forms (LWF), are defined over local sub domains Ω_s with local boundaries. They do not require any background mesh for either interpolation or integration purpose, hence, they are truly meshfree method (Atluri and Zhu, 1998a,b). The global compatibility is also not required in a local weak form method. These methods include the meshless local Petrov-Galerkin (MLPG) method (Atluri and Zhu, 1998a,b; Atluri and Shen, 2002b), the local boundary integral equation (LBIE) method (Zhu et al., 1998b), the finite spheres method (De and Bathe, 2000, 2001a), and the local radial point interpolation method (LRPIM) (Liu and Gu, 2001a), etc.

Based on the general concept of MLPG method, due to the formulation of weak form in MLPG over all local sub domains Ω_i , use of different types of meshless trial and test functions leads to a large variety of MLPG methods (Atluri and Shen, 2002a,b). They can easily be distinguished by the appropriate selection of the trial and test functions and the integration methods. Based on MLPG concept, the test functions over each local sub domain Ω_s can be selected through various ways (Atluri et al., 1999b; Atluri and Shen, 2005) as follows.

1. weight function in the MLS approximation
2. collocation Diracs Delta function
3. error function in differential equation, using discrete least squares
4. modified fundamental solution to differential equation
5. constant for 2nd order PDE (or linear function for 4th order PDE)
6. identical to trial function

Any other convenient functions may also be chosen as test functions. The above choice of test functions also differs in the type of integrals in weak form, and MPLG methods are characterized by (Atluri and Shen, 2002a, 2005) as shown in Table 2.1. In MLPG-1,

TABLE 2.1: MLPG methods

<i>Method</i>	<i>Test function</i>	<i>Integral type in weak form</i>
MLPG-1	MLS weight function	domain integral
MLPG-2	Dirac delta function	none
MLPG-3	Discrete least-squares	domain integral
MLPG-4	Fundamental solution	singular boundary integral
MLPG-5	Heaviside unit step function	regular boundary integral
MLPG-6	same as the trial function	domain integral

the test function is bell shaped and zero on the local boundary, as long as Ω_i does not intersect with the global boundary of Ω . MLPG-2 yields collocation method and integrals of LWF vanish, i.e., no integration required, and therefore strong form of the PDE is discretized. It is the simplest meshless method. MLPG-3 and MLPG-4 are analogous to the least-squares meshfree method (LSMM) and local boundary integral equation (LBIE), respectively. In MLPG-5, the test function is the characteristic function and thus constant over each local sub domain Ω_i . The volume integral are avoided from the stiffness matrix. MLPG-6 possesses an special case of a Bubnov-Galerkin method due to identical test and trial functions. The resulting method is similar to EFG and DEM but the latter work with GWF instead of LWF. If spheres are used for the sub domains, the method has also been referred to as the method of finite spheres (MFS) (De and Bathe, 2000).

Subsequently, LPIM (Liu and Gu, 2001a) was extended by Liu and Gu (2002a) and proposed a local radial point interpolation method (LRPIM). LPIM is, a truly meshless method, based on non-element interpolation and non-mesh integration. It construct polynomial interpolation function with delta function property using a group of arbitrarily distributed points. Due to property, essential boundary conditions can easily be implemented. Inappropriate choice of polynomial basis results in a badly conditioned matrix, which may be even invertible (Liu and Gu, 2001b). For alleviating this problem, LRPIM constructs shape functions using radial function basis is proposed. The shape functions so formulated possess delta function property. In this technique, the interpolation using radial function basis is stable and flexible. Similarly, Zhu et al. (1998b) developed a new class

of meshfree method, namely, local boundary integral equations (LBIE), which combines the features of three methods such as Galerkin finite element method (GFEM), boundary element method (BEM) and element free Galerkin methods (EFGM). It is illustrated to solve boundary value problems (BVP). Subsequently, it is included under MLPG category and termed as MLPG-6, which is based on the local symmetric weak form (LSWF) wherein test and trial functions are defined from the same space. Boundary integrals with strongly singular kernels require special attention in implementations of LBIE, for known boundary densities in case of MLS approximation (Sladeka et al., 2000). LBIE possesses a tremendous potential for solving nonlinear problems and/or problems with discontinuities (Zhu et al., 1998a). It is a promising method for solving BVP in solid mechanics (Atluri et al., 2000; Zhu, 2010).

Further, Atluri and Zhu (1998a) illustrated a MLPG method to solve the linear problems using Laplace and Poisson equations on patch test and potential flow problems. Conventional moving least squares (MLS) scheme has been used in meshless interpolation. They also solved nonlinear problems to ensure its truly meshless characteristics. They observed that both linear and quadratic bases $P(x)$, as well as spline and Gaussian weight functions $w_i(x)$ in the approximation/trial function yielded quite accurate numerical results. In most cases Gaussian weights functions with the quadratic basis produced better results, but quadratic basis may increase the computational cost (Atluri and Zhu, 1998b).

This MLPG method is further extended (Atluri et al., 1999a) to illustrate 4th order boundary problems for thin beams. A generalized moving least squares (GMLS) interpolation scheme was proposed by introduction of an independent variable in MLS scheme. A new numerical algorithm was also proposed for non-element local integration schemes such as MLS, PU and Shepard interpolations (Atluri et al., 1999a,b). They clearly distinguished between symmetric and unsymmetric matrices formed due to the varying sizes of test and trial domains of computing point I and influence point J to form the stiffness matrix K_{IJ} . They derived the transformation technique between the fictitious and actual nodal values (\hat{u}_I and \tilde{u}_I) for direct imposition of the essential BCs (Cai and Zhu, 2004). A new method to integrate within each sub-domain or over intersections of sub-domains has also been

presented. MLPG is successfully used to solve the applied mechanics problems such as 2-D elasto-statics, cantilever beam, plate with a hole, long beam, etc (Atluri et al., 1999b).

Interrelation of the various meshless approaches has been briefly presented by Atluri and Shen (2005). They summarized a variety of typical local interpolation schemes and MLS approximation. It was noted that the basis functions can be other than monomial. Local (truncated) least square (LSQ) scheme from the MLS have been pointed out by using heavy-side weight functions. The usual “element assembly” process, as in the conventional FEM is needed but the global stiffness matrix formation is not required in MLPG Atluri and Shen (2005).

In spite of the great success of the classical numerical approaches in both solid and fluid mechanics problems, meshfree methods is still a growing interest in development of new advanced numerical method due to their high adaptivity and low cost to prepare input data for numerical analysis. MLPG is effectively and efficiently applied in heat transfer problems with great extent.

2.3 MLPG application in diffusion problems

The variety of heat transfer problems have been solved by using MLPG methods such as heat conduction problem in an an-isotropic medium (Sladeka et al., 2004; Mazzia et al., 2014), 3-D transient heat conduction in a functionally graded thick plate (Qian and Batra, 2005). Stationary and transient heat conduction inverse problems in 2-D and 3-D axisymmetric bodies wherein singular value decomposition (SVD) is applied to solve ill-conditioned linear system of algebraic equations obtained from MLS approximation of the local integral equations (LIE) (Sladeka et al., 2006), local weak-forms for steady-state heat conduction problems (Hong and Quan, 2008), transient heat conduction problems (Sladeka et al., 2008; Slater, 2008; Chen and Liew, 2011; Slater, 2008), nonlinear heat conduction problems (Thakur et al., 2009). The diffusion equation with an integral condition in which a new approach based on MLPG and collocation methods has been used to treat the

parabolic PDEs with non-classical BCs ([Abbasbandy and Shirzadi, 2010](#)), steady-state heat conduction in heterogeneous materials ([Ahmadi et al., 2010](#)) and 3-D heat transfer analysis ([Tian and Rao, 2012](#)). The types of non-element interpolation techniques used in MLPG method and variety of linear and non-linear problems have been explained in the book ([Liu and Gu, 2005](#)).

2.4 MLPG application in convection-diffusion problems

Numerical simulation of convection-diffusion such as Navier-Stokes equation using MLPG in the fluid flow phenomena has been explored by numerous researchers. To avoid oscillatory behavior and numerical diffusion encountered in convection dominant flows, upwind techniques have been widely applied in the conventional CFD techniques viz FEM, FDM, FVM ([Zienkiewicz and Taylor, 2000](#)). Upwinding concept is used to stabilize the convection-dominated flow problems. [Hughes and Brooks \(1979\)](#) introduced the ‘streamline upwind (SU) method’, where the artificial diffusion operator is constructed to act only in the flow direction, and thereby, a-priori elimination of the possibility of any crosswind diffusion. It was observed that better solutions for the crosswind problem can be obtained by using the streamline upwind method, but several deficiencies remained [Brooks and Hughes \(1982\)](#). Most popular among such methods is the streamline upwind Petrov-Galerkin method (SUPG) [Brooks and Hughes \(1982\)](#) which consistently introduces an additional stability term in the upwind direction. This method has better stability and accuracy properties than the standard Galerkin formulation for convection-dominated flows.

In the Mfree approach first upwind scheme introduced by [Gingold and Monaghan \(1977\)](#) using SPH method which has been widely applied in incompressible flow and free surface flow problems. Onate and co-researchers ([Onate et al., 1996a,b](#)) solved flow problems using FPM, in conjunction with the finite increment calculus (FIC) method to handle the nonlinear convection term. Sadat and co-authors ([Sadat and Prax, 1996](#); [Sadat and](#)

Couturier, 2000) employed DEM to solve fluid flow problems. Wu and Liu (2003) applied the local radial point interpolation method (LRPIM) to solve 2-D natural convection heat transfer in a square cavity and in a concentric circular annulus for Rayleigh numbers up to 10^5 and 10^4 , respectively.

Subsequently, Fries and Matthies (2006b,a) developed the coupled method of EFG and FEM to compute incompressible fluid flow. Gunther et al. (2000) carried out multi-scale meshfree parallel computations of viscous compressible flows using RKPM method. Least-squares meshfree method (LSMFM) based on the first-order velocity-pressure-vorticity formulation for 2-D steady state incompressible viscous flow has been presented by Zhang et al. (2005). For solving the unsteady convection-diffusion equation, Chinchapatnam et al. (2006) used unsymmetric and symmetric meshless collocation techniques with radial basis functions. They suggested that the symmetric collocation is marginally better than the unsymmetric approach. Both collocation techniques required a very dense set of collocation points in order to achieve accurate results for high Peclet numbers.

The choice of the related parameters is still not so straight forward, because the MLPG method is based on a local weak form over a local sub-domain, there is a very convenient way to construct upwind scheme. The pioneering work proposed two kinds of upwind schemes of MLPG method to solve convection-diffusion (Lin and Atluri, 2000) and incompressible flow problems (Lin and Atluri, 2001), namely “MLPG Upwinding Scheme I (MLPG-US1)” and “MLPG Upwinding Scheme II (MLPG-US2)”. These techniques were applied upwinding in the streamline direction with skewing of the test function opposite to the streamline direction at very high Peclet numbers, $Pe = 1, 10, 100$ and 1000 . MLPG-US2 produced very good solutions when Pe is very large. In the MLPG2, larger size of support domain yields more accurate solutions and smaller size of support produces more spurious crosswind diffusion effect (Lin and Atluri, 2000). Further, they solved the incompressible Navier-Stokes (N-S) equations by MLPG method with addition of a “perturbation” term based on residual forms of the Euler-Lagrange equations in modified LWF of continuity equation (Lin and Atluri, 2001).

Subsequently, [Mohammadi \(2008\)](#) developed a new upwind scheme based on the meshless FVM to calculate incompressible flow problems with a vorticity-stream function formulation of NS equations. The heaviside step function was used as test function. Due to its simple implementation, it can be very conveniently extended in different meshless methods as an upwind scheme. MLPG method with some modifications is adopted by [Wu et al. \(2005\)](#) to simulate steady incompressible flow problems for a scattered nodal distribution. They solved vorticity-stream function equation to simulate natural convection in concentric annuli with different geometries at high Reynolds number ($10^4 < Re < 10^6$). [Wu et al. \(2010\)](#) further extended the MLPG method to solve the incompressible fluid flow problems. To overcome the oscillations behavior, the streamline upwind Petrov-Galerkin (SUPG) method was applied in convection-dominated flows. The pressure-stabilizing Petrov-Galerkin (PSPG) method is applied to satisfy the Babuka-Brezzi condition. A stabilization parameter τ ($\tau_{SUPG} = \tau_{PSPG}$) is used. For lid-driven square cavity flow and natural convection at $Re = 100, 400, 1000, 4000$ and 5000 , they found that upwind scheme suggested by [Lin and Atluri \(2000, 2001\)](#) could not obtain convergent solutions to at the high Re . [Wu et al. \(2012\)](#) used SUPG scheme to eliminate over- and under-shoots produced by the convection term, they analyzed the convection dominated problems. SUPG method is strongly dependent on the stability parameter, although it is more stable than MLPG-US1 and MLPG-US2.

Lid-driven cavity flow, backward-facing step flow and transient flow past a circular cylinder governed by 2-D Navier-Stokes equations in terms of the primitive variables at various Reynolds numbers (Re) has been solved by MLPG method ([Najafi et al., 2012](#); [Najafi and Enjilela, 2014](#)). Further, MLPG primitive variable-based method using CBS scheme was used to solve the laminar fluid flow and natural, forced and mixed convection heat transfer at higher Rayleigh, Reynolds, Peclet and Grashof numbers. Natural convection within a square cavity, forced convection flow over a tube bundle, and mixed convection within a lid-driven square cavity were also solved by their proposed method ([Najafi et al., 2013](#)).

Further, [Sataprahma and Luadsong \(2014\)](#) used MLPG (LSWF1) to solve unsteady incompressible fluid flow problem governed by Navier-Stokes equations at $Re = 10^2$, and

concluded that a classical Gaussian weight function of order 2 gives the most accurate results. Afterwards, problems such as flow between parallel plates, lid-driven cavity flow, Couette flow between two eccentric cylinders at high Peclet (Pe) and Reynolds (Re) numbers using Navier-Stokes equations were solved by using a stabilized two-step Taylor-characteristic-based MLPG (2S-TCB-MLPG) method (Enjilela and Arefmanesh, 2015). Recently, a two-level variational multi-scale MLPG (VMS-MLPG) method is proposed by Zheng et al. (2009). They obtained the stable and accurate solutions of convection-diffusion problems with large Peclet numbers. They also observed that the convection term may cause oscillation solutions at $Pe = 10^6$ and 10^8 (Chen et al., 2017b).

The present work has explored the effectiveness and ease of MLPG method for complex fluid flow problems. In particular, non-Newtonian fluid flow simulation using MLPG methods has been the focus of this work. This objective has been achieved by developing a MLPG based Complex fluid flow solver in C++ programming language. Due to the lack of information for such applications. The solver, first, has been validated and verified by using varieties of standard benchmark problems of CFD using MLPG method.

2.5 Conventional approaches in analysis of complex fluid flow

Non-Newtonian fluids are being processed in many chemical and process industries associated with varieties of transport phenomena. Voluminous literature is available on the non-Newtonian rheology and its applications in allied industries, e.g., (Chhabra and Richardson, 2008), etc. Conventional numerical and experimental approaches has been used on the novel work in the field of complex fluid rheology such as laminar flow of power law fluids in concentric annuli (Gupta and Chhabra, 1996), co-current flow of shear thinning liquids and air in horizontal and vertical pipes (Chhabra et al., 1982), prediction of flow pattern of non-Newtonian liquid in horizontal pipes (Chhabra and Richardson, 1984), particles velocity calculation (Chhabra, 1995), complex fluid flow in curve ducts

(Jayanti et al., 1990; Jayanti and Hewitt, 1992) and power law rheology in convection diffusion problems (Paliwal et al., 2003; Verma and Eswaran, 1996, 1997).

Recently, a numerical approach has been developed (Tian et al., 2014a; Tian, 2014) for complex non-Newtonian fluid flow and heat transfer phenomena involving moving boundaries and complex geometries, deforming-spatial-domain/stabilized space-time (DSD/SST) method. They observed that the flow and heat transfer characteristics are quite different if the moving boundaries are taken into account. More recently, a flow solver based on the fractional step implicit immersed boundary (IB) method is used to simulate the complex flows involving moving boundaries and complex geometries are having body force caused by the immersed body (Wang and Tian, 2017). The lid-driven cavity flow of a purely-viscous non-Newtonian fluid obeying Carreau-Yasuda rheological model by using the PIM meshfree method combined with characteristic-based split (CBS) algorithm has been numerically studied by Shamekhi and Sadeghy (2009). Their Results, in terms of velocity and pressure profiles at Reynolds numbers as high as 10^3 , revealed the strong effect of the shear-thinning behavior of a fluid on its flow kinematics within the cavity. They observed that results obtained using meshless methods are in close agreement with the conventional CFD approaches.

Therefore, having the added advantages of meshless methods over conventional CFD methods, MLPG based complex fluid flow solver will be very useful in analyzing and designing such applications at greater ease.

2.6 Summary of literature review

Based on the aforementioned discussion of the available literature, following remarks can be drawn:

1. To overcome the drawbacks of the classical methods (FDM, FEM and FVM), MLPG is the new numerical approach in the computational methods, and widely used in the different engineering problems.

2. Amongst the various meshfree approaches available in the literature, MLPG follow the truly meshless characteristics and gives liberty to avoid the mesh generation.
3. Due to the truly meshless characteristics, MLPG methods are widely adopted as an efficient computational technique to solve applied mechanics problems, heat transfer phenomena, fluid mechanics using the upwinding concept, to simulate the convective-diffusive transport and the incompressible Navier-Stokes equations, etc.
4. MLPG produces more accurate results due the truly meshless nature. Re-meshing approaches needed for some moving materials in the predefined numerical techniques and difficult to maintain accuracy without introducing an error into simulation.
5. MLPG is highly suitable for complex geometries than the classical methods (FDM, FEM and FVM) or
6. While having a number of advantages of MLPG methods, none of the efforts have been devoted to its application in complex fluid flows simulations.
7. Due to efficient and accurate PDE solver, MLPG method will be very useful to simulate the complex fluid rehology which is widely used in chemical and process industries.
8. While the meshless methods are advantageous in many aspect, there are certain notable problems due which, probably, these methods have grown up slowly. For instance, complexities in defining the shape function and their derivative at the boundary surfaces, the numerical stability issues at higher Reynolds number, etc.

2.7 Objectives

Based on the above analysis of the existing literature on meshless methods and their applications, the present work aims to explore the effectiveness and ease of MLPG method for complex fluid flow problems. This thesis particularly focuses on non-Newtonian fluid

flow simulation using MLPG methods. Based on the review of available MLPG and complex fluid flow literature, the following objectives are set up for the present research work.

1. The development of the general purpose CFD solved based on of the meshfree methods, namely, meshless local Petrov-Galerkin method (MLPG) in C++ programming language.
2. Validation and application of the solver for basic Newtonian fluid flow and heat transfer problems such as 1-D and 2-D space diffusion problems, convection problems, convection-diffusion problems and incompressible flow problems based on Navier-Stokes equations, etc.
3. Extension of developed MLPG solver to include non-Newtonian fluid rheology.
4. Validation and application of MLPG code to solve the non-Newtonian power-law fluid flow problems in 1-D and 2-D space.

The next chapter discusses the mathematical formulations for MLPG approach, suitable numerical integration technique using the standard PDEs to fulfill the objectives setup in the present work.



Chapter 3

Mathematical formulation and numerical methodology

This chapter presents the general idea about MLPG method, formation of local sub-domain, technique about imposition of essential and natural BCs with appropriate assumptions. Moving least square (MLS) formulations to construct the shape function approximation for data interpolation. A brief description about weight functions calculation at the computing points and Gauss point integration technique of quadrature domain is also presented herein.

3.1 Meshless local Petrov Galerkin (MLPG) method

The generalized form of the present numerical method such as meshless local Petrov Galerkin method (MLPG) is developed by [Atluri and Zhu \(1998a,b\)](#), which is one of the truly ‘mesh-free’ method, since it does not require any background cells for interpolation and integration purpose. Flexibility present by choosing between test and trial functions, the original formulation of the MLPG has subsequently been divided into various forms, namely, MLPG-1, MLPG-2, MLPG-3, MLPG-4, MLPG-5, and MLPG-6. Amongst them,

MLPG-1 is the original version. Both MLPG-1 and MLPG-5 seem to be the most promising formulations (Atluri and Shen, 2005). See subsection 2.2.3 and Table 2.1 for more details.

3.2 Concept of support and influence domains

The concept of the support domain/local sub-domain is used to interpolate the value at any point (or node) within the computational domain. A local domain includes many nodes inside it. These points, except the point of interest, are called as influencing points and the small local domain is often called as support domain/local sub-domain. Such influencing nodes come inside the support domain of a point x to be used to support or approximate the function value at x . It can have different shapes and sizes for different points of interest x . The support and problem domains are usually mentioned as Ω_I and Ω , respectively. Depicted in Figure 3.1 are the various types of influence and support domains. The most often used shapes, however, are circular or rectangular in 2-D space. The concept of support domain is always used to select the nodes for constructing the shape functions.

In the present work, a circular domain is used because (i) circle has no bias on direction and (ii) very easy to formulate the weight function, as depicted in Figure 3.1. The approximate

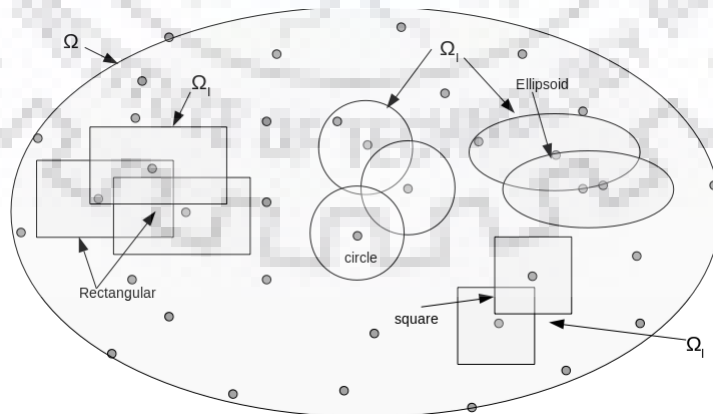


FIGURE 3.1: Types of support domain used

function calculated for any computing point x is often called as test function ϕ_{te}^I . In meshless methods, both support and influence domains can be clearly distinguished. A

influence domain is defined as a domain of neighboring nodes that a node exerts an influence upon (Liu, 2003a). Use of an influence domain is an alternative way to select nodes for interpolation. It works well for domains with highly non-regularly distributed nodes, but for the uniformly distributed nodes the domains are same having equal radius (r_i).

Figure 3.2 shows that the domains for nodes 1 to 5, when approximating the field variable at point x . Each domain is having different radius as r_1, r_2, \dots, r_5 , respectively. Nodes 2, 3, and 5 are within the support domain of computing point x . These circles represent the influence domain for the corresponding nodes. Note that the domains of points 1 and 4 are not the influence domain of point x . It can also be seen that node 4 is closer, in comparison

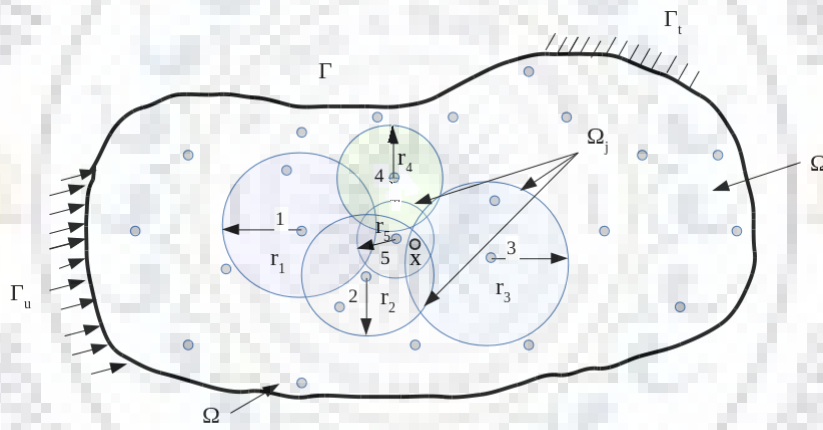


FIGURE 3.2: Influence domain of a point x

to node 3, to the point x but it does not have any influence on point x . These influence domains are usually represented as Ω_J and approximate function for these domains are called as trial function ϕ_{tr}^I . Here, I and J refer to the computing and influencing points, respectively.

3.3 Determination of dimension of a support domain

The accuracy of interpolation function depends on the nodes inside the support domain of computing point (which is often a quadrature point x or the center of integration cell). Therefore, a suitable choice of support domain is must to ensure a proper area of coverage

having sufficient number of nodes for interpolation. The dimension of the support domain (e.g., radius ds) for a point x is determined by

$$ds = \alpha \times dc \quad (3.1)$$

where α and dc are the dimensionless parameter of the support domain and a characteristic length relating to the nodal spacing near the point x . In general, dc can be defined as an average nodal spacing in the support domain of point x . It is simply the distance between two neighboring points for uniformly distributed points. Hence, the size of support domain may differ for each non-uniformly distributed computing point. The importance of α is in determining the size of local support domain such that the maximum number of nodes are selected for accurate interpolation at interpolation point with respect to nodal distance. Scaling parameter (α) are chosen in a way that yields non-singular matrix at every point inside the domain. Generally, good results are obtained for $2 \leq \alpha \leq 3$ (Liu, 2003a). However, it need to be predetermined by the analyst, usually, through numerical experimentation for the same class of problems having known solutions. The actual number of nodes (n) can be determined by counting all the nodes in the support domain.

An average nodal spacing for randomly distributed nodes can be determined as follows:

$$dc = \begin{cases} \frac{D_s}{n_{ds} - 1} & \text{for 1-D space} \\ \frac{\sqrt{A_{sd}}}{\sqrt{n_{A_{sd}} - 1}} & \text{for 2-D space} \\ \frac{\sqrt[3]{V_{sd}}}{\sqrt[3]{n_{V_{sd}} - 1}} & \text{for 3-D space} \end{cases} \quad (3.2)$$

where D_s and n_{ds} are the estimated ds and the number of nodes covered by a known domain of dimension D_s . A_{sd} and $n_{A_{sd}}$ are the estimated area covered by the support domain of dimension ds and the number of nodes covered by the estimated domain of area A_{sd} . V_{sd} and $n_{V_{sd}}$ are the estimated volume covered by the support domain of dimension ds and the number of nodes covered by the support domain of volume V_{sd} . An estimate of ds need not to be very accurate but a reasonably good. The dimension ds of the local sub-domain Ω_s for a interpolating point x in the physical domain Ω with non-uniformly

distributed nodes can easily be determined (Liu, 2003a), for instance by using Eq. (3.2) for 1-D space, as follows.

1. Estimate of ds with the help of dc gives Ds .
2. Count total number of nodes covered by Ds .
3. Calculate dc by using Eq. (3.2).
4. Finally, determine size of support domain (ds) by using Eq. (3.1) for optimally selected α .

Similar procedure can be used to determine ds in 2- and 3-D spaces by using Eqs. (3.3) and (3.4).

3.4 MLPG formulations

Prior to the constructions of MLPG methodology to calculate the approximate function to local interpolation, the following three basic formulations/steps are required.

- Basis function
- Weight function
- Shape function approximation using non-element interpolation technique such as MLS, PIM, RBS, etc.

MLS is categorized as a method of series representation of functions. It is popular technique due to the following two main features:

- Approximated field function in MLS is continuous and smooth in problem domain
- It can produce an approximation having desired order of consistency

MLS technique is, therefore, widely used as an alternative method for constructing functions approximation than any other interpolation technique. It is however very complex in nature (Lancaster and Salkauskas, 1986; Cleveland, 1993).

3.4.1 Basis function approximation

A predefined interpolation scheme, i.e., use of polynomials as basis function in the interpolation, has been widely used in FEM. In Mfree methods, basis functions play crucial role in defining the accuracy of numerical computations because shape functions approximation depend on them.

In MLPG formulations, the linear and quadratic basis functions for 1- and 2-D spaces are chosen as follow:

$$P^T(x) = \begin{cases} \{1 & x\} & p=1 \text{ for 1-D space} & (3.5) \\ \{1 & x & y\} & p=1 \text{ for 2-D space} & (3.6) \\ \{1 & x & x^2\} & p=2 \text{ for 1-D space} & (3.7) \\ \{1 & x & y & x^2 & xy & y^2\} & p=2 \text{ for 2-D space} & (3.8) \end{cases}$$

Here p is the order of polynomial. In general, the choice of basis function can be described as Pascal triangle for multi-dimensional space as shown in Figure 3.3. The general form of

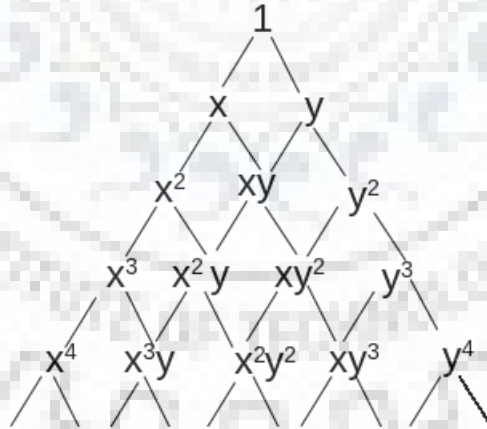


FIGURE 3.3: Pascal triangle for general selection of basis function (Liu and Gu, 2005).

polynomial basis function of order p can be written as follows.

$$P^T(x) = \begin{cases} \{1 & x & x^2 & \cdots & x^{p-1} & x^p\} & \text{for 1-D space} & (3.9) \\ \{1 & x & y & x^2 & xy & y^2 & \cdots & x^p & y^p\} & \text{for 2-D space} & (3.10) \end{cases}$$

3.4.2 Choice of weight functions

The weight functions are used to perform a sum, integral, or average in order to give some elements more weightage or influence on the result than other elements in the same set. They can be employed in both discrete and continuous manner. The performance of the MLS approximation is strongly dependent on weight functions. In general, it should be nonzero over a small neighborhood of a node x_I . It must also have compact support called as the support of node I . It is desirable that $w(x - x_I)$ be smooth. If $w(x - x_I)$ is C^1 continuous then the shape function $\phi_I(x)$ will also C^1 continuous for a polynomial basis (Lancaster and Salkauskas, 1981, 1986). Further, $w(x - x_I)$ should be large for smaller values of $(x_I - x)$ and relatively small for the larger distant x_I . The weight function

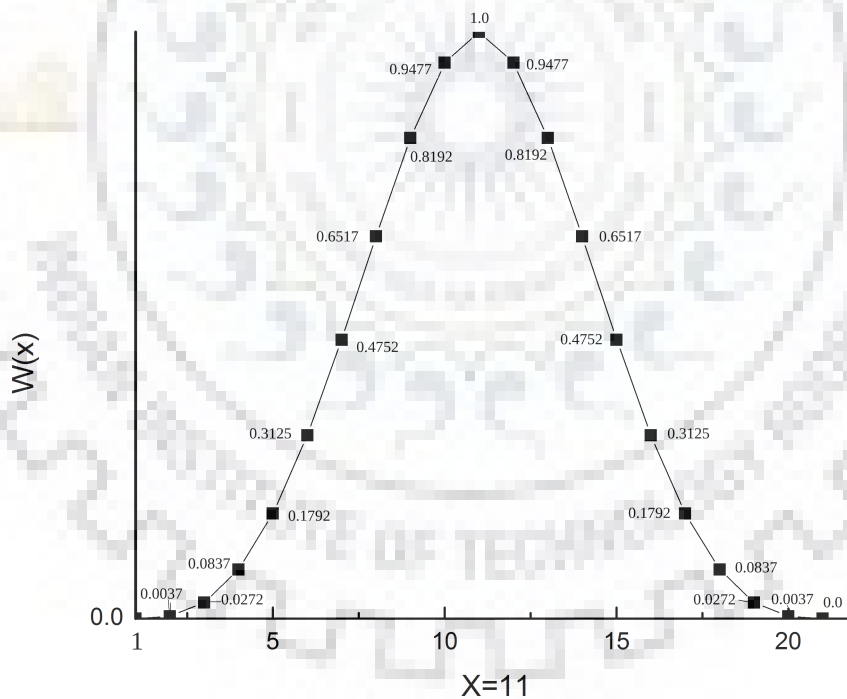


FIGURE 3.4: Weightage of influencing points in subdomain.

$w(x - x_I)$, hence, affects the resulting approximation function in MLPG and other meshless formulations. The major properties of weight function $w(x - x_I)$ are summarized (Liu and Gu, 2005) as follows.

- It should be finite (> 0) within support domain, Ω_s .
- It should be zero ($= 0$) outside support domain.
- It should monotonically decreases from the point of interest (x), and must have compact support, i.e. the 1st and 2nd derivative are all completely zero at the boundary of the support domain where $r_i = 1$.
- It must be sufficiently smooth, especially on the boundary of Ω_s .

The compact support is most important characteristic of the selection of weight function. The above properties are completely fulfilled in Figure 3.4, which is constructed using quadratic weight function for point of interest as $x = 11$ and $x_I = 1, 2, 3, \dots, 21$. A symbolic representation of weight function, which depends only on the distance between two points ($d = x - x_I$), can be given as follows.

$$w(x - x_I) = w_I(d) \quad (3.11)$$

The most commonly used weight functions in literature (Belytschko et al., 1996b) can be written as functions of normalized radius/distance ($r_d = |d|/d_s$) as follow. Irrespective of the type of weight function, $w_I(d) = 0$ for $r_d \geq 1$.

1. Cubic spline weight function

$$w_I(d) = \begin{cases} \frac{2}{3} - 4r_d^2 + 4r_d^3 & \text{for } r_d \leq \frac{1}{2} \\ \frac{4}{3} - 4r_d + 4r_d^2 - \frac{4}{3}r_d^3 & \text{for } \frac{1}{2} \leq r_d \leq 1 \end{cases} \quad (3.12)$$

2. Quartic spline weight function

$$w_I(d) = 1 - 6r_d^2 + 8r_d^3 - 3r_d^4 \quad \text{for } r_d \leq 1 \quad (3.13)$$

3. The exponential weight function (search)

$$w_I(d) = \exp\left(\frac{-r_d}{\alpha_o}\right)^2 \quad \text{for } r_d \leq 1 \quad (3.14)$$

where α_o is constant parameter. Generally, $\alpha_o = 0.3$ (Liu, 2003a).

Singh (2004) has extended the definition of weight function for multi-dimensional systems by defining the normalized distance (r_d) as follows.

$$r_d = \frac{\|x - x_I\|}{ds} \quad \text{where } x = \{x, y, z\} \quad \text{and } ds \text{ (Eq. 3.1)} \quad (3.15)$$

The weight function at any given point in multi-dimensional systems is then calculated as follows.

$$w_I(d) = \begin{cases} w_x & \text{for 1-D space} \\ w_x \times w_y & \text{for 2-D space} \\ w_x \times w_y \times w_z & \text{for 3-D space} \end{cases} \quad (3.16)$$

The formulation of shape functions (see Section 3.4.3.1) results in matrices **A** and **B**. To compute the spatial derivatives of these matrices, it is necessary to compute the spatial derivative of the weight function. The derivative of weight function for three dimensional space is represented as below.

$$w'_{I,x}(d) = w'(r_x) \cdot w(r_y) \cdot w(r_z) \quad (3.17)$$

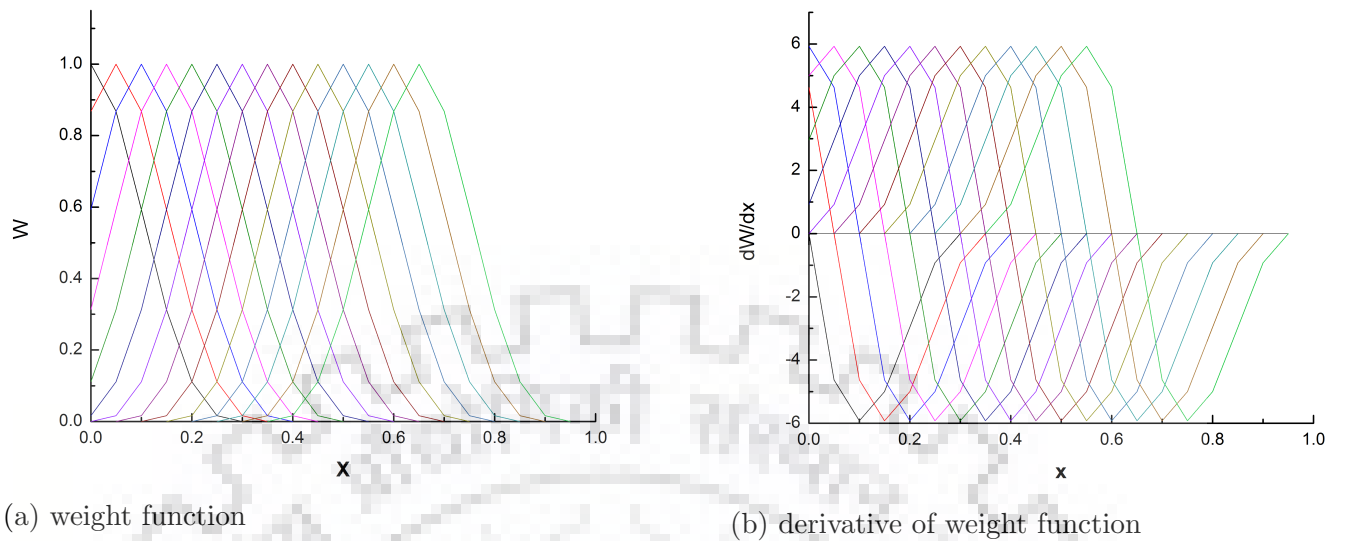
$$w'_{I,y}(d) = w(r_x) \cdot w'(r_y) \cdot w(r_z) \quad (3.18)$$

$$w'_{I,z}(d) = w(r_x) \cdot w(r_y) \cdot w'(r_z) \quad (3.19)$$

The x -derivative of cubic spline weight function (Dolbow and Belytschko, 1998) is evaluated as follows.

$$w'(r_x) = \frac{dw_I}{dx} = \frac{dw_I}{dr_x} \frac{dr_x}{dx} = \begin{cases} (-8r_x + 12r_x^2) \text{sign}(x - x_I) & \text{for } r_x \leq \frac{1}{2} \\ (-4 + 8r_x - 4r_x^2) \text{sign}(x - x_I) & \text{for } \frac{1}{2} < r_x \leq 1 \\ 0 & \text{for } r_x > 1 \end{cases} \quad (3.20)$$

Figure 3.5 illustrates an example of the quadratic weight function and its first derivatives in 1-D space Ω having 21 uniformly distributed nodes. The general idea of three types of

FIGURE 3.5: Weight function and its 1st derivatives

weight functions and their first derivatives is illustrated (Liu and Gu, 2005) in Figure 3.6. MLS approximation for the trial function at every sample point must ensure to have

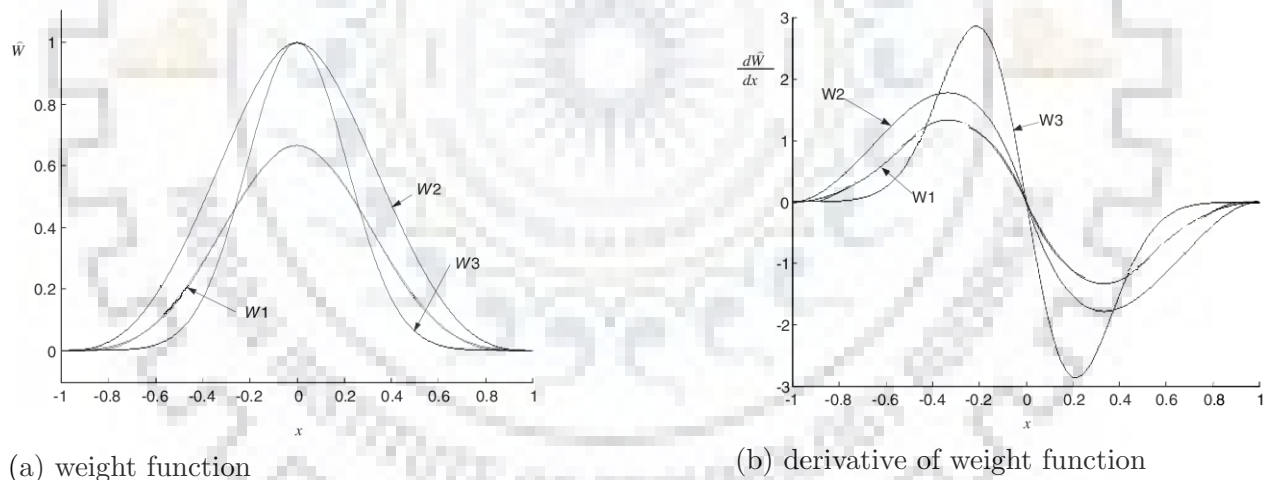


FIGURE 3.6: Weight function and their derivatives (W1: cubic spline; W2: quartic spline; W3: exponential) (Liu and Gu, 2005).

the non-singular matrix A . It can be achieved by suitable selection of weight function w_i associated with node i in such a way that the size of support domain ds for w_i should be optimally large enough having sufficient nodes to cover the domain of the MLS approximation. A very small ds may yield relatively large numerical errors to construct the system matrix. On the other hand, ds should be also small enough to maintain the local character for MLS approximation. As discussed earlier, the weight function affects

the resulting MLS approximation, if weight functions (w) are continuous then the shape functions (ϕ) are also continuous.

3.4.3 Moving least square (MLS) approximation

MLS approximation was developed for the data fitting and surface construction by [Lancaster and Salkauskas \(1986, 1981\)](#) and [Cleveland \(1993\)](#). An excellent description of MLS can be found elsewhere ([Lancaster and Salkauskas, 1981](#)). In MLS, the dependence of coefficient a on x makes continuous movement of weighted least squares approximation. Due to which, weighted least square (WLS) interpolation is termed as moving least square (MLS). Hence, MLS shape functions remain continuous in the entire global domain, as long as the weight functions are chosen properly ([Liu and Gu, 2005](#)). MLS approximation is widely used in meshfree methods for construction of shape functions for local sub domain. [Nayroles et al. \(1992\)](#) first constructed shape functions using MLS approximation for their diffuse element method (DEM) for mechanics problems. MLS approximation is popular due to two major features (see section 3.4). The procedure of constructing shape functions for MFree methods using MLS approximation is detailed in subsequent section.

3.4.3.1 Formulation of MLS shape functions

Meshless method, in general, is required to preserve the local character of the numerical implementation by using a local interpolation or approximation to represent the trial function with actual (or fictitious) values of unknown variables at randomly located nodes. Generalization of the meshless formulation needs a relatively direct local interpolation or approximation scheme with reasonably high accuracy and with an easy extension to n -dimensional problems. MLS approximation can be considered one of such schemes ([Atluri et al., 1999b](#)).

Let approximation $u^h(x)$ for any function $u(x)$ is posed as a m^{th} order polynomial having arbitrary coefficients. The order of polynomial is defined as the order of basis. In 1-D

space, a linear or quadratic basis $u^h(x)$ can be written as below.

$$u^h(x) = \begin{cases} a_0(x) + a_1(x)x & \text{(linear basis)} \\ a_0(x) + a_1(x)x + a_2(x)x^2 & \text{(quadratic basis)} \end{cases} \quad (3.21)$$

where unknown coefficients $a_j(x)$ vary with x . These approximations are called as moving least square (MLS) interpolants in curve and surface fitting ([Lancaster and Salkauskas, 1981](#)).

Let an unknown scalar function of defined field variable $u(x)$ in problem domain Ω , the neighborhood of point x is located in $\{x_I\}$ where Ω is divided in $I = 1, 2, 3 \dots n$ nodes. The MLS approximation of an unknown variable $u^h(x)$ of u can be defined by

$$u^h(\mathbf{x}) = \sum_{j=1}^m p_j(\mathbf{x})a_j(\mathbf{x}) = \mathbf{P}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \quad \forall x \in \Omega_s \quad (3.23)$$

where the vector of basis function, $\mathbf{P}(\mathbf{x})$, is chosen according to the problem space, which consists lowest order of monomials to ensure minimum completeness, and m is the number of terms in the monomials. The vector of nodal coefficients, $\mathbf{a}(\mathbf{x})$, is given by

$$\mathbf{a}^T(\mathbf{x}) = \{a_1(x) \quad a_2(x) \quad a_3(x) \quad \dots \quad a_m(x)\} \quad (3.24)$$

It can be obtained by minimizing the following weighted discrete L_2 norm ([Marosevic, 1996](#)).

$$J = \sum_{I=1}^n w(x - x_I)[u^h(\mathbf{x}, x_I) - u_I]^2 = \sum_{I=1}^n w(x - x_I)[\mathbf{P}^T(x_I)\mathbf{a}(\mathbf{x}) - u_I]^2 \quad (3.25)$$

where n is the number of nodes in the test domain of x for which $w(x - x_I) \neq 0$, and u_I is the nodal value of u at $x = x_I$. The above functional equation, i.e., a weighted residual, is constructed using the approximated values and nodal parameters of unknown field function. Since the nodes (n) used in MLS approximation is usually much larger than the number of basis function (m), the approximated function (u^h) does not pass through the nodal positions, as shown in [Figure 3.7](#).

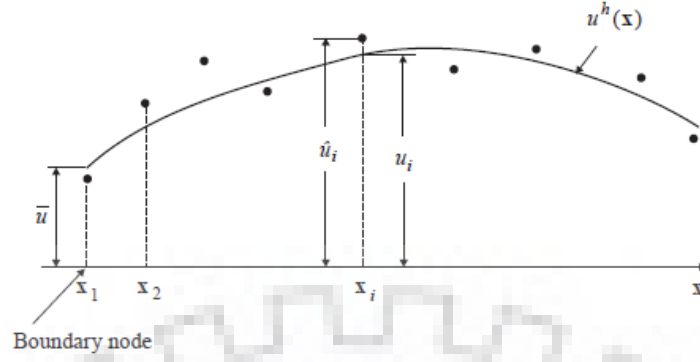


FIGURE 3.7: The appropriate function $u^h(x)$ and the nodal parameter u_i in the MLS approximation (Atluri and Zhu, 1998a).

Eq. (3.25) can be rewritten in the following form

$$J = (\mathbf{P} \mathbf{a} - \mathbf{u})^T \mathbf{w}(x) (\mathbf{P} \mathbf{a} - \mathbf{u}) \quad (3.26)$$

here $\mathbf{u}^T = \{u_1, u_2, u_3, \dots, u_n\}$ (3.27)

$$\mathbf{P} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{bmatrix} \quad \text{and} \quad (3.28)$$

$$\mathbf{w}(x) = \begin{bmatrix} w(x - x_1) & 0 & \cdots & 0 \\ 0 & w(x - x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(x - x_n) \end{bmatrix} \quad (3.29)$$

The function J can be minimized by its first derivative of J with respect to a , i.e.,

$$\frac{\partial J}{\partial \mathbf{a}} = 0 \quad (3.30)$$

It results in the following system of m equations.

$$\frac{\partial J}{\partial a_k} = 0 \quad \Rightarrow \quad \sum_{I=1}^n w(x - x_I) 2p_k(x_I) [\mathbf{p}^T(x_I) \mathbf{a}(x) - u_I] = 0 \quad 1 \leq k \leq m \quad (3.31)$$

Eq. (3.31) in vector notation,

$$\sum_{I=1}^n w(x - x_I) \mathbf{p}(x_I) [\mathbf{p}^T(x_I) \mathbf{a}(x) - u_I] = 0 \quad (3.32)$$

On simplification of Eq. (3.32), we have

$$\sum_{I=1}^n w(x - x_I) \mathbf{p}(x_I) \mathbf{p}^T(x_I) \mathbf{a}(x) = \sum_{I=1}^n w(x - x_I) \mathbf{p}(x_I) u_I \quad (3.33)$$

It leads to the following set of system matrices.

$$\mathbf{A}(\mathbf{x}) \mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x}) \mathbf{u}_s \quad (3.34)$$

where \mathbf{u}_s , vector of nodal parameters of field function in support domain Ω_s , is given as

$$\mathbf{u}_s = \{u_1 \quad u_2 \quad \cdots \quad u_n\}^T \quad (3.35)$$

and $\mathbf{A}(\mathbf{x})$, weighted *moment matrix*, is defined by

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^n w(\mathbf{x} - \mathbf{x}_I) \mathbf{p}(\mathbf{x}_I) \mathbf{p}^T(\mathbf{x}_I) \quad (3.36)$$

For a 1-D space with the linear basis ($m = 2$), \mathbf{A} is a 2×2 symmetric matrix that can explicitly be written as

$$\mathbf{A}_{2 \times 2}(\mathbf{x}) = \sum_{I=1}^n w_I(\mathbf{x}) \mathbf{p}(\mathbf{x}_I) \mathbf{p}^T(\mathbf{x}_I) = \sum_{I=1}^n w(x - x_I) \begin{bmatrix} 1 & x_I \\ x_I & x_I^2 \end{bmatrix} \quad (3.37)$$

Similarly, for 2-D space problems

$$\mathbf{A}_{3 \times 3}(\mathbf{x}) = \sum_{I=1}^n w_I(\mathbf{x}) \mathbf{p}(\mathbf{x}_I) \mathbf{p}^T(\mathbf{x}_I) = \sum_{I=1}^n w(x - x_I) \begin{bmatrix} 1 & x_I & y_I \\ x_I & x_I^2 & x_I y_I \\ y_I & x_I y_I & y_I^2 \end{bmatrix} \quad (3.38)$$

In a well defined MLS approximation, matrix \mathbf{A} must be non-singular. The matrix \mathbf{B} is defined as

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} w_1(\mathbf{x})\mathbf{p}(x_1) & w_2(\mathbf{x})\mathbf{p}(x_2) & \cdots & w_n(\mathbf{x})\mathbf{p}(x_n) \end{bmatrix} \quad (3.39)$$

which is $(3 \times n)$ matrix for 2-D space and can be expressed as

$$\begin{aligned} \mathbf{B}_{3 \times n}(\mathbf{x}) &= \begin{bmatrix} w(x-x_1) \begin{bmatrix} 1 \\ x_1 \\ y_1 \end{bmatrix} & w(x-x_2) \begin{bmatrix} 1 \\ x_2 \\ y_2 \end{bmatrix} & \cdots & w(x-x_n) \begin{bmatrix} 1 \\ x_n \\ y_n \end{bmatrix} \end{bmatrix} \\ &= \begin{bmatrix} w_1 & w_2 & \cdots & w_n \\ x_1 w_1 & x_2 w_2 & \cdots & x_n w_n \\ y_1 w_1 & y_2 w_2 & \cdots & y_n w_n \end{bmatrix}_{3 \times n} \end{aligned} \quad (3.40)$$

By solving Eq. (3.34) for $\mathbf{a}(\mathbf{x})$, we have

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u}_s \quad (3.41)$$

By substituting Eq. (3.41) into Eq. (3.23), we obtain

$$u^h(\mathbf{x}) = \sum_{i=1}^n \phi_i(x)u_i = \mathbf{\Phi}^T(\mathbf{x})\mathbf{u}_s \quad (3.42)$$

where $\mathbf{\Phi}(\mathbf{x})$ is the matrix obtained by MLS shape functions approximation corresponding to the influencing nodes present in support domain of the points x . It can be expressed as

$$\mathbf{\Phi}^T(\mathbf{x}) = \underbrace{\mathbf{p}^T(\mathbf{x})}_{1 \times 3} \underbrace{\mathbf{A}^{-1}(\mathbf{x})}_{3 \times 3} \underbrace{\mathbf{B}(\mathbf{x})}_{3 \times n} = \left\{ \phi_1(\mathbf{x}) \quad \phi_2(\mathbf{x}) \quad \cdots \quad \phi_n(\mathbf{x}) \right\}_{1 \times n} \quad (3.43)$$

The shape function $\phi_i(\mathbf{x})$ for i^{th} node is defined by

$$\begin{aligned} \phi_i(\mathbf{x}) &= [\phi_1(x) \quad \phi_2(x) \quad \phi_3(x) \cdots \phi_n(x)] \\ &= \sum_{j=1}^m p_j(\mathbf{x})(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}))_{ji} = (\mathbf{p}^T(\mathbf{x})(\mathbf{A}^{-1}\mathbf{B}))_i \end{aligned} \quad (3.44)$$

To obtain the partial derivatives of the shape functions (Belytschko et al., 1996b), Eq. (3.43) is re-written as follow.

$$\Phi^T(\mathbf{x}) = \gamma^T(\mathbf{x})\mathbf{B}(\mathbf{x}) \quad \text{where} \quad \gamma^T = \mathbf{p}^T \mathbf{A}^{-1} \quad (3.45)$$

Since \mathbf{A} is symmetric matrix, $\gamma(\mathbf{x})$ can be obtained from above Eq. (3.45).

$$\mathbf{A}\gamma = \mathbf{p} \quad (3.46)$$

The partial derivatives of γ can than be obtained by solving the following equations.

$$\mathbf{A}\gamma_{,i} = \mathbf{p}_{,j} - \mathbf{A}_{,i}\gamma \quad (3.47)$$

$$\mathbf{A}\gamma_{,ij} = \mathbf{p}_{,ij} - (\mathbf{A}_{,i}\gamma_{,j} + \mathbf{A}_{,j}\gamma_{,i} + \mathbf{A}_{,ij}\gamma) \quad (3.48)$$

$$\begin{aligned} \mathbf{A}\gamma_{,ijk} = & \mathbf{p}_{,ijk} - (\mathbf{A}_{,i}\gamma_{,jk} + \mathbf{A}_{,j}\gamma_{,ik} + \mathbf{A}_{,k}\gamma_{,ij} + \mathbf{A}_{,ij}\gamma_{,k} \\ & + \mathbf{A}_{,ik}\gamma_{,j} + \mathbf{A}_{,jk}\gamma_{,i} + \mathbf{A}_{,ijk}\gamma) \end{aligned} \quad (3.49)$$

where i , j and k denotes the x , y , z coordinates and a ‘comma’ indicates a partial derivatives with respect to the indicated coordinate that follows. Then, the partial derivatives of the shape function can be obtain using the following expressions.

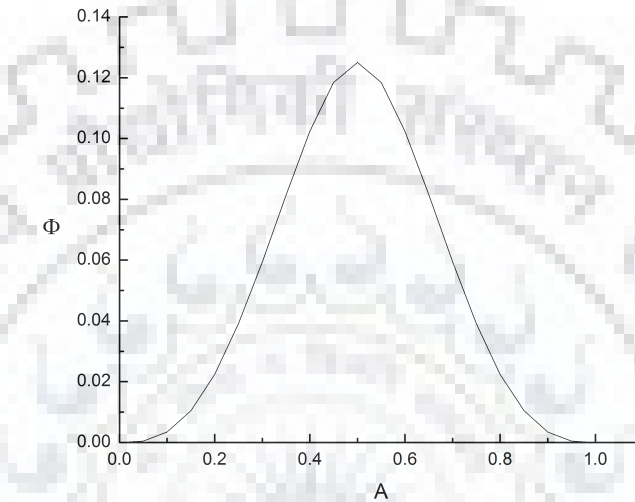
$$\Phi_{,i}^T = \gamma_{,i}^T \mathbf{B} + \gamma^T \mathbf{B}_{,i} \quad (3.50)$$

$$\Phi_{,ij}^T = \gamma_{,ij}^T \mathbf{B} + \gamma_{,i}^T \mathbf{B}_{,j} + \gamma_{,j}^T \mathbf{B}_{,i} + \gamma^T \mathbf{B}_{,ij} \quad (3.51)$$

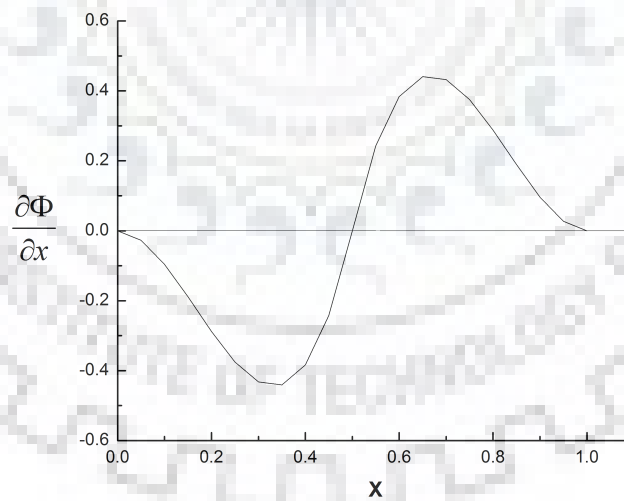
$$\begin{aligned} \Phi_{,ijk}^T = & \gamma_{,ijk}^T \mathbf{B} + \gamma_{,ij}^T \mathbf{B}_{,k} + \gamma_{,ik}^T \mathbf{B}_{,j} + \gamma_{,jk}^T \mathbf{B}_{,i} + \gamma_{,i}^T \mathbf{B}_{,jk} \\ & + \gamma_{,j}^T \mathbf{B}_{,ik} + \gamma_{,k}^T \mathbf{B}_{,ij} + \gamma^T \mathbf{B}_{,ijk} \end{aligned} \quad (3.52)$$

All the field nodes covered by the support domain are used to perform the MLS approximation for the unknown function at point x . The number of nodes n should be sufficient to avoid the singularity of the moment matrix \mathbf{A} , which is invertible, and to provide the interpolation stability. In order to ensure the existence of \mathbf{A}^{-1} as well as conditioned \mathbf{A} , we usually take $n \gg m$. Here n is determined through trial and error method by adjusting

the dimensionless parameter α . There is no theoretically best value of n present yet (Liu and Gu, 2005). The representative shape function and its first derivative constructed by using quadratic weight function at computational point $x = 11$ for 21 nodes is shown in Figures 3.8 and 3.9.



(a) shape function



(b) derivative of shape function

FIGURE 3.8: Shape function and its first derivative at point of interest $x = 11$ and $x_I = (1, 2, 3, \dots, 21)$

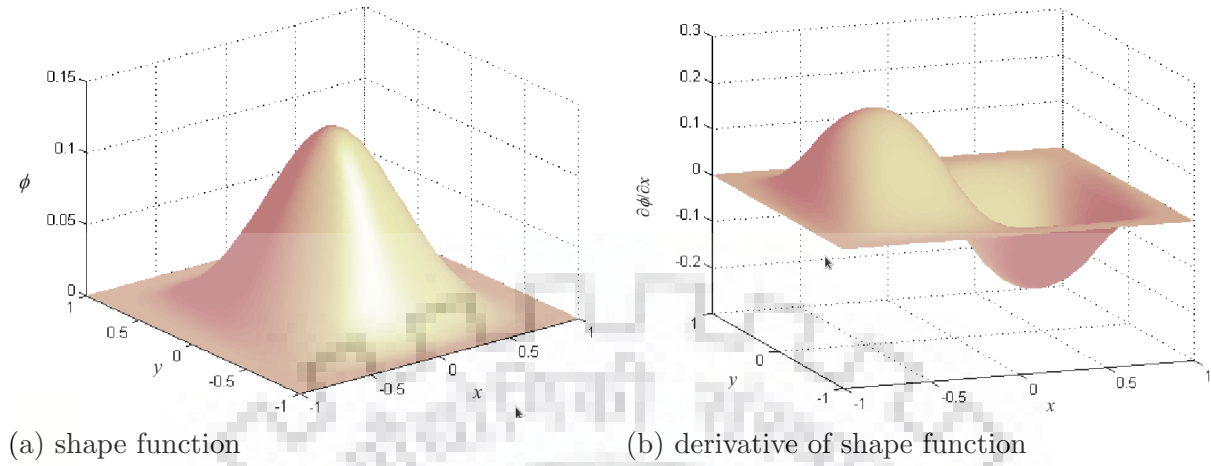


FIGURE 3.9: Shape function and its first derivative in 3-D space by using cubic spline weight function (Liu and Gu, 2005).

3.4.3.2 Properties of MLS shape functions

MLS shape functions have the following four characteristics.

1. **Consistency:** The consistency of MLS approximation depends on the complete order of the monomial used as the polynomial basis. The shape function possess C^q consistency for the q^{th} complete order of monomial (Krongauz and Belytschko, 1996; Liu, 2003a).
2. **Partition of unity:** The shape function $\phi_i(x)$ must have the partition of unity if any constant term is included in the basis, i.e.,

$$\sum_{i=1}^n \phi_i(x) = 1 \quad (3.53)$$

3. **Delta function property:** Since the function obtained by MLS approximation does not pass through the nodal values, hence shape functions do not satisfy the Kronecker delta condition, as in FEM. Thus

$$\phi_i(x_j) \neq \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (3.54)$$

This property makes the imposition of essential boundary conditions more complicated than that in FEM.

4. **Reproducibility:** The different types of basis functions can be used to construct the shape function. These basis functions used to construct the shape function can be reproduced(Liu, 2003a).

3.4.4 Flowchart of the shape function calculation

The flowchart for the calculation of the shape function by using the MLS approximation is presented in Figure 3.10.

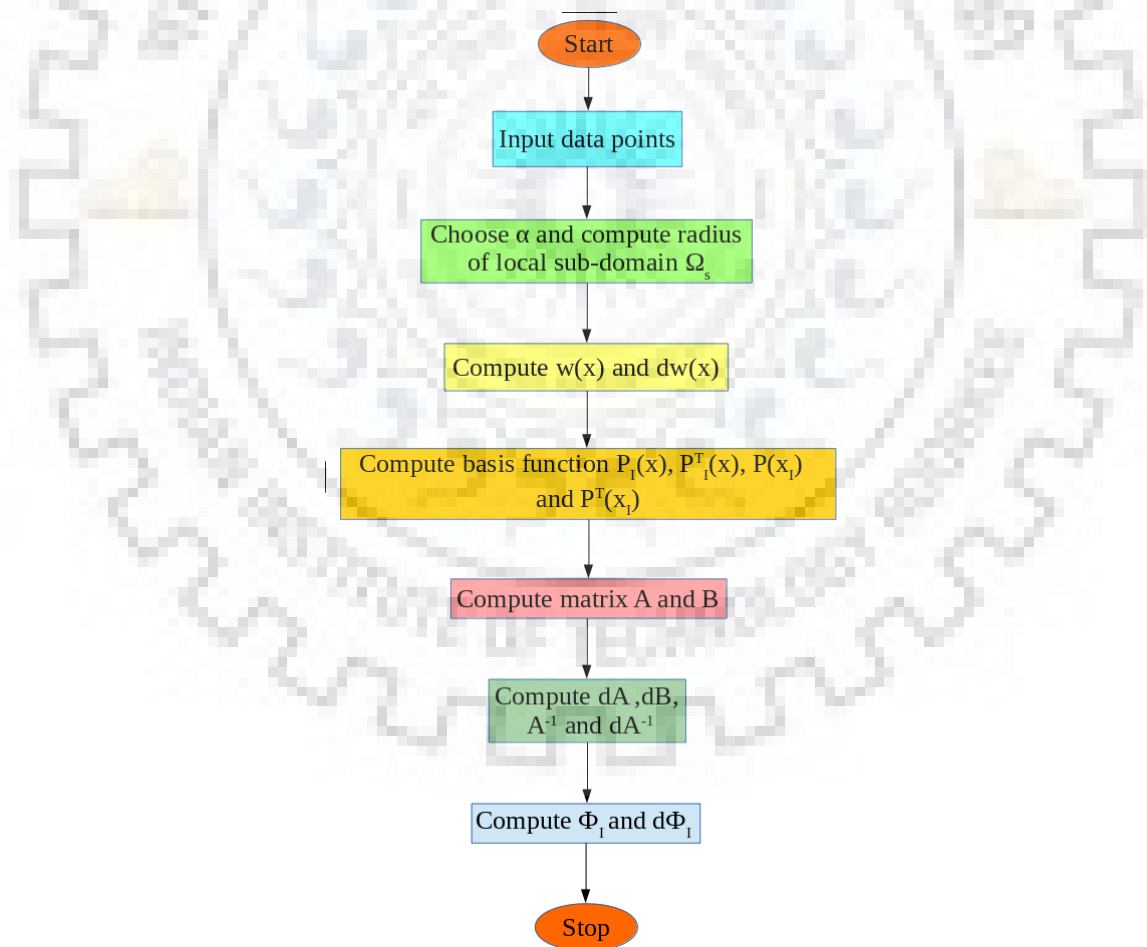


FIGURE 3.10: Flowchart for computation of MLS shape functions.

3.4.5 Imposition of boundary conditions

It is evident from the existing literature that some of the Mfree methods still have the difficulty in imposition of the Dirichlet BCs. It is because most of the Mfree shape functions are constructed by using MLS approximation. An implementation of the essential BCs becomes difficult due to use of such trial functions which do not satisfy the Kronecker delta function property.

Various numerical techniques have been proposed to enforce the essential BCs by previous researchers, such as the Lagrange multiplier method (Belytschko et al., 1994). The Lagrange multipliers λ can be viewed as smart forces constraining $T - \bar{T} = 0$. In order to obtain the discretized system of equations, the Lagrange multipliers λ , which are unknown functions of the coordinates, need to be interpolated by using their nodal values and shape functions for nodes on the essential boundaries.

$$\lambda_T^h = [\phi_1 \quad \phi_2 \dots \phi_{n_\lambda}] \begin{bmatrix} \lambda_{T_1} \\ \lambda_{T_2} \\ \vdots \\ \lambda_{T_{n_\lambda}} \end{bmatrix} = \phi_I(s)_{(1 \times n_\lambda)} \cdot \lambda_{(n_\lambda \times 1)} \quad (3.55)$$

where n_λ is the number of nodes used for the interpolation. ϕ_I is the shape function for the I^{th} node on essential boundary, s is the arc length along the essential boundary and λ is the vector of the nodal Lagrange multipliers of field nodes on the same essential boundary.

The penalty parameter approach (Gu and Liu, 2001a) is a convenient and alternative way for enforcing the essential BCs. In this approach, the diagonal elements of the stiffness matrix (K) are modified as follow.

$$\bar{K}_{ii} = \alpha \cdot K_{ii} \quad (3.56)$$

where α , the penalty factor, is much larger number (Gu and Liu, 2001a), i.e.,

$$\alpha = 10^4 - 10^{13} \times (K_{ii})_{max} \quad (3.57)$$

Such a modification may lead to ill-conditioned stiffness matrix. Zhu (2010) noted that the penalty parameter can be chosen in between 10^{10} to 10^{15} .

The direct interpolation method (Liu and Gu, 2004) and the transformation method (Atluri et al., 1999b; Atluri and Shen, 2002a) in which, as in the MLPG, the fictitious nodal values do not pass through the nodal data points, as depicted in Figure 3.7. However, meshless approximation can be re-interpreted from the point of view of an interpolation, which passes through the actual node values as $u^h(x) = u_i$ (Figure 3.11). Various possibilities

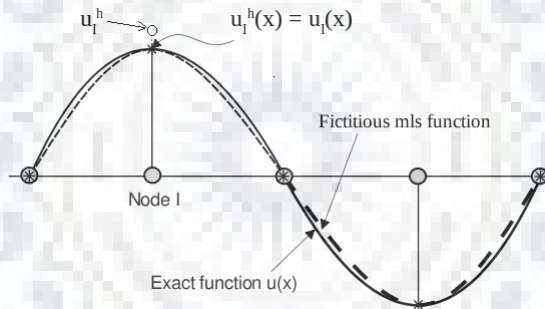


FIGURE 3.11: u_I^h is the fictitious nodal data at node I and u_I is the exact nodal value at node I for interpolation

have been explored in the literature for the correct implementation of boundary conditions in meshless methods. For instance, imposition of boundary conditions (Belytschko et al., 1996b; Fries and Matthies, 2004), enforcement of essential boundary conditions in meshless methods using finite elements (Krongauz and Belytschko, 1996), boundary conditions in the EFG method (Mukherjee and Mukherjee, 1997b), boundary conditions implementation in meshless methods (Gunther and Liu, 1998), implementation of essential boundary conditions (Gavete et al., 2000; Mendez and Huerta, 2004), and comparison between Lagrange multiplier and penalty methods for setting boundary conditions in Mfree methods (Ramirez et al., 2011), etc. are reported in literature.

In the present work, direct interpolation technique is used for essential boundary conditions due to its simplicity and easy to implementations. In the MLPG method, the weak form of a PDE facilitates to impose the Neumann and Robin BCs directly, but impositions of essential BC is very complex. To impose the essential BCs by direct interpolation method, consider the following simple diffusion equation in 2-D space,

$$k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = \hat{q} \quad \text{in } \Omega \quad (3.58)$$

subjected to the following essential BC

$$T = T_1 \quad \text{on } \Gamma_u \quad (3.59)$$

The weighted integral form of above equation in Ω_x can be written as

$$\int_{\Omega_s} \left[k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) - \hat{q} \right] w \partial\Omega = 0 \quad (3.60)$$

where \hat{q} , T , k and w represent the heat source per unit volume, temperature, thermal conductivity and weight function, respectively.

To reduce this high-order differentiability requirement on T , Eq. (3.60) can be integrated by parts. By using ‘Gauss divergence theorem’, the following local weak formulation equation is obtained.

$$\int_{\Omega_s} \left(k \frac{\partial T}{\partial x} \frac{\partial w}{\partial x} + k \frac{\partial T}{\partial y} \frac{\partial w}{\partial y} + \hat{q} w \right) \partial\Omega - \int_{\Gamma_{si}} \left(\frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} \right) n_j w \partial\Gamma \quad (3.61)$$

$$- \int_{\Gamma_{su}} \left(\frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} \right) n_j w \partial\Gamma - \int_{\Gamma_{st}} \left(\frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} \right) n_j w \partial\Gamma = 0 \quad (3.62)$$

where n_j the outward unit vector to $\Gamma_s (= \Gamma_{si} \cup \Gamma_{su} \cup \Gamma_{st})$. Γ_{si} represents the internal boundary of the quadrature domain where as Γ_{su} and Γ_{st} are the part of essential and natural BCs that intersect with the quadrature domain. Both Γ_{su} and Γ_{st} vanish when the local sub-domain is located entirely within the global domain (see Figure 3.12) and $\Gamma_s = \Gamma_{si}$. Unlike the Galerkin method, the Petrov-Galerkin method chooses the trial and

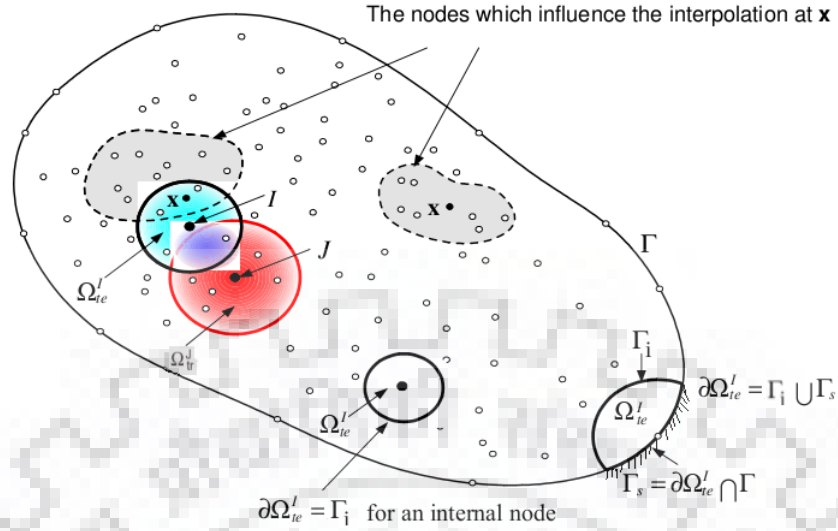


FIGURE 3.12: Schematics of the MLPG method and the boundary intersection domain [Atluri et al. \(1999b\)](#)

test functions from different spaces Ω_{tr}^J and Ω_{te}^I , respectively. The weight function w is purposely selected in such a way that it vanishes on Γ_{si} . By using this selected weight function w , Eq. (3.62) transforms as follow.

$$\int_{\Omega_s} \left(k \frac{\partial T}{\partial x} \frac{\partial w}{\partial x} + k \frac{\partial T}{\partial y} \frac{\partial w}{\partial y} + \hat{q}w \right) \partial\Omega - \int_{\Gamma_u} \left(\frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} \right) n_j w \partial\Gamma - \int_{\Gamma_t} \left(\frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} \right) n_j w \partial\Gamma = 0 \quad (3.63)$$

If the quadrature domain is located entirely in the global domain, the integral parts related to Γ_{su} and Γ_{st} vanish, and Petrov-Galerkin form (Eq. 3.63) can be simplified as

$$\int_{\Omega_s} \left(k \frac{\partial T}{\partial x} \frac{\partial w}{\partial x} + k \frac{\partial T}{\partial y} \frac{\partial w}{\partial y} + \hat{q}w \right) \partial\Omega = 0 \quad (3.64)$$

Eq. (3.64) is used to establish the discrete algebraic equations for those nodes whose quadrature domains falls entirely within the global domain Ω . Further, Eq. (3.62) is used to create the discrete algebraic equations for those nodes whose quadrature domain are intersects with the problem domain boundary.

Based on the concept of MLPG, the test functions over each local sub domain Ω_s can be chosen in a variety of ways ([Atluri and Shen, 2002a,b](#)), as summarized in Table 3.1.

TABLE 3.1: MLPG types and their characteristics

Methods	Test function w in Ω_{te}	local weak form over each Ω_s	Relation between Ω_{te} and Ω_{tr}	Integral to evalu- ate the weak form
MLPG-1	MLS weight function	LSWF	$\Omega_{te} < \Omega_{tr}$	domain itegral
MLPG-2	Kronecker delta $\delta(\mathbf{x}, \mathbf{x}_I)$	LUSWF1	Ω_{te} can be arbitrary	none
MLPG-3	least square $\phi_{ii}^I(x)(w_3)$	LUSWF1	$\Omega_{te} = \Omega_{tr}$	domain integral
MLPG-4	fundamental solution $u^*(w_4)$	LUSWF2	$\Omega_{te} < \Omega_{tr}$	singular boundary integral
MLPG-5	Constant (w_5)	LSWF	$\Omega_{te} < \Omega_{tr}$	regualr boundary in- tegral
MLPG-6	Same as trial function	LSWF	$\Omega_{te} = \Omega_{tr}$	domain integral

LSWF: local symmetric weak form; LUSWF1: local unsymmetric weak form; LUSWF2: local unsymmetric weak form with twice use of divergence theorem;

3.4.6 Direct interpolation method

In MLPG, the system equations are constructed node by node. Each row of the global stiffness matrix (\mathbf{K}) and the global force vector (\mathbf{F}) refers to each field node. With this structural feature of the system equation of MLPG, the direct interpolation method is used to enforce essential BCs as follow.

Assume the temperature (T) at I^{th} field node on the essential boundary is prescribed as below.

$$T_I^h = \bar{T} \quad (3.65)$$

A linear equation for I^{th} node can, therefore, be written as

$$\phi_1 T_1 + \phi_2 T_2 + \dots + \phi_n T_n = \bar{T}_I \quad (3.66)$$

It represents for all those nodes where the essential BCs are applicable but similar algebraic equations for other nodes should still be obtained from weighted residual form of PDE (Liu and Gu, 2004, 2005). Eq. (3.65) is assembled (stacked) into the system equation directly for all field nodes to obtain the global system equations given as below.

$$\mathbf{K}_{n \times n} \mathbf{T}_{n \times 1} = \mathbf{F}_{n \times 1} \quad (3.67)$$

Where \mathbf{K} , \mathbf{F} and \mathbf{T} refer to the global stiffness matrix, global force vector and the required variable parameter, respectively.

3.5 Integration techniques in MLPG

Integration techniques are frequently used in engineering computations, solid and fluid mechanics problems, etc. to evaluate integration of a function over in a certain domain. CFD techniques such as FEM, FVM require exact evaluation of integration to obtain the global (*stiffness*) matrix \mathbf{K} according to the following general formulation.

$$\mathbf{K} = \int_{\Omega} f(x) \partial\Omega \quad (3.68)$$

where Ω represents a domain of integration. To calculate the area or volume of the quadrature domain, various integration techniques have been developed by previous researchers. Most commonly used numerical integration techniques are not effective in meshfree method due to its complex nature. In the meshfree methods, numerical integration is performed by using *Gauss quadrature rule* consisting of quadrature points and weights, which are specifically generated for certain function and/or in domain. These integration points and weights are obtained by solving a set of functions within certain intervals. Such set of functions could be polynomials, trigonometric and basis functions of a particular space which define the particular domain (Mousavi et al., 2010).

The numerical integration in meshfree methods using the Galerkin weak form formulation is very complex and exhibits major computational cost. In FEM, numerical integration or numerical quadrature, of an element, involves approximation of integrand by a polynomial of sufficient degree, because the integral of a polynomial can be evaluated exactly. Hence, some researchers have reverted back to developing FE techniques incorporating certain aspects (basically integration) of the meshless methods. Although the integration by using background cell in other meshless methods such as EFG, PU and RKPM methods preserve symmetry of the sampling points, accurate solutions may not be achieved with too much of an under integration (Atluri et al., 1999b). It should be noted that well known meshfree

methods have utilized precious numerical integration (Belytschko et al., 1994; Liu and Jun, 1995; Beissel and Belytschko, 1996; Duarte and Oden, 1996a; Oden et al., 1998), but it requires a background mesh for integration. Hence, these are not the truly meshfree methods.

In the prospects of truly meshfree methods, Atluri et al. (1999a,b) presented a mapping technique in which intersecting region ($\Omega_{tr}^J \cap \Omega_{te}^I$) is transformed into unit circle to calculate the area using simple Gaussian quadrature. To obtain each K_{IJ} for each value of J , Ω_{tr}^J intersects only partially with Ω_{te}^I . Their numerical experimentation suggested that number of small partitions in a domain of integration, such as a sub-domain Ω_{te}^I , or $\Omega_{tr}^J \cap \Omega_{te}^I$, give better solutions, in comparison to the integrating over a whole domain with a large number of integration points (Atluri et al., 1999a,b). Further, numerical integration is performed without a mesh using Gauss quadrature rule termed as ‘method of finite square’ (De and Bathe, 2000, 2001a,b) a truly meshless technique by using MLS approximation. A good comparison of numerical integration rules for MLPG methods to preserve the solution accuracy in conjunction with reduced computational cost can be found elsewhere (Mazzia et al., 2007).

To avoid the use of global background cell for integration purpose, a so called local weak form (LWF) is used to develop the MLPG method. Some other variations of MLPG proposed based on LWFs are called meshfree LWFs methods. When LWF can be used for a field node, the numerical integration is carried out over a local quadrature domain defined for node \mathbf{x}_I , which can also be the local domain where the test functions is defined. All the sub-domains are schematically drawn in Figure 3.13. For instance, Ω_w is the domain of non-zero weight (test) function, i.e. $w_i(x) \neq 0$ and quadrature domain Ω_q of node I is a domain for the integration in Eq. (3.62). Although both Ω_w and Ω_q do not have to be same, we have to choose a regular and simple shape for local domain (such as circle, sphere, rectangle, squares, ellipsoids, etc.) for all nodes, as shown in Figure 3.13, and the integration can be done numerically within the local domain. Therefore, a global background mesh is not required in MLPG for either function approximation or integration. The procedure is quite similar to numerical methods based on the strong form formulation,

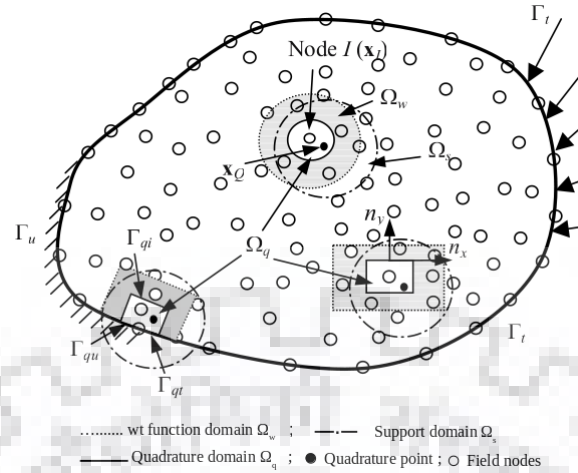


FIGURE 3.13: An integration domain modeled using the MFree local weak-form. Ω_q and Ω_w for field nodes, and the support domain Ω_s for a Gauss quadrature point x_q (Liu and Gu, 2005).

such as FDM. However, because MLS approximation is used in MLPG, special treatment is needed to enforce the essential BCs (Liu and Gu, 2005). For the local integration in Ω_q , the sample points or nodes are needed to employ the Gaussian quadrature scheme. For each quadrature points x_q situated in a local domain, MLS approximation is performed to compute the shape function and to obtain the integrand. Polar coordinate system is used to generate the sample points in the quadrature domain Ω_q and divided into 4 equal parts to integrate separately, as discussed in subsequent section.

3.5.1 Gaussian quadrature

The n -points Gaussian quadrature rule, is a quadrature rule constructed to yield an exact result for polynomials of degree $(2n - 1)$ or less by a suitable choice of the points x_i and weights w_i for $i = 1, 2, \dots, n$. The domain of integration for such a rule is conventionally taken as $[-1, 1]$, and the rule is stated as:

$$\int_{-1}^1 f(x)dx = \sum_{i=1}^n w_i f(x_i) \quad (3.69)$$

where $f(x)$ is the integrand of integral term. Gaussian quadrature as above will only produce good results if the function $f(x)$ is well approximated by a polynomial function

within the range $[-1, 1]$. The method is not suitable for functions with singularities. However, if the integrated function can be written as $f(x) = w(x)g(x)$, where $g(x)$ is approximated polynomial and $w(x)$ is known, then alternative weights w'_i and points x'_i that depend on the weighting function $w(x)$ may give better results. It can be written as

$$\int_{-1}^1 f(x)dx = \int_{-1}^1 w(x)g(x) \approx \sum_{i=1}^n w'_i g(x'_i) \quad (3.70)$$

The method of subdivision of quadrature domains works well when the number of subdivisions is sufficiently large (Atluri et al., 1999b).

In Present work, the method proposed by Liu (2003a) for complex domains is used. They used the coordinate transform to transfer non-rectangular quadrature domain (or sectors) into standard squares for the Gauss quadrature scheme, as shown in Figure 3.14. The

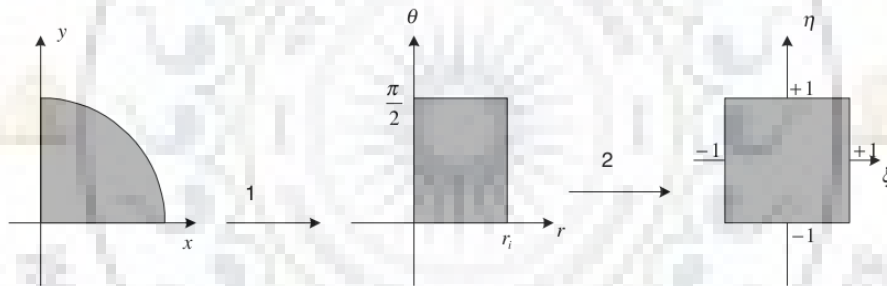


FIGURE 3.14: Transformation of first quarter of a circle into a standard square (Liu, 2003a)

Present work uses circles centered at the node as our integration cell. The following effective procedure for an accurate integration over circular quadrature domains by introducing a number of quadrature points x_Q in integration domain is adopted. The polar coordinate system is used to generate points (x_q) in quadrature domain. A circular quadrature domain Ω_q , either located entirely within the global domain Ω or intersecting with global domain boundary Γ , the transformation for the first quarter of a circle (Figure 3.14) consists of the following steps.

First, map the quarter $r - \theta$ plane into $x - y$ plane (rectangle region) by using

$$\begin{cases} x = r \cos \theta & r = a\xi + b; \\ y = r \sin \theta & \theta = A\eta + B \end{cases} \quad (3.71)$$

where r and θ varies in the ranges of $[-1, 1]$ and $[0, 2\pi]$, respectively. On solving for r and θ , we get

$$r = \frac{R_q}{2}(\xi + 1) \quad \text{and} \quad \theta = \frac{\pi}{4}(\eta + 1) \quad (3.72)$$

where R_q being the radius of the quadrature domain. Substitution of (Eq. 3.72) in Eq. (3.71) results

$$x = \frac{R_q}{2}(\xi + 1) \cos \left[\frac{\pi}{4}(\eta + 1) \right] \quad \text{and} \quad y = \frac{R_q}{2}(\xi + 1) \sin \left[\frac{\pi}{4}(\eta + 1) \right] \quad (3.73)$$

Eq. (3.73) will transform the circular quarter directly into standard square in a single operation. The general formulation for all four quadrants ($k = 1, 2, 3$ and 4) of the subdomain can be written concisely as follows.

$$x = \frac{R_q}{2}(\xi + 1) \cos \left[\frac{\pi}{4}(\eta + 2k - 1) \right] \quad \text{and} \quad y = \frac{R_q}{2}(\xi + 1) \sin \left[\frac{\pi}{4}(\eta + 2k - 1) \right] \quad (3.74)$$

The Jacobian, similar for all four quarters, resulting from this transformation can be expressed in a simpler form as follows.

$$|J| = \begin{vmatrix} \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \end{vmatrix} \quad (3.75)$$

The double integral of a function can than be performed using the Gauss quadrature scheme over the square domain

$$\int \int_{\Omega_Q} f(x, y) dx dy = \int_{-1}^1 \int_{-1}^1 f(\eta, \xi) d\eta d\xi = \sum_1^{n_g} \sum_1^{n_g} w_i w_j f(\eta_i, \xi_j) \quad (3.76)$$

where w_i are the Gauss weights at Gauss points ($i = 1, 2, \dots, n_g$) and

$$f(\eta, \xi) = f(x(\eta, \xi), y(\eta, \xi)) |J| \quad (3.77)$$

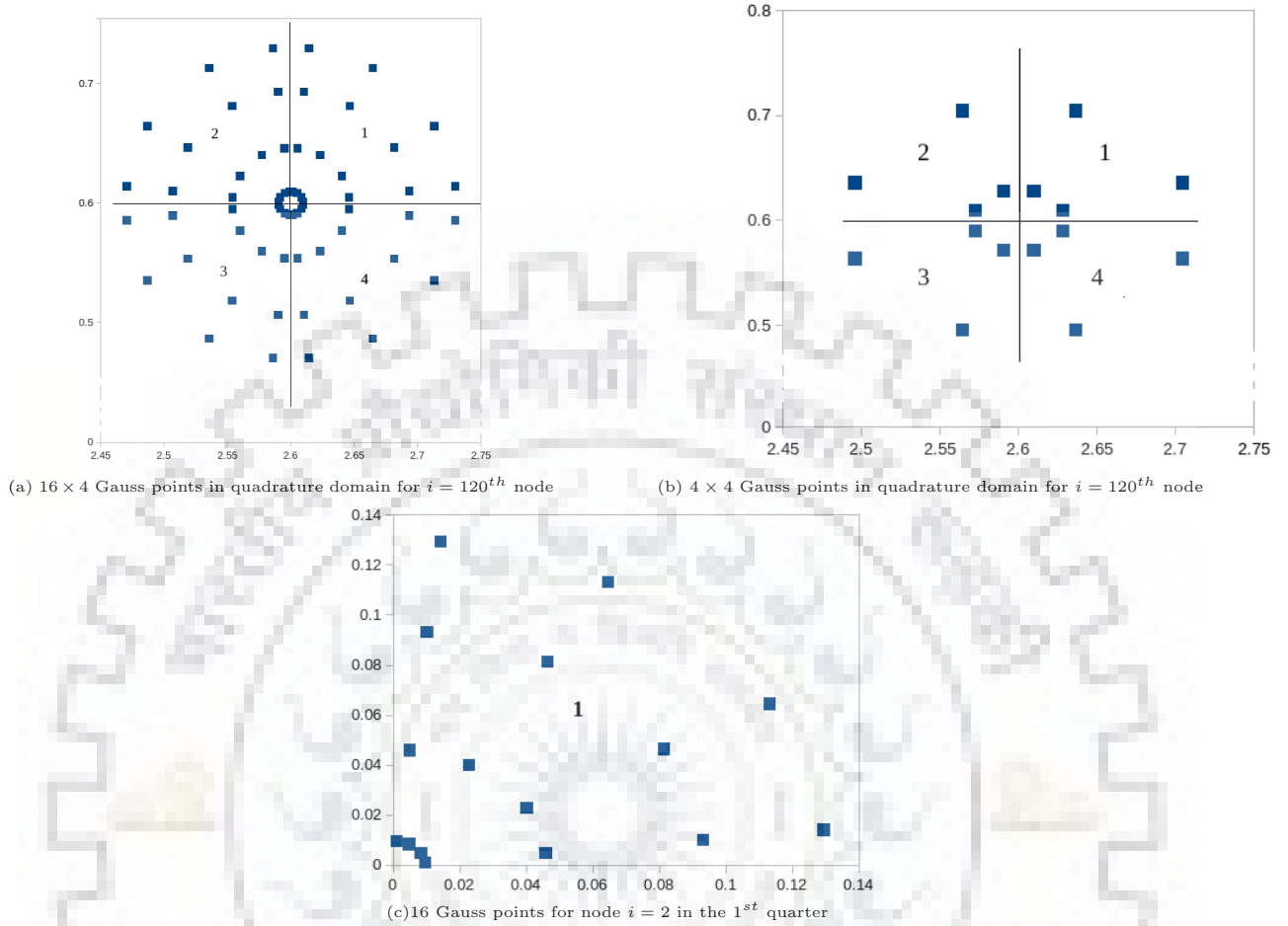


FIGURE 3.15: The sample of Gauss points in circular domain presented for $i = 2$ and $i = 120^{th}$ nodes in 156 uniformly distributed nodes in rectangle domain.

where J is the Jacobin corresponding to the coordinate transformation. The numerical integration will than becomes

$$\int_{\Omega_Q} f(x) d\Omega = \sum_{k=1}^M \left(\sum_{l=1}^{n_g} A_l J_l F_l^* \right)_k \quad (3.78)$$

where $A_l (= w_i w_j)$ is the weight for the numerical integration and M denotes the total number of partitions for integration.

3.6 MLPG algorithm

In general, flow and thermal problems can be categorized into (a) convection-diffusion problems and (b) Navier-Stokes equations based problems. In first case, the pressure does

not affect the flow field, while in second case, pressure is coupled with the velocity field. The solution algorithms using MLPG method for both the cases are outlined below.

3.6.1 Solution of convection-diffusion problems

The following sequential steps can be utilized to implement the MLPG method (Atluri and Zhu, 1998a).

1. Choose the total number of nodes in the global domain Ω including global boundaries.
2. Choose the basis function and suitable weight function according to problem domain such that MLS approximation is well defined.
3. Determine the local sub-domains (Ω_s) and their corresponding local boundaries ($\partial\Omega_s$) for each nodes.
4. Make a loop over all nodes located inside the global domain.
 - (a) Assuming the local domain represents for a quadrature domain, i.e., $\Omega_q \equiv \Omega_s$ for each node
 - (b) Determine Gaussian quadrature points x_q for each Ω_s and $\partial\Omega_s$
 - (c) Loop over quadrature points x_q in each quarter of Ω_q and on their $\partial\Omega_q$
 - i. Determine the nodes x_i located in the domain of definition of the MLS approximation for the trial function at point x_q , i.e., those nodes with $w_i(x_q) > 0$
 - ii. For each x_q , Calculate shape functions and their derivatives, i.e., $\phi_i(x_q)$ and $\phi'_i(x_q)$.
 - iii. Evaluate numerical integrals of discretized partial differential equations (Eq. 3.78).
 - iv. Assemble contributions of each computing node to the linear system, i.e., formation of global stiffness and global vector matrices (\mathbf{K} and \mathbf{F}).

5. Solve the linear system for the fictitious nodal values (Eq. 3.67).

A flow chart of MLPG algorithm is depicted in Figure 3.16 and 3.17.

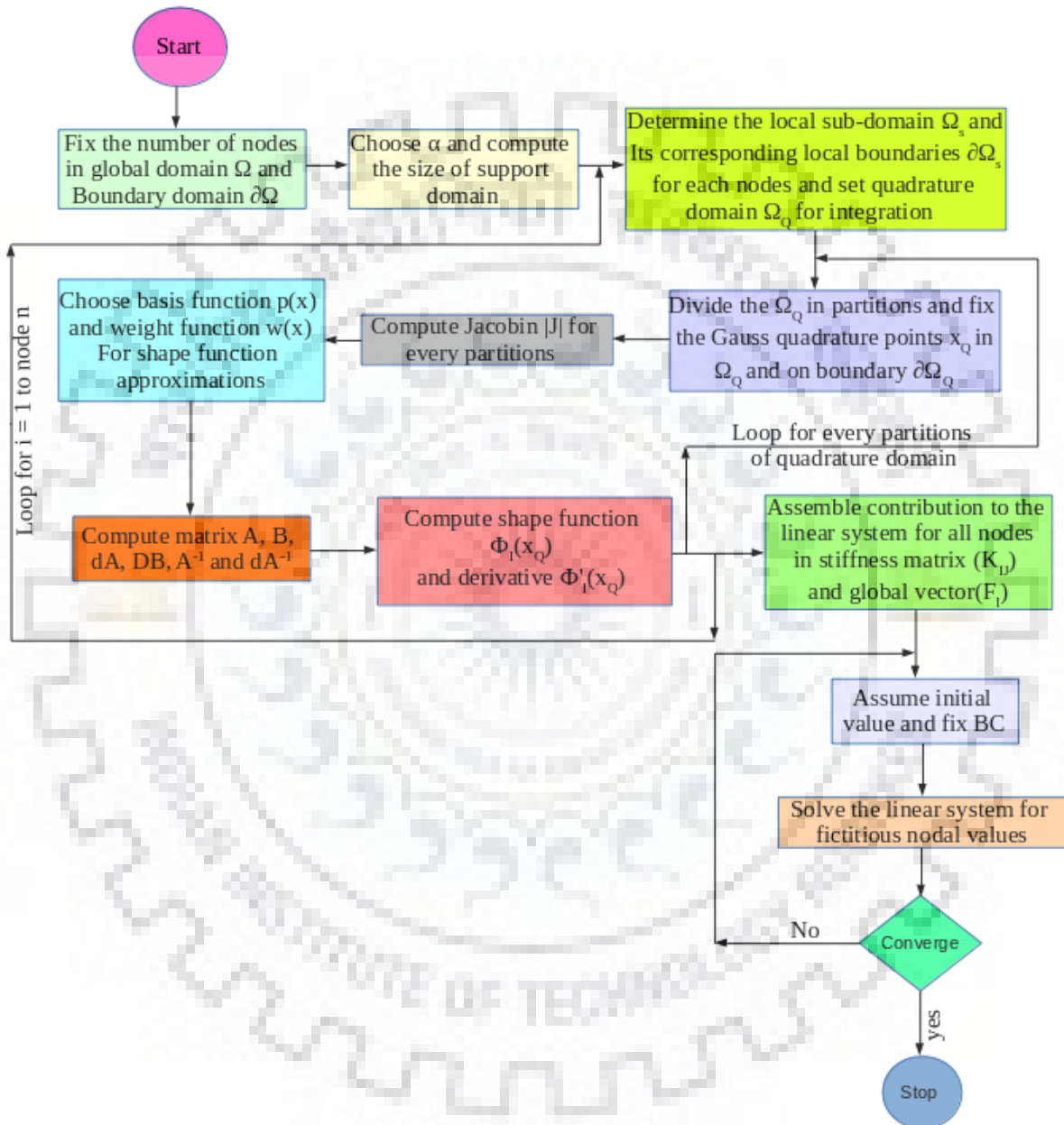


FIGURE 3.16: MLPG flow chart for any convection-diffusion problems.

3.6.2 Solution of Navier-Stokes equations

MLPG algorithm for solution of Navier-Stokes equations for Newtonian and non-Newtonian fluids can be written as follows.

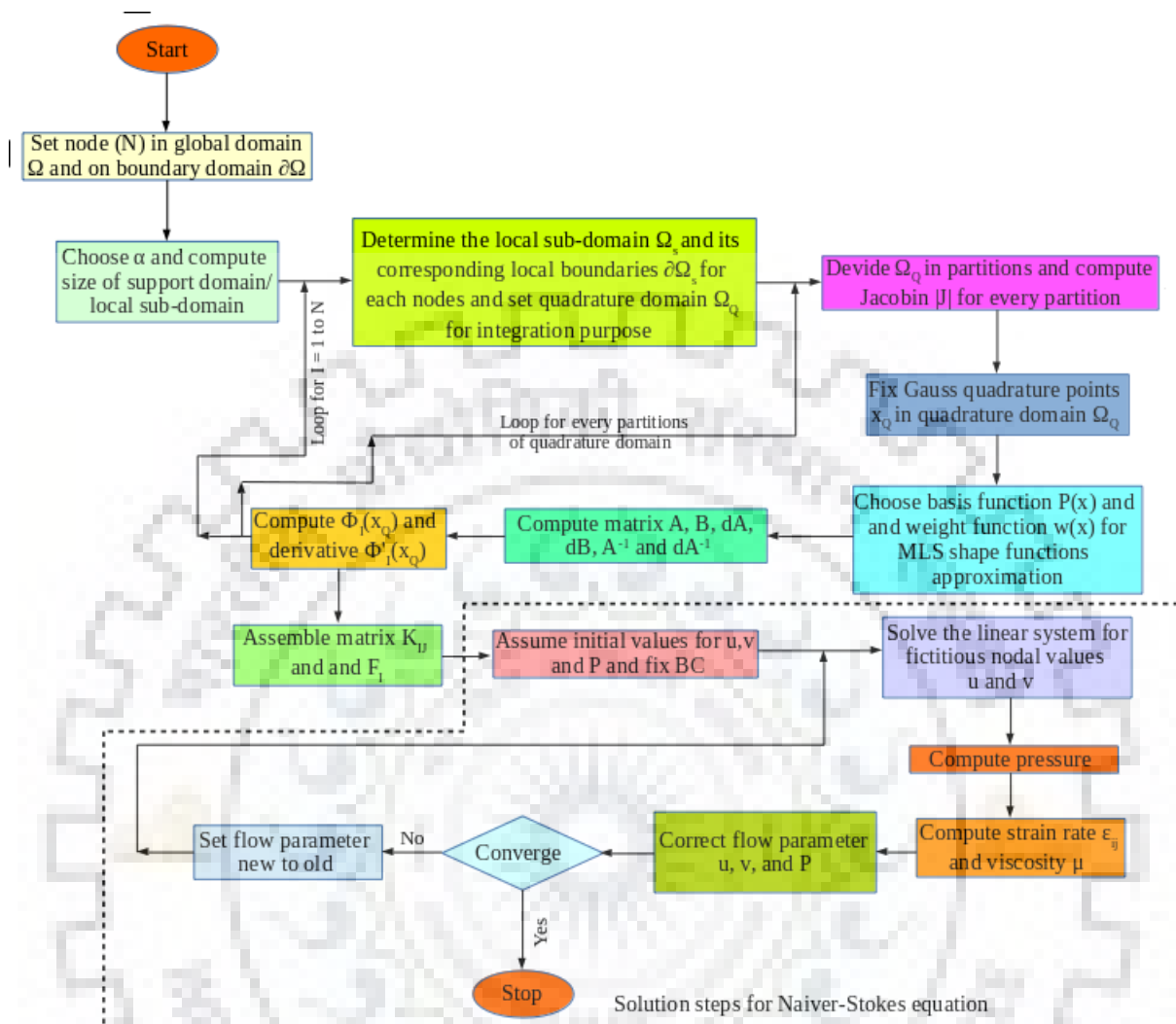


FIGURE 3.17: MLPG flow chart for any Navier-Stokes equations.

1. Steps 1 to 4 of MLPG algorithm for convection-diffusion problems.
2. Assume the initial values of u , v , P and fix the boundary conditions.
3. Make a loop over all nodes located inside the global domain.
 - (a) First solve the linear system for fictitious nodal value u and v .
 - (b) Compute pressure P
 - (c) Compute ε_{ij} and viscosity μ from calculated u and v
 - (d) Correction step for flow parameter, and set new value to old value till convergence.

These algorithms have been used in the present work to develop the in-house computational solver based on MLPG method in C++ programming language. The detailed validation and applications of in-house developed MLPG solver has been presented in the subsequent chapters. Prior to moving ahead, dimensionless parameters used herein are briefly discussed in next section.

3.7 Dimensionless parameters used

In the present work, various dimensional parameters have been used to characterize the flow and thermal behavior of the hydrodynamics problems. Prior to moving forward to discuss the validation of present MLPG solver, these dimensionless parameters are briefly defined below.

3.7.1 Reynolds Number (Re):

The Reynolds number is generally defined as follow.

$$Re = \frac{\text{Inertial force}}{\text{Viscous force}} = \frac{\rho U D}{\mu} \quad (3.79)$$

where, D , U , ρ and μ are the characteristic length, characteristic velocity and density and viscosity of fluid, respectively. Based on the values of Reynolds number, the flow behavior can be categorized as laminar and turbulent flows. In case of laminar flow, viscous forces dominate over the inertial ones. They are also characterized by smooth constant fluid motion. In case of turbulent flow, inertial forces dominate over the viscous forces and the flow streams tend to develop eddies, vortices and other flow instabilities.

3.7.2 Prandtl Number (Pr):

The Prandtl Number is defined as

$$Pr = \frac{\text{Momentum diffusivity}}{\text{thermal diffusion}} = \frac{\nu}{\alpha} = \frac{\mu/\rho}{k/c_p\rho} = \frac{c_p\mu}{k} \quad (3.80)$$

where μ , α , c_p , ρ , $\nu = (\frac{\mu}{\rho})$ and k are considered as dynamic viscosity, thermal diffusivity, heat capacity, density, momentum diffusivity or kinematic viscosity, and thermal conductivity. The thermal diffusivity dominates at $Pr \ll 1$, but in reverse when $Pr \gg 1$, the momentum diffusivity dominates the behavior.

3.7.3 Peclet Number (Pe):

It is used to study transport phenomena in a continuum and it is defined as.

$$Pe = \frac{\text{Advective transport}}{\text{Diffusive transport}} = RePr = \frac{LU}{\alpha} \quad (3.81)$$

where L , U , α , is the characteristic length, local flow velocity, thermal diffusivity, which is written as $\alpha = \frac{k}{\rho C_p}$, where k , ρ and c_p is the thermal conductivity, density and heat capacity.



Chapter 4

Application of MLPG solver in diffusion and convection-diffusion problems

MLPG methods have been successfully used to obtain the numerical solutions of different engineering problems governed by various kind of PDE, such as linear and nonlinear PDE, Navier-Stokes, Poisson and Laplace equations, etc. This chapter presents the validation of our in-house developed solver based on MLPG method with standard benchmark CFD problems. In particular, the simplest benchmark problem using diffusion equation constrained to various types of boundary conditions has first been validated to ascertain the accuracy and applicability of the present in-house solver. The essential boundary condition is enforced by the direct interpolation method, and MLS approximation is used as interpolation scheme. For all problems considered herein, quartic spline weight function has been chosen to select the nearest weighting nodes. Gaussian quadrature scheme is used for quadrature domain integration with suitable Gauss points selection.

Over the decades, various steady and transient heat conduction problems are well stated as benchmark problems by the previous researchers, e.g., heat conduction in an isotropic and inhomogeneous media (Sladeka et al., 2004; Wang et al., 2005; Ahmadi et al., 2010),

multi-dimensional transient heat conduction phenomena (Qian and Batra, 2005; Sladeka et al., 2006, 2008), steady-state heat conduction in 2-D irregular complex domain (Hong and Quan, 2008), and nonlinear heat conduction phenomena (Thakur et al., 2009; Tian and Rao, 2012). Various kinds of weight functions are used (Singh, 2004) such as cubic spline, quartic spline, Gaussian function, hyperbolic function, exponential, rational, and cosine type of functions to compute the shape function approximations in heat conduction phenomena. Abbasbandy and Shirzadi (2010) have presented new approach based on MLPG and collocation methods to treat the parabolic PDEs with non-classical boundary conditions and Chen and Liew (2011) used MLPG approach based on the moving Kriging interpolation for solving PDEs for heat flow in 2- and 3-D spaces.

The present solver has been validated by comparing the numerical solution for temperature distribution in steady-state heat conduction in 1-D and 2-D spaces with the corresponding analytical/experimental results available in the literature.

4.1 Solution of diffusion equation

Numerical solution of diffusion equation subjected to different BCs is obtained by using in-house MLPG solver and discussed in this section.

4.1.1 Steady state heat conduction

Consider a steady state heat conduction in an infinitely long uniform rod, whose both the ends are maintained at constant temperature (Figure 4.1). The length (L) of rod

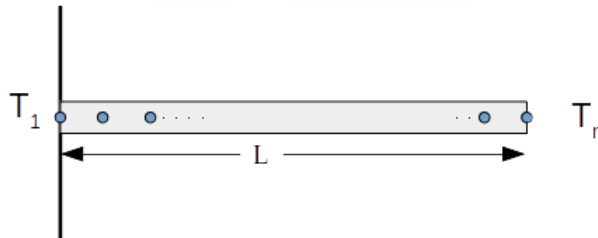


FIGURE 4.1: Heat conduction in a bar of length L .

is very large in comparison to both width (W) and height (H) of rod, i.e., $L \gg W$ and $L \gg H$. One of the boundary is exposed to higher temperature in comparison to the other boundary. Therefore, it can be considered as 1-D space heat transfer problem. In absence of all other phenomena like convection and radiation, the thermal conduction problem is governed by the following diffusion equation.

$$k_c \frac{d^2 T}{dx^2} = 0 \quad (4.1)$$

subjected to the essential BCs noted below.

$$T = \begin{cases} T_1 & \text{at } x = 0 \\ T_n & \text{at } x = L \end{cases} \quad (4.2)$$

where k_c and T are the thermal conductivity and temperature, respectively. The above equations (Eqs. 4.1 and 4.2) represent the simplest set of equations, whose analytical (and exact) solution can easily be obtained. The exact solution is a linear distribution between the specified boundary temperature computed as $T = T_n x + T_1$. It, therefore, can be considered as best choice to validate the newly developed computation flow physics solver.

The local weak form (LWF) of Eq. (4.1), with application of Gauss divergence theorem, can be written as

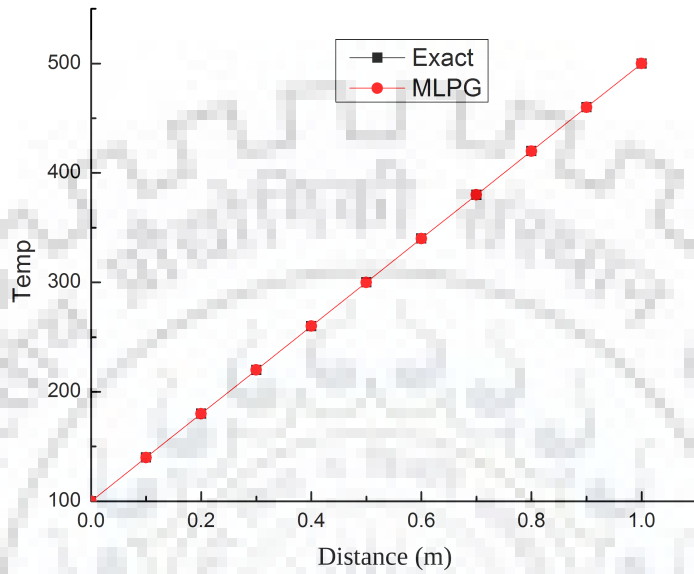
$$\int_{\Omega_s} \left(k_c \frac{dw}{dx} \frac{d\phi}{dx} \right) d\Omega - \int_{\Gamma_{su}} k_c w \frac{d\phi}{dx} \cdot n d\Gamma - \int_{\Gamma_{st}} k_c w \frac{d\phi}{dx} \cdot n d\Gamma - \int_{\Gamma_{sl}} k_c w \frac{d\phi}{dx} \cdot n d\Gamma = 0 \quad (4.3)$$

where Γ_{su} , Γ_{st} and Γ_{sl} belongs to essential, Neumann and internal surface BCs, as mentioned in Figure 3.12. If quadrature domains are completely inside the global boundary (Ω), then all the surface integral terms will vanish. The reduced form of Eq. (4.3) can than be expressed as below.

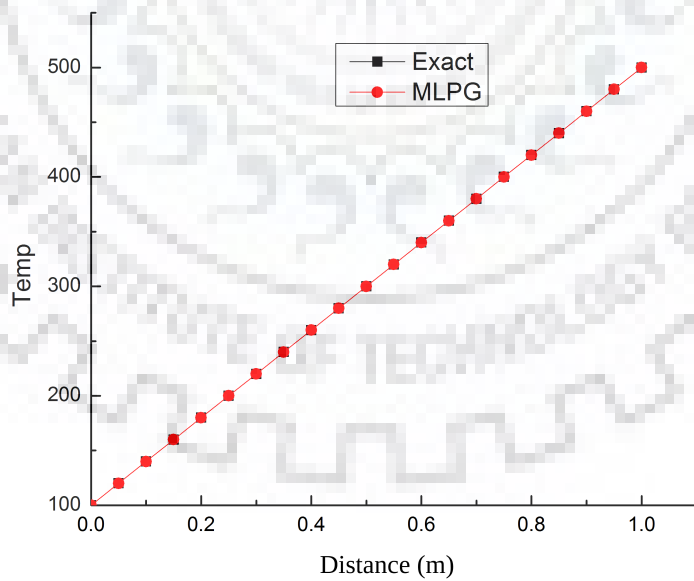
$$\int_{\Omega_s} \left(k_c \frac{dw}{dx} \frac{d\phi}{dx} \right) d\Omega = 0 \quad (4.4)$$

The complete equation (Eq. 4.3) will be used for those nodes which lie on the surface of global domain. The reduced form (Eq. 4.4) is used for internal nodes whose quadrature Ω_q does not intersect the global domain.

In-house MLPG solver has been used to solve this 1-D heat diffusion problem by using $L = 1$ for $n = 11$ and 22 nodes. The boundary values are randomly selected as $T_1 = 100$ and



(a) $n = 11$ nodes



(b) $n = 21$ nodes

FIGURE 4.2: Comparison of numerical results with analytical solution for $n = 11$ and 22 nodes.

$T_n = 500$. Both linear and quadratic basis functions, $p(X) = [1 \ x]$ and $p(X) = [1 \ x \ x^2]$,

are used for MLS shape function approximation. The quartic spline weight function is used for test domain. Four Gauss quadrature points are used in each quadrature sub-domain (Ω_q) for integration. The appropriate values of dimensionless parameter α_s and α_q are taken as 2.4 and 0.99, respectively. The convergence criteria for the solution variable is taken as 10^{-5} .

The nodal temperatures values calculated by the MLPG are compared in Figure 4.2 with the exact nodal temperature ($T = T_n x + T_1$) values for 11 and 22 nodes in the global domain. Both numerical and exact values are overlapping to each other. Qualitatively, numerical solution are as good as exact solution. Further attempts are made in Tables 4.1 and 4.2 to quantitatively analyse the relative deviation in the computed and exact values of nodal temperatures for two different basis functions. The percent relative deviation of the computed values with respect to the exact nodal values (Tables 4.1 and 4.2) is defined as

$$\epsilon_r = \sqrt{\frac{(T_{i,\text{cal}} - T_{i,\text{exact}})^2}{(T_{i,\text{exact}})^2}} \quad (4.5)$$

here $T_{i,\text{cal}}$ and $T_{i,\text{exact}}$ are the numerical (using MLPG solver) and exact values of temperature at i^{th} node, respectively. Table 4.1 shows that both exact and numerical

TABLE 4.1: Temperature distribution in a rod calculated for $n = 11$ nodes by using linear (Lin) and Quadratic (Quad) basis functions.

n	T_{exact}	$T_{\text{cal,Lin}}$	$T_{\text{cal,Quad}}$	$\% \epsilon_r$ (Lin)	$\% \epsilon_r$ (Quad)
1	100	100	100	0	0
2	140	139.999	140.023	7.1e-04	1.6e-02
3	180	179.999	180.035	5.5e-04	1.9e-02
4	220	219.999	220.064	4.5e-04	2.9e-02
5	260	259.999	260.101	3.8e-04	3.8e-02
6	300	299.999	300.102	3.3e-04	3.4e-02
7	340	339.999	340.094	2.9e-04	2.7e-02
8	380	379.999	380.099	2.6e-04	2.6e-02
9	420	419.999	420.090	2.3e-04	2.1e-02
10	460	459.999	460.051	2.1e-04	1.1e-02
11	500	500	500	0	0

values are very close ($\epsilon_r \approx 10^{-4}\%$) to each other for all the computing nodes (n). It

is further observed that the linear basis function yields better results than those with quadratic basis function for the problem under consideration. It indicates that the linear

TABLE 4.2: Temperature distribution in a rod calculated for $n = 21$ nodes by using linear (Lin) basis function.

n	T_{exact}	$T_{\text{cal,Lin}}$	$\% \epsilon_r$ (Lin)
1	100	100	0
2	120	119.9998	1.4e-04
3	140	139.9997	2.4e-04
5	180	179.9982	0.001
7	220	219.9976	0.001091
9	260	259.9972	0.00108
11	300	299.9972	0.000933
13	340	339.9974	0.00076
15	380	379.9979	0.0006
17	420	419.9985	0.00036
19	460	459.9992	0.00017
20	480	479.9996	8.33-005
21	500	500	0

basis function is more suitable than the quadratic basis function for the linear distribution of solution variable. In CFD studies, number of nodes also influences the computed solution. The influence of larger nodes ($n = 21$) on the numerical solution by using the linear basis function based in-house MLPG solver is investigated and reported in Table 4.2. As expected, the numerical solution has improved with increasing number of computational nodes. For example, ϵ_r at $x = 0.1$ has reduced from 7.1×10^{-4} to 1.4×10^{-4} with increasing nodes from 11 to 21, respectively. Both the exact and numerical values at each nodes are in very close agreement to each other. Such a close correspondance between the computed and exact values boosts us to further validate the present in-house MLPG solver with broader range of problems to ascertain its wider applicability. Subsequent section extends this 1-D heat conduction problem to little more realistic problem by including the heat source.

4.1.2 Steady state heat conduction with uniform heat generation

The in-house MLPG solver has been used to determine the solution of the previous problem (Section 4.1.1) in-conjunction with heat source. Consider a steady state heat transfer in an infinitely thick plate having uniform heat generation (q). The two ends of the plate is exposed to the isothermal conditions (Figure 4.3). Assuming that the dimensions in the

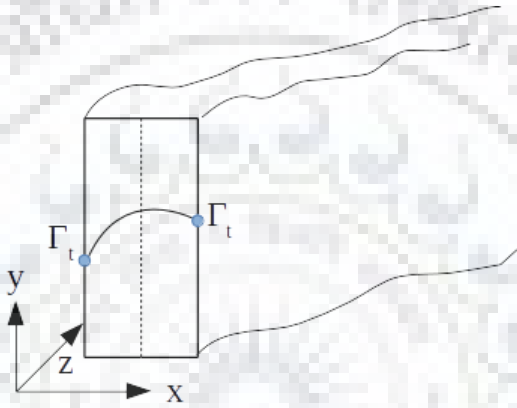


FIGURE 4.3: Heat conduction in large plate.

y - and z - directions are extremely large in comparison to dimension in x - direction, i.e., temperature gradients are significant in the x - direction only. The problem therefore can be considered as 1-D heat conduction with heating source. The governing equation can than be written as

$$k_c \frac{d}{dx} \left(\frac{dT}{dx} \right) + q = 0 \quad (4.6)$$

The essential BCs are

$$T = \begin{cases} T_1 & \text{at } x = 0 \\ T_n & \text{at } x = L \end{cases} \quad (4.7)$$

The exact solution of the above set of equations (Eqs. 4.6 and 4.7) is a quadratic distribution according to the Eqs. (4.8) between the specified boundary temperature as following (Versteeg and Malalasekera, 2007)

$$T = T_1 + \left[\frac{T_n - T_1}{L} + \frac{q}{2k_c}(L - x) \right] x \quad (4.8)$$

Eq. (4.6) in local weak form becomes,

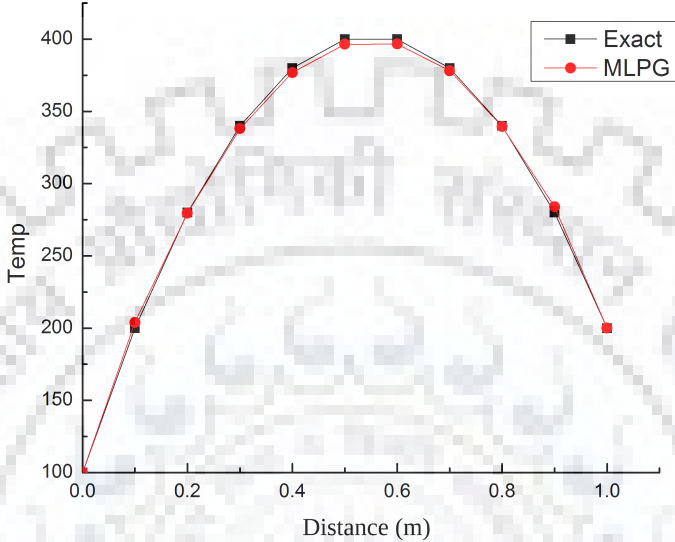
$$\int_{\Omega_s} \left(k_c \frac{dw}{dx} \frac{d\phi}{dx} \right) d\Omega - \int_{\Gamma_s} k_c w \frac{d\phi}{dx} n d\Gamma + \int_{\Omega_s} q w d\Omega = 0 \quad (4.9)$$

The surface integrals term will vanish when quadrature domains are completely inside the global boundary. The Dirichlet BCs can be imposed by direct method. Eq. (4.9) then becomes

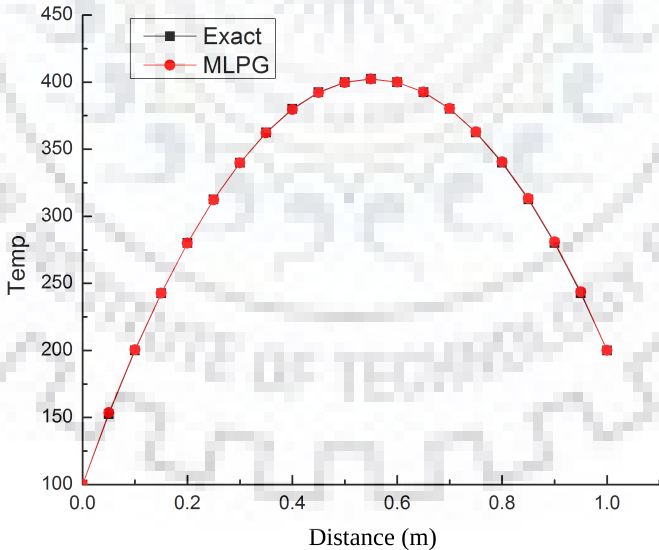
$$\int_{\Omega_s} \left(k_c \frac{dw}{dx} \frac{d\phi}{dx} \right) d\Omega + \int_{\Omega_s} q w d\Omega = 0 \quad (4.10)$$

Based on our experience with the previous problem, a linear basis function $p(\mathbf{x}) = [1 \ x]$ is used for MLS approximation and quartic spline weight function is used for shape function approximation. Direct interpolation method is used for essential BCs. The plate of unity length ($L = 1$) has been uniformly divided in two separate cases as $n = 11$ and $n = 21$ nodes, respectively. The optimized values of α_s and α_q are obtained as 2.295 and 0.98 to define the local sub-domain and quadrature domain for the best temperature distribution using $n = 11$ nodes. For $n = 21$ nodes, α_s and α_q are set as 2.2 and 0.99, respectively. The resulting matrix system is solved using the iterative procedure having the convergence criteria of 10^{-5} . The numerical results have been obtained for fixed thermal conductivity ($k_c = 0.5$ W/m.K) and uniform heat generation ($q = 1000$ kW/m³). The two ends of plate are maintained at $T_1 = 100$ K and $T_n = 200$ K, respectively.

Figure 4.4 compares the numerical results obtained using in-house MLPG solver with analytical solution for temperature distribution in a plate. Qualitatively, the numerical results are in close agreement with the exact solution of the considered problem. Quantification of closeness is analyzed in Tables 4.3 and 4.4 for 11 and 21 nodes, respectively. We can see from the Table 4.3 that both numerical and analytical results are close to each other for



(a) $n = 11$ nodes



(b) $n = 21$ nodes

FIGURE 4.4: Comparison of numerical MLPG results with analytical solution for temperature distribution in a plate.

TABLE 4.3: Temperature distribution in a plate calculated by using $n = 11$ nodes and linear (Lin) basis function.

n	T_{exact}	$T_{\text{cal,Lin}}$	$\% \epsilon_r$ (Lin)
1	100.00	100	0
2	200.00	203.971	1.985
3	280.00	279.598	0.143
4	340.00	338.094	0.560
5	380.00	376.873	0.822
6	400.00	396.527	0.868
7	400.00	396.884	0.779
8	380.00	378.112	0.496
9	340.00	339.582	0.122
10	280.00	283.944	1.408
11	200.00	200	0

TABLE 4.4: Temperature distribution in a plate calculated by using $n = 21$ nodes and linear (Lin) basis function.

n	T_{exact}	$T_{\text{cal,Lin}}$	$\% \epsilon_r$ (Lin)
1	100.000	100.000	0
3	200.000	200.442	0.221
5	280.000	279.989	0.004
7	340.000	339.753	0.073
9	380.000	379.662	0.089
11	400.000	399.740	0.065
13	400.000	399.956	0.011
15	380.000	380.265	0.070
17	340.000	340.618	0.182
19	280.000	280.887	0.317
21	200.000	200.000	0

all other nodes, except $n = 2$ and 10 . The possible reason behind it may be significantly higher, i.e., 2 to 4 folds, temperature in central region in comparison to BCs. Such larger gradients may not be correctly resolved due to small number of uniformly distributed computational nodes. The possible step may be taken to correct this erroneous solution by crowding of nodes in the global computational domain or in the local domain near the boundary surfaces. The number of uniformly distributed nodes in the global domain has therefore been increased to $n = 21$ in place of $n = 11$ to find out the improvement in the result on the questionable nodes of Table 4.3. The new results have been compared in analytical results in Table 4.4 for $n = 21$ nodes. To facilitate the comparison of values in Tables 4.3 and 4.4, alternate node values are listed in Table 4.4. We can observed that by increasing the number of nodes in computational domain, the numerical results are extremely close the analytical values. At distance $x = 0.1$, ϵ_r has reduced from 1.985 to 0.221 with increment in nodes from $n = 11$ to 21, respectively. Other corresponding error values have also improved. This validation further gives us confidence in our in-house MLPG solver.

4.1.3 Convective cooling of a circular fin

In-house MLPG solver has further been validated on the modified version of previous problem (see Section 4.1.2). Consider that a fin, of length (L) and uniform cross-sectional area (A), is exposed to the surrounding ambient (temperature T_∞). The base end of fin is maintained isothermally (temperature T_B) while the other end is insulated. The cooling of a fin is, therefore, achieved by means of convective heat transfer along the length (Figure 4.5). During this process, convective mode of heat transfer gives rise to heat source

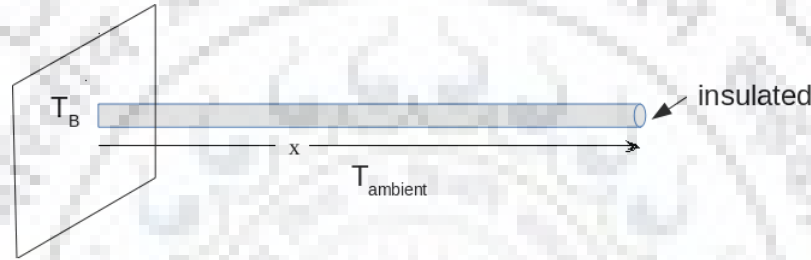


FIGURE 4.5: Convective cooling of a fin.

or sink, depending on the operating parameters. Under the above noted approximations, 1-D steady state heat transfer problem considered herein is governed by the following equation.

$$k_c A \frac{d^2 T}{dx^2} - hP(T - T_\infty) = 0 \quad (4.11)$$

where h , P and k_c are the convective heat transfer coefficient, perimeter of fin and the thermal conductivity of the fin material, respectively. The analytical solution of Eq. (4.11) (Versteeg and Malalasekera, 2007) is given by

$$\frac{T - T_\infty}{T_B - T_\infty} = \frac{\cosh[q(L - x)]}{\cosh(nL)} \quad \text{where} \quad q = \sqrt{\frac{hP}{k_c A}} \quad (4.12)$$

The discretize form of Eq. (4.11) can be written as follow.

$$\begin{aligned} \int_{\Omega_s} \left(\frac{dw}{dx} \frac{d\phi}{dx} \right) d\Omega - \int_{\Gamma_u} w \frac{d\phi}{dx} \cdot \eta d\Gamma - \int_{\Gamma_t} w \frac{d\phi}{dx} \cdot \eta d\Gamma \\ + \int_{\Omega_s} n^2 \phi w d\Omega = \int_{\Omega_s} n^2 T_\infty w d\Omega \end{aligned} \quad (4.13)$$

Since boundary node at the end of fin is insulated, surface term will not vanish in the above equation. The physically realistic BCs are given as follow.

$$T = T_B \quad \text{on } \Gamma_u \quad (\text{Dirichlet BC}) \quad (4.14)$$

$$\frac{dT}{dn} = t \quad \text{on } \Gamma_t \quad (\text{Neumann BC}) \quad (4.15)$$

The discretised form of the governing equation in conjunction of boundary conditions has been solved by using in-house MLPG solver. Similar to previous problem, a linear basis function $p(\mathbf{x}) = [1 \ x]$ is used for MLS approximation and quartic spline weight function is used for shape function approximation. The computational domain has been divided to 11 uniformly distributed nodes. Due to the lack of cronikar delta function property,

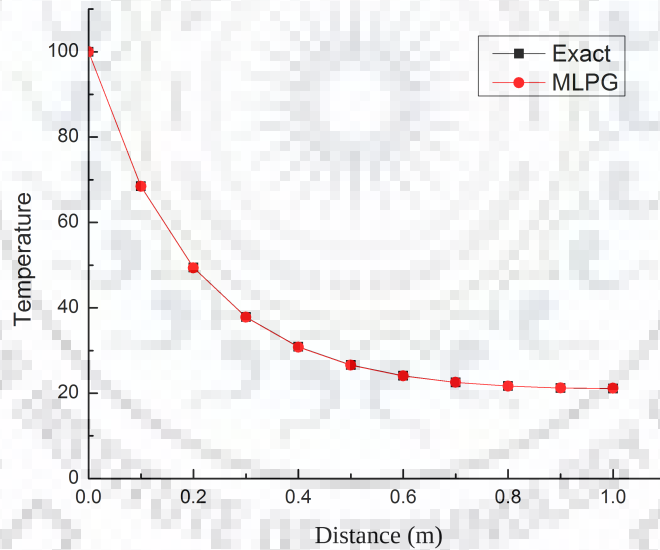


FIGURE 4.6: Temperature distribution along the length of fin.

the direct interpolation method is used to impose the essential BC. Each quadrature sub-domain Ω_q for integration has been populated with four Gauss quadrature points. The appropriate values of dimensionless parameters α_s and α_q are taken as 2 and 1.05 for best temperature distribution. The numerical and analytical results have been obtained for $L = 1$, $T_\infty = 20^\circ\text{C}$, $T_B = 100^\circ\text{C}$, $t = 0$ and $q^2 = 25$.

TABLE 4.5: Temperature distribution along the length of fin calculated by using $n = 11$ nodes and linear (Lin) basis function.

n	T_{exact}	$T_{\text{cal,Lin}}$	$\% \epsilon_r$ (Lin)
1	100	100	0
2	68.52	68.452	0.099
3	49.43	49.3622	0.137
4	37.86	37.7993	0.160
5	30.85	30.8011	0.158
6	26.61	26.5726	0.140
7	24.056	24.0292	0.111
8	22.54	22.5212	0.083
9	21.66	21.6589	0.005
10	21.21	21.2193	0.043
11	21.07	21.1394	0.329

The temperature is reduced, as expected, from base end to outer end of a fin. Qualitatively, both numerical and analytical temperature distributions are overlapping to each other (see Figure 4.6). Table 4.5 compares the MLPG solution with analytical solution (Eq. 4.12) to observe the quantitative performance of in-house MLPG solver. The insignificant deviations ($\epsilon_r < 0.5\%$) are obtained between the two results. The maximum deviation ($\epsilon_r \approx 0.33\%$) is seen to be at outer insulated end. It is certainly due the Neumann type of BC imposed at that end. This error can be minimize by increasing the number of nodes in the computational domain and, in particular, near the insulated end of fin.

In general, the in-house MLPG solver so far has produced accurate results of one-dimensional (1-D) conduction problems, with and without heat source/sink of fixed as well as variable magnitudes, subjected to both Dirichlet and Neumann type of boundary conditions. This reliable validation gives us confidence on present in-house solver to be tested on higher dimensional problems.

4.1.4 Heat conduction in a square slab

Consider the steady state heat conduction in an infinitely long square (side L) slab, as depicted in Figure 4.7. MLPG methodology used in deriving discretized equations in 1-D heat conduction case (see Section 4.1.1) can be easily extended to heat conduction in

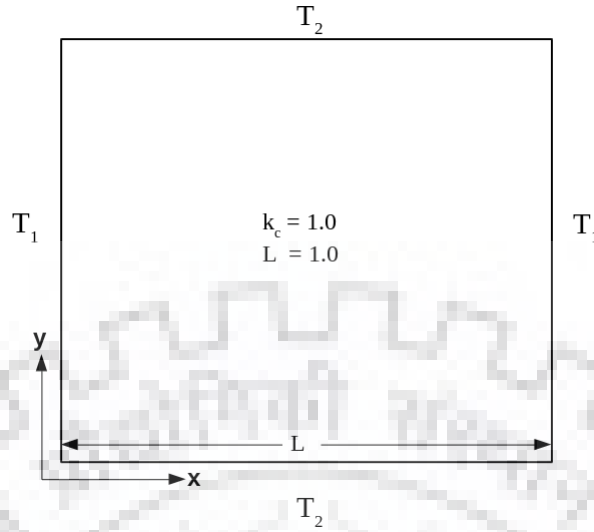


FIGURE 4.7: Square slab exposed to isothermal conditions.

two-dimensional square slab. In absence of any source/sink, the thermal field equation for the given problem, i.e., 2-D steady state heat conduction, is given by

$$k_c \frac{d^2 T}{dx^2} + k_c \frac{d^2 T}{dy^2} = 0 \quad (4.16)$$

The slab is subjected to isothermal conditions, as follow

$$T = \begin{cases} T_1 = 100 & \text{at } x = 0 \text{ and } L; 0 \leq y \leq L \\ T_2 = 25 & \text{at } y = 0 \text{ and } L; 0 \leq x \leq L \end{cases} \quad (4.17)$$

The discretized local weak form of PDE (Eq. 4.16) is written as follow.

$$\int_{\Omega_s} k_c \left(\frac{dw}{dx} \frac{d\phi}{dx} \right) d\Omega + \int_{\Omega_s} k_c \left(\frac{dw}{dy} \frac{d\phi}{dy} \right) d\Omega - \int_{\Gamma_u} w \left(\frac{d\phi}{dx} + \frac{d\phi}{dy} \right) \cdot \eta d\Gamma - \int_{\Gamma_t} w \left(\frac{d\phi}{dx} + \frac{d\phi}{dy} \right) \cdot \eta d\Gamma - \int_{\Gamma_l} w \left(\frac{d\phi}{dx} + \frac{d\phi}{dy} \right) \cdot \eta d\Gamma = 0 \quad (4.18)$$

It is subjected to the Dirichlet BCs $T_i = \hat{\phi}_i$ on Γ_u .

The temperature distribution in 2-D square slab has been investigated by using quartic spline weight function to calculate weightage of neighboring nodes. Quadratic basis function $p(\mathbf{x}) = [1 \ x \ y \ x^2 \ xy \ y^2]$ is used to calculate shape function using MLS approximation.

TABLE 4.6: Temperature distribution in square slab calculated by using $n = 25$ nodes.

n	T_{FLUENT}	T_{MLPG}	$\% \epsilon_r$
1	62.50	64.199	2.718
2	100.00	100.00	0
3	100.00	100.00	0
4	100.00	100.00	0
5	62.50	62.54	0.065
6	25.00	25.00	0
7	62.50	62.77	0.435
8	71.87	72.16	0.397
9	62.50	62.541	0.066
10	25.00	25.00	0
11	25.00	25.00	0
12	53.13	52.75	0.702
13	62.50	62.89	0.621
14	53.13	52.88	0.466
15	25.00	25.00	0
16	25.00	25.00	0
17	62.50	62.543	0.069
18	71.88	72.29	0.574
19	62.50	62.43	0.1069
20	25.00	25.00	0
21	62.50	62.54	0.0696
22	100.00	100.00	0
23	100.00	100.00	0
24	100.00	100.00	0
25	62.50	62.461	0.063

TABLE 4.7: Temperature distribution in square slab calculated by using $n = 121$ nodes.

n	T_{FLUENT}	T_{MLPG}	$\% \epsilon_r$
1	62.50	62.508	0.013
2	100.00	100.000	0.000
6	100.00	100.000	0.000
10	100.00	100.000	0.000
11	62.50	62.499	0.000
12	25.00	25.000	0.000
13	62.50	62.499	0.000
17	87.57	87.787	0.251
21	62.50	62.498	0.002
22	25.00	25.000	0.000
67	25.00	25.000	0.000
68	37.43	37.016	1.106
72	62.50	63.108	0.974
76	37.43	37.015	1.108
77	25.00	25.000	0.000
100	25.00	25.000	0.000
101	62.50	62.497	0.004
105	87.57	87.780	0.243
109	62.50	62.497	0.004
110	25.00	25.000	0.000
111	62.50	62.499	0.001
112	100.00	100.000	0.000
116	100.00	100.000	0.000
120	100.00	100.000	0.000
121	62.50	62.499	0.000

The computational domain of area $L^2 = 1$ has been populated with total 25 (i.e., 5×5) uniformly distributed nodes. The support and quadrature domains are set as $\alpha_s = 2.2$ and $\alpha_q = 0.95$. Four partitions have been considered for minimum error in quadrature domain integration such as volume and surface integrals. Each quadrature domain is populated with 4×4 Gauss points. The numerical results have been obtained by considering $k_c = 1.0$ with convergence criteria of $\delta = 10^{-5}$.

Table 4.6 lists the nodal temperature distributions calculated numerically by using MLPG method. To validate the present solver in 2-D space problems, the problem under consideration has also been solved by using a reliable commercial CFD solver (ANSYS FLUENT) for similar set of physical and geometrical parameters. The resulting temperature distribution (T_{FLUENT}) is also included in Table 4.6. Their relative deviations between the two results

(ϵ_r) have been calculated to analyze the validity of present in-house solver. Evidently, all nodal temperature values, except at one corner node 1, are extremely close ($\epsilon_r < 1\%$) to each other. The corner node ($n = 1$) shows largest deviation ($\epsilon_r \approx 2.7\%$), possibly due to the insufficient Gauss integration points. Its accuracy of the present results can further be enhanced by increasing the number of nodes in global domain and/or by increasing Gauss points for the evaluation of integral quantities. To observe this possibility of improvement in the numerical results, the computational domain has been uniformly divided into 121 (11×11) nodes. The calculated results using 121 nodes are reported in Table 4.7, where it can clearly be seen the near boundary values have been to the greater extent. However, central nodes are yielding little higher deviations, which is within the acceptable range. Based on the above validations with various problems, the in-house solver can be considered to working efficiently. Its validity is further being scrutinize in the subsequent sections.

4.1.5 Heat conduction in rectangular slab

The problem of steady heat conduction in square slab exposed to isothermal conditions (section 4.1.4) has been generalized herein as follows. Consider the steady state heat conduction in an infinitely long rectangular slab exposed to both isothermal and insulated conditions, as shown in Figure 4.8.

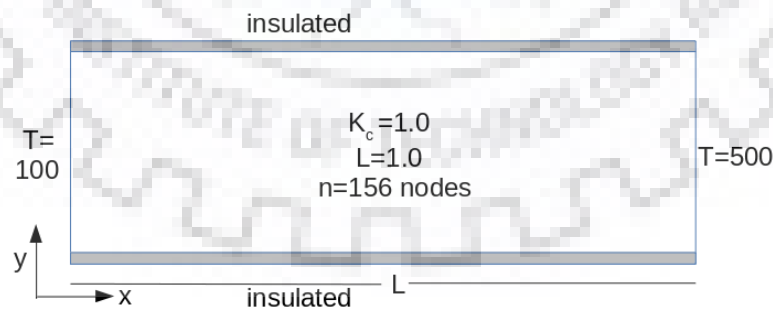


FIGURE 4.8: Rectangular slab exposed to both isothermal and insulated conditions.

The problem under consideration can mathematically be illustrated by the following 2-D steady state heat diffusion equation.

$$\frac{d}{dx} \left(k_c \frac{dT}{dx} \right) + \frac{d}{dy} \left(k_c \frac{dT}{dy} \right) = 0 \quad (4.19)$$

The discretized local weak form of PDE (Eq. 4.19) is written as follow.

$$\int_{\Omega_s} k_c \left(\frac{dw}{dx} \frac{d\phi}{dx} \right) d\Omega + \int_{\Omega_s} k_c \left(\frac{dw}{dy} \frac{d\phi}{dy} \right) d\Omega - \int_{\Gamma_u} w \left(\frac{d\phi}{dx} + \frac{d\phi}{dy} \right) \cdot \eta d\Gamma - \int_{\Gamma_t} w \left(\frac{d\phi}{dx} + \frac{d\phi}{dy} \right) \cdot \eta d\Gamma - \int_{\Gamma_l} w \left(\frac{d\phi}{dx} + \frac{d\phi}{dy} \right) \cdot \eta d\Gamma = 0 \quad (4.20)$$

The physically realistic BCs are given as follow.

$$\phi_i = \hat{\phi} \quad \text{on} \quad \Gamma_u \quad (\text{Dirichlet BC}) \quad (4.21)$$

$$\frac{d\phi_i}{dn_j} = \bar{t} \quad \text{on} \quad \Gamma_t \quad (\text{Neumann BC}) \quad (4.22)$$

where $\Gamma_s (= \Gamma_l \cup \Gamma_u \cup \Gamma_t)$ includes three parts of the local sub domain, i.e., Γ_u for the essential BCs, Γ_t for natural BCs and Γ_l for internal BCs of quadrature domain. When the quadrature domain is located entirely within the global domain, and the weight function is monotonically decreasing at the boundary of support domain equal to zero. If there is no BC specified on internal surfaces, then all the terms Γ_u , Γ_t and Γ_l vanish and for the boundary nodes Γ_l will also vanish.

The numerical solution using in-house MLPG solver is obtained by considering the computational domain as $x = [0, 5]$ and $y = [0, 1]$. The temperature is maintained as $T = 100$ and 500 at $x = 0$ and 5 , respectively. The horizontal boundaries ($y = 0$ and 1) are insulated and the thermal conductivity is taken as $k_c = 1$. The whole computational domain is populated with 256 uniformly distributed nodes. Each quadrature sub-domain Ω_q is considered as circular with radius $r_q = \Delta x \times 0.7$, in which $\Delta x = \Delta y$ is the nodal distance. The support sub-domain Ω_s is also chosen as a circle, having $\alpha_s = 2.3$ for both x and y directions. The rate and accuracy of convergence are related to the dimensionless size parameter, i.e., the support sub-domain Ω_s . The value $\alpha_q = 0.7$ was verified to be good by conducting a comparative analysis. A convergence criteria of $\delta = 10^{-5}$ is taken for the iterative process.

For the comparison and validation purpose of the present in-house MLPG solver, a finite volume based commercial CFD solver (ANSYS FLUENT) has also been used to simulate the problem under consideration for similar set of conditions. Figure 4.9 compares both

MLPG and FLUENT solutions for the nodal temperature distribution in a rectangular slab. Both results have shown good correspondence to each other within the acceptable order of deviations. The small differences observed in the nodal temperatures can further

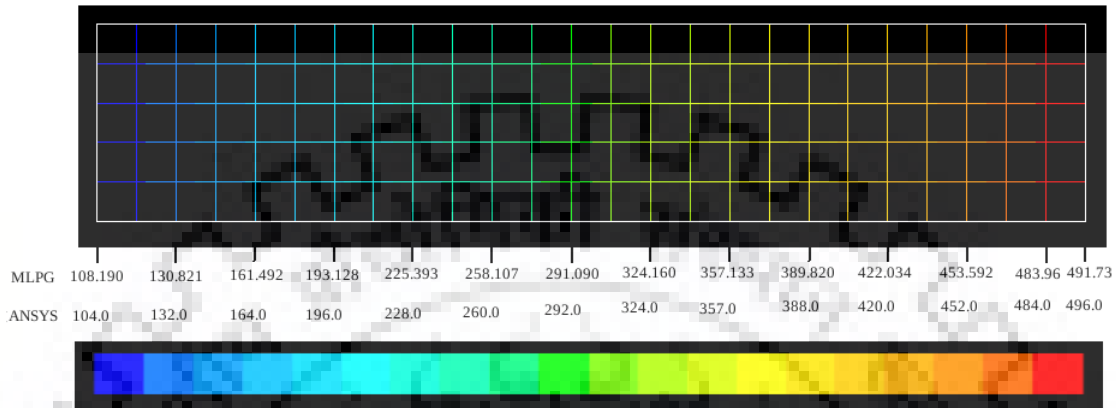


FIGURE 4.9: Temperature distribution in a rectangular plate

be reduced by increasing the number of nodes and the reducing/alternate the size of the sub-domains in MLPG method.

Based on the above applications of in-house MLPG solver for the solutions of 1-D and 2-D diffusion equations subjected to various types of BCs (Section 4.1.1 to 4.1.5), it can be concluded that promising in-house MLPG solver can efficiently be utilized in solving various engineering problems such heat conduction, mass diffusion controlled by the BCs of either or both Dirichlet and Neumann type. Having gained this confidence, further applications of the present solver in convection-diffusion is explored in subsequent section.

4.2 Solution of convection-diffusion equation

Numerical solution of convection-diffusion equation subjected to different BCs is obtained by using in-house MLPG solver and discussed in this section.

Traditional CFD methods, like FDM, FEM and FVM, have encountered many difficulties when dealing with some special problems such as large distortions, deformable boundaries, moving materials, and free surfaces, etc. To deal with such issues, MLPG numerical

technique has been used by [Lin and Atluri \(2000\)](#) to deal the basic convection dominated flow problems governed by Navier-Stokes equations at different Peclet numbers (Pe) and Reynolds number's (Re). They used a stabilization technique to deal the stabilization issues with the dominated convective terms at high Peclet numbers. Their results have shown good correspondence with the standard benchmarking problems. Meshless methods have been recorded by researchers as promising alternate tools in solving such special problems without appearance of difficulties faced by traditional techniques.

It is evident from the existing literature that the convection dominated flows are some of the most difficult problems to solve numerically. The non-linearities associated with convection term cause serious numerical difficulties which appear in the form of “wiggles” (oscillatory solutions). Various upwinding schemes have been proposed for traditional CFD techniques to overcome such instability problems. A review of such upwinding schemes can be found elsewhere ([Zienkiewicz and Taylor, 2000](#)). The exiting literature also suggests that optimal upwinding schemes may be designed to find the exact nodal solution in 1-D problems. However, in case of the multi-dimensional problems, generalizations of traditional upwinding schemes were unsuccessful due to the crosswind diffusion problem (see [Figure 4.10](#)). Prior to attempting the solution of convection dominated problems with

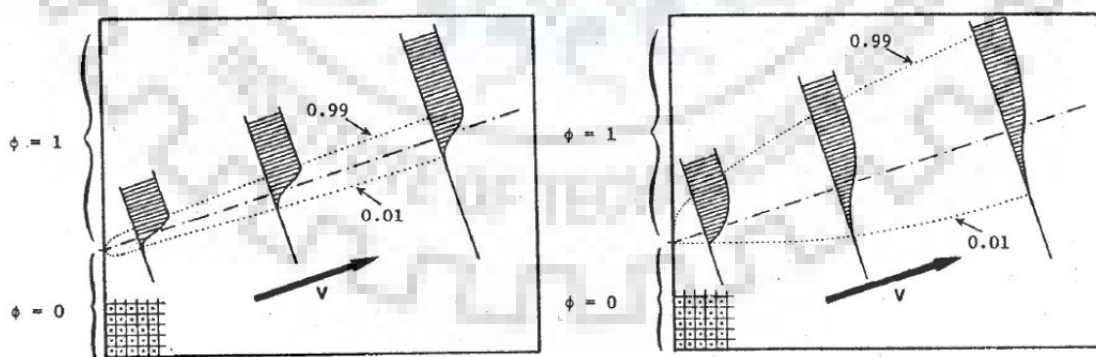


FIGURE 4.10: Illustration of the crosswind diffusion problem with $Pe = 100$ ([Leonard, 1979](#)). Exact (left) and optimal upwinding (right) solutions.

in-house MLPG solver, upwinding schemes proposed for meshless methods, in particular for MLPG methods, are briefed as follows.

4.2.1 Meshless upwind schemes

The streamline upwind (SU) method has first been introduced ([Hughes and Brooks, 1979](#)) to a-priori elimination of the possibility of any crosswind diffusion by constructing an artificial diffusion operator which acts only in the direction of flow. Most popular among such methods is the streamline upwind Petrov-Galerkin (SUPG) method ([Brooks and Hughes, 1982](#)) which consistently introduces an additional stability term in the upwind direction. While having some drawbacks, this method has better stability and accuracy properties than the standard Galerkin formulation for convection dominated flows. It is apparent that the upwind effects, arrived by whatever means, are significant only in the direction of flow.

In truly meshless methods, two upwinding schemes (US-I and US-II) for MLPG method were proposed ([Lin and Atluri, 2000](#)) by taking the similar kind of considerations to deal with convection dominated convection-diffusion problems at high Peclet numbers (Pe). The very general nature of truly meshless methods makes it easier to introduce the upwind concept more clearly and effectively. Amongst them, US-II is a very good scheme, not only because it always gives good solution, but also due to its very general nature. The desirable features of good upwinding scheme are (a) upwinding effect should be applied only in the streamline direction and (b) consistency for every term should be conserved. US-II also possesses these desirable features to be a good up-winding scheme ([Lin and Atluri, 2000](#)).

MLPG methods can easily handle the upwinding effects because they are based on a local weak form over a local sub-domain. MLPG method, in general, is based on Petrov-Galerkin weighting procedures. Different spaces for the test and trial functions can be used. One simple way to treat the upwinding effects is shifting of the local sub-domain opposite to the streamline direction. The same spaces for the trial and test functions are used, i.e., the same support and the same interpolation scheme (MLS) for the trial and test functions are employed. There is no need to modify the local weak form but have to shift the local

sub-domain when integration is implemented. Both US-I and US-II are briefly discussed herein for MLPG methods.

4.2.1.1 MLPG Upwinding Scheme I (US-I)

In MLPG method, different spaces for the test and trial functions can be used (see Figure 4.11). In order to apply upwinding in the stream line direction, skew the test

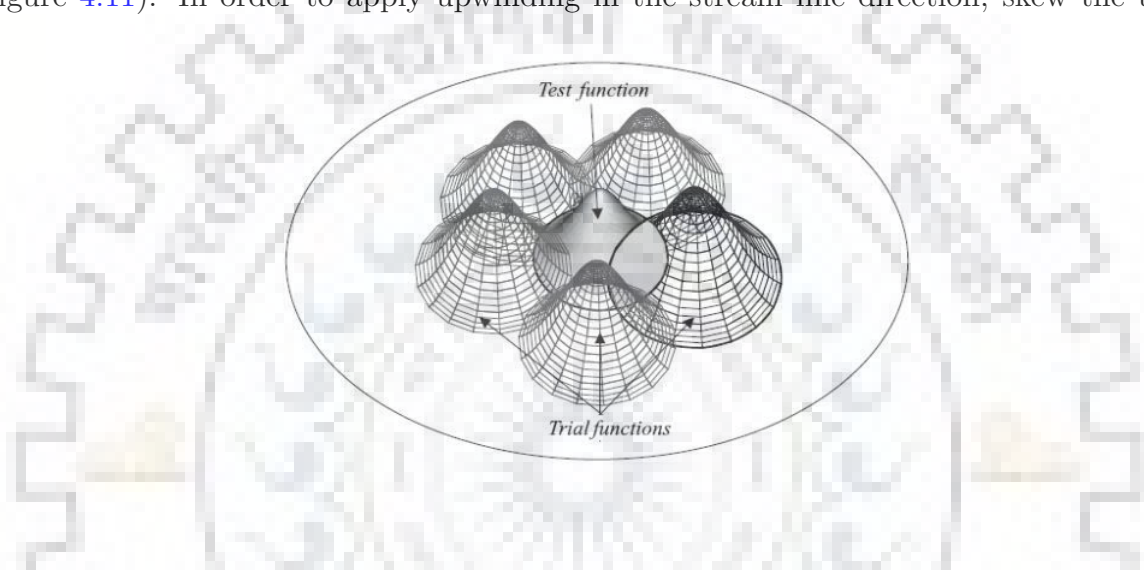


FIGURE 4.11: MLPG method without upwinding (Lin and Atluri, 2000).

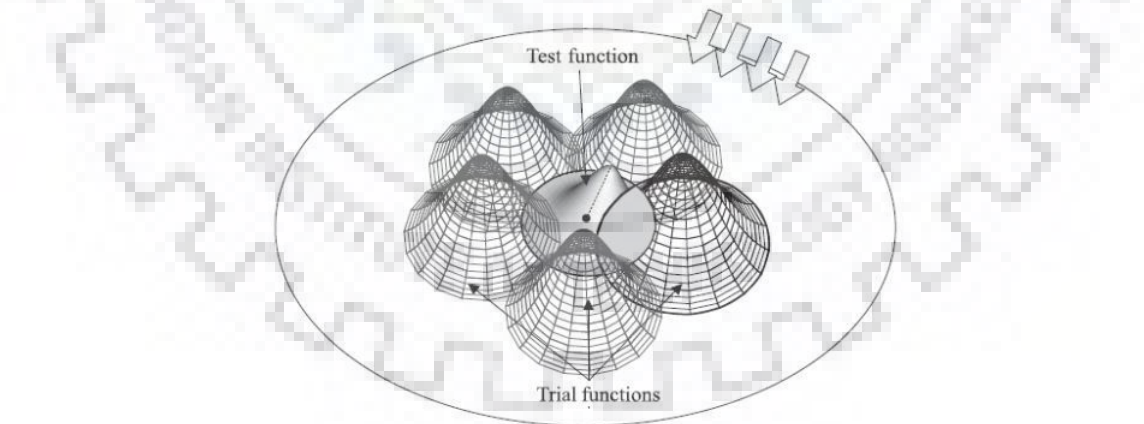


FIGURE 4.12: MLPG upwinding scheme I (US-I) (Lin and Atluri, 2000).

function opposite to the flow direction (Figure 4.12). It can be illustrated as follows: choose a skewed weight function as the test function. For the skewed weight function, the same weight function which is chosen for problem domain, shift the position of maximum

of w_i from x_i to $(x_i - \Delta)$, as shown in Figure 4.13. Here, $\Delta = (\gamma r_i) s_i$ is the shifting distance. Here, s_i and r_i are the unit vector of the streamline direction, and the radius of

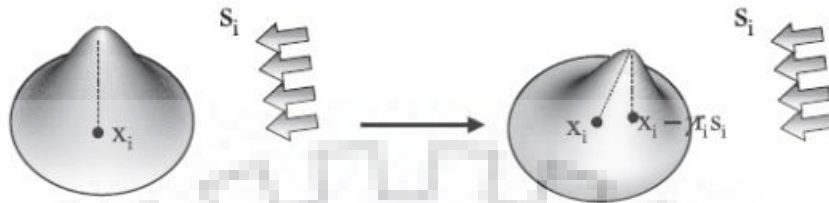


FIGURE 4.13: US-I illustration (Lin and Atluri, 2000).

the support for the test function, respectively, at x_i . The γ is given by

$$\gamma = \frac{1}{2} \coth\left(\frac{Pe}{2}\right) - \frac{1}{Pe} \quad \text{where} \quad Pe = \frac{ur_i}{K} \quad (4.23)$$

Here Pe is the local Peclet number. The size of the support for the trial function and local sub-domain at x_i is same as r_i and coincide with the support for the test function at x_i .

4.2.1.2 MLPG Upwinding Scheme II (US-II)

As MLPG methods are based on a local weak form over a local sub-domain, there is another very convenient way to design upwinding schemes. According to the Bubnov-Galerkin procedure to discretize the local weak form, shift the local sub-domain opposite to the streamline direction, as shown in Figure 4.14. The direction of the shifting is opposite to

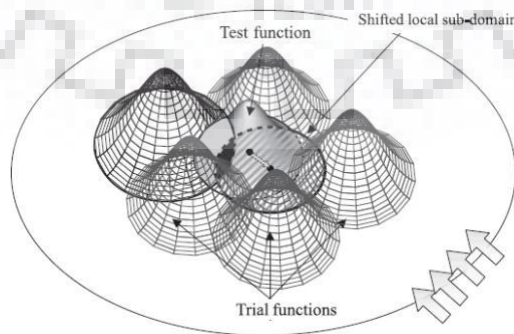


FIGURE 4.14: MLPG upwinding scheme II (US-II) (Lin and Atluri, 2000).

the streamline direction s_i at x_i (Figure 4.15). The shifting distance of the local sub-domain

can be specified as γr_i , where r_i is the size of the support for the test functions, which is equal to the size of the local domain at x_i and γ is given by

$$\gamma = \coth\left(\frac{Pe}{2}\right) - \frac{2}{Pe} \quad \text{where} \quad Pe = \frac{2ur_i}{K} \quad (4.24)$$

Here, same space for the test and trial functions are used. Also, same support and interpolation scheme (MLS) for the test and trial functions are employed. The local

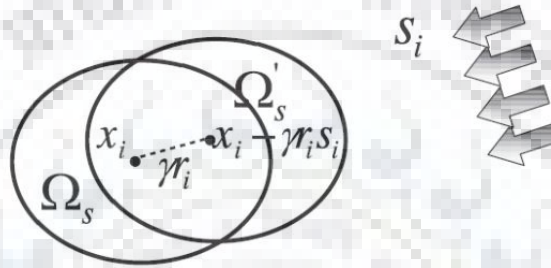


FIGURE 4.15: US-II illustration (Lin and Atluri, 2000).

sub-domain at x_i is no longer coinciding with the support for the test function at x_i , but the size is same. While MLPG chooses the test function such that the integration term along the boundary Γ_{sl} equals to zero, this is generally not true for the MLPG with US-II. Therefore, local weak form must include the integration term along the boundary Γ_{sl} .

The in-house MLPG solver has been equipped with both US-I and US-II. The subsequent sections validate the accuracy of the present solver with various convection-diffusion problems.

4.2.2 Steady state convection-diffusion

The steady state convection-diffusion problem in Ω is governed by the following equation.

$$v_j \frac{\partial \phi}{\partial x_j} = \frac{\partial}{\partial x_j} \left(K_d \frac{\partial \phi}{\partial x_j} \right) + f \quad (4.25)$$

where v_j is the flow velocity, K_d is the diffusivity coefficient, f is a source term. It is subjected to the following boundary conditions.

$$\phi = \hat{\phi} \quad \text{on} \quad \Gamma_u \quad (\text{Essential BC}) \quad (4.26)$$

$$K_d \frac{\partial \phi}{\partial x_j} n_j = \bar{t} \quad \text{on} \quad \Gamma_t \quad (\text{Natural BC}) \quad (4.27)$$

Here, $\hat{\phi}$ and \bar{t} are known values, n_j is the unit vector normal to Γ_l , and Γ_u and Γ_t are the subset of Γ_s . The local weighted residual form of Eq. (4.25) can be written as follows.

$$\int_{\Omega_s} \left[v_j \frac{\partial \phi}{\partial x_j} - \frac{\partial}{\partial x_j} \left(K_d \frac{\partial \phi}{\partial x_j} \right) - f \right] w \partial \Omega = 0 \quad (4.28)$$

for all w . By using integration by parts, the local weak form of above equation becomes as

$$\int_{\Omega_s} \left[v_j \frac{\partial \phi}{\partial x_j} + K_d \frac{\partial \phi}{\partial x_j} \frac{\partial w}{\partial x_j} - fw \right] \partial \Omega - \int_{\Gamma_s} K_d \frac{\partial \phi}{\partial x_j} n_j w \partial \Gamma = 0 \quad (4.29)$$

for all continuous test and trial functions (w and ϕ) in Ω . The boundary Γ_s of the local sub-domain Ω_s may intersect with the global domain Ω , therefore, $\Gamma_s = \Gamma_{sl} \cup \Gamma_{su} \cup \Gamma_{st}$. Here, Γ_{sl} is a part of Γ_s , which is inside the global domain, $\Gamma_{su} = \Gamma_s \cap \Gamma_u$, and $\Gamma_{st} = \Gamma_s \cap \Gamma_t$.

Now, consider the 1-D convection-diffusion flow problem (Figure 4.16) subjected to the essential BCs at both ends.

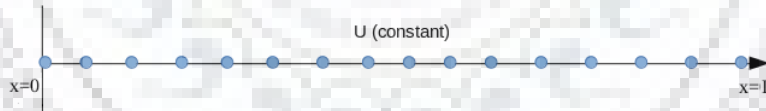


FIGURE 4.16: 1-D convection-diffusion flow domain of length L .

The governing equation (Eq. 4.25) for the considered case can be simplified as

$$u \frac{\partial \phi}{\partial x} - \frac{\partial}{\partial x} \left(K_d \frac{\partial \phi}{\partial x} \right) - f = 0 \quad (4.30)$$

The local Peclet number can be defined as $Pe = (uL)/K_d$, where L is the length of computational domain, u may be evaluated for given K_d for different Pe . The flow problem has been solved by using in-house MLPG, with and without upwinding, solver for the following two cases (Lin and Atluri, 2000):

- **Case 1:** $K_d = 1$, $f = 0$, $\phi_{x=0} = 0$ and $\phi_{x=L} = 1$
- **Case 2:** $K_d = 1$, $f = 100$, $\phi_{x=0} = 0$ and $\phi_{x=L} = 0$

The computational domain ($L = 1$) has been populated by $n = 11$ uniformly distributed nodes. The radius of support domains (r_i) is chosen as 2.3 times nodal distance ($2.3\Delta x$) to solve these problems. Quartic spline weight function is used for MLS shape function approximation. Direct interpolation method is used to impose the essential BCs.

Figure 4.17 depicts the numerical solutions of considered problems with Case-1 conditions by using MLPG, MLPG-US-I and MLPG-US-II methods for the wide range of Peclet number ($1 \leq Pe \leq 1000$). These results clearly show that MLPG is well accurate for low Pe but at high Pe oscillatory behavior appears in the solution. The upwind scheme is required at this stage to prevent the oscillations in the numerical solution. MLPG stabilized with both US-I and US-II produces very good results for overall range of Pe considered herein. While MLPG-US-I is very flexible, it is observed that results obtained from MLPG-US-II are superior in comparison with those obtained using MLPG-US-I.

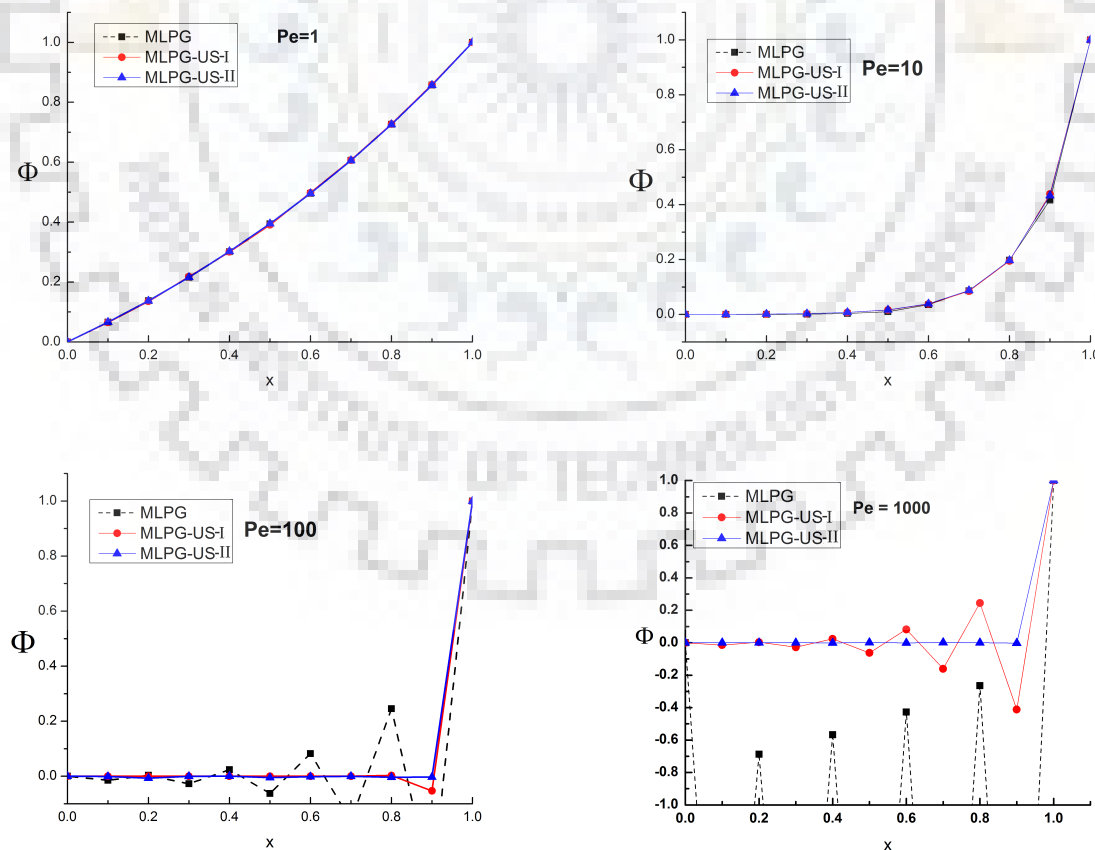


FIGURE 4.17: Results of Case 1 using MLPG, MLPG-US-I and MLPG-US-II for $1 \leq Pe \leq 1000$.

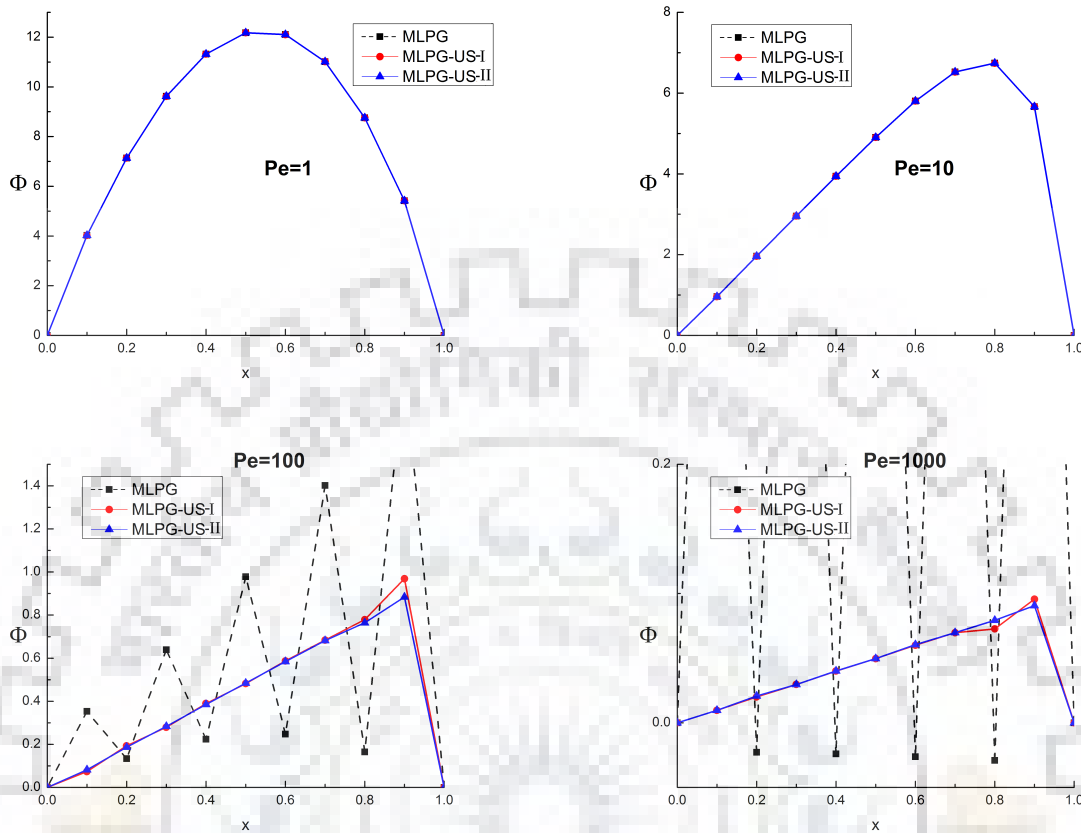


FIGURE 4.18: Results of Case 2 using MLPG, MLPG-US-I and MLPG-US-II for $1 \leq Pe \leq 1000$.

These observations are further strengthened with the help of case 2 conditions which is having finite source in addition to convection-diffusion process. The numerical results for case 2 are reported in Figure 4.18 using both MLPG and MLPG with upwinding methods. Similar to case predictions, the numerical schemes (MLPG, MLPG-US-I and MLPG-US-II) records stability and stabilization issues for case 2 at higher Pe . Both cases show that the MLPG-US-II is very successfully in dealing with convection-dominated flows. The size of support is observed plays a significant role in MLPG-US-II. Similar observations are also reported elsewhere (Lin and Atluri, 2000) for both cases considered herein.

4.2.3 Steady state convection-diffusion without source

Consider a property ϕ is transported by means of convection and diffusion through the one-dimensional space of length L . The problem is governed by Eq. (4.30) with source term

$f = 0$. The essential BCs are just reverse of case 1 of previous problem, i.e., $\phi_{x=0} = 1$ and $\phi_{x=L} = 0$. To validate their FVM solver, [Versteeg and Malalasekera \(2007\)](#) compared the distributions of $\phi(x)$ computed numerically and analytically by using FVM and Eq. (4.31), respectively, for $u = 0.1$ m/s (case A) and $u = 2.5$ m/s (case B).

$$\frac{\phi - \phi_0}{\phi_L - \phi_0} = \frac{\exp(\rho u x / K_d) - 1}{\exp(\rho u L / K_d) - 1} \quad (4.31)$$

An in-house MLPG solver has been used herein to find out the distribution of $\phi(x)$. The numerical solution has been compared with analytical solution to present the accuracy and acceptability of the present solver. The global computational domain of unit length ($L = 1$) has been populated with $n = 11$ uniformly distributed nodes. The quadratic $p(\mathbf{x}) = [1 \ x \ x^2]$ and quartic spline weight $w(x)$ functions are used for MLS shape function approximation for both the cases. A value of $\delta = 10^{-5}$ is used for the convergence of the iterative process. Four Gauss quadrature points are used in each quadrature sub-domain Ω_q . The appropriate optimized values of α_s are determined for both MLPG and MLPG-US-II methods, to match the results with exact solution. The value of $\alpha_s = 2.0$ and $\alpha_q = 0.64$ is used for MLPG-US-II.

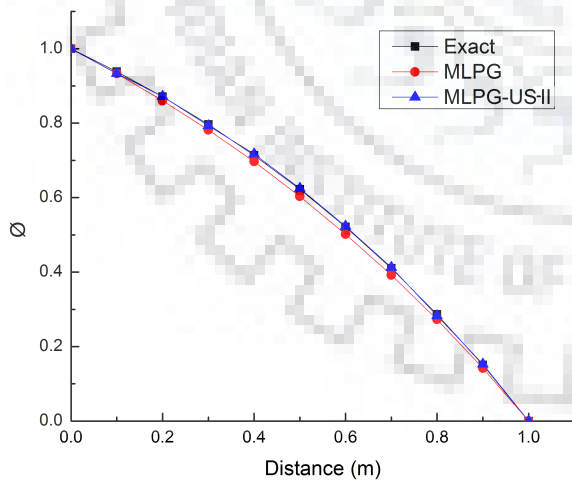


FIGURE 4.19: Distribution of $\phi(x)$ for case A.

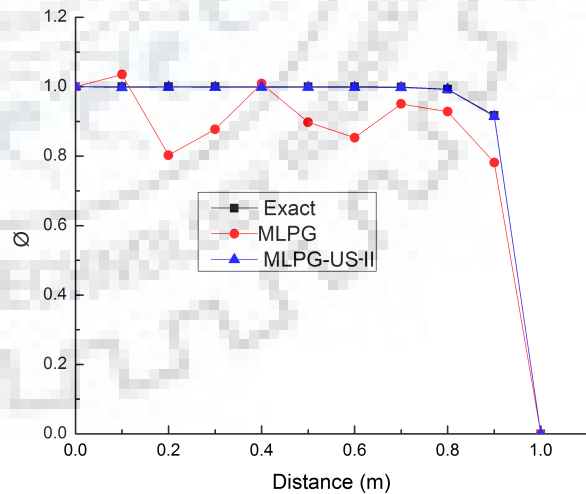


FIGURE 4.20: Distribution of $\phi(x)$ for case B.

Consider the density and thermal diffusivity as $\rho = 1.0$ kg/m³ and $K_d = 0.1$ m²/s, respectively. Figures 4.19 and 4.20 depicts the values of $\phi(x)$ computed using MLPG

and MLPG-US-II schemes for cases A and B, respectively. As seen earlier (section 4.2.2), MLPG results are qualitatively close to exact solution for case A but they are oscillating and far away from exact solution in case B. In both cases, MLPG-US-II results are, however, showing excellent correspondence with the exact solution.

Tables 4.8 and 4.9 show the numerical values of $\phi(x)$ for cases A and B, respectively, with their error quantification with results. The percent relative errors (ϵ_r) with respect to exact solution are also recorded in these tables. In both cases, the numerical results of

TABLE 4.8: Distribution of $\phi(x)$ for case A of convection-diffusion process using $n = 11$ computational nodes.

n	ϕ_{Exact}	ϕ_{MLPG}	$\epsilon_{r(\text{MLPG})}$	$\phi_{\text{MLPG-US-II}}$	$\epsilon_{r(\text{MLPG-US-II})}$
1	1.0000	1.0000	0.0000	1.0000	0.0000
2	0.9388	0.9332	0.5931	0.9339	0.5221
3	0.8711	0.8602	1.2576	0.8717	0.0662
4	0.7964	0.7819	1.8191	0.7930	0.4224
5	0.7138	0.6970	2.3554	0.7176	0.5308
6	0.6225	0.6035	3.0508	0.6247	0.3584
7	0.5215	0.5017	3.8031	0.5232	0.3194
8	0.4100	0.3922	4.3464	0.4129	0.6909
9	0.2868	0.2730	4.8102	0.2832	1.2488
10	0.1505	0.1420	5.6435	0.1528	1.5065
11	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE 4.9: Distribution of $\phi(x)$ for case B of convection-diffusion process using $n = 11$ computational nodes.

n	ϕ_{Exact}	ϕ_{MLPG}	$\epsilon_{r(\text{MLPG})}$	$\phi_{\text{MLPG-US-II}}$	$\epsilon_{r(\text{MLPG-US-II})}$
1	1.0000	1.0000	0.000	1.0000	0.0000
2	1.0000	1.0351	-3.510	0.9974	0.2612
3	1.0000	0.8026	19.738	0.9987	0.1261
4	1.0000	0.8776	12.239	0.9981	0.1913
5	1.0000	1.0090	-0.903	0.9984	0.1613
6	1.0000	0.8976	10.235	0.9982	0.1748
7	1.0000	0.8531	14.690	0.9983	0.1654
8	0.9994	0.9505	4.901	0.9978	0.1613
9	0.9933	0.9282	6.546	0.9922	0.1077
10	0.9179	0.7818	14.828	0.9144	0.3873
11	0.0000	0.0000	0.000	0.0000	0.0000

the given problems are poorly predicted by MLPG method, maximum deviations being $\epsilon_r \approx 5.6\%$ and 19.7% for cases A and B, respectively. The maximum deviations with

MLPG-US-II are $\epsilon_r \approx 1.5\%$ and 0.4% for cases A and B, respectively. The superiority of MLPG-US-II is, therefore, clearly reflected over MLPG scheme.

Due to dominant convection part, MLPG yields oscillatory behavior in the numerical solution of case B (see Figure 4.20 and Table 4.9). Upwinding scheme, as expected, have controlled the oscillations and yielded correct solution. It can, therefore, be safely concluded that both MLPG-US-I and MLPG-US-II are correctly implemented in the in-house MLPG solver. Based on the above validation of convection-diffusion problems, present solver with MLPG-US-II can accurately be used to solve the convection dominating flow problems, governed by Navier-Stokes equations, at large magnitudes of velocity.

Based on the confidence gained on the present in-house MLPG solver with the help of various diffusion and convection-diffusion problems subjected to varying set of boundary conditions in 1- and 2-D spaces, the next chapter attempts to verify the suitability of present solver for the flow problem coupled with pressure governing by Navier-Stokes equations.



Chapter 5

Application of MLPG solver in Navier-Stokes problems with Newtonian and non-Newtonian fluids

5.1 Introduction

This chapter explores the efficacy and suitability of the in-house MLPG solver for uniformly distributed nodes in the simulation of Navier-Stokes equations. In particular, the continuity and momentum equations in 1-D and 2-D spaces have been solved for the benchmarking problems. An adequate information is now available on most aspects of flow and heat transfer for Newtonian fluid flow using MLPG method. On the other hand, many industrial materials exhibit a range of non-Newtonian fluid behavior. For instance, most polymeric systems and slurry suspensions exhibit shear dependent viscosity, and thereby, display shear-thinning or shear-thickening, or both, under appropriate shear conditions. Despite of their wide occurrence in many industrial applications such as fiber reinforced resin processing, handling of pulp suspensions, fluidization of fibrous materials etc., none of the literature so far have used MLPG methods in computation of such non-Newtonian flows. Numerous investigations are reported in the literature on the flow of shear-thinning

and shear-thickening fluids, frequently modeled by the simple power-law model by using convention CFD methods, such as FDM, FEM, and FVM, e.g., non-Newtonian flow and applied rheology (Chhabra and Richardson, 2008), power law fluids flow across a circular and rectangular cylinders (Gupta et al., 2003; Bharti et al., 2006; Sahu et al., 2010), the cocurrent flow of shear thinning liquids and air in horizontal and vertical pipes (Chhabra et al., 1982), forced convection heat transfer from a sphere to non-Newtonian power law fluids (Dhole et al., 2006), laminar flow of power law fluids in concentric annuli (Gupta and Chhabra, 1996), etc. This chapter has also made the efforts to investigate non-Newtonian flow through a channel using in-house MLPG solver.

The main challenges in the development of a successful solver for non-Newtonian fluid rheology having shear dependent viscosity are: (a) effectively treatment of the nonlinearity of the dominant convection term at high Reynolds number and (b) treatment of incompressibility. In case of Newtonian fluids, the incompressible Navier-Stokes equations have been successfully solved by MLPG methods (Lin and Atluri, 2000, 2001; Wu et al., 2010; Sataprahma and Luadsong, 2014, etc.). None of the efforts are recording the applicability of MLPG in non-Newtonian fluid flow simulations. However, only effort was made by Shamekhi and Sadeghy (2009) using meshfree PIM method to investigate the non-Newtonian flow.

Shamekhi and Sadeghy (2009) have investigated lid-driven cavity flow of a purely viscous non-Newtonian fluid obeying Carreau-Yasuda rheological model by using the point interpolation method (PIM) meshfree method combined with the Characteristic-Based Split-A algorithm. They reported for the velocity and pressure profiles at Reynolds numbers $Re = 1000$.

Prior to the application of MLPG solver, the dimensionless form of 2-D flow governing equations in cartesian coordinates are introduced herein as follows.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (5.1)$$

$$\frac{\partial u}{\partial t} + \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = -\frac{\partial p}{\partial x} + \frac{\mu}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{2}{Re} \left(\varepsilon_{xx} \frac{\partial \mu}{\partial x} + \varepsilon_{yx} \frac{\partial \mu}{\partial y} \right) \quad (5.2)$$

$$\frac{\partial v}{\partial t} + \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = -\frac{\partial p}{\partial y} + \frac{\mu}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{2}{Re} \left(\varepsilon_{xy} \frac{\partial \mu}{\partial x} + \varepsilon_{yy} \frac{\partial \mu}{\partial y} \right) \quad (5.3)$$

Eq. (5.1) and Eqs. (5.2 - 5.3) are referred as continuity and momentum equations, respectively. All variables in Eqs. (5.1 - 5.3) are dimensionless. The length and velocity variables are made dimensionless by using the reference length (L_r) and reference velocity (V_r). Here u and v are the x - and y -components of the velocity, p and μ are the dimensionless pressure and viscosity, t is time. The Reynolds number (Re) and rate of strain tensor (ϵ_{ij}) are defined as follow.

$$Re = \frac{L_r V_r \rho}{\eta} \quad \text{and} \quad \epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial j} + \frac{\partial u_j}{\partial i} \right) \quad (i, j) \equiv (x, y) \quad (5.4)$$

where ρ is the density of fluid. The fluid viscosity (η) is related to shear stress and shear rate as follow:

$$\tau_{ij} = 2\eta\epsilon_{ij}$$

The rheology of the fluids is governed by the commonly used power-law rheological model (Bird et al., 2002; Chhabra and Richardson, 2008) as follow:

$$\mu = m(I_2/2)^{(n-1)/2} \quad \text{where} \quad I_2 = (1/2)(\epsilon : \epsilon) = (1/2) \sum_{i,j} \epsilon_{ij}^2 \quad (5.5)$$

All variables in Eq. (5.5) are dimensional. Here m and n are the flow consistency index and flow behavior index, respectively. The value of power-law index, $n < 1$, $n = 1$ and $n > 1$ represent the shear-thinning, Newtonian and shear-thickening fluids, respectively. I_2 is the second invariant of the rate of strain tensor whose components are available in the standard text book (Bird et al., 2002). The dimensionless viscosity (μ) is represented by the following relation:

$$\mu = (I_2/2)^{(n-1)/2}$$

Eqs. (5.1 - 5.5) are numerically solved by using MLPG solver for two flow problems in subsequent sections.

5.2 1-D fully developed flow of Newtonian and non-Newtonian fluids through a channel

Consider the steady fully developed flow of incompressible Newtonian and non-Newtonian fluids through a rectangular channel. The assumptions of the steady and fully developed nature simplifies the flow to be one-dimensional (1-D). Mathematically, it can be expressed as $\partial(u)/\partial x = 0 \Rightarrow u = u(y)$ and $v \approx 0$. Under these considerations, Eqs. (5.1 -5.3) can be simplified as follows.

$$\frac{\partial u}{\partial x} = 0 \quad \Rightarrow \quad u = u(y) \quad (5.6)$$

$$0 = -\frac{\partial p}{\partial x} + \frac{\mu}{Re} \left(\frac{\partial^2 u}{\partial y^2} \right) + \frac{2}{Re} \left(\varepsilon_{yx} \frac{\partial \mu}{\partial y} \right) \quad (5.7)$$

$$0 = -\frac{\partial p}{\partial y} \quad \Rightarrow \quad p = p(x) \quad (5.8)$$

All equation variables are dimensionless. Considering that the above dimensionless equations have used height of channel (H) and average flow velocity (u_{avg}) as reference length and velocity, respectively. Therefore, $0 \leq y \leq 1$ and $u_{avg} = 1$.

Evidently, Continuity equation (Eq. 5.6) itself satisfies. Therefore, Eq. (5.7) needs to be solved to determine the flow field. The following essential BCs are imposed to determine the flow field.

$$u_i = \bar{u}_i \quad \text{on} \quad \Gamma \quad (5.9)$$

By introduction of $\alpha = (\partial p/\partial x)$ and $u_p = (u/\alpha)$, Eq. (5.7) can be transformed as follow.

$$\frac{\mu}{Re} \left(\frac{\partial^2 u_p}{\partial y^2} \right) + \frac{2}{Re} \left(\varepsilon_{xx} \frac{\partial \mu_p}{\partial x} \right) - 1 = 0 \quad (5.10)$$

Here, μ_p is determined by using the modified velocity u_p . By using the definition of average velocity (u_{avg}), the pressure gradient term (α) can be determined as follow.

$$\alpha = \left[\int_0^1 u_p dy \right]^{-1} \quad (5.11)$$

To obtain local weak-forms of the momentum equation, Eq. (5.10) is weighted by test function w and integrated over local sub-domain Ω_s . The following integration system equation has been evolved.

$$\int_{\Omega_s} \left[\frac{\mu}{Re} \left(\frac{\partial^2 u_p}{\partial y^2} \right) + \frac{2}{Re} \left(\varepsilon_{xx} \frac{\partial \mu_p}{\partial x} \right) - 1 \right] w \partial \Omega = 0 \quad (5.12)$$

By using integration by parts, Eq. (5.12) can be rewritten as

$$\int_{\Omega_s} \frac{\mu}{Re} \left(\frac{\partial u_p}{\partial y} \frac{\partial w}{\partial y} \right) \partial \Omega - \int_{\Gamma_s} \frac{\mu}{Re} \frac{\partial u_p}{\partial y} w \cdot n \partial \Gamma - \frac{2}{Re} \int_{\Omega_s} \left(\varepsilon_{xx} \frac{\partial \mu}{\partial x} \right) w \partial \Omega + \int_{\Omega_s} w \partial \Omega = 0 \quad (5.13)$$

Using direct interpolation scheme for the essential BCs, Eq. (5.13) for internal nodes can be written in a system of algebraic equations as

$$\int_{\Omega_s} \frac{\mu}{Re} \left(\frac{\partial u_p}{\partial y} \frac{\partial w}{\partial y} \right) \partial \Omega = \frac{2}{Re} \int_{\Omega_s} \left(\varepsilon_{xx} \frac{\partial \mu}{\partial x} \right) w \partial \Omega - \int_{\Omega_s} w \partial \Omega \quad (5.14)$$

The above set of equations have been solved by using the following algorithm in MLPG solver for the given values of Re and n .

1. Assume u and u_p .
2. Calculate α by using the definition of average velocity.
3. Calculate μ , μ_p and ε_{xx} .
4. Determine u_p from Eq. (5.14).
5. Determine u from u_p .
6. Repeat steps 2 to 5 till convergence of u .

A linear basis function $p(\mathbf{x}) = [1 \quad x]$ with quartic spline weight function $w(x)$ is used for MLS shape function approximation in the in-house MLPG solver. Quadrature domain (Ω_q) integration has been performed by using 4 Gauss points in each part of Ω_q . For 1-D support sub-domain Ω_s , an appropriate value of α_s and α_q is selected from a numerical study using several trial experimentations. The local sub-domain and quadrature domain size parameters α_s and α_q are varying according to the power-law index value (n). Direct

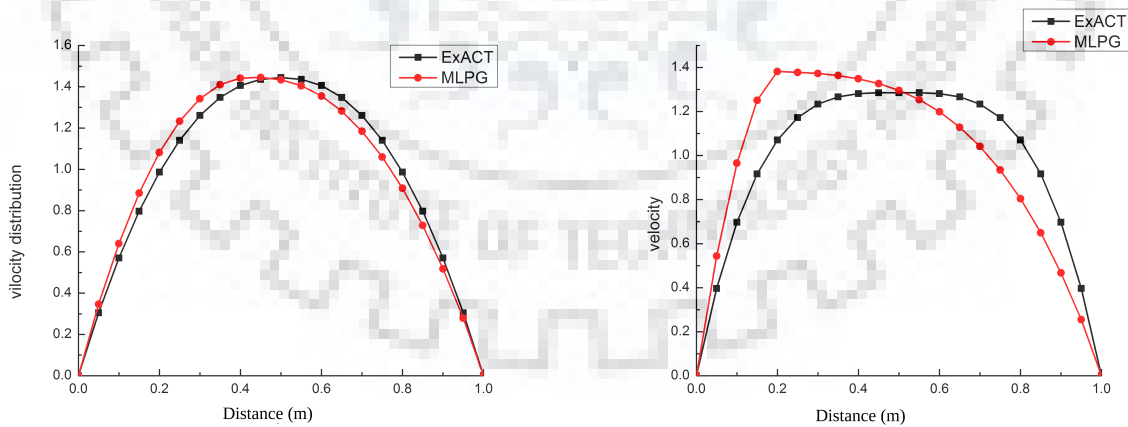
interpolation method is used to impose Dirichlet BCs $u = 0$ at the channel walls ($y = 0$ and 1). A convergence criteria of $\delta = 10^{-5}$ is used to obtain the computational results.

Extensive numerical results have been obtained by systematic variation of the value of the power-law index in the range as $0.4 \leq n \leq 1.8$ for a fixed value of Reynolds number ($Re = 1$) in the steady laminar flow regime. The computational domain has been uniformly discretized with $N = 21, 41$ and 51 nodes. The accuracy and applicability of the in-house MLPG solver has been assessed by comparing the numerical solution with analytical solution (Eq. 5.15) of the problem under consideration.

$$u(y) = u_{max} \left[1 - \left(\left| 1 - \frac{2y}{h} \right| \right)^{(n+1)/n} \right] \quad \text{and} \quad u_{max} = \frac{2n+1}{n+1} \quad (5.15)$$

where y and u_{max} are the dimensionless vertical distance ($0 \leq y \leq 1$) and dimensional maximum velocity, respectively. The dimensionless height of channel is unity ($h = 1$) in present case.

Figure 5.1 shows the comparisons of numerical and analytical results for two values of power-law index ($n = 0.8$ and 0.4) by using $N = 21$ computational nodes. The analytical



(a) Power-law index, $n = 0.8$.

(b) Power-law index, $n = 0.4$.

FIGURE 5.1: Velocity profiles for $n = 0.8$ and 0.4 by using 21 uniformly distributed computational nodes.

and numerical results for similar conditions are clearly not matching. It appears that the numerical velocity profiles are shifting to left side for selected values of power-law

index ($n < 1$) irrespective of sizes of the support and quadrature domains. It is known that the correctness of the velocity field depends the correctness of pressure field, i.e., pressure gradient in the present problem. It has been reported in the literature (Pierre, 1988; Franca and Frey, 1992, etc.) that conventional CFD approach like FEM also yields such erroneous results, even for Newtonian fluids ($n = 1$), due to the unstable pressure field evaluation. A perturbation term has been used (Pierre, 1988; Franca and Frey, 1992) to obtain the stabilized pressure field in FEM simulations.

The present MLPG approach also seems to require stabilization of pressure field by suitable means. The pressure gradient fluctuations to the neighboring nodes cause the maximum errors in solution. To overcome this situation, the pressure gradient term (α) herein is stabilized by using a correction factor (τ_p) as follow.

$$\alpha = \left[\int_0^1 u_p dy \right]^{-1} \times \tau_p \quad \text{where} \quad \tau_p = \frac{\beta}{r_s^2 \times n} \quad (5.16)$$

where r_s and n are the radius of support for the weight function, power-law fluid behavior index. The dimensionless parameter β depends on the number of computational nodes (N) and the power-law index (n). It reduces with increasing N and/or decreasing n , and/or both.

Figures 5.2 and 5.3 show the computed velocity profiles for shear-thinning ($0.2 \leq n \leq 1$) and shear-thickening ($1 \leq n \leq 1.8$) fluids, respectively, by using $N = 21$ computational nodes. These MLPG results have been obtained with the optimized value of dimensionless domain parameters α_s and α_q selected in between 2 – 2.1 and 0.9 – 1, respectively. These results have also been compared with the analytical results. It can clearly be seen that the introduction of correction factor (τ_p) improves the accuracy of MLPG solver in application of non-Newtonian fluid flow. The results shown are the best possible results with optimized value of β . For the given computational nodes ($N = 21$), the value of β is seen to increase ($0.5 < \beta < 4.5$) with increasing value of flow behavior index ($0.2 \leq n \leq 1.8$). It obviously suggests that the correction factor (τ_p) is one of possible solution to control the erroneous oscillation in the pressure field. A strong need of correction in the pressure field is also observed with the increasing tendencies of both shear-thinning and shear-thickening fluid

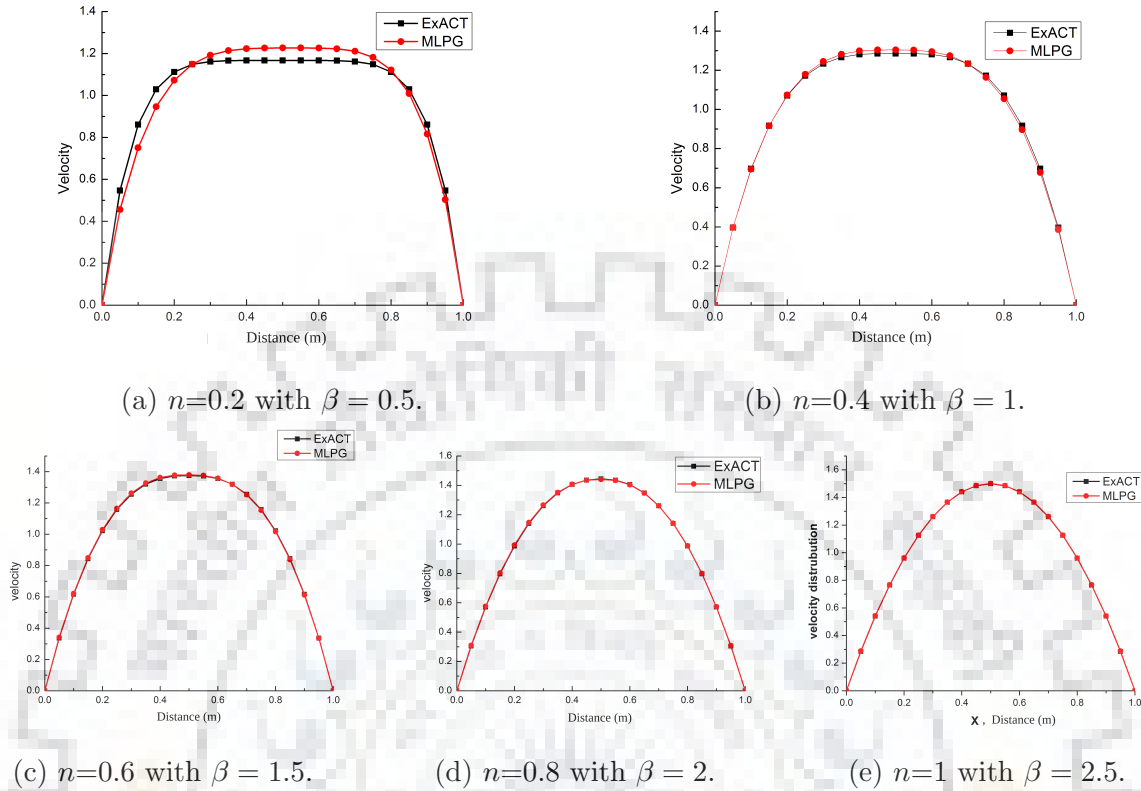


FIGURE 5.2: Velocity profiles for shear-thinning ($0.2 \leq n \leq 1$) fluids by using $N = 21$ computational nodes

behaviors. While MLPG results are well posed with analytical results for $0.4 \leq n \leq 1.6$, they are still showing larger extent of deviation at extremes of shear-thinning ($n < 0.4$, Figure 5.2) and shear-thickening ($n > 1.6$, Figure 5.3). These conditions requires resolution of sharper gradients during the computations. Due to the lesser N , distribution of the pressure gradient is highly influenced to the neighboring nodes. These deviations may, thus, be attributed to lesser number of computational nodes (N) in a given problem domain.

Figures 5.4 and 5.5 have explored the influence of nodal points (N) and requirement of the pressure correction factor (τ_p) in simulation of non-Newtonian fluid flow problem by using in-house MLPG solver. The values of α_s and α_q are selected in the range as 2.0 – 2.3 and 0.9 – 0.95, respectively, for both $N = 41$ and 51 computational nodes. The numerical results are observed to be very close to the analytical solutions for $N = 41$ nodes. The correspondence between the two results has further improved for $N = 51$ nodes. As expected, an increasing nodal points have reduced the value of β , under otherwise identical

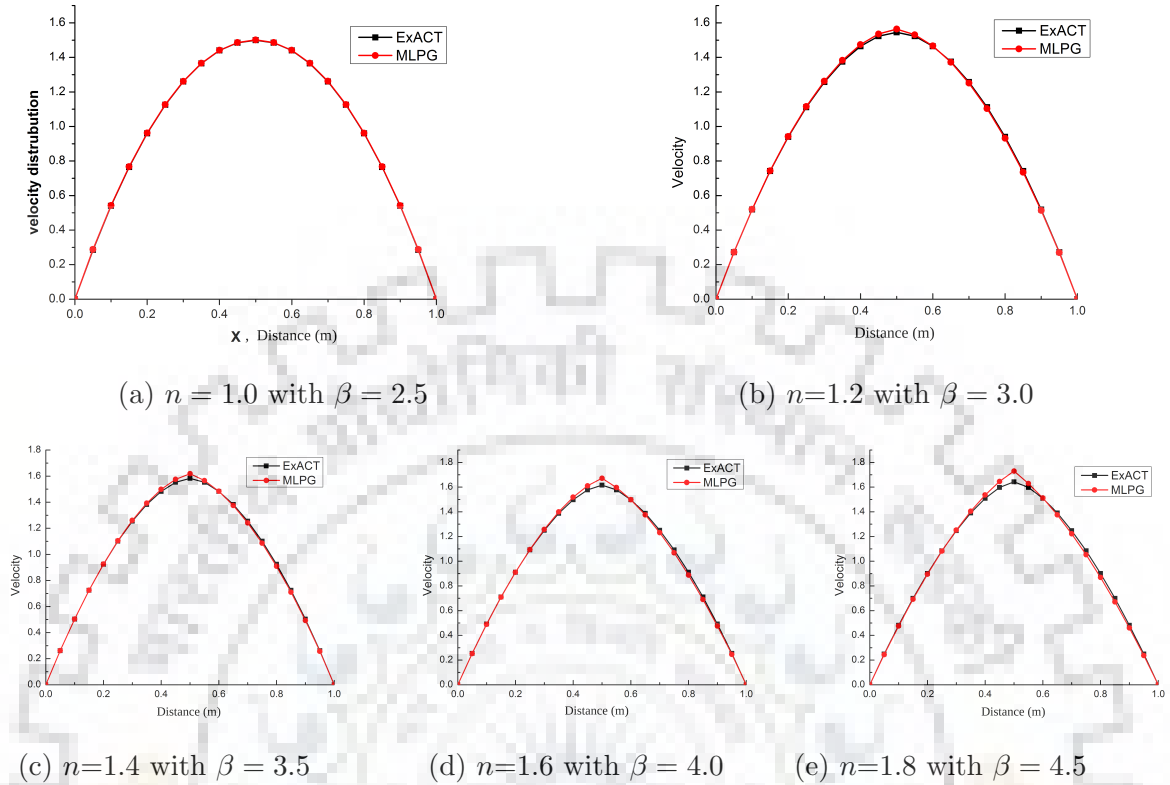


FIGURE 5.3: Velocity profiles for shear-thickening ($1 \leq n \leq 1.8$) fluids by using $N = 21$ computational nodes

flow conditions. For instance, β is seen to increase as $0.4 < \beta < 1.31$ and $0.16 < \beta < 0.64$ with increasing value of flow behavior index ($0.2 \leq n \leq 1.8$) for nodal points of $N = 41$ and 51 , respectively. Further, it is noted that the requirement of β diminish with increasing number of computational nodes (N) (see Table 5.1). With increasing number of nodes (N), β values have been reduced which suggests that the effect of pressure gradient to the neighboring nodes are minimized. The minimal fluctuations in pressure thereby yields correct velocity field for non-Newtonian fluid flow. It can, therefore, be concluded that while larger N enhances the computational cost, it is beneficial to obtain the accurate results with smooth convergence of MLPG solver.

Qualitative comparisons of the velocity fields computed by using MLPG solver with analytical solution (Figures 5.2 to 5.5) for power-law index ranging from shear-thinning to shear-thickening ($0.2 \leq n \leq 1.8$) suggests that the present in-house solver can successfully be utilized in simulation of non-Newtonian fluid flow. A representative quantitative

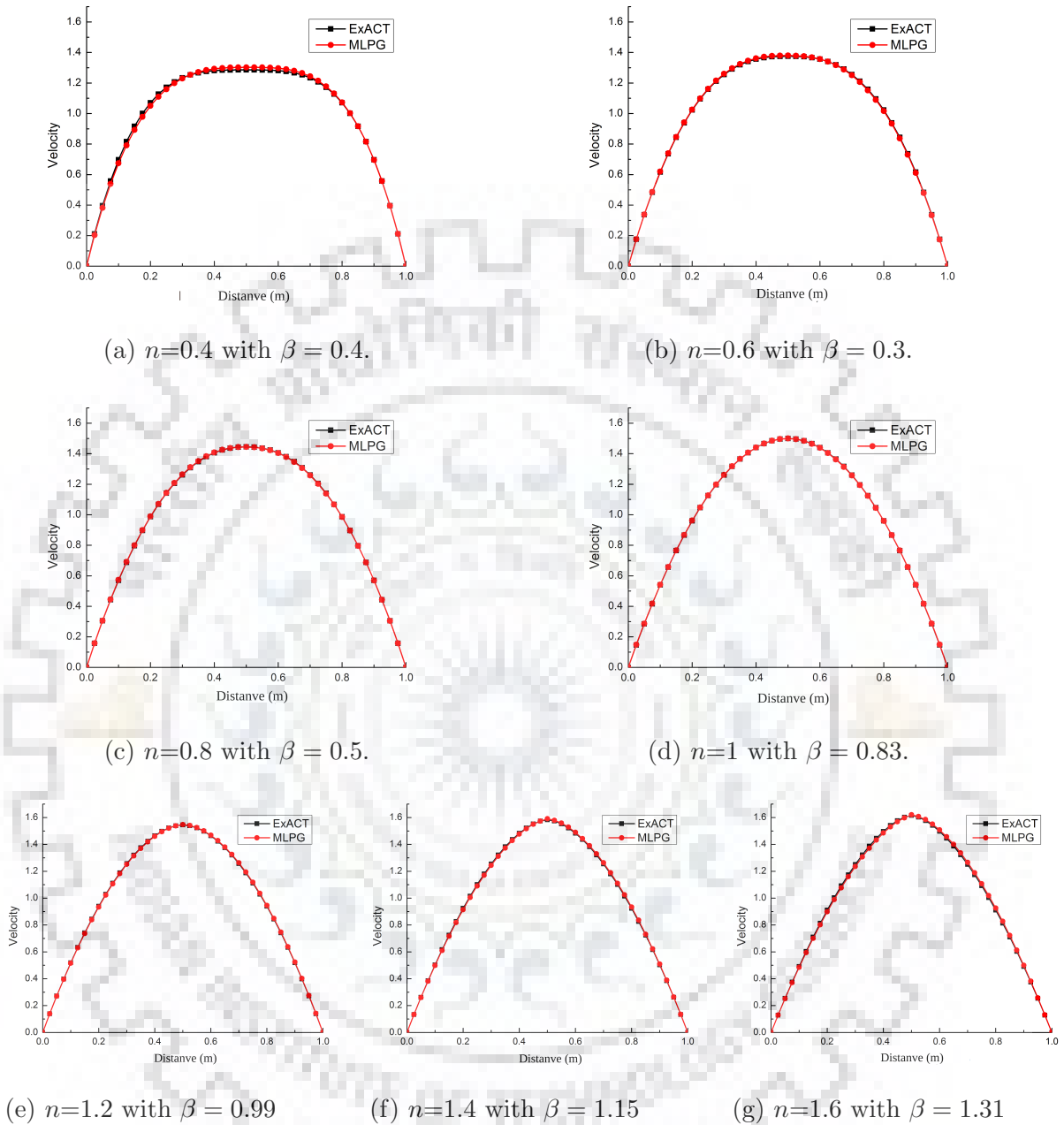


FIGURE 5.4: Velocity profiles for power-law fluids ($0.4 \leq n \leq 1.6$) fluids by using $N = 41$ computational nodes

comparison of the velocity, viscosity and shear-stress fields is also reported in Table 5.2 for power-law index of $n = 0.4$ and for 41 nodes in the computational domain. The numerical results are seen to well posed with analytical values within the acceptable degree of deviations. Table 5.2 shows core region have large deviations in comparison to near boundary region. Since the fluid viscosity is a function of shear rate, which is varying at each node and plug-like velocity profile of considered case, requires closer nodes for

TABLE 5.1: Optimized values of β for a range of power-law index (n) and computational nodes (N).

n	β		
	$N = 21$	$N = 41$	$N = 51$
0.2	0.5	-	-
0.4	1.0	0.40	0.16
0.6	1.5	0.30	0.24
0.8	2.0	0.50	0.32
1.0	2.5	0.83	0.40
1.2	3.0	0.99	0.48
1.4	3.5	1.15	0.56
1.6	4.0	1.31	0.64
1.8	4.5	-	-

TABLE 5.2: Velocity, viscosity and stress fields for shear-thinning ($n = 0.4$) fluid by using $N = 41$ nodes.

N	u_{Exact}	u_{MLPG}	$\% \epsilon_r$	μ_{Exact}	μ_{MLPG}	τ_{Exact}	τ_{MLPG}
1	0.000	0.000	0.000	0.289	0.295	7.931	7.646
2	0.211	0.204	3.473	0.289	0.295	7.931	7.646
3	0.396	0.382	3.589	0.313	0.319	6.929	6.690
4	0.557	0.538	3.453	0.341	0.346	6.007	5.841
6	0.815	0.791	2.961	0.411	0.413	4.396	4.361
8	1.001	0.977	2.318	0.509	0.501	3.077	3.157
10	1.127	1.110	1.477	0.653	0.625	2.029	2.185
12	1.207	1.198	0.674	0.882	0.812	1.232	1.413
14	1.253	1.254	0.083	1.282	1.117	0.660	0.830
16	1.275	1.284	0.685	2.109	1.683	0.288	0.419
18	1.284	1.298	1.088	4.424	2.949	0.083	0.164
20	1.285	1.302	1.282	17.943	7.858	0.008	0.032
22	1.285	1.302	1.307	17.943	15.742	0.008	0.010
24	1.284	1.300	1.270	4.424	3.876	0.083	0.104
26	1.275	1.290	1.146	2.109	1.925	0.288	0.335
28	1.253	1.264	0.931	1.282	1.208	0.660	0.729
30	1.207	1.214	0.634	0.882	0.846	1.232	1.319
32	1.127	1.131	0.384	0.653	0.642	2.029	2.089
34	1.001	1.003	0.166	0.509	0.504	3.077	3.124
36	0.815	0.816	0.004	0.411	0.409	4.396	4.420
38	0.557	0.557	0.052	0.341	0.341	6.007	6.002
39	0.396	0.395	0.179	0.313	0.313	6.929	6.904
40	0.211	0.212	0.457	0.288	0.288	7.930	7.916
41	0.000	0.000	0.000	0.288	0.288	7.930	7.916

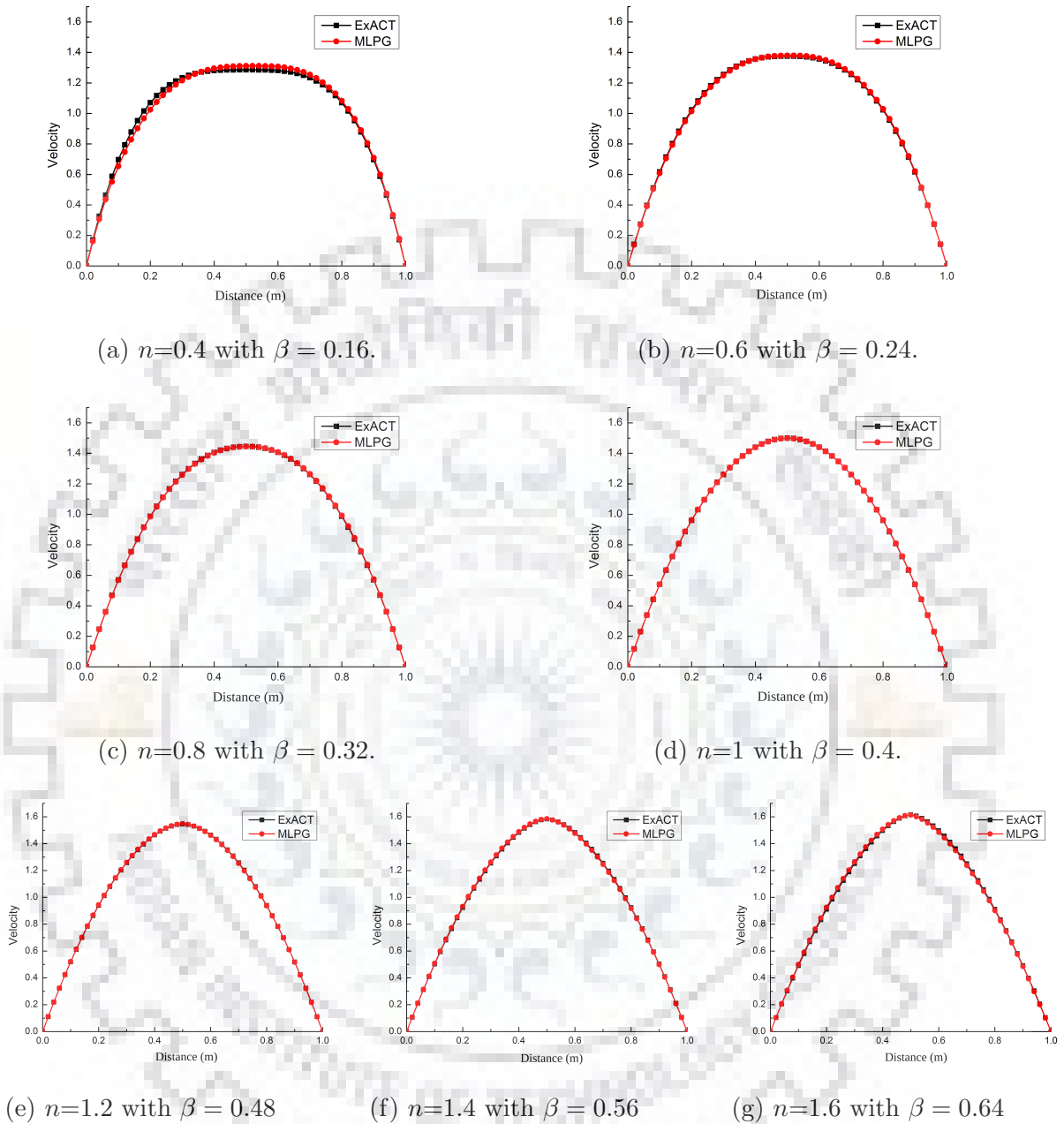


FIGURE 5.5: Velocity profiles for power-law fluids ($0.4 \leq n \leq 1.6$) fluids by using $N = 51$ computational nodes

correct resolution of shear rate. In addition to nodal distribution, the size of support and quadrature domains also plays an important role in MLPG methods.

Furthermore, pressure gradient as a function of power-law index has also been explored for two nodal distributions ($N = 41$ and 51) in Figure 5.6. The non-linear variation of pressure gradient with power-law index seems to smoothen with increasing N . While the

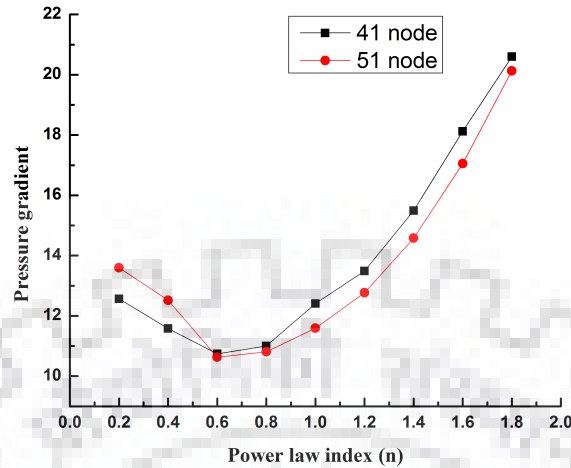


FIGURE 5.6: Pressure gradient as a function of power-law index ($0.2 \leq n \leq 1.8$) and computational nodes ($N= 41$ nad 51)

pressure gradient in shear-thickening ($n > 1$) fluids shows smooth variation with power-law index (n), shear-thinning ($n < 1$) fluids shows strong fluctuations. These fluctuations are certainly due to lower viscosity of shear-thinning fluids in comparison to shear-thickening fluids. Further refinement of the nodal distribution is needed to obtain the correct and smooth field.

Having gained the confidence on MLPG solver to simulate the non-Newtonian flows, subsequent section attempts to explore the efficacy of present in-house MLPG solver in 2-D Newtonian/non-Newtonian fluid flows.

5.3 2-D flow of Newtonian and non-Newtonian fluids through a channel

Consider the physical model of a two-dimensional (2-D), steady, laminar, incompressible non-Newtonian fluid in a channel (length L and width H) shown in Figure 5.7. The uniform flow stream is approaching through a sufficient long confined channel due to which it becomes fully developed at the exit. Both top and bottom walls of the channel are solid and impermeable. Since the flow field varies along the length and height of the channel,

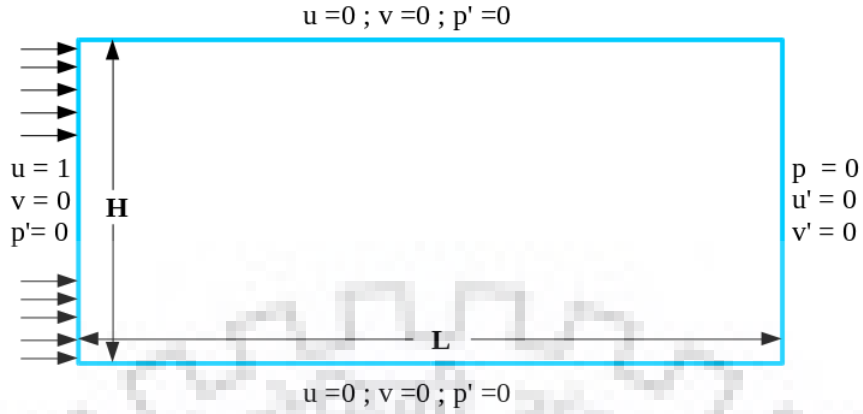


FIGURE 5.7: Flow through a channel.

the flow can be approximated as 2-D. Under the above noted approximations, the flow governing equations (Eqs. 5.1 - 5.3) can be simplified by eliminating the transient term $\partial(\)/\partial t = 0$. The conservative form of the simplified governing equations can be written as follows.

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (5.17)$$

$$u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\mu}{Re} \left(\frac{\partial^2 u_i}{\partial x_j^2} \right) - \frac{2}{Re} \left(\varepsilon_{ij} \frac{\partial \mu}{\partial x_j} \right) - f_i = 0 \quad (5.18)$$

where f_i represents for the extra body force term in i^{th} direction. The flow is subjected to the following Dirichlet and Neumann BCs

$$u_i = \bar{u}_i \quad \text{on} \quad \Gamma_u \quad (5.19)$$

$$\left[\frac{\mu}{Re} \left(\frac{\partial u_i}{\partial x_j} \right) n_j - p \right] n_j = \bar{t}_j \quad \text{on} \quad \Gamma_t \quad (5.20)$$

where \bar{u}_i and \bar{t}_i are fixed and n_j is outward unit vector normal at the surface. The local weak form of governing equations (Eqs. 5.17 and 5.18) is obtained by weighting them by a test function w_i and integrated over a local sub-domain Ω_s , as follows.

$$\int_{\Omega_s} \left(\frac{\partial u_i}{\partial x_i} \right) w \partial \Omega = 0. \quad (5.21)$$

$$\int_{\Omega_s} \left(u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\mu}{Re} \left(\frac{\partial^2 u_i}{\partial x_j^2} \right) - \frac{2}{Re} \left(\varepsilon_{ij} \frac{\partial \mu}{\partial x_j} \right) - f_i \right) w_i \partial \Omega = 0 \quad (5.22)$$

By using the integration by parts and applying the Gauss divergence theorem, Eq. (5.22) can be reformed in to local weak form for all the continuous test w and trial functions u_i as below.

$$\int_{\Omega_s} \left(u_j \frac{\partial u_i}{\partial x_j} w_i - p \frac{\partial w_i}{\partial x_i} + \frac{\mu}{Re} \left(\frac{\partial u_i}{\partial x_j} \frac{\partial w_i}{\partial x_j} \right) - \frac{2}{Re} \left(\varepsilon_{ij} \frac{\partial \mu_i}{\partial x_j} \right) w_i - f_i w_i \right) \partial \Omega - \int_{\Gamma_s} \left(\frac{\mu}{Re} \frac{\partial u_i}{\partial x_j} - p \right) w_i n_j \partial \Gamma = 0 \quad (5.23)$$

By imposing all BCs on a local weak form, Eq. (5.23) can be recasted as follow.

$$\int_{\Omega_s} \left(u_j \frac{\partial u_i}{\partial x_j} w_i - p \frac{\partial w_i}{\partial x_i} + \frac{\mu}{Re} \left(\frac{\partial u_i}{\partial x_j} \frac{\partial w_i}{\partial x_j} \right) - \frac{2}{Re} \left(\varepsilon_{ij} \frac{\partial \mu_i}{\partial x_j} \right) w_i - f_i w_i \right) \partial \Omega - \int_{\Gamma_{sl}} \left(\frac{\mu}{Re} \frac{\partial u_i}{\partial x_j} - p \right) w_i n_j \partial \Gamma - \int_{\Gamma_{su}} \left(\frac{\mu}{Re} \frac{\partial u_i}{\partial x_j} - p \right) w_i n_j \partial \Gamma - \int_{\Gamma_{st}} \left(\frac{\mu}{Re} \frac{\partial u_i}{\partial x_j} - p \right) w_i n_j \partial \Gamma = 0 \quad (5.24)$$

where, Γ_{sl} is the part of local sub-domain situated inside the global domain. As MLPG is based on a local weak form over a local sub domain in which the test and the trial functions come from the different space, the Bubnov-Galerkin procedure is used herein to discretize the local weak form (Lin and Atluri, 2000, 2001; Wu et al., 2010). It needs to compute the internal surface integral term Γ_{sl} of inner boundary of local sub domain Ω_s , which is inside the global domain Ω at boundary. MLPG methods usually choose the test function such that the integration term along the boundary Γ_{sl} is zero, whereas this term should not vanished or should be retained in the local weak form of MLPG-US-II. In the Bubnov-Galerkin discretization procedure, the same spaces for the test and trial functions are used for MLS (Lin and Atluri, 2000) and termed as MLPG-6 (see Chapter 3). Such formulations are based on mixed form. The essential BCs are imposed herein by using direct interpolation method, which is very easy to implement and effective. For pressure calculations at each points, the most popular Babuska-Brezzi(B-B) condition (Babuska, 1971; Brezzi, 1974) is used, which is based on the mixed formulations and a special condition is added to overcome the spurious pressure solutions. This Babuska-Brezzi(B-B) stability condition must be satisfied to obtain the effective solutions (Atluri and Shen, 2002b).

Furthermore, an approximate ‘perturbation’ term based on residual forms of the Euler-Lagrange equations is added. It is a straightforward approach to modify the standard mixed formulations. Eq. (5.21) can be reformed by applying this approach as follows.

$$\int_{\Omega_s} \frac{\partial u_i}{\partial x_i} w \partial \Omega + \tau \int_{\Omega_s} \left(u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\mu}{Re} \left(\frac{\partial^2 u_i}{\partial x_j^2} \right) - \frac{2}{Re} \left(\varepsilon_{ij} \frac{\partial \mu_i}{\partial x_j} \right) - f_i \right) \times \frac{\partial w}{\partial x_i} \partial \Omega = 0 \quad (5.25)$$

Here, τ is the stability parameter for pressure calculation. The optimized value of τ can solve the pressure stability issues without upsetting the consistency. Similar to stabilized FEM (Pierre, 1988; Franca and Frey, 1992, etc.), the stability parameter (τ) may be defined as

$$\tau = \begin{cases} \beta D^2 & Re_L < 1 \\ \frac{D}{2\|u\|} & Re_L \geq 1 \end{cases} \quad (5.26)$$

where, D is the diameter of the support domain for the weight function. The local Reynolds number ($Re_L = D\|u\|Re$) is related to local domain and velocity $\|u\| = \sqrt{u_i u_i}$. According to Eq. (5.26), τ may depend on global Re and the size of local sub-domain (Lin and Atluri, 2001). The best choice of parameter $\beta = 0.01$ is found for a particular problem (Lin and Atluri, 2001), but it may vary and independently be chosen for considered problems. Further, due to increasing cost of computations and irrelevant to calculate the 2nd order derivatives of the shape function, 2nd order derivative terms in Eq. (5.25) are ignored in present in-house solver. Similar strategy has been used in the existing literature on solution of Navier-Stokes equation for Newtonian fluid flow (Nayroles et al., 1992; Lin and Atluri, 2001; Wu et al., 2010; Tezduyar and Osawa, 2000)

For the formation of algebraic equation, MLS scheme is applied as field approximation and discretization technique. The field variables can be approximated using the following MLS approximation functions.

$$u^h(x) = \boldsymbol{\phi}^T(\mathbf{x}) \cdot \hat{\mathbf{u}} = \sum_{I=1}^N \phi(x) \cdot \hat{u}_I \quad \text{for } x \in \Omega_s \quad (5.27)$$

$$v^h(x) = \boldsymbol{\phi}^T(\mathbf{x}) \cdot \hat{\mathbf{v}} = \sum_{I=1}^N \phi(x) \cdot \hat{v}_I \quad \text{for } x \in \Omega_s \quad (5.28)$$

$$p^h(x) = \boldsymbol{\phi}^T(\mathbf{x}) \cdot \hat{\mathbf{p}} = \sum_{I=1}^N \phi(x) \cdot \hat{p}_I \quad \text{for } x \in \Omega_s \quad (5.29)$$

where $\phi(x)$ is the shape function for the local sub-domain Ω_s for the computations of field variables approximations. On substitution of Eqs. (5.27 - 5.29) into Eqs. (5.24 - 5.25) for all the nodes, the following system of matrices can be obtained.

$$\mathbf{K} \hat{\mathbf{u}}_j = \mathbf{F} \quad (5.30)$$

Where, $\hat{\mathbf{u}}_j$ is the vector of the unknown fictitious nodal values, $\hat{\mathbf{u}}_j = \{\hat{u}_j^1, \hat{u}_j^2, \dots, \hat{u}_j^N\}$.

The global stiffness matrix (\mathbf{K}) and the global vector (\mathbf{F}) are defined as follow.

$$K_{IJ} = \int_{\Omega_s} \left(u_j \frac{\partial u_i}{\partial x_j} w_i + \frac{\mu}{Re} \left(\frac{\partial u_i}{\partial x_j} \frac{\partial w_i}{\partial x_j} \right) \right) \partial \Omega - \int_{\Gamma_{sl}} \left(\frac{\mu}{Re} \frac{\partial u_i}{\partial x_j} \right) w_i n_j \partial \Gamma - \int_{\Gamma_{su}} \left(\frac{\mu}{Re} \frac{\partial u_i}{\partial x_j} \right) w_i n_j \partial \Gamma - \int_{\Gamma_{st}} \left(\frac{\mu}{Re} \frac{\partial u_i}{\partial x_j} \right) w_i n_j \partial \Gamma \quad (5.31)$$

$$F_I = \int_{\Omega_s} \left(p \frac{\partial w_i}{\partial x_i} + \frac{2}{Re} \left(\varepsilon_{ij} \frac{\partial \mu_i}{\partial x_j} \right) w_i + f_i w_i \right) \partial \Omega - \int_{\Gamma_{sl}} p w_i \partial \Gamma - \int_{\Gamma_{su}} p w_i \partial \Gamma - \int_{\Gamma_{st}} p w_i \partial \Gamma \quad (5.32)$$

The corresponding matrices components for pressure are evaluated as,

$$K_{IJ} = \tau \int_{\Omega_s} \left(\frac{\partial p}{\partial x_i} \frac{\partial w}{\partial x_i} \right) \partial \Omega \quad (5.33)$$

$$F_I = \tau \int_{\Omega_s} \left(\varepsilon_{ij} \frac{\partial \mu}{\partial x_j} - u_j \frac{\partial u_i}{\partial x_j} \right) \frac{\partial w}{\partial x_i} \partial \Omega - \int_{\Omega_s} \frac{\partial u_i}{\partial x_i} w \partial \Omega \quad (5.34)$$

The above set of equations have been solved by considering a square channel of ($0 \leq x, y \leq 1$) populated with $N = 11 \times 11$ and 21×21 with uniformly distributed nodes. The uniform stream ($u_\infty = 1$) of non-Newtonian power-law ($0.6 \leq n \leq 1.4$) fluids is

approaching the channel entrance. The exit velocity profile is fully developed. A quadratic basis function $\mathbf{P}^T(\mathbf{x}) = [1 \ x \ y \ x^2 \ xy \ y^2]$ with quartic spline weight function $w(\mathbf{x})$ is used for MLS shape function approximation. Each quarter of quadrature domain Ω_q is uniformly populated with 4 Gauss points for integration purpose. For the circular local support domain Ω_s , an appropriate value of α_s and α_q is selected from a numerical study using several trial tests. The sizes of both local sub domain r_s and quadrature domain r_q depends on the value of power-law index (n). A criteria of $\delta = 10^5$ is used for the convergence of the iterative process.

Extensive numerical results have been obtained by systematic variation of the power-law index in the range as $0.6 \leq n \leq 1.4$ for two value of Reynolds number ($Re = 1$ and 20). The results computed using in-house MLPG solver has been compared with the exact fully developed flow solution (Eq. 5.15) to ascertain the reliability and accuracy of the present methodology.

Figures 5.8 - 5.10 compare the exact and computational solutions for fully developed velocity, viscosity and strain tensor profiles at the outlet of channel for different n by using $N=121$ (i.e., 11×11) uniformly distributed nodes at $Re = 1$. Different values of both β and radius of support domain (r_s) are used for different n to obtain the best possible computational results by using MLPG and MLPG-US-II with 121 uniformly distributed nodal points. It is also observed that the smaller values of $\beta (< 1)$ produces better computational results. It is observed that, at the large value of β , the boundary behavior comes in to effect, due to which it may deteriorates the pressure at near the boundary in the mixed formulation system.

The fully developed velocity patterns (Figure 5.8) obtained using both MLPG and MLPG-US-II are seen to be qualitatively similar to the analytical solutions over the range of power-law index ($0.2 \leq n \leq 1$) at $Re = 1$. The Results obtained from MLPG method on viscosity and strain tensor have also been analyzed in Figures 5.9 and 5.10 to understand the issues pointed in viscosity profiles. The analysis of the profiles shows that the numerical results are not exact for smaller value of $n < 1$ (shear thinning) and higher value of $n > 1$

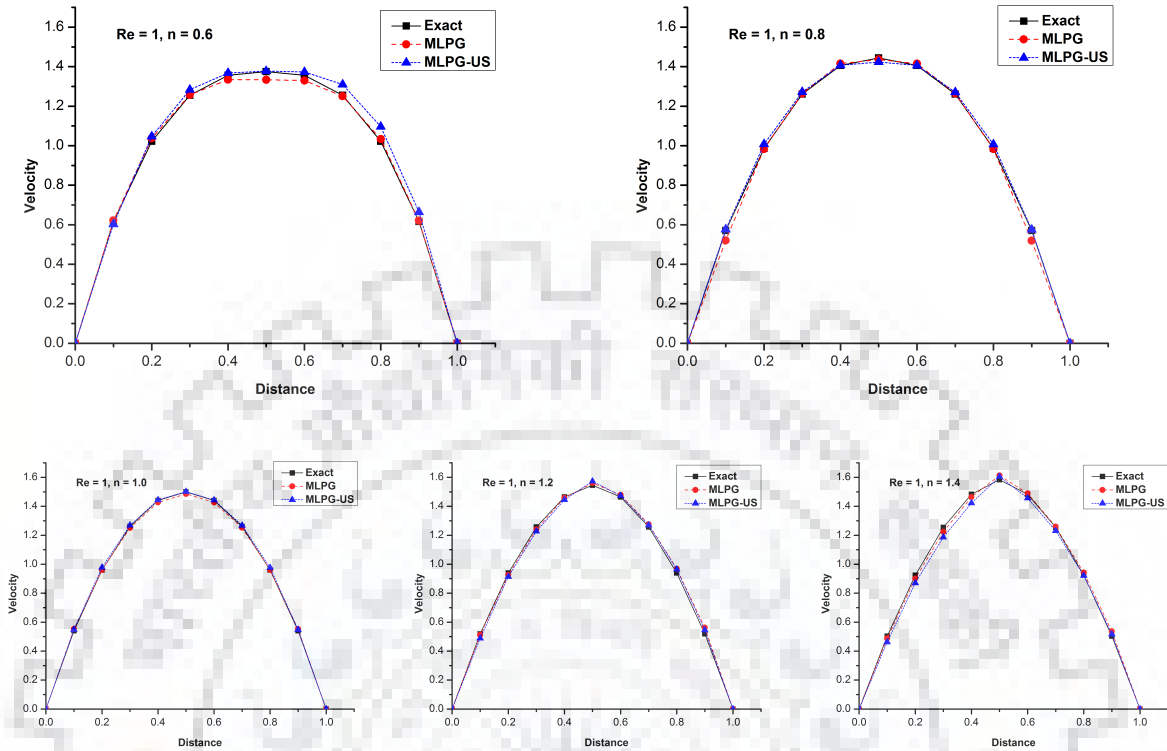


FIGURE 5.8: Fully developed velocity profile at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N=121$ nodes.

(shear thickening) due to (i) insufficient nodal points (ii) effect of parameter such as pressure and viscosity gradient and (iii) inaccuracy of upwinding scheme used to prevent the wiggle nature of flow. In general, the complexities in numerical solution of such complex flows appears due to shear-dependent viscosity. It requires sufficient number of nodes for correct resolution of both sharp as well as steep gradients of flow variables. Similar to velocity patterns, both local viscosity and strain patterns show excellent correspondence between the MLPG and exact results for $n = 1$ (Newtonian flow). The maximum deviations between the present and exact results appeared for both small as well as large values of n , i.e., under extremes of shear-thinning ($n \ll 1$) and shear-thickening ($n \gg 1$) fluid behaviors. One possible reason behind this deviation is that the viscosity becomes extremely large, say infinity, for small shearing rate in extreme shear-thinning fluids and vice versa, say zero, in shear-thickening fluids. Such situations are major challenges to handle in computations of non-Newtonian fluid flows. As discussed earlier, refinement in

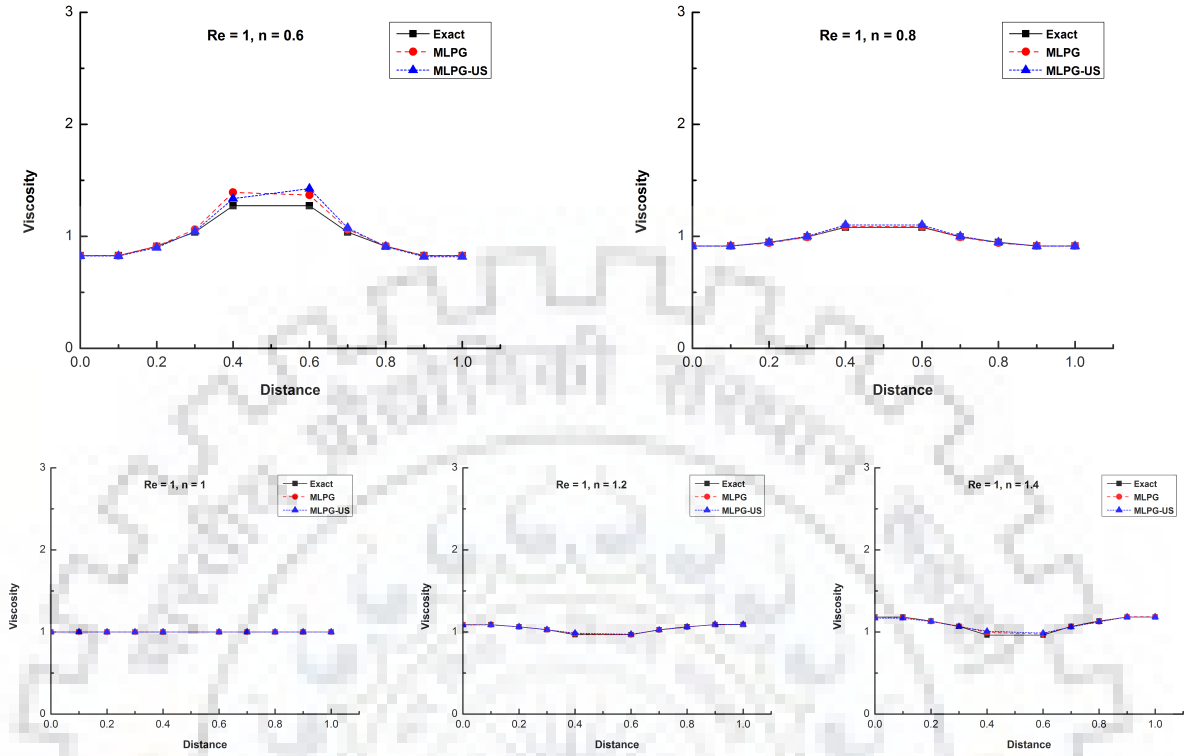


FIGURE 5.9: Viscosity profiles at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N=121$ nodes.

the nodes is one solution to control this errors. In addition to enhancement of the nodal points, these errors can also be reduced through refined optimized selection of β , radius of support and quadrature domains, etc. At $\beta = 0$ (i.e. no modification in the mixed formulation), the pressure shows an oscillatory nature. At larger β the boundary comes in to effect which suppresses the oscillation and yields smooth convergence.

Further, the computational nodes are doubled and $N = 441$ (i.e., 21×21) nodes are selected to analyze their influence on the flow patterns of power-law fluids ($0.6 \leq n \leq 1.4$) at $Re = 1$. The value of β is suitably adjusted for better stability parameter τ in the mixed formulation to control the non-linearity. Table 5.3 has summarized the optimized values of β , radius of support domain Ω_s and shifting distance of test domain $\Delta = \gamma r_i$ for upwinding schemes for different values of n at $Re = 1$ for $N = 121$ and 441 nodes. The velocity, viscosity and strain rate profiles are depicted in Figures 5.11 - 5.13, respectively. Similar to the case of $N = 121$ nodes, the numerical results obtained using both MLPG and

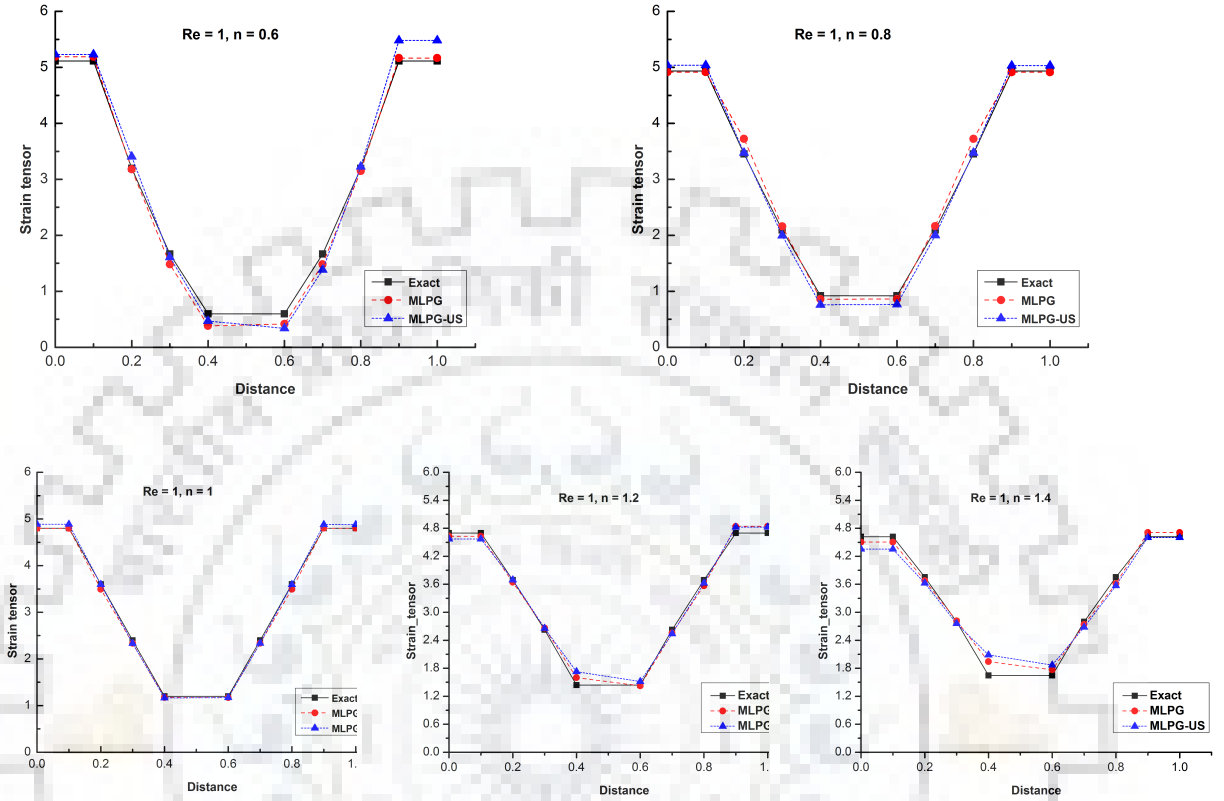


FIGURE 5.10: Strain tensor profiles at the outlet of channel for different n at $Re = 1$. Comparison of exact with numerical solutions for $N=121$ nodes.

TABLE 5.3: Optimized value of β and radius of support domains at $Re = 1$ for a range of power-law index ($0.6 \leq n \leq 1.4$). Here Δ is the shifting of test domain.

n	N	MLPG		MLPG-US-II		Δ
		r_s	β	r_s	β	
0.6	121	$2.21 \times h$	0.019	$2.20 \times h$	0.01	0.08
0.8	121	$2.29 \times h$	0.02	$2.30 \times h$	0.02	0.10
1.0	121	$2.10 \times h$	0.03	$2.25 \times h$	0.5	0.06
1.2	121	$2.15 \times h$	0.01	$2.31 \times h$	0.01	0.04
1.4	121	$2.18 \times h$	0.01	$2.30 \times h$	0.02	0.02
0.6	441	$2.30 \times h$	0.10	$2.15 \times h$	0.4	0.04
0.8	441	$2.22 \times h$	0.03	$2.15 \times h$	0.15	0.06
1.0	441	$2.16 \times h$	0.03	$2.15 \times h$	0.15	0.06
1.2	441	$2.10 \times h$	0.01	$2.15 \times h$	0.15	0.06
1.4	441	$2.10 \times h$	0.10	$2.25 \times h$	0.10	0.06

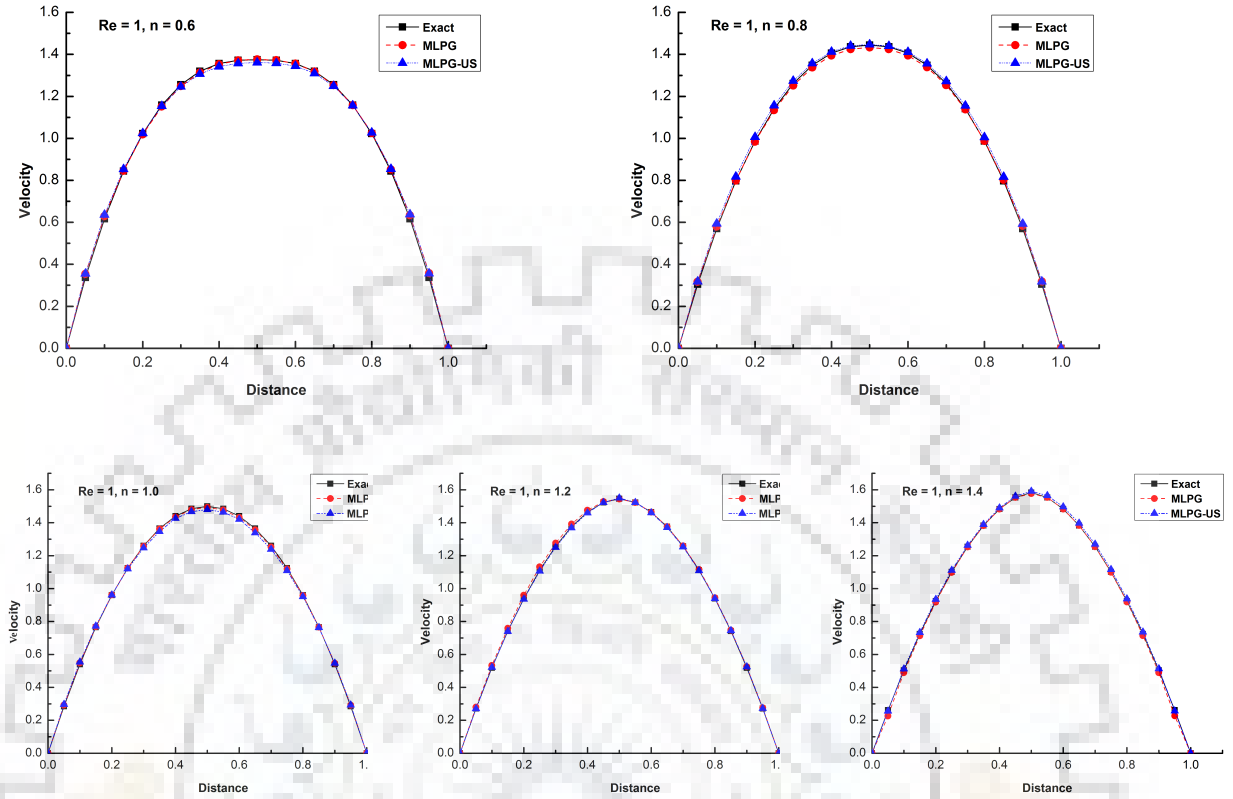


FIGURE 5.11: Fully developed velocity profile at the outlet of channel for different n at $Re = 1$. Comparison of exact with numerical solutions for $N=(441)$ nodes.

MLPG-US-II are compared with exact ones to discuss the suitability of present numerical solver in computation of complex fluids flow. The flow profiles clearly seem to be matching with the exact profiles. Based on the results of $N = 121$ and 441 nodes, it can also be summarized that the dense distribution of nodal points yield more accurate results in comparison to the sparse distribution. The maximum deviation in numerical and exact results appear near the boundary nodes, and it can be minimize on the best selection of radius of quadrature domain Ω_q , radius of support domain Ω_s and β . Further, the inaccuracies observed in the results of 121 nodes have also diminished with increasing nodal point in the computational domain. While both MLPG and MLPG-US-II works better in computation of non-Newtonian flow, the results of MLPG are much accurate. The effect of nodal points ($N = 11 \times 11$ and 21×21 nodes) on the optimized value of β is shown in Table 5.3 at $Re = 1$ and $0.6 \leq n \leq 1.4$. It is noted that the radius of support domain may not be fixed for both MLPG and MLPG-US-II at different n as in Figure 5.13.

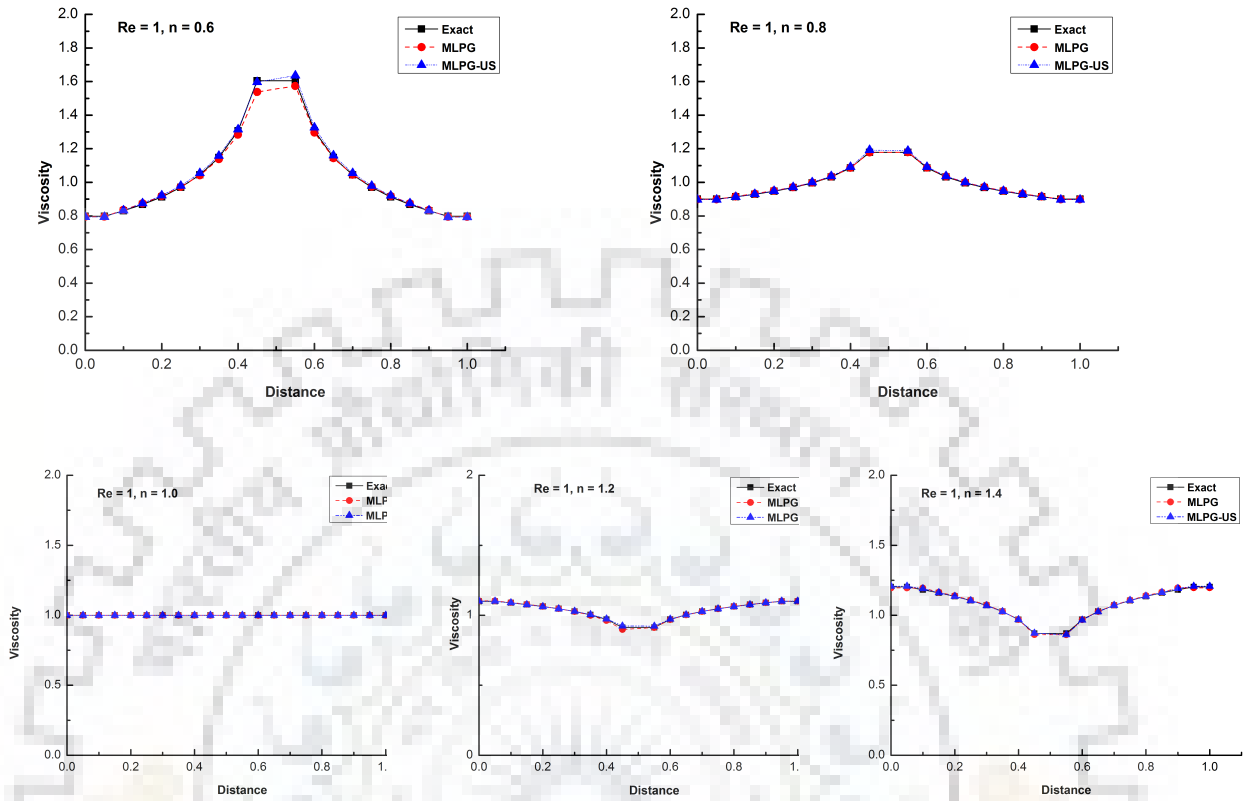


FIGURE 5.12: Viscosity profiles at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N=(21 \times 21)$ nodes.

It is also not straightforward to say that larger radius of support is best or small radius support for β . The maximum errors may occur from the corner nodes, because less points are covered by a support domain. But the results are very much satisfactory for 21×21 nodes. Hence, we can say that dense nodal distribution with refinement of corner points will produce excellent results.

At $Re=1$, MLPG yielded better results without any oscillations in comparison of MLPG-US-II. However, the effect of advective term is dominant with increasing Re , and more oscillatory nature appears in the solution. To stabilize, it is necessary to use the upwinding scheme to control the oscillatory behavior.

The standard upwind scheme in which $Pe = Re_L$, described by [Lin and Atluri \(2000\)](#) is not well imposed for the non-Newtonian fluid rheology in which viscosity and pressure plays an important role. Due to shear dependency of viscosity, the insignificant oscillatory

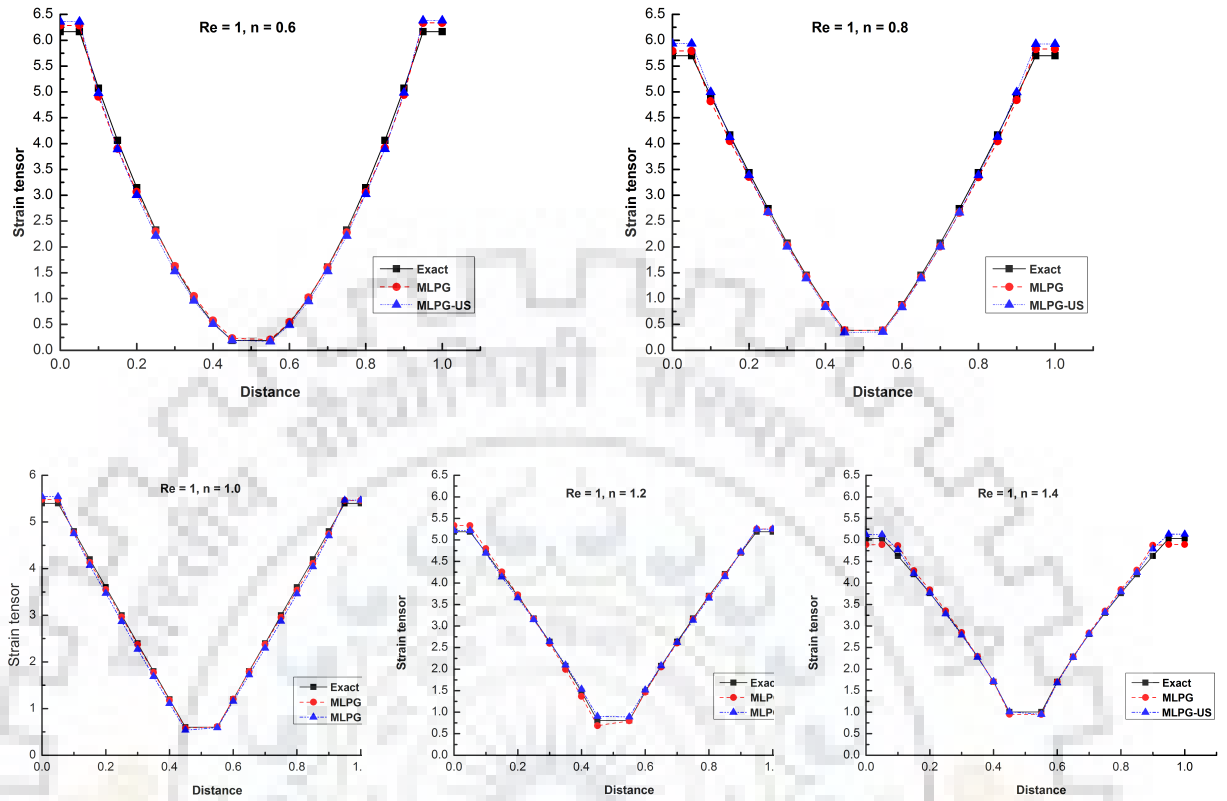


FIGURE 5.13: Strain tensor profiles at the outlet of channel for different n at fixed Re . Comparison of exact solution with numerical (MLPG and MLPG-US-II) solutions for $N=(21 \times 21)$ nodes.

nature appears at low Re . However, a suitable upwind scheme is necessary to control the oscillatory nature at sufficiently large Re . Subsequently, it was observed that at $Re = 20$, there is no existence of oscillatory nature, hence, only MLPG method is used to strengthen the applicability of our solver. The numerical solutions are validated with exact solutions. Again different values of β are chosen for our computations using MLPG method with mixed formulation system to get satisfactory results. Different values of correction factors f range between 0.8 to 1 are sufficient for better observations which are summarized in Table 5.4 at different power law index. Figures 5.14, 5.15, and 5.16 report the fully developed velocity, viscosity and strain rate profiles at the outlet of channel for $Re = 20$ and $0.6 \leq n \leq 1.8$ by using 441 computational nodes. In general, the observations are similar to the case of $Re = 1$. At the lower value of n , the viscosity tends to ∞ , which is very important to control the appearance of spurious results, as depicted at $n = 0.6$ and 1.8 in Figures 5.15 and 5.16. As seen in these figures, the numerical results seem to

TABLE 5.4: Optimized value of β , radius of support domains (r_s) and correction factor (f) for different n at $Re = 20$ by using MLPG with $N = 441$ computational nodes.

n	r_s	β	f
0.6	$2.13 \times h$	0.20	0.8
0.8	$2.25 \times h$	0.20	0.8
1.0	$2.11 \times h$	0.01	1.0
1.2	$2.23 \times h$	0.01	0.8
1.4	$2.30 \times h$	0.05	0.8
1.6	$2.30 \times h$	0.10	0.8
1.8	$2.30 \times h$	0.40	0.85

correspond excellently with the exact results. The treatment of corner points requires special attention to obtain the correct results in neighborhood of these points.

In the mixed formulation system, the pressure solution is depend on choice of β . [Lin and Atluri \(2001\)](#) suggested that at low values of β , the pressure solution appears more precise and smooth, and should always be ≤ 1 . In the figure [5.17](#) the pressure contours are shown on behalf of velocity accuracy at the outlet of channel at the several trial values of β . From the computational results, it is found that correction factor f is decreases from 10 to < 1 by increasing Re from 1 to 20, see table [5.4](#), which is always < 1 .

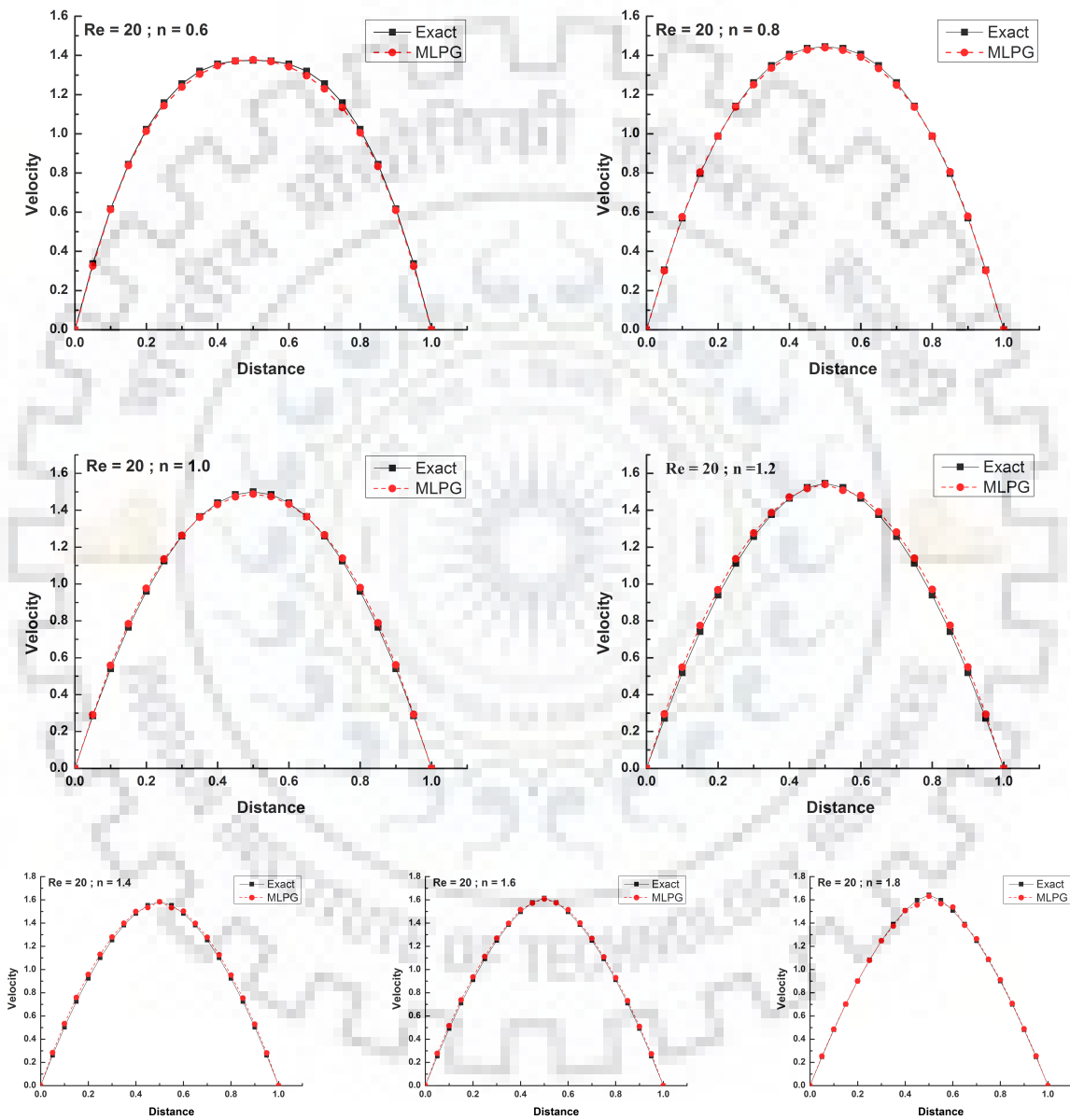


FIGURE 5.14: Fully developed velocity profile at the outlet of channel for different n at fixed $Re = 20$. Comparison of exact with MLPG solutions.

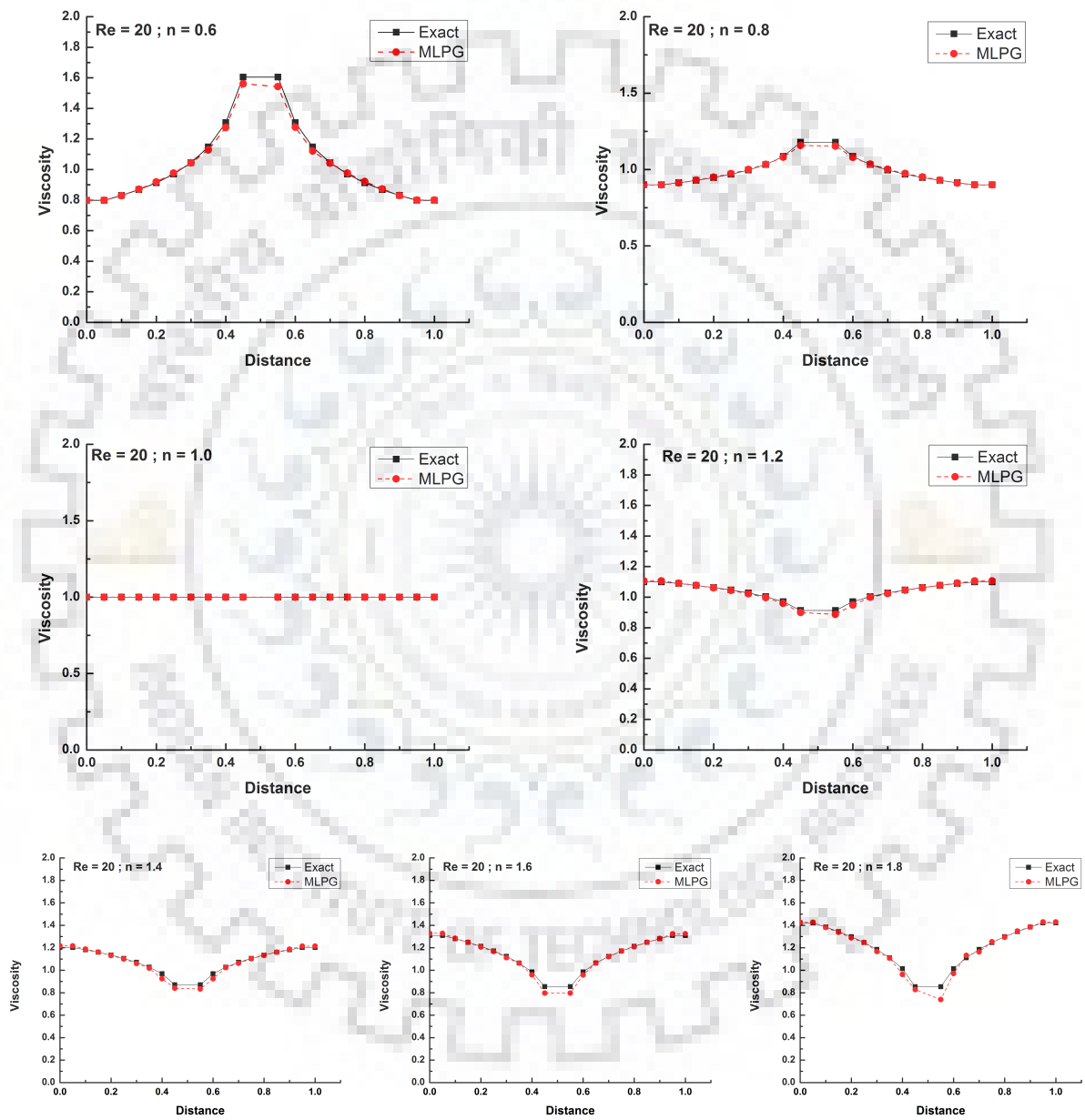


FIGURE 5.15: Viscosity profiles at the outlet of channel for different n at fixed $Re = 20$. Comparison of exact solution with MLPG solutions.

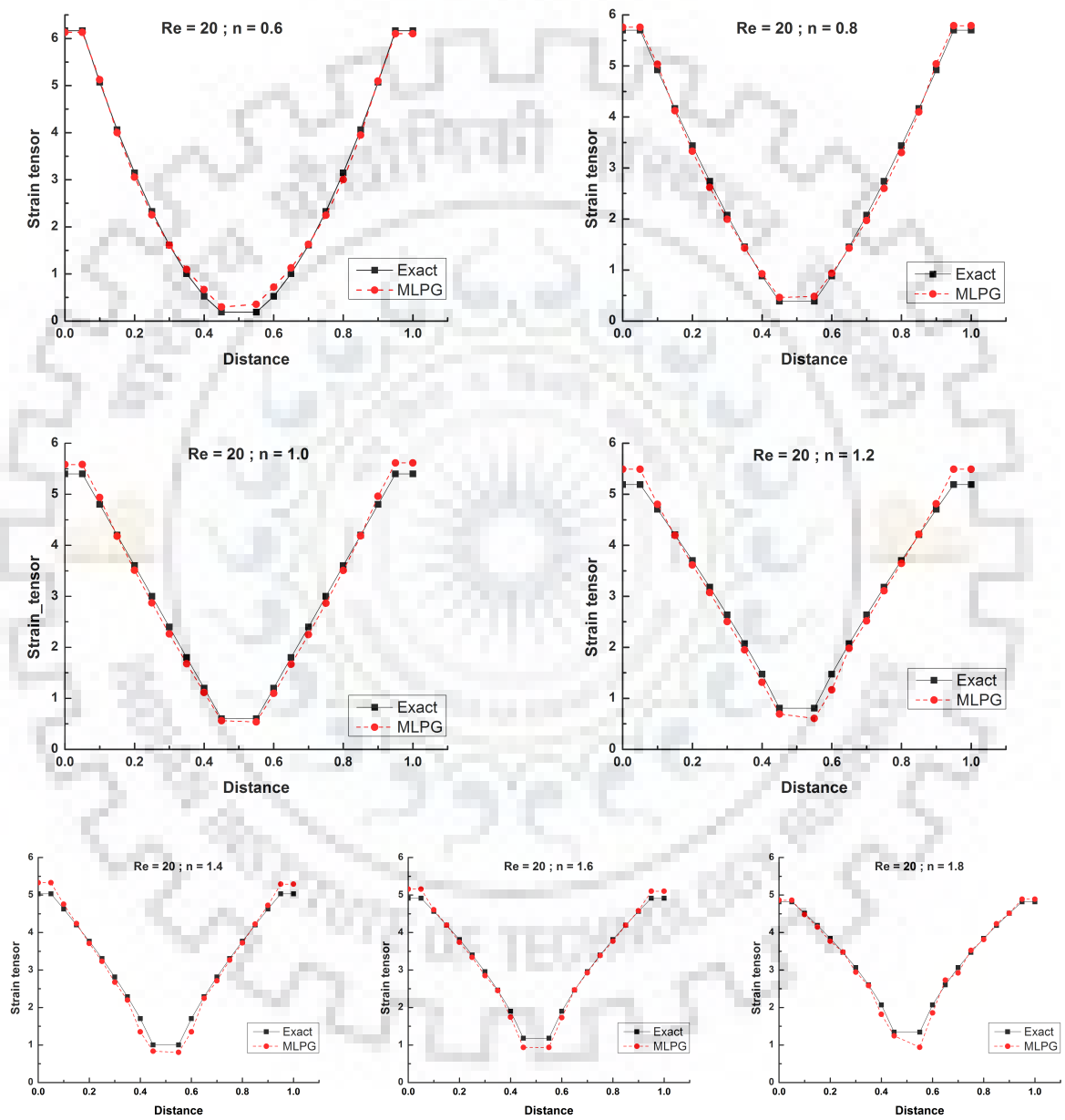


FIGURE 5.16: Viscosity profiles at the outlet of channel for different n at fixed $Re = 20$ with 21×21 nodes. Comparison of exact solution with MLPG solutions.

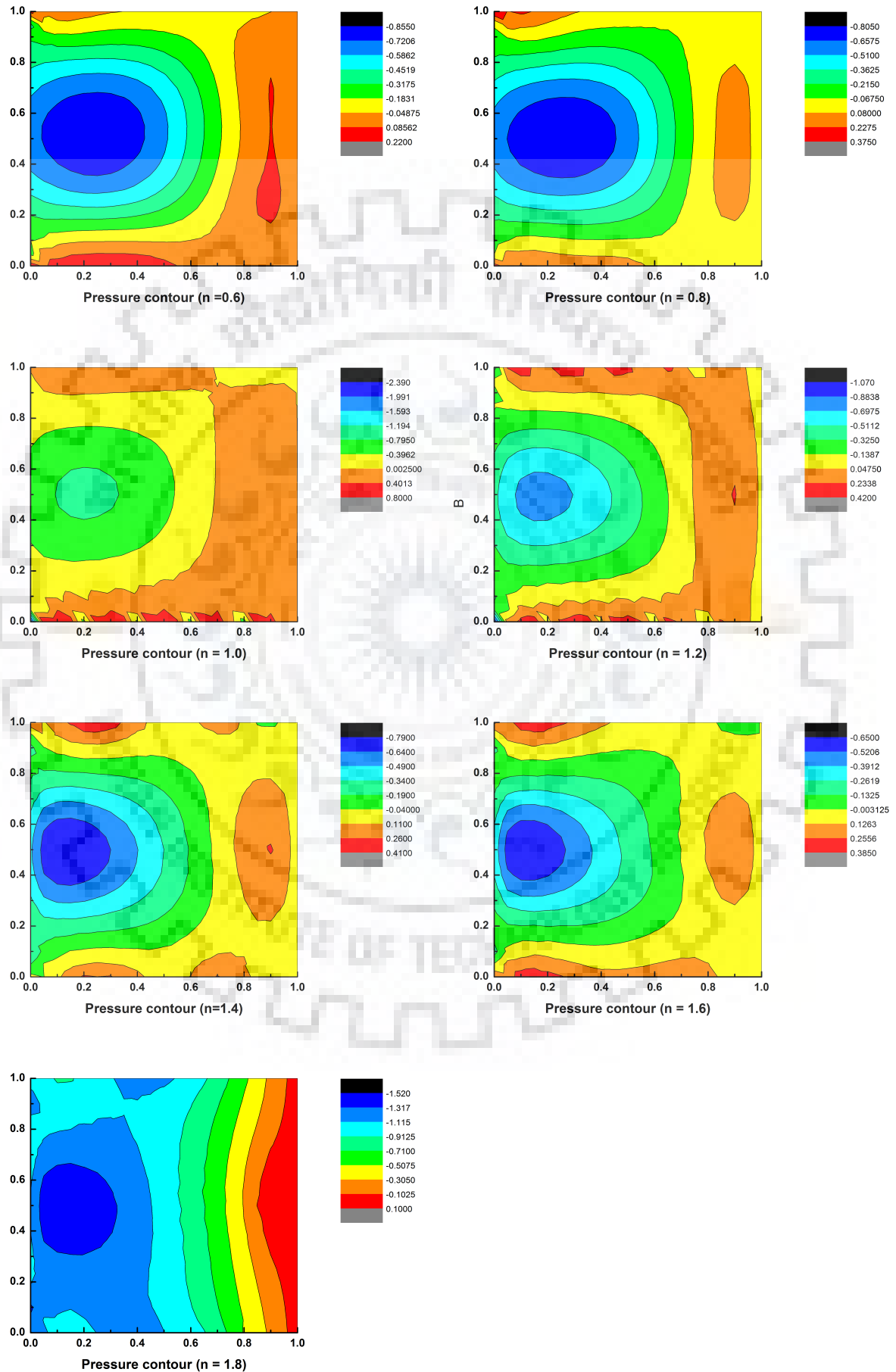


FIGURE 5.17: Pressure contour at different n by using MLPG with $N = (21 \times 21)$ nodes at $Re = 20$

TABLE 5.5: Comparison of local velocity and viscosity values with exact solution nodes at $Re = 1$ by using for 11×11 computational nodes. (Here A, B and C represents for exact, MLPG and MLPG-US-II solutions).

Node no.	n=0.6			n=0.8			n=1.0			n=1.2			n=1.4		
	u_A	u_B	u_C	u_A	u_B	u_C	u_A	u_B	u_C	u_A	u_B	u_C	u_A	u_B	u_C
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.616	0.621	0.602	0.570	0.520	0.576	0.540	0.552	0.549	0.518	0.512	0.488	0.503	0.488	0.462
3	1.022	1.037	1.046	0.986	0.983	1.007	0.960	0.959	0.977	0.939	0.926	0.913	0.923	0.900	0.870
4	1.255	1.257	1.283	1.260	1.264	1.271	1.260	1.250	1.268	1.257	1.241	1.227	1.254	1.224	1.187
5	1.356	1.334	1.367	1.405	1.415	1.406	1.440	1.426	1.444	1.464	1.458	1.445	1.483	1.464	1.422
6	1.375	1.333	1.377	1.444	1.436	1.423	1.500	1.485	1.501	1.545	1.560	1.572	1.583	1.612	1.604
7	1.356	1.330	1.373	1.405	1.415	1.406	1.440	1.427	1.443	1.464	1.479	1.472	1.483	1.489	1.456
8	1.255	1.250	1.310	1.260	1.264	1.270	1.260	1.251	1.267	1.257	1.275	1.269	1.254	1.259	1.230
9	1.022	1.033	1.096	0.986	0.982	1.006	0.960	0.960	0.976	0.939	0.968	0.964	0.923	0.941	0.920
10	0.616	0.620	0.664	0.570	0.519	0.575	0.540	0.552	0.548	0.518	0.562	0.544	0.503	0.536	0.516
11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
no.	η_A	η_B	η_C	η_A	η_B	η_C	η_A	η_B	η_C	η_A	η_B	η_C	η_A	η_B	η_C
1	0.828	0.826	0.825	0.913	0.913	0.911	1	1	1	1.089	1.087	1.086	1.182	1.176	1.168
2	0.828	0.826	0.825	0.913	0.913	0.911	1	1	1	1.089	1.087	1.086	1.182	1.176	1.168
3	0.910	0.911	0.899	0.946	0.939	0.946	1	1	1	1.063	1.061	1.063	1.134	1.129	1.126
4	1.037	1.061	1.044	0.995	0.992	1.000	1	1	1	1.027	1.029	1.028	1.069	1.070	1.066
5	1.273	1.392	1.336	1.080	1.088	1.101	1	1	1	0.967	0.977	0.985	0.961	0.994	1.008
6		2.459	2.382		2.256	1.964	1	1	1	1	0.742	0.766	0	0.574	0.611
7	1.273	1.369	1.426	1.080	1.087	1.100	1	1	1	0.967	0.966	0.972	0.961	0.975	0.986
8	1.037	1.061	1.076	0.995	0.992	1.000	1	1	1	1.027	1.024	1.024	1.069	1.064	1.060
9	0.910	0.913	0.908	0.946	0.939	0.946	1	1	1	1.063	1.059	1.061	1.134	1.125	1.122
10	0.828	0.827	0.817	0.913	0.914	0.911	1	1	1	1.089	1.092	1.092	1.182	1.186	1.181
11	0.828	0.827	0.817	0.913	0.914	0.911	1	1	1	1.089	1.092	1.092	1.182	1.186	1.181

TABLE 5.6: Comparison of local velocity and viscosity values with exact solution nodes at $Re = 1$ by using for 21×21 computational nodes. (Here A, B and C represents for exact, MLPG and MLPG-US-II solutions).

Node no.	n=0.6			n=0.8			n=1.0			n=1.2			n=1.4		
	u_A	u_B	u_C	u_A	u_B	u_C	u_A	u_B	u_C	u_A	u_B	u_C	u_A	u_B	u_C
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.617	0.629	0.636	0.570	0.579	0.594	0.540	0.548	0.554	0.519	0.533	0.522	0.503	0.489	0.504
5	1.023	1.019	1.025	0.987	0.984	1.006	0.960	0.962	0.961	0.940	0.959	0.936	0.924	0.918	0.936
7	1.256	1.248	1.246	1.261	1.252	1.273	1.260	1.258	1.248	1.257	1.276	1.250	1.254	1.253	1.277
9	1.356	1.353	1.342	1.406	1.394	1.413	1.440	1.435	1.416	1.465	1.475	1.460	1.483	1.482	1.519
11	1.375	1.376	1.361	1.444	1.433	1.447	1.500	1.493	1.470	1.545	1.544	1.550	1.583	1.577	1.633
13	1.356	1.356	1.344	1.406	1.395	1.412	1.440	1.432	1.411	1.465	1.464	1.461	1.483	1.482	1.519
15	1.256	1.253	1.249	1.261	1.254	1.272	1.260	1.253	1.239	1.257	1.260	1.253	1.254	1.254	1.276
17	1.023	1.025	1.028	0.987	0.988	1.006	0.960	0.957	0.951	0.940	0.944	0.940	0.924	0.919	0.936
19	0.617	0.634	0.638	0.570	0.583	0.593	0.540	0.546	0.547	0.519	0.525	0.525	0.503	0.489	0.504
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
no.	η_A	η_B	η_C	η_A	η_B	η_C	η_A	η_B	η_C	η_A	η_B	η_C	η_A	η_B	η_C
1	0.798	0.795	0.793	0.900	0.899	0.896	1	1	1	1.100	1.103	1.100	1.202	1.195	1.202
3	0.830	0.835	0.833	0.913	0.915	0.912	1	1	1	1.089	1.091	1.089	1.182	1.194	1.195
5	0.913	0.918	0.921	0.947	0.949	0.948	1	1	1	1.063	1.064	1.062	1.134	1.139	1.142
7	1.044	1.042	1.054	0.996	0.998	0.999	1	1	1	1.028	1.026	1.028	1.070	1.073	1.078
9	1.306	1.283	1.315	1.085	1.086	1.091	1	1	1	0.969	0.962	0.973	0.968	0.969	0.978
11		2.946	2.914		2.033	1.736	1	1	1	1	0.697	0.506	0	0.234	0.318
13	1.306	1.296	1.326	1.085	1.087	1.091	1	1	1	0.969	0.969	0.972	0.968	0.968	0.979
15	1.044	1.045	1.055	0.996	0.999	0.999	1	1	1	1.028	1.026	1.027	1.070	1.072	1.078
17	0.913	0.918	0.920	0.947	0.949	0.948	1	1	1	1.063	1.063	1.061	1.134	1.139	1.141
19	0.830	0.834	0.833	0.913	0.915	0.912	1	1	1	1.089	1.089	1.089	1.182	1.195	1.195
21	0.798	0.794	0.792	0.900	0.898	0.897	1	1	1	1.100	1.101	1.101	1.202	1.195	1.202

TABLE 5.7: Comparison of local velocity and viscosity values with exact solution nodes at $Re = 20$ by using for 21×21 computational nodes. (Here A, and B represents for exact, and MLPG solutions).

Node no.	n=0.6		n=0.8		n=1.0		n=1.2		n=1.4		n=1.6		n=1.8	
	u_A	u_B	u_A	u_B	u_A	u_B	u_A	u_B	u_A	u_B	u_A	u_B	u_A	u_B
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.617	0.613	0.570	0.576	0.540	0.558	0.519	0.549	0.503	0.533	0.491	0.516	0.482	0.486
5	1.023	1.013	0.987	0.988	0.960	0.976	0.940	0.968	0.924	0.957	0.911	0.936	0.901	0.901
7	1.256	1.239	1.261	1.250	1.260	1.263	1.257	1.276	1.254	1.280	1.251	1.270	1.248	1.249
9	1.356	1.348	1.406	1.393	1.440	1.431	1.465	1.471	1.483	1.500	1.497	1.515	1.508	1.507
11	1.375	1.378	1.444	1.440	1.500	1.487	1.545	1.540	1.583	1.583	1.615	1.608	1.643	1.631
13	1.356	1.342	1.406	1.391	1.440	1.433	1.465	1.479	1.483	1.503	1.497	1.515	1.508	1.537
15	1.256	1.230	1.261	1.249	1.260	1.266	1.257	1.281	1.254	1.278	1.251	1.268	1.248	1.265
17	1.023	1.005	0.987	0.989	0.960	0.980	0.940	0.971	0.924	0.952	0.911	0.930	0.901	0.912
19	0.617	0.610	0.570	0.579	0.540	0.552	0.519	0.549	0.503	0.529	0.491	0.510	0.482	0.489
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
no.	η_A	η_B	η_A	η_B	η_A	η_B	η_A	η_B	η_A	η_B	η_A	η_B	η_A	η_B
1	0.798	0.799	0.901	0.900	1.000	1.000	1.100	1.106	1.203	1.217	1.309	1.329	1.421	1.427
3	0.830	0.828	0.914	0.912	1.000	1.000	1.089	1.092	1.183	1.189	1.281	1.285	1.385	1.381
5	0.913	0.919	0.947	0.950	1.000	1.000	1.063	1.061	1.135	1.132	1.213	1.207	1.299	1.289
7	1.044	1.044	0.996	1.000	1.000	1.000	1.028	1.023	1.071	1.060	1.124	1.112	1.186	1.167
9	1.307	1.245	1.085	1.080	1.000	1.000	0.970	0.959	0.969	0.925	0.985	0.960	1.013	0.963
11		2.351		1.599	1.000	1.000	1.000	0.742	0.000	0.286	0.000	0.277	0.000	0.316
13	1.307	1.226	1.085	1.079	1.000	1.000	0.970	0.947	0.969	0.925	0.985	0.958	1.013	0.971
15	1.044	1.042	0.996	1.001	1.000	1.000	1.028	1.023	1.071	1.063	1.124	1.121	1.186	1.164
17	0.913	0.922	0.947	0.951	1.000	1.000	1.063	1.062	1.135	1.132	1.213	1.209	1.299	1.294
19	0.830	0.829	0.914	0.912	1.000	1.000	1.089	1.092	1.183	1.188	1.281	1.282	1.385	1.385
21	0.798	0.800	0.901	0.899	1.000	1.000	1.100	1.106	1.203	1.215	1.309	1.324	1.421	1.430

Finally, quantitative comparisons of the local velocity and viscosity values with the exact values, under identical conditions are also shown in Tables 5.5 to 5.7 for $Re = 1$ and 20 and for a broad range of power-law index ($0.6 \leq \leq 1.8$) by using the 11×11 and 21×21 uniformly distributed computational nodes. The comparisons have shown that MLPG method yields better results in comparison of MLPG-US-II method. The density of the nodal distribution also influences the solutions. The sufficiently refined nodal distribution is needed to obtain the correct results.

In summary, several incompressible N-S equations are solved before switching to non-Newtonian complex fluid flow simulations in which shear dependent viscosity and pressure plays an important role in fluid flow mechanism. The Babuska-Breezi condition is used for the MLPG method. The second order derivatives in the mixed formulation is vanished due to expensive computations. In the mixed formulation system, for pressure calculations, one stability parameter is included without upsetting consistency and to satisfy the mass continuity equation.

The numerical results suggested that the predefined upwinding scheme termed as MLPG-US-II is not suitable in the simulation of non-Newtonian viscous fluid flow. While the MLPG method is well established to solve the general fluid dynamics problems, but in the field of shear dependence viscous flow mechanism, further improvement is required for the precision computations of non-Newtonian fluid rheology at higher Reynolds number because of its non-linearity nature. Further study for higher Reynolds number is necessary to check the upwind schemes before any final conclusion, which will precise the way to investigate a better upwind scheme for a better results in non-Newtonian complex fluid flow simulations.



Chapter 6

Conclusions and future directions

This chapter concludes the work presented in this thesis. The recapitulation of purpose and findings have been summarized, followed by future possibilities of expansion of the work.

6.1 Recapitulation of purpose and findings

The development of the meshfree or meshless method is one of the exciting topics in computational mechanics in numerical methods. Several meshfree techniques have been developed by previous researchers, in which meshless local Petrov-Galerkin (MLPG) and linear boundary integral equation (LBIE) serves as a truly meshfree method amongst all other methods which required background mesh for integration purposes. Due to the truly meshfree characteristics, MLPG is an attractive computational method for the researchers, and widely applied in solid mechanics and fluid dynamics problems. It is proven to be an cost effective tool, as an alternative of convection CFD methods. It is often possible to find a numerical or approximate solution for a complex problem efficiently for the space/domain according to our interest.

Existing literature also suggested that the meshless methods, in particular, MLPG methods, have been successful in obtaining an accurate solutions of solid mechanics problems. Fluid mechanics problems have also been solved within the greater ease and accuracy. However, these meshless methods have not been used in obtaining a solution of complex fluid flow problems. It should, however, be noted here that non-Newtonian fluids are being processed in many chemical and process industries associated with varieties of transport phenomena in which flow properties differ in many ways from those of Newtonian fluids. The shear-dependent viscosity of non-Newtonian fluid is non-linear, in contrast to the Newtonian fluid where shear stress varies linearly with shear rate, passing through the origin and the proportionality constant being the coefficient of viscosity.

Due to lack of literature of application of MLPG in non-Newtonian fluid flow rheology, the present work aimed at exploration of applicability, effectiveness and ease of MLPG method in computational of complex fluid flow problems. This objective has been achieved by developing a MLPG complex fluid flow solver based on C++ programming language. The approach developed in MLPG complex fluid flow in-house solver will be very useful in analyzing and designing such applications at greater ease.

The solver, first, has been validated and verified by using various standard benchmark CFD problems based on the diffusion, convection and convection-diffusion equations. The excellent correspondence between the MLPG results and exact/literature results have motivated us to extend the development of solver for non-Newtonian flow computations. The detailed algorithm has also been developed herein, due to the lack of information for such applications. First, MLPG methods are used to solve one dimensional non-Newtonian flow problem with some mathematical modification to check its efficiency and accuracy. The results obtained using MLPG were quite smooth and accurate for different nodal distribution in computational domain. It further encouraged to apply for multi-dimensional space.

Subsequently, two-dimensional channel flow problem has been considered for assurance of MLPG in complex fluid flow simulations in which shear dependent viscosity and pressure

plays an important role on natural fluid dynamics phenomena. The channel flow problem is a standard bench mark problems in CFD and easy to analyze the effectiveness of variable parameters. The numerical results in terms of velocity, viscosity and strain rate are obtained for broader range of power-law index, including both shear-thinning as well shear-thickening fluid behaviors. The MLPG and MLPG with upwinding methods are used at fixed Re for initial observation using Ostwald-de waele power-law equation, which is used to describe the behavior of non-Newtonian fluid. MLPG with upwinding scheme is used to overcome the oscillatory behavior due to non-linearity at high Reynolds number is not well imposed at $Re > 1$. The results highlighted that the significant non-linear effects appear in the power-law fluids, particularly for extremely conditions of shear-thinning ($n < 1$) and shear-thickening ($n > 1$) fluid behaviors. Hence, it is necessary to improve the efficiency and accuracy for MLPG and MLPG with upwinding scheme to overcome the effect of non-linearity at corner nodes where more deviation occurred. It was also observed that MLPG is more accurate than the MLPG with upwinding, under otherwise identical conditions considered herein.

Having gained the excellent results at low Re , higher value of $Re = 20$ is also considered to present the robustness and applicability of in-house MLPG solver. The results found that MLPG is very promising and effective numerical approach for a complex fluid flow simulations under the laminar regime. It is further observed that irrespective of simulation conditions, sufficiently refined nodal distribution is required to obtain the correct results. A particular attention is required to resolve the gradients near the corner boundaries.

Finally, it can be concluded that the meshfree MLPG method have shown great potential to become a new generation numerical analysis tool for problems in complex fluid flow simulations. However, still many technical issues and challenges are need to be resolved. Further research is necessary to improve this meshfree method at higher Reynolds number where the fluid behavior posses the great challenges for the cases where local velocity may reach to zero or infinity, depending on the shear-thinning or shear-thickening fluid behaviors..

6.2 Recommendations for future research directions

Based on the work presented in this thesis, the following recommendation for future research directions are made:

- Extension of MLPG method in simulation of complex fluids under laminar regime.
- Extension to three-dimensional flow simulation of complex fluids
- Applicability of method in turbulent flow regime of non-Newtonian flow.
- Exploration of suitably convective upwinding scheme to resolve the non-linear convection terms.



Bibliography

- Abbasbandy, S., Shirzadi, A., 2010. A meshless method for two-dimensional diffusion equation with an integral condition. *Engineering Analysis with Boundary Elements* **34**, 1031–1037. (Cited on pages 21 and 66)
- Ahmadi, I., Sheikhy, N., Aghdam, M. M., Nourazar, S. S., 2010. A new local meshless method for steady-state heat conduction in heterogeneous materials. *Engineering Analysis with Boundary Elements* **34**, 1105–1112. (Cited on pages 21 and 65)
- Anderson, J., 1995. *Computational fluid dynamics: The basic with applications*. McGraw Hills, Inc. (Cited on pages 3 and 5)
- Atluri, S. N., Cho, J. Y., Kim, H., 1999a. Analysis of thin beams, using the meshless local petrov-galerkin method, with generalized moving least squares interpolations. *Computational Mechanics* **24**, 334–347. (Cited on pages 19 and 54)
- Atluri, S. N., Kim, H., Cho, J. Y., 1999b. A critical assessment of the truly meshless local petrov-galerkin(MLPG) and local boundary integral equation (LBIE) methods. *Computational Mechanics* **24**, 348–372. (Cited on pages iv, xi, 17, 19, 20, 39, 49, 51, 53, 54, and 56)
- Atluri, S. N., Shen, S., 2002a. The meshless local petrov-galerkin (MLPG) method: A simple and less-costly alternative to the finite element and boundary element methods. *CMES: Computer Modeling in Engineering & Sciences* **3**, 11–51. (Cited on pages 7, 17, 18, 49, and 51)

- Atluri, S. N., Shen, S., 2005. The basis of meshless domain discretization: The meshless local petrov galerkin (mlpg) method. *Advances in Computational Mathematics* **23**, 73–93. (Cited on pages iv, 7, 8, 17, 18, 20, and 30)
- Atluri, S. N., Shen, S. P., 2002b. *The Meshless Local Petrov Galerkin (MLPG) Method*. Crest. (Cited on pages 17, 51, and 109)
- Atluri, S. N., Sladek, J., Sladek, V., Zhu, T., 2000. The local boundary integral equation (LBIE) and it's meshless implementation for linear elasticity. *Computational Mechanics* **25**, 180–198. (Cited on pages 8 and 19)
- Atluri, S. N., Zhu, T., 1998a. A new meshless local petrov-galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* **22**, 117–127. (Cited on pages iv, xi, 6, 17, 19, 29, 41, and 59)
- Atluri, S. N., Zhu, T., 1998b. A new meshless local petrov galerkin (MLPG) approach to nonlinear problems in computational modeling and simulation. *Modeling Simulation in Engg.* **3**, 187–196. (Cited on pages 6, 17, 19, and 29)
- Atluri, S. N., Zhu, T., 2000. The meshless local petrov-galerkin (MLPG) approach for solving problems in elasto-statics. *Computational Mechanics* **25**, 169–179. (Cited on page 8)
- Babuska, I., 1971. Error bounds for finite element methods. *Numerische Mathematik* **16**, 322–333. (Cited on page 109)
- Babuska, I., Melenk, J. M., 1997. Partition of unity method. *International Journal for Numerical Methods in Engineering* **40**, 727–758. (Cited on pages 6 and 16)
- Beissel, S., Belytschko, T., 1996. Nodal integration of the element-free galerkin method. *Computer methods in applied mechanics and engineering* **139**, 49–74. (Cited on page 54)

-
- Belytschko, T., Krongauz, Y., Organ, Liu, W. K., 1996a. Smoothing and accelerated computations in the element free galerkin method. *Journal of Computational and Applied Mathematics* **74**, 111–126. (Cited on page 15)
- Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P., 1996b. Meshless methods: An overview and recent developments. *Computer methods in applied mechanics and engineering* **139**, 3–47. (Cited on pages 12, 17, 36, 44, and 49)
- Belytschko, T., Lu, Y. Y., Gu, L., 1994. Element free galerkin methods. *International Journal for Numerical Methods in Engineering* **37**, 229–256. (Cited on pages 6, 15, 48, and 54)
- Belytschko, T., Organ, D., Krongauz, Y., 1995. A coupled finite element-element-free galerkin method. *Computational Mechanics* **17**, 186–195. (Cited on page 15)
- Bharti, R. P., Chhabra, R. P., Eswaran, V., 2006. Steady flow of power law fluids across a circular cylinder. *The Canadian Journal of Chemical Engineering* **84**, 406–421. (Cited on pages 5 and 96)
- Bird, R. B., Stewart, W. E., Lightfoot, E. N., 2002. *Transport Phenomena*, 2nd Edition. John Wiley and Sons Inc., New York, NY. (Cited on page 97)
- Blazek, J., 2001. *Computational fluid dynamics: Principals and applications*, 3rd Edition. Elsevier. (Cited on pages 3 and 5)
- Brezzi, F., 1974. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *RAIRO analyse numerical, R2*, 129–151. (Cited on page 109)
- Brooks, A. N., Hughes, T. J. R., 1982. Streamline upwind petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering* **32**, 199–259. (Cited on pages 21 and 84)

- Cai, Y. C., Zhu, H. H., 2004. Direct imposition of essential boundary conditions and treatment of material discontinuities in the EFG method. *Computational Mechanics* **34** (4), 330–338. (Cited on page 19)
- Chen, J.-S., Hillman, M., Chi, S.-W., 2017a. Meshfree methods: progress made after 20 years. *Journal of Engineering Mechanics* **143** (4), 04017001 (1–38). (Cited on pages 7 and 12)
- Chen, L., Liew, K. M., 2011. A local petrov-galerkin approach with moving kriging interpolation for solving transient heat conduction problems. *Computational Mechanics* **47**, 455–467. (Cited on pages 20 and 66)
- Chen, Z. J., Xie, Z. Y. L. W. L., Wu, X. H., 2017b. A two-level variational multiscale meshless local petrov-galerkin (VMS-MLPG) method for convection-diffusion problems with large pecllet number. *Computers and Fluids* **00**, 1–10. (Cited on page 24)
- Chhabra, R. P., 1995. Calculating velocities of particles. *Chemical Engineering* **102**, 133–135. (Cited on page 24)
- Chhabra, R. P., Farooqi, S. I., Khatib, Z., Richardson, J. F., 1982. The cocurrent flow of shear thinning liquids and air in horizontal and vertical pipes. *Journal of Pipelines* **2**, 169–185. (Cited on pages 24 and 96)
- Chhabra, R. P., Richardson, J. F., 1984. Prediction of flow pattern for the co-current flow of gas and non-newtonian liquid in horizontal pipes. *The Canadian Journal of Chemical Engineering* **62**, 169–185. (Cited on page 24)
- Chhabra, R. P., Richardson, J. F., 2008. *Non-Newtonian Flow and Applied Rheology*, 2nd Edition. Butterworth-Heinemann, Oxford, UK. (Cited on pages 24, 96, and 97)
- Chinchapatnam, P. P., Djidjeli, K., Nair, P. B., 2006. Unsymmetric and symmetric meshless schemes for the unsteady convection–diffusion equation. *Computer Methods in Applied Mechanics and Engineering* **195**, 2432–2453. (Cited on page 22)
- Cleveland, W. S., 1993. Visualizing data. (Cited on pages 33 and 39)

- Dalal, A., Das, M. K., 2005. Laminar natural convection in an inclined complicated cavity with spatially variable wall temperature. *International Journal of Heat and Mass Transfer* **48**, 2986–3007. (Cited on page 5)
- Dalal, A., Eswaran, V., Biswas, G., 2008. A finite volume method for navier-stokes equations on unstructured meshes. *Numerical Heat Transfer: Part B: Fundamentals* **54**, 238–259. (Cited on page 5)
- De, A. K., Dalal, A., 2006. Numerical study of natural convection around a square, horizontal, heated cylinder placed in an enclosure. *International Journal of Heat and Mass Transfer* **49**, 4608–4623. (Cited on page 3)
- De, A. K., Dalal, A., 2007. Numerical study of laminar forced convection fluid flow and heat transfer from a triangular cylinder placed in a channel. *ASME: Journal of Heat Transfer* **129**, 646–656. (Cited on page 5)
- De, S., Bathe, K. J., 2000. The method of finite spheres. *Computational Mechanics* **25**, 329–45. (Cited on pages 17, 18, and 54)
- De, S., Bathe, K. J., 2001a. The method of finite spheres with improved numerical integration. *Computers and Structures* **79**, 2183–2196. (Cited on pages 17 and 54)
- De, S., Bathe, K. J., 2001b. Towards an efficient meshless computational technique: the method of finite spheres. *Engineering Computations* **18**, 170–192. (Cited on page 54)
- Dhole, S. D., Chhabra, R. P., Eswaran, V., 2006. Forced convection heat transfer from a sphere to non- newtonian power law fluids. *American Institute of Chemical Engineers Journal* **52**, 3658–3667. (Cited on pages 5 and 96)
- Dolbow, J., Belytschko, T., 1998. An introduction to programming the meshless element free galerkin method. *Archives of computational methods in engineering* **5**, 207–241. (Cited on page 37)

- Duarte, C. A., Oden, J. T., 1996a. Hp clouds - a meshless method to solve boundary-value problems. *Numerical methods for partial differential equations* **12**, 1–33. (Cited on pages 6, 15, and 54)
- Duarte, C. A., Oden, J. T., 1996b. Hp clouds: A h-p meshless method. *Numerical methods for partial differential equations* **12**, 673–705. (Cited on pages 6 and 15)
- Enjilela, V., Arefmanesh, A., 2015. Two-step taylor-characteristic-based MLPG method for fluid flow and heat transfer applications. *Engineering Analysis with Boundary Elements* **51**, 174–190. (Cited on page 24)
- Franca, L. P., Frey, S. L., 1992. Stabilized finite element methods: Ii. the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering* **99**, 209–233. (Cited on pages 101 and 110)
- Fries, T. P., Matthies, H. G., 2004. Classification and overview of meshfree methods. *Scientific Computing* . (Cited on pages 12, 17, and 49)
- Fries, T. P., Matthies, H. G., 2006a. A stabilized and coupled meshfree/meshbased method for the incompressible navier-stokes equations—part ii:coupling. *Computer Methods in Applied Mechanics and Engineering* **195**, 6191–204. (Cited on page 22)
- Fries, T. P., Matthies, H. G., 2006b. A stabilized and coupled meshfree/meshbased method for the incompressible navier-stokes equations—part i:stabilization. *Computer Methods in Applied Mechanics and Engineering* **195**, 6205–24. (Cited on page 22)
- Gavete, L., Benito, J. J., Falco, S., Ruiz, A., 2000. Implementation of essential boundary conditions in a meshless method. *International Journal for Numerical Methods in Biomedical Engineering* **16**, 409–421. (Cited on page 49)
- Gingold, R. A., Monaghan, J. J., 1977. Smoothed particle hydrodynamics, theory and applications to non-spherical stars. *Monthly notices of the royal astronomical society* **181**, 375–389. (Cited on pages iii, 6, 13, and 21)

-
- Gu, Y. T., Liu, G. R., 2001a. A coupled element free galerkin-boundary element method for stress analysis of two dimensional solids. *Computer methods in applied mechanics and engineering* **190**, 4405–4419. (Cited on pages 48 and 49)
- Gu, Y. T., Liu, G. R., 2001b. A meshless local petrov-galerkin (MLPG) formulation for static and free vibration analysis of thin plates. *CMES: Computer Modeling in Engineering & Sciences* **2** (4), 463–476. (Cited on page 8)
- Gunther, F., Liu, W. K., Diachin, D., Christon, M. A., 2000. Multi-scale meshfree parallel computations for viscous, compressible flows. *Computer Methods in Applied Mechanics and Engineering* **190**, 279–303. (Cited on page 22)
- Gunther, F. C., Liu, W. K., 1998. Implementation of boundary conditions for meshless methods. *Computer Methods in Applied Mechanics and Engineering* **163**, 205–230. (Cited on page 49)
- Gupta, A. K., Sharma, A., Chhabra, R. P., Eswaran, V., 2003. Two dimensional steady flow of a power law fluid past a square cylinder in a plane channel: Momentum and heat transfer characteristics. *Industrial and Engineering Chemistry Research* **42** (22), 5674–5686. (Cited on pages 3 and 96)
- Gupta, R. K., Chhabra, R. P., 1996. Laminar flow of power law fluids in concentric annuli. *American Institute of Chemical Engineers Journal* **42**, 2080–2083. (Cited on pages 24 and 96)
- Han, Z. D., Atluri, S. N., 2004a. A meshless local petrov-galerkin (MLPG) approach for 3-dimensional elasto-dynamics. *CMC: Computers, Materials & Continua* **1**, 129–140. (Cited on page 8)
- Han, Z. D., Atluri, S. N., 2004b. Meshless local petrov-galerkin (MLPG) approaches for solving 3d problems in elasto-statics. *CMES: Computer Modeling in Engineering & Sciences* **6**, 169–188. (Cited on page 8)

-
- Hong, W. X., Quan, T. W., 2008. Meshless method based on the local weak-forms for steady-state heat conduction problems. *International Journal of Heat and Mass Transfer* **51**, 3103–3112. (Cited on pages 20 and 66)
- Hughes, T. J. R., Brooks, A., 1979. A multidimensional upwind scheme with no crosswind diffusion. *Finite element methods for convection dominated flows* **34**, 19–35. (Cited on pages 21 and 84)
- Jayanti, S., Hewitt, G. F., 1992. A numerical study of bifurcation in laminar flow in curved ducts. *International Journal for Numerical Methods in Fluids* **14**, 253–266. (Cited on page 25)
- Jayanti, S., Hewitt, G. F., Kightley, J. R., 1990. Fluid flow in curved ducts. *International Journal for Numerical Methods in Fluids* **10**, 569–589. (Cited on page 25)
- Kansa, E. J., 1990a. Multiquadrics- a scattered data approximation scheme with applications computational fluid dynamics-i. *Computers & mathematics with applications* **19**, 127–145. (Cited on page 14)
- Kansa, E. J., 1990b. Multiquadrics- a scattered data approximation scheme with applications to computational fluid dynamics-ii. *Computers & mathematics with applications* **19**, 147–161. (Cited on page 14)
- Krongauz, Y., Belytschko, T., 1996. Enforcement of essential boundary conditions in meshless approximations using finite element. *Computer methods in applied mechanics and engineering* **131**, 133–145. (Cited on pages 46 and 49)
- Kumar, S. R., Sharma, A., Agrawal, A., 2008. Simulation of flow around a row of square cylinders. *Journal of Fluid Mechanics* **606**, 369–397. (Cited on page 5)
- Lancaster, P., Salkauskas, K., 1981. Surface generated by moving least squares methods. *Mathematics of computation* **37**, 141–158. (Cited on pages 15, 35, 39, and 40)
- Lancaster, P., Salkauskas, K., 1986. *Curve and surface fitting, an introduction*. Academic press. (Cited on pages iv, 15, 33, 35, and 39)

-
- Lei, C., Cheng, L., Kavanagh, K., 2000. A finite difference solution of the shear flow over a circular cylinder. *Ocean Engineering* **27** (3), 271–290. (Cited on page 3)
- Leonard, B. P., 1979. A survey of finite difference of opinion on numerical muddling of the incomprehensible defective confusion confusion equation. *Finite Element Methods for Convection Dominated Flows* **34**. (Cited on pages xii and 83)
- Libersky, L. D., Petscheck, A. G., Carney, T. C., Hipp, J. R., 1993. High strain lagrangian hydrodynamics. a three-dimensional sph code for dynamic material response. *journal of computational physics* **109**, 67–75. (Cited on pages 6 and 13)
- Lin, H., Atluri, S. N., 2000. Meshless local petrov-galerkin (MLPG) method for convection-diffusion problems. *CMES: Computer Modeling in Engineering & Sciences* **1**, 45–60. (Cited on pages v, xii, 8, 22, 23, 83, 84, 85, 86, 87, 88, 90, 96, 109, and 117)
- Lin, H., Atluri, S. N., 2001. The meshless local petrov-galerkin(MLPG) method for solving incompressible navier-stokes equations. *CMES: Computer Modeling in Engineering & Sciences* **2**, 117–142. (Cited on pages 8, 22, 23, 96, 109, 110, and 119)
- Liu, G. R., 2003a. *Meshfree Methods, moving beyond the finite element method*. Taylor & Francis. (Cited on pages iii, iv, xi, 11, 31, 32, 33, 37, 46, 47, and 56)
- Liu, G. R., Gu, Y. T., 1999. A point interpolation method. *Proceedings of 4th Asia Pacific Conference on Computational Mechanics, Singapore*, 1009–1014. (Cited on page 16)
- Liu, G. R., Gu, Y. T., 2001a. A local point interpolation method for stress analysis of two-dimensional solids. *International Journal of Structural Engineering and Mechanics* **11**, 221–236. (Cited on pages 16, 17, and 18)
- Liu, G. R., Gu, Y. T., 2001b. A local radial point interpolation method(LRPIM) for free vibration analyses of 2-D solids. *Journal of Sound Vibration* **246**, 29–46. (Cited on page 18)

-
- Liu, G. R., Gu, Y. T., 2002a. Point interpolation method based on local residual formulation using radial basis functions. *Structural Engineering and Mechanics* **14**, 713–732. (Cited on page 18)
- Liu, G. R., Gu, Y. T., 2002b. A truly meshless method based on the strong-weak form. advance in meshfree and X-FEM methods. *Proceeding of the 1st Asian workshop in meshfree methods. Singapore*, 259–261. (Cited on page 14)
- Liu, G. R., Gu, Y. T., 2003a. A matrix triangularization algorithm for point interpolation method. *Computer Methods in Applied Mechanics and Engineering* **192**, 2269–2295. (Cited on page 16)
- Liu, G. R., Gu, Y. T., 2003b. A meshfree method: Meshfree weak-strong (MWS) form method, for 2-D solids. *Computational Mechanics* **33**, 2–14. (Cited on page 14)
- Liu, G. R., Gu, Y. T., 2004. Boundary mesh-free methods based on the boundary point interpolation methods. *Engineering analysis with boundary elements* **28**, 475–487. (Cited on pages 49 and 52)
- Liu, G. R., Gu, Y. T., 2005. *An Introduction to Meshfree Methods and their Programming*. Springer Science & Business Media. (Cited on pages xi, 21, 34, 35, 38, 39, 45, 46, 52, and 55)
- Liu, G. R., Wu, Y. L., Ding, H., 2004. Meshfree weak strong (MWS) form method and its application to incompressible flow problems. *International journal for numerical methods in fluids* **46**, 1025–47. (Cited on page 14)
- Liu, W. K., Jun, S., 1995. Reproducing kernel particle methods. *International journal for numerical methods in fluids* **20**, 1081–1106. (Cited on pages 6, 15, and 54)
- Liu, W. K., Jun, S., Li, S., Adee, J., Belytschko, T., 1995b. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering* **38**, 1655–1679. (Cited on pages 6 and 15)

- Liu, X.-L., 2003b. A finite element/nodal volume technique for flow simulation of injection pultrusion. *Composites Part A: Applied Science and Manufacturing* **34** (7), 649–661. (Cited on page 4)
- Long, S., Atluri, S. N., 2002. A meshless local petrov-galerkin method for solving the bending problem of a thin plate. *CMES: Computer Modeling in Engineering & Sciences* **3**, 53–63. (Cited on page 8)
- Lucy, L. B., 1977. A numerical approach to the testing of the fission hypothesis. *The astronomical journal* **82**, 1013–1024. (Cited on pages iii, 6, and 13)
- Marosevic, T., 1996. A choice of norm in discrete approximation. *Mathematical Communications* **1** (2), 147–152. (Cited on page 40)
- Mazzia, A., Ferronato, M., Pini, G., Gambolati, G., 2007. A comparison of numerical integration rules for the meshless local petrov–galerkin method. *Numerical Algorithms* **45**, 61–74. (Cited on page 54)
- Mazzia, A., Pini, G., Sartoretto, F., 2014. Meshless techniques for anisotropic diffusion. *Applied Mathematics and Computation* **236**, 554–66. (Cited on page 20)
- McDonald, P. W., 1971. The computation of transonic flow through two-dimensional gas turbine cascades. *American Society of Mechanical Engineers* **71**, 71–GT–89. (Cited on page 4)
- Melenk, J. M., Babuska, I., 1996. The partition of unity finite element methods: Basic theory and applications. *Computer methods in applied mechanics and engineering* **139**, 289–314. (Cited on page 16)
- Mendez, S. F., Huerta, A., 2004. Imposing essential boundary conditions in mesh-free methods. *Computer methods in applied mechanics and engineering* **193**, 1257–1275. (Cited on page 49)

-
- Mittal, S., Kumar, V., 1999. Finite element study of vortex-induced cross-flow and in-line oscillations of a circular cylinder at low reynolds numbers. *International Journal for Numerical Methods in Fluids* **31** (7), 1087–1120. (Cited on page 4)
- Mohammadi, M. H., 2008. Stabilized meshless local petrov-galerkin (MLPG) method for incompressible viscous fluid flows. *CMES: Computer Modeling in Engineering & Sciences* **29**, 75–94. (Cited on page 23)
- Mousavi, S. E., Xiao, H., Sukumar, N., 2010. Generalized gaussian quadrature rules on arbitrary polygons. *International Journal for Numerical Methods in Engineering* **82**, 99–113. (Cited on page 53)
- Mukherjee, Y. X., Mukherjee, S., 1997a. The boundary node method for potential problems. *International Journal for Numerical Methods in Engineering* **40**, 797–815. (Cited on pages 6 and 17)
- Mukherjee, Y. X., Mukherjee, S., 1997b. On boundary conditions in the element-free galerkin method. *Computational Mechanics* **19**, 264–270. (Cited on page 49)
- Najafi, M., Arefmanesh, A., Enjilela, V., 2012. Meshless local petrov–galerkin method–higher reynolds numbers fluid flow applications. *Engineering Analysis with Boundary Elements* **36**, 1671–1685. (Cited on page 23)
- Najafi, M., Arefmanesh, A., Enjilela, V., 2013. Extending MLPG primitive variable-based method for implementation in fluid flow and natural, forced and mixed convection heat transfer. *Engineering Analysis with Boundary Elements* **37**, 1285–1299. (Cited on page 23)
- Najafi, M., Enjilela, V., 2014. Natural convection heat transfer at high rayleigh numbers – extended meshless local petrov–galerkin (MLPG) primitive variable method. *Engineering Analysis with Boundary Elements* **44**, 170–184. (Cited on page 23)
- Nayroles, B., Touzot, G., Villon, P., 1992. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics* **10**, 307–318. (Cited on pages 6, 15, 39, and 110)

-
- Neumann, J. V., Goldstine, H., 1947. Numerical inverting of matrices of high order. *Bulletin of the American Mathematical Society* **53**, 1021–1099. (Cited on page 1)
- Nguyen, V. P., Rabczuk, T., Bordas, S. P., Dufloy, M., 2008. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation* **79**, 763–813. (Cited on pages 12 and 17)
- Oden, J. T., Durate, C. A., Zienkiewicz, O. C., 1998. A new cloud-based hp finite element method. *Computer methods in applied mechanics and engineering* **153**, 117–126. (Cited on page 54)
- Onate, E., Idelsohn, S., Zienkiewicz, O. C., Taylor, R. L., 1996a. A finite point method in computational mechanics: Applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering* **39**, 3839–3866. (Cited on pages 6, 13, 15, and 21)
- Onate, E., Idelsohn, S., Zienkiewicz, O. C., Taylor, R. L., 1996b. A stabilized finite point method for analysis of fluid mechanics problems. *Computer methods in applied mechanics and engineering* **139**, 315–346. (Cited on pages 13 and 21)
- Paliwal, B., Sharma, A., Chhabra, R. P., Eswaran, V., 2003. Power-law flow past a square cylinder: momentum and heat transfer characteristics. *Chemical Engineering Science* **58**, 5315–5329. (Cited on pages 3 and 25)
- Perrin, A., Hu, H., 2006. An explicit finite-difference scheme for simulation of moving particles. *Journal of Computational Physics* **212** (1), 166–187. (Cited on page 3)
- Pierre, R., 1988. Simple c^0 approximation for the computation of the incompressible flows. *Computer methods in applied mechanics and engineering* **68**, 205–227. (Cited on pages 101 and 110)
- Price, H. S., Cavendish, J. C., Varga, R. S., 1968. Numerical methods of higher-order accuracy for convection-diffusion equations. *Society of Petroleum Engineers Journal* **8**, 293–303. (Cited on page 4)

- Qian, L. F., Batra, R. C., 2005. Three dimensional transient heat conduction in a functionally graded thick plate with a high order plate theory and a meshless local petrov galerkin method. *Computational Mechanics* **35**, 214–226. (Cited on pages 20 and 66)
- Ramirez, J. F. S., Meas, E. M., Branch, J. W. B., boulangier, P., 2011. A comparison between lagrange multiplier and penalty methods to enforce essential boundary conditions in meshfree methods. *Revista Avances en Sistemas e-Informatica* **8**, 51–56. (Cited on page 49)
- Ranjan, R., Dalal, A., Biswas, G., 2008. A numerical study of fluid flow and heat transfer around a square cylinder at incidence using unstructured grids. *Numerical Heat Transfer, Part A: Applications* **54**, 890–913. (Cited on page 5)
- Sadat, H., Couturier, S., 2000. Performance and accuracy of a meshless method for laminar natural convection. *Numerical Heat Transfer: Part B: Fundamentals* **37**, 455–467. (Cited on page 21)
- Sadat, H., Prax, C., 1996. Application of the diffuse approximation for solving fluid flow and heat transfer problems. *International journal of heat and mass transfer* **39**, 214–218. (Cited on page 21)
- Sahu, A., Chhabra, R. P., Eswaran, V., 2010. Two-dimensional laminar flow of a power-law fluid over a confined square cylinder. *Journal of Non-Newtonian Fluid Mechanics* **165**, 752–763. (Cited on pages 5 and 96)
- Sarkar, M., Paramane, S. B., Sharma, A., 2017. Periodically fully developed heat and fluid flow characteristics in a furrowed wavy channel. *Heat Transfer Engineering* **38**, 278–288. (Cited on page 5)
- Sataprahma, C., Luadsong, A., 2014. The meshless local petrov-galerkin method for simulating unsteady incompressible fluid flow. *Journal of the Egyptian Mathematical Society* **22**, 501–510. (Cited on pages 23 and 96)

- Shamekhi, A., Sadeghy, K., 2009. Cavity flow simulation of carreau–yasuda non-newtonian fluids using PIM meshfree method. *Applied Mathematical Modeling* **33**, 4131–4145. (Cited on pages 25 and 96)
- Sharma, A., 2016. *Introduction to computational fluid dynamics: development, application and analysis*. Wiley. (Cited on pages 3 and 5)
- Singh, I. V., 2004. A numerical solution of composite heat transfer problems using meshless method. *International Journal of Heat and Mass Transfer* **47**, 2133–2138. (Cited on pages 37 and 66)
- Sladeka, J., Sladeka, V., Atluri, S. N., 2004. Meshless local petrov–galerkin method for heat conduction problem in an anisotropic medium. *CMES: Computer Modeling in Engineering & Sciences* **6**, 309–318. (Cited on pages 20 and 65)
- Sladeka, J., Sladeka, V., Hon, Y. C., 2006. Inverse heat conduction problems by meshless local petrov–galerkin method. *Engineering Analysis with Boundary Elements* **30**, 650–661. (Cited on pages 20 and 66)
- Sladeka, J., Sladeka, V., Tan, C. L., Atluri, S. N., 2008. Analysis of transient heat conduction in 3D anisotropic functionally graded solids, by the MLPG method. *CMES: Computer Modeling in Engineering & Sciences* **32**, 161–174. (Cited on pages 20 and 66)
- Sladeka, V., Sladeka, J., Atluri, S. N., Keer, R. V., 2000. Numerical integration of singularities in meshless implementation of local boundary integral equations. *Computational Mechanics* **25**, 394–403. (Cited on page 19)
- Slater, J. C., 1934. Electronic energy bands in metals. *Physical Review* **45**, 794–801. (Cited on page 13)
- Slater, J. C., 2008. Meshless solution of a diffusion equation with parameter optimization and error analysis. *Engineering Analysis with Boundary Elements* **32**, 567–577. (Cited on page 20)

-
- Swegle, J. W., Hicks, D. L., Attaway, S. W., 1995. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics* **116**, 123–134. (Cited on pages 6 and 13)
- Tain, F. B., 2014. FSI modeling with the DSD/SST method for the fluid and finite difference method for the structure. *Computational Mechanics* **54** (2), 581–589. (Cited on page 3)
- Tezduyar, T. E., Osawa, Y., 2000. Finite element stabilization parameters computed from element matrices and vectors. *Computer methods in applied mechanics and engineering* **190**, 411–430. (Cited on page 110)
- Thakur, H., Singh, K. M., Sahoo, P. K., 2009. Meshless local petrov-galerkin method for nonlinear heat conduction problems. *Numerical Heat Transfer, Part B* **56**, 393–410. (Cited on pages 20 and 66)
- Tian, F. B., 2014. FSI modeling with the DSD/SST method for the fluid and finite difference method for the structure. *Computational Mechanics* **54**, 581–589. (Cited on page 25)
- Tian, F. B., Bharti, R. P., Xu, Y. Q., 2014a. Deforming-spatial-domain/stabilized space-time (DSD/SST) method in computation of non-newtonian fluid flow and heat transfer with moving boundaries. *Computational Mechanics* **53**, 257–271. (Cited on page 25)
- Tian, F. B., Bharti, R. P., Xu, Y. Q., 2014b. Deforming-spatial-domain/stabilized space-time (DSD/SST) method in computation of non-newtonian fluid flow and heat transfer with moving boundaries. *Computational Mechanics* **53** (2), 257–271. (Cited on page 4)
- Tian, F. B., Zhu, L., Fok, P. W., Lu, X. Y., 2013. Simulation of a pulsatile non-newtonian flow past a stenosed 2D artery with atherosclerosis. *Computers in biology and medicine* **43** (9), 1098–1113. (Cited on page 4)

-
- Tian, J., Rao, S. S., 2012. Meshless local petrov-galerkin method for three-dimensional heat transfer analysis. *Journal of Heat Transfer* **134**, 112701–112709. (Cited on pages 21 and 66)
- Trobec, R., Kosec, G., Sterk, M., Sarler, B., 2012. Comparison of local weak and strong form meshless methods for 2-d diffusion equation. *Engineering Analysis with Boundary Elements* **36**, 310–321. (Cited on page 13)
- Turner, M. J., Clough, R. W., Martin, H. C., Topp, L. J., 1956. Stiffness and deflection analysis of complex structures. *journal of the Aeronautical Sciences* **23**, 805–823. (Cited on page 3)
- Verma, A. K., Eswaran, V., 1996. Overlapping control volume method for convective-diffusive problems. *International journals for numerical methods in fluids* **23**, 865–882. (Cited on page 25)
- Verma, A. K., Eswaran, V., 1997. Bounded convection scheme for the overlapping control volume approach. *International journals for numerical methods in fluids* **25**, 1137–1161. (Cited on page 25)
- Versteeg, H. K., Malalasekera, W., 2007. *An introduction to computational fluid dynamics: the finite volume method*, 2nd Edition. Longman Scientific and Technical, New York, NY. (Cited on pages iv, 71, 75, and 91)
- Wan, D., Turek, S., 2007. An efficient multigrid-FEM method for the simulation of solid-liquid two phase flows. *Journal of Computational and Applied Mathematics* **203** (2), 561–580. (Cited on page 4)
- Wang, H., Qin, Q. H., Kang, Y. L., 2005. A new meshless method for steady-state heat conduction problems in anisotropic and inhomogeneous media. *Archive of Applied Mechanics* **74**, 563–579. (Cited on pages 8 and 65)
- Wang, J. G., Liu, G. R., 2000. Radial point interpolation method for elastoplastic problems. *ICSSD 2000: 1 st Structural Conference on Structural Stability and Dynamics*, 703–708. (Cited on page 16)

- Wang, J. G., Liu, G. R., 2002. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering* **54** (11), 1623–1648. (Cited on page 16)
- Wang, W. Q., Tian, Y. Y. F. B., 2017. A simple and efficient implicit direct forcing immersed boundary model for simulations of complex flow. *Applied mathematical modeling* **43**, 287–305. (Cited on pages 5 and 25)
- Wang, W. Q., Yan, Y., Tian, F. B., 2017. A simple and efficient implicit direct forcing immersed boundary model for simulations of complex flow. *Applied Mathematical Modeling* **43**, 287–305. (Cited on page 5)
- Wang, X., Dalton, C., 1991. Oscillating flow past a rigid circular cylinder: A finite-difference calculation. *Journal of Fluids Engineering* **113** (3), 377–383. (Cited on page 3)
- Wu, X. H., Dai, Y. J., Tao, W. Q., 2012. MLPG/SUPG method for convection-dominated problems. *Numerical Heat Transfer, Part B* **61**, 36–51. (Cited on page 23)
- Wu, X. H., Tao, W. Q., Shen, S. P., Zhu, X. W., 2010. A stabilized mlpg method for steady state incompressible fluid flow simulation. *journal of computational physics* **229**, 8564–8577. (Cited on pages 23, 96, 109, and 110)
- Wu, Y. L., Liu, G. R., 2003. A meshfree formulation of local radial point interpolation method (LRPIM) for incompressible flow simulation. *Computational Mechanics* **30**, 355–65. (Cited on page 22)
- Wu, Y. L., Liu, G. R., Gu, Y. T., 2005. Application of meshless local petrov galerkin (MLPG) approach to simulation of incompressible flow. *Numerical Heat Transfer , Part B* **48**, 459–475. (Cited on page 23)
- Wu, Z., 1992. Hermite-birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications* **8**, 1–10. (Cited on page 14)
- Zahiri, S., 2011. *Meshfree Methods, Numerical Analysis-Theory and Application*. INTECH Open Access Publisher. (Cited on page 13)

-
- Zhang, X. K., Kwon, K. C., Youn, S. K., 2005. The least-squares meshfree method for the steady incompressible viscous flow. *Journal of Computational Physics* **206**, 182–207. (Cited on page 22)
- Zhao, Y., Sharp, M., 1999. Finite element analysis of the lift on a slightly deformable and freely rotating and translating cylinder in two-dimensional channel flow. *Journal of Biomechanical Engineering* **121** (2), 148–152. (Cited on page 4)
- Zhao, Y., Sharp, M., 2000. Stability of elliptical cylinders in two-dimensional channel flow. *Journal of Biomechanical Engineering* **122** (5), 493–497. (Cited on page 4)
- Zheng, H. B., Hou, Y. R., Shi, F., Song, L. N., 2009. A finite element variational multiscale method for incompressible flows based on two local gauss integration's. *Journal of Computational Physics* **228**, 5961–71. (Cited on page 24)
- Zhu, T., Zhang, J., Atluri, S. N., 1998a. A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. *Computational Mechanics* **22**, 174–186. (Cited on pages 6 and 19)
- Zhu, T., Zhang, J. D., Atluri, S. N., 1998b. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. *Computational Mechanics* **21**, 223–235. (Cited on pages 6, 17, and 18)
- Zhu, T. L., 2010. The nonlinear meshless local petrov-galerkin (MLPG) method from the nonlinear regular local boundary integral equation. *International Journal for Computational Methods in Engineering Science and Mechanics* **11**, 123–132. (Cited on pages 19 and 49)
- Zienkiewicz, O. C., Taylor, R. L., 2000. *The Finite Element Method Volume 3: Fluid Dynamics*, 5th Edition. Butterworth-Heinemann, Barcelona. (Cited on pages 21 and 83)