

**UNIVERSITY OF GAZIANTEP**  
**GRADUATE SCHOOL OF NATURAL & APPLIED**  
**SCIENCES**

**AN OBJECT ORIENTED PROGRAM OF MESHFREE**  
**METHODS FOR ANALYZING OF 2D AND 3D**  
**ELASTOSTATIC PROBLEMS**

**MECHANICAL ENGINEERING**

**Ph.D. THESIS**

**MUSTAFA MURAT YAVUZ**

**SEPTEMBER 2015**

**An Object Oriented Program of Meshfree Methods for  
Analyzing of 2D and 3D Elastostatic Problems**

**University of Gaziantep  
Mechanical Engineering  
Ph.D. Thesis**

**Supervisor  
Prof.Dr. Bahattin KANBER**

**Mustafa Murat YAVUZ  
September 2015**

© 2015 [Mustafa Murat YAVUZ]

T.C.  
UNIVERSITY OF GAZİANTEP  
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES  
MECHANICAL ENGINEERING DEPARTMENT

Name of the thesis: An Object Oriented Program of Meshfree Methods for  
Analyzing of 2D and 3D Elastostatic Problems

Name of the student: Mustafa Murat YAVUZ

Exam date: 11/09/2015


Approval of the Graduate School of Natural and Applied Sciences

  
Prof. Dr. Metin BEDİR  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of  
Doctor of Philosophy.

  
Prof. Dr. Sait SÖYLEMEZ  
Head of Department

This is to certify that I have read this thesis and it is fully adequate, in scope and  
quality, as a thesis for the degree of Doctor of Philosophy.

  
Prof. Dr. Bahattin KANBER  
Supervisor

This is to certify that we have read this thesis and that in our consensus/majority  
opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor  
of Philosophy.

Examining Committee Members:

signature

Prof. Dr. İbrahim H. GÜZELBEY

Prof. Dr. Alaattin AKTAŞ

Prof. Dr. Bahattin KANBER

Assoc. Prof. Dr. Serkan NOHUT

Assist. Prof. Dr. Ö. Yavuz BOZKURT



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

**Mustafa Murat YAVUZ**

**ABSTRACT**  
**AN OBJECT ORIENTED PROGRAM OF MESHFREE METHODS FOR**  
**ANALYZING OF 2D AND 3D ELASTOSTATIC PROBLEMS**

YAVUZ, Mustafa Murat

Ph.D. in Mechanical Engineering

Supervisor: Prof. Dr. Bahattin KANBER

September 2015

153 pages

In this study, several object oriented and structured programs are developed using Matlab and Fortran programming languages. In these programs, various meshfree methods are used such as LRPIM, MLPG and RPIM with different integration schemes. The performance of each program, efficiency of integration schemes and shape parameters of RPIM are investigated in details.

In the first part of this study, LRPIM and MLPG methods are used inside an object oriented program to test the classes and objects. Developed program is applied in the solutions of 2D and 3D elasto-static problems. The modularity of the program is main target in this part of the study.

In the second part, a nodal integration scheme is used in RPIM with a structured program for solution of 2D and 3D elasto-static problems. Higher order terms of Taylor series expansion are used for increasing solution accuracy. Various integration domain shapes are used for integrations of regular and irregular distributed nodes in the problem domains. Optimum shape parameters of RPIM and support domain sizes are determined.

In the last part, stress fluctuations in nodal integration results have been decreased with three different applications.

**Key Words:** Object oriented programming (OOP), meshfree methods, RPIM, nodal integration, Matlab

## ÖZET

### 2 VE 3 BOYUTLU ELASTOSTATİK PROBLEMLERİN ANALİZİ İÇİN AĞSIZ YÖNTEMLERİN NESNE TABANLI BİR PROGRAMI

YAVUZ, Mustafa Murat

Doktora Tezi, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Bahattin KANBER

Eylül 2015

153 sayfa

Bu çalışmada, Matlab ve Fortran programlama dilleri kullanılarak çeşitli nesne tabanlı ve yapısal programlar geliştirilmiştir. Bu programlar içerisinde LRPIM, MLPG ve RPIM gibi çeşitli ağsız yöntemler farklı integrasyon şemaları ile kullanılmıştır. Her programın performansı, integrasyon şeması ve RPIM şekil parametrelerinin verimliliği detaylı olarak araştırılmıştır.

Çalışmanın ilk kısmında, LRPIM ve MLPG yöntemleri bir nesne tabanlı program içerisinde sınıfları ve nesnelere test etmek için kullanılmıştır. Geliştirilen program 2 ve 3 boyutlu elasto-statik problemlerin çözümüne uygulanmıştır. Programın modülerliği bu bölümün ana hedefidir.

İkinci bölümde, noktasal integrasyon yöntemi yapısal bir program ile 2 ve 3 boyutlu elasto-statik problemlerin çözümü için RPIM içerisinde kullanılmıştır. Yüksek dereceli Taylor serisi açılımı terimleri çözüm hassasiyetinin artırılması için kullanılmıştır. Problem alanı içerisinde düzenli ve düzensiz dağılmış noktaların integrasyonu için çeşitli integrasyon alanı şekilleri kullanılmıştır. Uygun RPIM şekil parametreleri ve destek alanı büyüklüğü belirlenmiştir.

Son bölümde, noktasal integrasyon sonuçları içerisindeki gerilme dalgalanmaları 3 farklı yöntem kullanılarak azaltılmıştır.

**Anahtar Kelimeler:** Nesne yönetimli programlama, ağsız yöntemler, RPIM, noktasal integrasyon, Matlab

## **ACKNOWLEDGEMENTS**

I express sincere appreciation to my supervisor Prof. Dr. Bahattin Kanber for all helping. I cannot complete this thesis without his support and effort. His advices provide to continue and prevent plugging of my thesis in all steps.

I want to thank my family; my parents Nebi and Selma Yavuz, raised and always help me. They support morale and everything all my life.



## CONTENTS

	Page
ABSTRACT .....	v
ÖZET.....	vi
ACKNOWLEDGEMENTS .....	vii
CONTENTS .....	viii
LIST OF TABLES .....	xi
LIST OF FIGURES .....	xii
LIST OF SYMBOLS/ABBREVIATIONS .....	xix
1. INTRODUCTION .....	1
1.1 General Introduction .....	1
1.2 Research Objectives and Tasks .....	2
1.3 Layout of Thesis.....	3
2. LITERATURE REVIEW.....	4
2.1 Introduction .....	4
2.2 Meshfree Methods.....	4
2.2.1 Solid Mechanics .....	5
2.2.2 Large Deformation .....	7
2.2.3 Crack and Fracture .....	7
2.2.4 Thin and Thick Plates.....	8
2.2.5 Composites .....	9
2.2.6 Contact .....	10
2.2.7 Impact.....	11
2.2.8 Fluid Mechanics and Heat Transfer .....	11
2.2.9 Developments on Applications .....	12
2.2.10 Radial Basis Functions and Other Applications.....	13
2.3 Stress Stabilization on Meshfree Methods .....	15
2.4 OOP in Numerical Methods.....	16
2.5 Conclusion of Literature Review .....	22
3. MESHFREE INTERPOLATION METHODS .....	25
3.1 Introduction .....	25
3.2 Moving Least Square Interpolation Method .....	25

3.3 Radial Point Interpolation Method.....	28
4. OBJECT-ORIENTED PROGRAMMING IN MESHFREE ANALYSIS .....	34
4.1 Introduction .....	34
4.2 Object-oriented Programming (OOP) and Matlab .....	34
4.3 Meshfree Classes and Their Implementations in a Matlab Program .....	35
4.3.1 mc (Model Constructor) Class .....	37
4.3.2. inte (Integration Cell) Class.....	39
4.3.3. sdo (Support Domain) Class.....	40
4.3.4 sf (Shape Function) Class.....	42
4.3.5 bc (Boundary Condition) Class .....	49
4.3.6 solv (Solution) Class .....	51
4.3.7 post (Results) Class .....	53
4.4 Example Solution and Discussions .....	55
4.4.1 Key Design Principles and Solution Procedure .....	55
4.4.2 Solutions and Discussion .....	59
4.5 Conclusion.....	61
5. NODAL INTEGRATION WITH TAYLOR SERIES EXPANSION .....	63
5.1 Introduction .....	63
5.2 2D Nodal Integration Scheme .....	64
5.3 3D Nodal Integration Scheme .....	67
6. INTEGRATIONS WITH VARIOUS TYPES OF CELLS.....	71
6.1 Introduction .....	71
6.2 Rectangular Shaped Taylor Integration Cells for 2D Cases .....	71
6.2.1 Numerical Examples.....	71
6.2.1.1 Axial Loaded Bar .....	73
6.2.1.2 Cantilever Beam .....	77
6.2.1.3 Simply Supported Beam.....	79
6.2.2 Results and Discussions.....	82
6.3 Triangular Shaped Taylor Integration Cells for 2D Cases.....	83
6.4 Hexahedral Shaped Taylor Integration Cells for 3D Cases .....	86
6.4.1 Numerical Examples.....	86
6.4.1.1 Torsion Bar .....	89
6.4.1.2 Simply Supported Beam.....	94
6.4.1.3 L-Shaped Beam .....	100
6.4.2 Results and Discussions.....	106

6.5 Tetrahedral Shaped Taylor Integration Cells for 3D Cases .....	107
6.5.1 Numerical Examples .....	111
6.5.1.1 Axial Loaded Bar .....	114
6.5.1.2 Cantilever Beam .....	117
6.5.1.3 Torsion of Cylindrical Shaft.....	121
6.5.1.4 Comparison of Solution Times.....	122
6.5.2 Results and Discussions .....	124
7. METHODS OF STRESS STABILIZATION .....	126
7.1 Introduction .....	126
7.2 Average Stress Distribution for Each Local Support Domain .....	126
7.3 Stiffness Effect on Boundary Locations .....	126
7.4 Least Square Stabilizations .....	127
7.5 Numerical Examples .....	127
7.6 Results and Discussions .....	129
7.7 Conclusions .....	133
CONCLUSIONS.....	135
REFERENCES.....	137
CURRICULUM VITAE .....	153

## LIST OF TABLES

	Page
Table 6.1. Comparison of solution times of RPIM with nodal and Gaussian integration techniques for axial loaded beam models of 33 and 1111 nodes.....	76
Table 6.2. Components of edges for subdivided Taylor Integration cells for each node (triangular geometry).....	84
Table 6.3. Direction of triangular cell and used bounds of integral in natural coordinates. ....	85
Table 6.4. Comparison of shear stress results of a 3D non-circular bar under torsion (case I) for analytical, FEM and nodal int. techniques .....	90
Table 6.5. Comparison of effect of $\alpha_c$ on shear stress results of a 3D non-circular bar under torsion (case I) for nodal int. (2 <sup>nd</sup> order terms) and 4×4 Gauss int.....	93
Table 6.6. Comparison of effect of sd on shear stress results of a 3D non-circular bar under torsion (case I) for nodal int. (2 <sup>nd</sup> order terms) and 4×4 Gauss int.....	93
Table 6.7. Comparison of effect of $q$ on shear stress results of a 3D non-circular bar under torsion (case I) for nodal int. (2 <sup>nd</sup> order terms) and 4×4 Gauss int.....	93
Table 6.8. Comparison of solution times of RPIM with nodal and Gauss integration techniques for simply supported beam model of 189 nodes.....	100
Table 6.9. Components of edges for subdivided Taylor Integration cells for each node (tetrahedral geometry) .....	109
Table 6.10. Direction of tetrahedral cell and used bounds of integral in natural coordinates .....	111

## LIST OF FIGURES

	Page
Figure 3.1. Binomial expansion for 3D polynomial terms .....	30
Figure 3.2. Different types of local support domains for point of interests (Liu and Gu, 2005).....	30
Figure 4.1. <code>mc</code> class hierarch.....	38
Figure 4.2. Matlab codes of <code>mc</code> , its inherited classes and member functions.....	38
Figure 4.3. <code>inte</code> class hierarch.....	39
Figure 4.4. Matlab codes of <code>inte</code> , its inherited classes and member functions.....	40
Figure 4.5. <code>sdo</code> class hierarch .....	41
Figure 4.6. Matlab codes of <code>sdo</code> class and its inherited classes and member functions .....	42
Figure 4.7. <code>sf</code> class hierarch.....	47
Figure 4.8. Matlab codes of <code>sf</code> and its inherited classes and member functions.....	48
Figure 4.8. Matlab codes of <code>sf</code> and its inherited classes and member functions (continue) .....	49
Figure 4.9. <code>bc</code> class hierarch.....	50
Figure 4.10. Matlab codes of <code>bc</code> , its inherited classes and member functions.....	51
Figure 4.11. <code>solv</code> class hierarch.....	52
Figure 4.12. Matlab codes of <code>solv</code> , its inherited classes and member functions.....	53
Figure 4.13. <code>post</code> class hierarch.....	54
Figure 4.14. Matlab codes of <code>post</code> , its inherited classes and member functions.....	55
Figure 4.15. Engineering (a), meshfree (b) and FEM (c) models of axial loaded and cantilever beam problems and their boundary conditions.....	55
Figure 4.16. Overall flow and communications of classes during the solution for local quadrature domain .....	57
Figure 4.17. The displacement results of analytic, ANSYS, original meshfree (Liu and Gu, 2005) and OOP meshfree programs for cantilever beam problem (upper) and for axial loaded beam problem (lower).....	60
Figure 4.18. The stress results of analytic, ANSYS, original meshfree (Liu and Gu, 2005) and OOP meshfree programs for cantilever beam problem (upper), and for axial loaded bar problem (lower).....	61
Figure 6.1. The models are used in analytical solution for; a) axial loaded beam (case I), cantilever beam (case II), b) simply supported beam problem with distributed loaded (case III) .....	72

Figure 6.2. The models are used in FEM solutions; a) 10 elements model, b) 20 elements model.....	73
Figure 6.3. The models are used in NI-RPIM solutions for 2D shaped Taylor integration cells; a) 18 nodes model, b) 33 nodes model .....	73
Figure 6.4. Comparison of different solution techniques on vertical deformation of 18 nodes 2D axial loaded beam .....	74
Figure 6.5. Comparison of different solution techniques on vertical deformation of 33 node 2D axial loaded beam.....	74
Figure 6.6. Effect of Taylor series expansion terms on vertical deformation of 18 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells .....	75
Figure 6.7. Effect of Taylor series expansion terms on vertical deformation of 33 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells .....	75
Figure 6.8. Effect of Taylor series expansion terms on axial stress of 18 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells.....	75
Figure 6.9. Effect of Taylor series expansion terms on axial stress of 33 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells.....	76
Figure 6.10. Comparison of different solution techniques on vertical deformation of 18 nodes 2D cantilever beam .....	77
Figure 6.11. Comparison of different solution techniques on vertical deformation of 33 nodes 2D cantilever beam .....	78
Figure 6.12. Effect of Taylor series expansion terms on vertical deformation of 18 nodes cantilever beam .....	78
Figure 6.13. Effect of Taylor series expansion terms on vertical deformation of 33 nodes cantilever beam with 2D rectangular shaped Taylor integration cells .....	78
Figure 6.14. Effect of Taylor series expansion terms on axial stress of 18 nodes cantilever beam with 2D rectangular shaped Taylor integration cells ....	79
Figure 6.15. Effect of Taylor series expansion terms on axial stress of 33 node cantilever beam with 2D rectangular shaped Taylor integration cells ....	79
Figure 6.16. Comparison of different solution techniques on vertical deformation of 18 nodes 2D simply supported beam .....	80
Figure 6.17. Comparison of different solution techniques on vertical deformation of 33 nodes 2D simply supported beam .....	80
Figure 6.18. Effect of Taylor series expansion terms on vertical deformation of 18 nodes simply supported beam with 2D rectangular shaped Taylor integration cells .....	80
Figure 6.19. Effect of Taylor series expansion terms on vertical deformation of 33 nodes simply supported beam with 2D rectangular shaped Taylor integration cells .....	81

Figure 6.20. Effect of Taylor series expansion terms on axial stress of 18 nodes simply supported beam with 2D rectangular shaped Taylor integration cells .....	81
Figure 6.21. Effect of Taylor series expansion terms on axial stress of 33 nodes simply supported beam.....	82
Figure 6.22. Sub-division of triangular background cell for each node.....	83
Figure 6.23. Transformation of a triangular integration cell from global (a) to natural (b) coordinates.....	84
Figure 6.24. The used models in the analyses and boundary conditions, a) bar under torsion (case I) b) simply supported beam problem with distributed loaded (case II), c) L-shaped beam under uniformly varying load (case III) .....	87
Figure 6.25. 3D FEM models of analysed cases; a) beam for case II, b) L shaped beam for case III.....	87
Figure 6.26. 3D meshfree models of analysed cases; a) beam for case II, b) L shaped beam for case.....	88
Figure 6.27. Comparison of twist angle results of a non-circular bar under torsion (case I) for analytical, FEM and nodal int. techniques .....	89
Figure 6.28. Comparison of twist angle results of a non-circular bar under torsion (case I) for analytical, Gauss and nodal int. (2 <sup>nd</sup> order terms) techniques.....	89
Figure 6.29. Comparison of effect of on twist angle results of a 3D non-circular bar under torsion (case I) for 4×4 gauss int.....	90
Figure 6.30. Comparison of effect of $\alpha_c$ on twist angle results of a non-circular bar under torsion (case I) for nodal int. (2 <sup>nd</sup> order terms) .....	91
Figure 6.31. Comparison of effect of sd on twist angle results of a 3D non-circular bar under torsion (case I) for 4×4 Gauss int. ....	91
Figure 6.32. Comparison of effect of sd on twist angle results of a 3D non-circular bar under torsion (case I) for nodal int. (2 <sup>nd</sup> order terms).....	92
Figure 6.33. Comparison of effect of on twist angle results of a 3D non-circular bar under torsion (case I) for 4×4 Gauss int.....	92
Figure 6.34. Comparison of effect of on twist angle results of a 3D non-circular bar under torsion (case I) for nodal int. (2 <sup>nd</sup> order terms) .....	92
Figure 6.35. Comparison of vertical deformation results of distributed loaded simply supported beam (case II) for analytical, FEM and nodal int. techniques.....	94
Figure 6.36. Comparison of vertical deformation results of distributed loaded simply supported beam (case II) for analytical, Gauss and nodal int. (2 <sup>nd</sup> order terms) techniques .....	94
Figure 6.37. Comparison of bending stress results of distributed loaded simply supported beam (case II) for analytical, FEM and nodal int. techniques.....	95

Figure 6.38. Comparison of bending stress results of distributed loaded simply supported beam (case II) for analytical, Gauss and nodal int. (2 <sup>nd</sup> order terms) techniques .....	95
Figure 6.39. Comparison of effect of $\alpha_c$ on vertical deformation results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.....	96
Figure 6.40. Comparison of effect of $\alpha_c$ on vertical deformation results of distributed loaded simply supported beam (case II) for nodal int. (2 <sup>nd</sup> order terms) .....	96
Figure 6.41. Comparison of effect of sd on vertical deformation results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.....	96
Figure 6.42. Comparison of effect of sd on vertical deformation results of distributed loaded simply supported beam (case II) for nodal int. (2 <sup>nd</sup> order terms) .....	97
Figure 6.43. Comparison of effect of $q$ on vertical deformation results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.....	97
Figure 6.44. Comparison of effect of $q$ on vertical deformation results of distributed loaded simply supported beam (case II) for nodal int. (2 <sup>nd</sup> order terms) .....	97
Figure 6.45. Comparison of effect of $\alpha_c$ on bending stress results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.....	98
Figure 6.46. Comparison of effect of $\alpha_c$ on bending stress results of distributed loaded simply supported beam (case II) for nodal int. (2 <sup>nd</sup> order terms).....	98
Figure 6.47. Comparison of effect of sd on bending stress results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.....	99
Figure 6.48. Comparison of effect of sd on bending stress results of distributed loaded simply supported beam (case II) for nodal int. (2 <sup>nd</sup> order terms).....	99
Figure 6.49. Comparison of effect of $q$ on bending stress results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.....	99
Figure 6.50. Comparison of effect of $q$ on bending stress results of distributed loaded simply supported beam (case II) for nodal int. (2 <sup>nd</sup> order terms).....	100
Figure 6.51. Comparison of vertical deformation results of L-shaped beam (case III) for FEM and nodal int. techniques.....	100
Figure 6.52. Comparison of vertical deformation results of L-shaped beam (case III) for FEM, Gauss and nodal int. (2 <sup>nd</sup> order terms) techniques .....	101



Figure 6.53. Comparison of bending stress results of L-shaped beam (case III) for FEM and nodal int. techniques .....	101
Figure 6.54. Comparison of bending stress results of L-shaped beam (case III) for FEM, Gauss and nodal int. (2 <sup>nd</sup> order terms) techniques.....	102
Figure 6.55. Comparison of effect of $\alpha_c$ on vertical deformation results of L-shaped beam (case III) for 4×4 Gauss int.....	102
Figure 6.56. Comparison of effect of $\alpha_c$ on vertical deformation results of L-shaped beam (case III) for nodal int. (2 <sup>nd</sup> order terms).....	102
Figure 6.57. Comparison of effect of sd on vertical deformation results of L-shaped beam (case III) for 4×4 Gauss int.....	103
Figure 6.58. Comparison of effect of sd on vertical deformation results of L-shaped beam (case III) for nodal int. (2 <sup>nd</sup> order terms).....	103
Figure 6.59. Comparison of effect of $q$ on vertical deformation results of L-shaped beam (case III) for 4×4 Gauss int.....	104
Figure 6.60. Comparison of effect of $q$ on vertical deformation results of L-shaped beam (case III) for nodal int. (2 <sup>nd</sup> order terms).....	104
Figure 6.61. Comparison of effect of $\alpha_c$ on bending stress results of L-shaped beam (case III) for 4×4 Gauss int.....	104
Figure 6.62. Comparison of effect of $\alpha_c$ on bending stress results of L-shaped beam (case III) for nodal int. (2 <sup>nd</sup> order terms) .....	105
Figure 6.63. Comparison of effect of sd on bending stress results of L-shaped beam (case III) for 4×4 Gauss int.....	105
Figure 6.64. Comparison of effect of sd on bending stress results of L-shaped beam (case III) for nodal int. (2 <sup>nd</sup> order terms) .....	105
Figure 6.65. Comparison of effect of $q$ on bending stress results of L-shaped beam (case III) for 4×4 Gauss int.....	106
Figure 6.66. Comparison of effect of $q$ on bending stress results of L-shaped beam (case III) for nodal int. (2 <sup>nd</sup> order terms) .....	106
Figure 6.67. Sub-division of tetrahedral background cell for each node .....	108
Figure 6.68. Transformation of a tetrahedral integration cell (ic II in Fig.6.67) from global (a) to natural (b) coordinates .....	110
Figure 6.69. The models used in the analyses and boundary conditions, a) an axial loaded bar (case I) b) cantilever beam (case II), c) torsion of cylindrical shaft (case III) .....	112
Figure 6.70. FEM models for; a) case I and II, b) case III.....	113
Figure 6.71. Meshfree models of analysed cases for; a) case I and II, b) case III...	113
Figure 6.72. Comparison of axial deformation of bar (case I) for analytical, FEM and nodal int. techniques .....	114

Figure 6.73. Comparison of axial stress of bar (case I) for analytical, FEM and nodal int. techniques.....	114
Figure 6.74. Comparison of effect of sd on deformation results of axial loaded bar (case I) for nodal int. (4 <sup>th</sup> order terms) .....	115
Figure 6.75. Comparison of effect of sd on stress results of axial loaded bar (case I) for nodal int. (4 <sup>th</sup> order terms) .....	115
Figure 6.76. Comparison of effect of $\alpha_c$ (alfc) on deformation results of axial loaded bar (case I) for nodal int. (4 <sup>th</sup> order terms) .....	116
Figure 6.77. Comparison of effect of $\alpha_c$ (alfc) on stress results of axial loaded bar (case I) for nodal int. (4 <sup>th</sup> order terms) .....	116
Figure 6.78. Comparison of effect of $q$ on deformation results of axial loaded bar (case I) for nodal int. (4 <sup>th</sup> order terms) .....	117
Figure 6.79. Comparison of effect of $q$ on stress results of axial loaded bar (case I) for nodal int. (4 <sup>th</sup> order terms) .....	117
Figure 6.80. Comparison of deflection of cantilever beam (case II) for analytical, FEM and nodal int. techniques .....	117
Figure 6.81. Comparison of bending stress results of cantilever beam (case II) for analytical, FEM and nodal int. techniques .....	118
Figure 6.82. Comparison of effect of sd on deflection of cantilever beam (case II) for nodal int. (4 <sup>th</sup> order terms) .....	119
Figure 6.83. Comparison of effect of sd on bending stress results of cantilever beam (case II) for nodal int. (4 <sup>th</sup> order terms) .....	119
Figure 6.84. Comparison of effect of $\alpha_c$ (alfc) on deflection results of cantilever beam (case II) for nodal int. (4 <sup>th</sup> order terms) .....	119
Figure 6.85. Comparison of effect of $\alpha_c$ (alfc) on bending stress results of cantilever beam (case II) for nodal int. (4 <sup>th</sup> order terms) .....	120
Figure 6.86. Comparison of effect of $q$ deflection results of cantilever beam (case II) for nodal int. (4 <sup>th</sup> order terms).....	120
Figure 6.87. Comparison of effect of $q$ on bending stress results of cantilever beam (case II) for nodal int. (4 <sup>th</sup> order terms) .....	121
Figure 6.88. Comparison of twist angle results of circular shaft under torsion (case III) for analytical, FEM and nodal int. techniques .....	121
Figure 6.89. Comparison of effect of sd on twist angle results of circular shaft under torsion (case III) for nodal int. (4 <sup>th</sup> order terms) .....	122
Figure 6.90. Comparison of effect of $\alpha_c$ (alfc) on twist angle results of circular shaft under torsion (case III) for nodal int. (4 <sup>th</sup> order terms) .....	122
Figure 6.91. Comparison of effect of $q$ on twist angle results of circular shaft under torsion (case III) for nodal int. (4 <sup>th</sup> order terms) .....	122

Figure 6.92. Effects of order of Taylor series terms on CPU time against number of nodes with standard Gauss elimination solver.....	123
Figure 6.93. Effects of order of Taylor series terms on CPU time against number of nodes with Intel Math Kernel Library (MKL).....	124
Figure 6.94. Comparison of CPU time of NI-RRIM (0 <sup>th</sup> order term) with Intel Math Kernel Library (MKL) and ANSYS solvers.....	124
Figure 7.1. The used axial loaded bar model geometry and meshfree model for 2D case analysis .....	128
Figure 7.2. The used cantilever beam model geometry and meshfree model for 3D cases.....	128
Figure 7.3. Axial deformation results of the bar (2D case) for NI-RPIM with 0 <sup>th</sup> term and analytical solution .....	129
Figure 7.4. Axial stress results of the bar (2D case) for NI-RPIM with 0 <sup>th</sup> term and analytical solution.....	130
Figure 7.5. Comparison of deflection results of cantilever beam for analytical and pure nodal int. technique .....	130
Figure 7.6. Comparison of bending stress results of cantilever beam (3D case) for analytical and pure/unstabilized nodal int. technique.....	131
Figure 7.7. Comparison of bending stress results of cantilever beam (3D case) for analytical and nodal int. technique with ASD .....	131
Figure 7.8. Comparison of bending stress results of cantilever beam (3D case) for analytical and nodal int. technique of 0 <sup>th</sup> order term with ASD and LSS .....	131
Figure 7.9. Comparison of bending stress results of cantilever beam (3D case) for analytical and nodal int. technique of 0 <sup>th</sup> order term with ASD/ASD+BC Stiffness E.....	132
Figure 7.10. Comparison of effect of sd on bending stress results of cantilever beam (3D case) for nodal int. with ASD (0 <sup>th</sup> order term) .....	132
Figure 7.11. Comparison of effect of alfc ( $\alpha_c$ ) on bending stress results of cantilever beam (3D case) for nodal int. with ASD (0 <sup>th</sup> order term).....	133
Figure 7.12. Comparison of effect of q on bending stress results of cantilever beam (3D case) for nodal int. with ASD (0 <sup>th</sup> order term) .....	133

## LIST OF SYMBOLS/ABBREVIATIONS

$u(x)$	=	a scalar field function
$P(x)$	=	polynomial basis functions
$R(x)$	=	radial basis functions
$R_q$	=	moment matrix of radial basis functions
$a$	=	coefficients of radial basis functions
$b$	=	coefficients of polynomial basis functions
$d_s$	=	local support domain size
$d_c$	=	average nodal spacing near the point of interest
$\alpha_s$	=	dimensionless size of the local support domain
$U_s$	=	nodal displacement vector of local domain
$\alpha_c, q$	=	shape parameters of radial basis functions
$a(x)$	=	coefficient vector
$\Omega$	=	problem domain
$m$	=	number of the basis functions
$n$	=	number of nodes in the local support domain
$A(x)$	=	weighted moment matrix
$\hat{W}_i$	=	weight function at related node
$\hat{V}_i$	=	derivatives of weight functions
$\Phi(x)$	=	vector of shape functions
$K_i$	=	stiffness matrix
$f_i$	=	force matrix

$\Gamma_{qi}$  = not intersection locations of local quadrature domain with global boundary

$\Gamma_{qr}$  = intersection locations with natural boundary

$\Gamma_{qu}$  = intersection locations of essential boundary locations

$J_{qi}, J_{qu}$  = Jacobean matrices at curve boundary locations

$L^T$  = differential operator

$\sigma$  = stress vector

$u$  = displacement vector

$b$  = body force vector

$\bar{t}$  = traction on the natural boundaries

$\bar{u}$  = prescribed displacement on the essential boundary

$R_n$  = total error between real and its Taylor expansion results

$\gamma$  = stabilization parameter

$I$  = moment of inertia

$L$  = length of the beam

$c$  = distance between upper surface and natural axis of beam

$A$  = cross-section area of the bar

$E$  = Young's modulus

$\nu$  = Poisson's ratio

FEM = finite element method

BEM = boundary element method

FVM = finite volume method

FDM = finite difference method

SPH = smoothed particle hydrodynamics

DEM	diffuse element method
MLS	moving least square
EFG	element-free Galerkin
RKPM	reproducing kernel particle methods
MLPG	meshless local Petrov-Galerkin
PIM	point interpolation method
RPIM	radial point interpolation method
LC-PIM	linearly conforming point interpolation method
NI-RPIM	nodal integration radial point interpolation method
NS-PIM	node-based smoothed point interpolation method
ES-PIM	edge-based smoothed point interpolation method
CS-RPIM	cell-based smoothed radial point interpolation method
LPIM	local point interpolation method
FPM	finite point method
MQ	multi-quadric
ALE	arbitrary Lagrangian Eulerian
NEM	natural element method
LRPIM	local radial point interpolation method
FGM	functionally graded materials
MPC	meshless point collocation
MPM	material point methods
RBFs	radial basis functions
BKM	boundary knot method

LSS	least-square stabilization
TEBS	Taylor series expansion based stabilization
FIG	finite increment gradient stabilization
OOP	object-oriented programming
PL	programming language
OOBM	object-oriented blackboard model
PDEs	partial differential equations

## CHAPTER 1

### INTRODUCTION

#### 1.1 General Introduction

Computational mechanics is one of the numerical analysing techniques and it is becoming widespread by means of developed computational science and numerical methods. On the contrary, usage of experimental methods for the investigation of solid mechanics problems, numerical methods and solution techniques can be used, which are usually fast and they do not require any experimental equipment and materials. Construction of the numerical model and application of required initial and boundary conditions are enough for the solution of the problems. Solution results are taken fast and new conditions are easily adapted into the analyses. Different numerical methods are adapted on computer programs. Some examples of numerical analysis methods can be given as; finite element method (FEM), boundary element method (BEM), finite difference method (FDM), finite volume method (FVM) and so on. They are easily applied on the most of engineering problems. However some assumptions and simplifications are required, especially for the analysis model, which includes complex shapes, nonlinear behaviour and time dependent responses. Some differences occur between simplified and real model, which must be taken under control.

FEM are widely used on the solutions of solid mechanics problems. The analysed model is divided into small elements, which are called finite elements and the solution is going over on these elements. The elements form meshes and it is known that analyse results are directly influenced by quality of these meshes. Rough, less or insufficient meshes cause to give less sensitive or incorrect results. Also it may cause to not convergence of the finite element analyses (FEA) with respect to analyse conditions. Hence, mesh convergence is also studied for satisfying stable results, which consumes more time again.



Different studies are investigated for eliminating meshing operation of FEM. One of the investigations is meshfree method. Meshfree method is one of numerical analysing techniques, which gains popularity with respect to new improvements. In meshfree or meshless methods, a mesh is not commonly used, at least interpolation operations. Only placing nodes into the analysed system is enough. However, meshfree method is a new method when its development stage is compared with FEM and BEM. It needs more improvements, especially for the construction of relations between nodes before and during the analyses. Most of the researchers focus on this subject and they develop some new techniques.

Besides the developments on numerical techniques, it is also known that their programming in computer programs is important. A well-structured programming is needed for using numerical methods efficiently in computational solutions. Misidentifications, incorrect operation cycles, unwanted changes of constant parameters and conflicts lead to undesirable results during the application. Hence another programming technique must be used for prevention of conflicts, especially for the programs in developing stage. Object oriented programming (OOP) technique is one of the programming technique and used especially in the programming of complex and developing programs. Transactions with objects avoid the complexity and classification of similar structures increases efficiency in OOP. Inheritance, polymorphism and encapsulation properties of OOP provide to decrease conflicts in programs, especially complex types.

## **1.2 Research Objectives and Tasks**

The main objective of this study is constructing an object oriented meshfree program for analysing of solid mechanics problems. The followed stages of this study can be summarized as;

- I. to investigate the studies about meshfree methods and OOP in numerical methods in literature
- II. to determine a suitable meshfree program in literature to apply in OOP programming
- III. to construct classes of OOP programming of meshfree methods for solution of 2D elasto-static problems

- IV. to set classes of RPIM and MLS methods for construction of shape functions
- V. to add randomly distributed nodes property in the determination of node coordinates
- VI. to determine class properties and objects
- VII. to set a main program for running the classes in harmony
- VIII. to solve 2D elastostatic plane problems subjected to axial and bending loading with OOP meshfree program
- IX. to improve 2D meshfree program for using in 3D applications
- X. to solve 3D elasto-static solid mechanics problems
- XI. to improve 2D and 3D Matlab programs for new meshfree techniques

### **1.3 Layout of Thesis**

A general introduction about the thesis is given in chapter one. The previous studies are mentioned in literature review about meshfree methods, their applications and OOP in numerical methods, which are given in chapter two. The formulations of MLS and RPIM on the construction of shape functions are summarized in chapter three, which are used in many meshfree methods. In chapter four, application of OOP in MLPG and LRPIM is given for 2D elastostatic problems. Classes of OOP are defined in detail. In chapter five, nodal integration scheme is summarized for 2D and 3D structures. Nodal integration needs integration cells. In chapter six, application of nodal integration on 2D/3D RPIM for different integration cells is given. Stress results in nodal integration include fluctuations. In chapter seven, stress stabilization methods are given for decreasing fluctuations in nodal integration stress results. In the last chapter, chapter eight includes main findings in this study.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

Previous studies are searched in literature about meshfree methods and object oriented programming. Also, their applications on different subjects are searched and summarized in this section.

#### **2.2 Meshfree Methods**

Nowadays, meshfree methods are very popular and used in many different areas such as solution of 2D/3D solid and fluid mechanics, thermal and heat transfer problems, explosion modelling, etc. The main reason is flexibility of meshfree methods. Solutions are generally carried out without using any predefined elements and only nodes are used in modelling.

The idea of meshfree methods in numerical analyses has been used nearly since 1977 by the development of SPH (smoothed particle hydrodynamics) (Lucy, 1977; Gingold and Monaghan, 1977), which is applied on the solutions of astrophysical problems. Further developments are continuously carried on and new methods are developed by different researchers. DEM (diffuse element method) (Nayroles et al., 1992) is developed at the further development stages which requires no mesh generation. It includes MLS (moving least square) functions, which is used further development stage of EFG (element-free Galerkin) (Belytschko et al., 1994) method. The usage of MLS functions as trial and test functions in EFG method prevents volumetric locking on the solutions. Liu et al. (1995) develop Reproducing Kernel Particle methods (RKPM). Also partitions of unity (Babuska and Melenk, 1996) and H-p clouds (Duarte and Oden, 1996) methods are developed by using MLS functions. Atluri and Zhu (1998) study on meshless local Petrov-Galerkin (MLPG) method for solution of solid mechanics problems. PIM (point interpolation method) (Liu and Gu, 2001) and RPIM (radial point interpolation method) (Wang and Liu, 2002a) are developed for solution of solid mechanics problems.

These methods can be also combined with other methods. PIM and PRIM is included inside of BEM (Liu and Gu, 2004) for construction of boundary integral equations. PIM and RPIM are further used with different methods and integration schemes; linearly conforming point interpolation method (LC-PIM) (Liu et al., 2005), nodal integration radial point interpolation method (NI-RPIM) (Liu et al., 2007), NS-PIM (node-based smoothed point interpolation method) (Zhang et al., 2007), edge-based smoothed point interpolation method (ES-PIM) (Liu and Zhang, 2008) and CS-RPIM (cell-based smoothed radial point interpolation method) (Liu and Zhang, 2009) are available in literature. The name of LC-PIM (Liu et al., 2005) was lately changed to NS-PIM (Zhang et al., 2007) to make it consistent with other types of smoothing methods. Most of these methods are constructed based on weak form and provide to eliminate pre-defined mesh dependency for construction of shape functions. However, more researches are needed for fully eliminating mesh generation on interpolations and integrations. Investigations on meshfree methods still continue. Therefore, it is not fully developed with respect to FEM and BEM.

Researchers are trying to solve the problem about meshing cost of FEM or BEM with application of meshfree methods on different subjects. Some of the application areas are summarized;

### **2.2.1 Solid Mechanics**

Numerical methods have been used on the investigations of solid mechanics problems for a long time. Different methods are developed and they are used on the solution of 2D/3D solid mechanics problems. Bobaru and Mukherjee (2001) investigate element free Galerkin method on 2D shape optimization problems and compare their results with FEM results. They mention that MLS in EFG method has higher computational cost than FEM, but no remeshing property of meshfree methods close the difference in their analyses. Besides the usage of meshless local Petrov-Galerkin method, local point interpolation method (LPIM) is developed (Gu and Liu, 2001) which has lower computational cost than MLS method. Onate et al. (2001) use finite point method (FPM) for the investigation of linear elastic solid mechanics problems. They achieve better results than a standard FEM analysis. Wang and Liu (2002b) investigate the effect of different shape parameters of RPIM on a cantilever beam problem by using different basis functions. They determine that

optimum shape parameters are  $q=1.03$  and  $R=1.42$ , when MQ (multi-quadric) basis functions are used. If Gaussian basis functions are used, the optimum shape parameter has a range between 0.003 and 0.03 (Wang and Liu, 2002b). Liu (2002) develops a method, point assembly method for investigations of two-dimensional solids. The method (Liu, 2002) does not need cells and search neighbour points, but it cannot be used in compressible elastic material problems with respect to volumetric locking condition. Also Lui (2002) warns that overlapping of integrations of nodes can be seen in this method, which needs high point density for achieving accurate stress results.

Noguchia and Kawashima (2004) use ALE-EFG (arbitrary Lagrangian Eulerian-EFG) method for investigation of cable-reinforced membrane structures. Although they get acceptable results, they mention that the used method is nearly five or ten times slower than a FEM analysis. Liu et al. (2005) investigate solutions of 2D solid mechanics problems by using LC-PIM. They achieve more accurate results than FEM analysis. Dai et al. (2006) use RPIM for investigation of 2D solids with both regular and irregular node distributions. They achieve good agreements between compared results and both regular and irregular distributed node results in their analyses. Computational cost of meshfree methods is main problem on solid mechanics problems. It is tried to be decreased with the both usage of Heaviside test function and MLPG method in the study of Hu et al. (2006). Heaviside test function is also used by Zheng and Dai (2011) for providing less computational costs. They use local Moving Kriging method in their analyses. Yanan et al. (2011) investigate 2D elastic problems with b-spline method. They use different scales of subdomains for saving computational costs. Fooladi et al. (2011) investigate time dependent dynamic loads on solid mechanics with using local radial interpolation method. Besides the usage of Gaussian quadrature, they use Taylor series expressions for decreasing the number of field nodes and integration points. In some cases, irregular node distributions cause to increase numerical errors in meshfree analyses and the problem can be prevented by increasing the number of nodes (Abdollahifar et al., 2012). Plasticity analyses of solids are also investigated by Chen et al. (2011). They use kinematic hardening assumption and reduced basis technique to achieve more realistic and stable solutions.

### **2.2.2 Large Deformation**

The most weakness of FEM is observed in large deformation analyses, because of highly distorted elements occur. Chen et al. (1996) use reproducing kernel particle methods for ignoring mesh distortion problems of mesh based techniques. They investigate large deformation problems of non-linear elastic and inelastic structures. Shapiro and Tsukanov (1999) use meshless methods for investigation of deforming domains with r-functions method. They do not use predetermined elements, but it causes a significant computational cost in their analyses. Liew et al. (2002) use reproducing kernel particle method for the investigation of large deformations of elasto-static beam problems. They add penalty method for eliminating the lack of Kronecker delta property of RKPM. Shangwu et al. (2003) investigate strain rolling of plane with RKPM method. They mention that rectangular influence domain is more suitable than circular influence domain. Gu et al. (2007) use local Kriging method on the solutions of 2D deformable bodies. Filice et al. (2009) use NEM (natural element method) on the extrusion processes, but they mention that NEM has more computational cost than FEM. Hanoglu et al. (2011) investigate rolling process with a strong form meshfree technique and they gain good agreements with compared DEFORM-2D results.

### **2.2.3 Crack and Fracture**

Crack and fracture analyses are mainly discussed and studied for preventing the failure of mechanical systems. High distorted elements may occur, if mesh based methods are used in fracture analyses. Ponthot and Belytschko (1998) use arbitrary Lagrangian-Eulerian formulation for the investigation of dynamic crack propagation. They clarify that redefining of node locations are easily handled by ALE. Hao et al. (2000) use RKPM for investigation of fracture problems. They gain higher resolutions on shear band pattern in RKPM solution than FEM solution. Rao and Rahman (2001) use both meshless and FEM on crack analyses and use same shape functions on the interaction of meshless and FEM models. Li et al. (2002) use meshfree Galerkin method on the investigation of dynamic crack propagations. They achieve to see curved shear band path in meshfree analyses as nearly same as compared experimental results, which are not captured with FEM. Rabczuk and Belytschko (2007) study on crack grow by using meshfree methods. They model the

crack as a collection of particles and the unity of these particles are ruined on the direction of crack propagation. Gu and Zhang (2008) use both of meshfree and FEM for crack analyses. They set meshfree particles on crack sides for prevention of remeshing, but they use FEM at far away of crack for less decreasing computational efficiency. In the study of Rabczuk et al. (2010), particles are set to be ruined easily by using meshless method when cracking criteria occurs. Wena and Aliabadi (2011) investigate stress intensity factors by using element free Galerkin method with both usage of RBF and MLS methods. Zhang et al. (2011) investigate 3D crack analysis. They use radial integration method for transferring the domain integral to boundary integrals over the crack. Liu et al. (2011) investigate crack problems with cell based smoothed radial point interpolation method. They set the number of gauss points as 4, for gaining optimum solution accuracy in their analyses. Pathak et al. (2014) investigate 3D crack problems with a coupled FE and EFG methods. They use EFG method on crack surfaces and FEM on other analysed locations.

#### **2.2.4 Thin and Thick Plates**

Another application area of meshfree methods is investigation of thin and thick plate problems. Donning and Liu (1998) investigate thin and thick plates with meshless methods. They try to eliminate shear locking of thin members with Galerkin method. Static and free vibration analyses of thin shells are investigated with MLS shape functions by Liu et al. (2002). They assign these shape functions into Galerkin weak form for construction of system equations. Liew et al. (2004) investigate thin plates by using MLS functions and they recommend that more order degree of polynomials are needed in irregular distributed node analyses than regular distributed node analyses. Li et al. (2005) study on a locking-free meshless local Petrov–Galerkin formulation on shear flexible thick plates in thin plate limits. They prevent shear locking with changing two dependent variables in governing equations rather than using assumed strain and reduced integration techniques. Bending behaviour of shear deformable shallow shells is investigated in the study of Sladek et al. (2007) by using meshless local Petrov-Galerkin method. They see that regular distributed nodes give more stable results than irregular ones. Xia et al. (2009) use local radial point interpolation method (LRPIM) on the elasto-dynamic solution of moderate thickness plates. They use low order polynomials for gaining time, however their results have low accuracy. Dynamic fracture of plates is (Gato, 2010) also investigated and the

connectivity of fractured parts is easily removed in calculations by the usage of meshfree methods. RezaeiMojdehi et al. (2011) investigate static and dynamic thick plates by using cubic domains in support domain and MLPG method with MLS shape functions. They eliminate domain integrals in the calculation of stiffness matrix by using Heaviside test functions. Heaviside test functions are used for converting the local integral equations to local sub-domains (RezaeiMojdehi et al., 2011). Elasto-plastic behaviour of thick plates is investigated by Xi et al. (2011). They use polynomial radial basis functions inside of weighted residual method and use quartic spline functions as test functions. Liu et al. (2011) use adaptive analysing method for increasing sensitivity in sheet metal forming with reproducing kernel particle method. Wu et al. (2011) investigate plane stress and strain problems of plates with meshless collocation method. Dinis et al. (2011) investigate the dynamic behaviour of 3D thin walled structures with natural neighbour meshless method. The determination of neighbour nodes provides less influence effects on the variations of boundary conditions. Wang et al. (2011) try to eliminate surface mesh on BEM by using hybrid boundary node method and MLS functions. Results of their study show good agreements with compared analytical solutions on the solutions of 3D thick walled cylinder problems. Similar usage of meshfree methods can be seen for the solutions of nonhomogeneous problems by Chen et al. (2011).

### **2.2.5 Composites**

Investigations of composite plates with different combinations, material properties and other parameters cause to consume large amount of resources and lose lots of time in experimental analyses. In this sense, meshfree methods are also applied on the investigation of composite structures. Bending and buckling of thick rectangular laminates are investigated by using MLS methods (Liew and Huang, 2003). Liew and Huang (2003) observe that if enough support is achieved for integration, the accuracy of the solution is not affected dominantly by node irregularities and support sizes. EFGM is used on the investigations of plates and laminates (Belinha et al., 2007), but this method has bigger computational cost than other methods. Liu et al. (2008) use RPIM for analysing composite plates, which are subjected to static and free vibration loads. Besides the usage of Gauss integration, Liu et al. (2008) use nodal integration which prevents shear locking condition on the analyses. Galerkin meshfree formulation is used for the investigation of composites by Wang et al.



(2009). They set different integration subdomains for different materials. Ferreira et al. (2011) use RBF collocation method in the investigations of composites. Different researchers investigate FGM (functionally graded materials) and composites by using meshless methods (Wu and Yang, 2011; Wu et al., 2011; Wu and Chiu, 2011) and suitable results are achieved. Li et al. (2011) achieve more realistic models on the investigation of 3D orthogonal woven fabric composites by using radial basis and moving kriging interpolation functions. Orthotropic enrichment functions are used on fracture problems of composites by Ghorashi et al. (2011) for increasing the accuracy of the solution. Roque et al. (2011) investigate composite plates by using finite difference methods with radial basis functions and they mention that shape parameters are important and affect the solutions.

### **2.2.6 Contact**

The interaction of bodies may give different responses on different conditions. Meshfree methods can be used in contact mechanic problems with different contact algorithms. Campbell et al. (2000) use SPH and develop a contact algorithm by using penalty method. They see in their investigations that oscillations in SPH may cause to inaccurate results. Moreover, in some cases, the contact algorithm begins without interaction of nodes (Campbell et al., 2000). They determine that it is caused from interaction of node domains. Three different radial basis functions are investigated by Xiao et al. (2005), which are MQ, EXP and TSP on the solutions of 2D contact problems. They see that MQ gives the best and EXP gives the worst results. However, shape parameters of all these basis functions are sensitive to distance of nodes without TSP basis functions (Xiao et al., 2005). Kwon and Youn (2006) investigate a rigid plastic contact analysis. Hu et al. (2007) use two different test functions for two different cases. When nodes do not contact, Heaviside test function is selected for simplifying domain integrals (Hu et al., 2007). Otherwise, if the contact occurs, Kronecker delta test function is used (Hu et al., 2007). Nianfei et al. (2009) investigate an adaptive RKPM method on contact problems. Especially, they use adaptive domain sizes for elasto-plastic large deformations analyses. Hence, they achieve optimum support sizes for different nodes interactions and preventing usage of unnecessary nodes in integration. Amirani and Nemati (2011) investigate a meshless contact algorithm with FEM in the investigations of 2D frictionless interaction problems. They use FEM shape functions in contact.

### **2.2.7 Impact**

Besides static analysis, impact analysis is also examined by different researchers. Geometrical discontinuities, fragmentation of geometry and up to hypervelocity impact effects may cause to difficulties when FEM is used in the analyses. Liu et al. (1999) use reproducing kernel particle method on impact problems. Li et al. (2001) investigate an impact problem on a Taylor bar with a contact-detection algorithm, which is mainly derived from moment method in meshfree discretization. Ma et al. (2002) use material point method on hypervelocity impact problems, which does not need to search neighbour nodes in the analyses and avoids tensile instability. Rabczuk and Eibl (2006) investigate dynamic behaviour of concrete with SPH, which supports fragmentation easily. Recio et al. (2007) investigate hourglass condition and different forms of locking by using EFGM. Wang et al. (2009) try to prevent hourglass condition and large computational time by using a local boundary integration operation with both usage of meshfree and FEM shape functions. Johnson et al. (2011) use a coupled particle and finite element algorithm on the investigations of high velocity impact problems. The used technique prevents tensile instabilities in the particles. Zhang et al. (2011) use both meshfree and FEM for investigation of impact analysis. They use SPH for meshfree method and particle contact algorithm for contact definition. They clarify that contact begins when the SPH domain of node contacts with finite element. However, the SPH node has not directly interacted with finite element yet (Zhang et al., 2011), which may give less accurate results. Gong et al. (2012) investigate impact behaviour of an aluminium plate by using material point method, which provides good agreements with compared experimental solutions.

### **2.2.8 Fluid Mechanics and Heat Transfer**

Fluid mechanics and heat transfer problems are another usage area of meshfree methods. Rabczuk et al. (2004) study on Lagrangian and Eulerian kernels with stable particle methods. They mention that Lagrangian kernels are more suitable for small deformation and failure problems of solids. If fluid-solid interaction problem is investigated, it may be useful to define solid structures with Lagrangian kernels and fluid structures with Eulerian kernels (Rabczuk et al., 2004). Vorobyev et al. (2011) investigate sloshing fluid analyses with SPH. Hashemi and Jahangirian (2011) use Taylor series least square method for interpolations in compressible flow

investigations. They use local time stepping and residual smoothing methods for increasing computational efficiency. Li (2011) develops a generalized boundary node method for investigation of incompressible viscous fluid flows, which provides the application of boundary conditions easily. Darbani et al. (2011) investigate broken dam problems with natural element method and achieve suitable results. Swathi and Eldho (2014) study on ground water flow simulation by using MLPG method, which has much less consuming time for pre-processor step than FDM and FVM.

Xue-Hong and Wen-Quan (2008) use meshless MLPG method for the investigation of steady state heat conduction problems. They achieve accurate results for 2D irregular complex domains. Wu et al. (2009) use node based smoothed point interpolation method (NS-PIM) for 3D heat transfer problems. The used gradient smoothing technique enables to achieve suitable results with low ordered shape functions (Wu et al., 2009). Thakur et al. (2010) investigate nonlinear heat conduction problem on irregular domains with MLPG method. Shibahara and Atluri (2011) investigate welding process by using MLPG method. They see that increasing the number of nodes cause to decrease the effect of shape parameters. Khosravifard et al. (2011) use an improved RPIM method for heat conduction analyses of FGM. Skouras et al. (2011) investigate heat conduction problem on isotropic and functionally graded materials with meshless point collocation (MPC) method. They mention that the accuracy of solution is directly affected by the order of meshless approximation functions and their derivatives. Heat conduction problems are also investigated by Cheng and Liew (2012). They use RKPM and achieve good agreements with compared analytical solutions.

### **2.2.9 Developments on Applications**

Some weaknesses of meshfree methods are tried to be eliminated by different researchers. They use different methods and observe their efficiencies on the solution of different problems. Liu et al. (1997) use spectral methods with moving least square reproducing method and see that this method needs more calculation time and computational resources. Liu and Gu (2002) investigate three different integration methods, which are LPIM, MLPG and LR-PIM. They see that LPIM has the best and LR-PIM has the least efficiency. Liu and Gu (2004) study on boundary meshfree methods and determine optimum number of nodes in the support domain and shape

parameters for MQ radial basis functions. Kwon et al. (2006) investigate the performance of integrations of Galerkin method and determine that increasing integration points in nodal supports provides less integration errors. RPIM is very effective and becomes widespread in the meshfree methods. It is used in DRM (dual reciprocity method) with hybrid boundary node method (HBNM) and seen that using RPIM increases the accuracy (Yan et al., 2010). The study shows that the usage of MLS shape functions increases CPU time, which is decreased by construction of shape functions with RPIM. Tang et al. (2011) include automatic mesh generation technique with Delaunay method on node based smoothed point interpolation method. The applied adaptive refinement method in the study effectively achieves accurate stress concentration results. He et al. (2011) use meshfree weighted least square method for achieving less computational cost on MLS method. The used method avoids numerical integration, which increases computational efficiency. They try to prevent the ill-conditioned formulations of MLS method for 2D Helmholtz problems. Liana et al. (2011) use material point method (MPM) and FEM together with hybrid nodes. FEM is used in small deformation and MPM are used in large deformation sections. Moosavi et al. (2011) use orthogonal weighted basis functions to solve singularity problems on meshless finite volume method. Xiang et al. (2011) investigate 3D partial differential equations with radial point collocation method. They study on different radial basis functions and try to calculate optimum shape parameters. It is mentioned that if the number of partial differential equations increases, it will cause to decrease the accuracy of the solutions on meshless collocation method. Cheng et al. (2014) investigate 2D plasticity problems with EFG method. They include interpolating moving least square method to construct shape functions, which supports property of Kronecker delta function.

### **2.2.10 Radial Basis Functions and Other Applications**

RBFs (radial basis functions) include an important level in some meshfree methods and are widely used in meshfree approximations. They are used for development of BKM (boundary knot method) (Chen and Tanaka, 2002). However, in some meshfree methods, the usage of RBF cannot fully increase accuracy. It is observed in the study of Zhang (2006) that whether radial basis functions are used, the accuracy and efficiency of Galerkin method is less than FEM. The interpolation of nodes can cause oscillations.

Xiao et al. (2007) investigate homogeneous and isotropic thick elastic plates with a meshless local Petrov-Galerkin (MLPG) method and a higher-order shear and normal deformable plate theory. They mention that the usage of RBFs provides essential boundary conditions without application of any special treatment, which reduces computational cost. Demirkaya et al. (2008) observe in their study that node placements are very important and affect the convergence of RBF solutions. Regular node placement provides the convergence of solution quickly in RBFs. It is also detected by Demirkaya et al. (2008) that the matrices may be ill-conditioned, if too much number of nodes is used. Increasing number of nodes causes to increase the size of Jacobian and other matrices in the solution and cause to decrease the computational speed. Kanber and Bozkurt (2008) investigate a diagonal offset algorithm to overcome singular moment matrix problem by using polynomial point interpolation method. Another meshfree method is NNR-PIM (natural neighbour radial point interpolation method), which is used for investigation of 3D solids (Dinis et al., 2007), plates (Dinis et al., 2008) and thin structures (Dinis et al., 2010) by using RBF, which achieves enough accuracy.

RBF meshless method is simple to use, because of RBFs only depend on the Euclidean distance between nodes. It is also used for the solutions of radiative transport equations by Kindelan et al. (2010). Both local and global domains can be used in RBF. But, local domains in the formulation of system matrix have less computational cost on the contrary to global domain (Senturk, 2011). Zhou et al. (2011) try to improve ill-conditioned interpolation matrices in the calculations of RBF by applying different shape parameter variation schemes. Ferreira et al. (2011) investigate elasto-static, vibration and buckling analysis of laminate plates by radial basis function collocation method. Also this method is used for solutions of Burgers' equations by Islam et al. (2012). RBF includes different shape parameters and their effects (Wang and Liu, 2002b; Kanber et al., 2013; Bozkurt et al., 2013) are also searched in some studies. They can affect solution results and suitable parameters must be searched before their usage. This condition is also observed in the study of Wang et al. (2010) that the accuracy of solutions in meshless methods is affected by the number of nodes, selection of nodes and used shape parameters. More developments are continuously carrying on meshfree methods.

### 2.3 Stress Stabilization on Meshfree Methods

Different techniques are developed and adapted for increasing the applicability of meshfree methods. One of the integration schemes; nodal integration scheme can increase applicability and be effective for compact usage inside of RPIM, which is explained in detail in the study of Liu et al. (2007).

The application of nodal integration is fast and suitable for complex geometries. This method uses Taylor series expansion in integration and the solution results are mainly affected with the used order of Taylor expansion. However, it is observed that the stress results of pure/unstabilized nodal integration (Duan and Belytschko, 2008; Yavuz and Kanber, 2015) include high fluctuations. Even if two close nodes are selected in the analysed model, their stress results can be highly different. Hence application of different methods is investigated for trying to decrease the stress fluctuations.

In literature, behaviour of nodal integration includes instabilities in some cases. Initially Biessel and Belytschko (1996) include additional a stabilization term on potential energy function of EFG method for nodal integration. It achieves instability problems of nodal integration in weak form formulations. Chen et al. (2001) also focus on instability of direct nodal integration and observe that integration constraints include errors in direct nodal integration and they cannot be satisfied enough in Gaussian integration at irregular discretization. They eliminate this problem with including a strain smoothing stabilization. Zhou et al. (2003) include square residual of equilibrium equation into potential energy function in stabilized nodal integration scheme. The used Voronoi diagram with supporting Delaunay triangles increases accuracy of volume assignment of nodal integration. Liu et al. (2006) use least square stabilization technique on radial point collocation method (RPCM). The used stabilization technique and easily added nodes in analysed models with respect to adaptive analysis give accurate results. Van et al. (2007) use conforming nodal integration into finite element formulation of laminate plates, which prevents shear locking. Han (2010) uses stabilized conforming nodal integration method on elasto-plastic analysis of metal forming process. Nodal integration with strain smoothing stabilization is used, which prevents instabilities of integration of Galerkin weak form formulations. Elmer et al. (2012) use a stable

nodal integration method on nearly incompressible solids. Xu (2014) uses stabilized nodal integration for cracking particles method (CPM), which supports stability and computational efficiency. Cui et al. (2015) investigate 3D elasticity problems with RPIM. Gradient smoothing method is applied on smoothed Galerkin weak form of analysed equations, which gives accurate results and provides to not necessary of derivatives of shape functions.

Duan and Belytschko (2008) summarize and compare 3 different stabilization techniques; which are least-square stabilization (LSS) (Biessel and Belytschko, 1996; Fries and Belytschko, 2008), Taylor series expansion based stabilization (TEBS) (Liu et al., 1985; Nagashima, 1999; Liu et al., 2007) and the finite increment gradient stabilization (FIG) (Bonet and Kulasegaram, 2000), which can adjust stress instabilities. However, they recommend using LSS and TEBS rather than FIG. These methods can be tried to reduce stress fluctuations in nodal integration.

## **2.4 OOP in Numerical Methods**

Besides the structure of procedural programming, different operators and techniques are available in OOP (object oriented programming) for increasing modularity. Objects and classes form main structure and all processes mainly continue with respect to them. An object is used inside of calculations and related data are stored as its assigned properties. Classes are used for grouping similar objects which support object properties in a main framework for various objects. Functions in the classes can also handle related calculations. These structures in OOP support high programming capacity. Also some extraordinary properties are available in OOP than procedural programming, which are encapsulation, inheritance and polymorphism. Encapsulation hides the implementation details of the objects. Input and output processes are mainly concerned at the outside of the class. Inheritance is methodology to form new classes in terms of old classes. Polymorphism is the ability to use an operator or method in different ways.

Further developments or extension sections of numerical methods can be easily applicable with OOP. This structure is well used in FEM with defining classes and their methods. This programming technique (Mackie, 1992) provides to better code writing with its supplied modularity in FEM. Various design structures are

constructed in OOP of FEM in different subjects. Zimmermann et al. (1992) use OOP to FEA of 2D truss elements by using Smalltalk PL (programming language). They define 5 classes for whole analyses; `component`, `domain`, `tool`, `linear system` and `dof` classes, which are subclasses of a main class. Dubois-Pelerin et al. (1992) develop an OO (object-oriented) FEM program by using Smalltalk PL from extension study of Zimmermann et al. (1992). OOP methodology is used for increasing readability, modularity and reusability. The OO program includes two different kinds of methods; instance and class methods. It is clarified that objects use instance methods, but classes use class methods.

The usage of C++ PL in OOP is widely and most of the OO FEA codes are written with this language. Ohtsubo et al. (1993) develop an OO finite element program with capable of fully automated sub-division of mesh property by using C++ PL. When model is changed, only changing of related object can satisfy the new changes without changing the whole model. They define the analysed problem with three main objects, which are `geometry`, `analytical conditions` and `relation` objects and applied to 3D shell elements. Kong and Chen (1995) study on an OO FEA program with C++ PL. They use different block objects on different block hierarchies. Adeli and Kao (1996) design object-oriented blackboard model (OOBM) for design of large structures with C++ PL. OOP mainly manages, isolates and encapsulates knowledge bases of objects, which are specific objects for OOBM and control the main flow of algorithm. Sheehy and Grosse (1997) also study OOBM for FEM and analysis of multichip modules with C++ PL.

Zimmermann et al. (1998) study OO FEA environment and its key features with both Smalltalk and C++ PL. They construct 5 major classes; `FEMComponent`, `Dof`, `Domain`, `LinerSystems` and `GaussPoint`. Pantale (2002) uses OO FEM to impact simulations with C++ PL. Peters and Dziugys (2002) use OOP for 3D numerical particle motion simulation of granular materials. They use C++ PL and mention that encapsulation and inheritance are very attractive for this research. Ma and Feng (2002) develop OO FEA software with visual C++ PL for providing Windows-type graphical user interfaces (GUI). Yu and Kumar (2001) construct an OO framework to FEM for decreasing duplication between analyse components with C++ PL. They use inheritance and polymorphism to provide the framework as



abstract class in OO FE program. Patzak and Bittnar (2001) study an OO FE program and construct material-element and analysis frame design with C++ PL.

Nonlinear behaviour responses in numerical analyses are programmed with OO C++ PL in some studies. Menetrey and Zimmermann (1994) apply OO FEA to plasticity analyses with C++ PL. Newton-Raphson and stress integration algorithms are included to existing objects. Dubois-Pelerin and Pegon (1998) introduce OOP formation for nonlinear FEA. They use C++ PL and define three classes; `problem`, `nonlinear solver` and `linear solver`. Zarabas and Srikanth (1999) use OO C++ PL for FEA in inelastic large deformation analyses. Lages et al. (1999) study an OO nonlinear FEA with C++ PL. It can solve elasto-plastic and geometrically nonlinear beam model with Newton-Raphson and arc-length nonlinear solution schemes. Commend and Zimmermann (2001) develop OO FE program with including Von Mises plasticity. They use C++ PL in OO coding. Tabatabai (2002) uses OO FEA for steel weight minimizations with respect to plasticity theory with C++ PL. Wegner and Peczak (2010) use OOP method for non-linear FEA with including a strain energy density function. C++ PL is used in coding.

Peskin and Hardin (1996) develop and OO program of FEA for fluid dynamic problems with C++ PL. Their written code can solve continuum equations of fluid dynamics, heat and mass transfer in chemical engineering applications. Munthe and Langtangen (2000) study an application of OO FEA for CFD with C++ PL. Sampath and Zarabas (1999) use OO C++ programming method on FEM for investigation of solidification process with including heat and mass transfer. Qiao (2006) uses OOP technique to program BEM for 2D heat transfer analyses with C++ PL. It is applied for solution of Laplace and diffusion equations. Moreno and Ramaswamy (2003) use OO FEA with C++ PL for investigation of natural convective flows. Different class hierarchies are included inside of geometrical and mathematical management tools. Morgan and Henda (2006) develop an OO program with Java and C++ PL for investigation of thermal radiation simulator. It has graphical user interface (GUI) and is able to work on Macintosh, Windows and Linux platforms. Liu et al. (1996) develop a 2D OO program with both adaptive FEM and FVM by using C++ PL. Their similar components like linear system solvers, data structures and error analyse modules are easily grouped with OOP. Phongthanapanich and Dechaumphai (2006)

develop an OO FEM and FVM program with graphical interface. It can be used in the solutions of heat transfer, solid and fluid mechanic problems. They use both Visual Basic and C++ PL in programming. Marczak (2006) constructs a structure of OO numerical integration scheme for FEM and BEM by using C++ PL. Lage (1998) uses an OOP technique to BEM with C++ PL. Wang et al. (1999) develop OO BEM by using C++ PL.

Gil and Bugeda (2001) include damage models and sensitivity analyses in their OO FE program by using C++ PL. Pantale et al. (2004) apply OOP technique to FEA of metal forming and impact analyses. They use C++ PL. Chung et al. (2005) use OOP technique for investigation of FEA of fibre metal laminates with C++ PL. Kromer et al. (2004) apply OO FE to multibody system analyses with C++ PL. Wang and Kolditz (2007) investigate multi-field problems with OO FEA. C++ PL is used in their study. OOP can easily manage different types of partial differential equations (PDEs). Geometrical, topological and process-related data are encapsulated. The code is applied to thermo-hydro-mechanical coupled problems for geotechnical applications. Dadvand et al. (2010) develop an OO FEA codes for multi-disciplinary problems with C++ PL.

Programming language and efficiently usage of computers are also important cases in computational numerical methods. Dubois-Pelerin and Zimmermann (1993) carry the developed OO FEM structure (Zimmermann et al., 1992; Dubois-Pelerin et al., 1992) from Smalltalk PL to OO C++ programming for increasing efficiency. Because it is mentioned in their study that Smalltalk PL has very slow executing time, approximately 100 times slower than a Fortran code. Cary et al. (1997) investigate the differences between C++ and Fortran 90 PL in OOP platform. C++ satisfies OOP properties, however Fortran 90 is not capable of OOP features. It is thought that it supports a small section with type and module features without inheritance. The lack of high reusable libraries in Fortran decrease modularity in OOP. Archer et al. (1999) develop a new OOP structure for FEA which can solve nonlinear static and dynamic conditions with C++ PL. It is observed in the study that Fortran code is %20 faster than the prepared OO program, which may be caused from young age OO program and lack of a fast numerical library. Kromer et al. (2005) study an OO FE code for multibody systems. The code includes the solution

capacity of 3D multibody flexible mechanism with large rotations and large strains. It is mentioned that efficiency of OO C++ has lower performance than Fortran compilers because of polymorphism, which needs dynamic binding.

Carry et al. (1994) study on an OO FE solver for parallel computing with C++ PL. Guidec et al. (1996) construct an object oriented framework for increasing efficiency of usage of distributed computing systems (DCS). Re-usage of encapsulated objects in parallel computing have benefits in calculations. Master et al. (1997) study transient heat conduction problems with solidification in FE. They use C++ PL with OOP and parallel computing technique. Shared and distributed memory in parallel computing allows increasing computational performance and using for large meshes. They construct two main classes; mathematical and geometrical classes. Mukunda et al. (1998) study parallel computing of OO FE framework with C++ PL. Base and Carey (1999) construct data structure classes on FEM with parallel computing and C++ PL. Chen and Archer (2001) develop an OO FE program with combination of multi-data scheme and message passing diagram. Niekamp and Stein (2002) develop 2D and 3D adaptive OO FE program with a refinement algorithm. C++ is used with parallel computing in their study. Patzak and Rypil (2012) study a parallel load-balancing framework for FEA with OOP technique.

The libraries and storage modules in programming have lots of benefits. The usage of OOP in FEM (Forde, 1990) increases the performance of maintaining and decreases complexity of design. Different methods like banded matrix storage on global stiffness matrix can be easily applied for decreasing computational cost. The program of Forde (1990) is prepared for solutions of 2D elastic problems and enables to modification with the prepared framework. Yu and Adeli (1993) develop an OO enhanced entity relationship data model for FEA and create a class library by using C++ PL. Also Zeglinski et al. (1994) study OO matrix classes for FEA with C++ PL. Their matrix library includes various matrix objects and different sparse storage schemes for various matrix types. Sparse, banded, symmetric and unsymmetrical full, upper triangular and lower triangular matrices are classified. Kong (1996) studies a data design structure for decreasing data coupling and enhancing hidden data in OO program of FEA with C++ PL. Dubois-Pelerin and Pegon (1997) study on improving modularity of an OO FE program. Data structure and linear equation

solvers are prepared with alternative schemes in OOP technique. Besson and Foerch (1997) develop a large scale OO program for FEA. They use C++ PL with a core library. Mackie (1998) uses OOP for developing a fully interactive FE program, which have multi-threading options. Also Mackie (1999) studies on an OO FE program with respect to data modelling. At the further stage, Mackie (2000) investigates calculation controls on an OO FE program. Three different calculation methods are implanted easily with respect to flexible property of OOP, which are standard linear solution, sub-structure based calculations and concurrent processing. Desitter et al. (2000) construct 1D, 2D and 3D ground-water models with OO FE framework. Mackie (2002) investigates effect of objects calculation control on an OO FEA program. Marczak (2004) constructs an OO framework for BEM. In his study, C++ PL is used and three main classes are constructed. Lejeune et al. (2012) develop an object oriented high order non-linear solver with C++ PL. They focus on OO asymptotic numerical method library.

Other programming languages are used in OOP as; Al-Assaf and Saffarini (2004) construct an OO program for optimizing concrete slabs with Java PL. Optimal design parameters are easily calculated. Nikishkow (2006) develops an OO FEA code with Java PL. The code includes ten Java packages and three methods, which are used for construction of FEM models, solution and visualization. Alves et al. (2013) study on GFEM (generalized FEM) with object oriented environment. GFEM has a relation with partition of unity method for interpolations. Both FEM and GFEM and their similarities and differences are easily constructed in OOP environment. They use Java PL. Anacleto et al. (2013) investigate OO programming of BEM with self-regular formulation, which includes four integral equations; displacement boundary integral, Somigliana's integral identities for displacement, stress and strain. They use Java PL. Nota et al. (2013) study and design OO classes for GFEM, which includes local enrichment functions for improving solution. Python PL is used. Heng and Mackie (2009) construct 5 different design patterns for OO FEA by using C#. Nie et al. (2010) develop an OO FE program for multi-physic applications. It includes adaptive meshing. They use Fortran 2003 with OOP technique. They solve heat transfer, fluid mechanics and structure mechanics problems. Adaptive meshing is used in required local sections. Rauson et al. (2010) mention object construction and basic design patterns of OOP for Fortran 2003.

Some of OOP applications on meshfree methods are available in literature. Krysl and Belytschko (2001) design a library for EFG shape functions with both OO and procedural programming. EFG shape functions have more complicated than FEM shape functions and encapsulation of OOP can decrease the complexity. Seidl and Schmidt (2005) study on OOP for both meshless FEM and classical FEM with C++ PL. Zhang and Subbarayan (2006) develop an OO meshless analyses software with b-spline. They construct a framework with Java PL. Sanchez and Gonçalvesw (2010) study on OOP for meshfree shape functions. They focus on shape functions, which are derived from MLS. They use Java PL and construct a main framework of classes. Barbieri and Meo (2012) use OOP technique with Matlab for RKPM. They mainly focus on increasing speed on construction of shape functions.

## **2.5 Conclusion of Literature Review**

Meshfree methods are mainly developed to avoid the meshing problems in mesh dependent methods like FEM, FDM and other similar methods. The usage of only nodes in this method supports flexibility and simplicity for modelling of problem domain. In large deformation, fracture and impact problems, finite elements show high distortions. In meshfree methods, problems including high distortions in the analysed model are overcome by using only nodes. Re-meshing requirement in many finite element solutions is also eliminated with the usage of meshfree methods.

Some findings are summarized and highlighted from literature review, which are observed commonly in the most of the studies. In meshfree analyses, the analysed model consists of only nodes, which can be distributed random or regular in the problem domain. Literature studies have revealed that regular distributed nodes give more stable, quickly converging and less erroneous results than irregular ones. If irregular node distributions are used, the studies recommend to increase the used number of nodes or the order degree of polynomials for decreasing numerical errors. However the usage of too much number of nodes may cause ill-conditioned matrices and increases the size of Jacobian and other matrices, which will cause higher computational cost. Hence enough and optimum number of nodes must be determined in the analyses. If enough support is achieved for integration, the accuracy of the solution is not affected dominantly by node irregularities and support sizes. The findings in literature show that domain shapes can affect solution results.

Rectangular support domain recommended rather than the usage of circular support domains.

Many different techniques are developed on meshfree methods for increasing accuracy. The usage of RBF in meshfree methods is one of the applications, which is widely used by many researchers. RBF improves accuracy of solutions in the local and global domain applications, but it causes increasing the solution time. The usage of RBF is simple, which only depends on the Euclidean distance between nodes and it contains shape parameters. The effect of shape parameters on the solution has also an important concept in the meshfree studies. In literature, shape parameters are generally selected according to type of the problem and there is no a general optimum shape parameter values. Three different radial basis functions are commonly examined, which are MQ, EXP and TSP. Optimum shape parameters are  $q=1.03$  and  $R=1.42$ , when MQ (multi-quadric) basis functions are used in some literature studies, which gives the best results.

Different interpolation methods are developed and adapted into solution of different problems. The early developed method; EFG contains MLS functions, which prevents volumetric locking. However, this method is not as widely used as FEM, because of its higher computational cost. Also in MLS shape functions, Kronecker delta function property is not satisfied and it requires some extra algorithms in the applications of essential boundary conditions. RPIM shape functions satisfy Kronecker delta function property and can be able to give better results rather than MLS shape functions. But more solution time is also required in the RPIM solutions.

Solution time and computational costs are another critical concept for meshfree methods. Most of them have lower performance than FEM. Nevertheless, researchers improve their solution speed by using low order polynomials. But it cause low accuracy. Both meshless and FEM are used in some cases for decreasing solution time of pure meshfree analyses.

Programming of numerical methods is also considered in the literature studies. Different programming techniques are mentioned and one of them, OOP is widely used, especially in the programming of FEM. Object oriented programming is especially used in complex problems to solve them in a simple and modular way.

C++ programming language is generally preferred in their programming. In old Fortran versions, object oriented programming is not supported. Fortran 2003 partially supports it, but it is not used as much as C++ in OOP. Java is also used to develop object oriented numerical programs that works on different systems such as Windows, Linux and Macintosh.

Some shortcomings are predicted that nodal integration scheme can be used effectively, if its applicability can be developed, especially on the solution of 3D problems. It usually includes fluctuations on stress results. OOP has lots of benefits and is effectively used on FEM. Its usage on meshfree methods can be useful for the spread of the method. Therefore, in this work, our main aim is developing an OO meshfree program. Matlab is a platform that fully supports OOP, which includes suitable functions in its library.

## CHAPTER 3

### MESHFREE INTERPOLATION METHODS

#### 3.1 Introduction

Construction of shape functions is one of the most critical sections in meshfree methods. Their quality must be checked carefully, which directly influences the solution results. MLS and RPIM interpolation methods are commonly used in construction of shape functions in meshfree methods, which are used in this study. Their structure is summarized in this section.

#### 3.2 Moving Least Square Interpolation Method

One of the most used interpolation technique in meshfree shape function construction is MLS interpolation method. This method is widely used, especially in EFG. MLS shape functions (Liu and Gu, 2005) can be derived with initially creation of a field function  $u(x)$  in the problem domain,  $\Omega$ .  $u(x)$  approximation variable is in Eq. 3.1.

$$u^h(x) = \sum_{j=1}^m p_j(x) \times a_j(x) = p^T(x)a(x) \quad (3.1)$$

where  $p(x)$  represents basis functions in spatial coordinates and  $x^T = [x, y]$  for 2D problems, which is usually built with monomials from Pascal triangle to provide minimum completeness.  $m$  represents the number of the basis functions.  $a(x)$  is coefficient vector, which depends on  $x$  and is given in Eq. 3.2 as;

$$a^T(x) = \{a_1(x) \ a_2(x) \dots a_m(x)\} \quad (3.2)$$

$a$  can be (Liu and Gu, 2005) calculated by minimizing weighted discrete  $L_2$  norm. The weight residual function, which includes approximated values and the nodal parameters of the unknown field function is given in Eq. 3.3.

$$J = \sum_{i=1}^n \hat{W}(x-x_i) [p^T(x_i)a(x) - u_i]^2 \quad (3.3)$$



which  $n$  represents number of nodes in the support domain of  $x$ . It is valid for which the weight function is not equal to zero;  $\hat{W}(x-x_i) \neq 0$ .

The partial derivative of  $J$  with respect to  $a(x)$  equals zero

$$\partial J / \partial a = 0 \quad (3.4)$$

That provides the following linear relations.

$$A(x)a(x) = B(x)U_s \quad (3.5)$$

where  $U_s$  includes the nodal parameters of field function for all the nodes in the related support domain.

$$U_s = \{u_1 \ u_2 \ \dots \ u_n\}^T \quad (3.6)$$

$A(x)$  in Eq. 3.5 represents weighted moment matrix, which can be represented as,

$$A(x) = \sum_{i=1}^n \hat{W}_i(x) p(x_i) p^T(x_i) \quad (3.7)$$

where

$$\hat{W}_i(x) = \hat{W}(x-x_i) \quad (3.8)$$

When linear basis is used ( $m=3; p^T(x) = \{1 \ x \ y\}$ ),  $A$  is a symmetric matrix, which has a dimension of  $3 \times 3$ . It can be explicitly written as

$$A_{3 \times 3}(x) = \sum_{i=1}^n \hat{W}_i(x) p(x_i) p^T(x_i) \quad (3.9)$$

$$\begin{aligned} A_{3 \times 3}(x) = & \hat{W}(x-x_1) \begin{bmatrix} 1 & x_1 & y_1 \\ x_1 & x_1^2 & x_1 y_1 \\ y_1 & x_1 y_1 & y_1^2 \end{bmatrix} + \hat{W}(x-x_2) \begin{bmatrix} 1 & x_2 & y_2 \\ x_2 & x_2^2 & x_2 y_2 \\ y_2 & x_2 y_2 & y_2^2 \end{bmatrix} + \\ & \dots + \hat{W}(x-x_n) \begin{bmatrix} 1 & x_n & y_n \\ x_n & x_n^2 & x_n y_n \\ y_n & x_n y_n & y_n^2 \end{bmatrix} \end{aligned} \quad (3.10)$$

$$A_{3 \times 3}(x) = \begin{bmatrix} \sum_{i=1}^n \hat{W}_i & \sum_{i=1}^n x_i \hat{W}_i & \sum_{i=1}^n y_i \hat{W}_i \\ \sum_{i=1}^n x_i \hat{W}_i & \sum_{i=1}^n x_i^2 \hat{W}_i & \sum_{i=1}^n x_i y_i \hat{W}_i \\ \sum_{i=1}^n y_i \hat{W}_i & \sum_{i=1}^n x_i y_i \hat{W}_i & \sum_{i=1}^n y_i^2 \hat{W}_i \end{bmatrix}_{(3 \times 3)} \quad (3.11)$$

$B$  matrix in Eq. 3.5 can be expressed as

$$B(x) = [\hat{W}_1(x)p(x_1) \hat{W}_2(x)p(x_2) \dots \hat{W}_n(x)p(x_n)] \quad (3.12)$$

When linear basis is used,  $B$  matrix, which supports a dimension of  $3 \times n$  matrix can be expressed as;

$$B_{3 \times n}(x) = \left[ \hat{W}(x-x_1) \begin{bmatrix} 1 \\ x_1 \\ y_1 \end{bmatrix} \hat{W}(x-x_2) \begin{bmatrix} 1 \\ x_2 \\ y_2 \end{bmatrix} \dots \hat{W}(x-x_n) \begin{bmatrix} 1 \\ x_n \\ y_n \end{bmatrix} \right] \quad (3.13)$$

$$B_{3 \times n}(x) = \begin{bmatrix} \hat{W}_1 & \hat{W}_2 & \dots & \hat{W}_n \\ x_1 \hat{W}_1 & x_2 \hat{W}_2 & \dots & x_n \hat{W}_n \\ y_1 \hat{W}_1 & y_2 \hat{W}_2 & \dots & y_n \hat{W}_n \end{bmatrix}_{(3 \times n)} \quad (3.14)$$

If Eq. 3.5 is solved for  $a(x)$ , it can be written as;

$$a(x) = A^{-1}(x)B(x)U_s \quad (3.15)$$

Eq. 3.15 is substituted into Eq. 3.1, new form can be written as;

$$u^h(x) = \sum_{i=1}^n \phi_i(x)u_i = \Phi^T(x)U_s \quad (3.16)$$

where  $\Phi^T(x)$  represents (Liu and Gu, 2005) vector of shape functions, which is derived from MLS. It is written for  $n$  number of nodes in the support domain of the point  $x$  as;

$$\Phi^T(x) = \{\phi_1(x) \phi_2(x) \dots \phi_n(x)\}_{(1 \times n)} = p^T(x)_{(1 \times 3)} A^{-1}(x)_{(3 \times 3)} B(x)_{(3 \times n)} \quad (3.17)$$

For a selected node  $i$ th, the shape function  $\phi_i(x)$  is given as

$$\phi_i(x) = \sum_{j=1}^m p_j(x) (A^{-1}(x)B(x))_j = p^T(x) (A^{-1}B)_i \quad (3.18)$$

MLS shape functions can be continuous in the entire global domain, if suitable weight functions are used. Hence Eq. 3.17 can be rewritten in 3.19 for obtaining the partial derivatives of shape functions;

$$\Phi^T(x) = \gamma^T(x)B(x) \quad (3.19)$$

where

$$\gamma^T(x) = p^T A^{-1} \quad (3.20)$$

Eq. 3.20 can be regulated, since  $A$  matrix is symmetric;

$$A\gamma = p \quad (3.21)$$

The partial derivatives of  $\gamma$  can be created as;

$$A\gamma_{,i} = p_{,i} - A_{,i}\gamma \quad (3.22)$$

$$A\gamma_{,ij} = p_{,ij} - (A_{,i}\gamma_{,j} + A_{,j}\gamma_{,i} + A_{,ij}\gamma) \quad (3.23)$$

$$A\gamma_{,ijt} = p_{,ijt} - (A_{,i}\gamma_{,jt} + A_{,j}\gamma_{,it} + A_{,t}\gamma_{,ij} + A_{,ij}\gamma_{,t} + A_{,it}\gamma_{,j} + A_{,jt}\gamma_{,i} + A_{,ijt}\gamma) \quad (3.24)$$

$i, j$  and  $t$  in Eq. 3.22, 3.23 and 3.24 represent coordinates in  $x$  and  $y$ . Comma represents partial derivative with respect to the indicated spatial coordinate. It is applied for partial derivatives of the shape function  $\Phi$ , which is given in below equations.

$$\Phi^T_{,i} = \gamma^T_{,i}B + \gamma^TB_{,i} \quad (3.25)$$

$$\Phi^T_{,ij} = \gamma^T_{,ij}B + \gamma^T_{,i}B_{,j} + \gamma^T_{,j}B_{,i} + \gamma^TB_{,ij} \quad (3.26)$$

$$\Phi^T_{,ijt} = \gamma^T_{,ijt}B + \gamma^T_{,ij}B_{,t} + \gamma^T_{,it}B_{,j} + \gamma^T_{,jt}B_{,i} + \gamma^T_{,i}B_{,jt} + \gamma^T_{,j}B_{,it} + \gamma^T_{,t}B_{,ij} + \gamma^TB_{,ijt} \quad (3.27)$$

### 3.3 Radial Point Interpolation Method

Relation between nodes in meshfree methods is mainly obtained with interpolations. Various interpolation methods are available in literature. RPIM (Wang and Liu, 2002a) is one of interpolation technique and widely used. It includes PIM (Liu and Gu, 2001) with radial basis functions. When shape functions have been constructed, a field function  $u(x)$  is created and given as in Eq. 3.28, which is consist of polynomial and basis functions.

$$u(x) = \sum_{i=1}^n R_i(x) \times a_i + \sum_{j=1}^m P_j(x) \times b_j = R^T(x) \times a + P^T(x) \times b \quad (3.28)$$

$R_i(x)$  and  $P_j(x)$  represent radial basis and polynomial basis functions, respectively.  $a_i$  and  $b_j$  are related constants,  $n$  is the number of field nodes in the local support domain and  $m$  is the number of polynomial terms. Interpolations between nodes are mainly accomplished within the local support domain for each node or point of interests.

Various RBF (Liu and Gu, 2005; Liu, 2009) are available in literature, like multi-quadrics (MQ), the Gaussian (Exp), the thin plate spline (TSP) and logarithmic radial basis functions. MQ is (Liu and Gu, 2005; Dinis et al., 2007) used as radial basis function in Eq. 3.29.

$$R_i(x) = (r_i^2 + (\alpha_c \times d_c)^2)^q \quad (3.29)$$

where  $d_c$  is usually set as average nodal spacing near the point of interest at  $x$ ;  $\alpha_c$  and  $q$  are shape parameters. It is (Liu and Gu, 2005; Liu, 2009) recommended to use  $q$  as 1.03 and  $\alpha_c$  as 3.00 for MQ basis function. The radial distance is given in Eq. 3.30 for 2D and in Eq. 3.31 for 3D cases.

$$r_i(x) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (3.30)$$

$$r_i(x) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \quad (3.31)$$

The polynomial terms are given in Eq. 3.32 for 2D and Eq. 3.33 for 3D cases which are mainly derived from binomial expansion.

$$p^T(x) = \{1, x, y, x^2, xy, y^2, \dots\} \quad (3.32)$$

$$p^T(x) = \{1, x, y, z, x^2, xy, y^2, yz, z^2, zx, \dots\} \quad (3.33)$$

The binomial expansion is given in Fig. 3.1 for 3D cases. Its  $x^n$  and  $y^n$  components are used for 2D cases.

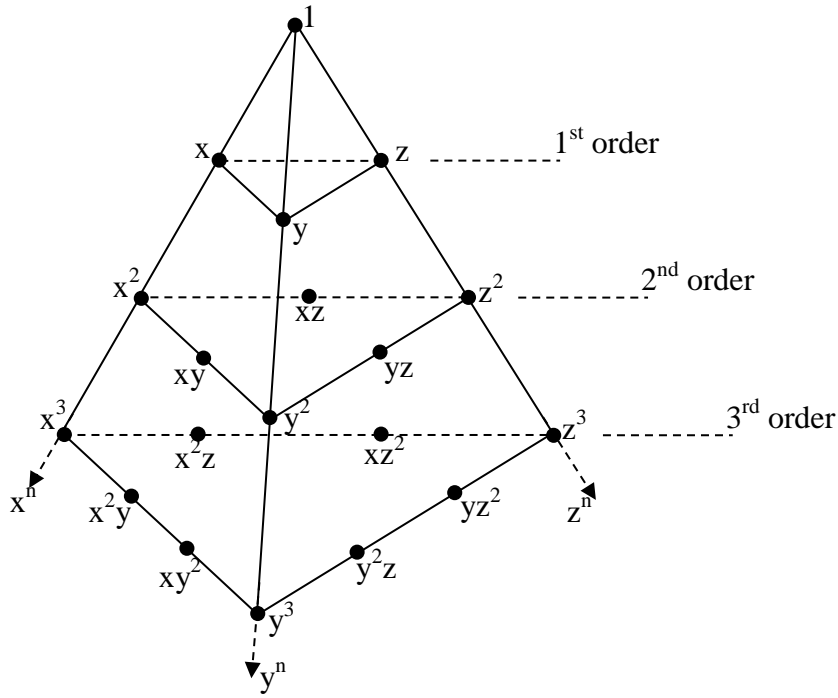


Figure 3.1. Binomial expansion for 3D polynomial terms

Interpolation is applied in a support domain for point of interest. Different support domain geometries can be used, like circular, elliptical, triangular or rectangular. Illustration of circular and rectangular local support domain for point of interest is given in Fig. 3.2.

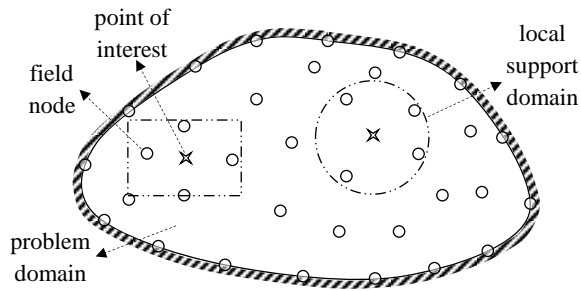


Figure 3.2. Different types of local support domains for point of interests (Liu and Gu, 2005)

A circular local support domain is used and its covered area is given by radius of circle ( $d_s$ ), which is given in Eq. 3.34.

$$d_s = \alpha_s \times d_c \quad (3.34)$$

where  $d_c$  is average nodal spacing and  $\alpha_s$  is a positive real number of dimensionless size of the local support domain. Its value (Liu and Gu, 2005; Dinis et al., 2007) is commonly used between 2.00 and 3.00. The unknown constants of field function of  $a_i$  and  $b_j$  in Eq. 3.28 can be determined by enforcing the field function to pass through all  $n$  field nodes in the local support domain. At the  $k^{th}$  point or last point in a local support domain, field function can be written in Eq. 3.35 for 2D and in Eq. 3.36 for 3D cases:

$$u(x_k, y_k) = \sum_{i=1}^n R_i(x_k, y_k) \times a_i + \sum_{j=1}^m P_j(x_k, y_k) \times b_j \quad k=1,2,\dots,n \quad (3.35)$$

$$u(x_k, y_k, z_k) = \sum_{i=1}^n R_i(x_k, y_k, z_k) \times a_i + \sum_{j=1}^m P_j(x_k, y_k, z_k) \times b_j \quad k=1,2,\dots,n \quad (3.36)$$

The matrix form of Eq. 3.35 or Eq. 3.36 can be expressed as;

$$U_e = R_q \times a + P_m \times b = \{u_1 \ u_2 \ \dots \ u_n\}^T \quad (3.37)$$

where  $U_e$  is the vector of function values at the nodes in the local support domain.

$R_q$  is the moment matrix of RBF and it is given in Eq. 3.39.

$$R_q = \begin{bmatrix} R_1(r_1) & R_2(r_1) & \dots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \dots & R_n(r_2) \\ \dots & \dots & \dots & \dots \\ R_1(r_n) & R_2(r_n) & \dots & R_n(r_n) \end{bmatrix}_{(n \times n)} \quad (3.38)$$

$P_m$  is the polynomial moment matrix (Liu and Gu, 2005), which are given in Eq. 3.39 and 3.40 for 2D and 3D cases, respectively.

$$P_m = \begin{bmatrix} 1 & x_1 & y_1 & \dots & \dots & P_m(r_1) \\ 1 & x_2 & y_2 & \dots & \dots & P_m(r_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & y_n & \dots & \dots & P_m(r_n) \end{bmatrix}_{(n \times m)} \quad (3.39)$$

$$P_m = \begin{bmatrix} 1 & x_1 & y_1 & z_1 & \dots & P_m(r_1) \\ 1 & x_2 & y_2 & z_2 & \dots & P_m(r_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & y_n & z_n & \dots & P_m(r_n) \end{bmatrix}_{(n \times m)} \quad (3.40)$$

$a$  and  $b$  are vectors of unknown coefficients for radial and polynomial basis functions respectively. They are given in Eq. 3.41 and 3.42.

$$a^T = \{a_1 a_2 \dots a_n\} \quad (3.41)$$

$$b^T = \{b_1 b_2 \dots b_n\} \quad (3.42)$$

For solution of field function, unknown parameter  $a$  in Eq. 3.37 must satisfy in polynomial function,

$$\sum_{j=1}^n p_j(x_i) \times a_i = P_m^T \times a = 0 \quad j=1,2,\dots,m \quad (3.43)$$

Combination of Eq. 3.37 and Eq. 3.28 yields the following equations in the matrix form:

$$\tilde{U}_e = \begin{bmatrix} U_e \\ 0 \end{bmatrix} = \begin{bmatrix} R_q & P_m \\ P_m^T & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix} = G \times a_0 \quad j=1,2,\dots,m \quad (3.44)$$

where

$$\tilde{U}_e = \begin{bmatrix} U_e \\ 0 \end{bmatrix} = \{a_1 a_2 \dots a_n 0 0 \dots 0\}^T \quad (3.45)$$

Unique solution is obtained, if inverse of matrix  $G$  exists:

$$a_0 = \begin{Bmatrix} a \\ b \end{Bmatrix} = G^{-1} \times \tilde{U}_e \quad (3.46)$$

Substituting Eq. 3.46 into Eq. 3.28, interpolation with respect to field function can be expressed as;

$$u(x) = \{R^T(x) \times P^T(x)\} \times G^{-1} \times \tilde{U}_e = \tilde{\varphi}(x) \times \tilde{U}_e \quad (3.47)$$

Finally (Liu and Gu, 2005; Dinis et al., 2007), RPIM shape functions for the corresponding  $n$  field nodes can be obtained as;

$$\varphi^T(x) = \{\varphi_1(x) \varphi_2(x) \dots \varphi_n(x)\} \quad (3.48)$$

The approximation function can be written as;

$$u(x) = \varphi^T(x) \times U_e = \sum_{i=1}^n \varphi_i \times u_i \quad (3.49)$$

The derivatives of  $u(x)$  can be easily obtained as

$$u_{i,k}(x) = \varphi_{,k}^T(x) \times U_e \quad (3.50)$$

where  $k$  denotes the coordinates  $x$  or  $y$  in 2D and  $x$ ,  $y$  or  $z$  in 3D cases. Partial differentiation is taken with respect to that defined coordinated by  $k$ .



## CHAPTER 4

### OBJECT-ORIENTED PROGRAMMING IN MESHFREE ANALYSIS

#### 4.1 Introduction

The usage and implementation of numerical methods on computer programs are also important, as well as their developments. Computer coding of numerical methods is a very different kind of study and applicability of numerical methods depends on the coding performance. A well designed program contains less error and is easily handled. There are lots of codes which have different kinds of structures, even if they have small sizes. Procedural programming is commonly used in coding of these programs, which includes less procedure. But addition of more codes, new functions, elements and items causes increasing conflicts, even if subroutines or other auxiliary modules are used. One of the programming techniques, object-oriented programming (OOP) is used for preventing complexity, which has capable of key-lock property and can group similar structures. This key-lock property supports to avoid unnecessary usages of programs. Hence OOP can be capable to use in programming of numerical methods. Developments in numerical methods provide different solution techniques like FDM, BEM, FEM and so on. The usage of FEM is widely known, especially in the solutions of solid mechanics problems. Solution procedure includes some structures (interpolations, transformations, construction of shape functions and application of essential and natural boundary conditions) which must work in harmony together. Any conflicts may cause wrong results. Hence object-oriented programming techniques are widely used in finite and boundary element in literature. Further developments or extension sections of numerical methods can be easily applied with OOP.

#### 4.2 Object-oriented Programming (OOP) and Matlab

OOP is an appropriate programming technique for large codes and based on a hierarchy of classes, which cooperate their objects. It mainly includes three major components; encapsulation, inheritance and polymorphism. The working structure of

object is protected and hidden by encapsulation. Input/output functions are mainly considered. Inheritance can construct new classes in terms of old classes. Polymorphism is the ability to use an operator or method in different ways.

Matlab is a programming environment for algorithm development, data analysis, visualization, numerical computation and widely used in scientific and engineering applications. It also supports OOP technique with various class operators. There are two different ways to define classes in Matlab. The first way is using `value class` and the second is using `handle class`. A `value class` constructor returns an instance that is associated with the variable, which it assigned. A `handle class` constructor returns a handle object that is a reference to the object created. `handle class` is used as a base class of all developed classes in this study. This class also controls hierarchy of its subclasses.

### 4.3 Meshfree Classes and Their Implementations in a Matlab Program

In this work, seven different classes are mainly used. However, three of them are especially related with meshfree techniques; `inte`, `sdo` and `sf` classes. They have their own sub-classes. Some sub-classes have further sub-classes. In the subsections of this chapter, they are discussed in detail. Basic and special class properties are explained. Classes and its properties mainly manage objects and programming schemes. Defined properties in the class will be assigned to objects. Hence required and related properties must be determined for setting efficient object management. Thus the programmed method and its logic must be well known.

The basic concept of elasto-static problems is the solution of equilibrium equations with respect to applied natural and essential boundary conditions. Their equations are given in Eq. 4.1, 4.2 and 4.3, respectively, which are mainly concerned in the application of object-oriented meshfree program for solution of elasto-static problems.

$$L^T \sigma + b = 0 \quad \text{in } \Omega \quad (4.1)$$

$$\sigma n = \bar{t} \quad \text{on } \Gamma_t \quad (4.2)$$

$$u = \bar{u} \quad \text{on } \Gamma_u \quad (4.3)$$

There are several methods available for solution of these equations. Strong and weak form formulations are defined for solution of partial differential equations (PDEs). LRPIM and MLPG methods can handle weak form PDEs. Hence, the equilibrium equation (Liu and Gu, 2005) in Eq. 4.1 can be written in weak form with respect to weighted residual methods for LRPIM as

$$\int_{\Omega_q} \hat{W}_I (\sigma_{ij,j} + b_i) d\Omega = 0 \quad (4.4)$$

where  $\hat{W}_I$  is the weight function at related node. The application of natural and essential boundary conditions on weak form of equilibrium equation for local quadrature domain of a related node is given in Eq. 4.5. This local quadrature domain can intersect with natural and essential boundaries, which need a special formulation on construction of stiffness matrix that related locations. Three boundary locations are determined with respect to natural and essential boundary conditions.  $\Gamma_{qi}$  represents not intersection locations of local quadrature domain with global boundary,  $\Gamma_{qt}$  represents intersection locations with natural boundary and  $\Gamma_{qu}$  represents intersection locations of essential boundary locations. More detail information can be achieved in the study of Liu and Gu (2005).

$$\int_{\Gamma_{qi}} \hat{W}_I n_j \sigma_{ij} d\Gamma + \int_{\Gamma_{qu}} \hat{W}_I n_j \sigma_{ij} d\Gamma + \int_{\Gamma_{qt}} \hat{W}_I n_j \sigma_{ij} d\Gamma - \int_{\Omega_q} [\hat{W}_{I,j} \sigma_{ij} - \hat{W}_I b_i] d\Omega = 0 \quad (4.5)$$

Eq. 4.5 can be written with application of stresses. Application of tractions at inside of natural boundary conditions is given in Eq. 4.6.

$$\int_{\Omega_q} \hat{W}_{I,j} \sigma_{ij} d\Omega - \int_{\Gamma_{qi}} \hat{W}_I t_i d\Gamma - \int_{\Gamma_{qu}} \hat{W}_I t_i d\Gamma = \int_{\Gamma_{qt}} \hat{W}_I \bar{t}_i d\Gamma + \int_{\Omega_q} \hat{W}_I b_i d\Omega \quad (4.6)$$

Similar equilibrium equation is valid for MLPG method and its local weak formulation is given in Eq. 4.7. Same applications of natural and essential boundary conditions are valid as LRPIM method. In addition, equilibrium equation of MLPG method includes a curve integral for application of essential boundary conditions where local quadrature domain intersects with essential boundary  $\Gamma_{qu}$ . But, on the contrary, to use RPIM shape functions, MPLG method uses MLS shape functions, which do not support Kronecker delta function property. Hence essential boundary

conditions are enforced with penalty method. A penalty method with factor ( $\alpha$ ) is included, which is used as  $10^8$ .

$$\int_{\Omega_q} \hat{W}_I(\sigma_{ij} + b_i) d\Omega - \alpha \int_{\Gamma_{qu}} \hat{W}_I(u_i - \bar{u}_i) d\Gamma = 0 \quad (4.7)$$

The solution of these equations includes various components, which are classified for decreasing complexity in classes of OOP.

### 4.3.1 mc (Model Constructor) Class

mc (model constructor) class is mainly used for construction of basic numerical model, has two main properties: *young* and *anu*. They represent Young's modulus and Poisson's ratio, respectively. Its hierarchy diagram and Matlab codes are given in Fig. 4.1 and 4.2. It does not include any methods. Its properties are transferred to classes of material and nodes. Material properties must be known for calculation of strain and stresses with respect to Hooke's law, which is given in Eq. 4.8.

$$\sigma = D \varepsilon \quad (4.8)$$

material class is defined for materials of the model and has two own properties: *you* and *dmat*. *dmat* includes material matrix for isotropic materials in its property with respect to plane stress and strain, which are given in Eq. 4.9 and 4.10, respectively. In addition to them, different material models can be easily added with *dmat* property.

$$D = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)^2 \end{bmatrix} \quad (4.9)$$

$$D = \frac{E(1-\nu)}{(1-2\nu)(1+\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \quad (4.10)$$

The class of *nodes* is used to read all data, which are related geometrical and meshfree parameter properties for the model and has three different sub-classes:

xlsinputnodes, textinputnodes and randtextinputnodes. xlsinputnodes class is used to read all its data from Excel files. If data is given in a text file, they can be read from textinputnodes class. Node coordinate can be obtained randomly by using Matlab statements in the class of randtextinputnodes.

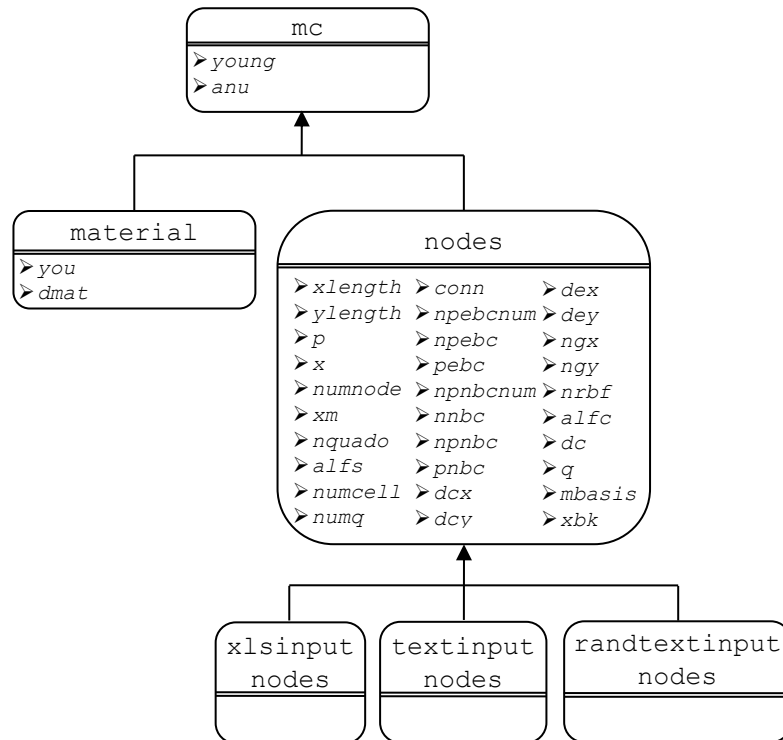


Figure 4.1. mc class hierarch

```

classdef mc < handle
    properties (GetAccess = public, SetAccess = protected)
        young; anu
    end
end
classdef material < mc
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Mat=material(young,anu)...
        function Mat=dmatcalc(Mat)...
    end
end
classdef nodes < mc
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Node=nodes(ylength)...
    end
end
classdef randtextinputnodes < nodes
    methods...
end
classdef textinputnodes < nodes
    methods...
end
classdef xlsinputnodes < nodes
    methods...
end
end
  
```

Figure 4.2. Matlab codes of mc, its inherited classes and member functions

### 4.3.2. inte (Integration Cell) Class

Integration is one of the basic procedures in meshfree methods. Different integration methods are available and integration domains can be variable. Therefore, `inte` class is defined for local and global quadrature domains as shown in Fig. 4.3. Its Matlab codes are given in Fig. 4.4. The properties are inherited to the `qdomain`, `gausscoefficient`, `domaingausspoints`, `localquadraturedomain` and `globalquadraturedomain` classes. The size of quadrature domain is calculated in `qdomain` class and recorded in `ds` property. The weights are calculated with respect to the number of sampling points in `gausscoefficient` class and recorded in `gauss` property. A local quadrature domain is determined for a sampling point in `localquadraturedomain` class. The determined local quadrature domain is further subdivided in `subdivisionlqdomain` class. In `domaingausspoints` class, gauss points are calculated for a divided quadrature domain and recorded in `gss` property. Global quadrature domain can be used directly with `globalquadraturedomain` class.

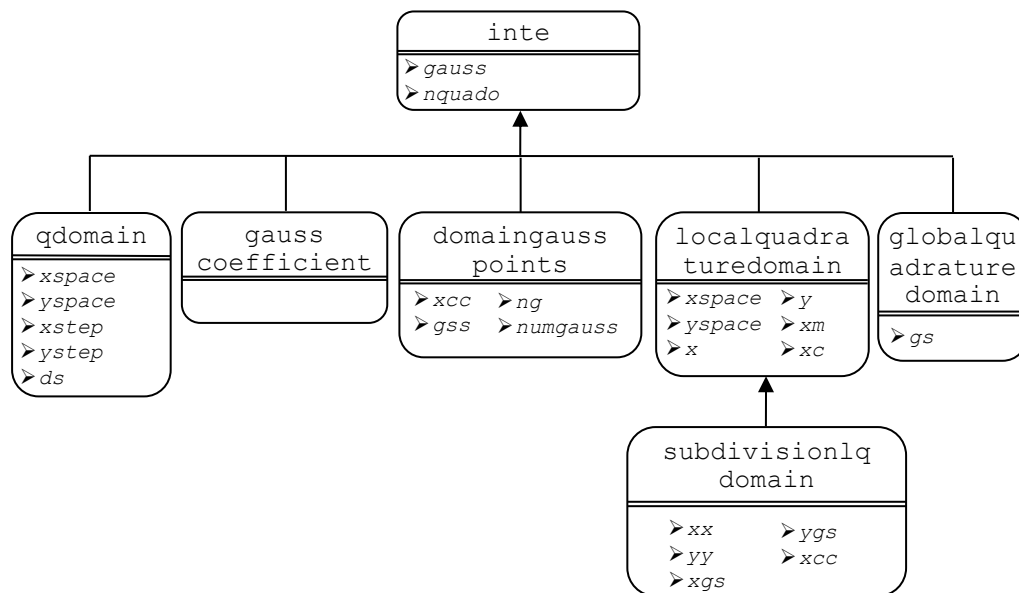


Figure 4.3. `inte` class hierarch

```

classdef inte < handle
    properties (GetAccess = public, SetAccess = protected)
        gauss
        nquado
    end
end
classdef qdomain < inte
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Qnode=qdomain(nquado)...
        function Qnode=qdomaincalc(Qnode,Node)...
    end
end
classdef gausscoefficient < inte
    methods
        function Gaussco=gausscoefficient(nquado)...
        function Gaussco=gausscoefficientcalc(Gaussco)...
    end
end
classdef localquadraturedomain < inte
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Qdomain=localquadraturedomain(xspace,yspace,x,y,xm)...
        end
        function Qdomain=localquadraturedomaincalc(Qdomain)...
    end
end
end
...

```

Figure 4.4. Matlab codes of inte, its inherited classes and member functions

**4.3.3. sdo (Support Domain) Class**

The number of nodes in a support domain and their coordinates are determined in sdo class. Its hierarchy diagram and Matlab codes are given in Fig. 4.5 and 4.6. It has a single basic property, gpos. It is inherited to its sub-classes of supportdomain and testfunc. supportdomain class determines the number of nodes and their labels in a support domain. In this process, three different support domain geometries can be used. They are given in rectangulardomain, triangulardomain and circulardomain classes. In Eq. 4.11, size of support domain is given with circular distance ( $d_s$ ) for a circular support domain.  $d_c$  is usually selected as average nodal spacing and  $\alpha_s$  is a constant and used as (Liu and Gu, 2005; Liu, 2009) between 2.00 and 3.00 generally.

$$d_s = \alpha_s \times d_c \tag{4.11}$$

The number of nodes and their labels in support domain for integration are stored in ndex and nv properties. testfunc class computes the weights and stores them in w property. Different weight functions (Liu and Gu, 2005) are available like the cubic spline function ( $W_1$ ), quartic spline weight function ( $W_2$ ) and other weight functions.  $W_1$  and  $W_2$  are given in Eq. 4.12 and 4.13, respectively. Both LRPIM and

MPLG methods can use these weight functions in the solution of equilibrium equation.

$$\hat{W}_i(x) = \begin{cases} 2/3 - 4\bar{r}_i^2 + 4\bar{r}_i^3 & \bar{r}_i \leq 0.5 \\ 4/3 - 4\bar{r}_i + 4\bar{r}_i^2 + 4/3\bar{r}_i^3 & 0.5 \leq \bar{r}_i \leq 1.0 \\ 0 & 1 < \bar{r}_i \end{cases} \quad (4.12)$$

$$\hat{W}_i(x) = \begin{cases} 1 - 6\bar{r}_i^2 + 8\bar{r}_i^3 - 3\bar{r}_i^4 & \bar{r}_i \leq 1.0 \\ 0 & \bar{r}_i > 1.0 \end{cases} \quad (4.13)$$

Matrices of weight functions ( $\hat{W}$ ) and their derivatives ( $\hat{V}_t$ ) of related point are given in Eq. 4.14 and 4.15.

$$\hat{W}_t = \hat{W}(x, x_t)_{(2 \times 2)} = \begin{bmatrix} \hat{W}(x, x_t) & 0 \\ 0 & \hat{W}(x, x_t) \end{bmatrix} \quad (4.14)$$

$$\hat{V}_t = \hat{V}(x, x_t)_{(3 \times 2)} = \begin{bmatrix} \hat{W}_{,x}(x, x_t) & 0 \\ 0 & \hat{W}_{,y}(x, x_t) \\ \hat{W}_{,y}(x, x_t) & \hat{W}_{,x}(x, x_t) \end{bmatrix} \quad (4.15)$$

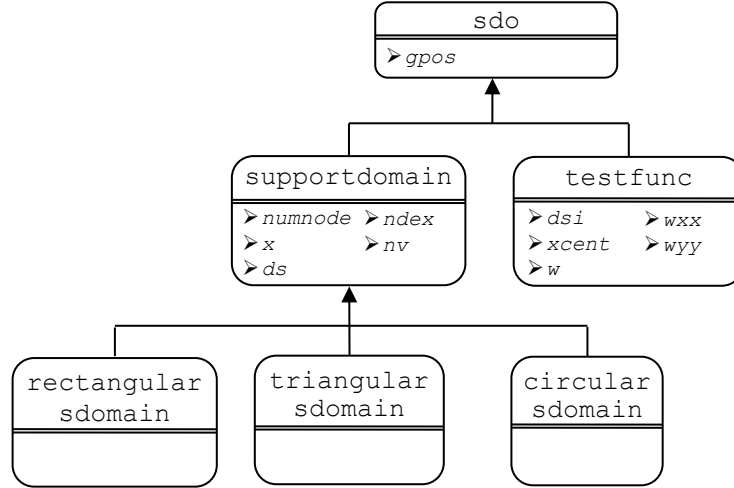


Figure 4.5. sdo class hierarch



```

classdef sdo < handle
    properties (GetAccess = public, SetAccess = protected)
        gpos
    end
end
classdef testfunc < sdo
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Test=testfunc(dsi,xcent,gpos)...
            function Test=testfuncalc(Test)...
        end
    end
end
classdef supportdomain < sdo
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Sdomain=supportdomain(numnode,gpos,x,ds,ndex)...
            function Sdomain=supportdomaincalc(Sdomain)...
        end
    end
end
classdef rectangulardomain < supportdomain
    methods
        function Rectsdomain=rectangulardomain(numnode,gpos,x,ds,ndex)...
            function Rectsdomain=rectangulardomaincalc(Rectsdomain)...
        end
    end
end
...

```

Figure 4.6. Matlab codes of sdo class and its inherited classes and member functions

#### 4.3.4 sf (Shape Function) Class

Most meshfree methods are classified according to shape function construction. Therefore, sf class is defined for RPIM and MLS shape functions as shown in Fig. 4.7. Their Matlab codes are given in Fig. 4.8. It has seven properties: ndex, mbasis, gpos, x, nv, phi and numnode. Formulation of interpolation methods are also given again for pointing sf class operations and properties. In rpim class, shape functions and their derivatives are obtained by computeradialbasis class. computeradialbasis includes basis functions and polynomial terms with their derivatives in rk property. Polynomial terms are given in Eq. 4.16, which are mainly derived from binomial expansion.

$$p^T(x) = \{1, x, y, x^2, xy, y^2, \dots\} \quad (4.16)$$

Radial basis functions are calculated (Liu and Gu, 2005) using one of its sub-classes of mq (multi-quadrics), exp (Gaussian) or tsp (thin plate spline). multi-quadrics (MQ), the Gaussian (Exp), the thin plate spline (TSP) function are given in Eq. 4.17, 4.18 and 4.19, respectively.

$$R_i(x, y) = \exp \left[ -\alpha_c \left( \frac{r_i}{d_c} \right)^2 \right] \quad (4.17)$$

$$R_i(x, y) = r_i^n \quad (4.18)$$

$$R_i(x, y) = r_i^n \log r_i \quad (4.19)$$

$d_c$  is the average nodal spacing near the point of interest at  $x$ ;  $\alpha_c$  and  $n$  are two arbitrary real numbers of dimensionless parameters, which are called shape parameters. It is (Liu and Gu, 2005; Liu, 2009) recommended to use  $q$  as 1.03 and  $\alpha_c$  as 3.00 for MQ basis function. The radial distance is given in Eq. 4.20 for 2D.

$$r_i(x) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (4.20)$$

Shape functions can be constructed by using a field function,  $u(x)$  which includes polynomial and basis functions and is given in Eq 4.21.

$$u(x) = \sum_{i=1}^n R_i(x) \times a_i + \sum_{j=1}^m P_j(x) \times b_j = R^T(x) \times a + P^T(x) \times b \quad (4.21)$$

$R_i(x)$  and  $P_j(x)$  represent radial basis and polynomial basis functions, respectively.  $a_i$  and  $b_j$  are related constants,  $n$  is the number of field nodes in the local support domain and  $m$  is the number of polynomial terms. Interpolations between nodes are mainly accomplished within the local support domain for each node or point of interests.

The matrix form of the above equation can be expressed in Eq. 4.22.  $U_e$  is the vector of function values at nodes in the local support domain.  $R_q$  is the moment matrix and  $P_m$  is the polynomial moment matrix (Liu and Gu, 2005), which are given in Eq. 4.23 and 4.24, respectively.

$$U_e = R_q \times a + P_m \times b = \{u_1 \ u_2 \ \dots \ u_n\}^T \quad (4.22)$$

$$R_q = \begin{bmatrix} R_1(r_1) & R_2(r_1) & \dots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \dots & R_n(r_2) \\ \dots & \dots & \dots & \dots \\ R_1(r_n) & R_2(r_n) & \dots & R_n(r_n) \end{bmatrix}_{(n \times n)} \quad (4.23)$$

$$P_m = \begin{bmatrix} 1 & x_1 & y_1 & \dots & P_m(x_1) \\ 1 & x_2 & y_2 & \dots & P_m(x_2) \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & y_n & \dots & P_m(x_n) \end{bmatrix}_{(n \times m)} \quad (4.24)$$

$a$  is the vector of unknown coefficients for RBF and  $b$  is the vector of unknown coefficients for polynomial basis functions. They are given in Eq. 4.25 and 4.26.

$$a^T = \{a_1 \ a_2 \ \dots \ a_n\} \quad (4.25)$$

$$b^T = \{b_1 \ b_2 \ \dots \ b_n\} \quad (4.26)$$

For solution of field function, unknown parameter  $a$  in Eq. 4.21 must satisfy in polynomial function;

$$\sum_{i=1}^n p_j(x_i) \times a_i = P_m^T \times a = 0 \quad j=1, 2, \dots, m \quad (4.27)$$

Combination of Eq. 4.21 and Eq. 4.22 supports the Eq. 4.28.

$$\tilde{U}_e = \begin{bmatrix} U_e \\ 0 \end{bmatrix} = \begin{bmatrix} R_q & P_m \\ P_m^T & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix} = G \times a_0 \quad j=1,2,\dots,m \quad (4.28)$$

where

$$\tilde{U}_e = \begin{bmatrix} U_e \\ 0 \end{bmatrix} = \{a_1 \ a_2 \ \dots \ a_n \ 0 \ 0 \ \dots \ 0\}^T \quad (4.29)$$

Unique solution is obtained if inverse of matrix  $G$  exists:

$$a_0 = \begin{Bmatrix} a \\ b \end{Bmatrix} = G^{-1} \times \tilde{U}_e \quad (4.30)$$

Substituting Eq. 4.30 into Eq. 4.21, interpolation with respect to field function can be expressed as;

$$u(x) = \{R^T(x) \times P^T(x)\} \times G^{-1} \times \tilde{U}_e = \tilde{\varphi}(x) \times \tilde{U}_e \quad (4.31)$$

Finally (Liu and Gu, 2005), RPIM shape functions for the corresponding  $n$  field nodes can be obtained as;

$$\varphi^T(x) = \{\varphi_1(x) \ \varphi_2(x) \ \dots \ \varphi_n(x)\} \quad (4.32)$$

The approximation function can be written as;

$$u(x) = \varphi^T(x) \times U_e = \sum_{i=1}^n \varphi_i \times u_i \quad (4.33)$$

The derivatives of  $u(x)$  can be easily obtained as;

$$u_{i,k}(x) = \varphi_{,k}^T(x) \times U_e \quad (4.34)$$

where  $k$  denotes the coordinates and partial differentials are taken with respect to that defined coordinated by  $k$ .

In mls class, the mls shape functions and their derivatives are obtained by using `compute_basis` and `compute_ab` sub-classes. `compute_basis` class computes base functions and their derivatives. `compute_ab` class computes the weights by using `weight_w1` and `weight_w2` and finally computes shape functions and their derivatives. When MLS shape functions are constructed, the field function  $u(x)$  can be written with approximate formation of MLS in Eq. 4.35.  $a(x)$  is unknown coefficients and they are shown in Eq. 4.36. These coefficients are included to  $L_2$  norm for finding their values in Eq. 4.37. When minimizing  $L_2$  norm with applying  $\partial J / \partial \alpha = 0$ , a linear equation can be achieved in Eq. 4.38.

$$u(x) = \sum_{j=1}^m P_j(x) \times a_j(x) = P^T(x) \times a(x) \quad (4.35)$$

$$a^T = \{a_1(x) \ a_2(x) \ \dots \ a_m(x)\} \quad (4.36)$$

$$J = \sum_{i=1}^n \hat{W}(x - x_i) [P^T(x_i) a(x) - u_i]^2 \quad (4.37)$$

$$A(x) a(x) = B(x) U_s \quad (4.38)$$

As same as right hand side of Eq. 4.22, field functions can be expressed as;

$$U_s = \{u_1 \ u_2 \ \dots \ u_n\}^T \quad (4.39)$$

In Eq. 4.38,  $A(x)$ ,  $B(x)$  and  $a(x)$  can be expressed as in Eq. 4.40, 4.41 and 4.42, respectively;

$$A(x) = \sum_{i=1}^n \hat{W}_i(x) p(x_i) p^T(x_i) \quad (4.40)$$

$$B(x) = [\hat{W}_1(x)p(x_1) \ \hat{W}_2(x)p(x_2) \ \dots \ \hat{W}_n(x)p(x_n)] \quad (4.41)$$

$$a(x) = A^{-1}(x)B(x)U_s \quad (4.42)$$

Also  $a(x)$  can supply field function and the field function can be written with respect to shape functions as;

$$u^h(x) = \sum_{i=1}^n \phi_i(x)u_i = \Phi^T(x)U_s \quad (4.43)$$

$\Phi(x)$  represents the constructed shape functions. When they are regulated, they can be written as;

$$\Phi^T(x) = \{\phi_1(x) \ \phi_2(x) \ \dots \ \phi_n(x)\}_{(1 \times n)} = P^T(x)A^{-1}(x)B(x) \quad (4.44)$$

All calculated shape functions and their derivatives are stored in the property of `phi`. Strain matrix is constructed with RPIM or MLS shape functions and it is given in Eq. 4.45.

$$B_{(3 \times 2n)} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} & 0 & \frac{\partial \phi_2}{\partial x} & 0 & \dots & \frac{\partial \phi_n}{\partial x} & 0 \\ 0 & \frac{\partial \phi_1}{\partial y} & 0 & \frac{\partial \phi_2}{\partial y} & \dots & 0 & \frac{\partial \phi_n}{\partial y} \\ \frac{\partial \phi_1}{\partial y} & \frac{\partial \phi_1}{\partial x} & \frac{\partial \phi_2}{\partial y} & \frac{\partial \phi_2}{\partial x} & \dots & \frac{\partial \phi_n}{\partial y} & \frac{\partial \phi_n}{\partial x} \end{bmatrix} \quad (4.45)$$

Further step will be construction of stiffness and force matrices with respect to constructed shape functions. Stiffness formulation of LRPIM is given in Eq. 4.46 and it is regulated with Gauss quadrature in Eq. 4.47 for integration.  $J_{qi}$  and  $J_{qu}$  are Jacobean matrices at curve boundary locations.

$$K_I = \int_{\Omega_q} \hat{V}_I^T DBd\Omega - \int_{\Gamma_{qi}} \hat{W}_I^T nDBd\Gamma - \int_{\Gamma_{qu}} \hat{W}_I^T nDBd\Gamma \quad (4.46)$$

$$K_I = \sum_{k=1}^{n_g} \hat{w}_k \hat{V}_I^T(x_{Qk})DB(x_{Qk})|J_q^D| - \sum_{k=1}^{n_{qt}} \hat{w}_k \hat{W}_I^T(x_{Qk})nDB(x_{Qk})|J_{qi}^B| - \sum_{k=1}^{n_{qt}} \hat{w}_k \hat{W}_I^T(x_{Qk})nDB(x_{Qk})|J_{qu}^B| \quad (4.47)$$

Applied traction and body forces are integrated in Eq. 4.48 and it is regulated with

Gauss quadrature for integration in Eq. 4.49.

$$f_I = \int_{\Gamma_{qt}} \hat{W}_I^T \bar{t} d\Gamma + \int_{\Omega_q} \hat{W}_I^T b d\Omega \quad (4.48)$$

$$f_I = \sum_{k=1}^{n_{gt}} \hat{w}_k \hat{W}_I^T(x_{Qk}) \bar{t} |J_{qt}^B| + \sum_{k=1}^{n_g} \hat{w}_k \hat{W}_I^T(x_{Qk}) b |J_q^D| \quad (4.49)$$

Similar stiffness and force matrices can be integrated in MLS shape functions, which are given in Eq. 4.50 and 4.51.

$$K_I = \int_{\Omega_q} \hat{V}_I^T DB d\Omega - \int_{\Gamma_{qi}} \hat{W}_I^T n DB d\Gamma - \int_{\Gamma_{qu}} \hat{W}_I^T n DB d\Gamma + \alpha \int_{\Gamma_{qu}} \hat{W}_I^T \Phi d\Gamma \quad (4.50)$$

$$f_I = \int_{\Gamma_{qt}} \hat{W}_I^T \bar{t} d\Gamma + \int_{\Omega_q} \hat{W}_I^T b d\Omega + \alpha \int_{\Gamma_{qu}} \hat{W}_I^T \bar{u} d\Gamma \quad (4.51)$$

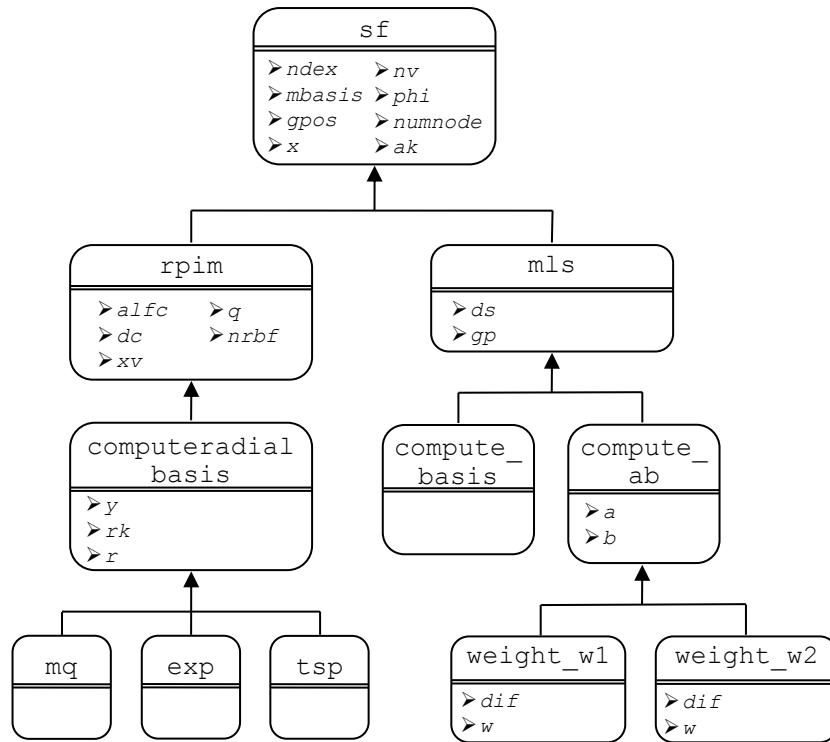


Figure 4.7. sf class hierarchy

```

classdef sf < handle
    properties (GetAccess = public, SetAccess = protected)
        ndex; mbasis; gpos; x; nv; phi; numnode
    end
end
classdef mls < sf
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Shape=mls(gpos,x,nv,ds,numnode,ndex,mbasis)...
            function Shape=mlscalcalc(Shape)...
    end
end
classdef compute_basis < mls
    methods
        function Comp_basis=compute_basis(gpos,x,nv,ds,numnode,ndex,mbasis)...
            function Comp_basis=compute_basispre(Comp_basis,gpos,mbasis)...
            function Comp_basis=compute_basiscalc(Comp_basis)...
    end
end
classdef compute_ab < mls
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Comp_ab=compute_ab(gpos,x,nv,ds,numnode,ndex,mbasis)...
            function Comp_ab=compute_abpre(Comp_ab,gpos,x,nv,ds,
                numnode,ndex,mbasis)...
            function Comp_ab=compute_abcalc(Comp_ab)...
    end
end
end

```

Figure 4.8. Matlab codes of `sf` and its inherited classes and member functions

```

classdef weight_w1 < compute_ab
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Weight1=weight_w1(gpos,x,nv,ds,numnode,ndex,mbasis)...
        function Weight1=weight_w1pre(Weight1,dif,nv,ds,ndex,numnode)...
        function Weight1=weight_w1calc(Weight1)...
    end
end
classdef weight_w2 < compute_ab
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Weight2=weight_w2(gpos,x,nv,ds,nx,numnode,ndex,mbasis)...
        function Weight2=weight_w2pre(Weight2,dif,nv,ds,nx,ndex,numnode)...
        function Weight2=weight_w2calc(Weight2)...
    end
end
classdef rpim < sf
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Shape=rpim(gpos,x,nv,numnode,ndex,alfc,dc,q,nrbf,mbasis)...
        function Shape=rpimcalc(Shape)...
    end
end
classdef computeradialbasis < rpim
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Rbasis=computeradialbasis(gpos,x,nv,numnode,ndex,
            alfc,dc,q,nrbf,mbasis)...
        function Rbasis=computeradialbasispre(Rbasis,x,y,xv,ndex,
            R,q,nrbf,mbasis)...
        function Rbasis=computeradialbasiscalc(Rbasis)...
    end
end
classdef mq < computeradialbasis
    methods
        function Mqbasis=mq(gpos,x,nv,numnode,ndex,alfc,dc,q,nrbf,mbasis)...
        function Mqbasis=mqrbasispre(Mqbasis,x,y,xv,ndex,R,q,nrbf,mbasis)...
        function Mqbasis=mqrbasiscalc(Mqbasis)...
    end
end
classdef expo < computeradialbasis
    methods
        function Expbasis=expo(gpos,x,nv,numnode,ndex,alfc,dc,q,nrbf,mbasis)...
        function Expbasis=expbasispre(Expbasis,x,y,xv,ndex,R,q,nrbf,mbasis)...
        function Expbasis=expbasiscalc(Expbasis)...
    end
end
classdef tsp < computeradialbasis
    methods
        function Tspbasis=tsp(gpos,x,nv,numnode,ndex,alfc,dc,q,nrbf,mbasis)...
        function Tspbasis=tspbasispre(Tspbasis,x,y,xv,ndex,R,q,nrbf,mbasis)...
        function Tspbasis=tspbasiscalc(Tspbasis)...
    end
end
end

```

Figure 4.8. Matlab codes of `sf` and its inherited classes and member functions (continue)

### 4.3.5 bc (Boundary Condition) Class

`bc` is defined for the essential and natural boundary conditions as shown in Fig. 4.9. It has a single property of `x`. It inherits this property to `integration_bcqt`, `integration_bcquqi` and `essentialbc` classes in Fig. 4.10. `integration_bcquqi` is mainly used integration of Eq. 4.52, which is inside of stiffness matrix terms in Eq. 4.46 and 4.50.



$$-\int_{\Gamma_{qi}} \hat{W}_I^T n DB d\Gamma - \int_{\Gamma_{qu}} \hat{W}_I^T n DB d\Gamma \quad (4.52)$$

integration\_bcqt class is used integration of Eq. 4.53, which represents integration of traction at the inside of Eq. 4.48 and 4.51.

$$\int_{\Gamma_{qi}} \hat{W}_I^T \bar{t} d\Gamma \quad (4.53)$$

In integration\_bcqt class, applied traction is transferred to the nodes and stored as nodal values in `f` property. Then, `f` is used to calculate force vector. integration\_bcquqi class is used in the calculation boundary integrals in a local quadrature domain. Essential boundary conditions are handled by using essentialbc class and all calculated values are recorded into `ak` property.

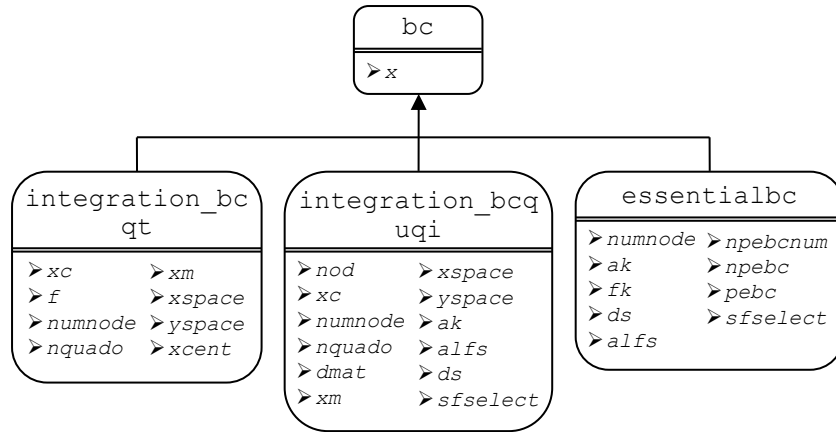


Figure 4.9. bc class hierarch

```

classdef bc < handle
    properties (GetAccess = public, SetAccess = protected)
        x
    end
end
classdef integration_bcqt < bc
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Integbcqt=integration_bcqt(xc,f,x,numnode,nquado,
            xm,xspace,yspace,xcent)...
        function Integbcqt=integration_bcqtcalc(Integbcqt)...
    end
end
classdef integration_bcquqi < bc
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Integbcquqi=integration_bcquqi(nod,xc,x,numnode,nquado,
            dmat,xm,xspace,yspace,ak,alfs,ds,sfselect)...
        function Integbcquqi=integration_bcquqicalc(Integbcquqi,Node)...
    end
end
classdef essentialbc < bc
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Essenbc=essentialbc(x,numnode,ak,fk,ds,alfs,
            npebcnum,npebc,pebc,sfselect)...
        function Essenbc=essentialbccalc(Essenbc,Node)...
    end
end
end

```

Figure 4.10. Matlab codes of bc, its inherited classes and member functions

**4.3.6 solv (Solution) Class**

The target in developing solv class is encapsulation of all classes that are related with solutions. Therefore, it has no property and it is the base class of dobmaxcalculator, solverband, sparsestorage and getinverse as shown in Fig. 4.11. gausseqsolver and bandsolver are subclasses of solverband. Its Matlab codes are partially given in Fig. 4.12. All these classes are generally used in the basic matrix operations and transfer their results to the class properties where they are called. bandsolver class is used solution of banded matrices and gausseqsolver class is used as alternatively usage of bandsolver for standard gauss elimination. matlabsolver class includes direct and different iterative methods for solution of linear equations ( $A*x=b$ ), which are available in Matlab library. Some iterative solvers are directly imported as class functions and ready for calling, which are Biconjugate gradients method (bicg), Biconjugate gradients stabilized method (bicgstab), Biconjugate gradients stabilized (l) method (bicgstabl), Conjugate gradients squared method (cgs), Generalized minimum residual method (gmres), LSQR method (lsqr), Minimum residual method (minres), Preconditioned conjugate gradients method (pcg), Quasi-minimal residual method (qmr), Symmetric LQ method (symmlq) and Transpose-free quasi-minimal

residual method (tfqmr). These method are used for solution of linear equations ( $A*x=b$ ) with iterations. Dimensions of matrices must be square ( $n \times n$ ) and symmetric for term of A in minres, pcg and symmlq solvers. Other methods can be used with asymmetric matrices. Only lsqr method can be capable to solve  $m \times n$  matrix dimensions of A. It is pointed (Liu and Gu, 2005) that stiffness matrices are generally sparse, banded and asymmetric. Hence all iterative solvers can be used compatible without minres, pcg and symmlq solvers.

Solution of complex geometries mostly causes to increase number of nodes in the solution. This phenomena cause to increase size of global stiffness matrix in meshfree methods. In most of the cases, global stiffness matrix holds zero values and its storage decrease computational effort. sparsestorage class is used for construction of sparse matrices, if it includes lots of zero terms. Sparse ratio function can calculate sparsity, which is the ratio of number of zero terms to all terms. Sparse matrices are also constructed with sparse function in Matlab alternatively. If only a linear equation is solved with small sizes, sparse function can be used without sparsestorage class.

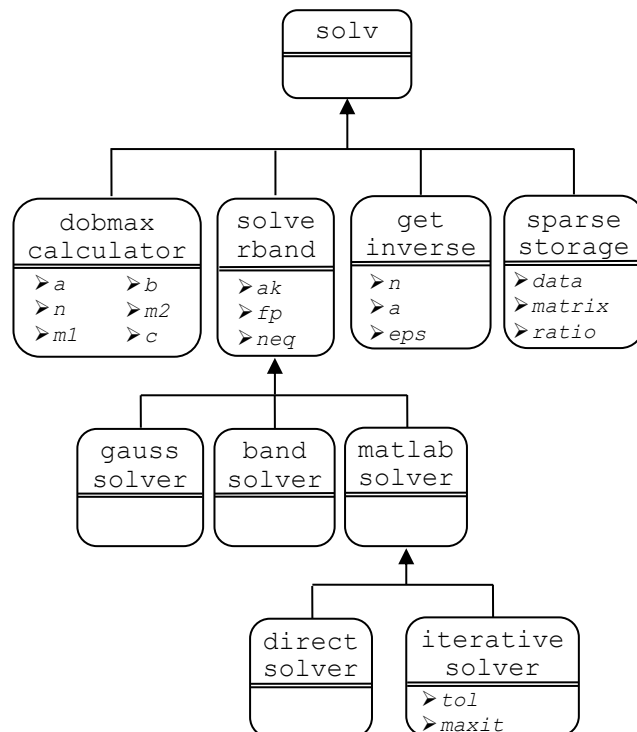


Figure 4.11. solv class hierarchy

```

classdef solv < handle
end
classdef dobmaxcalculator < solv
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Dobmax=dobmaxcalculator(A,N,M1,B,M2)...
            function Dobmax=dobmaxcalculatorcalc(Dobmax)...
        end
    end
end
classdef solverband < solv
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Sband=solverband(ak,fp,neq)...
            function Sband=solverbandcalc(Sband)...
        end
    end
end
classdef bandsolver < solv
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Bsolv=bandsolver(ak,fp,neq)...
            function Bsolv=bandsolvercalc(Bsolv)...
        end
    end
end
classdef gausssolver < solv
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Gaussolv=gaussolver(ak,fp,neq)...
            function Gaussolv=gaussolvercalc(Gaussolv)...
        end
    end
end
classdef getinverse < solv
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Getinv=getinverse(n,a,eps)...
            function Getinv=getinversecalc(Getinv)...
        end
    end
end
classdef sparsestorage < solv
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Sparsestr=sparsestorage(data,matrix,ratio)...
            function Sparsestr=sparseratio(Sparsestr)...
            function Sparsestr=sparsestoragecalc(Sparsestr)...
        end
    end
end
...

```

Figure 4.12. Matlab codes of solv, its inherited classes and member functions

**4.3.7 post (Results) Class**

All methods, which are related with results of solution, are encapsulated in a single class; post. It is shown in Fig. 4.13. Its Matlab codes are partially given in Fig. 4.14. It has three basic properties, x, numnode, u2. It inherits its properties to getdisplacement, getnodestress, output and getengerror classes. Displacements and stresses are calculated with respect to Eq. 4.54 and 4.55.

$$\epsilon_{(3 \times 1)} = B_{(3 \times 2n)} u_{(2n \times 1)} \tag{4.54}$$

$$\sigma_{(3 \times 1)} = D_{(3 \times 3)} \epsilon_{(3 \times 1)} = D_{(3 \times 3)} B_{(3 \times 2n)} u_{(2n \times 1)} \tag{4.55}$$

getdisplacement class computes the nodal displacement under action of given

loading condition and records them in `u2` and `disp` properties. The nodal stresses are computed in `getnodestress` class and recorded in `stress` property. In the output class, calculated displacements and stresses are written in a text file using its methods. The class also inherits all its properties and methods to `getengerror` that computes the energy error in the solution together with `totalgausspoints`.

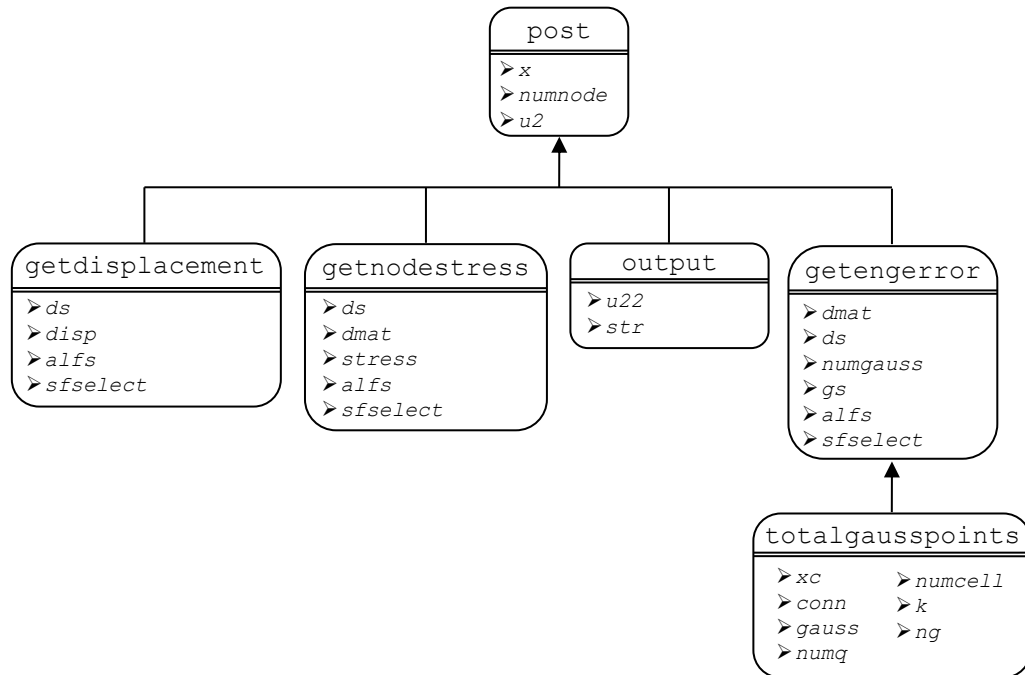


Figure 4.13. `post` class hierarch

```

classdef post
    properties (GetAccess = public, SetAccess = protected)
        x; numnode; u2
    end
end
classdef getdisplacement < post
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Getdisp=getdisplacement(x,ds,u2,alfs,numnode,sfselect)...
        function Getdisp=getdisplacementcalc(Getdisp,Node)...
    end
end
classdef getnodestress < post
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Getnstress=getnodestress(x,ds,Dmat,u2,alfs,numnode,sfselect)...
        function Getnstress=getnodestresscalc(Getnstress,Node)...
    end
end
classdef output < post
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Out=output(x,numnode,u2,u22,str)...
        function Out=outputcalc(Out)...
    end
end
classdef getengerror < post
    properties (GetAccess = public, SetAccess = protected)...
    methods
        function Geteng=getengerror(x,numnode,u2,dmat,ds,numgauss,gs,alfs,sfselect)...
        function Geteng=getengerrorcalc(Geteng,Node)...
    end
end
end ...

```

Figure 4.14. Matlab codes of `post`, its inherited classes and member functions

#### 4.4 Example Solution and Discussions

A cantilever beam with an end load (case I) and an axial loaded bar (case II) are solved by using LRPIM and MLPG methods in developed program. In these solutions, rectangular support domains and local rectangular integration cells are used. The constructed model has a length of 48 cm and a height of 12 cm. Unit thickness is used in the analyses. The used material properties have Young modulus of  $30.10^6$  N/cm<sup>2</sup> and Poisson's ratio of 0.0. Regularly distributed nodes have a nodal spacing of 4.8 cm in x (horizontal) and 3.00 cm in y (vertical) directions. Number of sub-division of quadrature domains is equal to 2 for both x and y directions. Number of gauss points is equal to 4 for each domain. 40 integration cells are used. Influence domain constant is selected as 3.00. Radial basis function parameters are selected as `nrbf` of 1.00, `alfc` of 1.00, `dc` of 3.00 and `q` of 1.03. The used number of basis functions is equal to 3. In both cases, the force ( $P$ ) is taken as 1000 N as shown in Fig. 4.15. The shape parameters are taken from the compared study (Liu and Gu, 2005) for getting same solution procedure. The number of regular distributed nodes is equal to 55 in all solutions.

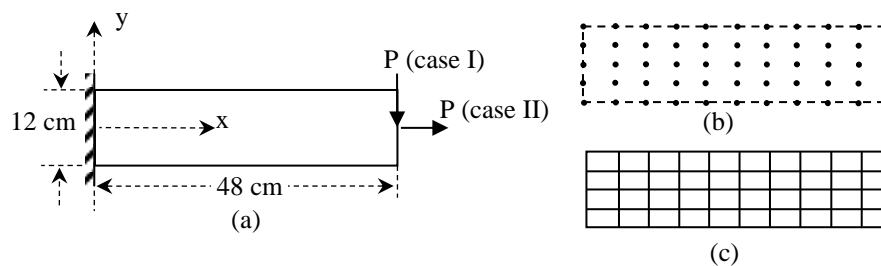


Figure 4.15. Engineering (a), meshfree (b) and FEM (c) models of axial loaded and cantilever beam problems and their boundary conditions

##### 4.4.1 Key Design Principles and Solution Procedure

A main framework is constructed to manage all objects and all base classes use the Matlab `handle` class. The overall scheme and interaction of classes can be seen in Fig. 4.16. `nodes` and `material` objects are used almost everywhere in the program. Therefore, they are created at the beginning of the solution. Construction of other classes is realized in sequence after these classes. All other classes are created by sending required data to their constructor functions. All computations are carried out inside of member functions and results are stored in objects as class properties. If

local variables are not defined as a property of that class, they are cleared at the end of the class function operations. Hence, it prevents unnecessary usage of memory. Only related and re-used from outside of class operators are assigned as class property.

Node coordinates, geometry and boundary conditions, meshfree parameters and other properties are read from files with `mc` class and stored in `nodes` object. Then `material` object is created for construction of material matrix with respect to plane stress or plane strain. When model operations in the classes are finished, calculation of quadrature domain of nodes begins with `inte` class. `qdomain` object stores the quadrature domain size and `gausscoefficient` object stores the gauss weights. In the solutions, local quadrature domain integrations are used. Therefore, `localquadraturedomain` object stores the quadrature domain boundaries and it is divided to subdomains. `nodes` object is called inside of this class and node coordinates are directly used without any conflicts. It is commonly used in procedural programming that transferring parameters with assigning new parameter names between programs and subroutines which causes to conflicts, especially inside of much code sizes. However, parameters are stored as properties of class objects and re-assignment is not required. `subdivisionlqdomain` object stores the divided local quadrature domain and `gauss node positions` are stored in `domaingausspoints` object.

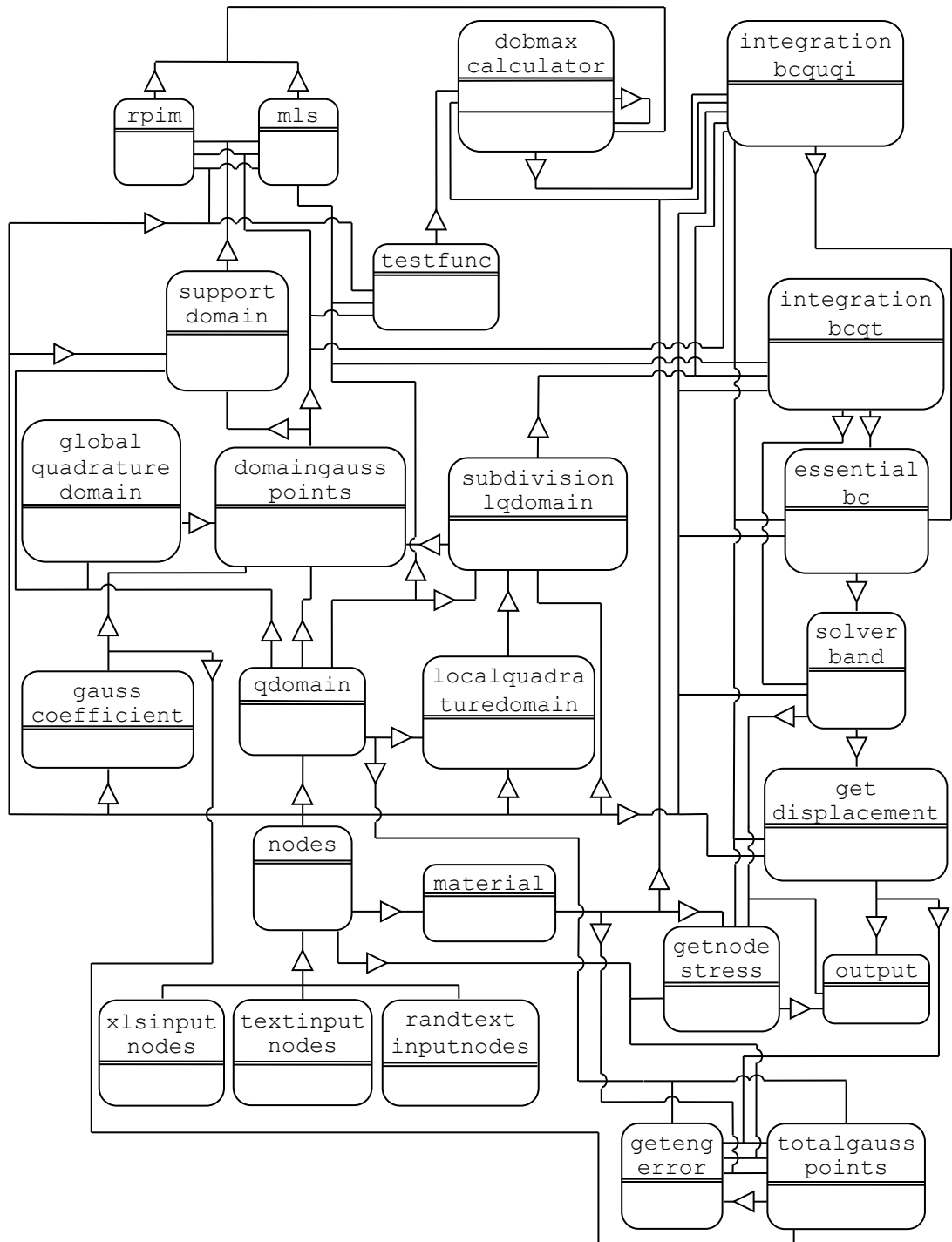


Figure 4.16. Overall flow and communications of classes during the solution for local quadrature domain

The type of support domain geometry is selected as rectangular. The number of nodes and node numbers in a support domain of a gauss point are stored in supportdomain object. Cubic spline test functions are used and their weights are stored inside of testfunc object. Next step is the calculation of shape functions. They are calculated and stored in rpim object. computeradialbasis class



includes the derivatives of radial basis functions and it is used by `rpim` object. 3 different basis function classes are available; `mq`, `exp` and `tsp`. `mq` class is used in this analysis. It can be also calculated by using `mls` object using MLPG method. If `mls` method is used, the calculated shape functions are stored in `mls` object. Inheritance property has large benefits on the developments of meshfree methods, especially similar structures and their sub-links are available. Superclass properties are directly transferred as properties of subclasses. Redefining of subclass properties is not required. Besides of that subclass includes its own special properties. Including new methods in shape function class can be easily handled by OOP, which can be defined as a subclass. Main properties like shape functions and their derivative values property `phi` directly assigns that new method.

OOP can be able to provide to work RPIM and MLS shape functions together for construction of shape functions. Both methods have their own benefits and accuracy in the calculations. One of them can be used in determined locations and the other can be used in other locations. This procedure can more easily accomplished by OOP. A switch is used to select the method for the construction of shape functions. `dobmaxcalculator` object is used for arrangements and calculations of shape functions. `integration_bcqt` object stores the applied force at related nodes. Also `integration_bcquqi` object stores the calculations of boundary integrals. All the calculations are repeated for each node and stored in the related class object. When the calculations finish, essential boundary conditions are applied and they are stored in `essentialbc` object.

Stiffness and force matrix includes lots of zero terms with respect to interpolation. `sparsestorage` class can be used for construction of sparse stiffness and force matrices. Also sparse function in Matlab can be directly used without using `sparsestorage` class. However, either using sparse function or using `sparsestorage` class consumes time for construction of sparse matrix. In this study, sparse function is directly used. Because solution model has few nodes and their properties are easily stored in memory. `sparsestorage` class also consumes time, especially %25 greater solution times in `solvr` class when comparing solution with sparse function in this solution. `solvrband` object is used for the solution of linear equations. After the solution, the nodal displacements are stored in

getdisplacement object. getnodestress object stores the nodal stresses, which are calculated with respect to nodal displacements. output object stores the output of the getdisplacement and getnodestress object for writing them to an external file. Energy error is calculated by using totalgausspoints and getengerror objects and stored inside of getengerror object.

#### 4.4.2 Solutions and Discussion

Analytical and FEM solutions are available in the examined cases. The deformation and axial stress formulations are given in Eq. 4.56 and 4.57 for axial loaded bar problem. A is cross-section area of the bar, E is Young's modulus of material, x is horizontal distance from one end of the bar and P represents applied force.

$$\delta = \frac{P \times x}{A \times E} \quad (4.56)$$

$$\sigma = \frac{P}{A} \quad (4.57)$$

For the cantilever beam problem, the displacements are plotted for the upper side where tensional stresses occur. The deflection and axial stresses of the cantilever beam are given in Eq. 4.58 and 4.59. I is moment of inertia, L is length of the beam and c is the distance between upper surface and natural axis of the beam.

$$y = \frac{P}{6 \times E \times I} (x^3 - 3 \times L \times x^2) \quad (4.58)$$

$$\sigma = \frac{M \times c}{I} \quad (4.59)$$

In ANSYS, same geometrical model is constructed by using PLANE182 elements. These elements have 4 nodes at the edges for rectangular type. Plane stress assumption is used. Similar boundary conditions are applied as analytical solutions.

The original procedural meshfree program (Liu and Gu, 2005) gives the same displacement results as OOP meshfree program as shown in Fig. 4.17 for cantilever beam and axial loaded bar problems. The analytical and ANSYS solutions are also given and it is seen that they are in good agreements. RPIM solutions are a little bit closer to analytical solutions than MLS results.

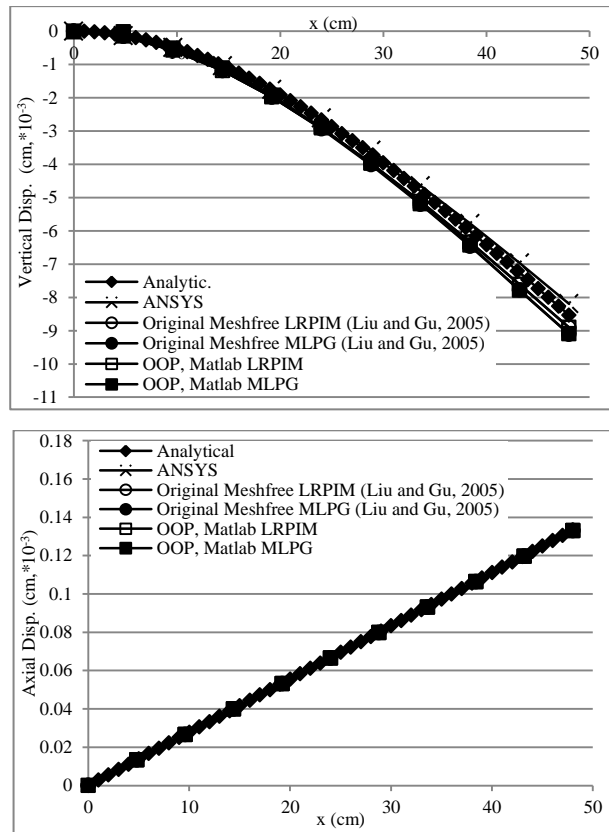


Figure 4.17. The displacement results of analytic, ANSYS, original meshfree (Liu and Gu, 2005) and OOP meshfree programs for cantilever beam problem (upper) and for axial loaded beam problem (lower)

Stress results are given in Fig. 4.18 for cantilever beam and axial loaded bar problems. It is seen that both original procedural meshfree program and OOP program nearly give the same results. All numerical results have good agreements with analytical solutions. A little difference can be seen in the results of RPIM method at the left corner, but same difference exists in the original procedural program. All stress results of bar problem are in perfect agreements with analytical and ANSYS results.

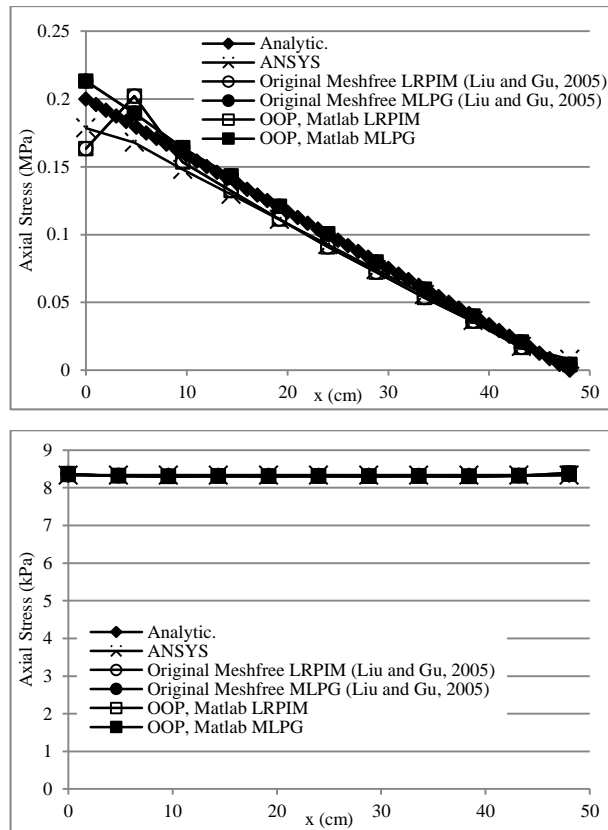


Figure 4.18. The stress results of analytic, ANSYS, original meshfree (Liu and Gu, 2005) and OOP meshfree programs for cantilever beam problem (upper), and for axially loaded bar problem (lower)

#### 4.5 Conclusion

Object-oriented programming of meshfree methods are discussed in detail for solution of elasto-static problems. Since new meshfree techniques are continuously improved by different researchers, the goal is developing a program that can be easily adapted to new meshfree techniques. Therefore, a program is mainly organised based on types of support domains, integrations and shape functions. It can handle a solution in different ways:

- using different integration cells; either global or local rectangular integration cells,
- using different support domains; rectangular, circular or triangular support domains,
- using different shape functions; either RPIM or MLS shape functions.
- different Matlab functions are included for increasing modularity at solution

A new technique, for example about shape functions, can be easily adapted by

creating its classes and objects without changing whole program structure. The source of a problem can be easily detected by only dealing with relevant classes.

In a procedural program, the program is developed using functions and subroutines. The parameters that are transferred to them must be carefully selected and general flow must be designed without causing a “spaghetti code” which is hard to understand and maintain. All variables in a procedural program are initially defined as public. However, in an object oriented program, the variables inside the classes are initially defined as private. The program is constructed based on objects and works by their interactions. Objects are instances of classes. Each class encapsulates its properties and methods. So, an encapsulated class can be modified and improved without considering the whole program structure.

The flexibility and modularity of OOP technique are used during developing of Meshfree code in Matlab. However, it was shown that Matlab OOP codes are slower against procedural Fortran codes. The reason of slowness of Matlab is based on slowness of it doing some certain operations such as loops (Berry et al., 2006; Eklund et al., 2012; Wang and Hopke, 2001). The results of Matlab OOP program, on the other hand, are in perfect agreement with results of Fortran procedural program.

## CHAPTER 5

### NODAL INTEGRATION WITH TAYLOR SERIES EXPANSION

#### 5.1 Introduction

Integration is an important phenomena and it is widely used in most of the mathematical applications of analysed systems. However, integration of problems is usually complex and very hard to solve. Hence numerical integration techniques are used to solve the complex conditions. These methods include some errors which must be checked for accuracy. Different researchers investigate and develop numerical integrations on different subjects. Lorenzini and Passoni (1999) develop numerical integration technique to decrease computational efficiency on the calculations of tropospheric chemistry. Monegato and Scuderi (1999) try to decrease the singularities on boundary integrals. Hadjifotinou (2002) uses numerical integration for the calculation of satellite orbits. A numerical integration (Ochiai, 2003) is investigated for meshless BEM method on the subject of no requirement of internal cells. Masserey et al. (2005) solve electromagnetic problems by decreasing the singularity of integration. Marczak (2006) uses object oriented programming technique to gain more efficient on numerical integration of FEM and BEM. Saritas and Filippou (2009) develop an integration technique for investigation of 3D plastic damage of concrete models. Also, unlike other integrations techniques, wavelet methods (Hashish et.al, 2009) are used for numerical integration. Leontyev (2010) develops a numerical integration technique on the solution of dynamic problems. Bettaieb et al. (2012) investigate elastoplastic strains of a single crystal by means of their developed numerical algorithm. Halilovic et al. (2009) develops an explicit numerical integration scheme for increasing efficiency on boundary value problems.

Gaussian integration is one of the integration schemes and widely used in the literature studies, especially in FEM. The series expansion technique of Gaussian integration gain lots of benefits. The integration is directly influenced by Gaussian weights. New developments are continuously carried on. Rathod et al. (2007a) study on a new gauss integration technique that includes two sections. One of the sections

provides new gauss weights to increase efficiency. Also Rathod et al. (2007b) study on symmetric gauss integration technique for setting more gauss integration sections. They achieve better results without using high order Gaussian integration rules. Islam and Hossain (2009) use a composite numerical integration on FEM and achieve solutions which are not gained analytically. Babuška et al. (2009) estimate an integration formula for increase the efficiency of the Gaussian integration on meshfree methods.

Taylor series expansion can be alternatively used on integrations. Liu et al. (2007) use Taylor series expansion on the solution of 2D elasto-static problems by using NI-RPIM. The applicability of this method can be developed, especially on 3D cases. Both 2D and 3D integration schemes are mentioned in this section. Basic equilibrium equations and main solution structure are given again for pointing how the integration scheme is used.

The approximated solution must be adapted to equilibrium equation with respect to applied boundary conditions. General form of equilibrium equation (Liu and Gu, 2005; Liu et al., 2007) and natural and essential boundary conditions are given in Eq. 5.1, 5.2 and 5.3, respectively.

$$L^T \times \sigma + b = 0 \quad (5.1)$$

$$\sigma \cdot n = \bar{t} \text{ on } \tau_i \quad (5.2)$$

$$u = \bar{u} \text{ on } \tau_u \quad (5.3)$$

$L^T$  is (Liu and Gu, 2005) differential operator,  $\sigma$  is the stress vector,  $u$  is the displacement vector,  $b$  is the body force vector,  $\bar{t}$  is prescribed traction on the natural boundaries,  $\bar{u}$  is prescribed displacement on the essential boundaries and  $n$  is the vector of unit outward normal on the natural boundary.

## 5.2 2D Nodal Integration Scheme

$L^T$ ,  $\sigma$ ,  $u$  and  $b$  are given in Eq. 5.4, 5.5, 5.6 and 5.7, respectively.

$$L^T = \begin{bmatrix} \partial/\partial x & 0 & \partial/\partial y \\ 0 & \partial/\partial y & \partial/\partial x \end{bmatrix} \quad (5.4)$$

$$\sigma^T = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \tau_{xy} \end{bmatrix} \quad (5.5)$$

$$u = \begin{bmatrix} u \\ v \end{bmatrix} \quad (5.6)$$

$$b = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \quad (5.7)$$

Equilibrium equation and natural and essential boundary conditions are represented with respect to weak form formulation in RPIM. Hence, the equilibrium equation, Eq. 5.1 can be defined as in Galerkin weak formulation in Eq. 5.8,

$$\int (L \times \delta u)^T \times (D \times L \times u) d\Omega - \int (\delta u^T \times b) d\Omega - \int (\delta u^T \times t) d\Gamma = 0 \quad (5.8)$$

$D$  matrix is material coefficient matrix and it is given for plane stress and plane strain in Eq. 5.9 and 5.10 as;

$$D = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)^2 \end{bmatrix} \quad (5.9)$$

$$D = \frac{E(1-\nu)}{(1-2\nu)(1+\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \quad (5.10)$$

where  $E$  is Young's modulus and  $\nu$  is Poisson's ratio (Liu and Gu, 2005). When substituting the approximated function in Eq. 3.49 or 4.33, into Eq. 5.8, general form of stiffness and force matrices are obtained.

$$K \times u = f \quad (5.11)$$

In Eq. 5.12, stiffness matrix formulation is represented. Force matrix formulation is given in Eq. 5.13. In addition, strain matrix, which includes derivatives of shape functions, is given in Eq. 5.14.

$$K_{ij} = \int B_i^T \times D \times B_j d\Omega \quad (5.12)$$



$$f_i = \int \varphi_i \times t d\Gamma + \int \varphi_i \times b d\Omega \quad (5.13)$$

$$B_i = \begin{bmatrix} \varphi_{i,x} & 0 \\ 0 & \varphi_{i,y} \\ \varphi_{i,y} & \varphi_{i,x} \end{bmatrix} \quad (5.14)$$

A suitable integration method is needed to solve these equations. Various integration techniques are available. In NI-RPIM (Liu et al., 2007), a series integration scheme is used with Taylor series. Series are widely used in mathematical operations, especially in numerical studies. An unknown value of a valid function can be estimated with a known value with series operations. One of the series is Taylor series and it is widely used in computational fluid dynamics with respect to finite difference method (FDM). An example of its value estimation from  $x_0$  to  $x_0 + h$  can be defined serial expansion of functions and it is given in Eq. 5.15.  $R_n$  is the total error between value of  $f(x_0 + h)$  and its Taylor expansion results. In general, the degree of used terms in FDM increases the accuracy. In Eq. 5.15, Taylor series are expanded and derived to  $n$  th order.

$$\begin{aligned} f(x_0 + h) = & f(x_0) + \frac{f'(x_0)}{1!} \times h + \frac{f''(x_0)}{2!} \times h^2 \\ & + \frac{f'''(x_0)}{3!} \times h^3 + \dots + \frac{f^{(n)}(x_0)}{n!} \times h^n + R_n \end{aligned} \quad (5.15)$$

Equations of integrations are expanded with respect to Taylor series expansion as Eq. 5.15. Stiffness matrix is set as approximate function for 2D cases as  $f(x, y)$ , which is given in Eq. 5.16,

$$\begin{aligned} f(x, y) = & f(x_0, y_0) + (x - x_0) \times \frac{\partial f(x_0, y_0)}{\partial x} + (y - y_0) \times \frac{\partial f(x_0, y_0)}{\partial y} + \\ & + \frac{1}{2!} \times (x - x_0)^2 \times \frac{\partial^2 f(x_0, y_0)}{\partial x^2} + \frac{1}{2!} \times 2 \times (x - x_0) \times (y - y_0) \\ & \times \frac{\partial^2 f(x_0, y_0)}{\partial x \partial y} + \frac{1}{2!} \times (y - y_0)^2 \times \frac{\partial^2 f(x_0, y_0)}{\partial y^2} + \frac{1}{3!} \times (x - x_0)^3 \\ & \times \frac{\partial^3 f(x_0, y_0)}{\partial x^3} + \frac{1}{3!} \times (y - y_0)^3 \times \frac{\partial^3 f(x_0, y_0)}{\partial y^3} + \dots \end{aligned} \quad (5.16)$$

The nodal integration of Eq. 5.16 can be written as Eq. 5.17,

$$\begin{aligned}
\int f(x, y)d\Omega = & \int (f(x_0, y_0) + (x - x_0) \times \frac{\partial f(x_0, y_0)}{\partial x} + (y - y_0) \times \frac{\partial f(x_0, y_0)}{\partial y} \\
& + \frac{1}{2!} \times (x - x_0)^2 \times \frac{\partial^2 f(x_0, y_0)}{\partial x^2} + \frac{1}{2!} \times 2 \times (x - x_0) \times (y - y_0) \\
& \times \frac{\partial^2 f(x_0, y_0)}{\partial x \partial y} + \frac{1}{2!} \times (y - y_0)^2 \times \frac{\partial^2 f(x_0, y_0)}{\partial y^2} + \quad (5.17) \\
& + \frac{1}{3!} \times (x - x_0)^3 \times \frac{\partial^3 f(x_0, y_0)}{\partial x^3} + \frac{1}{3!} \times (y - y_0)^3 \times \frac{\partial^3 f(x_0, y_0)}{\partial y^3} \\
& + \frac{1}{3!} \times 3 \times (x - x_0)^2 \times (y - y_0) \times \frac{\partial^3 f(x_0, y_0)}{\partial x^2 \partial y} + \dots)d\Omega
\end{aligned}$$

If Eq. 5.18 is separated and arranged, the following form can be obtained,

$$\begin{aligned}
\int f(x, y)d\Omega = & \int f(x_0, y_0)d\Omega + \int (x - x_0) \times \frac{\partial f(x_0, y_0)}{\partial x} d\Omega \\
& + \int (y - y_0) \times \frac{\partial f(x_0, y_0)}{\partial y} d\Omega + \int \frac{1}{2!} \times (x - x_0)^2 \\
& \times \frac{\partial^2 f(x_0, y_0)}{\partial x^2} d\Omega + \int \frac{1}{2!} \times 2 \times (x - x_0) \times (y - y_0) \quad (5.18) \\
& \times \frac{\partial^2 f(x_0, y_0)}{\partial x \partial y} d\Omega + \int \frac{1}{2!} \times (y - y_0)^2 \times \frac{\partial^2 f(x_0, y_0)}{\partial y^2} d\Omega \\
& + \int \frac{1}{3!} \times (x - x_0)^3 \times \frac{\partial^3 f(x_0, y_0)}{\partial x^3} d\Omega + \dots
\end{aligned}$$

$$\begin{aligned}
\int f(x, y)d\Omega = & f(x_0, y_0) \times \int 1d\Omega + f_x(x_0, y_0) \times \int (x - x_0)d\Omega \\
& + f_y(x_0, y_0) \times \int (y - y_0)d\Omega + f_{xx}(x_0, y_0) \times \frac{1}{2!} \times \int (x - x_0)^2 d\Omega \quad (5.19) \\
& + f_{yy}(x_0, y_0) \times \int (y - y_0)^2 d\Omega + \dots
\end{aligned}$$

In Eq. 5.19,  $x - x_0$  represents  $\Delta x$  and  $y - y_0$  represents  $\Delta y$ . The distance is calculated from midpoint of related field node and integration cell, which is called Taylor integration cells.  $d\Omega$  is equal to area of Taylor integration cell. Each Taylor integration cell for each field node includes an area that does not interact with other field nodes of Taylor integration cells.

### 5.3 3D Nodal Integration Scheme

The mentioned integration rule in chapter 5.2 is expanded for 3D cases in this section. The given equations of  $L^T$ ,  $\sigma$ ,  $u$  and  $b$  for 2D cases in Eq. 5.4, 5.5, 5.6 and 5.7 are expanded for 3D cases in Eq. 5.20, 5.21, 5.22 and 5.23, respectively.

$$L^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \quad (5.20)$$

$$\sigma^T = [\sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \tau_{yz} \quad \tau_{xz} \quad \tau_{xy}] \quad (5.21)$$

$$u = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (5.22)$$

$$b = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (5.23)$$

Same Galerkin weak formulation of equilibrium equation is valid in 3D cases, which is given in Eq. 5.8.  $D$  matrix is material coefficient matrix and it is given for isotropic solids as;

$$D = \frac{E \times (1-\nu)}{(1+\nu) \times (1-2\nu)} \times \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2 \times (1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2 \times (1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2 \times (1-\nu)} \end{bmatrix} \quad (5.24)$$

$B$  matrix of 3D cases includes differences from 2D cases with respect to its sizes, which given in below.

$$B_i = \begin{bmatrix} \varphi_{i,x} & 0 & 0 \\ 0 & \varphi_{i,y} & 0 \\ 0 & 0 & \varphi_{i,z} \\ 0 & \varphi_{i,z} & \varphi_{i,y} \\ \varphi_{i,z} & 0 & \varphi_{i,x} \\ \varphi_{i,y} & \varphi_{i,x} & 0 \end{bmatrix} \quad (5.25)$$

Taylor series are also used in 3D cases. Stiffness matrix is set as approximate function as  $f(x, y, z)$ , which is given in Eq. 5.26,

$$\begin{aligned}
f(x, y, z) = & f(x_0, y_0, z_0) + (x - x_0) \times \frac{\partial f(x_0, y_0, z_0)}{\partial x} + (y - y_0) \times \\
& \frac{\partial f(x_0, y_0, z_0)}{\partial y} + (z - z_0) \times \frac{\partial f(x_0, y_0, z_0)}{\partial z} + \frac{1}{2!} \times (x - x_0)^2 \times \\
& \frac{\partial^2 f(x_0, y_0, z_0)}{\partial x^2} + \frac{1}{2!} \times 2 \times (x - x_0) \times (y - y_0) \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial x \partial y} + \\
& \frac{1}{2!} \times (y - y_0)^2 \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial y^2} + \frac{1}{2!} \times 2 \times (y - y_0) \times (z - z_0) \times \\
& \frac{\partial^2 f(x_0, y_0, z_0)}{\partial y \partial z} + \frac{1}{2!} \times (z - z_0)^2 \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial z^2} + \frac{1}{2!} \times 2 \times \\
& (z - z_0) \times (x - x_0) \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial z \partial x} + \frac{1}{3!} \times (x - x_0)^3 \times \\
& \frac{\partial^3 f(x_0, y_0, z_0)}{\partial x^3} + \frac{1}{3!} \times (y - y_0)^3 \times \frac{\partial^3 f(x_0, y_0, z_0)}{\partial y^3} + \frac{1}{3!} \times (z - z_0)^3 \times \\
& \frac{\partial^3 f(x_0, y_0, z_0)}{\partial z^3} + \frac{1}{3!} \times 3 \times (x - x_0)^2 \times (y - y_0) \times \frac{\partial^3 f(x_0, y_0, z_0)}{\partial x^2 \partial y} + \dots \quad (5.26)
\end{aligned}$$

The nodal integration of Eq. (5.26) can be written as Eq. (5.27),

$$\begin{aligned}
\int f(x, y, z) d\Omega = & \int \left( f(x_0, y_0, z_0) + (x - x_0) \times \frac{\partial f(x_0, y_0, z_0)}{\partial x} + (y - y_0) \times \right. \\
& \frac{\partial f(x_0, y_0, z_0)}{\partial y} + (z - z_0) \times \frac{\partial f(x_0, y_0, z_0)}{\partial z} + \frac{1}{2!} \times (x - x_0)^2 \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial x^2} + \\
& \frac{1}{2!} \times 2 \times (x - x_0) \times (y - y_0) \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial x \partial y} + \frac{1}{2!} \times (y - y_0)^2 \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial y^2} + \\
& \frac{1}{2!} \times 2 \times (y - y_0) \times (z - z_0) \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial y \partial z} + \frac{1}{2!} \times (z - z_0)^2 \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial z^2} + \quad (5.27) \\
& \frac{1}{2!} \times 2 \times (z - z_0) \times (x - x_0) \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial z \partial x} + \frac{1}{3!} \times (x - x_0)^3 \times \frac{\partial^3 f(x_0, y_0, z_0)}{\partial x^3} + \\
& \frac{1}{3!} \times (y - y_0)^3 \times \frac{\partial^3 f(x_0, y_0, z_0)}{\partial y^3} + \frac{1}{3!} \times (z - z_0)^3 \times \frac{\partial^3 f(x_0, y_0, z_0)}{\partial z^3} + \frac{1}{3!} \times 3 \times \\
& \left. (x - x_0)^2 \times (y - y_0) \times \frac{\partial^3 f(x_0, y_0, z_0)}{\partial x^2 \partial y} + \dots \right) d\Omega
\end{aligned}$$

If Eq. (5.27) is separated and arranged, the following form can be obtained,

$$\begin{aligned}
\int f(x, y, z)d\Omega &= \int f(x_0, y_0, z_0)d\Omega + \int (x - x_0) \times \frac{\partial f(x_0, y_0, z_0)}{\partial x} d\Omega + \int (y - y_0) \times \\
&\frac{\partial f(x_0, y_0, z_0)}{\partial y} d\Omega + \int (z - z_0) \times \frac{\partial f(x_0, y_0, z_0)}{\partial z} d\Omega + \int \frac{1}{2!} \times (x - x_0)^2 \times \\
&\frac{\partial^2 f(x_0, y_0, z_0)}{\partial x^2} d\Omega + \int \frac{1}{2!} \times 2 \times (x - x_0) \times (y - y_0) \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial x \partial y} d\Omega + \\
&\int \frac{1}{2!} \times (y - y_0)^2 \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial y^2} d\Omega + \int \frac{1}{2!} \times 2 \times (y - y_0) \times (z - z_0) \times \\
&\frac{\partial^2 f(x_0, y_0, z_0)}{\partial y \partial z} d\Omega + \int \frac{1}{2!} \times (z - z_0)^2 \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial z^2} d\Omega + \int \frac{1}{2!} \times 2 \times \\
&(z - z_0) \times (x - x_0) \times \frac{\partial^2 f(x_0, y_0, z_0)}{\partial z \partial x} d\Omega + \int \frac{1}{3!} \times (x - x_0)^3 \times \frac{\partial^3 f(x_0, y_0, z_0)}{\partial x^3} \\
&d\Omega + \dots
\end{aligned} \tag{5.28}$$

$$\begin{aligned}
\int f(x, y, z)d\Omega &= f(x_0, y_0, z_0) \times \int 1d\Omega + f_x(x_0, y_0, z_0) \times \int (x - x_0)d\Omega + \\
&f_y(x_0, y_0, z_0) \times \int (y - y_0)d\Omega + f_z(x_0, y_0, z_0) \times \int (z - z_0)d\Omega + \\
&f_{xx}(x_0, y_0, z_0) \times \frac{1}{2!} \times \int (x - x_0)^2 d\Omega + f_{xy}(x_0, y_0, z_0) \times \\
&\int (x - x_0) \times (y - y_0)d\Omega + \dots
\end{aligned} \tag{5.29}$$

In Eq. 5.29,  $x - x_0$  represents  $\Delta x$ ,  $y - y_0$  represents  $\Delta y$  and  $z - z_0$  represents  $\Delta z$ .  $d\Omega$  is equal to volume of Taylor integration cell.

## CHAPTER 6

### INTEGRATIONS WITH VARIOUS TYPES OF CELLS

#### 6.1 Introduction

It is required (Liu et al., 2007) to construction of integration domains for each node in the NI-RPIM of analysed model. These domains should not coincide with each other and must be fully integrated. Taylor integration domains are constructed with irregular triangular (2D cases) and tetrahedral (3D cases) geometries for irregular distributed nodes. These irregular domain construction supports to more flexible domains, rather than hexahedral cells. Delaunay triangulation method is used for creating triangular or tetrahedral cells between nodes by using Matlab.

#### 6.2 Rectangular Shaped Taylor Integration Cells for 2D Cases

Various integration domains are available for integration. Rectangular integration domains are one of them and widely used especially regular distributed nodes in the analysed domains. General form of integrations for 2D domains is mentioned in the previous chapter of 5.2 and the integration equation is given in 5.19 for 2D domains. The represented integrals are taken on the constructed domains for field node.

##### 6.2.1 Numerical Examples

Three different case studies are examined for application of rectangular shaped Taylor integration cells. Axial loaded, cantilever and simply supported beam with distributed loading problems have analytical solutions. Numerical techniques like FEM can be applied on these types of problems. Also nodal integration by using Taylor series expansion can be used with similar conditions. Increasing the orders of Taylor series expansions terms on these types of problems are examined in these cases.

Three different loading conditions are formed on a beam. Beam models have same geometrical and material properties. The only changed parameters are applied boundary conditions. Beam is modelled unit thickness and in the calculations, plane

stress assumption is used. Young modulus of 200 GPa and Poisson's ratio of 0.0 are used in material properties. The selection of Poisson's ratio as 0.0 is mainly used for providing the similar condition as analytical solutions. The behaviour of material is assumed fully elastic.

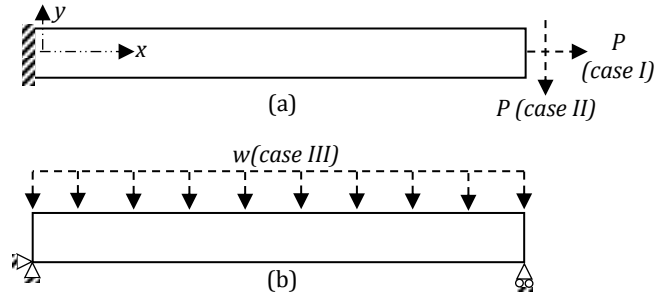


Figure 6.1. The models are used in analytical solution for; a) axial loaded beam (case I), cantilever beam (case II), b) simply supported beam problem with distributed loaded (case III)

Beam models are given for analytical solution in Fig. 6.1. The beam has a length of 1.00 m and a height of 0.1 m.  $P$  is applied as 1000 N and  $w$  is applied as 1000 N/m. In axial loaded beam, deformation and normal stress have more dominant effect on mechanical robustness. Results are taken at the natural axis of the beam in  $x$ -direction. The deformation and  $x$ -stress (axial stress) are given in Eq. 6.1 and 6.2, respectively for analytical solution.

$$\delta = \frac{P \times x}{A \times E} \quad (6.1)$$

$$\sigma = \frac{P}{A} \quad (6.2)$$

$P$  represents applied force,  $A$  is area,  $E$  is Young's modulus in the notifications.  $x$ -stress is also dominant effect on cantilever and simply supported beam problem. However, they occur at the maximum values on the surfaces. Hence the  $x$ -stress results are taken on the upper surface of the beams. They are given in Eq. 6.3 for cantilever and simply supported beam models.  $c$  represents the distance between natural axis and upper surface of the beam.

$$\sigma = \frac{M \times c}{I} \quad (6.3)$$

On the contrary to check  $x$ -direction deformation,  $y$ -direction deflection is considered in cantilever and simply supported beams. Deflection is given in Eq. 6.4

for cantilever beam problem.  $I$  is inertia of the beam. Deflection of distributed loaded simply supported beam model is given in Eq. 6.5, where  $w$  is applied distributed force.

$$y = \frac{P}{6 \times E \times I} (x^3 - 3 \times L \times x^2) \quad (6.4)$$

$$y = -\frac{w}{24 \times E \times I} \times (x^4 - 2 \times L \times x^3 - L^3 \times x) \quad (6.5)$$

Numerical models of FEM are given in Fig. 6.2 for 10 elements (a) and 20 elements (b). 10 element model have 18 nodes and 20 element model have 33 nodes. Same boundary conditions as analytical models are applied on FEM models. Plane182 element, which has 4 nodes for rectangular shape, is used in ANSYS.

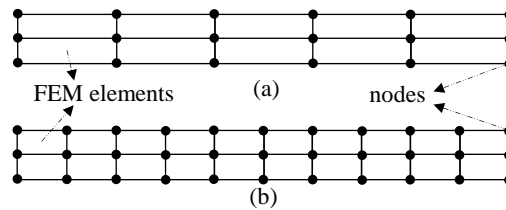


Figure 6.2. The models are used in FEM solutions; a) 10 elements model, b) 20 elements model

In Fig. 6.3, NI-RPIM models are given for 18 nodes and 33 nodes. These NI-RPIM models are both used for Gaussian and Taylor integration. In 18 nodes model, a nodal interval of 0.05 m in height and a nodal interval of 0.2 m in length are used. In 33 nodes model, a nodal interval of 0.05 m in height and a nodal interval of 0.1 m in length are used. The used Taylor series expansion terms are up to 4<sup>th</sup> and if Gaussian integration is used, 2\*2 gauss points are available in each background cell.

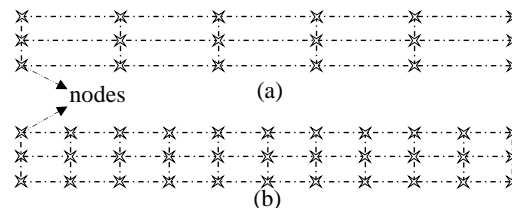


Figure 6.3. The models are used in NI-RPIM solutions for 2D shaped Taylor integration cells; a) 18 nodes model, b) 33 nodes model

### 6.2.1.1 Axial Loaded Bar

In Fig. 6.4, analytical, FEM (10 elements), RPIM (18 nodes) with Gaussian integration and Taylor integration (2<sup>nd</sup> order NI-RPIM) results are given for axial



loaded beam problem. Whole compared numerical results have good agreements with analytical solution. However, results of NI-RPIM (2<sup>nd</sup> order terms) have larger deformation than analytical results at the force application location of the beam. But, general characteristics of compared results behave similar and the difference can be accepted. Further increasing the number of nodes to 33, the results of NI-RPIM (2<sup>nd</sup> order terms) get better and are drawn to analytical results, especially the force applied location in Fig. 6.5 FEM and Gaussian int. of PRIM results have good agreement with analytical results.

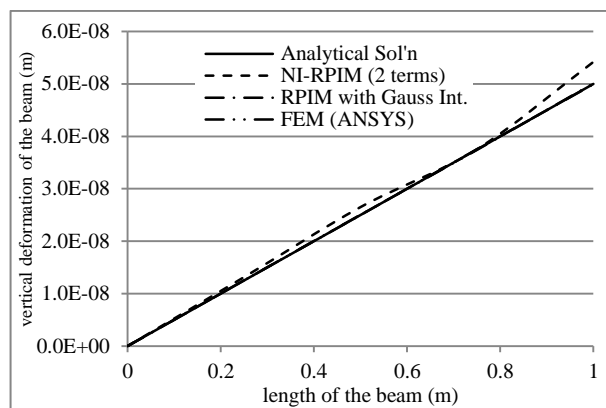


Figure 6.4. Comparison of different solution techniques on vertical deformation of 18 nodes 2D axial loaded beam

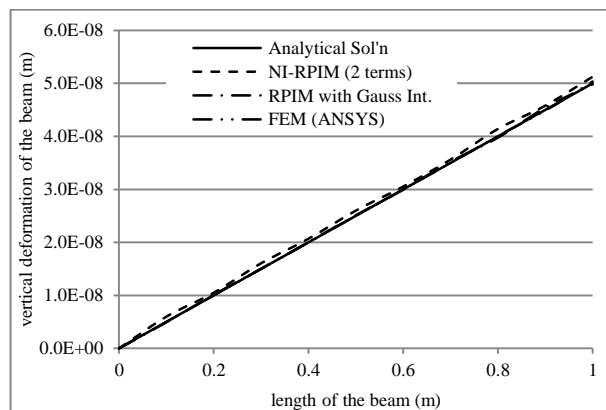


Figure 6.5. Comparison of different solution techniques on vertical deformation of 33 node 2D axial loaded beam

The effect of used terms in Taylor Series expansion is given in Fig. 6.6 for 18 nodes and they are compared with only analytical results. The usage of 1<sup>st</sup> order term in Taylor series expansion in NI-RPIM causes to a fluctuation at the vertical deformation results of the beam. The fluctuation disappears and the vertical deformation results are improved with increasing the used order of Taylor series expansion terms. However, the vertical deformations of tip locations of beam are same in all used terms of NI-RPIM scheme.

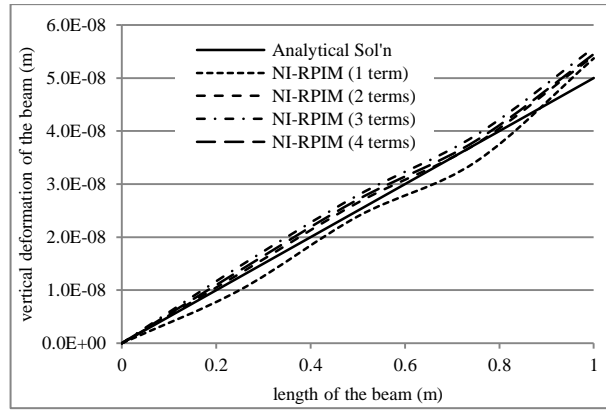


Figure 6.6. Effect of Taylor series expansion terms on vertical deformation of 18 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells

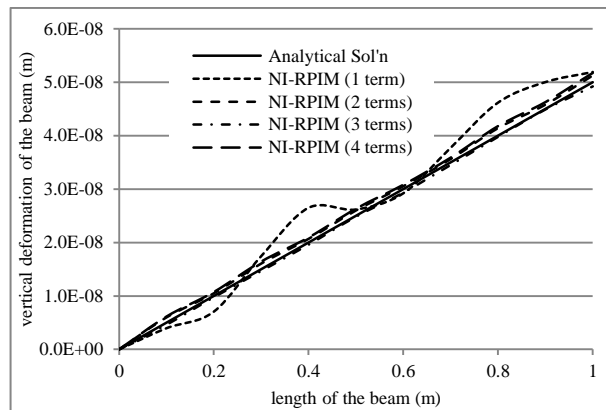


Figure 6.7. Effect of Taylor series expansion terms on vertical deformation of 33 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells

The increased node results in NI-RPIM are given in Fig. 6.7. It has seen that 2<sup>nd</sup> and greater terms of NI-RPIM results have good agreements with compared analytical solution. However, the used 1<sup>st</sup> term in NI-RPIM gives greater fluctuation results than 1<sup>st</sup> term of 15 node NI-RPIM results. Besides of that vertical deformation of tip of the beam approximately overlaps with analytical result.

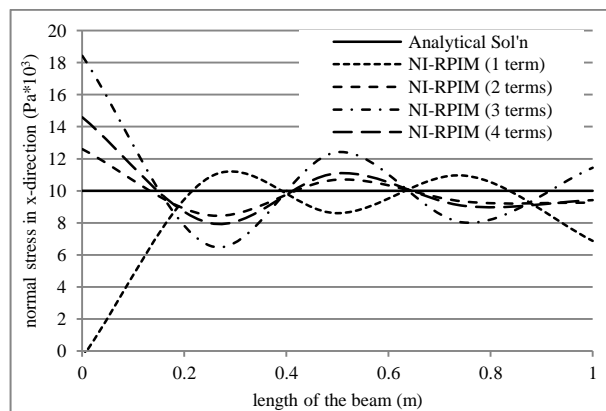


Figure 6.8. Effect of Taylor series expansion terms on axial stress of 18 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells

The effect of used terms can be seen more easily in stress results. Normal stress results in x-direction are given for 18 nodes in Fig. 6.8. NI-RPIM stress results at the support location of the beam have a great difference from analytical solution. This location influences the further sections results of the beam. But the fluctuation decreases at the far sections of support location. This local high stresses may be related with Saint Venant's principle. The results are improved with increasing the order of Taylor series expansion terms. In Fig. 6.9, increasing the number of nodes provide more suitable and better normal stress results. 4<sup>th</sup> order terms of NI-RPIM results are the best results and have the least fluctuation. Besides of that, using only 1<sup>st</sup> term in NI-RPIM gives the worst stress results.

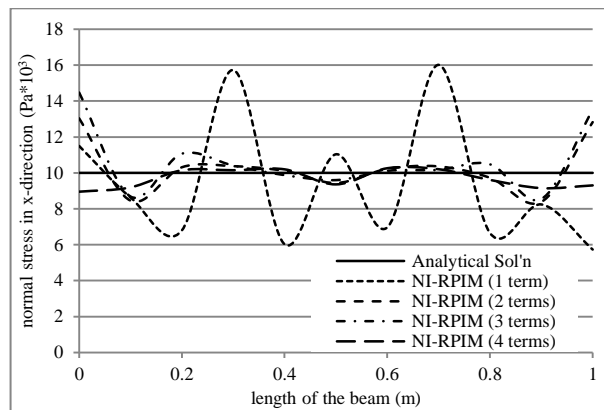


Figure 6.9. Effect of Taylor series expansion terms on axial stress of 33 nodes axial loaded beam with 2D rectangular shaped Taylor integration cells

Table 6.1. Comparison of solution times of RPIM with nodal and Gaussian integration techniques for axial loaded beam models of 33 and 1111 nodes

RPIM with nodal integration		Sol'n time (in Secs)	Sol'n time (in Secs)	RPIM with Gauss integration	
33 nodes	1 term	0.442	0.187	2*2	33 nodes
	2 terms	0.474	0.249	3*3	
	3 terms	0.528	0.39	4*4	
	4 terms	0.73			
1111 nodes	1 term	22.58	22.259	2*2	1111 nodes
	2 terms	24.064	42.557	3*3	
	3 terms	27.033	70.98	4*4	
	4 terms	31.944			

Solution time and computational performance are also important subjects for a numerical analysis. These properties can affect the usage of numerical analysis, if their requirements in the solution need high solution time and cause high computational cost. Hence, solution times of RPIM with nodal and Gaussian integration are investigated and their results are given in table 6.1. A computer has a

memory of 3GB and a processor speed of 2.4 GHz, is used in the analysis. 2 different numbers of nodes (33 and 1111) are used for observing the solution times. It is seen that in the results of 33 nodes, Gaussian integration works faster than nodal integration, even if 4\*4 number of Gauss points are used in the each background cell. But all solution cases are completed in less than one second.

However, when increasing the used number of nodes to 1111, the results of 1<sup>st</sup> order term of nodal integration and 2\*2 gauss points of Gaussian integration have similar solution times. When increasing the number of gauss points (3\*3 and 4\*4), the solution time increases rapidly rather than the increment of terms of nodal integration. The increment of order of Taylor series expansion in nodal integration is not dominant as the used number of gauss points in Gaussian integration for solution time. Nearly, the solution of RPIM with 4<sup>th</sup> order terms of nodal integration needs half of solution time of Gaussian integration (4\*4 gauss point).

### 6.2.1.2 Cantilever Beam

The deflection results of cantilever beam for different methods are given in Fig. 6.10. The compared methods give sufficient results with respect to analytical solution. However, there is a small gap between analytical and NI-RPIM results, which have 2<sup>nd</sup> order Taylor series expansion terms. This gap between the compared results is eliminated with increasing the number of nodes in Fig. 6.11. Nearly whole of the compared methods have perfectly agreement with analytical solution.

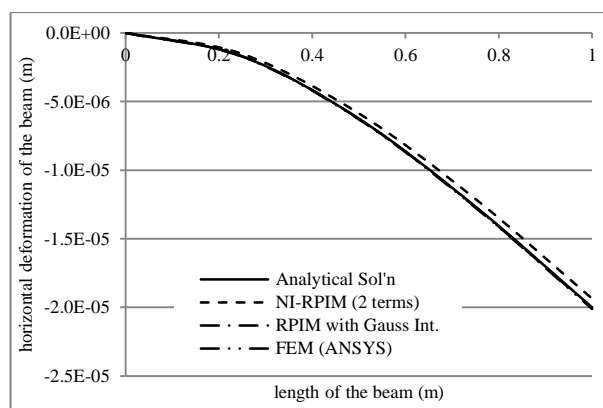


Figure 6.10. Comparison of different solution techniques on vertical deformation of 18 nodes 2D cantilever beam

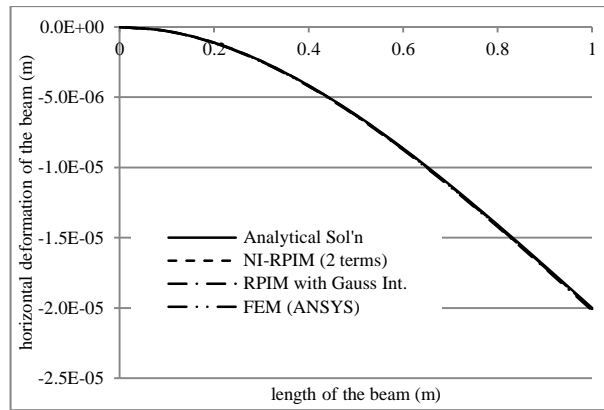


Figure 6.11. Comparison of different solution techniques on vertical deformation of 33 nodes 2D cantilever beam

The effect of used terms in horizontal deformation of the beam is given in Fig. 6.12. 1<sup>st</sup> order term of NI-RPIM gives insufficient results. When increasing the order of terms, the results become close enough to analytical results.

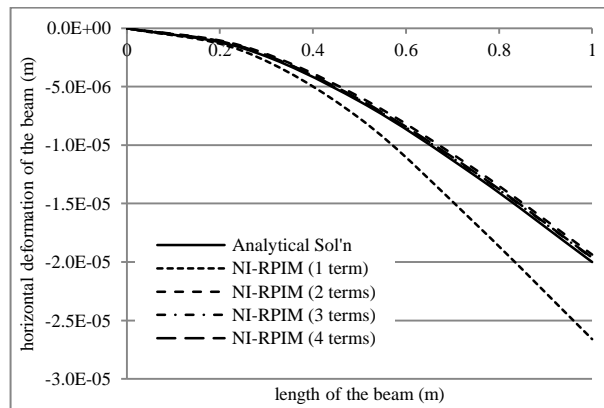


Figure 6.12. Effect of Taylor series expansion terms on vertical deformation of 18 nodes cantilever beam

But increasing the number of nodes to 33, the results are approximately same as analytical solution without 1<sup>st</sup> order results in Fig. 6.13.

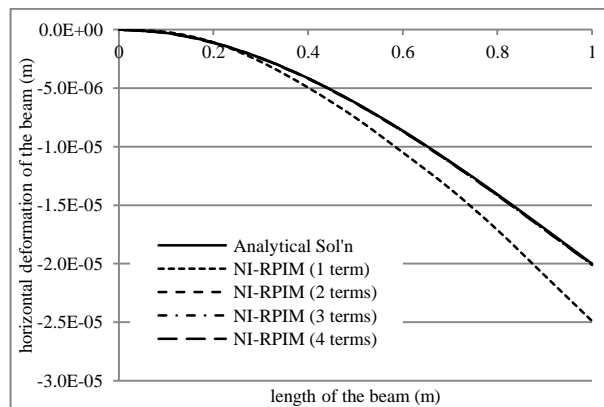


Figure 6.13. Effect of Taylor series expansion terms on vertical deformation of 33 nodes cantilever beam with 2D rectangular shaped Taylor integration cells

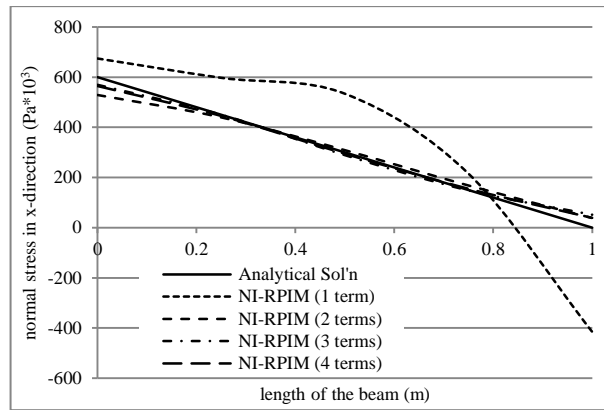


Figure 6.14. Effect of Taylor series expansion terms on axial stress of 18 nodes cantilever beam with 2D rectangular shaped Taylor integration cells

The stress results of cantilever beam for 18 nodes are given in Fig. 6.14. The results of 1<sup>st</sup> order term give insufficient results. However increasing the order of NI-RPIM terms gives similar results as analytical solution. A high distortion on the results of 1<sup>st</sup> order terms is seen in Fig. 6.15 and causes to not behave similar behaviour as analytical results. But 2<sup>nd</sup> and greater order of NI-RPIM terms nearly gives the same results as analytical solution.

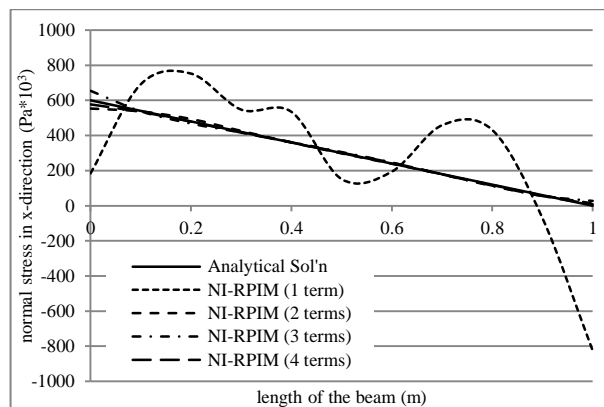


Figure 6.15. Effect of Taylor series expansion terms on axial stress of 33 node cantilever beam with 2D rectangular shaped Taylor integration cells

### 6.2.1.3 Simply Supported Beam

The horizontal deformation results of simply supported beam are given for 18 nodes in Fig. 6.16. FEM results are the closest results to analytical solution. Besides, the results of 2<sup>nd</sup> order terms in NI-RPIM are the farthest results, general behaviour of the results has similar characteristics with analytical solution. When increasing the number of nodes, nearly all the compared methods overlap with the analytical solution in Fig. 6.17.

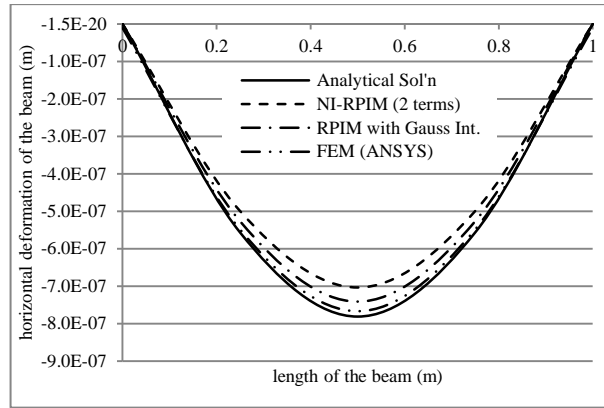


Figure 6.16. Comparison of different solution techniques on vertical deformation of 18 nodes 2D simply supported beam

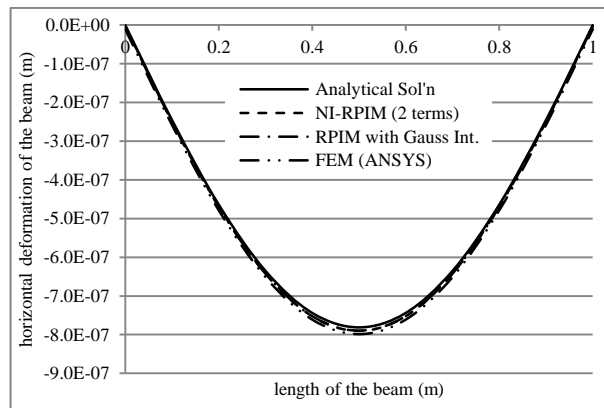


Figure 6.17. Comparison of different solution techniques on vertical deformation of 33 nodes 2D simply supported beam

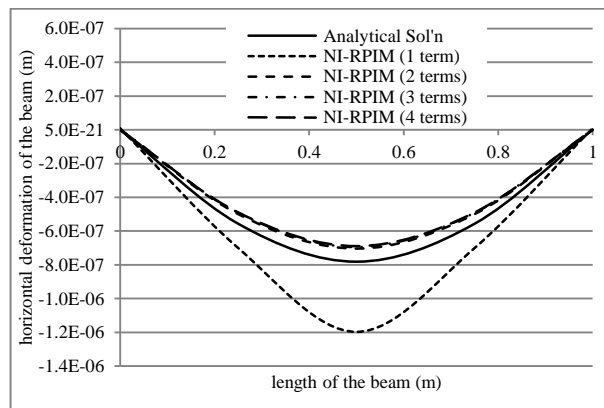


Figure 6.18. Effect of Taylor series expansion terms on vertical deformation of 18 nodes simply supported beam with 2D rectangular shaped Taylor integration cells

The effect of terms of NI-RPIM is given in Fig. 6.18 for 18 node results. It is seen that 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> order terms give similar and less deformation values than analytical solution. But 1<sup>st</sup> order term results have great difference from analytical solution and greater deformation values than analytical solution. When increasing number of nodes to 33, the results of 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> order terms approximately

overlap with analytical solution in Fig. 6.19. The difference between 1<sup>st</sup> order term and analytical results decreases, but it has not vanished.

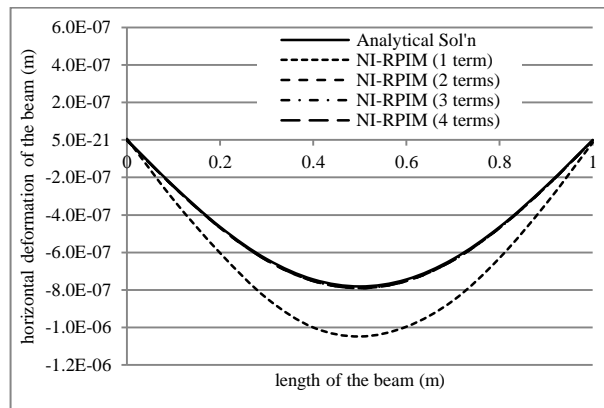


Figure 6.19. Effect of Taylor series expansion terms on vertical deformation of 33 nodes simply supported beam with 2D rectangular shaped Taylor integration cells

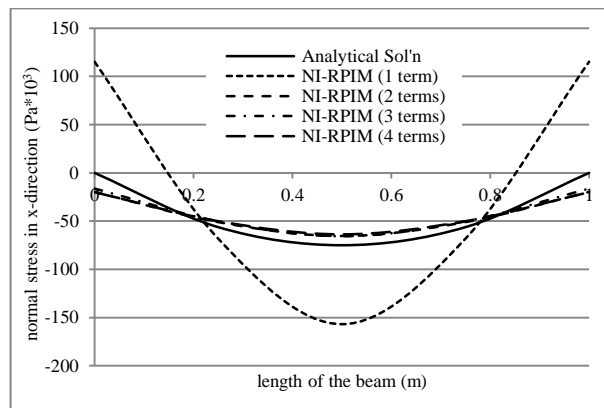


Figure 6.20. Effect of Taylor series expansion terms on axial stress of 18 nodes simply supported beam with 2D rectangular shaped Taylor integration cells

Similar condition can be seen in the stress results of simply supported beam. In Fig. 6.20, the results show that 1<sup>st</sup> order term solution has insufficient results. However, 2<sup>nd</sup> and greater order terms have closed enough results to analytical solution. In Fig. 6.21, nearly 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> order terms stress results are very close to analytical solution. The difference between 1<sup>st</sup> order terms results and analytical results decreases, but the stability of 1<sup>st</sup> order terms vanishes and a fluctuation occurs.



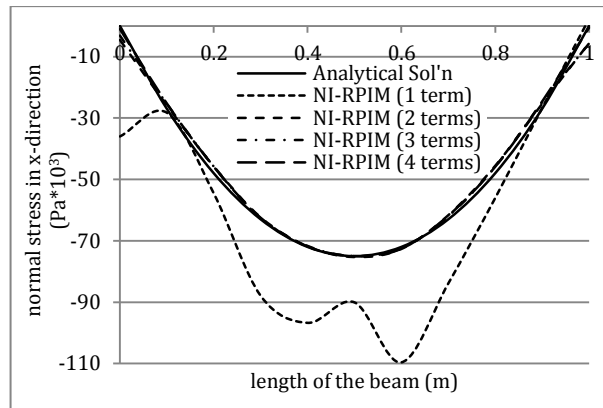


Figure 6.21. Effect of Taylor series expansion terms on axial stress of 33 nodes simply supported beam

It has been reported that first-order finite difference scheme (Anderson, 1995) does not give sufficient results in most of the computational fluid dynamics (CFD) analysis and second-order terms is suggested. Similar condition is detected in this section for 2D elastic problems.

### 6.2.2 Results and Discussions

In this section, the effect of higher order terms of Taylor series expansion is investigated up to 4<sup>th</sup> order within the method of NI-RPIM, which is used by Liu et al. (2007). Rectangular shaped Taylor integration cells are used. Three different case studies for elastic problems are examined with plane stress assumption. Also the effect of number of nodes is observed by changing the used number of nodes.

Some findings can be summarized that;

- increasing the number of nodes gives better results
- it has been detected that 2<sup>nd</sup> order Taylor terms can satisfy suitable result with respect to analytical solution
- fluctuation may occur in the 1<sup>st</sup> order terms in NI-RPIM
- in axial loaded beam results, higher order terms give more realistic results, especially on the investigation of stress conditions
- FEM, NI-RPIM with Gaussian and Taylor integration can give similar results as analytical solution
- on the contrary, high solution times in fewer nodes; NI-RPIM with Taylor series expansion has benefits and faster than Gaussian integration in the solution times, especially using lots of numbers in the analysis

### 6.3 Triangular Shaped Taylor Integration Cells for 2D Cases

The usage of rectangular shaped Taylor integration cells may not be available, when irregular distributed nodes are used. Highly distorted node distribution creates rectangular integration cells with rough shapes. In that case, the usage of triangular shaped Taylor integration cells supports more flexible and easily created integration cells. The cells need only three nodes, which is enough for the integration operation. Its application is mentioned in the below.

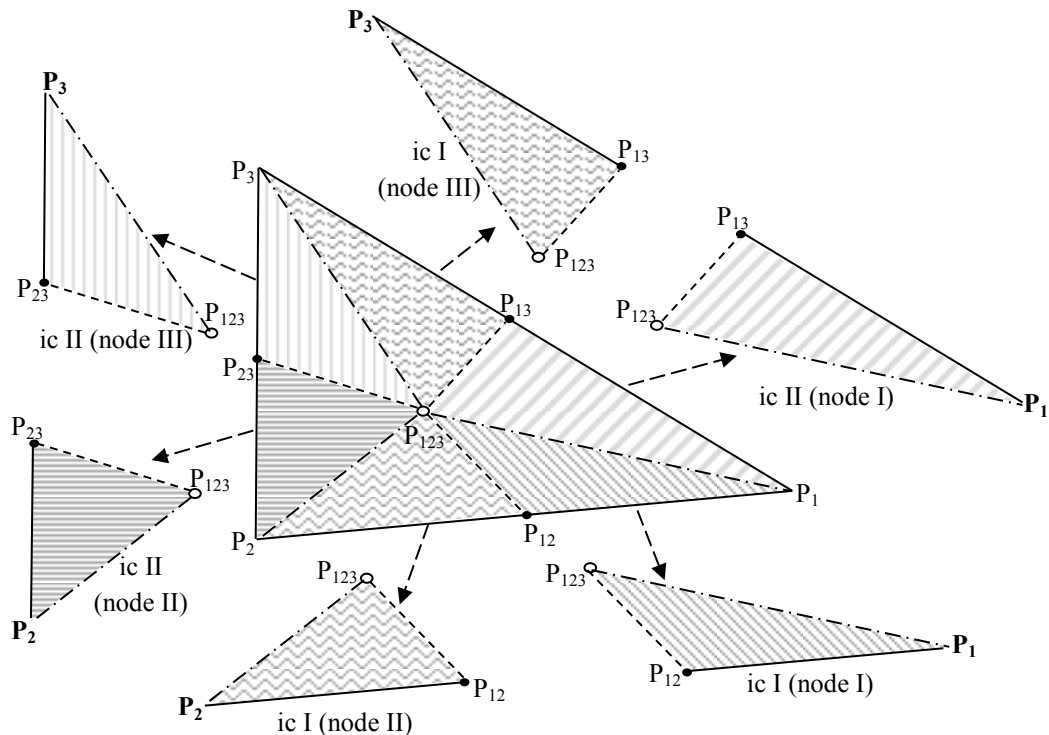


Figure 6.22. Sub-division of triangular background cell for each node

In Fig. 6.22, a triangular background cell is given. The cell which is formed by field nodes of  $P_1 (x_1, y_1)$ ,  $P_2 (x_2, y_2)$  and  $P_3 (x_3, y_3)$  is further divided into 2 different integration cells (ic I and II) for each node. So, the total number of integration cells for a field node can be calculated by multiplying 2 by the number of triangular cells that are connected to the field node.  $P_{12}$ ,  $P_{13}$  and  $P_{23}$  are centre point of each side line of triangular cell.  $P_{123}$  is centre point of surface area. Subdivision of tetrahedral integration cell components for each node is tabulated in Table 6.2.

Table 6.2. Components of edges for subdivided Taylor Integration cells for each node (triangular geometry)

		integration cell (ic)	components of edges		
P <sub>1</sub>	I	P <sub>1</sub>	P <sub>12</sub>	P <sub>123</sub>	
	II	P <sub>1</sub>	P <sub>123</sub>	P <sub>13</sub>	
P <sub>2</sub>	I	P <sub>2</sub>	P <sub>123</sub>	P <sub>12</sub>	
	II	P <sub>2</sub>	P <sub>23</sub>	P <sub>123</sub>	
P <sub>3</sub>	I	P <sub>3</sub>	P <sub>123</sub>	P <sub>13</sub>	
	II	P <sub>3</sub>	P <sub>23</sub>	P <sub>123</sub>	

The integrations in Eq. 5.19,  $\int 1d\Omega$ ,  $\int (\Delta x)d\Omega$ ,  $\int (\Delta y)d\Omega$ ,  $\int (\Delta x^2)d\Omega \dots$  must be calculated for triangular background cells. However, integration of terms of  $x, y, x^2, \dots$  includes complex operations for irregular triangular shapes. Hence, geometry can be transformed into natural coordinates by using Jacobian transformation. In Fig. 6.23, the transformation of triangular geometry from global to natural coordinates is shown. P<sub>1</sub>', P<sub>123</sub>' and P<sub>13</sub>' represent the transformed natural coordinates from global coordinates of P<sub>1</sub>, P<sub>123</sub> and P<sub>13</sub> points, respectively.

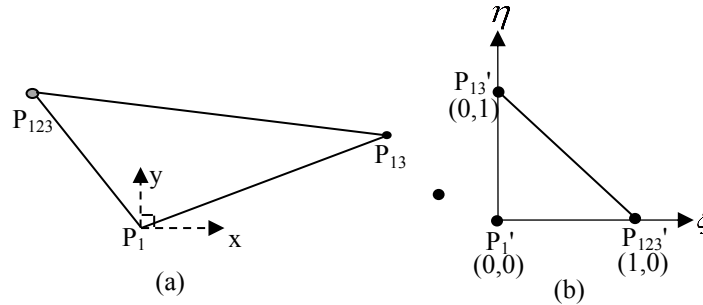


Figure 6.23. Transformation of a triangular integration cell from global (a) to natural (b) coordinates

The bounds of integral starts from zero to upper natural coordinate. Hence P<sub>1</sub>, P<sub>2</sub> and P<sub>3</sub> are placed at P<sub>1</sub>' in integral calculations. This condition can provide to no usage of parallel axis theorem to carry integration results to related edge node. Hence, x<sub>1</sub> and y<sub>1</sub> in Eq. 5.17, 5.18 and 5.19 are assigned x and y coordinates of P<sub>1</sub>, P<sub>2</sub> and P<sub>3</sub> in related integral node calculation.

The transformation of coordinates is given in Eq. 6.6 and 6.7 (Bhowmick and Shontz, 2012).  $\xi$  and  $\eta$  are natural coordinates.

$$x = x_1 + (x_2 - x_1) \cdot \xi + (x_3 - x_1) \cdot \eta \quad (6.6)$$

$$y = y_1 + (y_2 - y_1) \cdot \xi + (y_3 - y_1) \cdot \eta \quad (6.7)$$

Determinant of Jacobian (J) transformation matrix are given in Eq. 6.8.

$$DET(J) = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} = 2 \cdot Area \quad (6.8)$$

The transformation of integration of uniform geometry in natural coordinates is given in Eq. 6.9. Its application to tetrahedron geometry is given in Eq. 6.10. Hence integration in terms of  $x, y, x^2, \dots$  can be easily calculated as follows:

$$\int f(x, y) d\Omega = \int f[x(\xi, \eta), y(\xi, \eta)] * |DET(J)| d\Omega \quad (6.9)$$

$$\int_D f(x, y) dD = |DET(J)| * \int_0^1 d\xi \int_0^{1-\xi} f[x(\xi, \eta), y(\xi, \eta)] d\eta \quad (6.10)$$

Even if suitable orientation is used in Table 6.2, in some cases node coordinate orientations changes with respect to construction of triangular cells in Delaunay triangulation technique. Its symptoms can be determined with negative value of determinant of Jacobian matrix. For this reason, upper integration bounds in Eq. 6.9 are updated with respect to where the subdivided tetrahedral cell exists, which are given in Table 6.3.

Table 6.3. Direction of triangular cell and used bounds of integral in natural coordinates.

	$\Delta x_{12}$	$\Delta y_{13}$	$\int_0^1 f[x, y] d\eta$	$\int_0^1 f[x, y] d\xi$
I	+	+	$1 - \xi$	+1
II	+	-	$-(1 - \xi)$	+1
III	-	+	$1 + \xi$	-1
IV	-	-	$-(1 + \xi)$	-1

Where  $\Delta x_{12}$  is the sign of difference between  $x_2 - x_1$  and  $\Delta y_{13}$  is the sign of difference between  $y_3 - y_1$  for each sub-divided integration cell. The signs of  $\xi$  and  $\eta$  in Eq. 6.6 and 6.7 is also regulated with respect to signs of  $\Delta x_{12}$  and  $\Delta y_{13}$ .

These calculations are applied for each triangular cell and the results are summed for the corresponding field node. Hence, a node includes more than one triangular

integration cells, which looks like a polyhedron geometry. However, all main structure of integrations is carried on each subdivided triangular cells for related nodes.

#### **6.4 Hexahedral Shaped Taylor Integration Cells for 3D Cases**

Rectangular shaped Taylor integration cells are extended and adapted for 3D solution by using hexahedral shaped Taylor integration cells. Each integration domains include 8 field nodes.

##### **6.4.1 Numerical Examples**

Three different case studies are examined for solution of nodal integration by using hexahedral shaped Taylor integration cells. Effects of Taylor expansion terms in nodal integration are investigated by using a program that is developed in Fortran. The code is written based on the studies in literature (Liu and Gu, 2005; Liu et al., 2007). A computer has a processor speed of 2.4 GHz and a memory capacity of 3 GB is used in the analyses.

A simply supported beam with distributed load, a torsion bar and an L-shaped beam under uniform load are examined. They are given in Fig. 6.24. A linear elastic material is used with a Young Modulus of 200 GPa and Poisson's ratio of 0.0. The selection of Poisson's ratio as 0.0 is aimed for providing similar conditions as analytical solutions.

A bar/beam with square section of  $0.1 \text{ m} \times 0.1 \text{ m}$  and a length of 1.00 m is used in case I and II. L shaped beam has a length of 1.00 m in the x and 0.5 m in the z-direction. A force couple is used for applied torque in case I, which is equal to 75 Nm. Applied distributed load is  $22500 \text{ N/m}^2$  in case II. Uniformly varying load is used as  $2250 \text{ N/m}$  in case III. Node distributions are same in FEM and meshfree models.

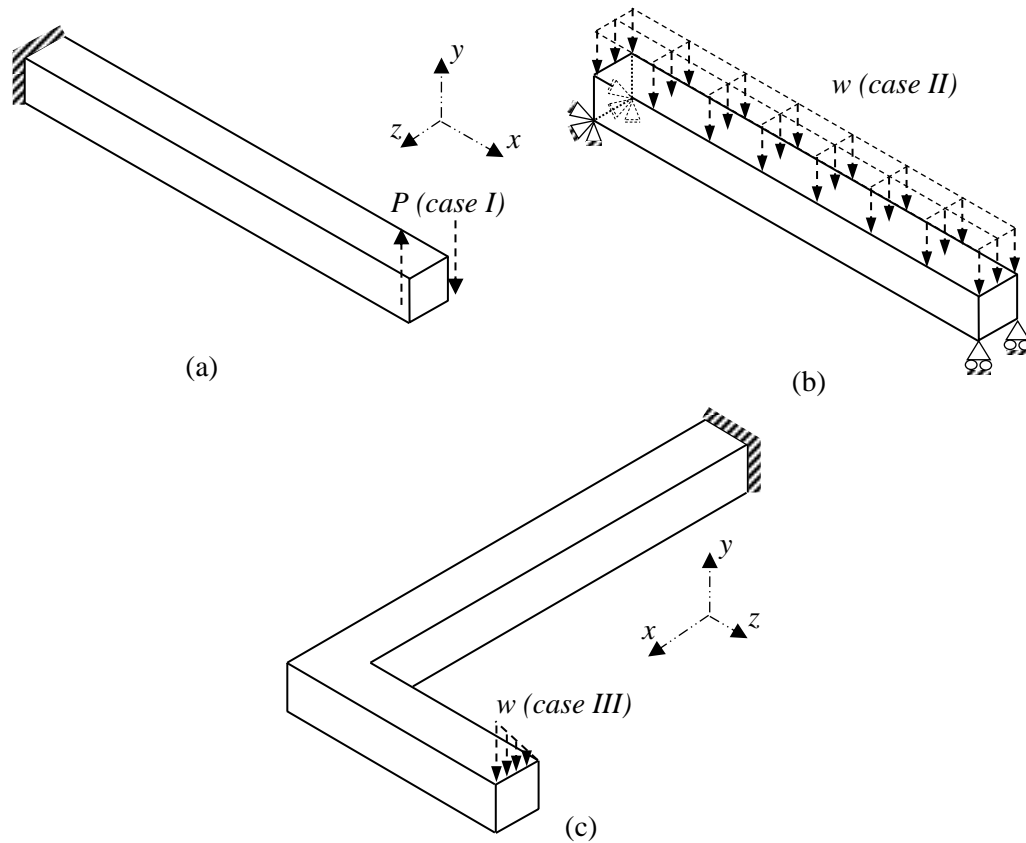


Figure 6.24. The used models in the analyses and boundary conditions, a) bar under torsion (case I) b) simply supported beam problem with distributed loaded (case II), c) L-shaped beam under uniformly varying load (case III)

SOLID 185 finite elements are used in FEM models, which have 8 nodes at the edges. ANSYS 14 software package program is used in FEM pre-processor, solution and post-processor steps. An element length of 0.05 m is selected in models as shown in Fig. 6.25.

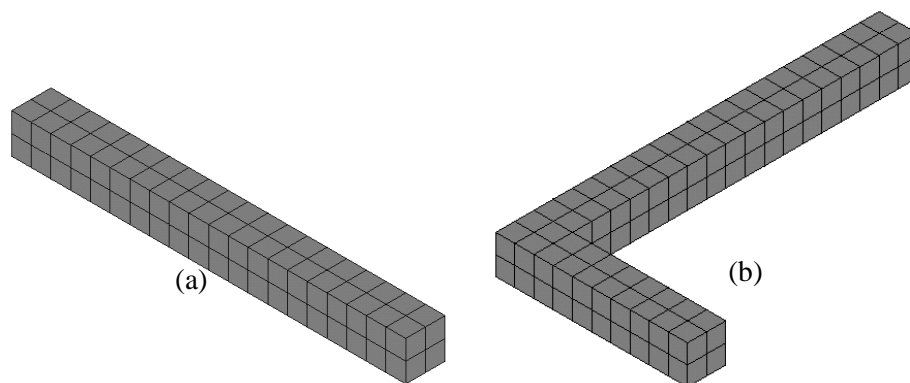


Figure 6.25. 3D FEM models of analysed cases; a) beam for case II, b) L shaped beam for case III

Meshfree models are given in Fig. 6.26, which have 189 nodes in case II and 270 nodes in case III. FEM and meshfree models of case I have 6561 nodes with an element length of 0.0125 m.

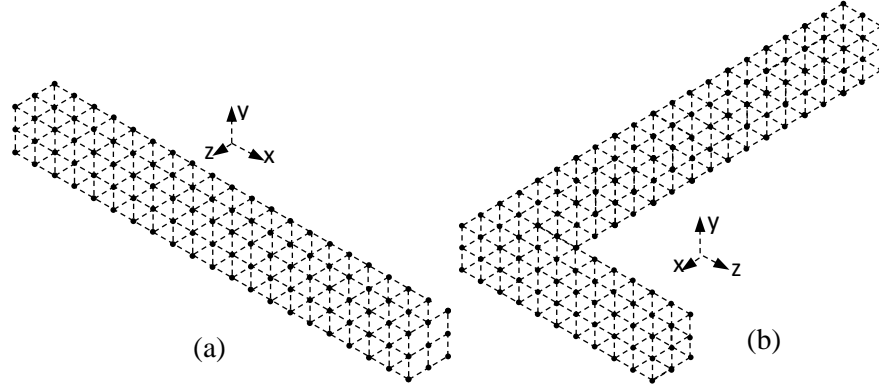


Figure 6.26. 3D meshfree models of analysed cases; a) beam for case II, b) L shaped beam for case

The vertical deflection of simply supported beam (Beer et al., 2009) with distributed loading (case II) is given as follows;

$$y = -\frac{w}{24 \times E \times I} \times (x^4 - 2 \times L \times x^3 - L^3 \times x) \quad (6.11)$$

where  $w$  is applied distributed load and  $I$  represents inertia of the beam.  $E$  is the Young's Modulus and  $x$  is the distance from fixed support location.  $L$  is also total length of beam. Its bending stress at the upper and below surfaces is given as;

$$\sigma = \frac{M \times c}{I} \quad (6.12)$$

where  $c$  is the distance between upper surface and neutral axis of the beam.

The twist angle ( $\Phi$ ) and maximum shear stress ( $\tau_{\max}$ ) in a torsion bar (case I) are analytically given (Beer et al., 2009; Timoshenko and Goodier, 1951) as follows;

$$\Phi = \frac{T \times c}{c_2 \times a \times b^3 \times G} \quad (6.13)$$

$$\tau_{\max} = \frac{T}{c_1 \times a \times b^2} \quad (6.14)$$

where  $T$  is applied torque,  $a$  and  $b$  are height and width of the beam, which are equal to 0.1 m and 0.1 m.  $c_1$  and  $c_2$  are constants for non-circular torsion of members, which are equal to 0.208 and 0.1406 with respect to square cross section.

$\alpha_c$  is used as 3.00, support domain size is used as 1.30 and  $q$  is used as 1.03 as a default parameters in the solutions of RPIM method.

### 6.4.1.1 Torsion Bar

Torsion of a non-circular bar with square cross section is investigated. Twist angle results are taken from side surface of bar, where the maximum shear stresses occur. Shear stress results are taken from the upper surface at the mid-length of bar.

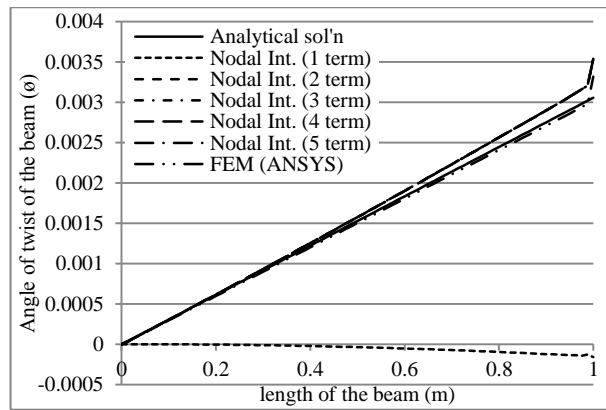


Figure 6.27. Comparison of twist angle results of a non-circular bar under torsion (case I) for analytical, FEM and nodal int. techniques

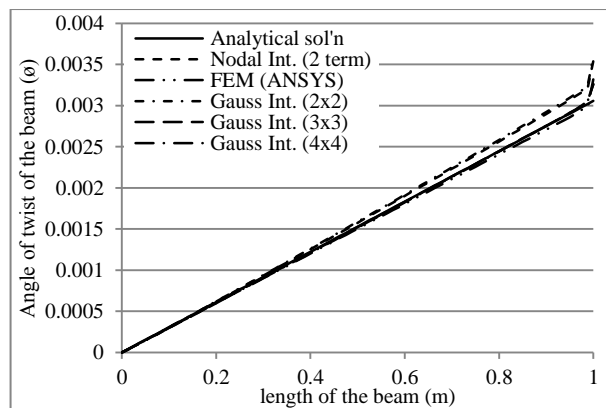


Figure 6.28. Comparison of twist angle results of a non-circular bar under torsion (case I) for analytical, Gauss and nodal int. (2<sup>nd</sup> order terms) techniques

Twist angle results (case I) are given in Fig. 6.27 and 6.28 for nodal and Gauss integrations. It is observed that 1<sup>st</sup> terms of nodal integration includes serious errors with respect to analytical solution. 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup> and 5<sup>th</sup> orders of nodal integration



results have similar characteristics with analytical results. But FEM results are closer than RPIM to nodal integration results. RPIM with Gauss integrations give similar results as 2<sup>nd</sup> and higher order terms results of nodal integration in Fig. 6.28. The number of sampling points has no effect on the results for this case.

Table 6.4. Comparison of shear stress results of a 3D non-circular bar under torsion (case I) for analytical, FEM and nodal int. techniques

Solution technique	$\tau_{xy}$ (kPa)
FEM (ANSYS)	352.19
Nodal Int. (1 term)	388.90
Nodal Int. (2 terms)	366.11
Nodal Int. (3 terms)	365.96
Nodal Int. (4 terms)	366.04
Nodal Int. (5 terms)	366.04
Gauss Int. (2x2)	369.88
Gauss Int. (3x3)	359.47
Gauss Int. (4x4)	354.01
Analytical solution	360.57

Shear stress results are compared in Table 6.4. 2<sup>nd</sup> and further order terms in nodal integration and Gauss integration with 3x3 and 4x4 sampling points give better results than FEM compared to analytical solution. Gauss integration with 3x3 sampling points gives the closest results to analytical results. 1<sup>st</sup> order terms of nodal integration and 2x2 sampling points of Gauss integration results include less sensitivity.

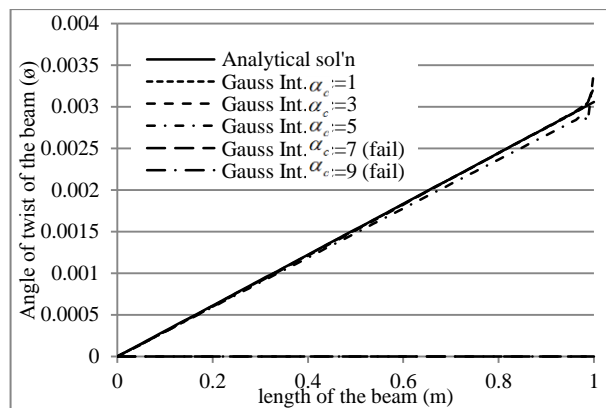


Figure 6.29. Comparison of effect of on twist angle results of a 3D non-circular bar under torsion (case I) for 4×4 gauss int.

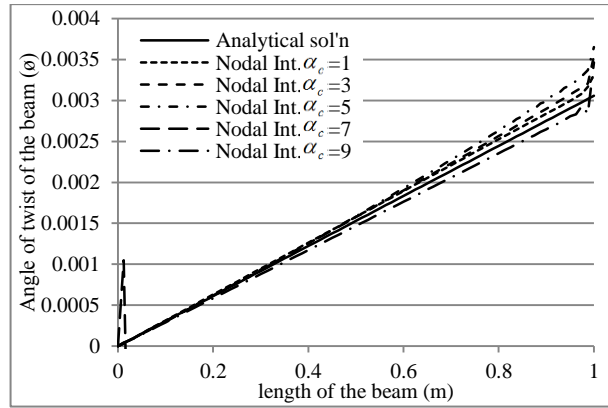


Figure 6.30. Comparison of effect of  $\alpha_c$  on twist angle results of a non-circular bar under torsion (case I) for nodal int. ( $2^{\text{nd}}$  order terms)

Effect of  $\alpha_c$  on twist angle results (case I) are given in Fig. 6.29 and 6.30. Nearly  $\alpha_c$  values of 1.00, 3.00 and 5.00 have the same results on the solutions of Gauss integration.  $\alpha_c$  values of 7.00 and 9.00 cause failure of shape function construction.  $\alpha_c$  values of 1.00, 3.00, 7.00 and 9.00 in nodal integration have similar results and enough accuracy with analytical solution.  $\alpha_c$  value of 5.00 in nodal integration has the least accuracy.

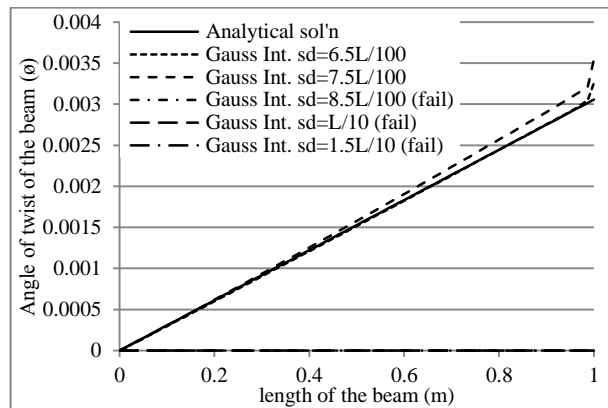


Figure 6.31. Comparison of effect of sd on twist angle results of a 3D non-circular bar under torsion (case I) for  $4 \times 4$  Gauss int.

Effects of support domain size are given in Fig. 6.31 and 6.32 for case I. Support domain size of  $6.5 \times L/100$  has similar results as analytical solution in Gauss integration. The usage of support domain size of  $7.5 \times L/100$  decreases accuracy. Further increments of support domain sizes cannot satisfy construction of shape functions and fail. Results of different support domain sizes have similar results and enough accuracy without  $1.5 \times L/10$  in nodal integration.

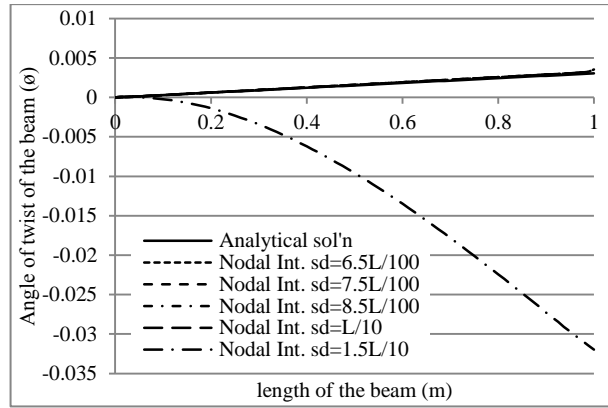


Figure 6.32. Comparison of effect of  $sd$  on twist angle results of a 3D non-circular bar under torsion (case I) for nodal int. ( $2^{nd}$  order terms)

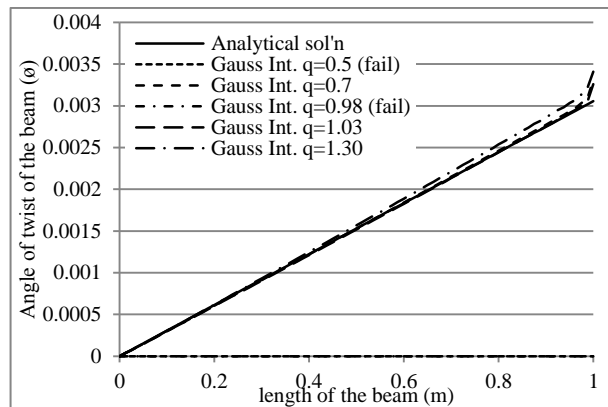


Figure 6.33. Comparison of effect of  $q$  on twist angle results of a 3D non-circular bar under torsion (case I) for  $4 \times 4$  Gauss int.

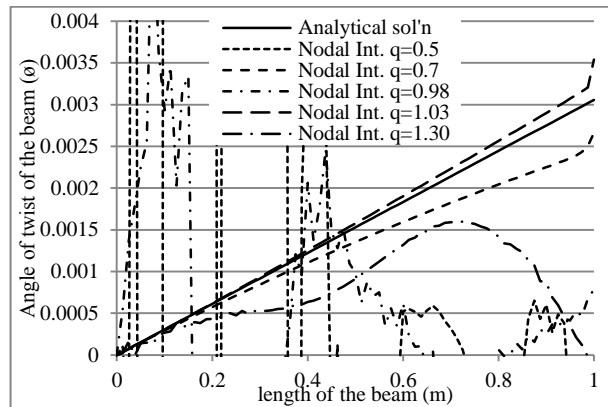


Figure 6.34. Comparison of effect of  $q$  on twist angle results of a 3D non-circular bar under torsion (case I) for nodal int. ( $2^{nd}$  order terms)

Effects of  $q$  on twist angle results are given in Fig. 6.33 and 6.34.  $q$  values of 0.5 and 0.98 cause to failure in construction of shape functions in RPIM with Gauss integration.  $q$  values of 0.7 and 1.03 have better accuracy than 1.30. In nodal integration results, all  $q$  values have high fluctuations when it is used a different value from 1.03.

Table 6.5. Comparison of effect of  $\alpha_c$  on shear stress results of a 3D non-circular bar under torsion (case I) for nodal int. (2<sup>nd</sup> order terms) and 4×4 Gauss int.

Effect of $\alpha_c$	$\tau_{xy}$ (kPa)	$\tau_{xy}$ (kPa)	Effect of $\alpha_c$
Nodal Int. ( $\alpha_c=1$ )	379.95	362.64	Gauss Int. ( $\alpha_c=1$ )
Nodal Int. ( $\alpha_c=3$ )	366.11	354.01	Gauss Int. ( $\alpha_c=3$ )
Nodal Int. ( $\alpha_c=5$ )	357.56	350.63	Gauss Int. ( $\alpha_c=5$ )
Nodal Int. ( $\alpha_c=7$ )	681.29	-	Gauss Int. ( $\alpha_c=7$ )
Nodal Int. ( $\alpha_c=9$ )	344.64	-	Gauss Int. ( $\alpha_c=9$ )

Table 6.6. Comparison of effect of sd on shear stress results of a 3D non-circular bar under torsion (case I) for nodal int. (2<sup>nd</sup> order terms) and 4×4 Gauss int.

Effect of sd	$\tau_{xy}$ (kPa)	$\tau_{xy}$ (kPa)	Effect of sd
Nodal Int. (sd=6.5×L/100)	366.11	354.01	Gauss Int. (sd=6.5×L/100)
Nodal Int. (sd=7.5×L/100)	366.11	366.93	Gauss Int. (sd=7.5×L/100)
Nodal Int. (sd=8.5×L/100)	366.11	-	Gauss Int. (sd=8.5×L/100)
Nodal Int. (sd=L/10)	366.11	-	Gauss Int. (sd=L/10)
Nodal Int. (sd=1.5×L/10)	370.82	-	Gauss Int. (sd=1.5×L/10)

Table 6.7. Comparison of effect of  $q$  on shear stress results of a 3D non-circular bar under torsion (case I) for nodal int. (2<sup>nd</sup> order terms) and 4×4 Gauss int.

Effect of $q$	$\tau_{xy}$ (kPa)	$\tau_{xy}$ (kPa)	Effect of $q$
Nodal Int. ( $q=0.5$ )	176.78	-	Gauss Int. ( $q=0.5$ )
Nodal Int. ( $q=0.7$ )	356.05	357.75	Gauss Int. ( $q=0.7$ )
Nodal Int. ( $q=0.98$ )	-796.27	-	Gauss Int. ( $q=0.98$ )
Nodal Int. ( $q=1.03$ )	366.11	354.01	Gauss Int. ( $q=1.03$ )
Nodal Int. ( $q=1.30$ )	80.77	363.03	Gauss Int. ( $q=1.30$ )

Effects of  $\alpha_c$ , sd and  $q$  on shear stress results (case I) are given in Table 6.5, 6.6 and 6.7, respectively. Analytical solution results of shear stress are equal to 360.57 kPa for Table 6.5, 6.6 and 6.7.  $\alpha_c$  values of 7.00 and 9.00, sd values of 8.5×L/100, L/10 and 1.5×L/10,  $q$  values of 0.5 and 0.98 in Gauss integration cannot satisfy construction of shape functions and fail.  $\alpha_c$  value of 1.00 has the closest results. When increasing its value, accuracy decreases. sd values of 6.5×L/100 and 7.5×L/100 have similar accuracy. Different values of  $q$  results have closed enough to analytical solution. When nodal integration results are discussed,  $\alpha_c$  value of 5.00 has the closest result to analytical solution. Other results of  $\alpha_c$  values have less accuracy, especially  $\alpha_c$  value of 7.00. All results of sd values are same without sd

value of  $1.5 \times L/10$ . Results of the highest sd value has less accuracy.  $q$  values of 0.7 and 1.03 have good accuracy. However, other  $q$  values, especially  $q$  value of 0.98, have wrong results.

### 6.4.1.2 Simply Supported Beam

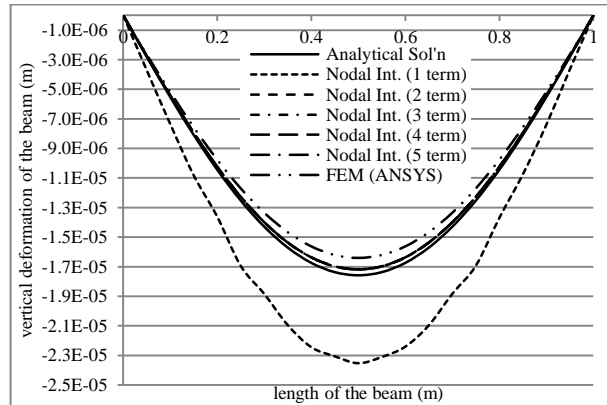


Figure 6.35. Comparison of vertical deformation results of distributed loaded simply supported beam (case II) for analytical, FEM and nodal int. techniques

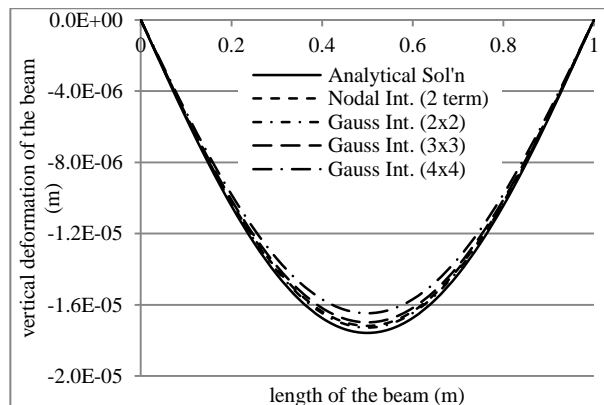


Figure 6.36. Comparison of vertical deformation results of distributed loaded simply supported beam (case II) for analytical, Gauss and nodal int. ( $2^{\text{nd}}$  order terms) techniques

Effect of different solution techniques are given in Fig. 6.35 and 6.36 for case study of II. The usage of only  $1^{\text{st}}$  order term of Taylor series expansion in nodal integration gives less accurate results with respect to analytical solution. However using more than one term in nodal integration rapidly provides accuracy, which has greater conformity with analytical solution than FEM (ANSYS). Results of two or more terms in nodal integration nearly show similar characteristics. Gauss integration results are also compatible with analytical solution.

Bending stress results are given in Fig. 6.37 and 6.38 for simply supported beam solution. Results of first term in nodal integration do not provide accuracy and

includes high fluctuations. But usage of two or more terms in nodal integration provides accuracy with respect to analytical solution. Gauss integration results have similar characteristics with second term results of nodal integration.

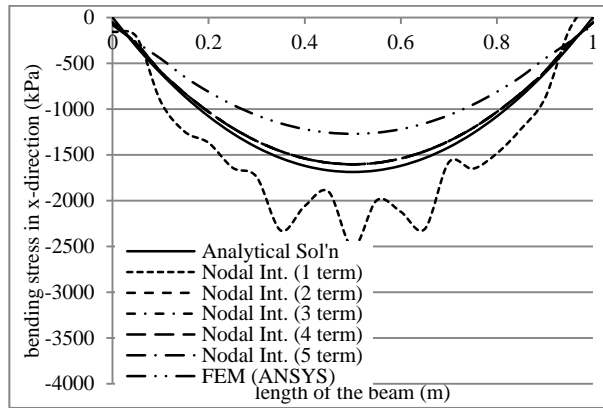


Figure 6.37. Comparison of bending stress results of distributed loaded simply supported beam (case II) for analytical, FEM and nodal int. techniques

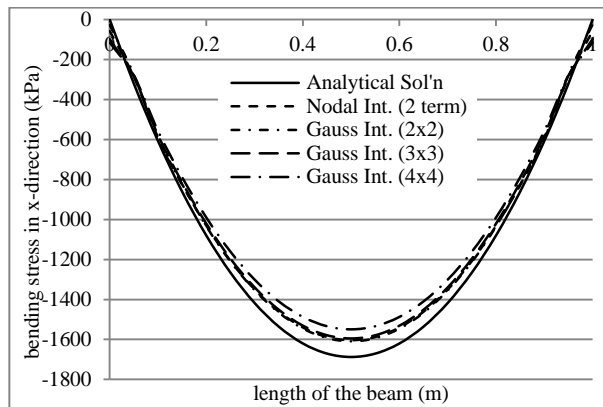


Figure 6.38. Comparison of bending stress results of distributed loaded simply supported beam (case II) for analytical, Gauss and nodal int. (2<sup>nd</sup> order terms) techniques

Effects of  $\alpha_c$  on bending stress results of simply supported beam are given in Fig. 6.39 and 6.40. It is observed that  $\alpha_c$  values of 1.00, 3.00, 5.00 and 9.00 have same values in Gauss integration results. However, results of  $\alpha_c$  values of 7.00 are only different than other values of  $\alpha_c$  results and includes less accuracy. Similar responses of  $\alpha_c$  values occur in nodal integration results. However results of  $\alpha_c$  values include greater deformation values than analytical solution. In Gauss integration, deformation results are less than analytical solution.

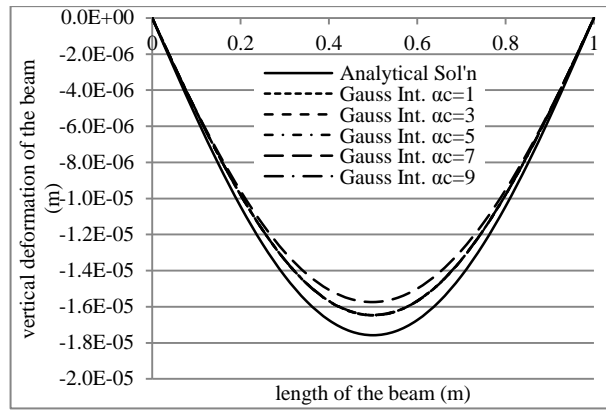


Figure 6.39. Comparison of effect of  $\alpha_c$  on vertical deformation results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.

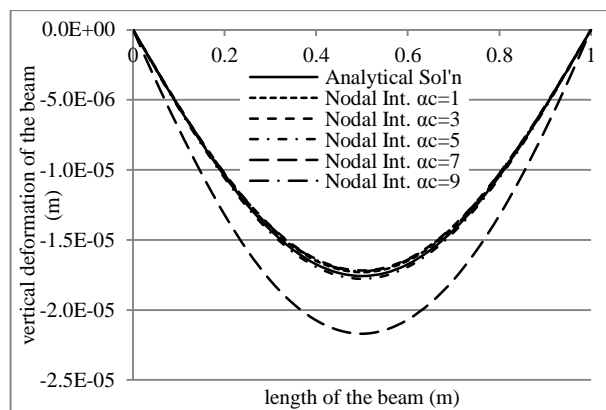


Figure 6.40. Comparison of effect of  $\alpha_c$  on vertical deformation results of distributed loaded simply supported beam (case II) for nodal int. (2<sup>nd</sup> order terms)

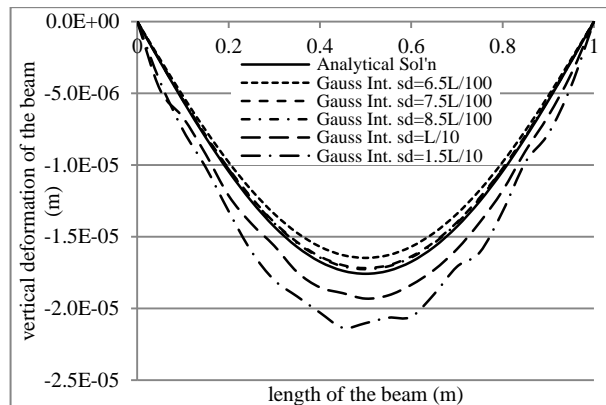


Figure 6.41. Comparison of effect of sd on vertical deformation results of distributed loaded simply supported beam (case II) for 4×4 Gauss int.

Effects of support domain size are given on bending stress results in Fig. 6.41 and 6.42. Results of sd value of  $6.5 \times L/100$  are compatible with analytical solution in Gauss integration results. Increasing support domain size to  $7.5 \times L/100$  and  $8.5 \times L/100$  increases accuracy. But larger support domain sizes as  $L/10$  and  $1.5 \times L/10$

decrease accuracy of solution. Same results of various support domain sizes occur in nodal integration.

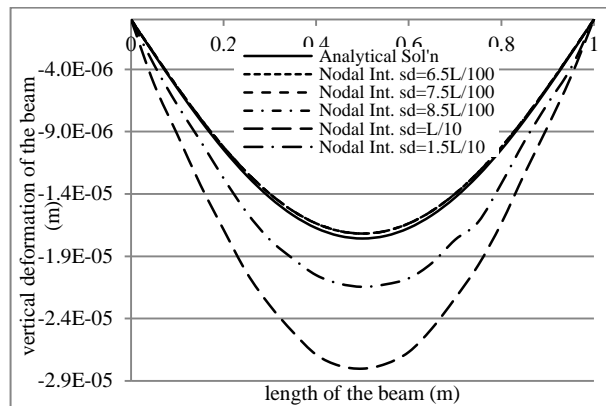


Figure 6.42. Comparison of effect of sd on vertical deformation results of distributed loaded simply supported beam (case II) for nodal int. ( $2^{\text{nd}}$  order terms)

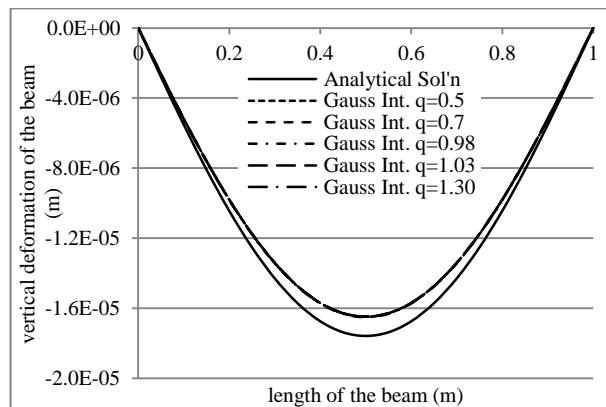


Figure 6.43. Comparison of effect of  $q$  on vertical deformation results of distributed loaded simply supported beam (case II) for  $4 \times 4$  Gauss int.

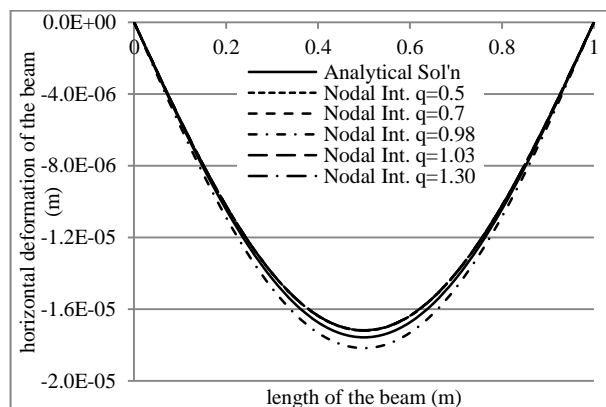


Figure 6.44. Comparison of effect of  $q$  on vertical deformation results of distributed loaded simply supported beam (case II) for nodal int. ( $2^{\text{nd}}$  order terms)

Effects of  $q$  on deformation results of simply supported beam (case II) are given in Fig. 6.43 and 6.44.  $q$  does not have any effect in Gauss integration results. Various



$q$  values have not effect in nodal integration without  $q$  value of 0.98.  $q$  value of 0.98 decreases accuracy of solution.

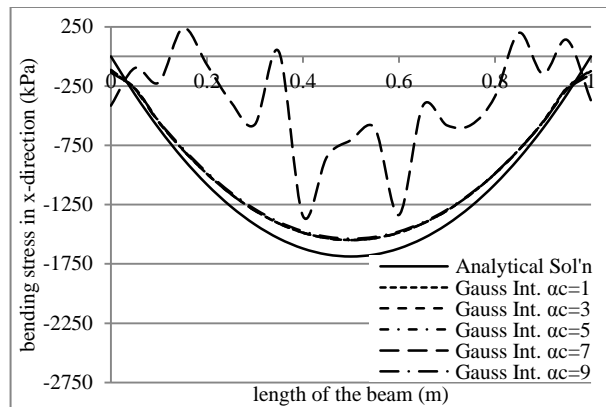


Figure 6.45. Comparison of effect of  $\alpha_c$  on bending stress results of distributed loaded simply supported beam (case II) for  $4 \times 4$  Gauss int.

Results of bending stress distribution (case II) for various  $\alpha_c$  values are given in Fig. 6.45 and 6.46. It is observed that results of  $\alpha_c$  values of 1.00, 3.00, 5.00 and 9.00 have nearly same result in Gauss integration. But result of  $\alpha_c$  value of 7.00 has high fluctuations and cannot satisfy accuracy. Same  $\alpha_c$  responses are available in nodal integration. However,  $\alpha_c$  value of 5.00 gives the best responses in simply supported beam analyses.

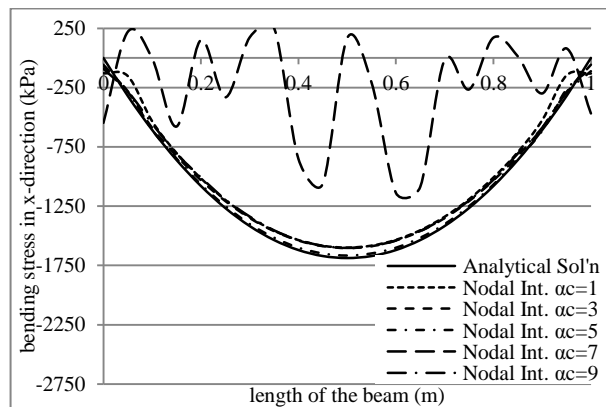


Figure 6.46. Comparison of effect of  $\alpha_c$  on bending stress results of distributed loaded simply supported beam (case II) for nodal int. ( $2^{\text{nd}}$  order terms)

Effect of support domain sizes on bending stress results are given in Fig. 6.47 and 6.48 for case II. sd sizes of  $7.5 \times L/100$  and  $8.5 \times L/100$  have the highest accuracy in the results of Gauss integration. But also sd sizes of  $7.5 \times L/100$  satisfy required accuracy with respect to analytical solution. Results of sd sizes of  $L/10$  and  $1.5 \times L/10$

have high fluctuations. Same responses on results of sd size are observed in nodal integration.

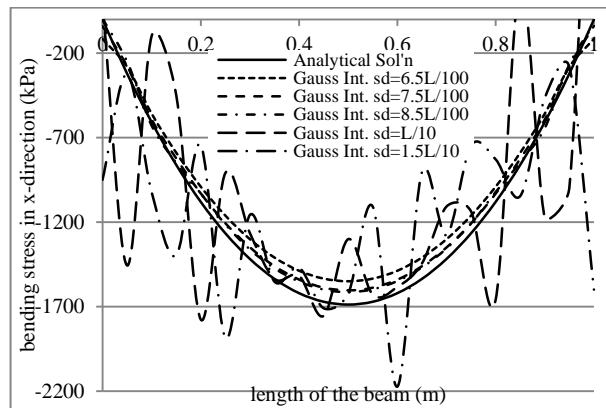


Figure 6.47. Comparison of effect of sd on bending stress results of distributed loaded simply supported beam (case II) for  $4 \times 4$  Gauss int.

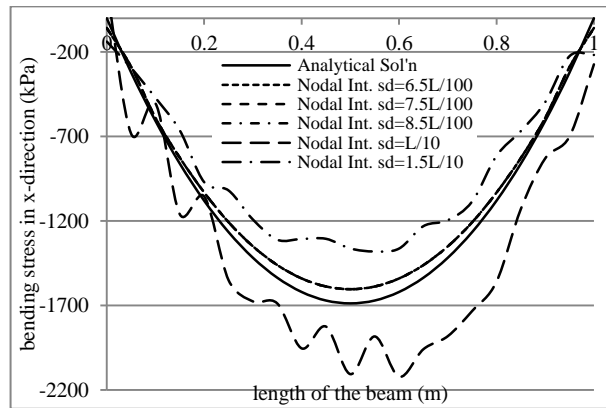


Figure 6.48. Comparison of effect of sd on bending stress results of distributed loaded simply supported beam (case II) for nodal int. ( $2^{\text{nd}}$  order terms)

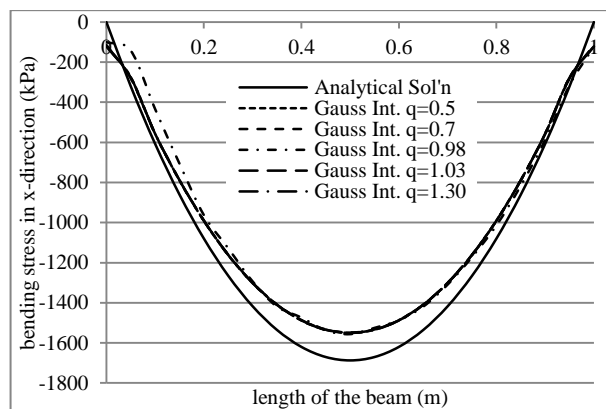


Figure 6.49. Comparison of effect of  $q$  on bending stress results of distributed loaded simply supported beam (case II) for  $4 \times 4$  Gauss int.

$q$  is not effective on Gauss integration results of bending stress in Fig. 6.49 for simply supported beam (case II). However  $q$  value of 0.98 gives different stress responses in nodal integration in Fig. 6.50.

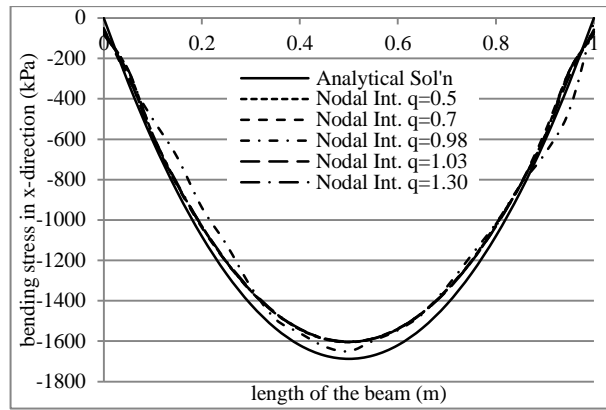


Figure 6.50. Comparison of effect of  $q$  on bending stress results of distributed loaded simply supported beam (case II) for nodal int. ( $2^{\text{nd}}$  order terms)

Solution times of RPIM with nodal and Gauss integration techniques are approximately given in Table 6.8 for simply supported beam. It is seen that usage of only first term in nodal integration has the best solution time. When the used terms in nodal integration increases, solution time in nodal integration increases. The increment rate of solution time in nodal integration is greater than Gauss integration.

Table 6.8. Comparison of solution times of RPIM with nodal and Gauss integration techniques for simply supported beam model of 189 nodes

RPIM with nodal integration		Sol'n time (in Secs)	Sol'n time (in Secs)	RPIM with Gauss integration	
189 nodes	1 term	1.669	1.794	$2 \times 2 \times 2$	189 nodes
	2 terms	4.337	3.635	$3 \times 3 \times 3$	
	3 terms	11.107	6.084	$4 \times 4 \times 4$	
	4 terms	30.466			
	5 terms	55.724			

#### 6.4.1.3 L-Shaped Beam

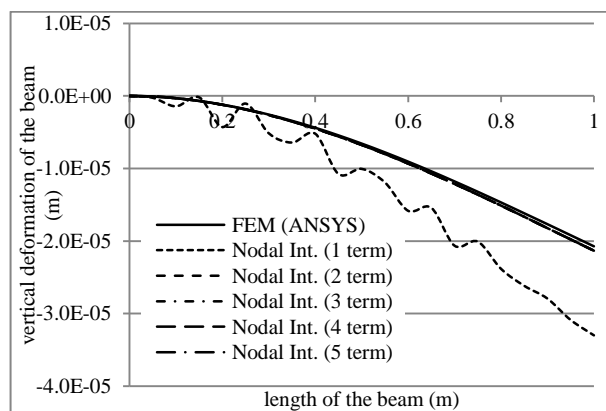


Figure 6.51. Comparison of vertical deformation results of L-shaped beam (case III) for FEM and nodal int. techniques

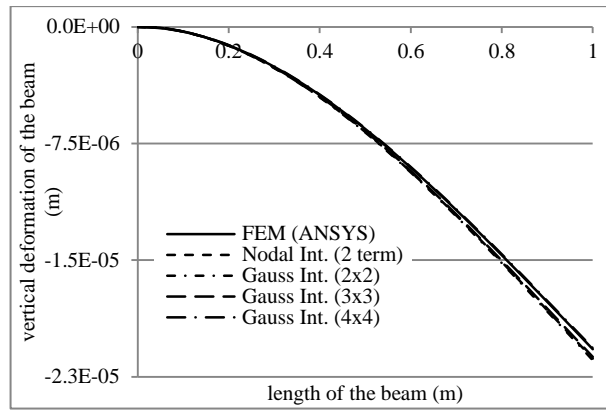


Figure 6.52. Comparison of vertical deformation results of L-shaped beam (case III) for FEM, Gauss and nodal int. (2<sup>nd</sup> order terms) techniques

Displacement results of L-shaped beam (case III) are taken from centre of the beam in x-direction. Bending stress results are taken from middle of upper surface of the beam. RPIM with nodal and Gauss integration results are compared with FEM (ANSYS) solutions in Fig. 6.51 and 6.52. 1<sup>st</sup> order nodal integration includes fluctuations and cannot satisfy accuracy. But further orders of nodal integration and Gauss integration results with sampling points of 2×2, 3×3 and 4×4 have similar results as FEM (ANSYS).

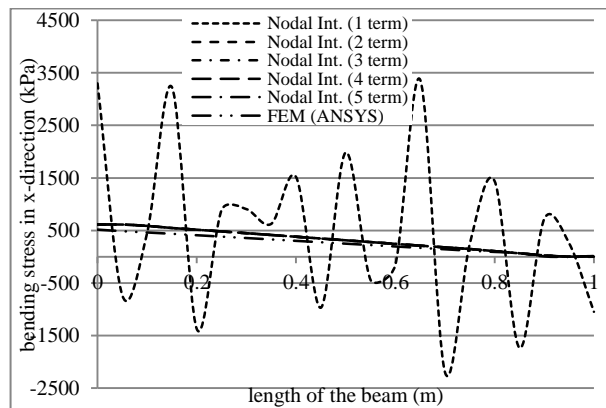


Figure 6.53. Comparison of bending stress results of L-shaped beam (case III) for FEM and nodal int. techniques

Bending stress results for L-shaped beam (case III) are given in Fig. 6.53 and 6.54. Nodal integration with 1<sup>st</sup> term cannot satisfy accuracy and includes high fluctuations. 2<sup>nd</sup> and further terms of nodal integration and Gauss integration results give suitable results with respect to FEM (ANSYS), but a little difference occurs in the results.

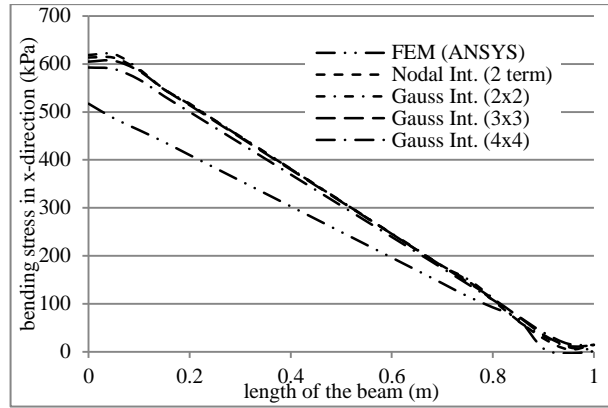


Figure 6.54. Comparison of bending stress results of L-shaped beam (case III) for FEM, Gauss and nodal int. ( $2^{\text{nd}}$  order terms) techniques

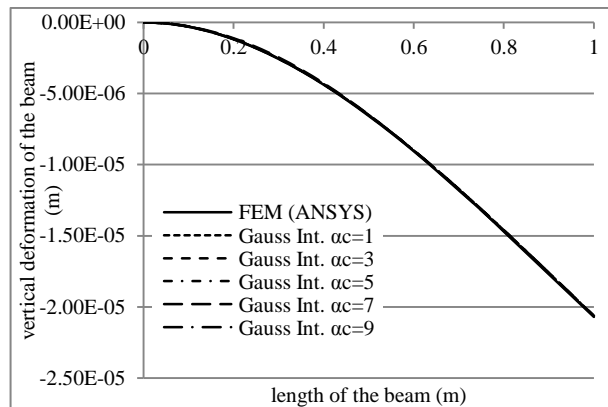


Figure 6.55. Comparison of effect of  $\alpha_c$  on vertical deformation results of L-shaped beam (case III) for  $4 \times 4$  Gauss int.

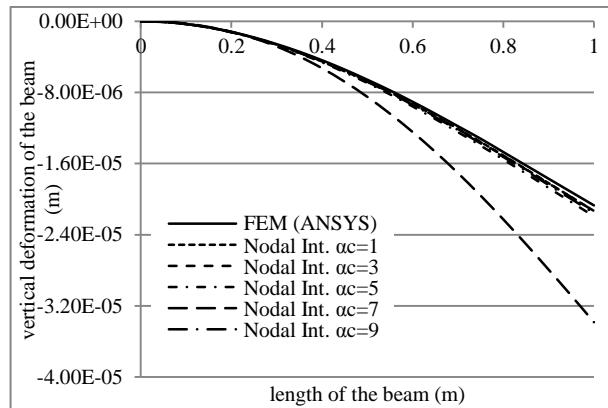


Figure 6.56. Comparison of effect of  $\alpha_c$  on vertical deformation results of L-shaped beam (case III) for nodal int. ( $2^{\text{nd}}$  order terms)

Effects of  $\alpha_c$  on deformation results of L-shaped beam (case III) are given in Fig. 6.55 and 6.56. It is shown that there is no  $\alpha_c$  effect in Gauss integration results.  $\alpha_c$  values of 1.00, 3.00, 5.00 and 9.00 have same deformation results, but  $\alpha_c$  value of 7.00 has less accuracy in results of nodal integration.

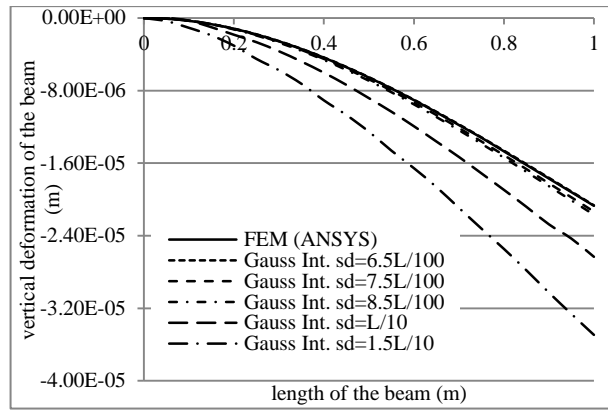


Figure 6.57. Comparison of effect of sd on vertical deformation results of L-shaped beam (case III) for 4×4 Gauss int.

Effects of support domain size (sd) on deformation of L-shaped beam (case III) are given in Fig. 6.57 and 6.58. sd values of  $6.5 \times L/100$ ,  $7.5 \times L/100$  and  $8.5 \times L/100$  in Gauss and nodal integration have same results with FEM (ANSYS) results. However, accuracy of solution decreases when sd value is increased to  $L/10$  and  $1.5 \times L/10$ .

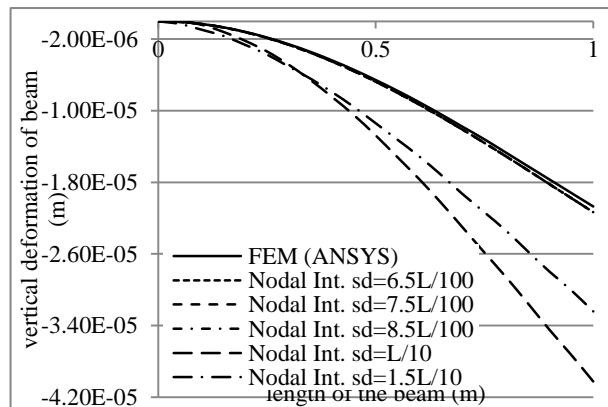


Figure 6.58. Comparison of effect of sd on vertical deformation results of L-shaped beam (case III) for nodal int. ( $2^{\text{nd}}$  order terms)

In Fig. 6.59 and 6.60, effect of  $q$  for nodal and Gauss integration results are given for deformation results of L-shaped beam (case III). There is no effect of  $q$  on results of Gauss integration.  $q$  value of 0.98 has a difference on deformation results in nodal integration. Other deformation results of  $q$  values have same results.

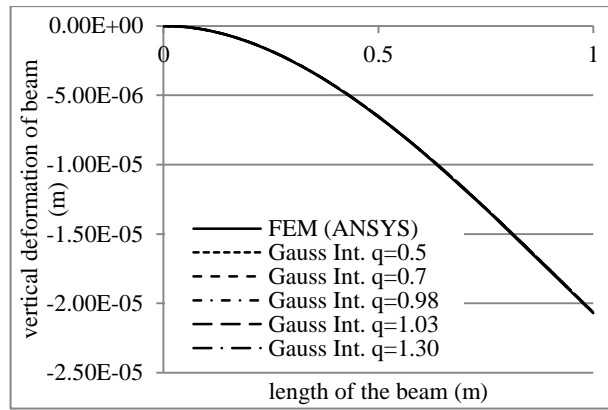


Figure 6.59. Comparison of effect of  $q$  on vertical deformation results of L-shaped beam (case III) for  $4 \times 4$  Gauss int.

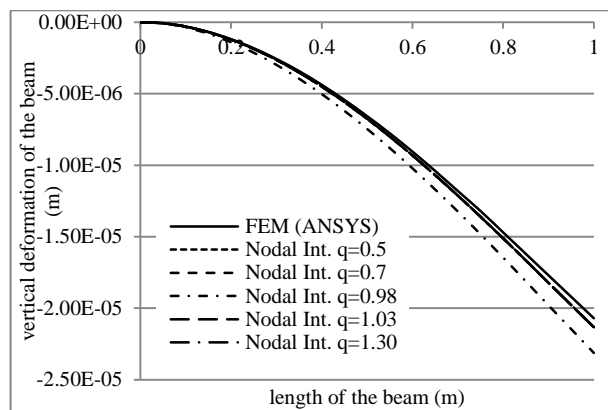


Figure 6.60. Comparison of effect of  $q$  on vertical deformation results of L-shaped beam (case III) for nodal int. ( $2^{\text{nd}}$  order terms)

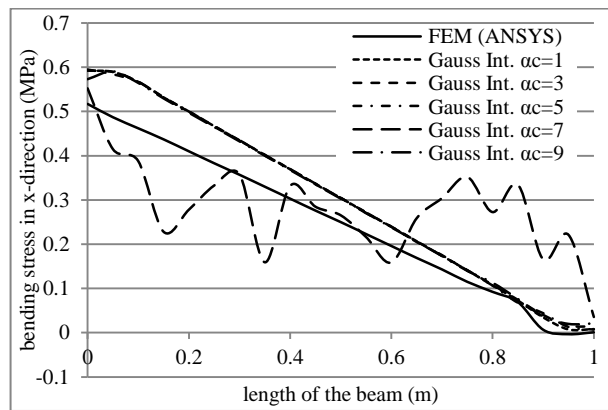


Figure 6.61. Comparison of effect of  $\alpha_c$  on bending stress results of L-shaped beam (case III) for  $4 \times 4$  Gauss int.

Bending stress results of L-shaped beam (case III) are given in Fig. 6.61 and 6.62 for various  $\alpha_c$  values. It is detected that changes on  $\alpha_c$  has no positive effects on both Gauss and nodal integrations. When  $\alpha_c$  value of 7.00 is used, high fluctuations on the results occur and accuracy is lost in nodal integration.

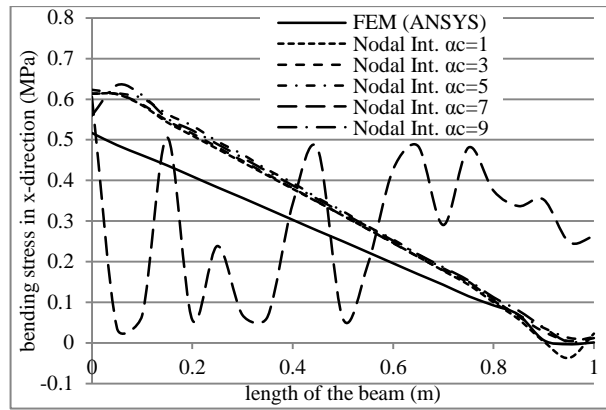


Figure 6.62. Comparison of effect of  $\alpha_c$  on bending stress results of L-shaped beam (case III) for nodal int. ( $2^{\text{nd}}$  order terms)

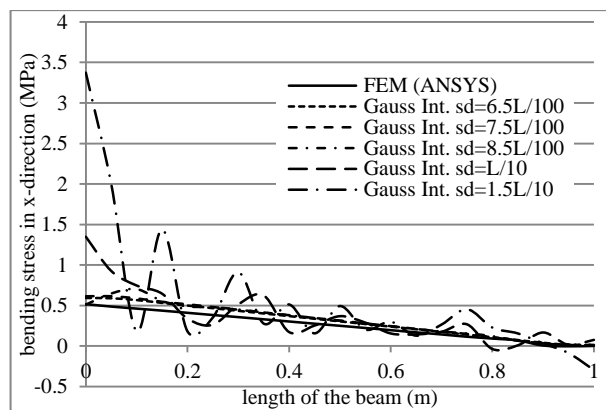


Figure 6.63. Comparison of effect of sd on bending stress results of L-shaped beam (case III) for  $4 \times 4$  Gauss int.

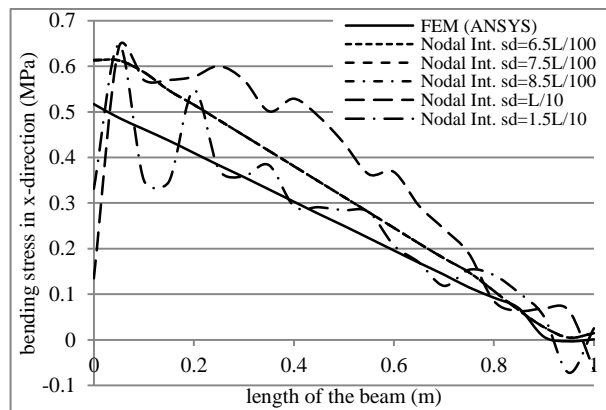


Figure 6.64. Comparison of effect of sd on bending stress results of L-shaped beam (case III) for nodal int. ( $2^{\text{nd}}$  order terms)

Effect of support domain sizes are given for L-shaped beam (case III) in Fig. 6.63 and 6.64. Results of sd values of  $6.5 \times L/100$ ,  $7.5 \times L/100$  and  $8.5 \times L/100$  have similar results and give sufficient accuracy in both Gauss and nodal integrations. However sd values of  $L/10$  and  $1.5 \times L/10$  affect negatively and cause high fluctuations on the results.



Effects of various  $q$  values are given in Fig. 6.65 and 6.66 for bending stress of L-shaped beam. Without  $q$  value of 0.98, all  $q$  values have similar results in both nodal and Gauss integration.

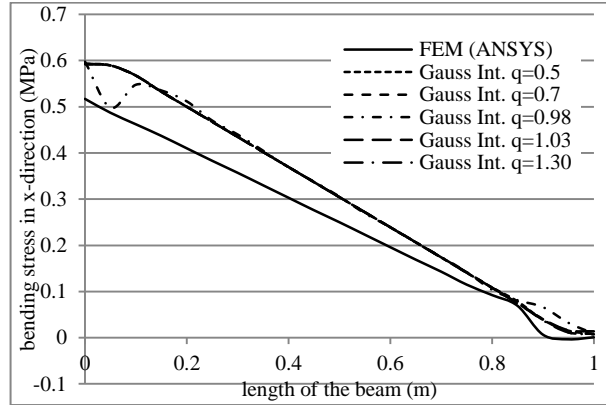


Figure 6.65. Comparison of effect of  $q$  on bending stress results of L-shaped beam (case III) for  $4 \times 4$  Gauss int.

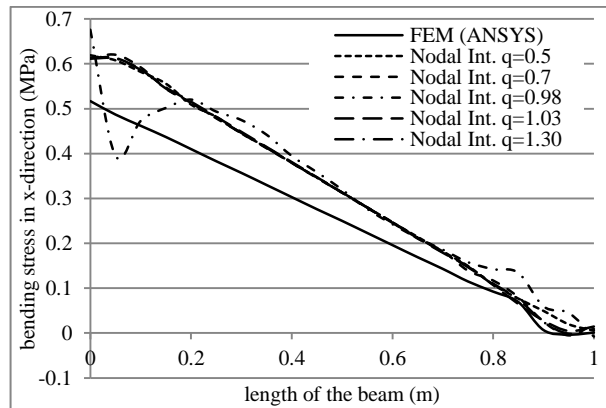


Figure 6.66. Comparison of effect of  $q$  on bending stress results of L-shaped beam (case III) for nodal int. ( $2^{\text{nd}}$  order terms)

### 6.4.2 Results and Discussions

Two different (nodal and Gauss) integration techniques are used for RPIM solutions and their effects on 3D elasto-static solutions are investigated. Taylor series expansion is used in nodal integration and investigated up to  $5^{\text{th}}$  order.  $2 \times 2$ ,  $3 \times 3$  and  $4 \times 4$  numbers of sampling points are used in Gauss integration. In addition, effects of shape parameters and support domain sizes are also investigated. Results are summarized as follows;

- usage of single term of Taylor series expansion in nodal integration cannot satisfy accuracy with respect to analytical solutions

- nodal integrations with 2<sup>nd</sup> and higher order of terms give sufficient results and they have better performance than FEM (ANSYS) in some cases
- 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup> and 5<sup>th</sup> order terms of nodal integration give nearly the same results
- Gauss integration in RPIM also gives good results in all numbers of sampling points
- the recommendation of 2<sup>nd</sup> order terms (Anderson, 1995) in CFD is also valid in the results of this section
- shape parameter  $\alpha_c$  is nearly not effective on Gauss integration results in cases II and III. However when  $\alpha_c$  is equal to 7.00 in nodal integration results, accuracy of the solution vanishes.  $\alpha_c$  values of 7.00 and 9.00 cause failure in construction of shape function in case I
- support domain sizes of  $6.5 \times L/100$ ,  $7.5 \times L/100$  and  $8.5 \times L/100$  of both nodal and Gauss integration give good agreements with analytical and FEM solutions in cases II and III. However, increasing support domain size to  $L/10$  and  $1.5 \times L/10$  cause to decrease in accuracy. sd values of  $8.5 \times L/100$ ,  $L/10$  and  $1.5 \times L/10$  in Gauss integration cannot support construction of shape functions in case I
- shape parameter  $q$  is not effective in Gauss integration. Similar results are also obtained in nodal integration. But when  $q$  is equal to 0.98, accuracy decreases in nodal integration solutions.  $q$  values of 0.5 and 0.98 cause to failures in Gauss integration in case I. When  $q$  is used as a different value from 1.03, the results include high fluctuations

### **6.5 Tetrahedral Shaped Taylor Integration Cells for 3D Cases**

The creation of tetrahedral shaped Taylor integration cells is more flexible and easy than hexahedral shaped Taylor integration cells. They are mentioned in this section.

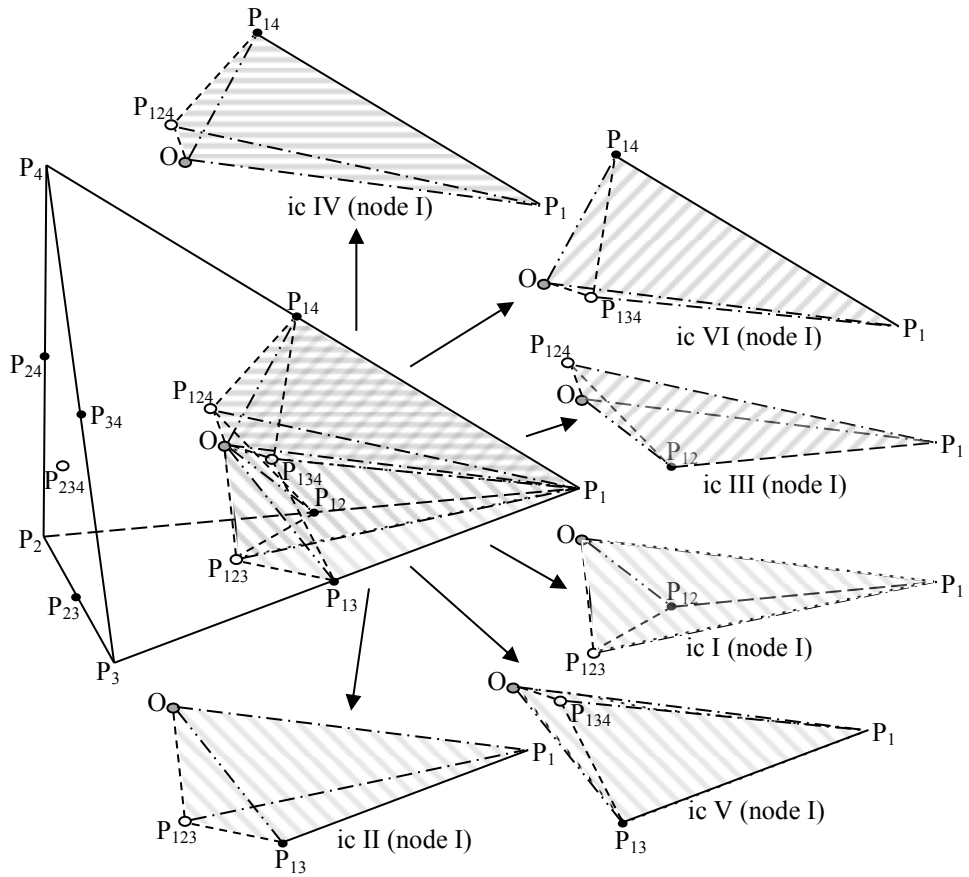


Figure 6.67. Sub-division of tetrahedral background cell for each node

In Fig. 6.67, a tetrahedral background cell is given. The cell which is formed by field nodes of  $P_1 (x_1, y_1, z_1)$ ,  $P_2 (x_2, y_2, z_2)$ ,  $P_3 (x_3, y_3, z_3)$  and  $P_4 (x_4, y_4, z_4)$  is further divided into 6 different integration cells (ic I, II...). So, the total number of integration cells for a field node can be calculated by multiplying 6 by the number of tetrahedral cells that are connected to the field node.  $P_{12}$ ,  $P_{13}$ ,  $P_{14}$ ,  $P_{23}$ ,  $P_{24}$  and  $P_{34}$  are centre point of each side line of tetrahedral cell.  $P_{123}$ ,  $P_{124}$ ,  $P_{134}$  and  $P_{234}$  are centre points of surface areas.  $O$  is the centre point. Subdivision of tetrahedral integration cell components for each node is tabulated in Table 6.9. Subdivision of tetrahedral cell can cause negative volumes (Toron, 2004; Kovalev, 2005). Therefore, the orientation of subdivided cell nodes is placed with an order of preventing negative volume results.

Table 6.9. Components of edges for subdivided Taylor Integration cells for each node (tetrahedral geometry)

	integration cell (ic)	components of edges					integration cell (ic)	components of edges			
Node I (P <sub>1</sub> )	I	P <sub>1</sub>	P <sub>12</sub>	P <sub>123</sub>	O	Node III (P <sub>3</sub> )	I	P <sub>3</sub>	P <sub>13</sub>	P <sub>123</sub>	O
	II	P <sub>1</sub>	P <sub>123</sub>	P <sub>13</sub>	O		II	P <sub>3</sub>	P <sub>123</sub>	P <sub>23</sub>	O
	III	P <sub>1</sub>	P <sub>124</sub>	P <sub>12</sub>	O		III	P <sub>3</sub>	P <sub>134</sub>	P <sub>13</sub>	O
	IV	P <sub>1</sub>	P <sub>14</sub>	P <sub>124</sub>	O		IV	P <sub>3</sub>	P <sub>34</sub>	P <sub>134</sub>	O
	V	P <sub>1</sub>	P <sub>13</sub>	P <sub>134</sub>	O		V	P <sub>3</sub>	P <sub>23</sub>	P <sub>234</sub>	O
	VI	P <sub>1</sub>	P <sub>134</sub>	P <sub>14</sub>	O		VI	P <sub>3</sub>	P <sub>234</sub>	P <sub>34</sub>	O
Node II (P <sub>2</sub> )	I	P <sub>2</sub>	P <sub>123</sub>	P <sub>12</sub>	O	Node IV (P <sub>4</sub> )	I	P <sub>4</sub>	P <sub>124</sub>	P <sub>14</sub>	O
	II	P <sub>2</sub>	P <sub>23</sub>	P <sub>123</sub>	O		II	P <sub>4</sub>	P <sub>24</sub>	P <sub>124</sub>	O
	III	P <sub>2</sub>	P <sub>12</sub>	P <sub>124</sub>	O		III	P <sub>4</sub>	P <sub>14</sub>	P <sub>134</sub>	O
	IV	P <sub>2</sub>	P <sub>124</sub>	P <sub>24</sub>	O		IV	P <sub>4</sub>	P <sub>134</sub>	P <sub>34</sub>	O
	V	P <sub>2</sub>	P <sub>234</sub>	P <sub>23</sub>	O		V	P <sub>4</sub>	P <sub>234</sub>	P <sub>24</sub>	O
	VI	P <sub>2</sub>	P <sub>24</sub>	P <sub>234</sub>	O		VI	P <sub>4</sub>	P <sub>34</sub>	P <sub>234</sub>	O

The first integration;  $\int 1d\Omega$  represents the volume of tetrahedron and can also be calculated as determinant of edge distances (Bhowmick and Shontz, 2012) in Eq. 6.15.

$$Volume = \frac{1}{6} \cdot \begin{vmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{vmatrix} \quad (6.15)$$

Jacobian transformation is used as similar as 2D cases for integration of the terms of  $x, y, z, x^2, \dots$ . In Fig. 6.68, the transformation of tetrahedron geometry from global to natural coordinates is shown for ic II of field node I. P<sub>1</sub>', P<sub>123</sub>', P<sub>13</sub>' and O' represent the transformed natural coordinates from global coordinates of P<sub>1</sub>, P<sub>123</sub>, P<sub>13</sub> and O points, respectively. Also for other integration cells, P<sub>1</sub>', P<sub>123</sub>', P<sub>13</sub>' and O' in Fig. 6.68 correspond to second, third, fourth and fifth columns in Table 6.9, respectively.

The bounds of integral starts from zero to upper natural coordinate. Hence P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub> and P<sub>4</sub> are placed at P<sub>1</sub>' in integral calculations. This condition can provide to no usage of parallel axis theorem to carry integration results to related edge node. Hence, x<sub>1</sub>, y<sub>1</sub> and z<sub>1</sub> in Eq. 6.16, 6.17 and 6.18 are assigned x, y and z coordinates of P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub> and P<sub>4</sub> in related integral node calculation.

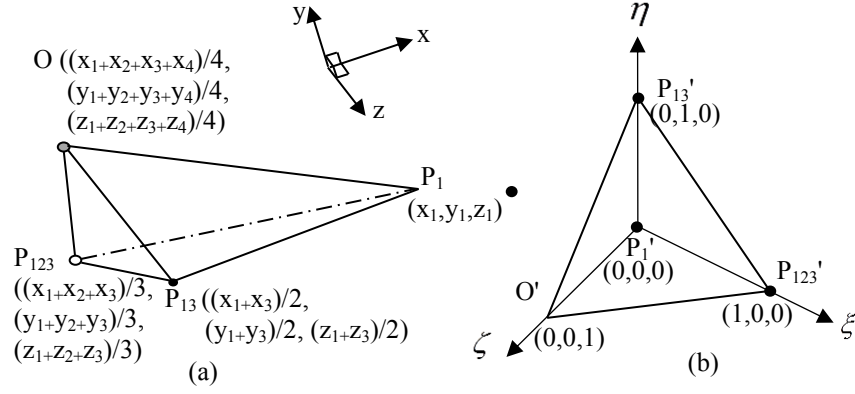


Figure 6.68. Transformation of a tetrahedral integration cell (ic II in Fig.6.67) from global (a) to natural (b) coordinates

The transformation of coordinates is given in Eq. 6.16, 6.17 and 6.18 (Bhowmick and Shontz, 2012).  $\eta$ ,  $\xi$  and  $\zeta$  are natural coordinates.

$$x = x_1 + (x_2 - x_1) \cdot \xi + (x_3 - x_1) \cdot \eta + (x_4 - x_1) \cdot \zeta \quad (6.16)$$

$$y = y_1 + (y_2 - y_1) \cdot \xi + (y_3 - y_1) \cdot \eta + (y_4 - y_1) \cdot \zeta \quad (6.17)$$

$$z = z_1 + (z_2 - z_1) \cdot \xi + (z_3 - z_1) \cdot \eta + (z_4 - z_1) \cdot \zeta \quad (6.18)$$

Determinant of Jacobian (J) transformation matrix are given in Eq. 6.19.

$$DET(J) = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{vmatrix} = 6 \cdot Volume \quad (6.19)$$

The transformation of integration of uniform geometry in natural coordinates is given in Eq. 6.20. Its application to tetrahedron geometry is given in Eq. 6.21. Hence integration in terms of  $x, y, z, x^2, \dots$  can be easily calculated as follows:

$$\int f(x, y, z) d\Omega = \int f[x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta)] * |DET(J)| d\Omega' \quad (6.20)$$

$$\int_D f(x, y, z) dD = |DET(J)| * \int_0^1 d\xi \int_0^{1-\xi} d\eta \int_0^{1-\xi-\eta} f[x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta)] d\zeta \quad (6.21)$$

Even if suitable orientation is used in Table 6.9, in some cases node coordinate orientations can be changed with respect to construction of tetrahedral cells in

Delaunay triangulation technique. For this reason, upper integration bounds in Eq. 6.21 are updated with respect to where the subdivided tetrahedral cell exists, which are given in Table 6.10.

Table 6.10. Direction of tetrahedral cell and used bounds of integral in natural coordinates

	$\Delta x_{12}$	$\Delta y_{13}$	$\Delta z_{14}$	$\int_0^1 f[x, y, z]d\zeta$	$\int_0^1 f[x, y, z]d\eta$	$\int_0^1 f[x, y, z]d\xi$
I	+	+	+	$1-\xi-\eta$	$1-\xi$	+1
II	+	+	-	$-(1-\xi-\eta)$	$1-\xi$	+1
III	+	-	+	$1-\xi+\eta$	$-(1-\xi)$	+1
IV	+	-	-	$-(1-\xi+\eta)$	$-(1-\xi)$	+1
V	-	+	+	$1+\xi-\eta$	$1+\xi$	-1
VI	-	+	-	$-(1+\xi-\eta)$	$1+\xi$	-1
VII	-	-	+	$1+\xi+\eta$	$-(1+\xi)$	-1
VIII	-	-	-	$-(1+\xi+\eta)$	$-(1+\xi)$	-1

Where  $\Delta x_{12}$  is the sign of difference between  $x_2 - x_1$ ,  $\Delta y_{13}$  is the sign of difference between  $y_3 - y_1$  and  $\Delta z_{14}$  is the sign of difference between  $z_4 - z_1$  for each subdivided integration cell. The signs of  $\eta$ ,  $\xi$  and  $\zeta$  in Eq. 6.16, 6.17 and 6.18 is also regulated with respect to signs of  $\Delta x_{12}$ ,  $\Delta y_{13}$  and  $\Delta z_{14}$ .

### 6.5.1 Numerical Examples

Three different case studies are investigated with different boundary conditions by using NI-RPIM. An axial loaded bar problem (case I), a cantilever beam problem (case II) and torsion of cylindrical shaft problem (case III) are examined. Tetrahedral integration cells and subdivision calculations are accomplished with Matlab and effects of Taylor series expansion terms in nodal integration are investigated by using a program that is developed in Fortran.

The used material properties have linear elastic behaviour with a Young's Modulus of 200 GPa and Poisson's ratio of 0.0. The selection of Poisson's ratio as 0.0 is aimed for providing similar conditions as analytical solutions.

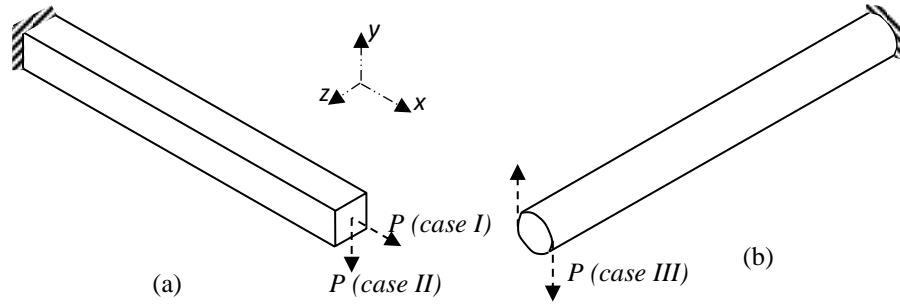


Figure 6.69. The models used in the analyses and boundary conditions, a) an axial loaded bar (case I) b) cantilever beam (case II), c) torsion of cylindrical shaft (case III)

All used model geometries and boundary conditions are illustrated in Fig. 6.69. The used bar/beam in case I and II have a length of 1.0 m with square section of  $0.1\text{m} \times 0.1\text{m}$ . A cylindrical shaft with a length of 1.0 m and a diameter of 0.1 m is used in case III. Applied force ( $P$ ) in case I and II is 20000 N. A force couple is applied for creating a torque, which is equal to 100 Nm in case III. Other end edges of bar, beam and shaft are fixed.

Axial deformation ( $\delta$ ) of bar (case I) is given in Eq. 6.22. Where  $P$  is applied force,  $x$  is distance from fixed support location,  $A$  is cross-sectional area and  $E$  is Young's Modulus.

$$\delta = \frac{P \times x}{A \times E} \quad (6.22)$$

Axial stress ( $\sigma$ ) of bar (case I) is given in Eq. 6.23,

$$\sigma = \frac{P}{A} \quad (6.23)$$

For case II, vertical deflection of cantilever beam (Beer et al., 2009) is given in Eq. 6.24, where  $I$  represents inertia of beam.

$$y = \frac{P \times x^2}{6 \times E \times I} (3 \times L - x) \quad (6.24)$$

The bending stress at upper and lower surfaces of beam is given as;

$$\sigma = \frac{M \times c}{I} \quad (6.25)$$

$c$  is the distance between upper/lower surface and natural axis of the beam. Angle of twist ( $\phi$ ) for torsion (Timoshenko and Goodier, 1951) of cylindrical shaft (case III) is given in Eq. 6.26.

$$\phi = \frac{T \times x}{J \times G} \quad (6.26)$$

where  $T$  is applied torque,  $J$  is polar moment of inertia and  $G$  is shear modulus of shaft.

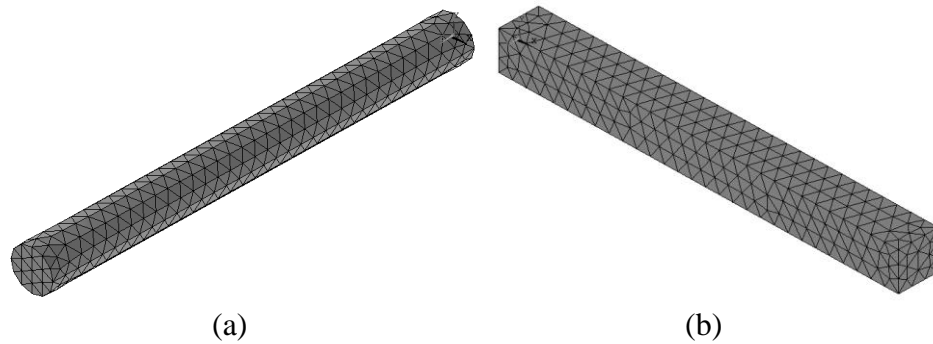


Figure 6.70. FEM models for; a) case I and II, b) case III

FEM models are given in Fig. 6.70, which are constructed with SOLID185 tetrahedral shaped finite elements. ANSYS 14 package program is used in FEM pre-processor, solution and post processor steps.

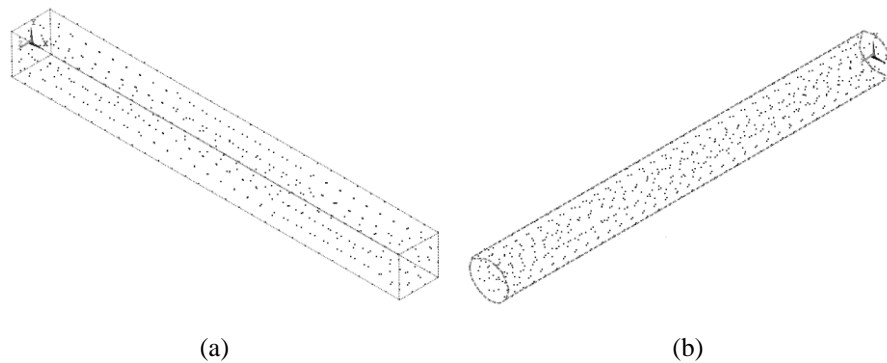


Figure 6.71. Meshfree models of analysed cases for; a) case I and II, b) case III

Meshfree models are shown in Fig. 6.71, which have 489 nodes in case I and II and 659 nodes in case III. Same numbers of nodes are used in FEM. In meshfree solutions,  $\alpha_c$  is used as 3.00, dimensionless support domain size ( $\alpha_s$ ) parameter is used as 1.30 and  $q$  is used as 1.03 as default parameters.



### 6.5.1.1 Axial Loaded Bar

An axial loaded bar problem (case I) is investigated with different orders of Taylor series terms in NI-RPIM. Orders of Taylor series expansion are basically represented in Eq. 5.15, where the term,  $f(x_0)$ , is called 0<sup>th</sup> order. When it includes  $f'(x_0)$ , in Eq. 5.15, it is called 1<sup>st</sup> order and so on.

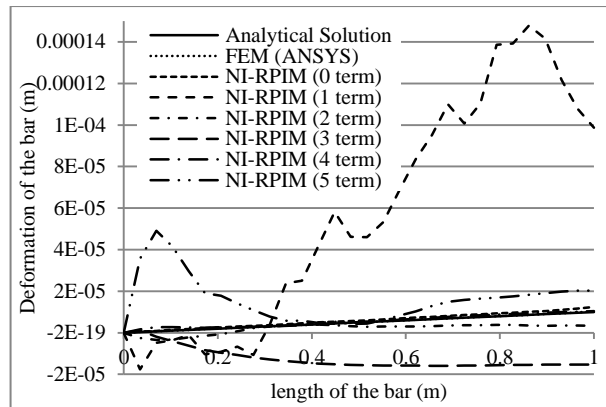


Figure 6.72. Comparison of axial deformation of bar (case I) for analytical, FEM and nodal int. techniques

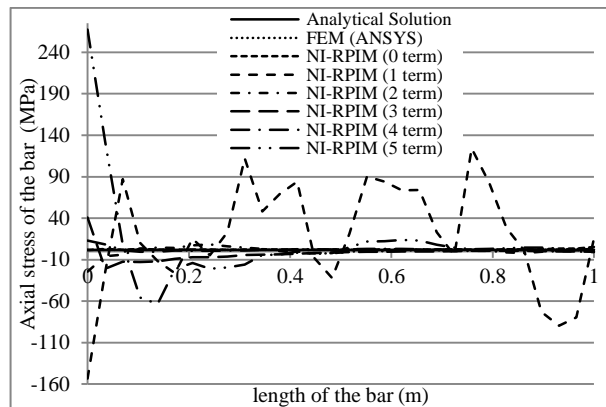


Figure 6.73. Comparison of axial stress of bar (case I) for analytical, FEM and nodal int. techniques

NI-RPIM, analytical and FEM solutions of case I are given for axial deformation in Fig. 6.72 and for axial stress in Fig. 6.73. Deformation and stress results of FEM solution nearly give same results as analytical solution. 0<sup>th</sup> order term of NI-RPIM has similar characteristic results as analytical solution but less close range than FEM results on deformation. However, stress results are same as FEM. 1<sup>st</sup> order term causes the highest fluctuation and the least accurate results in both deformation and stress results. The usage of 2<sup>nd</sup> order terms decreases the consisting fluctuations and difference. 3<sup>rd</sup> order terms have less fluctuation, but decreases solution accuracy. 4<sup>th</sup>

order terms of NI-RPIM results are well achieved same results as analytical solution. Accuracy decreases when 5<sup>th</sup> order terms are used.

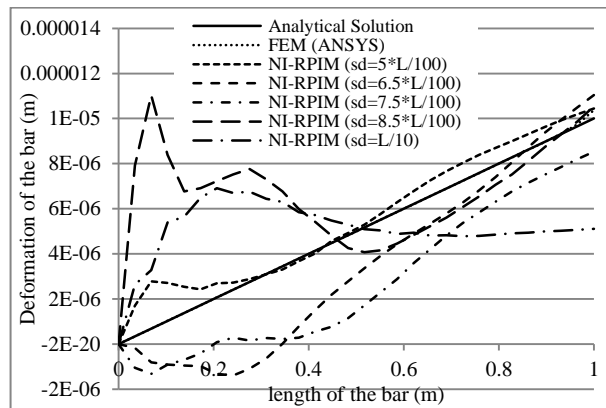


Figure 6.74. Comparison of effect of sd on deformation results of axial loaded bar (case I) for nodal int. (4<sup>th</sup> order terms)

Effect of support domain size (sd) on deformation and stress results of case I are given in Fig. 6.74 and 6.75, respectively. It is observed that increment on support domain size causes to fluctuations and decreases accuracy. This behaviour is similarly mentioned in the study (Liu, 2009), which can be caused from formation of ill-conditioned moment matrix. Hence least, but enough number of nodes must be consisted in the local support domain. Approximately between 14-16 nodes are consisted in sd size of  $5*L/100$ .

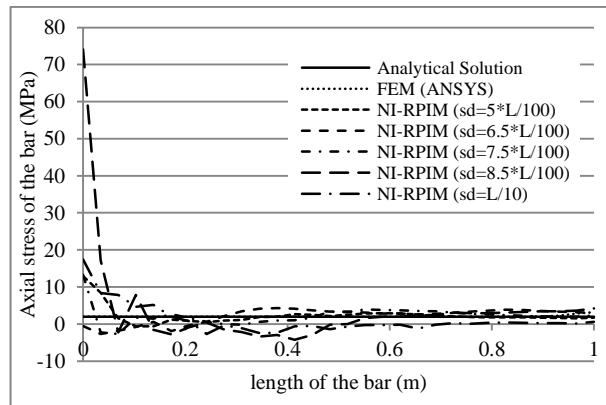


Figure 6.75. Comparison of effect of sd on stress results of axial loaded bar (case I) for nodal int. (4<sup>th</sup> order terms)

Effect of  $\alpha_c$  on case I, which is used inside RBF, is investigated for deformation in Fig. 6.76 and stress in Fig. 6.77.  $\alpha_c$  value of 7.00 has the least accuracy and includes fluctuations.  $\alpha_c$  value of 1.00 also includes less accuracy but less fluctuations. Both  $\alpha_c$  values of 1.00 and 7.00 should not be used.  $\alpha_c$  values of 5.00 and 9.00 has similar

characteristics with analytical solution, but their accuracy is low.  $\alpha_c$  value of 3.00 gives the best suitable results in both deformation and stress.

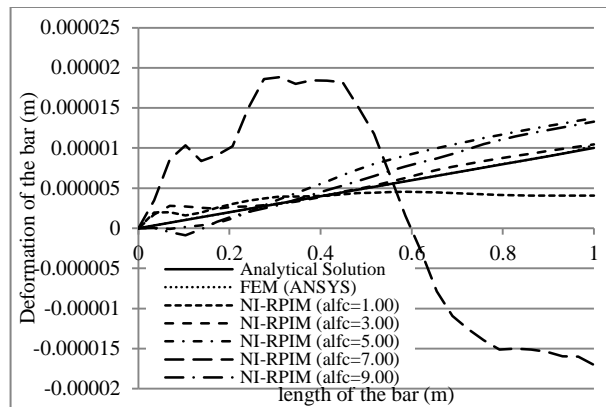


Figure 6.76. Comparison of effect of  $\alpha_c$  (alfc) on deformation results of axial loaded bar (case I) for nodal int. (4<sup>th</sup> order terms)

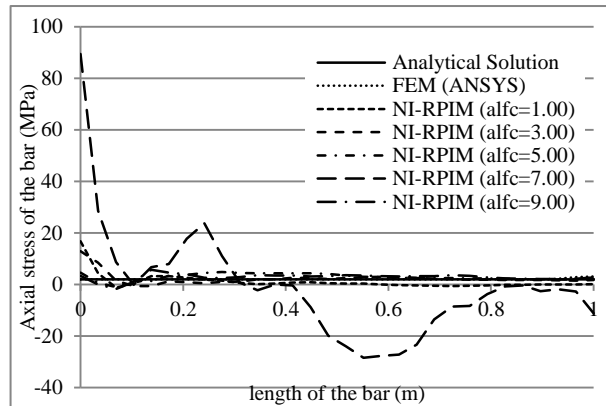


Figure 6.77. Comparison of effect of  $\alpha_c$  (alfc) on stress results of axial loaded bar (case I) for nodal int. (4<sup>th</sup> order terms)

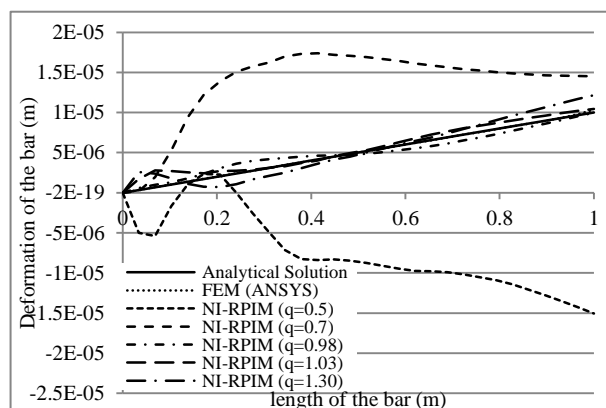


Figure 6.78. Comparison of effect of  $q$  on deformation results of axial loaded bar (case I) for nodal int. (4<sup>th</sup> order terms)

Effect of  $q$  on deformation and stress results of case I is given in Fig. 6.78 and 6.79, respectively.  $q$  value of 0.5 contains errors and is not accurate. Also  $q$  value of 0.7 does not have enough accuracy and contains fluctuations. The best accuracy is achieved with  $q$  values of 0.98 and 1.03.  $q$  value of 1.30 has also enough accuracy but less than  $q$  values of 0.98 and 1.03.

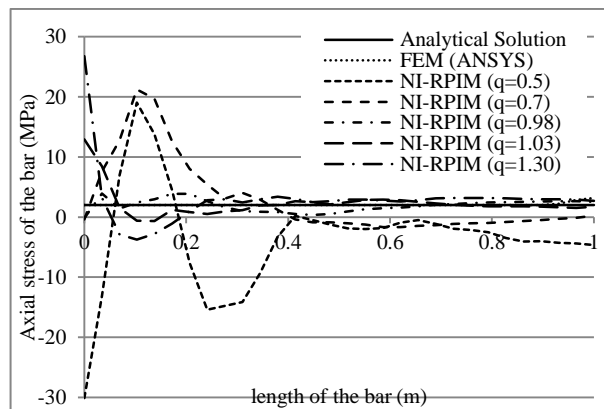


Figure 6.79. Comparison of effect of  $q$  on stress results of axial loaded bar (case I) for nodal int. (4<sup>th</sup> order terms)

### 6.5.1.2 Cantilever Beam

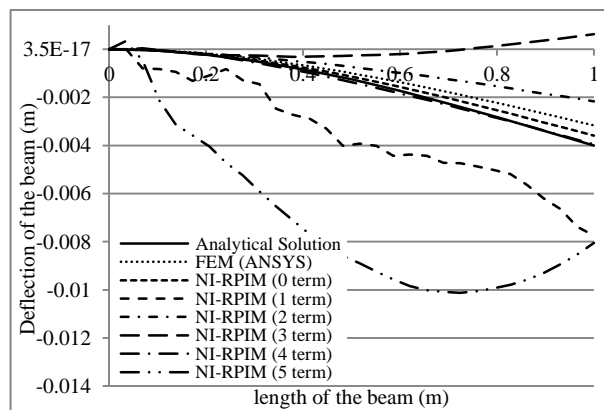


Figure 6.80. Comparison of deflection of cantilever beam (case II) for analytical, FEM and nodal int. techniques

Effects of orders of Taylor terms in NI-RPIM are investigated with analytical and FEM results for case II. Deformation and stress results are given in Fig. 6.80 and 6.81. 0<sup>th</sup> order terms have more accurate deformation and stress results than results of FEM. However, usage of 1<sup>st</sup> order term losses accuracy of solution and fluctuates. 2<sup>nd</sup> order terms provide to decrease fluctuations and errors, but it has less accurate results

than FEM. The usage 3<sup>rd</sup> order terms gets deflection results away from analytical results. However, the usage of 4<sup>th</sup> order terms gives the best results, which is nearly as same deformation results as analytical solution. Accuracy is lost with usage of 5<sup>th</sup> order terms.

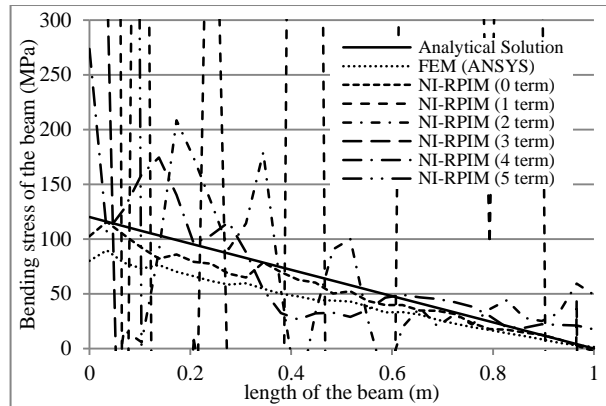


Figure 6.81. Comparison of bending stress results of cantilever beam (case II) for analytical, FEM and nodal int. techniques

Stress results include high fluctuations, even if displacement results are stable. The main reason is the structure of nodal integrations in calculations of strain matrices. This problem is commonly detected in literature. Stabilized conforming technique (Chen et al., 2001) is the one of the method for avoiding these fluctuations. This technique substantially reduce the fluctuations, which is also applied on different point interpolation methods; NS-PIM (Zhang et al., 2007), ES-PIM (Liu and Zhang, 2008) and CS-RPIM (Liu and Zhang, 2009). However, no stabilized conforming/smoothing method is used in this study for NI-RPIM (Liu et al., 2007) and stress fluctuations are tried to be reduced by selecting appropriate Taylor series terms and shape parameters.

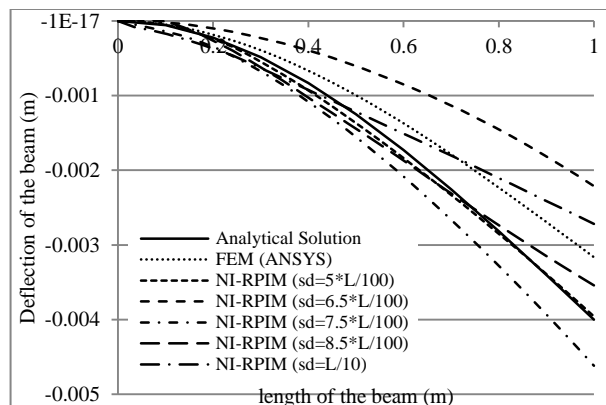


Figure 6.82. Comparison of effect of sd on deflection of cantilever beam (case II) for nodal int. (4<sup>th</sup> order terms)

Effect of support domain size (sd) is given in Fig. 6.82 and 6.83 for deformation and stress results of case II. The best result is achieved, when less support domain size is used. Increment of support domain size decreases accuracy results of deformation and increases fluctuations in stress results.

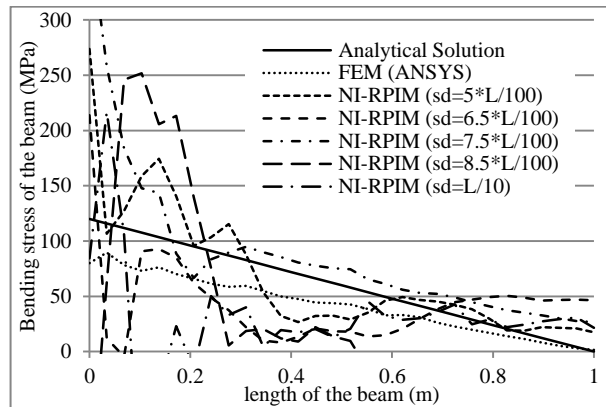


Figure 6.83. Comparison of effect of sd on bending stress results of cantilever beam (case II) for nodal int. (4<sup>th</sup> order terms)

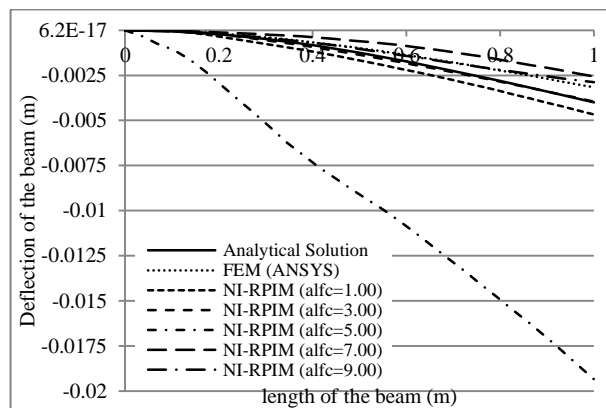


Figure 6.84. Comparison of effect of  $\alpha_c$  (alfc) on deflection results of cantilever beam (case II) for nodal int. (4<sup>th</sup> order terms)

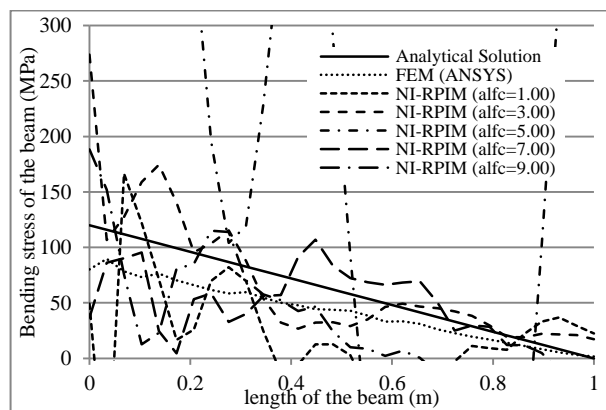


Figure 6.85. Comparison of effect of  $\alpha_c$  (alfc) on bending stress results of cantilever beam (case II) for nodal int. (4<sup>th</sup> order terms)

Effect of  $\alpha_c$  on deformation and stress results of case II is given in Fig. 6.84 and 6.85, respectively.  $\alpha_c$  values of 1.00, 3.00, 7.00 and 9.00 give similar deformation results as analytical and FEM solution.  $\alpha_c$  value of 5.00 has the least accurate deformation results and highly fluctuates in stress results.  $\alpha_c$  value of 1.00 gives good accuracy. The best accuracy is achieved with  $\alpha_c$  value of 3.00.

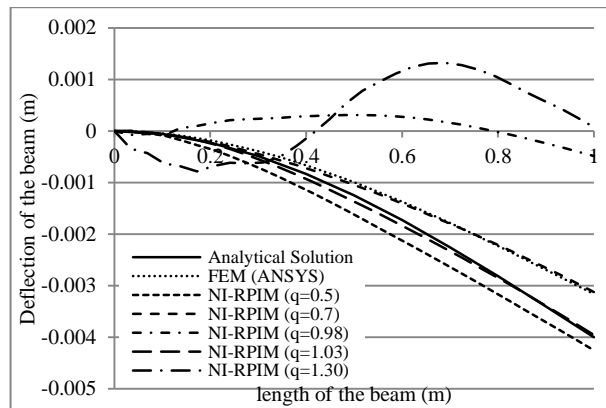


Figure 6.86. Comparison of effect of  $q$  deflection results of cantilever beam (case II) for nodal int. (4<sup>th</sup> order terms)

Effect of  $q$  on deformation and stress results of case II is given in Fig. 6.86 and 6.87.  $q$  values of 0.98 and 1.30 have the least accurate results in deformation results and include high fluctuations in stress results. The best accuracy is achieved with  $q$  value of 1.03.  $q$  of 0.7 has same deformation results with FEM.  $q$  value of 0.5 has also less accurate than  $q$  value of 1.03.

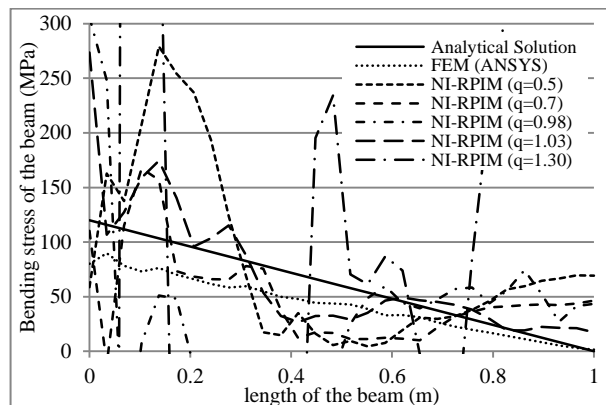


Figure 6.87. Comparison of effect of  $q$  on bending stress results of cantilever beam (case II) for nodal int. (4<sup>th</sup> order terms)

### 6.5.1.3 Torsion of Cylindrical Shaft

Torsion of cylindrical shaft problem (case III) is solved with NI-RPIM method and effects of orders of Taylor terms in are given in Fig. 6.88 for angle of twist results. Results of 0<sup>th</sup> order term have good accuracy with analytical solution, which is similar as FEM results. However, 1<sup>st</sup> order term decreases accuracy and fluctuates. 2<sup>nd</sup> and 3<sup>rd</sup> order terms have also less accurate but fluctuation in the results decreases. The best result is achieved with the usage of 4<sup>th</sup> order terms, which is nearly as same as analytical results. 5<sup>th</sup> order terms decrease accuracy again.

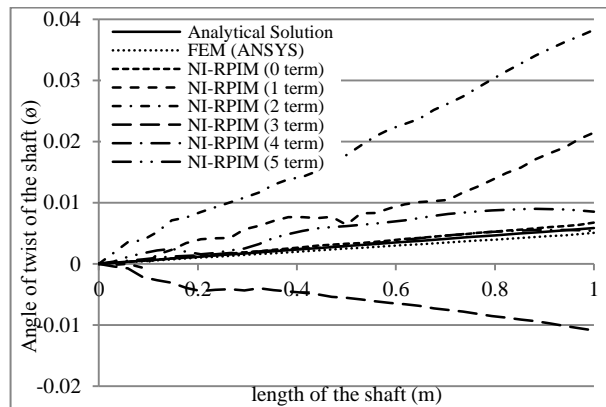


Figure 6.88. Comparison of twist angle results of circular shaft under torsion (case III) for analytical, FEM and nodal int. techniques

Effect of support domain size (sd) is given in Fig. 6.89 for case III. The best accuracy is achieved with support domain size of  $5*L/100$ , which is the smallest support domain size. It includes 14-17 nodes in the local support domain. Support domain size of  $7.5*L/100$  also gives closer results to analytical solution. Increasing the support domain size decreases accuracy.

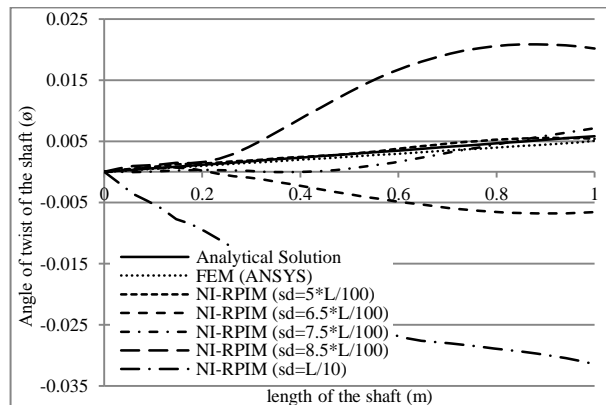




Figure 6.89. Comparison of effect of  $\alpha_c$  on twist angle results of circular shaft under torsion (case III) for nodal int. (4<sup>th</sup> order terms)

$\alpha_c$  values on angle of twist results for case III are given in Fig. 6.90.  $\alpha_c$  values of 1.00 and 7.00 have errors and fluctuations.  $\alpha_c$  value of 5.00 is not suitable in case III, which give negative value results. The best accuracy is achieved with  $\alpha_c$  value of 3.00.

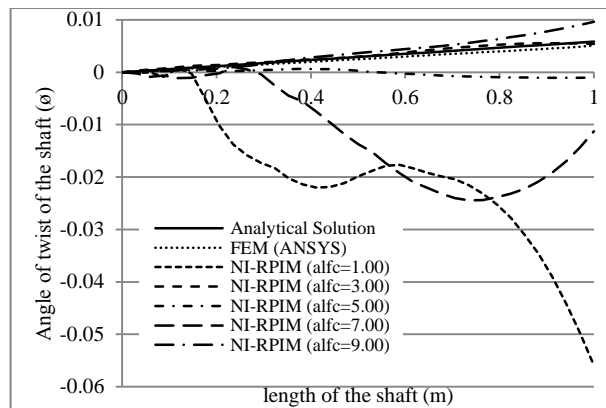


Figure 6.90. Comparison of effect of  $\alpha_c$  (alfc) on twist angle results of circular shaft under torsion (case III) for nodal int. (4<sup>th</sup> order terms)

Effect of  $q$  on angle of twist results are given in Fig. 6.91 for case III. The best results are obtained with  $q$  value of 1.03, which has the closest results with analytical solution. Remaining  $q$  values have less accuracy and cause fluctuations in some locations.

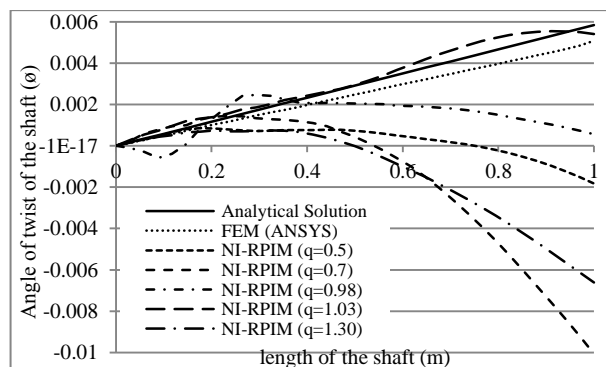


Figure 6.91. Comparison of effect of  $q$  on twist angle results of circular shaft under torsion (case III) for nodal int. (4<sup>th</sup> order terms)

#### 6.5.1.4 Comparison of Solution Times

The CPU time (central processing unit time) of the NI-RPIM is another important parameter and it is obviously related with the solution algorithm. In this study, two different solution algorithms are used: a standard Gauss elimination solver and a linear equation solver that is a part of the Intel Math Kernel Library (MKL) (Intel®, 2014). In Fig. 6.92, results of case II with standard Gauss elimination solver are given for different order Taylor expansion terms with a fixed local support domain size. A computer is used in all solutions, which has a memory of 3 GB and a processor speed of 2.4 GHz. The CPU time increases parabolic when the number of nodes increases. It also increases when the order of Taylor Series terms increases.

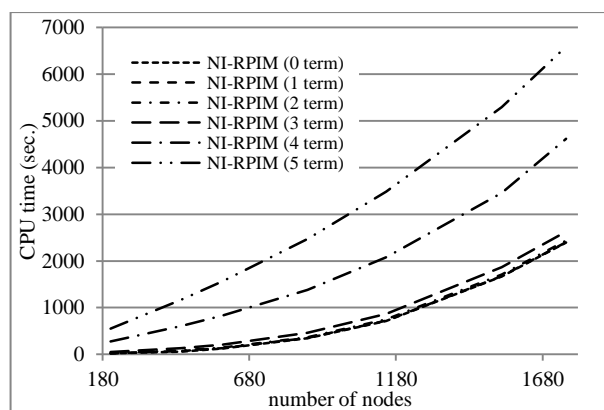


Figure 6.92. Effects of order of Taylor series terms on CPU time against number of nodes with standard Gauss elimination solver

The rate of increment suddenly changes when order becomes greater than 3. The CPU time is seriously decreased when the solver of MKL is used as shown in Fig. 6.93. On the contrary, to standard Gauss elimination solver, CPU time of MKL increases linearly as the number of nodes increases. It is concluded that the usage of higher order terms in NI-RPIM causes using higher order derivatives of radial basis functions and consequently it causes increasing CPU time.

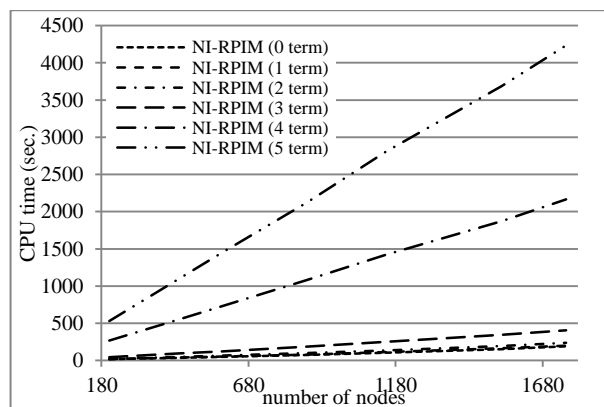


Figure 6.93. Effects of order of Taylor series terms on CPU time against number of nodes with Intel Math Kernel Library (MKL)

CPU time results of NI-RPIM are also compared with a finite element CPU time results (ANSYS) in Fig. 6.94. When the used number of nodes is smaller than 207, ANSYS and NI-RPIM show similar characteristics. However, when it is greater than 207, ANSYS solutions are completed in a short time compared with NI-RPIM solutions. The reason is that ANSYS use sparse direct and pre-condition conjugate gradient (CG) solvers.

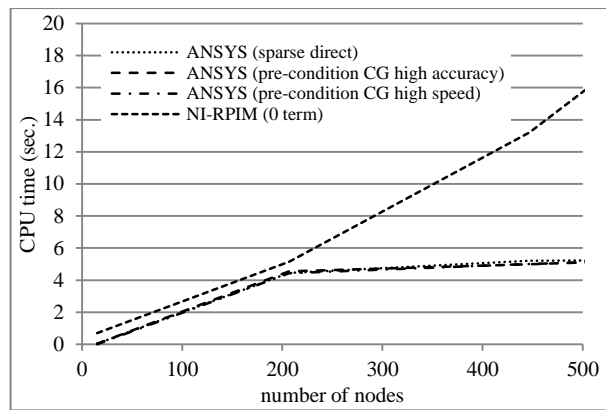


Figure 6.94. Comparison of CPU time of NI-RRIM (0<sup>th</sup> order term) with Intel Math Kernel Library (MKL) and ANSYS solvers

### 6.5.2 Results and Discussions

Tetrahedral background cells are used for the nodal integration of RPIM. Effects of orders of Taylor series terms on solution accuracy of 3D elasto-static problems are investigated. The terms are used up to 5<sup>th</sup> order. Iso-parametric transformation is used for irregular tetrahedral cells for higher order integration calculations in NI-RPIM. Negative volume calculations are prevented with respect to orientation of tetrahedral cells and bounds of integration. Also effect of support domain size and RBF terms ( $\alpha_c$  and  $q$ ) are investigated. Results can be summarized as follows:

- the usage of various orders of Taylor terms directly influences solution results
- the best accuracy in orders of Taylor terms in NI-RPIM results is achieved with 4<sup>th</sup> order, which can capable of giving more accurate results than FEM.
- even order terms of Taylor series expansion generally gives more accurate results than odd order terms of Taylor series expansion in NI-RPIM with tetrahedral background cells

- the usage of small local support domain size (sd) with enough dimension can give less fluctuation and more accurate results
- approximately, 14-16 nodes in case I and case II and 14-17 nodes in case III are used, which can be able to satisfy enough accuracy
- shape parameters of RBF ( $\alpha_c$  and  $q$ ) influences the results and it is recommend that value of 3.00 and value of 1.03 can be used in most of NI-RPIM solutions
- 2<sup>nd</sup> order scheme gives suitable results similarly rather than 1<sup>st</sup> order, which is recommended for CFD analyses by Anderson (1995). It is valid in this section
- beside of accuracy of solution, CPU time is another important section of a numerical analyses and it can be improved by using different solution methods. However, more improvements are required
- stress results include high fluctuations, even if displacement results are stable. The main reason is the structure of nodal integrations in calculations of strain matrices. In literature, different smoothing methods are used to minimize these fluctuations. However, in this section, they are tried to be reduced by selecting appropriate Taylor series terms and shape parameters. This problem will be investigated in detail and tried to be fixed using different algorithms in the future work/sections of this study

Today's computer technology is continuously advancing and high capable computers take place in market. By high-speed computers, complex problems can be solved easily. In numerical methods such as finite element method, the most difficult stage is obtaining a good mesh structure. Generally, analysts spent important amount of times to overcome this stage. New numerical methods such as NI-RPIM, completely overcome the mesh dependency problem and can obtain solutions directly from nodes. 3D laser scanners generally give high-resolution clouds of points for a solid model in a short time. Therefore, in spite of high CPU time disadvantage of NI-RPIM, it can be an important alternative method to other well-known numerical methods.

## CHAPTER 7

### METHODS OF STRESS STABILIZATION

#### 7.1 Introduction

In the previous sections of NI-RPIM results, it is observed that the nodal integration of radial point interpolation method (RPIM) based on Taylor series terms with tetrahedral background cells include some fluctuations in stress results. Therefore, it is attempted to stabilize stresses using average stress distributions for local support domain of each node and least square stress stabilization method in this section.

#### 7.2 Average Stress Distribution for Each Local Support Domain

It is observed in nodal integration that high stress fluctuations occur at far away from application locations of force and boundary conditions. It is not expected that formation of these high fluctuations occur at that conditions with respect to Saint Venant's principle.

For decreasing the fluctuations, average stress distributions are taken for each local support domain. The stresses of nodes in a local support domain of a related node is summed and averaged for number of nodes in that support domain.

#### 7.3 Stiffness Effect on Boundary Locations

In the previous section of NI-RPIM, some stress peak points are observed, especially on the application location of boundary conditions. Hence local stiffness matrix of

nodes on boundary condition locations are changed for decreasing stress peak points. Its effects are investigated.

#### 7.4 Least Square Stabilizations

In order to decrease fluctuations, least square stabilization method (LSS) (Biessel and Belytschko, 1996; Fries and Belytschko, 2007; Duan and Belytschko, 2008) is used in nodal integration scheme. This method adds two equations ( $K^\gamma$  and  $f^\gamma$ ) in Galerkin weak form of governing equation for providing stress stabilization. The main governing equation is previously given in Eq. 4.1/5.1. Addition of stress stabilization equations and detail transformation operations about governing equations can be reached from the study of Duan and Belytschko (2008). The simplified equation is given in Eq. 7.1.  $K^\gamma$  and  $f^\gamma$  equations are given in Eq. 7.2 and 7.3.

$$(K + K^\gamma + \beta K^P)d = (f + f^\gamma + \beta f^P) \quad (7.1)$$

$$K^\gamma = \int_{\Omega} \gamma (L^T DLN)^T (L^T DLN) D \Omega \quad (7.2)$$

$$f^\gamma = - \int_{\Omega} \gamma (L^T DLN)^T b D \Omega \quad (7.3)$$

$\gamma$  is the stabilization parameter and it is given in Eq. 7.4.

$$\gamma = \frac{2\alpha_s l_c^2}{E} \quad (7.4)$$

$\alpha_s$  is the dimensionless stabilization parameter and used as 0.3 in this section.  $l_c$  is used as nodal spacing, which is nearly equal to  $0.04 \cdot L/100$ .

#### 7.5 Numerical Examples

Two different case studies are examined for investigation of NI-RPIM with stress stabilization methods. A 2D axial loaded bar is used in the first case. Its model geometry and meshfree model are given in Fig. 7.1. The model includes 179 irregularly distributed field nodes. The axial loaded bar has a length of 1m and a height of 0.1m. The axial loaded force is applied as 5000 N.

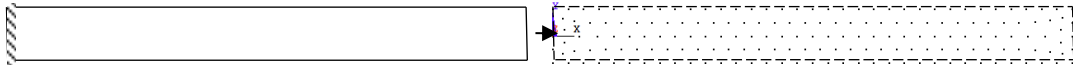


Figure 7.1. The used axial loaded bar model geometry and meshfree model for 2D case analysis

In the 2<sup>nd</sup> case study, a 3D cantilever beam problem is examined for investigation of stress stabilization methods. All the nodal integration of RPIM solutions is mainly prepared with respect to the study of Liu et al. (2007). The solutions of formulations, creation of tetrahedral integration cells and subdivision calculations are accomplished with Matlab.

The geometry and meshfree model is given in Fig. 7.2. The used beam has a length of 1.0 m with square section of 0.1m×0.1m. The used meshfree model has 878 irregularly distributed nodes.

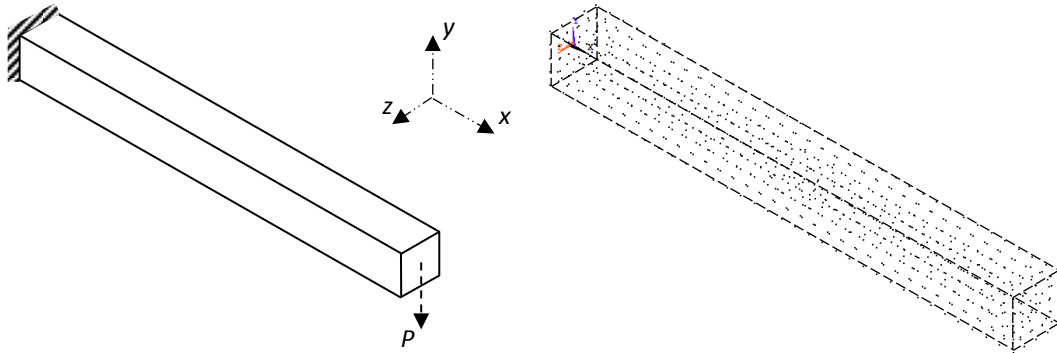


Figure 7.2. The used cantilever beam model geometry and meshfree model for 3D cases

The used material properties have linear elastic behaviour with a Young's Modulus of 200 GPa and Poisson's ratio of 0.0. Applied force (P) is used as 29000 N in cantilever beam problem and other side of force application location is determined as application of boundary condition location. This location is fixed.

Formulation of axial deformation and stress (Beer et al., 2009) of the bar is given in Eq. 7.5 and 7.6.  $A$  represents cross-section area of the bar.

$$\delta = \frac{P \times x}{A \times E} \quad (7.5)$$

$$\sigma = \frac{P}{A} \quad (7.6)$$

Vertical deflection equation (Beer et al., 2009) of cantilever beam is given in Eq. 7.7. Where  $P$  is applied force,  $x$  is distance from fixed support location,  $E$  is Young's Modulus and  $I$  represents inertia of beam.

$$y = \frac{P \times x^2}{6 \times E \times I} (3 \times L - x) \quad (7.7)$$

The bending stress at upper and lower surfaces of beam is given as;

$$\sigma = \frac{M \times c}{I} \quad (7.8)$$

$c$  is the distance between upper/lower surface and natural axis of the beam. In meshfree solutions,  $\alpha_c$  is used as 3.00, dimensionless support domain size ( $\alpha_s$ ) parameter is used as 1.30 and  $q$  is used as 1.03 as default parameters.

## 7.6 Results and Discussions

Axial deformation results of 2D bar are given in Fig 7.3. It is observed that NI-RPIM with 0<sup>th</sup> order term gives suitable results, when they are compared with analytical solution.

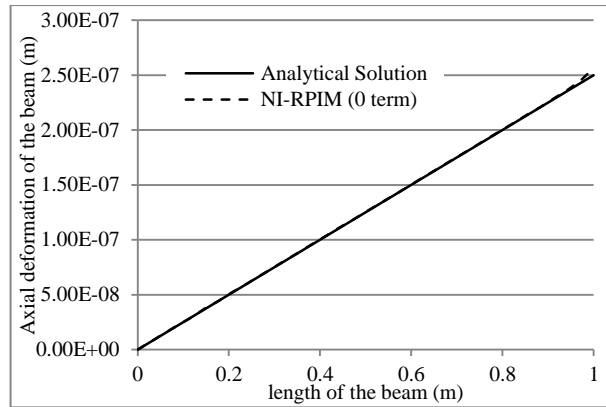


Figure 7.3. Axial deformation results of the bar (2D case) for NI-RPIM with 0th term and analytical solution

Stress results of analytical solution, NI-RPIM with unstabilized and averaged stress distributed (ASD) method are shown in Fig. 7.4. It is observed that the usage of ASD method in stress stabilization decreases fluctuations and high stress peak points at the BC.



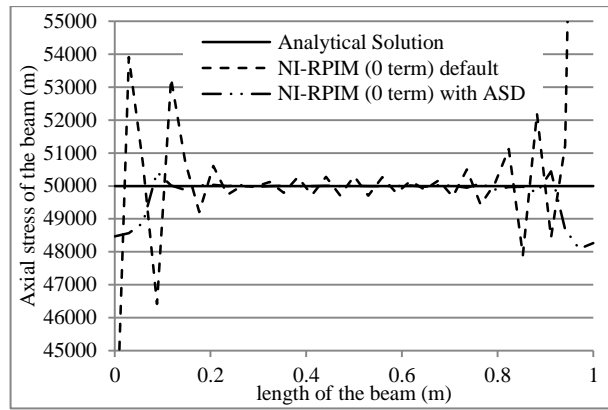


Figure 7.4. Axial stress results of the bar (2D case) for NI-RPIM with 0<sup>th</sup> term and analytical solution

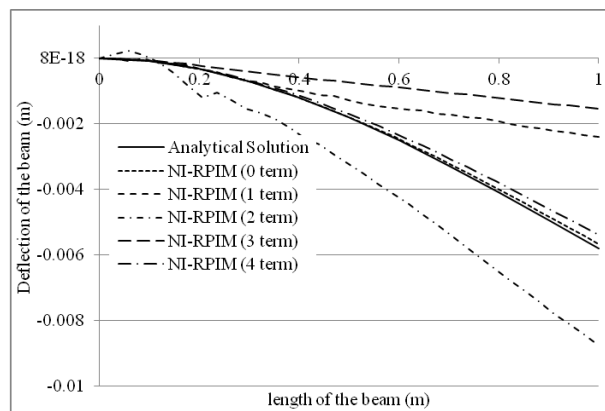


Figure 7.5. Comparison of deflection results of cantilever beam for analytical and pure nodal int. technique

Deflection results of cantilever beam is shown in Fig. 7.5 for NI-RPIM with 0<sup>th</sup>, 1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> order Taylor terms. It is observed that 0<sup>th</sup> and 4<sup>th</sup> order terms give the best results when they are compared with analytical solution. There is no fluctuation occurs on deflection results of these order terms. However the used other terms includes less accuracy and small distortions at the results.

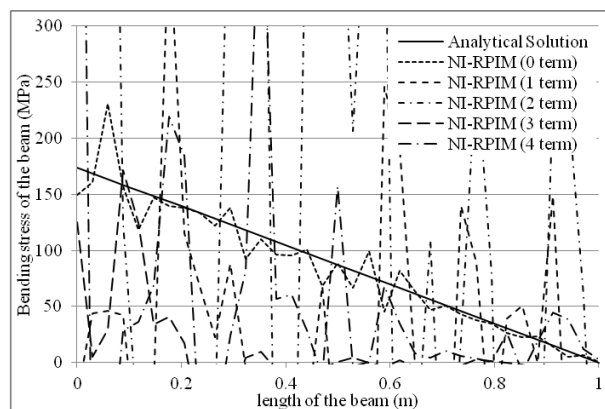


Figure 7.6. Comparison of bending stress results of cantilever beam (3D case) for analytical and pure/unstabilized nodal int. technique

The bending stress results are given in Fig. 7.6 for pure/unstabilized nodal integration, which include highly fluctuation results. The best results achieved with 0<sup>th</sup> order terms, when results of all terms are compared with each other. The stress results of 1<sup>st</sup> and 4<sup>th</sup> order terms include high stress values at the location of boundary conditions.

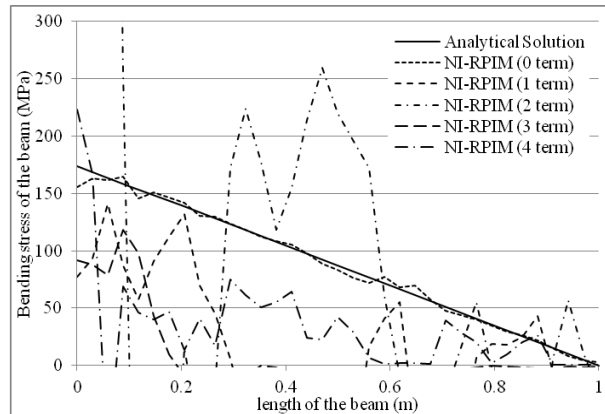


Figure 7.7. Comparison of bending stress results of cantilever beam (3D case) for analytical and nodal int. technique with ASD

Stress results are given in Fig. 7.7, when averaged stress distribution (ASD) of local support domain method is applied for each node. It is observed that fluctuations of stress results are decreased. The best accuracy is achieved with the usage of 0<sup>th</sup> order term. However, results of higher terms without 0<sup>th</sup> order term have poor accuracy and include fluctuations.

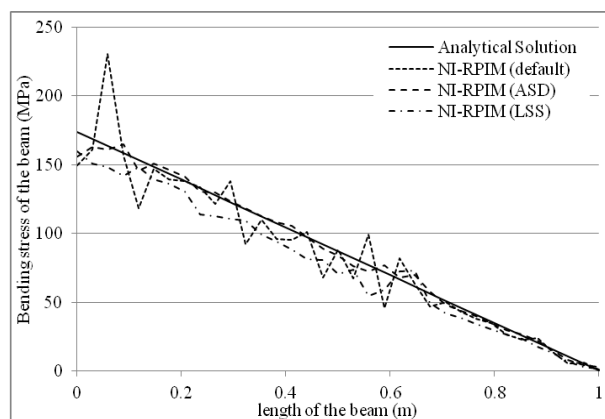


Figure 7.8. Comparison of bending stress results of cantilever beam (3D case) for analytical and nodal int. technique of 0<sup>th</sup> order term with ASD and LSS

In Fig.7.8, 0<sup>th</sup> order term results of NI-RPIM with unstabilized, ASD and LSS methods are compared. It is observed that the usage of both stress stabilization methods decreases fluctuations in the results.

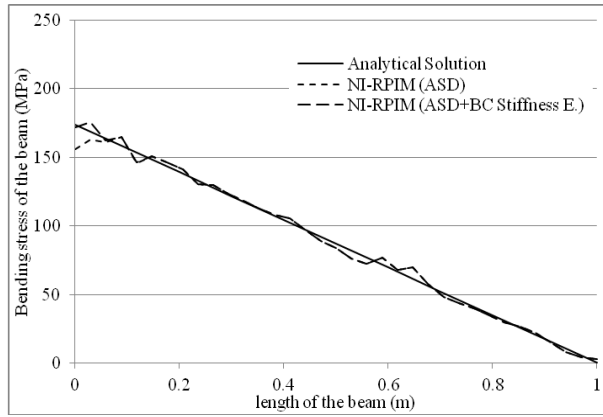


Figure 7.9. Comparison of bending stress results of cantilever beam (3D case) for analytical and nodal int. technique of 0<sup>th</sup> order term with ASD/ASD+BC Stiffness E. The effect of local stiffness matrix changes is given in Fig. 7.9. NI-RPIM results of 0<sup>th</sup> order term have a difference from analytical solution at the application location of boundary conditions (BC). NI-RPIM results have less stress values than analytical solution at BC. Hence, the stiffness values are decreased about %30 at this location for achieving same results at BC.

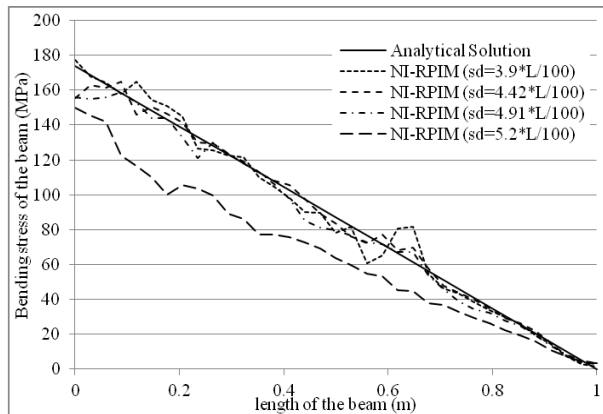


Figure 7.10. Comparison of effect of sd on bending stress results of cantilever beam (3D case) for nodal int. with ASD (0<sup>th</sup> order term) Effect of support domain size is shown for 0<sup>th</sup> order term results of NI-RPIM with ASD in Fig. 7.10. When sd value of 3.9\*L/100 is used, there is high fluctuation occurs about at a beam length of 0.6 m. sd size of 4.42\*L/100 gives better stress results, which is used default sd size in other solutions in this study. The average number of nodes in the local support domains is approximately equal to 44. When sd is increased, the accuracy begins decreasing.

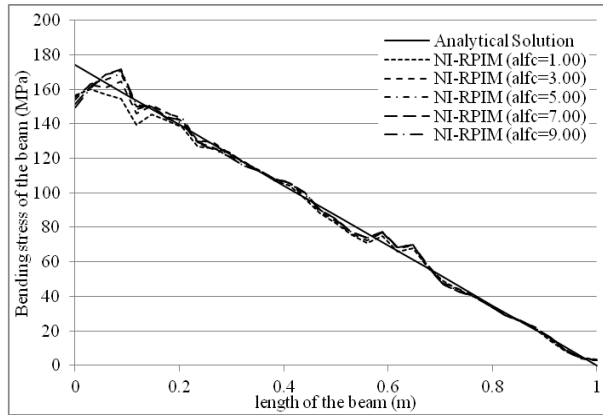


Figure 7.11. Comparison of effect of  $\text{alfc} (\alpha_c)$  on bending stress results of cantilever beam (3D case) for nodal int. with ASD ( $0^{\text{th}}$  order term)

Effect of  $\text{alfc} (\alpha_c)$  results are given in Fig. 7.11. The only dominant effect is observed at BC and its near locations. Different  $\text{alfc} (\alpha_c)$  results have similar characteristics.

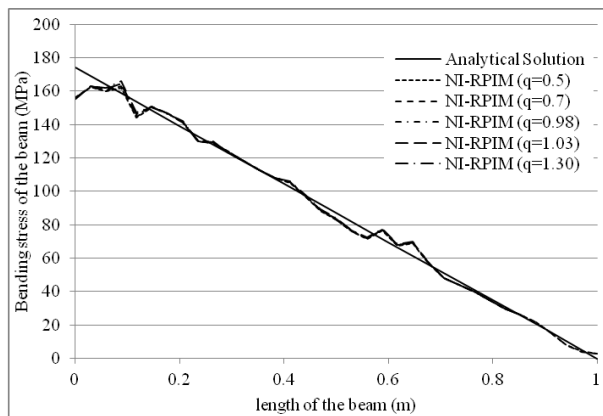


Figure 7.12. Comparison of effect of  $q$  on bending stress results of cantilever beam (3D case) for nodal int. with ASD ( $0^{\text{th}}$  order term)

Effect of  $q$  results is given in Fig. 7.12. Nearly all the stress results of different  $q$  values have same values.

## 7.7 Conclusions

Triangular and tetrahedral background cells are used for the nodal integration of RPIM. The fluctuation problem in stress results of nodal integration method is investigated with application of different methods. Averaged stress distribution (ASD) on local support domain and least square stabilization (LSS) methods are used. Effect of orders of Taylor terms, support domain size and RBF terms are investigated at NI-RPIM with ASD.

Results can be summarized as follows:

- the usage of various orders of Taylor terms directly affects deflection and stress results
- there is no fluctuation on deflection results of 0<sup>th</sup> and 4<sup>th</sup> order terms, which gives good accuracy with respect to analytical solution
- less fluctuated stress values are obtained with the usage of 0<sup>th</sup> order Taylor series terms in NI-RPIM when no stabilization method is used
- fluctuations on stress results are decreased with the usage of ASD and LSS methods
- 0<sup>th</sup> order term results of NI-RPIM with ASD gives the best stress results and less fluctuation values in the solutions
- high stress fluctuations on boundaries where essential boundary conditions are applied can be decreased with changing local stiffness values of corresponding nodes
- support domain size (sd) can affect stress results of NI-RPIM with ASD method
- changes of alfc ( $\alpha_c$ ) with ASD method do not have significant effect on NI-RPIM stress results in the regions where the essential boundary conditions are not included. Different values of q show similar stress distribution characteristics on NI-RPIM with ASD method.

## CONCLUSIONS

The results of literature review shows that meshfree methods need more developments and its applicability in computers must be increased with suitable programming structures. In the first part of this study, an object oriented program is constructed with including different classes and objects for the solution of 2D elasto-static problems by using LRPIM and MLPG methods. Inheritance and encapsulation properties are fully applied for increasing modularity. Some properties are included inside of the program that;

- subclasses use its superclass properties which prevents redefining similar properties again, especially on construction of shape functions, different support domain shapes/sizes, etc..
- different integration cells, new support domain sizes/shapes, shape function methods are easily included with prepared object oriented program
- no properties or assigned value can be changed outside of class

In the second part, a structural program is prepared, which includes nodal integration inside of RPIM for solution of 2D and 3D elasto-static problems. Higher order terms of Taylor series expansion are used. Various integration domain shapes are used for integrations of regular and irregular distributed nodes in 2D/3D problem domains. It is obtained that;

- 2<sup>nd</sup> order Taylor expansion terms can satisfy suitable results in 2D rectangular and 3D hexahedral integration domains
- 4<sup>th</sup> order Taylor expansion terms give the best accuracy in NI-RPIM results for 3D tetrahedral integration domains
- in general, the usage of too large support domain sizes decreases accuracy,
- increasing the used number of nodes in the analysed model generally increases the accuracy of solutions

Optimum shape parameters of RPIM are determined. Generally  $q$  value of 1.03 and  $\alpha_c$  value of 3.00 is recommended in the most of the examined cases.

In the last part, stress fluctuations in nodal integration results have been decreased with three different applications, which provide;

- fluctuations on stress results are decreased with the usage of ASD and LSS methods
- high stress fluctuations on boundaries where essential boundary conditions are applied can be decreased with changing local stiffness values of corresponding nodes

## REFERENCES

- Abdollahifar, A., Nami, M. R., Shafiei, A. R. (2012). A new MLPG method for elasto static problems. *Engineering Analysis with Boundary Elements*. **36**, 451–457.
- Adeli, H., Kao, W. M. (1996). Object-oriented blackboard models for integrated design of steel structures. *Computers & Structures*. **61(3)**, 545-561.
- Al-Assaf, A., Saffarini, H. S. (2004). Optimization of slabs using object oriented programming. *Computers & Structures*. **82**, 741–752.
- Alves, P. D., Barros, F. B., Pitangueira, R. L. S. (2013). An object-oriented approach to the generalized finite element method. *Advances in Engineering Software*. **59**, 1–18.
- Amirani, M. C., Nemati, N. (2011). Simulation of two dimensional unilateral contact using a coupled FE/EFG method. *Engineering Analysis with Boundary Elements*. **35**, 96–104.
- Anacleto, F. E. S., Ribeiro, T. S. A., Ribeiro, G. O., Pitangueira, R. L. S., Penna, S. S. (2013). An object-oriented tridimensional self-regular boundary element method implementation. *Engineering Analysis with Boundary Elements*. **37**, 1276–1284.
- Anderson, J. D. (1995). Computational fluid dynamics, the basics with applications. McGraw-Hill Press, New York.
- Archer, G. C., Fenves, G., Thewalt, C. (1999). A new object-oriented finite element analysis program architecture. *Computers and Structures*. **70**, 63-75.
- Atluri, S. N., Zhu, T. (1998). A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*. **22**, 117-127.
- Babuška, I., Banerjee, U., Osborn, J. E., Zhang, Q. (2009). Effect of numerical integration on meshless methods. *Comput. Methods Appl. Mech. Engrg.* **198**, 2886–2897.
- Babuska, I., Melenk, J. (1996). The partition of the unity finite element method: basic theory and applications. *Nuklearmedizin Nuclear Medicine*. **139**, 198-204.
- Barbieri, E., Meo, M. (2012). A fast object-oriented Matlab implementation of the reproducing kernel particle method. *Comput Mech*. **49**, 581–602.
- Beer, F. P., Johnston-Jr, E. R., Dewolf, J. T., Mazureket, D. (2009). Mechanics of Materials. (5<sup>th</sup> edn.) McGraw-Hill Press, Singapore.
- Belinha, J., Dinis, L. M. J. S. (2007). Nonlinear analysis of plates and laminates using the element free Galerkin method. *Composite Structures*. **78**, 337–350.
- Belytschko, T., Lu, Y. Y., Gu, L. (1994). Element-Free Galerkin methods. *International Journal for Numerical methods in Engineering*. **37**, 229-256.
- Berry, A., Bordat, J., Heggernes, P., Simonet, G., Villanger, Y. (2006). A wide-range algorithm for minimal triangulation from an arbitrary ordering. *Journal of*



*Algorithms*. **58**, 33-66.

Besson, J., Foerch, R. (1997). Large scale object-oriented finite element code design. *Comput. Methods Appl. Mech. Engrg.* **142**, 165-187.

Bettaieb, M. B., Débordes, O., Dogui, A., Duchêne, L., Keller, C. (2012). On the numerical integration of rate independent single crystal behavior at large strain. *International Journal of Plasticity*. **32**, 184–217.

Bhowmick, S., Shontz, S. M. (2012). Towards high-quality, untangled meshes via a force-directed graph embedding approach. *Procedia Computer Science*. **1**, 357–366.

Biessel, S., Belytschko, Y. (1996). Nodal integration of the element-free Galerkin method. *Comput. Methods Appl. Mech. Engrg.* **139**, 49-74.

Bobaru, F., Mukherjee, S. (2001). Shape sensitivity analysis and shape optimization in planar elasticity using the element-free Galerkin method. *Comput. Methods Appl. Mech. Engrg.* **190**, 4319-4337.

Bonet, J., Kulasegaram, S. (2000). Finite increment gradient stabilization of point integrated meshless methods for elliptic equations. *Communications in Numerical Methods in Engineering*. **16(7)**, 475–483.

Bose, A., Carey, G. F. (1999). A class of data structures and object-oriented implementation for finite element methods on distributed memory systems. *Comput. Methods Appl. Mech. Engrg.* **171**, 109-121.

Bozkurt, O. Y., Kanber, B., Asik, M. Z. (2013). Assessment of RPIM shape parameters for solution accuracy of 2D geometrically nonlinear problems. *International Journal of Computational Methods*. **10(3)**, 1-26.

Campbell, J., Vignjevic, R., Libersky, L. (2000). A contact algorithm for smoothed particle hydrodynamics. *Comput. Methods Appl. Mech. Engrg.* **184**, 49-65.

Carey, G., Schmidt, J., Singh, V., Yelton, D. (1994). A prototype scalable, object-oriented finite element solver on multicomputers. *Journal of Parallel and Distributed Computing*. **20**, 357-379.

Cary, J. R., Shasharina, S. G., Cummings, J. C., Reynders, J. V. W., Hinker, P. J. (1997). Comparison of c++ and Fortran 90 for object-oriented scientific programming. *Computer Physics Communications*. **105**, 20-36.

Chen, H., Archer, G. C. (2001). A distributed object-oriented finite element analysis program structure. *Computer-Aided Civil and Infrastructure Engineering*. **16**, 326-336.

Chen, J., Pan, C., Wu, C., Liu, W. K. (1996). Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Comput. Methods Appl. Mech. Engrg.* **139**, 195-227.

Chen, J., Wu, C., Yoon, S., You, Y. (2001). A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*. **50**, 435-466.

Chen, S., Liu, Y., Li, J., Cen, Z. (2011). Performance of the MLPG method for static shakedown analysis for bounded kinematic hardening structures. *European Journal of Mechanics A/Solids*. **30**, 183-194.

- Chen, W., Lin, J., Wang, F. (2011). Regularized meshless method for nonhomogeneous problems. *Engineering Analysis with Boundary Elements*. **35**, 253–257.
- Chen, W., Tanaka, M. (2002). Meshless, integration-free, and boundary-only rbf technique. *Computers and Mathematics with Applications*. **43**, 379-391.
- Cheng, R. J., Liew, K. M. (2012). A meshless analysis of three-dimensional transient heat conduction problems. *Engineering Analysis with Boundary Elements*. **36**, 203–210.
- Cheng, Y. M., Bai, F. N., Peng, M. J. (2014). A novel interpolating element-free Galerkin (IEFG) method for two-dimensional elastoplasticity. *Applied Mathematical Modelling*. **38**, 5187–5197.
- Chung, D. B., Gutierrez, M. A., Borst, R. (2005). Object-oriented stochastic finite element analysis of fibre metal laminates. *Comput. Methods Appl. Mech. Engrg.* **194**, 1427–1446.
- Commend, S., Zimmermann, T. (2001). Object-oriented nonlinear finite element programming: a primer. *Advances in Engineering Software*. **32**, 611-628.
- Cui, X. Y., Feng, H., Li, G. Y., Feng, S. Z. (2015). A cell-based smoothed radial point interpolation method (CS-RPIM) for three-dimensional solids. *Engineering Analysis with Boundary Elements*. **50**, 474–485.
- Dadvand, P., Rossi, R., Oñate, E. (2010). An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Arch Comput Methods Eng.* **17**, 253–297.
- Dai, K.Y., Liu, G. R., Han, X., Li, Y. (2006). Inelastic analysis of 2D solids using a weak-form RPIM based on deformation theory. *Comput. Methods Appl. Mech. Engrg.* **195**, 4179–4193.
- Darbani, M., Ouahsine, A., Villon, P., Naceur, H., Smaoui, H. (2011). Meshless method for shallow water equations with free surface flow. *Applied Mathematics and Computation*. **217**, 5113–5124.
- Demirkaya, G., Soh, C. W., Ilegbusi, O. J. (2008). Direct solution of Navier–Stokes equations by radial basis functions. *Applied Mathematical Modelling*. **32**, 1848–1858.
- Desitter, A., Bates, P. D., Anderson, M. G., Hervouet, J. M. (2000). Development of one, two and three-dimensional finite element groundwater models within a generalized object-oriented framework. *Hydrol. Process.* **14**, 2245-2259.
- Dinis, L. M. J. S., Jorgea, R. M. N., Belinha, J. (2007). Analysis of 3D solids using the natural neighbour radial point interpolation method. *Comput Methods Appl Mech Engrg.* **196**, 2009–2028.
- Dinis, L. M. J. S., Jorgea, R. M. N., Belinha, J. (2008). Analysis of plates and laminates using the natural neighbour radial point interpolation method. *Engineering Analysis with Boundary Elements*. **32**, 267–279.
- Dinis, L. M. J. S., Jorgea, R. M. N., Belinha, J. (2010). A 3D shell-like approach using a natural neighbour meshless method: isotropic and orthotropic thin structures. *Composite Structures*. **92**, 1132–1142.

- Dinis, L. M. J. S., Natal Jorge, R. M., Belinha, J. (2011). A natural neighbour meshless method with a 3D shell-like approach in the dynamic analysis of thin 3D structures. *Thin-Walled Structures*. **49**, 185–196.
- Donning, B. M., Liu, W. K. (1998). Meshless methods for shear-deformable beams and plates. *Comput. Methods Appl. Mech. Engrg.* **152**, 47-71.
- Duan, Q., Belytschko, T. (2008). On the stabilization of stress-point integration in the element free Galerkin method, *Meshfree Methods for Partial Differential Equations IV Lecture Notes in Computational Science and Engineering*. **65**, 47-68.
- Duarte, C. A., Oden, J. T. (1996). H-p clouds - an h-p meshless method. *Numerical Methods for Partial Differential Equations*. **12**, 673-705.
- Dubois-Pelerin, Y., Pegon, P. (1997). Improving modularity in object-oriented finite element programming. *Communications in Numerical Methods in Engineering*. **13**, 193-198.
- Dubois-Pelerin, Y., Pegon, P. (1998). Object-oriented programming in nonlinear finite element analysis. *Computers and Structures*. **67**, 225-241.
- Dubois-Pelerin, Y., Zimmermann, T. (1993). Object-oriented finite element programming: iii. an efficient implementation in C++. *Computer Methods in Applied Mechanics and Engineering*. **108**, 165-183.
- Dubois-Pelerin, Y., Zimmermann, T., Bomme, P. (1992). Object oriented finite element programming: ii. a prototype program in smalltalk. *Computer Methods in Applied Mechanics and Engineering*. **98**, 361-397.
- Eklund, A., Andersson, M., Knutsson, H. (2012). fMRI analysis on the GPU - possibilities and challenges. *Computer Methods and Programs in Biomedicine*. **105**, 145-161.
- Elmer, W., Chen, J. S., Puso, M., Tacioglu, E. (2012). A stable meshfree nodal integration method for nearly incompressible solids. *Finite Elements in Analysis and Design*. **51**, 81–85.
- Ferreira, A. J. M., Roque, C. M. C., Carrera, E., Cinefra, M., Polit, O. (2011). Radial basis functions collocation and a unified formulation for bending, vibration and buckling analysis of laminated plates, according to a variation of Murakami's zig-zag theory. *European Journal of Mechanics A/Solids*. **30**, 559-570.
- Ferreira, A. J. M., Roque, C. M. C., Neves, A. M. A., Jorge, R. M. N., Soares, C. M. M., Liew, K. M. (2011). Buckling and vibration analysis of isotropic and laminated plates by radial basis functions. *Composites: Part B*. **42**, 592–606.
- Filice, L., Alfarob, I., Gagliardi, F., Cueto, E., Micaric, F., Chinestad, F. (2009). A preliminary comparison between finite element and meshless simulations of extrusion. *Journal of Materials Processing Technology*. **209**, 3039–3049.
- Fooladi, M., Golbakhshi, H., Mohammadic, M., Soleimani, A. (2011). An improved meshless method for analyzing the time dependent problems in solid mechanics. *Engineering Analysis with Boundary Elements*. **35**, 1297–1302.
- Forde, B. W. R., Foschi, R. O., Stiemer, S. F. (1990). Object-oriented finite element analysis. *Computers & Structures*. **34(3)**, 355-374.

- Fries, T., Belytschko, T. (2008). Convergence and stabilization of stress-point integration in mesh-free and particle methods. *International Journal for Numerical Methods in Engineering*. **74(7)**, 1067-1087.
- Gato, C. (2010). Meshfree analysis of dynamic fracture in thin-walled structures. *Thin-Walled Structures*. **48**, 215–222.
- Ghorashi, S. S., Mohammadi, S., Sabbagh-Yazdi, S. (2011). Orthotropic enriched element free Galerkin method for fracture analysis of composites. *Engineering Fracture Mechanics*. **78**, 1906–1927.
- Gil, L., Bugeda, G. (2001). A c++ object-oriented programming strategy for the implementation of the finite element sensitivity analysis for a non-linear structural material model. *Advances in Engineering Software*. **32**, 927-935.
- Gingold, R. A., Monaghan, J. J. (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* **181**, 375–389.
- Gong, W., Liu, Y., Zhang, X., Ma, H. (2012). Numerical investigation on dynamical response of aluminum foam subject to hypervelocity impact with material point method. *Cmes-Computer Modeling In Engineering & Sciences*. **83**, 527-545.
- Gu, Y. T., Liu, G. R. (2001). A local point interpolation method for static and dynamic analysis of thin beams. *Comput. Methods Appl. Mech. Engrg.* **190**, 5515-5528.
- Gu, Y. T., Wang, Q. X., Lam, K. Y. (2007). A meshless local kriging method for large deformation analyses. *Comput. Methods Appl. Mech. Engrg.* **196**, 1673–1684.
- Gu, Y. T., Zhang, L. C. (2008). Coupling of the meshfree and finite element methods for determination of the crack tip fields. *Engineering Fracture Mechanics*. **75**, 986–1004.
- Guidic F., Jezequel, J. M., Pacherie, J. L. (1996). An object-oriented framework for supercomputing. *J. Systems Software*. **33**, 239-251.
- Hadjifotinou, K. G. (2002). Numerical integration of the variational equations of satellite orbits. *Planetary and Space Science*. **50**, 361 – 369.
- Halilovic, M., Vrha, M., Stok, B. (2009). Nice - An explicit numerical scheme for efficient integration of nonlinear constitutive equations. *Mathematics and Computers in Simulation*. **80**, 294–313.
- Han, K. (2010). Efficient meshfree analysis using stabilized conforming nodal integration for metal forming simulation. *Journal of the Korean Society of Marine Engineering*. **34(7)**, 943-950.
- Hanoglu, U., Islam, S. U., Sarler, B. (2011). Thermo-mechanical analysis of hot shape rolling of steel by a meshless method. *Procedia Engineering*. **10**, 3173–3178.
- Hao, S., Liu, W. K., Chang, C. T. (2000). Computer implementation of damage models by finite element and meshfree methods. *Comput. Methods Appl. Mech. Engrg.* **187**, 401-440.
- Hashemi, Y., Jahangirian, A. (2011). Implicit fully mesh-less method for compressible viscous flow calculations. *Journal of Computational and Applied Mathematics*. **235**, 4687–4700.

- Hashish, H., Behiry, S. H., El-Shamy, N. A. (2009). Numerical integration using wavelets. *Applied Mathematics and Computation*. **211**, 480–487.
- He, Z., Li, P., Li, Z., Liu, G. (2011). Dispersion and pollution of the improved meshless weighted least-square (IMWLS) solution for the Helmholtz equation. *Engineering Analysis with Boundary Elements*. **35**, 791–801.
- Heng, B. C. P., Mackie, R. I. (2009). Using design patterns in object-oriented finite element programming. *Computers & Structures*. **87**, 952–961.
- Hu, D. A., Long, S. Y., Han, X., Li, G. Y. (2007). A meshless local Petrov–Galerkin method for large deformation contact analysis of elastomers. *Engineering Analysis with Boundary Elements*. **31**, 657–666.
- Hu, D. A., Long, S. Y., Liu, K. Y., Li, G. Y. (2006). A modified meshless local Petrov–Galerkin method to elasticity problems in computer modeling and simulation. *Engineering Analysis with Boundary Elements*. **30**, 399–404.
- Intel® Math Kernel Library Reference Manual. (2014). Document Number: 630813-065US.
- Islam, M. D., Hossain, M. A. (2009). Numerical integrations over an arbitrary quadrilateral region. *Applied Mathematics and Computation*. **210**, 515–524.
- Islam, S., Šarler, B., Vertnik, R., Kosec, G. (2012). Radial basis function collocation method for the numerical solution of the two-dimensional transient nonlinear coupled Burgers’ equations. *Applied Mathematical Modelling*. **36**, 1148-1160.
- Johnson, G. R., Beissel, S. R., Gerlach, C. A. (2011). Another approach to a hybrid particle-finite element algorithm for high-velocity impact. *International Journal of Impact Engineering*. **38**, 397-405.
- Kanber, B., Bozkurt, O. Y. (2008). A diagonal offset algorithm for the polynomial point interpolation method. *Communications in Numerical Methods in Engineering*. **24**, 1909-1922.
- Kanber, B., Bozkurt, O. Y., Erklig, A. (2013). Investigation of RPIM shape parameter effects on the solution accuracy of 2D elastoplastic problems. *International Journal for Computational Methods in Engineering Science and Mechanics*. **14**, 354–366.
- Khosravifard, A., Hematiyan, M. R., Marin, L. (2011). Nonlinear transient heat conduction analysis of functionally graded materials in the presence of heat sources using an improved meshless radial point interpolation method. *Applied Mathematical Modelling*. **35**, 4157–4174.
- Kindelan, M., Bernal, F., González-Rodríguez, P., Moscoso, M. (2010). Application of the RBF meshless method to the solution of the radiative transport equation. *Journal of Computational Physics*. **229**, 1897–1908.
- Kong, X. A. (1996). A data design approach for object-oriented fem programs. *Computers & Structures*. **61(3)**, 503-513.
- Kong, X. A., Chen, D. P. (1995). An object-oriented design of fem programs. *Computers & Structures*. **57(1)**, 157-166.
- Kovalev, K. (2005). Unstructured hexahedral non-conformal mesh generation. PhD thesis, Faculty of Engineering Vrije Universiteit Brussel, Belgium.

- Kromer, V., Dufosse, F., Gueury, M. (2004). An object-oriented design of a finite element code: application to multibody systems analysis. *Advances in Engineering Software*. **35**, 273–287.
- Kromer, V., Dufosse, F., Gueury, M. (2005). On the implementation of object-oriented philosophy for the design of a finite element code dedicated to multibody systems. *Finite Elements in Analysis and Design*. **41**, 493–520.
- Krysl, P., Belytschko, T. (2001). Esflib: a library to compute the element free Galerkin shape functions. *Comput. Methods Appl. Mech. Engrg.* **190**, 2181–2205.
- Kwon, K., Park, S., Youn, S. (2006). The support integration scheme in the least-squares mesh-free method. *Finite Elements in Analysis and Design*. **43**, 127–144.
- Kwon, K., Youn, S. (2006). The least-squares meshfree method for rigid-plasticity with frictional contact. *International Journal of Solids and Structures*. **43**, 7450–7481.
- Lage, C. (1998). The application of object-oriented methods to boundary elements. *Comput. Methods Appl. Mech. Engrg.* **157**, 205–213.
- Lages, E. N., Paulino, G. H., Menezes, I. F. M., Silva, R. R. (1999). Nonlinear finite element analysis using an object-oriented philosophy - application to beam elements and to the cosserat continuum. *Engineering with Computers*. **15**, 73–89.
- Lee, S. K., Kim, H. J., Song, Y., Lee, C. (2009). MT2DInvMatlab - a program in Matlab and Fortran for two-dimensional magnetotelluric inversion. *Computers & Geosciences*. **35**, 1722–1734.
- Lejeune, A., Béchet, F., Boudaoud, H., Mathieu, N., Potier-Ferry, M. (2012). Object-oriented design to automate a high order non-linear solver based on asymptotic numerical method. *Advances in Engineering Software*. **48**, 70–88.
- Leontyev, V. A. (2010). Direct time integration algorithm with controllable numerical dissipation for structural dynamics: Two-step Lambda method. *Applied Numerical Mathematics*. **60**, 277–292.
- Li, L. Y., Wen, P. H., Aliabadi, M. H. (2011). Meshfree modeling and homogenization of 3D orthogonal woven composites. *Composites Science and Technology*. **71**, 1777–1788.
- Li, Q., Soric, J., Jarak, T., Atluri, S. N. (2005). A locking-free meshless local Petrov–Galerkin formulation for thick and thin plates. *Journal of Computational Physics*. **208**, 116–133.
- Li, S., Liu, W. K., Rosakis, A. J., Belytschko, T., Hao, W. (2002). Meshfree Galerkin simulations of dynamic shear band propagation and failure mode transition. *International Journal of Solids and Structures*. **39**, 1213–1240.
- Li, S., Qian, D., Liu, W. K., Belytschko, T. (2001). A meshfree contact detection algorithm. *Comput. Methods Appl. Mech. Engrg.* **190**, 3271–3292.
- Li, X. (2011). Development of a meshless Galerkin boundary node method for viscous fluid flows. *Mathematics and Computers in Simulation*. **82**, 258–280.
- Liana, Y. P., Zhanga, X., Liua, Y. (2011). Coupling of finite element method with material point method by local multi-mesh contact method. *Comput. Methods Appl. Mech. Engrg.* **200**, 3482–3494.

- Liew, K. M., Huang, Y. Q. (2003). Bending and buckling of thick symmetric rectangular laminates using the moving least-squares differential quadrature method. *International Journal of Mechanical Sciences*. **45**, 95–114.
- Liew, K. M., Huang, Y. Q., Reddy, J. N. (2004). Analysis of general shaped thin plates by the moving least-squares differential quadrature method. *Finite Elements in Analysis and Design*. **40**, 1453–1474.
- Liew, K. M., Ng, T. Y., Wu, Y. C. (2002). Meshfree method for large deformation analysis—a reproducing kernel particle approach. *Engineering Structures*. **24**, 543–551.
- Liu, G. R. (2002). A point assembly method for stress analysis for two-dimensional solids. *International Journal of Solids and Structures*. **39**, 261–276.
- Liu, G. R. (2009). *Meshfree Methods: Moving Beyond the Finite Element Method*. (2<sup>nd</sup> edn.) CRC Press.
- Liu, G. R., Gu, Y. T. (2001). A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering*. **50**, 937–951.
- Liu, G. R., Gu, Y. T. (2002). Comparisons of two meshfree local point interpolation methods for structural analyses. *Computational Mechanics*. **29(2)**, 107–121.
- Liu, G. R., Gu, Y. T. (2004). Boundary meshfree methods based on the boundary point interpolation methods. *Engineering Analysis with Boundary Elements*. **28**, 475–487.
- Liu, G. R., Gu, Y. T. (2005). *An Introduction to Meshfree Methods and Their Programming*. Springer, Berlin.
- Liu, G. R., Jiang, Y., Chen, L., Zhang, G. Y., Zhang, Y. W. (2011). A singular cell-based smoothed radial point interpolation method for fracture problems. *Computers and Structures*. **89**, 1378–1396.
- Liu, G. R., Kee, B. B. T., Chun, L. (2006). A stabilized least-squares radial point collocation method (LS-RPCM) for adaptive analysis. *Comput. Methods Appl. Mech. Engrg.* **195**, 4843–4861.
- Liu, G. R., Zhang, G. Y. (2008). Edge-based smoothed point interpolation methods. *International Journal of Computational Methods*. **5(4)**, 621–646.
- Liu, G. R., Zhang, G. Y. (2009). A normed G space and weakened weak (W2) formulation of cell-based smoothed point interpolation method. *International Journal of Computational Methods*. **6(1)**, 147–179.
- Liu, G. R., Zhang, G. Y., Dai, K. Y., Wang, Y. Y., Zhong, Z. H., Li, G. Y., Han, X. (2005). A linearly conforming point interpolation method (LC-PIM) for 2D solid mechanics problems. *International Journal of Computational Methods*. **2(4)**, 645–665.
- Liu, G. R., Zhang, G. Y., Wang, Y. Y., Zhong, Z. H., Li, G. Y., Han, X. (2007). A nodal integration technique for meshfree radial point interpolation method (NI-RPIM). *International Journal of Solids and Structures*. **44**, 3840–3860.
- Liu, G. R., Zhao, X., Dai, K.Y., Zhong, Z. H., Li, G. Y., Han, X. (2008). Static and free vibration analysis of laminated composite plates using the conforming radial point interpolation method. *Composites Science and Technology*. **68**, 354–366.

- Liu, H., Xing, Z., Sun, Z., Bao, J. (2011). Adaptive multiple scale meshless simulation on springback analysis in sheet metal forming. *Engineering Analysis with Boundary Elements*. **35**, 436–451.
- Liu, J., Lin, I., Shih, M., Chen, R., Hsieh, M. (1996). Object-oriented programming of adaptive finite element and finite volume methods. *Applied Numerical Mathematics*. **21**, 439-467.
- Liu, L., Liu, G. R., Tan, V. B. C. (2002). Element free method for static and free vibration analysis of spatial thin shell structures. *Comput. Methods Appl. Mech. Engrg.* **191**, 5923–5942.
- Liu, W. K., Hao, S., Belytschko, T., Li, S., Chang, C. T. (1999). Multiple scale meshfree methods for damage fracture and localization. *Computational Materials Science*. **16**, 197-205.
- Liu, W. K., Jun, S., Zhang, Y. F. (1995). Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*. **20**, 1081-1106.
- Liu, W. K., Ong, J. S., Uras, R. A. (1985). Finite element stabilization matrices - a unification approach. *Computer Methods in Applied Mechanics and Engineering*. **53(1)**, 13–46.
- Liu, W., Li, S., Belytschko, T. (1997). Moving least-square reproducing kernel methods (I) methodology and convergence. *Comput. Methods Appl. Mech. Engrg.* **143**, 113-154.
- Lorenzini, R., Passoni, L. (1999). Test of numerical methods for the integration of kinetic equations in tropospheric chemistry. *Computer Physics Communications*. **117**, 241-249.
- Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *Astron. J.* **82**, 013–1024.
- Ma, S., Zhang, X., Qiu, X. M. (2002). Comparison study of MPM and SPH in modeling hypervelocity impact problems. *International Journal of Impact Engineering*. **36**, 272–282.
- Ma, Y., Feng, W. (2002). Object-oriented finite element analysis and programming in VC++. *Applied Mathematics and Mechanics*. **23(12)**, 1437- 1443.
- Mackie, R. I. (1998). An object-oriented approach to fully interactive finite element software. *Advances in Engineering Software*. **29(2)**, 139-149.
- Mackie, R. I. (1999). Object-oriented finite element programming-the importance of data modelling. *Advances in Engineering Software*. **30**, 775–782.
- Mackie, R. I. (2000). An object-oriented approach to calculation control in finite element programs. *Computers & Structures*. **77**, 461-474.
- Mackie, R. I. (2002). Using objects to handle calculation control in finite element modelling. *Computers & Structures*. **80**, 2001–2009.
- Mackie, R. J. (1992). Object oriented programming of the finite element method. *International Journal for Numerical Methods in Engineering*. **35**, 425-436.
- Marczak, R. J. (2004). An object-oriented programming framework for boundary integral equation methods. *Computers & Structures*. **82**, 1237–1257.



- Marczak, R. J. (2006). Object-oriented numerical integration-a template scheme for fem and bem applications. *Advances in Engineering Software*. **37**, 172–183.
- Masserey, A., Rappaz, J., Rozsnyo, R., Swierkosz, M. (2005). Numerical integration of the three-dimensional Green kernel for an electromagnetic problem. *Journal of Computational Physics*. **205**, 48–71.
- Masters, I., Usmani, A. S., Cross, J. T., Lewis, R. W. (1997). Finite element analysis of solidification using object-oriented and parallel techniques. *International Journal for Numerical Methods in Engineering*. **40**, 2891-2909.
- Menetrey, P., Zimmermann, T. (1994). Object-oriented non-linear finite element analysis: application to j2 plasticity. *Computers & Structures*. **49(5)**, 767-177.
- Monegato, G., Scuderi, L. (1999). Numerical integration of functions with boundary singularities. *Journal of Computational and Applied Mathematics*. **112**, 201-214.
- Moosavi, M. R., Delfanian, F., Khelil, A. (2011). Orthogonal meshless finite volume method in elasticity. *Thin-Walled Structures*. **49**, 708–712.
- Moreno, R., Ramaswamy, B. (2003). Object-oriented implementation of the Galerkin finite element method and its application to the numerical study of natural convective flows in enclosures. *International Journal of Computational Fluid Dynamics*. **17(1)**, 39–60.
- Morgan, D., Henda, R. (2006). Object-oriented thermal radiation simulator. *Education for Chemical Engineers*. **1**, 101–115.
- Mukunda, G. R., Sotelino, E. D., Hsieh, S. (1998). Distributed finite element computations using object-oriented techniques. *Engineering with Computers*. **14**, 59-72.
- Munthe, O., Langtangen, H. P. (2000). Finite elements and object-oriented implementation techniques in computational fluid dynamics. *Comput. Methods Appl. Mech. Engrg*. **190**, 865-888.
- Nagashima, T. (1999). Node-by-node meshless approach and its applications to structural analysis. *Int. J. Numer. Meth. Engrg*. **46(3)**, 341–385.
- Nayroles, G. T., Pierre V. P. (1992). Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*. **10**, 307-318.
- Neto, D. P., Ferreira, M. D. C., Proença, S. P. B. (2013). An object-oriented class design for the generalized finite element method programming. *Latin American Journal of Solid and Structures*. **10**, 1267-1291.
- Nguyen-Van, H., Duy, N. M., Tran-Cong, T. (2007). A simple and accurate four-node quadrilateral element using stabilized nodal integration for laminated plates, *CMC*. **6(3)**, 159-175.
- Nianfei, G., Guangyao, L., Shuyao, L. (2009). 3D adaptive RKPM method for contact problems with elastic–plastic dynamic large deformation. *Engineering Analysis with Boundary Elements*. **33**, 1211–1222.
- Nie, J. H., Hopkins, D. A., Chen, Y. T., Hsieh, H. T. (2010). Development of an object-oriented finite element program with adaptive mesh refinement for multi-physics applications. *Advances in Engineering Software*. **41**, 569–579.

- Niekamp, R., Stein, E. (2002). An object-oriented approach for parallel two and three dimensional adaptive finite element computations. *Computers & Structures*. **80**, 317–328.
- Nikishkov, G. P. (2006). Object oriented design of a finite element code in Java. *Computer Modeling in Engng and Sciences*. **11**, 81-90.
- Noguchia, H., Kawashima, T. (2004). Meshfree analyses of cable-reinforced membrane structures by ALE–EFG method. *Engineering Analysis with Boundary Elements*. **28**, 443–451.
- Ochiai, Y. (2003). Multidimensional numerical integration for meshless BEM. *Engineering Analysis with Boundary Elements*. **27**, 241–249.
- Ohtsubo, H., Kawamura, Y. (1993). Development of the object-oriented finite element modeling system - modify. *Engineering with Computers*. **9**, 187-197.
- Onate, E., Perazzo, F., Miquel, J. (2001). A finite point method for elasticity problems. *Computer & Structures*. **79**, 2151-2163.
- Pantale, O. (2002). An object-oriented programming of an explicit dynamics code: application to impact simulation. *Advances in Engineering Software*. **33**, 297–306.
- Pantale, O., Caperaa, S., Rakotomalala, R. (2004). Development of an object-oriented finite element program: application to metal-forming and impact simulations. *Journal of Computational and Applied Mathematics*. **168**, 341–351.
- Pathak, H., Singh, A., Singh, I. V. (2014). Simulation of 3-D thermo-elastic fracture problems using coupled FE-EFG approach. *Procedia Materials Science*. **6**, 1927-1935.
- Patzak, B., Bittnar, Z. (2001). Design of object oriented finite element code. *Advances in Engineering Software*. **32**, 759-767.
- Patzák, B., Rypl, D. (2012). Object-oriented, parallel finite element framework with dynamic load balancing. *Advances in Engineering Software*. **47**, 35–50.
- Peskin, A. P., Hardin, G. R. (1996). An object-oriented approach to general purpose fluid dynamics software. *Computers chem. Engng*. **20(8)**, 1043-1058.
- Peters, B., Dziugys, A. (2002). Numerical simulation of the motion of granular material using object-oriented techniques. *Comput. Methods Appl. Mech. Engrg*. **191**, 1983–2007.
- Phongthanapanich, S., Dechaumphai, P. (2006). EasyFEM-An object-oriented graphics interface finite element/finite volume software. *Advances in Engineering Software*. **37**, 797–804.
- Ponthot, J. P., Belytschko, T. (1998). Arbitrary lagrangian-eulerian formulation for Element-Free Galerkin method. *Comput. Methods Appl. Mech. Engrg*. **152**, 19-46.
- Qiao, H. (2006). Object-oriented programming for the boundary element method in two-dimensional heat transfer analysis. *Advances in Engineering Software*. **37**, 248–259.
- Rabczuk, T., Belytschko, T. (2007). A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Comput. Methods Appl. Mech. Engrg*. **196**, 2777–2799.

- Rabczuk, T., Belytschko, T., Xiao, S. P. (2004). Stable particle methods based on Lagrangian kernels. *Comput. Methods Appl. Mech. Engrg.* **193**, 1035–1063.
- Rabczuk, T., Eibl, J. (2006). Modelling dynamic failure of concrete with meshfree methods. *International Journal of Impact Engineering.* **32**, 1878–1897.
- Rabczuk, T., Zi, G., Bordas, S., Nguyen-Xuan, H. (2010). A simple and robust three-dimensional cracking-particle method without enrichment. *Computer Methods in Applied Mechanics and Engineering.* **199**, 2437–2455.
- Rahman, T., Valdmán, J. (2013). Fast Matlab assembly of fem matrices in 2D and 3D: Nodal elements. *Applied Mathematics and Computation.* **219(13)**, 7151–7158.
- Rao, B. N., Rahman, S. (2001). A coupled meshless-finite element method for fracture analysis of cracks. *International Journal of Pressure Vessels and Piping.* **78**, 647-657.
- Rathod, H. T., Nagaraja, K. V., Venkatesudu, B. (2007a). On the application of two symmetric Gauss Legendre quadrature rules for composite numerical integration. *Applied Mathematics and Computation.* **190**, 21–39.
- Rathod, H. T., Nagaraja, K. V., Venkatesudu, B. (2007b). Symmetric Gauss Legendre quadrature formulas for composite numerical integration over a triangular surface. *Applied Mathematics and Computation.* **188**, 865–876.
- Recio, D. P., Jorge, R. M. N., Dinis, L. M. S. (2007). On the use of element-free Galerkin method for problems involving incompressibility. *Engineering Analysis with Boundary Elements.* **31**, 103–115.
- RezaeiMojdehi, A., Darvizeh, A., Basti, A., Rajabi, H. (2011). Three dimensional static and dynamic analysis of thick functionally graded plates by the meshless local Petrov–Galerkin (MLPG) method. *Engineering Analysis with Boundary Elements.* **35**, 1168–1180.
- Roque, C. M. C., Cunha, D., Shu, C., Ferreira, A. J. M. (2011). A local radial basis functions—Finite differences technique for the analysis of composite plates. *Engineering Analysis with Boundary Elements.* **35**, 363–374.
- Rouson, D. W. I., Xia, J., Xu, X. (2010). Object construction and destruction design patterns in Fortran 2003. *Procedia Computer Science.* **1**, 1495–1504.
- Sampath, R., Nicholas, Z. (1999). An object oriented implementation of a front tracking finite element method for directional solidification processes. *Int. J. Numer. Meth. Engng.* **44**, 1227-1265.
- Sánchez, J. M. M., Gonçalves, P. B. (2010). Shape function object modeling for meshfree methods. *Mecánica Computacional.* **29**, 4753-4767.
- Saritas, A., Filippou, F. C. (2009). Numerical integration of a class of 3d plastic-damage concrete models and condensation of 3d stress-strain relations for use in beam finite elements. *Engineering Structures.* **31**, 2327-2336.
- Seidl, A., Schmidt, T. (2005). Object oriented approach to consistent implementation of meshless and classical fem. *Systemics, Cybernetics and Informatics.* **4(1)**, 58-64.
- Senturk, U. (2011). Modeling nonlinear waves in a numerical wave tank with localized meshless RBF method. *Computers & Fluids.* **44**, 221–228.

- Shangwu, X., Liu, W. K., Cao, J., Rodrigues, J. M. C., Martins, P. A. F. (2003). On the utilization of the reproducing kernel particle method for the numerical simulation of plane strain rolling. *International Journal of Machine Tools & Manufacture*. **43**, 89–102.
- Shapiro, V., Tsukanov, I. (1999). Meshfree simulation of deforming domains. *Computer-Aided Design*. **31**, 459–471.
- Sheehy, M., Grosse, I. R. (1997). An object-oriented blackboard based approach for automated finite element modeling and analysis of multichip modules. *Engineering with Computers*. **13**, 197-210.
- Shibahara, M., Atluri, S. N. (2011). The meshless local Petrov-Galerkin method for the analysis of heat conduction due to a moving heat source, in welding. *International Journal of Thermal Sciences*. **50**, 984-992.
- Skouras, E. D., Bourantas, G. C., Loukopoulos, V. C., Nikiforidis, G. C. (2011). Truly meshless localized type techniques for the steady-state heat conduction problems for isotropic and functionally graded materials. *Engineering Analysis with Boundary Elements*. **35**, 452–464.
- Sladek, J., Sladek, V., Krivacek, J., Aliabadi, M. H. (2007). Local boundary integral equations for orthotropic shallow shells. *International Journal of Solids and Structures*. **44**, 2285–2303.
- Sokos, E. N., Zahradnik, J. (2008). Isola a Fortran code and a Matlab gui to perform multiple-point source inversion of seismic data. *Computers & Geosciences*. **34**, 967–977.
- Swathi, B., Eldho, T. I. (2014). Ground water flow simulation in unconfined aquifers using meshless local Petrov–Galerkin method. *Engineering Analysis with Boundary Elements*. **48**, 43–52.
- Tabatabai, S. M. R. (2002). Object-oriented finite element-based design and progressive steel weight minimization. *Finite Elements in Analysis and Design*. **39**, 55–76.
- Tang, Q., Zhang, G. Y., Liu, G. R., Zhong, Z. H., He, Z. C. (2011). A three-dimensional adaptive analysis using the meshfree node-based smoothed point interpolation method (NS-PIM). *Engineering Analysis with Boundary Elements*. **35**, 1123–1135.
- Thakur, H., Singh, K. M., Sahoo, P. K. (2010). MLPG analysis of nonlinear heat conduction in irregular domains. *Cmes-Computer Modeling In Engineering & Sciences*. **68**, 117-149.
- Timoshenko, S., Goodier J. N. (1951). *Theory of Elasticity*. McGraw-Hill Book Company, Toronto.
- Tonon, F. (2004). Explicit exact formulas for the 3-D tetrahedron inertia tensor in terms of its vertex coordinates. *Journal of Mathematics and Statistics*. **1(1)**, 8-11.
- Vorobyev, A., Kriventsev, V., Maschek, W. (2011). Simulation of central sloshing experiments with smoothed particle hydrodynamics (SPH) method. *Nuclear Engineering and Design*. **241**, 3086–3096.
- Wang, D., Sun, Y., Li, Ling. (2009). A discontinuous Galerkin meshfree modeling of material interface. *Cmes-Computer Modeling In Engineering & Sciences*. **45**, 57-82.

- Wang, H., Wu, C., Guo, Y., Botkin, M. E. (2009). A coupled meshfree/finite element method for automotive crashworthiness simulations. *International Journal of Impact Engineering*. **36**, 1210–1222.
- Wang, J. G., Liu, G. R. (2002a). A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*. **54**, 1623-1648.
- Wang, J. G., Liu, G. R. (2002b). On the optimal shape parameters of radial basis functions used for 2-D meshless methods. *Comput Methods Appl Mech Engrg*. **191**, 2611–2630.
- Wang, J., Hopke, P. K. (2001). Equation-oriented system: an efficient programming approach to solve multilinear and polynomial equations by the conjugate gradient algorithm. *Chemometrics and Intelligent Laboratory Systems*. **55**, 13-22.
- Wang, Q., Zheng, J. J., Miao, Y., Lv, J. H. (2011). The multi-domain hybrid boundary node method for 3D elasticity. *Engineering Analysis with Boundary Elements*. **35**, 803–810.
- Wang, W., Ji, X., Wang, Y. (1999). Object-oriented programming in boundary element methods using c++. *Advances in Engineering Software*. **30**, 127-132.
- Wang, W., Kolditz, O. (2007). Object-oriented finite element analysis of thermo-hydro-mechanical (thm) problems. *Int. J. Numer. Meth. Engng*. **69**, 162–201.
- Wang, Z., Qin, X., Wei, G., Su, L., Wang, L., Fang, B. (2010). Meshless method with ridge basis functions. *Applied Mathematics and Computation*. **217**, 1870–1886.
- Wegner, T., Peczak, A. (2010). Implementation of a strain energy-based nonlinear finite element in the object-oriented environment. *Computer Physics Communications*. **181**, 520–531.
- Wena, P. H., Aliabadi, M. H. (2011). A variational approach for evaluation of stress intensity factors using the element free Galerkin method. *International Journal of Solids and Structures*. **48**, 1171–1179.
- Wu, C., Chiu, K. (2011). RMVT-based meshless collocation and element-free Galerkin methods for the quasi-3D free vibration analysis of multilayered composite and FGM plates. *Composite Structures*. **93**, 1433–1448.
- Wu, C., Chiu, K., Wang, Y. (2011). RMVT-based meshless collocation and element-free Galerkin methods for the quasi-3D analysis of multilayered composite and FGM plates. *Composite Structures*. **93**, 923–943.
- Wu, C., Yang, S. (2011). RMVT-based meshless collocation and element-free Galerkin methods for the approximate 3D analysis of multilayered composite and FGM circular hollow cylinders. *Composites: Part B*. **42**, 1683–1700.
- Wu, C., Yang, S., Wang, Y., Hu, H. (2011). A meshless collocation method for the plane problems of functionally graded material beams and plates using the DRK interpolation. *Mechanics Research Communications*. **38**, 471– 476.
- Wu, S. C., Liu, G. R., Zhang, H. O., Xu, X., Li, Z. R. (2009). A node-based smoothed point interpolation method (NS-PIM) for three-dimensional heat transfer problems. *International Journal of Thermal Sciences*. **48**, 1367–1376.

- Xi, P., Long, S. Y., Wei, K. X. (2011). An analysis for the elasto-elastic problem of the moderately thick plate using the meshless local Petrov–Galerkin method. *Engineering Analysis with Boundary Elements*. **35**, 908–914.
- Xia, P., Long, S., Cui, H. (2009). Elastic dynamic analysis of moderately thick plate using meshless lrpim. *Acta Mechanica Solida Sinica*. **22(2)**, 1-9.
- Xiang, S., Jin, Y., Jiang, S., Bi, Z., Wang, K. (2011). Meshless global radial point collocation method for three-dimensional partial differential equations. *Engineering Analysis with Boundary Elements*. **35**, 289–297.
- Xiao, J. R., Batra, R. C., Gilhooley, D. F., Gillespie Jr, J. W., McCarthy, M. A. (2007). Analysis of thick plates by using a higher-order shear and normal deformable plate theory and MLPG method with radial basis functions. *Comput. Methods Appl. Mech. Engrg.* **196**, 979–987.
- Xiao, J. R., Gama, B. A., Gillespie, J. W., Kansad, E. J. (2005). Meshless solutions of 2D contact problems by subdomain variational inequality and MLPG method with radial basis functions. *Engineering Analysis with Boundary Elements*. **29**, 95–106.
- Xu, S. (2014). Stable cracking particles method based on stabilized nodal integration and updated Lagrangian kernel. *Mathematical Problems in Engineering*. **2014**, 1-10.
- Xue-Hong, W., Wen-Quan, T. (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.
- Yan, F., Feng, X., Zhou, H. (2010). Meshless method of dual reciprocity hybrid radial boundary node method for elasticity. *Acta Mechanica Solida Sinica*. **23(5)**, 447-458.
- Yanan, L., Liang, S., Yinghua, L., Zhangzhi, C. (2011). Multi-scale b-spline method for 2-D elastic problems. *Applied Mathematical Modelling*. **35**, 3685–3697.
- Yavuz, M. M., Kanber, B. (2015). On the usage of tetrahedral background cells in nodal integration of RPIM for 3D elasto-static problems. *International Journal of Computational Methods*. **12(6)**, 1-30.
- Yavuz, M. M., Kanber, B. (2015). On the usage of tetrahedral background cells in nodal integration of RPIM for 3D elasto-static problems, *International Journal of Computational Methods*. **12(6)**, 1-30.
- Yu, G., Adeli, H. (1993). Object-oriented finite element analysis using eer model. *J. Struct. Eng.* **119**, 2763-2781.
- Yu, L., Kumar, A. V. (2001). An object-oriented modular framework for implementing the finite element method. *Computers and Structures*. **79**, 919-928.
- Zabaras, N., Srikanth, A. (1999). An object-oriented programming approach to the Lagrangian fem analysis of large inelastic deformations and metal forming processes. *Int. J. Numer. Meth. Engng.* **45**, 399–445.
- Zeglinski, G. W., Han, R. P. S., Aitchison, P. (1994). Object oriented matrix classes for use in a finite element code using c++. *International Journal for Numerical Methods in Engineering*. **37**, 3921-3937.

- Zhang, C., Cui, M., Wang, J., Gao, X. W., Sladek, J., Sladek, V. (2011). 3D crack analysis in functionally graded materials. *Engineering Fracture Mechanics*. **78**, 585–604.
- Zhang, G. Y., Liu, G. R., Wang, Y. Y., Huang, H. T., Zhong, Z. H., Li, G. Y., Han, X. (2007). A linearly conforming point interpolation method (LC-PIM) for three dimensional elasticity problems. *International Journal for Numerical Methods in Engineering*. **72**, 1524-1543.
- Zhang, X., Subbarayan, G. (2006). Inurbs: an object-oriented, symbolic framework for integrated, meshless analysis and optimal design. *Advances in Engineering Software*. **37**, 287–311.
- Zhang, Y. (2006). Solve partial differential equations by two or more radial basis functions. *Applied Mathematics and Computation*. **181**, 793–801.
- Zhang, Z., Qiang, H., Gao, W. (2011). Coupling of smoothed particle hydrodynamics and finite element method for impact dynamics simulation. *Engineering Structures*. **33**, 255–264.
- Zheng, B., Dai, B. (2011). A meshless local moving kriging method for two-dimensional solids. *Applied Mathematics and Computation*. **218**, 563–573.
- Zhou, F., Zhang, J., Sheng, X., Li, G. (2011). Shape variable radial basis function and its application in dual reciprocity boundary face method. *Engineering analysis with Boundary Elements*. **35**, 244–252.
- Zhou, J. X., Wen, J. B., Zhang, H. Y., Zhang, L. (2003). A nodal integration and post-processing technique based on Voronoi diagram for Galerkin meshless methods. *Comput. Methods Appl. Mech. Engrg.* **192**, 3831–3843.
- Zimmermann, T., Bomme, P., Eyheramendy, D., Vernier, L., Commend, S. (1998). Aspects of an object-oriented finite element environment. *Computers and Structures*. **68**, 1-16.
- Zimmermann, T., Dubois-Pelerin, Y., Bomme, P. (1992). Object oriented finite element programming: i. governing principles. *Computer Methods in Applied Mechanics and Engineering*. **98**, 291-303.

## CURRICULUM VITAE

### PERSONAL INFORMATION

Name and Surname: Mustafa Murat YAVUZ

Nationality: Turkish (T.C.)

Date and Place of Birth: 16/07/1985, Gaziantep

Phone: +90 555 238 57 40

Email: muraty27@hotmail.com

### EDUCATION

	Institution	Year of Graduation
M.Sc	Gaziantep University, Graduate School of Natural and Applied Sciences	2010
B.S	Gaziantep University, Department of Mechanical Engineering	2008
High School	Gaziantep Anatolian High School	2003

### WORK EXPERIENCE

2009-Present Research Assistant (Osmaniye K.A. University)

### PUBLICATIONS

Yavuz, M. M., Kanber, B. (2015). On the usage of tetrahedral background cells in nodal integration of RPIM for 3D elasto-static problems. *International Journal of Computational Methods*. **12(6)**, 1-30. doi: 10.1142/S021987621550036X

Kanber, B., Yavuz, M. M. (2015). Object-oriented programming in meshfree analysis of elastostatic problems. *International Journal of Engineering & Applied Sciences (IJEAS)*. **7(2)**, 1-18. ISSN 1309-0267

Yavuz, M. M., Kanber, B. Application of least squares stress stabilization method on nodal integration of RPIM with tetrahedral background cells. *The 6<sup>th</sup> International Conference on Computational Methods (ICCM 2015)*, 14-17 July 2015, Auckland, New Zealand, 1-20.

Yavuz, M. M., Kanber, B. (2015). Effect of higher order Taylor series expansion terms of the NI-RPIM on the solution accuracy of 2D elastic problems. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*. **21(1)**, 1-10. doi: 10.5505/pajes.2014.29591