

Durham E-Theses

Nonlinear solid mechanics analysis using the parallel selective element-free Galerkin method

ULLAH, ZAHUR

How to cite:

ULLAH, ZAHUR (2013) *Nonlinear solid mechanics analysis using the parallel selective element-free Galerkin method*, Durham theses, Durham University. Available at Durham E-Theses Online:
<http://etheses.dur.ac.uk/6978/>

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

Academic Support Office, Durham University, University Office, Old Elvet, Durham DH1 3HP
e-mail: e-theses.admin@dur.ac.uk Tel: +44 0191 334 6107
<http://etheses.dur.ac.uk>

Durham University

Nonlinear solid mechanics analysis using the parallel selective element-free Galerkin method

Thesis by

Zahur Ullah

Submitted as partial consideration towards
the degree of Doctor of Philosophy



Computational Mechanics Group
School of Engineering & Computing Sciences
Durham University
United Kingdom

April 2013

Nonlinear solid mechanics analysis using the parallel selective element-free Galerkin method

Zahur Ullah

Abstract

A variety of meshless methods have been developed in the last fifteen years with an intention to solve practical engineering problems, but are limited to small academic problems due to associated high computational cost as compared to the standard finite element methods (FEM). The main objective of this thesis is the development of an efficient and accurate algorithm based on meshless methods for the solution of problems involving both material and geometrical nonlinearities, which are of practical importance in many engineering applications, including geomechanics, metal forming and biomechanics. One of the most commonly used meshless methods, the element-free Galerkin method (EFGM) is used in this research, in which maximum entropy shape functions (max-ent) are used instead of the standard moving least squares shape functions, which provides direct imposition of the essential boundary conditions.

Initially, theoretical background and corresponding computer implementations of the EFGM are described for linear and nonlinear problems. The Prandtl-Reuss constitutive model is used to model elasto-plasticity, both updated and total Lagrangian formulations are used to model finite deformation and consistent or algorithmic tangent is used to allow the quadratic rate of asymptotic convergence of the global Newton-Raphson algorithm. An adaptive strategy is developed for the EFGM for two- and three-dimensional nonlinear problems based on the Chung & Belytschko error estimation procedure, which was originally proposed for linear elastic problems. A new FE-EFGM coupling procedure based on max-ent shape functions is proposed for linear and geometrically nonlinear problems, in which there is no need of interface elements between the FE and EFG regions or any other special treatment, as required in the most previous research. The proposed coupling procedure is extended to become adaptive FE-EFGM coupling for two- and three-dimensional linear and nonlinear problems, in which the Zienkiewicz & Zhu error estimation procedure with the superconvergent patch recovery method for strains and stresses recovery are used in the FE region of the problem domain, while the Chung & Belytschko error estimation procedure is used in the EFG region of the problem domain. Parallel computer algorithms based on distributed memory parallel computer architecture are also developed for different numerical techniques proposed in this thesis. In the parallel program, the message passing interface library is used for inter-processor communication and open-source software packages, METIS and MUMPS are used for the automatic domain decomposition and solution of the final system of linear equations respectively. Separate numerical examples are presented for each algorithm to demonstrate its correct implementation and performance, and results are compared with the corresponding analytical or reference results.

Declaration

The work in this thesis is based on research carried out in the Computational Mechanics Group, School of Engineering and Computing Sciences, Durham University. No part of this report has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Parts of this work have been published in the following:

Journals

Z. Ullah and C.E. Augarde. Finite deformation elasto-plastic modelling using an adaptive meshless method. *Computers & Structures*, in press, available online 21 April 2012. [dx.doi.org/10.1016/j.compstruc.2012.04.001](https://doi.org/10.1016/j.compstruc.2012.04.001).

Conferences

Z. Ullah and C. E. Augarde. Solution of elasto-statics problems using the element-free Galerkin method with local maximum entropy shape functions. In *Proceedings of the 18th UK Conference of the Association for Computational Mechanics in Engineering (ACME)*, Southampton University, Southampton, UK, 29-31 March 2010, 161-164.

Z. Ullah, C. E. Augarde, R. S. Crouch and W. M. Coombs. FE-EFGM Coupling using maximum entropy shape functions and its application to small and finite deformation. In *Proceedings of the 19th UK Conference of the Association for Computational Mechanics in Engineering (ACME)*, Heriot-Watt University, Edinburgh, UK, 5-6 April 2011, 277-280.

Z. Ullah, C. E. Augarde and W. M. Coombs. Adaptive modelling of finite strain shear band localization using the element-free Galerkin method. In *Proceedings of the 20th UK Conference of the Association for Computational Mechanics in Engineering (ACME)*, University of Manchester, Manchester, UK, 26-28 March 2012, 251-254.

Z. Ullah, C. E. Augarde and W. M. Coombs. Parallel element-free Galerkin method algorithm with application to three-dimensional nonlinear adaptive analysis in solid mechanics. *10th World Congress on Computational Mechanics*, São Paulo, Brazil, 8-13 July 2012.

Z. Ullah, C. E. Augarde and W. M. Coombs. Three-dimensional FE-EFGM adaptive coupling with application to nonlinear adaptive analysis. In *International Conference on Computational Mechanics (CM13)*, University of Durham, Durham, UK, 25-27 March 2013 (Accepted).

Copyright © 2013 by Zahur Ullah.

“The copyright of this thesis rests with the author. No quotations from it should be published without the authors prior written consent and information derived from it should be acknowledged.”

Acknowledgements

The research work in this thesis would not have been possible without the financial support of the Overseas Research Students Awards Scheme from the School of Engineering and Computing Sciences, Durham University, UK. Their generous support is highly appreciated.

First of all, I would like to offer my deepest gratitude to my supervisor Dr. Charles Augarde for offering me the fully funded PhD opportunity and for his inspiring supervision, general advice, and encouragements throughout my PhD. His in-depth knowledge in a variety of fields, including solid mechanics, numerical analysis and computer programming has been invaluable. In spite of his busy schedule, he has been always available for discussion of the findings and rationalization of new ideas, which was critical to the successful completion of this thesis. I would also like to thank him for his patience and the time he spends reading this thesis along with other publications, listening to my presentations and providing numerous suggestions and advices for improvements.

My sincere thanks also go to Professor Roger Crouch (my former second supervisor), whose broad knowledge of computer programming and nonlinear solid mechanics, including finite deformation and elasto-plasticity has offered countless critical and very interesting suggestions for my research. Special thanks go to Dr. Will Coombs (my current second supervisor) for providing his nonlinear solid mechanics compact finite element MATLAB code, which was a great source of learning and without which this thesis would not have been completed. His exceptional knowledge of the numerical nonlinear solid mechanics was very helpful and has offered numerous suggestion and advices to improve this research work. His help on the L^AT_EX and CorelDRAW was also very helpful in compiling this thesis along with other publications.

Thanks must go to Dr. Henk Slim (Technical Lead, Scientific and High Performance Computing, Durham University) for conducting training courses on the Hamilton cluster and parallel programming; and later for his continuous support in dealing with my queries, which were of great help for the completion of this thesis. I would also like to mention the current and former colleagues, especially Professor Jon Trevelyan, Dr. Yiqian He, Dr. Claire Heaney and Dr. Xiaoying Zhuang for their endless help and support particularly at the start of my PhD.

Finally, I gratefully acknowledge my beloved family for their much needed constant support, encouragement and love throughout my life.

Zahur Ullah
Durham, Feb 2013

Contents

Abstract	i
Declaration	ii
Acknowledgements	iii
List of Figures	vi
List of Tables	xi
Nomenclature & Abbreviations	xii
1 Introduction	1
1.1 Overview	1
1.2 Scope and outline	5
1.3 Notation	7
2 Numerical implementation of meshless methods for linear elasto-statics problems	8
2.1 Introduction	8
2.2 Governing equations for linear elasticity	8
2.3 Strong and weak forms	10
2.4 Moving least squares approximations	12
2.4.1 Weight functions	14
2.4.2 Numerical demonstration	16
2.5 Element-free Galerkin method	18
2.6 Maximum entropy shape functions	21
2.6.1 Information and informational entropy	22
2.6.2 Polygonal interpolants or global max-ent shape functions	23
2.6.3 Local maximum entropy shape functions	25
2.6.4 Numerical demonstration	26
2.7 Element-free Galerkin method with max-ent shape functions	28
2.8 Numerical examples	29
2.9 Run time comparison between max-ent and MLS shape functions	38
2.10 Concluding remarks	39
3 Modelling nonlinearities with element-free Galerkin method	41
3.1 Introduction	41
3.2 Classical theory of plasticity	43
3.2.1 Elasto-plastic tangent matrix	44
3.2.2 Principle of stress integration	45
3.2.3 Consistent or algorithmic tangent	46
3.2.4 Numerical implementation algorithm	47
3.3 Theory of plasticity with finite deformation	49
3.3.1 Updated Lagrangian formulation	50
3.3.2 Total Lagrangian formulation	51
3.4 Numerical examples	52

3.4.1	Infinite plate strip	52
3.4.2	Elastic cantilever Beam	54
3.4.3	Double notched tensile specimen	56
3.4.4	Thick wall cylinder expansion	57
3.5	Concluding remarks	60
4	Error estimation and adaptivity in element-free Galerkin method	62
4.1	Introduction	62
4.2	Adaptivity in linear-elastic problems	63
4.2.1	Error estimation	63
4.2.2	Refinement strategy	67
4.2.3	Domain of influence based on Voronoi diagram	69
4.2.4	Numerical examples	70
4.3	Adaptivity in nonlinear problems	76
4.3.1	Error estimation	77
4.3.2	Refinement strategy & data transfer	80
4.3.3	Numerical examples	80
4.4	Extension to three-dimensional nonlinear problems	89
4.4.1	Numerical examples	92
4.5	Concluding remarks	96
5	Finite element-element-free Galerkin method coupling	99
5.1	Introduction	99
5.2	FE-EFGM coupling using interface elements	100
5.2.1	Shape functions	101
5.2.2	FE-EFGM for one- and three-dimensional problems	104
5.3	FE-EFGM coupling using max-ent shape functions	105
5.4	Numerical examples	106
5.4.1	One-dimensional bar	106
5.4.2	Two-dimensional cantilever beam	108
5.4.3	Flexible strip footing	110
5.4.4	Three-dimensional cantilever beam problem with end load	113
5.5	Extension to nonlinear problems	116
5.5.1	Numerical examples	117
5.6	Concluding remarks	120
6	Adaptive coupling of finite element and element-free Galerkin methods	122
6.1	Introduction	122
6.2	FE error estimation for two-dimensional linear elastic problems	123
6.2.1	Superconvergent patch recovery method	123
6.2.2	Error estimation	126
6.3	Extension of FE error estimation to three-dimensional and nonlinear problems	126
6.4	Adaptive FE-EFGM coupling for two-dimensional linear-elastic problems	129
6.4.1	Numerical examples	132
6.5	Implementation of the total Lagrangian formulation	136
6.6	Adaptive FE-EFGM coupling for three-dimensional nonlinear problems	139
6.6.1	Numerical examples	141
6.7	Concluding remarks	144
7	Parallel computations	146
7.1	Introduction	146
7.2	Parallel computer architecture	148
7.3	Message passing interface (MPI)	150
7.4	Performance of parallel programs	151
7.5	Automatic partitioning of the problem domain with METIS	152
7.6	MULTifrontal Massively Parallel Solver	152
7.7	Parallel algorithm for linear elastic problems	153

7.7.1	Numerical examples	154
7.8	Parallel algorithm for adaptive nonlinear problems	159
7.8.1	Numerical examples	161
7.9	Final demonstration problem	164
7.10	Concluding remarks	169
8	Conclusions and recommendations for future work	174
8.1	Overall assessment	174
8.2	Recommendations for future work	177
	References	179
	Appendix	201
A	Isoparametric formulations and Jacobian	201
	Appendix	203
B	Neighbouring search with kd-tree	203

List of Figures

2.1	Three-dimensional solid subjected to forces and boundary conditions	9
2.2	Moving least squares approximations	14
2.3	Geometry with nodes and the corresponding rectangular and circular domain of influence	15
2.4	1D cubic spline weight functions and derivatives for $d_{max} = 2.0$	16
2.5	1D cubic spline weight functions and derivatives for $d_{max} = 3.0$	17
2.6	1D MLS shape functions and derivatives for $d_{max} = 2.0$	17
2.7	1D MLS shape functions and derivatives for $d_{max} = 3.0$	17
2.8	2D problem domain, cubic spline weight function and derivatives for a node at $\mathbf{x} = [0, 0]$	18
2.9	2D MLS shape function and derivatives for a node at $\mathbf{x} = [0, 0]$	19
2.10	The EFGM problem discretization	20
2.11	Convergence path of the Newton's method for different points in one-dimensional domain for max-ent shape functions	26
2.12	1D max-ent shape functions and derivatives for $d_{max} = 3.0$	27
2.13	Convergence path of the Newton's method for different points in two-dimensional domain for max-ent shape functions	28
2.14	2D max-ent shape function and derivatives for a node at $\mathbf{x} = [0, 0]$	29
2.15	Geometry and loading for 1D bar problem	30
2.16	Displacement and stress for 1D bar problem	31
2.17	Rate of convergence for $\ e_u\ $ and $\ e\ $ for 1D bar problem	31
2.18	Geometry, boundary condition and loading for 2D beam problem	32
2.19	Discretization and deflections for 2D beam problem	33
2.20	Normal and shear stresses for 2D beam problem	34
2.21	Rate of convergence for $\ e_u\ $ and $\ e\ $ for 2D beam problem	34
2.22	Geometry, boundary condition and loading for a 2D hole in an infinite plate problem	35
2.23	Problem discretization with background cells, nodes, Gauss points and deflected nodes for the hole in an infinite plate problem	36
2.24	Stresses for the hole in an infinite plate problem	37
2.25	Rate of convergence for $\ e_u\ $ and $\ e\ $ for the hole in an infinite plate problem	37
2.26	3D beam problem geometry, discretization and neutral axis deflection	38
2.27	Normal and shear stresses for 3D cantilever beam problem	39
2.28	Sample discretizations used for the run time comparison of max-ent and MLS shape functions	39
2.29	Run time comparison of max-ent and MLS shape functions	40
3.1	Stress return procedure	46
3.2	Convergence of the Newton-Raphson method for an iteration $n + 1$	47
3.3	Relationship between the current and reference configuration	49
3.4	Infinite plate strip subjected to uniformly distributed load	53
3.5	Discretization, imposition of essential boundary conditions and undeformed and deformed configuration for the infinite plate strip problem	54
3.6	Pressure versus displacement for the infinite plate strip problem	54
3.7	Deflected profile or w/h versus x for the infinite plate strip problem	55
3.8	Geometry with undeformed and deformed configuration of the elastic cantilever beam problem with UDL	55
3.9	Pressure versus u_x and u_y for the elastic cantilever beam problem with UDL	56

3.10	Geometry with undeformed and deformed configuration of the elastic cantilever beam problem with tip load	56
3.11	P versus u_x , u_y and θ for the elastic cantilever beam problem with tip load	57
3.12	Geometry, boundary conditions, loading and discretization for double notched tensile specimen	58
3.13	Plastic gauss points at different increments for double notched tensile specimen	59
3.14	u_x , u_y contours and normalized net stress, $\bar{\sigma}/\sigma_y$ versus edge deflection for double notched tensile specimen	59
3.15	Geometry, discretization and boundary conditions for the thick walled cylinder problem .	60
3.16	Internal pressure versus current internal radius for the thick walled cylinder problem . . .	60
4.1	Comparison of EFGM, projected and analytical stress at Gauss points using $d_{max}^a = 1.4$ and $d_{max}^p = 1.2$ for one-dimensional bar problem	66
4.2	Effectivity index (θ) for different d_{max}^a and d_{max}^p values for one-dimensional bar problem .	66
4.3	Relative error (η) for different d_{max}^a and d_{max}^p values for the one-dimensional bar problem	67
4.4	Problem discretizations with background cells and nodes for the hole in an infinite plate problem	68
4.5	Comparison of estimated and exact cells-wise error in energy norm for 121 nodes for $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ for the hole in an infinite plate problem	68
4.6	Comparison of estimated and exact cells-wise error in energy norm for 441 nodes for $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ for the hole in an infinite plate problem	68
4.7	Effectivity index (θ) and relative error (η) for different d_{max}^a and d_{max}^p values for 121 nodes for the hole in an infinite plate problem	69
4.8	Step by step refinement of the integration cells	69
4.9	Problem discretization, Voronoi diagram and corresponding nodal domains of influence . .	70
4.10	Geometry, boundary condition and loading for the cantilever beam problem	71
4.11	Step by step adaptive refinement for the cantilever beam problem	71
4.12	Step by step Voronoi diagram for the adaptively refined cantilever beam problem	72
4.13	Step by step influence domains for the adaptively refined cantilever beam problem	72
4.14	Step by step uniform refinement for the cantilever beam problem	72
4.15	Step by step Voronoi diagram for the uniformly refined cantilever beam problem	73
4.16	Step by step influence domains for the uniformly refined cantilever beam problem	73
4.17	Relative error (η) % versus degrees of freedom for the cantilever beam problem	73
4.18	Geometry, boundary condition and loading for the square block problem	74
4.19	Step by step adaptive refinement for the square block problem	74
4.20	Step by step Voronoi diagram for the adaptively refined square block problem	74
4.21	Step by step influence domains for the adaptively refined square block problem	75
4.22	Step by step uniform refinement for the square block problem	75
4.23	Relative error (η) % versus degrees of freedom for the square block problem	75
4.24	Geometry, boundary condition and loading for the L-shaped plate problem	76
4.25	Step by step adaptive refinement for the L-shaped plate problem	76
4.26	Step by step Voronoi diagram for the adaptively refined L-shaped plate problem	77
4.27	Step by step influence domains for the adaptively refined L-shaped plate problem	77
4.28	Step by step uniform refinement for the L-shaped plate problem	77
4.29	Relative error (η) % versus degrees of freedom for the L-shaped plate problem	78
4.30	Nonlinear adaptive algorithm	79
4.31	Geometry, boundary condition and loading for the strip footing problem	82
4.32	Step by step discretizations for the strip footing problem	82
4.33	Step by step Voronoi diagrams for the strip footing problem	82
4.34	Step by step influence domains for the strip footing problem	83
4.35	Effective plastic strain contours for the strip footing problem	83
4.36	Normalized pressure vs displacement for different discretizations for the strip footing problem	84
4.37	Geometry, boundary condition and loading for the perforated tensile specimen problem .	85
4.38	Step by step discretizations for the perforated tensile specimen problem	85
4.39	Effective plastic strain contours for the perforated tensile specimen problem	86
4.40	Normalized pressure vs displacement for different discretizations for the perforated tensile specimen problem	86

4.41	Geometry, boundary condition and loading for the footing loaded on a slope problem . . .	87
4.42	Step by step discretizations for the footing loaded on a slope problem	87
4.43	Effective plastic strain contours for the footing loaded on a slope problem	88
4.44	Normalized pressure vs displacement for different discretizations for the footing loaded on a slope problem	88
4.45	Geometry, boundary condition and loading for the footing loaded on a vertical cut problem	89
4.46	Step by step discretizations for the footing loaded on a vertical cut problem	90
4.47	Effective plastic strain contours for the footing loaded on a vertical cut problem	90
4.48	Normalized pressure vs displacement for different discretizations for the footing loaded on a vertical cut problem	91
4.49	Three-dimensional problem discretization, Voronoi diagram and corresponding nodal domains of influence	91
4.50	Three dimensional refinement strategy	92
4.51	Geometry, boundary condition and loading for the 3D plate with a hole problem	93
4.52	Step by step discretizations for the 3D plate with a hole problem	93
4.53	Step by step Voronoi diagrams for the 3D plate with a hole problem	94
4.54	Step by step influence domains for the 3D plate with a hole problem	94
4.55	Effective plastic strain contours for the 3D plate with a hole problem	95
4.56	Reaction vs displacement for different discretizations for the 3D plate with a hole problem	95
4.57	Geometry, boundary condition and loading for the 3D vertical cut problem	96
4.58	Step by step discretizations for the 3D vertical cut problem	96
4.59	Step by step Voronoi diagrams for the 3D vertical cut problem	97
4.60	Step by step influence domains for the 3D vertical cut problem	97
4.61	Effective plastic strain contours for the 3D plate with a hole problem	97
4.62	Reaction vs displacement for different discretizations for the 3D vertical cut problem . . .	98
5.1	FE-EFGM coupling with interface elements	101
5.2	Four node iso-parametric quadrilateral element in natural coordinates	101
5.3	Sample one-dimensional ramp $R(\mathbf{x})$ and blending $\beta(\mathbf{x})$ functions	103
5.4	Sample two-dimensional ramp and blending functions	104
5.5	Two node linear element in natural coordinates	105
5.6	Eight node iso-parametric hexahedral element in natural coordinates	105
5.7	FE-EFGM coupling using max-ent shape functions	106
5.8	Discretizations for the one-dimensional bar problem	108
5.9	Displacements for the one-dimensional bar problem	108
5.10	Shape functions and shape function derivatives for the one-dimensional bar problem coupling using MLS with interface elements	109
5.11	Shape functions and shape function derivatives for the one-dimensional bar problem coupling using max-ent without interface elements	110
5.12	Shape functions and shape function derivatives for the one-dimensional bar problem coupling using MLS without interface elements	111
5.13	Convergence plots for different discretizations for the one-dimensional bar problem	112
5.14	Discretizations for the two-dimensional beam problem	112
5.15	Displacements and stresses for the two-dimensional beam problem at $d_{max} = 2.5$	113
5.16	Displacements and stresses for the two-dimensional beam problem at $d_{max} = 3.5$	114
5.17	Convergence plots for different discretizations and at different d_{max} for the two-dimensional beam problem	114
5.18	Geometry, boundary condition and loading for the flexible strip footing problem	115
5.19	Discretizations for the flexible strip footing problem	115
5.20	Surface deflection for different discretization for the flexible strip footing problem	116
5.21	Displacement (u_y) contours at $d_{max} = 3.8$ for the flexible strip footing problem	116
5.22	Three-dimensional beam problem discretizations with interface region (a) FE; (b) EFGM; (c) EFGM-FE; (d) FE-EFGM	117
5.23	Three-dimensional beam problem discretizations without interface region (a) EFGM-FE; (b) FE-EFGM	118
5.24	Three-dimensional beam problem neutral axis deflection u_y	118
5.25	Problem regions and results for the nonlinear infinite plate strip	119

5.26	Problem regions and results for the nonlinear elastic cantilever beam	120
6.1	Sample one-dimensional linear and quadratic elements patches [334]	126
6.2	Sample two-dimensional elements patches [334]	127
6.3	Sample three-dimensional elements patches	128
6.4	Stress calculation for boundary nodes	128
6.5	Adaptively coupled FE-EFGM algorithm for linear elastic problems	131
6.6	Step by step refinement strategy for the adaptive coupled FE-EFGM	132
6.7	Step by step discretizations for the square block problem	133
6.8	Step by step Gauss points for the square block problem	133
6.9	u_x contours for the square block problem	134
6.10	Von Misses stress contours for the square block problem	134
6.11	Step by step FE uniform refinement for the square block problem	135
6.12	Relative error (η) % for the square block problem	135
6.13	Step by step discretizations for the L-shaped plate problem	136
6.14	Step by step Gauss points for the L-shaped plate problem	136
6.15	u_x contours for the L-shaped plate problem	137
6.16	Von Misses stress contours for the L-shaped plate problem	137
6.17	Step by step FE uniform refinement for the L-shaped plate problem	138
6.18	Relative error (η) % for the L-shaped plate problem	138
6.19	Simulation times versus degrees of freedom for the updated and total Lagrangian formulations	139
6.20	Reaction versus displacement for the updated and total Lagrangian formulations	140
6.21	Step by step discretizations for the three-dimensional plate with a hole problem	142
6.22	Displacements and effective plastic strain contours for the three-dimensional plate with a hole problem	143
6.23	Reaction versus displacement for the three-dimensional plate with a hole problem	143
6.24	Step by step discretizations for the three-dimensional vertical cut problem problem	144
6.25	Displacement u_y and effective plastic strain contours for the three-dimensional vertical cut problem	144
6.26	Reaction versus displacement for the three-dimensional vertical cut problem	145
7.1	The von Neumann conventional sequential computer architecture [247]	148
7.2	Block diagram of the shared memory system architecture [247]	150
7.3	Block diagram of the distributed memory system architecture [247]	150
7.4	Traditional and multilevel partitioning algorithms [151]	153
7.5	Sample partitions for two processors	155
7.6	Linear elastic parallel algorithm	156
7.7	Selective Metis partitions for the discretization with 13202 DOFs for the two-dimensional beam problem	157
7.8	Deflections for the selective nodes for the two-dimensional beam problem	158
7.9	Performance on the Hamilton cluster [1] using 1-25 processors for the two-dimensional beam problem	158
7.10	Selective Metis partitions for the discretization with 14762 DOFs for the hole in an infinite plate problem	159
7.11	Displacements for selective nodes for the hole in an infinite plate problem	160
7.12	Performance on the Hamilton cluster [1] using 1-30 processors for the hole in an infinite plate problem	160
7.13	Nonlinear adaptive parallel algorithm	162
7.14	Selective Metis partitions for 6075 degrees of freedom for the 3D plate with a hole problem	163
7.15	Selective Metis partitions for 2793 degrees of freedom for the 3D plate with a hole problem	163
7.16	FEM reference mesh and reaction versus displacement for the 3D plate with a hole problem	164
7.17	Performance on the Hamilton cluster [1] using 1-8 processors for the the 3D plate with a hole problem	165
7.18	Step by step metis partitions for two processors for the adaptive 3D plate with a hole problem	166
7.19	Step by step metis partitions for five processors for the adaptive 3D plate with a hole problem	166

7.20	Final displacements and reaction versus displacement for the adaptive 3D plate with a hole problem	167
7.21	Performances on the Hamilton cluster [1] using 1-6 processors for the adaptive 3D plate with a hole problem	167
7.22	Step by step discretizations for the “adaptive” case for the final demonstration problem .	170
7.23	Contours of u_x and u_y over the final deformed geometry for the “adaptive” case for the final demonstration problem	170
7.24	Time distribution in different part of the code	171
7.25	Step by step discretizations for the “adaptive-1” case for the final demonstration problem	171
7.26	Contours of u_x and u_y over the final deformed geometry for the “adaptive-1” case for the final demonstration problem	172
7.27	Reference FEM meshes for the final demonstration problem	172
7.28	Contours of u_y over the final deformed geometry for the reference FEM meshes for the final demonstration problem	173
7.29	Reaction versus displacement for the final demonstration problem	173
A.1	Mapping of a rectangular element into an isoparametric quadrilateral element	201
B.1	Sample space partitioning and corresponding kd-tree [3]	205
B.2	Implementation of kd-tree with background mesh algorithm	205
B.3	Simulation times versus degrees of freedom for the full search, kd-tree and kd-tree with background mesh algorithms	206

List of Tables

2.1	Convergence of Newton's method for different points in 1D problem domain	27
2.2	Convergence of Newton's method for different points in 2D problem domain	27
2.3	Run time comparison of max-ent and MLS shape functions	39
6.1	Simulation times versus degrees of freedom for the updated and total Lagrangian formulations	139
7.1	Final demonstration problem results summary	169
B.1	Simulation times versus degrees of freedom for the full search, kd-tree and kd-tree with background mesh algorithms	204

Nomenclature & Abbreviations

Abbreviations

ALU	arithmetic and logical unit
BEM	boundary element method
BNM	boundary node method
CAD	computer aided design
CAE	computer aided engineering
CPU	central processing unit
CSPH	corrected smooth particle hydrodynamics
CUDA	compute unified device architecture
DEM	discrete element methods
DOFs	degrees of freedom
EFGM	element-free Galerkin method
FDM	finite difference method
FE-EFGM	coupled finite element-element-free Galerkin method coupling
FEA	finite element analysis
FEM	finite element method
FEs	finite elements
FETI	finite element tearing and interconnecting
FMM	free mesh method
FPM	finite point method
GFDM	general finite difference method
GPGPU	general-purpose graphic processing units
GPUs	graphic processing units
LBIE	local boundary integral equation
LS	least squares
max-ent	maximum entropy
MIMD	multiple instruction multiple data
MISD	multiple instruction single data
MLPG	meshless local Petrov-Galerkin method
MLS	moving least squares
MPI	message passing interface
MPR	minimal patch recovery
MUMPS	multifrontal massively parallel solver
MWS	meshfree weak-strong
NEM	natural element method
PDEs	partial differential equations
PIM	point interpolation method

PUFEM	partition of unity finite element method
PUM	partition of unity method
PVM	parallel virtual machine
RBF	radial basis functions
RHBNM	regular hybrid boundary node method
RKPM	reproducing kernel particle method
SIMD	single instruction multiple data
SISD	single instruction single data
SPH	smoothed particle hydrodynamics
SPR	superconvergent patch recovery
TL	total Lagrangian formulations
UDL	uniformly distributed load
UL	updated Lagrangian formulations

Symbols

α	penalty factor
$\bar{t}_x, \bar{t}_y, \bar{t}_z$	traction components in the x, y and z directions
$\beta(\mathbf{x})$	blending function
$\boldsymbol{\lambda}$	vector of Lagrange multipliers
ϕ	shape functions
$\boldsymbol{\sigma}$	Cauchy stress vector
$\boldsymbol{\sigma}(\mathbf{x})$	exact stress at a point \mathbf{x}
$\boldsymbol{\sigma}^*$	continuous FE stress field
$\boldsymbol{\sigma}^e(\mathbf{x})$	exact error in the stress at a point \mathbf{x}
$\boldsymbol{\sigma}^h(\mathbf{x})$	numerical stress at a point \mathbf{x}
$\boldsymbol{\sigma}^p(\mathbf{x})$	projected stress at a point \mathbf{x}
$\boldsymbol{\sigma}^t$	trial stress
$\boldsymbol{\sigma}_p^e(\mathbf{x})$	approximate error in the stress at a point \mathbf{x}
$\boldsymbol{\tau}$	Kirchhoff stress
$\boldsymbol{\varepsilon}$	total strain vector
$\boldsymbol{\varepsilon}(\mathbf{x})$	exact strain at a point \mathbf{x}
$\boldsymbol{\varepsilon}^e$	elastic strain vector
$\boldsymbol{\varepsilon}^e(\mathbf{x})$	exact error in the strain at a point \mathbf{x}
$\boldsymbol{\varepsilon}^h(\mathbf{x})$	numerical strain at a point \mathbf{x}
$\boldsymbol{\varepsilon}^p$	plastic strain vector
$\boldsymbol{\varepsilon}^p(\mathbf{x})$	projected strain at a point \mathbf{x}
$\boldsymbol{\varepsilon}_p^e(\mathbf{x})$	approximate error in the strain at a point \mathbf{x}
$\boldsymbol{\varepsilon}_t^e$	trial elastic strain
$\delta \mathbf{x}$	backward Euler increment
δ_{ij}	Kronecker delta

$\dot{\gamma}$	plastic multiplier	\mathbf{L}	derivative of the logarithm of \mathbf{b}_t^e with respect to its component
$(\dot{\cdot})$	rate of (\cdot)	\mathbf{n}	vector of an outward unit normal
η	relative error in energy norm	\mathbf{oobf}	out-of-balance (residual) force vector
Γ	problem boundary	\mathbf{P}	non-symmetric first Piola-Kirchhoff stress
Γ_t	traction boundary	\mathbf{p}	polynomial basis function
Γ_u	essential boundary	\mathbf{R}	rotation matrix
Γ_E	boundary of the interface region on the EFG side	\mathbf{U}	right stretch matrix
Γ_F	boundary of the interface region on the FE side	\mathbf{u}	displacement vector
$(\cdot)_{,ii}$	the second partial derivative of (\cdot) with respect to i	\mathbf{v}	left stretch matrix
$(\cdot)_{,i}$	partial derivative with respect to i	\mathbf{x}	spatial coordinates vector
$\langle g_r(x) \rangle$	expectation of $g_r(x)$	∇	differential operator
$\ \bar{e}\ $	estimated error in energy norm	ν	Poisson's ratio
$\ e\ $	L_2 norm of error in energy	Ω	problem domain
$\ e_u\ $	L_2 norm of error in displacement	Ω_E	EFG region
$\ U\ $	energy norm	Ω_F	FE region
\mathbb{E}_σ	stress admissible region	Ω_I	interface region
\mathbb{S}	stress space	Ω_I^e	interface element
$\bar{\mathbf{t}}$	traction vector	$\bar{\epsilon}^P$	effective plastic strain
$\bar{\mathbf{u}}$	specified displacement vector	$\bar{\eta}$	permissible relative error in energy norm
$\tilde{\mathbf{a}}$	MLS unknown parameters	$\overline{\ e\ }_k$	permissible error in each background cell
\mathbf{A}	Hessian matrix	$\bar{\phi}_i$	shape functions of the interface elements
\mathbf{a}	isotropic spatial consistent or algorithmic tangent	$\partial\mathbb{E}_\sigma$	boundary of yield surface
\mathbf{B}	strain matrix	Π	total potential energy
\mathbf{b}	left Cauchy-Green strain matrix	σ_{ij}	Cauchy stress tensor
\mathbf{b}^e	elastic left Cauchy-Green strain matrix	σ_p^*	polynomial expansion of stress
\mathbf{b}_f	body force vector	σ_{VM}	von Mises stress
\mathbf{b}_r	backward Euler residuals	$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$	normal Cauchy stresses
\mathbf{b}_{tr}^e	trial elastic left Cauchy-Green strain matrix	$\sigma_{xy}, \sigma_{yz}, \sigma_{zx}$	shear Cauchy stresses
\mathbf{C}	right Cauchy-Green strain matrix	σ_y	yield strength
\mathbf{C}^e	elastic compliance matrix	$\text{int}(\mathbb{E}_\sigma)$	region inside the yield surface
\mathbf{D}	elastic stiffness matrix	θ	effectivity index
\mathbf{D}^{alg}	consistent or algorithmic tangent	Δ	increment
\mathbf{D}^{ep}	infinitesimal, instantaneous elastoplastic tangent matrix	ε_{ij}	strain tensor
\mathbf{e}_σ	error in stresses	$\tilde{\mathbf{A}}$	isotropic material stiffness tangent matrix
\mathbf{e}_ε	error in strains	$\tilde{x}_i, \tilde{y}_i, \tilde{z}_i$	shifted spatial coordinates
\mathbf{e}_u	error in displacements	ξ, η, ψ	element natural or parent coordinates
\mathbf{F}	deformation gradient	b_x, b_y, b_z	body force components in the x, y and z directions
\mathbf{f}	force vector	c	cohesion or shear strength
\mathbf{f}^{ext}	external force vector	C_n	cost of the parallel computation on n processors
\mathbf{f}^{int}	internal force vector	d_{max}	influence domain's scaling parameter
\mathbf{G}	full stain-displacement matrix	d_{max}^a	scaling parameters for the domain of influence for analysis
\mathbf{I}_3	(3×3) unit matrix	d_{max}^p	scaling parameters for the domain of influence for projection
\mathbf{I}_6	(6×6) identity matrix	d_{mi}	size of influence domain for node i
\mathbf{J}	Jacobian matrix	d_n^a	domain of influence for analysis
\mathbf{K}	stiffness matrix	d_n^p	domain of influence for projection

E	modulus of elasticity	P_{lim}	limiting load
E_n	efficiency	r	normalized radius
E_s	strain energy	$R(\mathbf{x})$	ramp function
F	convex potential function	S_n	speedup
f	yield function	T_1	run time of a parallel program on one processor
g	plastic potential function	T_n	run time of a parallel program on n processors
H	informational entropy	u^{EFG}	EFG approximate displacement
H_{max}	maximum informational entropy	u^{FE}	FE approximate displacement
I	second moment of area	u^h	approximate displacement
J	determinant of the deformation gradient	u_x, u_y, u_z	displacement components in the x , y and z directions
J_2	second invariant of the deviatoric stress	u_{max}	maximum applied displacement
m	number of monomial	$V(i)$	Voronoi cell of node i
N	FE shape functions	W_f	work done by external forces
n_x, n_y, n_z	outward unit normals in the x , y and z directions	w_i	weight function
n_Q	number of Gauss points in each background cell	x, y, z	spatial coordinates
p	probability	Z	partition function in statistical mechanics

Chapter 1

Introduction

1.1 Overview

In computational solid mechanics, the responses of a structure subjected to externally applied loads and boundary conditions, are modelled using partial differential equations (PDEs). Therefore, the solution of these equations is of prime importance to determine the physical behaviour of a structure. Ideally, these equations should be solved analytically/exactly but due to mathematical complications, analytical/exact solutions are only possible for simple cases with simple geometries and boundary conditions [80]. Therefore, for practical engineering problems, which are modelled with more general PDEs with complicated geometries and boundary conditions, approximate/numerical methods are commonly used. According to [285], the history of the development of numerical methods in general, is categorized into three different periods. In the first period (1960-1975), new strategies were developed for the finite element method (FEM), which covers static linear elastic, stability, vibration, hydrostatic and hydrodynamics analysis. In the second period (1976-1990), a variety of new error estimation procedures, including *a priori* and *a posteriori*, were presented for the FEM. During this time, significant contributions were also made in the boundary element method (BEM) with applications to different classes of problems. In the third period (1991-until now), major contributions include, new FEM techniques for fracture modelling, micromechanical adaptive analysis, multiphysics problems modelling, solution of complex biomechanics problems, automatic and efficient mesh generation and adaptivity techniques, new iterative and direct solvers, new visualization tools for post-processing, including three-dimensional animations and virtual reality, generalized FEM and BEM with different interpolation schemes. Most meshless/meshfree methods have also been developed and used during this period.

The main goal of computational mechanics is to develop new numerical methods, which can accurately and efficiently solve practical engineering problems. The process of the development of numerical methods can be divided into four steps [329], i.e. development of mathematical models to describe the underlying physics of the problems, generation of the discretized set of equations from the mathematical model, solution of these equations and verifications of the methods with experimental or reference results. Researchers in theoretical mechanics mainly work on the development of new mathematical models, while researchers in computational mechanics are dealing with the other three steps. Different numerical methods have been proposed in the past few decades, which are widely used to solve complicated real-world problems. In the following, the historical development and applications of the mostly used numerical methods, i.e. the finite difference method (FDM), FEM, BEM, discrete element methods (DEM) and meshless methods are briefly summarized.

Finite difference method

The FDM is one of the oldest numerical methods used to solve structural mechanics problems, in which differential equations are replaced by finite difference equations using Taylor's theorem. As compared to the standard FEM, the FDM can provide a quick numerical solution to certain types of PDEs [21]. The main limitation of the FDM is that it cannot easily handle problems with complicated geometries, which restricts its use to simple academic problems [80].

Finite element method

The FEM is the most widely used and well established numerical methods for the solution of structural mechanics problems. Most of the commercial structural mechanics codes, e.g. ANSYS, Abaqus, etc. are based on the FEM. In FEM, the problem domain is divided into a finite number of subdomains known as "finite elements" and the vertices of each element are known as "nodes". In the FEM modelling, the PDEs describing the underlying physics of the problems are initially converted into integral equations, which are then converted to the discretized set of algebraic equations using polynomial interpolations associated with finite elements (FEs), the solution of which gives the required unknown at nodes (e.g. displacements in solid mechanics) [204].

The full historical development of the FEM is described in number of references, e.g. [4, 204, 329] and is briefly summarized here. The first use of the FEM in structural mechanics dates back to the 1940s [132], where it was used for stress analysis. The use of the energy principle in the FEM for the first time can be found in [11]. The direct method to calculate the stiffness matrix was introduced in [298] and was used for truss and two-dimensional triangular and rectangular elements for plane stress problems. The name "finite element" was first used in [65], in which triangular and rectangular elements were used for two-dimensional stress analysis. The first uses of the FEM for plate and shell analyses can be found in [215] and [111] respectively. The two-dimensional FEM was next extended to three-dimensional problems in a number of references [213, 216], and a new family of three-dimensional elements, including tetrahedral elements were introduced. Furthermore, special case of three-dimensional analysis, i.e. axisymmetric analysis was presented in [66]. The static linear elastic FEM was also extended to nonlinear cases, e.g. finite deformation [299], elasto-plasticity [101] and buckling analysis [102]. The FEM was also considered for the nonlinear structural dynamics analysis in [34] and for biomechanics in [136].

Boundary element methods

In the BEM, the PDEs are initially converted to integral equations, consisting of both surface and volume integrals, which are then converted into boundary integral equations consisting of only surface integrals [51]. As compared to the FEM's domain discretization, in the BEM only boundaries are required to be discretized, which results in very few elements and leads to computational efficiency. Furthermore, the BEM changes the dimensionality of problems, i.e. two- and three-dimensional problems are converted to one- and two-dimensional problems respectively [51]. The BEM is an ideal choice for high stress gradient and linear elastic problems, e.g. problems in linear elastic fracture mechanics. On the other hand, the resulting fully populated matrices and difficulty in modelling nonlinear material response are the two main drawbacks of the BEM [275]. The historical development of the BEM can be approximately dated back to 1963 [140] but then no significant contributions were observed for a long time, due to the popularity of the FEM. Later in [170], an effective and simple way was proposed for the calculation of singular integrals, which appears in the BEM formulations, when the distance between the point of interest inside the body (at which displacements and stresses are required) and surface approaches zero [227]. For the BEM development, the 1980s are considered as the peak time, during which it was

extensively used in fracture mechanics, varieties of new solvers were developed, and the commercial software package BEASY was also introduced. The BEM has been used in a number of applications, e.g. fracture mechanics [251, 275, 276, 309], acoustics [297], structural optimization [51, 295, 296], stress analysis [98], fluid mechanics [240] and heat transfer [239].

Discrete element methods

The DEM is an ideal set of numerical methods for the solution of multi body dynamics, which can handle both rigid and deformable bodies with arbitrary shapes and with continuously varying contacts [43]. For rigid bodies, only the contact laws between the bodies are important but for deformable bodies a proper continuum based, constitutive law must also be considered, which may include elasticity, plasticity, etc. Any numerical method can come under the umbrella of the DEM, if it can allow large displacements and rotations or even complete separations of discrete bodies and can calculate continuously varying contact conditions during the solution. Methods belonging to the DEM can be classified based on contact detection algorithms, types of bodies (rigid or deformable), deformation (small or finite), number (small or large), packing (loose or dense), potential for fragmentation, etc. The DEM is more suitable to simulate the behaviour of granular materials, e.g. sand, soil, powder, jointed rocks, stones, bricks, analyzing slopes, modelling excavation, flow simulation in silos, etc. According to [28], the DEM was first introduced in [74] and was later used in [76] to study granular material using simple shaped bodies, i.e. cylinders, spheres and ellipsoids. Varieties of new contact detection strategies were developed in [229, 310] and the underlying physics of hard and soft contacts associated with rigid and deformable bodies respectively, were presented in [75]. The idealized simple shaped bodies which are commonly used cannot accurately model materials with complicated shaped bodies; therefore, the effects of irregular shaped bodies are also studied both in two-dimensional [13, 267] and in three-dimensional [79].

Meshless methods

In spite of all the previously mentioned positive points of the FEM, it still suffers from high pre-processing time, low accuracy of stresses, difficulty in incorporating adaptivity and it is also not an ideal tool for certain classes of problems, e.g. large deformation, material damage, projectile penetration, fragmentation, crack growth and moving boundaries [196, 329]. To accurately model these problems, remeshing is required at different stages during the solution. Although robust mesh generation and mesh adaptivity algorithms are available, it is a complicated and computationally expensive task, especially for three-dimensional problems. Furthermore, due to remeshing, path dependent variables are required to be transferred between the consecutive meshes, which produces interpolation or mapping errors leading to final degraded accuracy. Meshless or meshfree methods are therefore an ideal choice for these problems, because only a set of nodes is required for the problem discretization. According to [193], the ideal requirements for meshless methods are

“No mesh is necessary at all throughout the process of solving the problem of given arbitrary geometry governed by partial differential system equations subject to all kinds of boundary conditions.”

The methods developed to the point under the umbrella of meshless methods, do not entirely fit into this definition due to numerical complications and are further classified according to the mesh requirement as [193]:

- Methods which use a global background mesh for integration.
- Methods which use a local background mesh for integration.

- Methods which use no background mesh at all, but these methods are not always stable.

In [196], meshless methods are also classified according to the formulation procedures:

- Methods which use the weak form.
- Methods which use the direct strong form.
- Methods which use combine weak and strong form.

Construction of meshless shape functions is an active research area, and a number of meshless shape functions have been developed so far. According to [193] meshless shape functions should ideally possess the following properties:

- handling of arbitrary distribution of nodes,
- stability,
- required minimum consistency,
- compact support,
- Kronecker delta property,
- global problem domain compatibility.

In the past few decades, a variety of new meshless methods have been developed. The history, details, recent developments, shape functions construction and computer implementations of meshless methods can be found in a number of review papers [36, 99, 188, 233, 289] and some of the development is also described later in this thesis. A number of dedicated monographs and books are also available, including [17, 58, 189, 193, 196] for in-depth explanations of different meshless methods.

Smoothed particle hydrodynamics (SPH) is the oldest meshless method, and was first proposed to solve three-dimensional astrophysics problems [109, 209] in open space. Later, it was also extended to solve problems in hydrodynamics [222], compressible gas flow [223] and structural mechanics [191]. In the SPH method, the problem domain is represented using a set of particles and then using the SPH approximation the strong form of the governing equations is discretized. The SPH is based on the Lagrangian formulation, in which each individual particle carries all the required properties. The SPH method however suffers from tensile instability and inconsistency, which respectively creates problems in modelling solid mechanics problems and leads to inferior accuracy. With different proposed improvements, the SPH has already been used for the solution of real-world problems [100], including flow modelling, heat and mass transfer, shock modelling, fracture mechanics, metal forming, impact and explosion. A number of detailed review papers and dedicated books are also available on the SPH, e.g. [42, 100, 224]. Based on the SPH method, a more accurate reproducing kernel particle method (RKPM) was proposed in [201], which uses a correction function in the original SPH approximation.

As compared to SPH strong form formulations, the element-free Galerkin method (EFGM) was proposed in [38], which is based on the global weak form. It is one of the most commonly used meshless methods and is based on the earlier version diffuse element method [231]. In the EFGM, moving least squares (MLS) shape functions were used for the approximation of the field variables; a background mesh was used for numerical integration and Lagrange multipliers were used for the imposition of essential boundary conditions. The EFGM has already been applied to a wide variety of problems, e.g. 2D linear elasticity [38, 85, 208], 3D linear and non-linear problems [31, 37], static and dynamic fracture mechanics

[39, 41], plate and shell analysis [167, 168], vibration [57, 200, 308], electromagnetics [63], heat transfer [38, 277–281], metal forming [319, 321], biomechanics [131, 311], geomechanics [323], etc.

The meshless local Petrov-Galerkin method (MLPG), which is based on a local weak form of governing equations was proposed in [18, 19]. As compared to the EFGM, in the MLPG method, there is no need of a global background mesh for numerical integration and is therefore termed a “truly meshless method”. In the MLPG method, numerical integration was carried out using local integration cells belonging to overlapping subdomains associated with each node. A detailed summary of the progress and applications of the MLPG method can be found in two monographs [16, 17]. The most commonly used meshless methods are:

Based on the strong form

General finite difference method (GFDM) [192], SPH, meshfree collocation methods [148, 202, 325, 339], finite point method (FPM) [234], etc.

Based on the global weak form

Diffuse element method, EFGM, point interpolation method (PIM) [194, 196], radial point interpolation method (RPIM) [196, 307], RKPM, hp-cloud method [88], partition of unity method (PUM) [23], etc.

Based on the local weak form

MLPG, Local point interpolation method (LPIM) [121], local radial point interpolation method (LRPIM) [198], meshless local Kriging method [122], etc.

Based on the combined strong and weak form

Meshfree weak-strong (MWS) form method [83, 195], etc.

Based on the boundary representation

Boundary node method (BNM) [228], local boundary integral equation (LBIE) method [328], boundary point interpolation method (BPIM) [118], regular hybrid boundary node method (RHBNM) [324], etc.

1.2 Scope and outline

Problems involving both material and geometrical nonlinearities are of practical importance in many engineering applications, e.g. geomechanics, metal forming and biomechanics. The main purpose of this thesis is to develop and implement numerical techniques based on meshless methods, which can efficiently and accurately model these problems. This thesis is divided into three different sections:

1. Chapter 1 presents a general introduction and description of the historical development of different numerical methods used in structural mechanics.
2. Chapters 2 and 3 deal with the theoretical background and numerical implementations of the EFGM for linear elastic and nonlinear problems respectively.
 - In Chapter 2, fundamentals required for the numerical implementation of the EFGM are reviewed, which includes influence domain and weight functions. Complete formulations are derived for the MLS shape functions and the conventional EFGM [38], which uses MLS shape functions for the approximation of the field variables. The global and local maximum entropy shape (max-ent) functions are also described and are used in the EFGM. A variety of one-, two- and three-dimensional linear elastic numerical examples are given at the end, to verify and validate the implementation.

- Chapter 3 provides detailed formulations and numerical implementations of small and finite strain elasto-plasticity. The main components of the small strain elasto-plasticity are described in detail, which includes elasto-plastic tangent matrix, principle of stress integration, consistent tangent and numerical implementation algorithm. The small strain elasto-plastic formulations are then extended to the large strain case, where both updated and total Lagrangian formulations are discussed. The full numerical implementations are validated with four two-dimensional benchmarked numerical examples.
3. Chapter 4 to 7 present the main contributions of this thesis, in which a variety of new numerical techniques have been developed for accurate and efficient modelling of problems with both material and geometrical nonlinearities. These new techniques include, adaptive EFGM, coupled FE-EFGM, adaptive FE-EFGM coupling and parallel implementations.
- Chapter 4 develops two- and three-dimensional nonlinear adaptive procedures for the EFGM based on the Chung & Belytschko error estimator. Initially, the standard Chung & Belytschko error estimation procedure, corresponding refinement strategy and adaptive algorithm are described for two-dimensional linear elastic problems. The calculation of domains of influence based on the Voronoi diagram is also explained. The implementation and performance of the two-dimensional linear elastic adaptive procedure is demonstrated with three numerical examples. The two-dimensional linear elastic adaptive procedure is then extended to two- and three-dimensional nonlinear cases, and the associated incremental error estimation procedure, refinement strategy and data transfer procedure between the consecutive discretizations are described. For the feasibility and accuracy of these methods, a number of two- and three-dimensional test problems are also included.
 - Chapter 5 presents a new FE-EFGM coupling procedure based on the max-ent shape functions. The conventional FE-EFGM coupling procedure for linear elastic problems, based on the interface or transition elements [40] is described in detail for one-, two- and three-dimensional cases. The proposed coupling procedure based on the max-ent shape functions for linear elastic problems is given next, which is then extended to geometrically nonlinear problems. A number of one- and two-dimensional linear elastic and geometrically nonlinear numerical are given for demonstration.
 - Chapter 6 extends the FE-EFGM coupling procedure presented in Chapter 5 to adaptive FE-EFGM coupling for both two- and three-dimensional linear and nonlinear problems. Initially, the standard FE error estimation procedure, the Zienkiewicz & Zhu with the superconvergent patch recovery (SPR) method for stress recovery is reviewed. The adaptive FE-EFGM coupling procedure is first described for two-dimensional linear elastic problems and is then extended to three-dimensional nonlinear problems. The feasibility and accuracy of these algorithms are tested on linear and nonlinear, two- and three-dimensional problems. In this chapter, the performance of the updated and total Lagrangian formulations are also compared in the case of the EFGM.
 - Chapter 8 develops distributed memory parallel computer implementations of the different numerical techniques developed in this thesis. Common parallel computer architectures are summarized and important concepts related to a parallel programming, i.e. the Message passing interface (MPI) library, performance parameters, METIS and MULTIFRONTAL MASSIVELY PARALLEL SOLVER (MUMPS) are described in detail. A parallel EFGM algorithm is initially presented for linear elastic problems and is then extended to adaptive nonlinear problems. A
-

variety of two- and three-dimensional numerical examples are also given to demonstrate the implementation and performance of these algorithms on the Durham University's Hamilton cluster. The developed parallel algorithms are then extended to three-dimensional nonlinear adaptive FE-EFGM coupling case and is tested on the final deformation problem.

Conclusions and recommendations for future work are given in Chapter 8.

1.3 Notation

Throughout this thesis, all mathematical formulations are presented in either matrix/vector or index notation, where essential for clear and better presentation. Due to symmetry in the shear components, for strains and stresses, instead of the four and nine component tensors for two- and three-dimensional cases, three and six components vectors are used respectively. All two-dimensional linear and nonlinear (except the two-dimensional parallel EFGM), and linear-elastic three-dimensional algorithms are implemented in MATLAB, while FORTRAN is used as a programming language for the two-dimensional parallel EFGM and the rest of the nonlinear three-dimensional algorithms.

Chapter 2

Numerical implementation of meshless methods for linear elasto-statics problems

2.1 Introduction

In this chapter, numerical implementation of meshless methods, especially the EFGM, is described in detail. Firstly, the conventional implementation with the MLS shape functions [38] is described. The MLS shape functions do not possess the Kronecker delta property, which complicates the imposition of essential boundary conditions and Lagrange multipliers are used for this purpose. Imposition of the essential boundary conditions is an important issue in meshless methods because of the lack of the Kronecker delta property of the MLS shape functions. Different methods can be found in the literature for the imposition of the essential boundary conditions, e.g. Lagrange multipliers in [38], penalty method in [19] and singular weight functions in [147] and [232]. In [95] a general overview of different methods for the imposition of essential boundary conditions in meshless methods is given, e.g. Lagrange multipliers, penalty method, Nitsche's method, and coupling with finite elements. As compared to MLS shape functions, max-ent shape functions are strictly positive and possess a weak Kronecker delta property at the boundaries, which facilitates the direct imposition of the essential boundary conditions as in the case of the FEM.

The outline of this chapter is as follows. Basic equations of elasticity are summarized in §2.2, and the concept of weak and strong forms are reviewed in §2.3. The complete formulation of the MLS shape functions and weight functions with one- and two-dimensional demonstrations are given in §2.4. The conventional EFGM formulations with MLS shape functions and Lagrange multipliers to impose the essential boundary conditions are given in §2.5. Maximum entropy formulations are given in §2.6 with one- and two-dimensional implementation followed by its use in the EFGM in §2.7. One-, two- and three-dimensional linear elastic numerical examples are presented in §2.8. Finally a comparative study of the run time of the max-ent and MLS shape functions are performed in §2.9 and concluding remarks are given in §2.10.

2.2 Governing equations for linear elasticity

These are commonly used equations in the computational solid mechanics, which are extensively used throughout this thesis and are summarized here for easy reference. Three-dimensional governing equations

for linear elasticity are presented first, which are then transformed into two-dimensional plane stress and plane strain cases. Consider a three-dimensional solid with volume Ω and bounded by boundary Γ as shown in Figure 2.1. This body is subjected to traction $\bar{\mathbf{t}} = \begin{bmatrix} \bar{t}_x & \bar{t}_y & \bar{t}_z \end{bmatrix}^T$ acting over the traction boundary Γ_t and body force $\mathbf{b}_f = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T$ acting over the volume of the body, where the subscripts x , y and z associated with components of $\bar{\mathbf{t}}$ and \mathbf{b}_f shows their directions in the Cartesian coordinates. A displacement $\bar{\mathbf{u}}$ is also specified over the part of the boundary, Γ_u . The nine stress

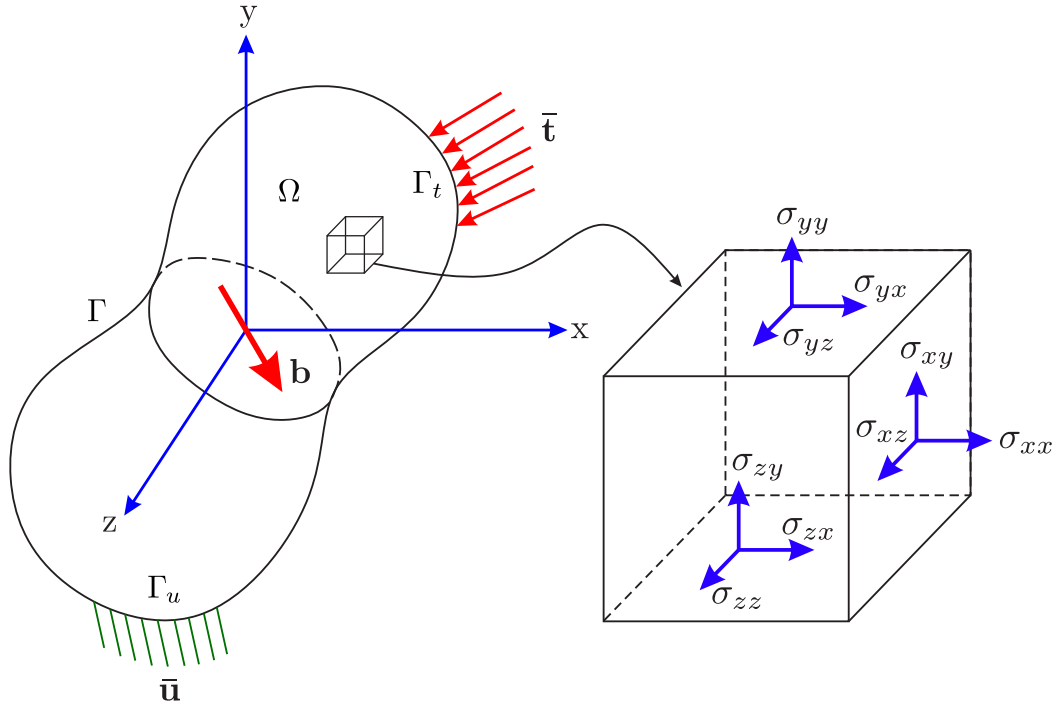


Figure 2.1: Three-dimensional solid subjected to forces and boundary conditions

components over an infinitesimal cube inside the body are also shown in Figure 2.1, where the first subscript indicates the face and the second shows the direction. Out of these nine stress components, only six are independent due to symmetry in shear stresses, i.e.

$$\sigma_{xy} = \sigma_{yx}, \quad \sigma_{yz} = \sigma_{zy}, \quad \sigma_{zx} = \sigma_{xz}. \quad (2.1)$$

The relationship between stress and strain vectors, which is also known as constitutive equation is written as

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix} = \mathbf{D}\boldsymbol{\varepsilon} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{Bmatrix} \quad (2.2)$$

here \mathbf{D} is the matrix of material constants, ν is the Poisson's ratio and E is the modulus of elasticity. In Equation (2.2), σ_{xx} , σ_{yy} and σ_{zz} are normal stresses and ε_{xx} , ε_{yy} and ε_{zz} are the corresponding normal strains, while σ_{xy} , σ_{yz} and σ_{xz} are shear stresses and ε_{xy} , ε_{yz} and ε_{xz} are the corresponding shear strains.

The relationship between strain vector $\boldsymbol{\varepsilon}$ and displacement vector $\mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$, where u_x , u_y and u_z are the displacements in x , y and z directions, is written as

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{yz} \\ \varepsilon_{xz} \end{Bmatrix} = \nabla \mathbf{u} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix}, \quad (2.3)$$

where ∇ is the differential operator. The equilibrium equation, which is the relationship between the external forces and stresses is written as

$$\nabla^T \boldsymbol{\sigma} + \mathbf{b}_f = 0. \quad (2.4)$$

The boundary conditions associated with Equation (2.4) are the traction and essential boundary conditions, which are given as

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_t, \quad (2.5a)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_u, \quad (2.5b)$$

where $\mathbf{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T$ are the outward unit normal to the boundary Γ . The governing equations of the three-dimensional linear elasticity can be converted to two-dimensional case using the plane stress and plane strain concepts. In the plane stress case, dimension of the body in the z direction is very small as compared to the dimensions in x and y directions, and all the stress components in the z direction can be ignored, i.e. $\sigma_{zz} = \sigma_{yz} = \sigma_{xz} = 0$. On the other hand, in the plane strain case, the dimension of the body in the z direction is very large as compared to the dimensions in x and y directions and all the strain components in the z direction can be ignored, i.e. $\varepsilon_{zz} = \varepsilon_{yz} = \varepsilon_{xz} = 0$. For the plane strain and plane stress cases \mathbf{D} matrix can be written as

$$\mathbf{D} = \frac{\bar{E}}{1 - \bar{\nu}^2} \begin{bmatrix} 1 & \bar{\nu} & 0 \\ \bar{\nu} & 1 & 0 \\ 0 & 0 & \frac{1 - \bar{\nu}}{2} \end{bmatrix}, \quad (2.6)$$

$$\bar{E} = \begin{cases} E & \text{for plane stress,} \\ \frac{E}{1 - \nu^2} & \text{for plane strain,} \end{cases} \quad (2.7a)$$

$$\bar{\nu} = \begin{cases} \nu & \text{for plane stress,} \\ \frac{\nu}{1 - \nu} & \text{for plane strain.} \end{cases} \quad (2.7b)$$

2.3 Strong and weak forms

In meshless methods, the concepts of strong and weak formulations of the governing equations are the same as in the FEM, i.e. the terms “strong” and “weak” refer to requirements on the consistency of the approximation functions. Equation (2.4) is said to be in the strong form, because the displacements approximation functions required for its solution, must have a second-order consistency in the entire problem domain, which is equal to the order of the equation. To obtain a solution for the strong

form of the governing equations is not possible for practical engineering problems due to mathematical complications. On the other hand, for the solution of the weak form of Equation (2.4), only first order consistency is required for the displacement approximation functions, which is one order less than the order of Equation (2.4). For a general case with $2k^{th}$ order differential equation, for direct solution of the strong form, the function of the field variables must be differentiable $2k$ times, while for the solution of the weak form, it needs only to be differentiable k times. Most numerical methods developed to date with a few exceptions, are based on weak form formulations for stability and accuracy. Different methods however are used throughout the literature to derive the weak form of the governing equations, i.e. weighted residual, variational principle, Hamilton principle and principle of total minimum potential energy [193]. Here the principle of total minimum potential energy is used for this purpose, which states that for a given admissible displacement, the total potential energy Π is conserved, i.e. $\delta\Pi = 0$. The total potential energy is written as $\Pi = E_s - W_f$, where E_s is strain energy and W_f is work done by external forces and are given as

$$E_s = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} d\Omega, \quad (2.8a)$$

$$W_f = \int_{\Omega} \mathbf{u}^T \mathbf{b}_f d\Omega + \int_{\Gamma_t} \mathbf{u}^T \bar{\mathbf{t}} d\Gamma. \quad (2.8b)$$

Now according to the principle of minimum total potential energy

$$\delta\Pi = \delta \left(\frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \mathbf{u}^T \mathbf{b}_f d\Omega - \int_{\Gamma_t} \mathbf{u}^T \bar{\mathbf{t}} d\Gamma \right) = 0. \quad (2.9)$$

After simplification [196]

$$\int_{\Omega} \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b}_f d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma = 0. \quad (2.10)$$

Equation (2.10) can also be termed as a “principle of virtual work”, i.e. the virtual work done by internal stresses is the same as the virtual work done by all external forces. If the body is subjected to a virtual displacement $\delta \mathbf{u}$, then in Equation (2.10) the first, second and third terms are the virtual work done by the internal stresses, body forces and tractions respectively. Using the stresses-strains and strains-displacements relationship, Equation (2.10) is written as

$$\int_{\Omega} \delta (\nabla \mathbf{u})^T \mathbf{D} (\nabla \mathbf{u}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b}_f d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma = 0, \quad (2.11)$$

which is known as the Galerkin weak form of the governing equations. Generally, in meshless methods, the assumed displacement functions do not satisfy the Kronecker delta property and so-called constrained Galerkin weak forms are used instead of Equation (2.11). Lagrange multipliers and the penalty method are the two commonly used procedures in the literature to write the constrained Galerkin weak form. The constrained Galerkin weak form, using the Lagrange multipliers method, is written [196]

$$\int_{\Omega} \delta (\nabla \mathbf{u})^T (\mathbf{D}) (\nabla \mathbf{u}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b}_f d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma - \int_{\Gamma_u} \delta \boldsymbol{\lambda}^T (\mathbf{u} - \bar{\mathbf{u}}) d\Gamma - \int_{\Gamma_u} \delta \mathbf{u}^T \boldsymbol{\lambda} d\Gamma = 0, \quad (2.12)$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers, physically viewed as a force, which will enforce the essential boundary conditions, i.e. $\mathbf{u} - \bar{\mathbf{u}} = 0$. Lagrange multipliers are unknown variables, the use of which increases the number of equations and correspondingly the number of unknowns in the final system of linear equations. The constrained Galerkin weak form, using the penalty method, is written [196]

$$\int_{\Omega} \delta (\nabla \mathbf{u})^T \mathbf{D} (\nabla \mathbf{u}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b}_f d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma - \delta \int_{\Gamma_u} \frac{1}{2} (\mathbf{u} - \bar{\mathbf{u}})^T \alpha (\mathbf{u} - \bar{\mathbf{u}}) d\Gamma = 0, \quad (2.13)$$

where α is a positive or negative constant [15] known as the penalty factor. In Equation (2.13) the term $(\mathbf{u} - \bar{\mathbf{u}})^T \alpha (\mathbf{u} - \bar{\mathbf{u}})$ is always positive and $\frac{1}{2}$ is used to cancel the 2, which appears later in the simplification. As compared to the Lagrange multipliers, the penalty factor is known constant and will not increase the size of the final discretized system of equations. Ideally, the penalty factor should be infinity, which would exactly satisfy the essential boundary conditions but is impossible to use in the numerical computations. The value of the penalty factor should be carefully selected for a particular problem, in the literature [193], the following values are suggested

$$\alpha = 1.0 \times 10^{4 \sim 13} \times (\text{maximum diagonal element in the stiffness matrix}),$$

or

$$\alpha = 1.0 \times 10^{5 \sim 8} \times \text{Young's modulus}.$$

2.4 Moving least squares approximations

MLS was first presented in [178] as an improvement over the non-interpolating conventional least squares (LS) method for interpolation of scattered data in multi-dimensions. Later on, MLS became a very important component of the family of the meshless methods [19, 38, 193, 196] for the approximations of the field variable (i.e. displacements in the case of solid mechanics). The MLS approximation u^h of a scalar function u at point \mathbf{x} is given as

$$u^h(\mathbf{x}) = \sum_{j=1}^m p_j(\mathbf{x}) \tilde{a}_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \tilde{\mathbf{a}}(\mathbf{x}), \quad (2.14)$$

where $\mathbf{p}(\mathbf{x})$ is a polynomial basis function of the spatial coordinates and m is the number of monomial terms in the basis function. $\tilde{\mathbf{a}}(\mathbf{x})$ are unknown parameters and in the case of the MLS approximation are also a function of spatial coordinates \mathbf{x} as shown in Equation (2.14), while in the case of conventional LS $\tilde{\mathbf{a}}(\mathbf{x})$ is constant over the whole problem domain. This localized feature of the MLS over the conventional LS makes it suitable for the approximation of field variables (even with strong gradients). The polynomial basis function $\mathbf{p}(\mathbf{x})$ is build from Pascal's triangle and Pascal's pyramid for two- and three-dimensional problems respectively. In the one-dimensional case, the linear and quadratic basis functions are given as

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x) = [1, x], \quad m = 2, \quad (2.15a)$$

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x) = [1, x, x^2], \quad m = 3, \quad (2.15b)$$

two-dimensional linear and quadratic basis functions are given as

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x, y) = [1, x, y], \quad m = 3, \quad (2.16a)$$

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x, y) = [1, x, y, xy, x^2, y^2], \quad m = 6, \quad (2.16b)$$

and three-dimensional linear and quadratic basis functions are given as

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x, y, z) = [1, x, y, z], \quad m = 4, \quad (2.17a)$$

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x, y, z) = [1, x, y, z, xy, yz, zx, x^2, y^2, z^2], \quad m = 10. \quad (2.17b)$$

The unknown coefficients $\tilde{\mathbf{a}}(\mathbf{x}) = [\tilde{a}_1(\mathbf{x}) \ \tilde{a}_2(\mathbf{x}) \ \cdots \ \tilde{a}_m(\mathbf{x})]^T$ in Equation (2.14) can be found by minimizing the following weighted discrete L_2 norm.

$$J(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) (\mathbf{p}^T(\mathbf{x}_i) \tilde{\mathbf{a}}(\mathbf{x}) - u_i)^2, \quad (2.18)$$

where $w_i(\mathbf{x})$ is a weight function of compact support for node i evaluated at a point \mathbf{x} , n is the total number of nodes in the support of a point \mathbf{x} , such that $w_i(\mathbf{x}) \neq 0$ and u_i is the nodal value of u at $\mathbf{x} = \mathbf{x}_i$. The minimum of J with respect to $\tilde{\mathbf{a}}(\mathbf{x})$ is found by

$$\frac{\partial J}{\partial \tilde{\mathbf{a}}} = 0, \quad (2.19)$$

which leads to the following system of linear equations

$$\mathbf{A}(\mathbf{x}) \tilde{\mathbf{a}}(\mathbf{x}) = \mathbf{B}(\mathbf{x}) \mathbf{u}. \quad (2.20)$$

Here $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are $(m \times m)$ and $(m \times n)$ matrices respectively and are given as

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i), \quad (2.21a)$$

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} w_1(\mathbf{x}) \mathbf{p}(\mathbf{x}_1) & w_2(\mathbf{x}) \mathbf{p}(\mathbf{x}_2) & \cdots & w_n(\mathbf{x}) \mathbf{p}(\mathbf{x}_n) \end{bmatrix}, \quad (2.21b)$$

and \mathbf{u} is $(n \times 1)$ vector and is given as

$$\mathbf{u} = [u_1, u_2, \dots, u_n]^T. \quad (2.22)$$

$\tilde{\mathbf{a}}(\mathbf{x})$ can be found using Equation (2.20), i.e.

$$\tilde{\mathbf{a}}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}) \mathbf{u}. \quad (2.23)$$

Putting $\tilde{\mathbf{a}}(\mathbf{x})$ from Equation (2.23) into Equation (2.14)

$$u^h(\mathbf{x}) = \underbrace{\mathbf{p}^T(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x})}_{\phi^T(\mathbf{x})} \mathbf{u} = \phi^T(\mathbf{x}) \mathbf{u}. \quad (2.24)$$

Here $\mathbf{p}^T(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}) = \phi^T(\mathbf{x})$, where $\phi(\mathbf{x})$ is a vector of shape functions. The first derivative of the shape function with respect to the spatial coordinates is also required for the numerical implementation of the solid mechanics problem and is given as

$$\phi_{,i} = \mathbf{P}_{,i}^T \mathbf{A}^{-1} \mathbf{B} + \mathbf{P}^T (\mathbf{A}_{,i}^{-1} \mathbf{B} + \mathbf{A}^{-1} \mathbf{B}_{,i}), \quad (2.25)$$

where

$$\mathbf{A}_{,i}^{-1} = -\mathbf{A}^{-1} \mathbf{A}_{,i} \mathbf{A}^{-1}, \quad (2.26)$$

and the index after the comma is a spatial derivative. The MLS shape functions do not satisfy the Kronecker delta property, i.e. $\phi_i(\mathbf{x}_j) \neq \delta_{ij}$ and are termed as approximants instead of interpolants. The values obtained from the MLS approximation are therefore, not the same as the nodal values, i.e. $u^h(\mathbf{x}_i) \neq u_i$ and are known as nodal parameters or fictitious nodal values as shown in Figure 2.2.

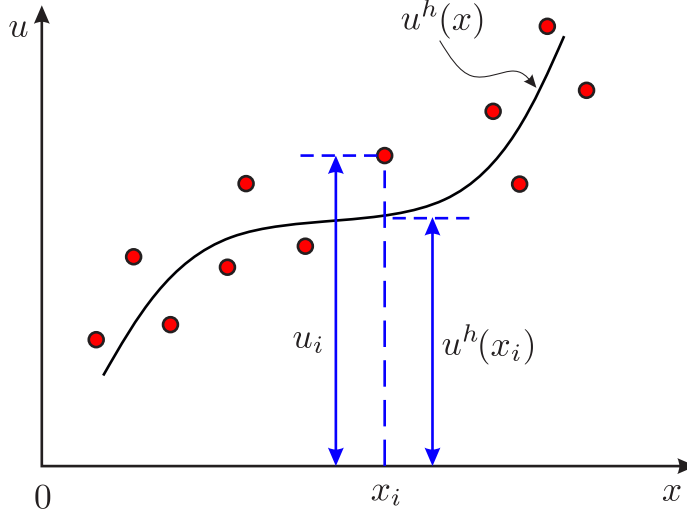


Figure 2.2: Moving least squares approximations

2.4.1 Weight functions

It is clear from §2.4, that the nodal weight functions are an important component of the MLS approximations. There are no predefined rules to select the weight functions for a particular application but the weight functions used for meshless methods have the following properties.

1. Its value should be maximum at the node and decrease with the distance $\|\mathbf{x} - \mathbf{x}_i\|$ from the node.
2. Smooth and non-negative.
3. It should have a compact support, i.e. non-zero over a small neighbourhood of a node, this compact support is known as the domain of influence of a node.

Domain of influence of a node is a very important concept in meshless methods, as it determines the region in which it has influence. The size of domain of influence for a node i is $d_{mi} = d_{max}c_i$, where d_{max} is a scaling parameter, for static analysis its value ranges from 2.0 to 4.0 [85] and c_i is determined by searching for enough neighbour nodes such that matrix \mathbf{A} in Equation (2.20) is invertible. For uniformly distributed nodes c_i can be chosen as the distance between two neighboring nodes. The following are the most common weight functions used in the literature [85, 193].

- The cubic spline weight function in one-dimension

$$w(x - x_i) = w(r) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & \text{for } r \leq \frac{1}{2}, \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3 & \text{for } \frac{1}{2} < r \leq 1, \\ 0 & \text{for } r > 1. \end{cases} \quad (2.27)$$

- The quartic spline weight function in one-dimension

$$w(x - x_i) = w(r) = \begin{cases} 1 - 6r^2 + 8r^3 - 3r^4 & \text{for } r \leq 1, \\ 0 & \text{for } r > 1. \end{cases} \quad (2.28)$$

Here $r = \frac{d_i}{d_{mi}}$ is the normalized radius, where $d_i = \|x - x_i\|$ is the distance between the node i and point of interest x . The weight function derivatives with respect to the spatial coordinates are also required for the shape function derivatives as given in Equation (2.25) and are given as follows.

1. The cubic spline weight function derivative in one dimension

$$\frac{dw}{dx} = \frac{dw}{dr} \frac{dr}{dx} = \begin{cases} (-8r + 12r^2) \operatorname{sign}(x - x_i) & \text{for } r \leq \frac{1}{2}, \\ (-4 + 8r - 4r^2) \operatorname{sign}(x - x_i) & \text{for } \frac{1}{2} < r \leq 1, \\ 0 & \text{for } r > 1. \end{cases} \quad (2.29)$$

2. The quartic spline weight function derivative in one dimension

$$\frac{dw}{dx} = \frac{dw}{dr} \frac{dr}{dx} = \begin{cases} (-12r + 24r^2 - 12r^3) \operatorname{sign}(x - x_i) & \text{for } r \leq 1, \\ 0 & \text{for } r > 1. \end{cases} \quad (2.30)$$

The domain of influence used in the literature is usually circular or rectangular in two dimensions, and hexahedral or spherical in three dimensions. For illustrative purposes, a sample 2D grid of nodes is shown in Figure 2.3(a) and the corresponding rectangular and circular domains of influence (reduced in size for clarity) are shown in Figures 2.3(b) and 2.3(c) respectively. For the rectangular and hexahedral influence

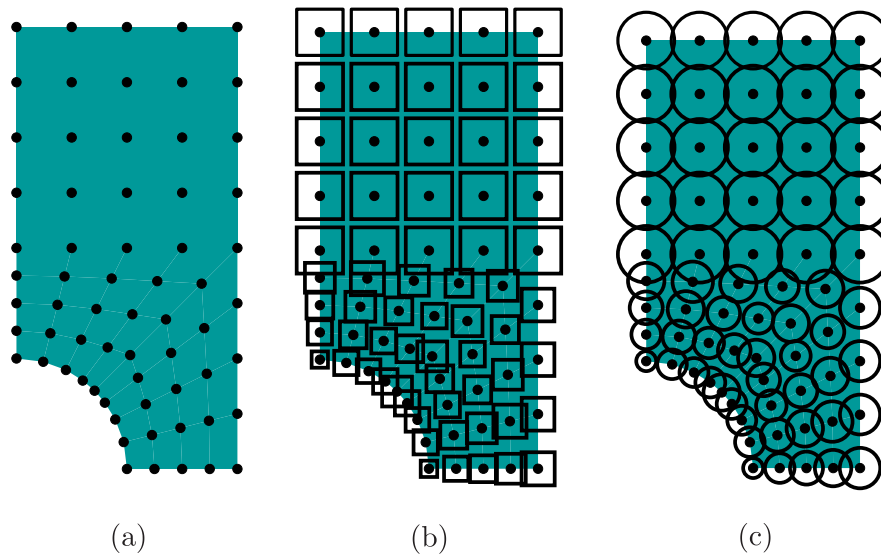


Figure 2.3: Geometry with nodes and the corresponding rectangular and circular domain of influence

domain, a tensor product of one-dimensional form [85, 193] is used. The tensor product weight function of a node i at a point \mathbf{x} can be found as

$$w(\mathbf{x} - \mathbf{x}_i) = \begin{cases} w(r_x) w(r_y) = w_x w_y & \text{for 2D,} \\ w(r_x) w(r_y) w(r_z) = w_x w_y w_z & \text{for 3D,} \end{cases} \quad (2.31)$$

where, e.g. $r_x = \frac{\|x-x_i\|}{(d_{mi})_x}$ and $(d_{mi})_x$ is the domain of influence in x direction. The two-dimensional tensor product weight function derivative can be found as

$$\frac{dw}{dx} = \frac{dw_x}{dx} w_y, \quad (2.32a)$$

$$\frac{dw}{dy} = w_x \frac{dw_y}{dy}, \quad (2.32b)$$

and in three dimensions

$$\frac{dw}{dx} = \frac{dw_x}{dx} w_y w_z, \quad (2.33a)$$

$$\frac{dw}{dy} = w_x \frac{dw_y}{dy} w_z, \quad (2.33b)$$

$$\frac{dw}{dz} = w_x w_y \frac{dw_z}{dz}, \quad (2.33c)$$

where w_x , w_y , w_z and $\frac{dw}{dx}$, $\frac{dw}{dy}$, $\frac{dw}{dz}$ can be found using r_x , r_y and r_z instead of r in Equations (2.27) to (2.30).

2.4.2 Numerical demonstration

In the following cubic spline weight functions and the corresponding MLS shape functions are given in one and two dimensions for demonstration. A linear basis function is used in the calculation of the MLS shape functions.

One dimension

Consider eleven equally distributed nodes in a problem domain of length one unit. Plots of the weight functions and the corresponding derivatives for $d_{max} = 2.0$ are given in Figures 2.4(a) and 2.4(b) respectively while weight functions and derivatives for $d_{max} = 3.0$ are given in Figures 2.5(a) and 2.5(b) respectively. The plots of MLS shape functions and derivatives for $d_{max} = 2.0$ are given in Figures 2.6(a) and 2.6(b) respectively and for $d_{max} = 3.0$ in Figure 2.7(a) and 2.7(b) respectively. It is clear from these plots that the weight function and the corresponding shape function for a node with larger d_{max} is non zero over a larger region. The first spatial derivative of the cubic spline weight functions and the corresponding MLS shape functions are smooth. It is also clear from Figures 2.6(a) and 2.7(a) that MLS shape functions do not possess the Kronecker delta property.

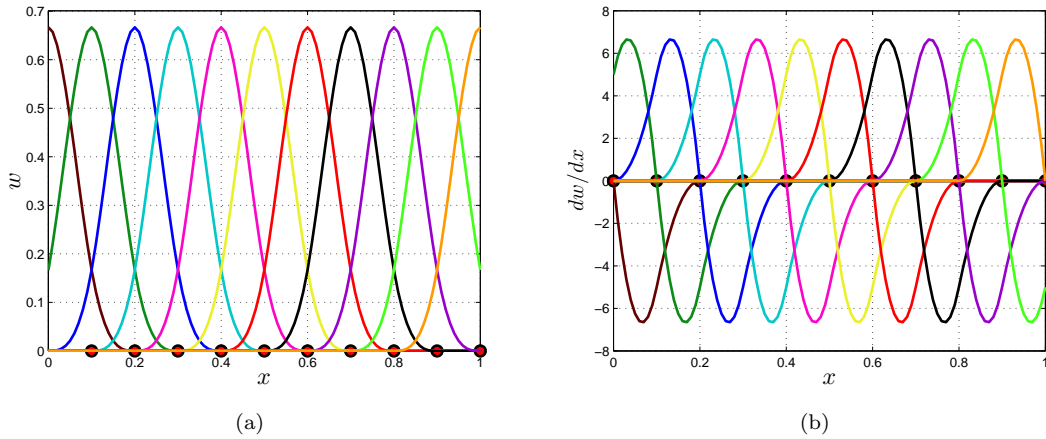
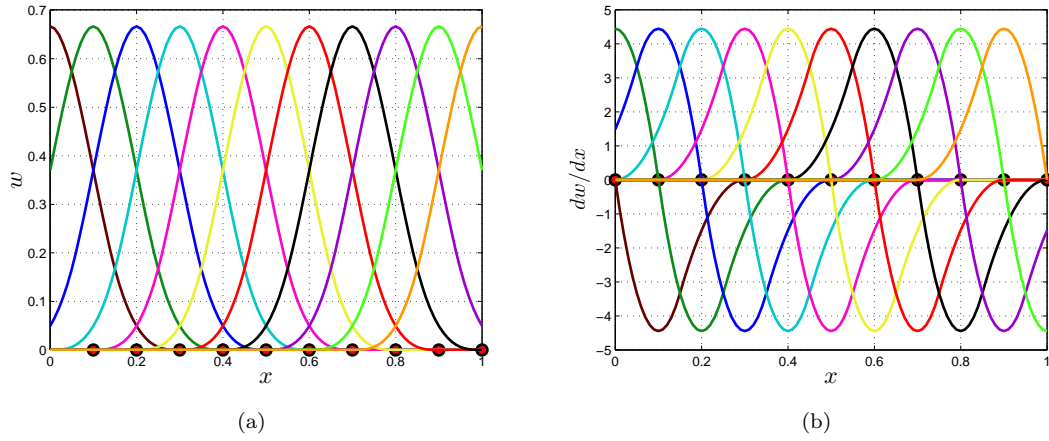
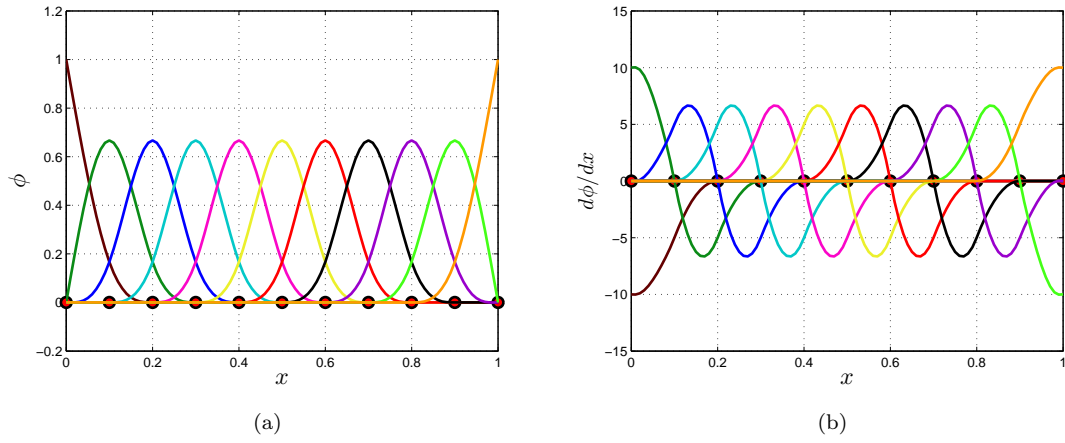
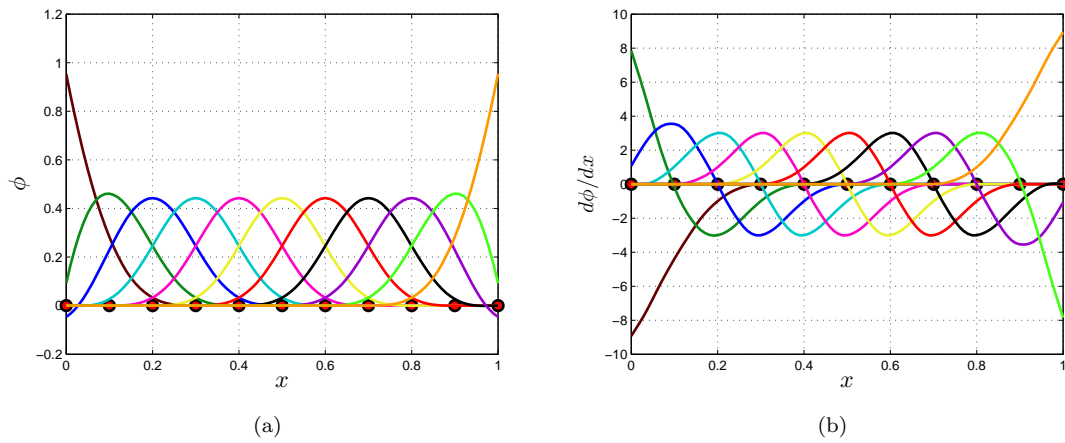


Figure 2.4: 1D cubic spline weight functions and derivatives for $d_{max} = 2.0$

Figure 2.5: 1D cubic spline weight functions and derivatives for $d_{max} = 3.0$ Figure 2.6: 1D MLS shape functions and derivatives for $d_{max} = 2.0$ Figure 2.7: 1D MLS shape functions and derivatives for $d_{max} = 3.0$

Two dimensions

Consider a square domain containing 49 equally distributed nodes. The problem domain with nodes is shown in Figure 2.8(a). The MLS shape functions are calculated using $d_{max} = 3.0$. The weight function

for a node at $\mathbf{x} = [0, 0]$ is shown in Figure 2.8(b) and its derivatives w.r.t. x and y are shown in Figure 2.8(c) and 2.8(d) respectively. The corresponding MLS shape function is shown in Figure 2.9(a) and its derivative w.r.t. x and y are shown in Figure 2.9(b) and 2.9(c) respectively. The smoothness of the first spatial derivative of the weight functions and the corresponding MLS shape functions, and the lack of Kronecker delta property is also clear in 2D case.

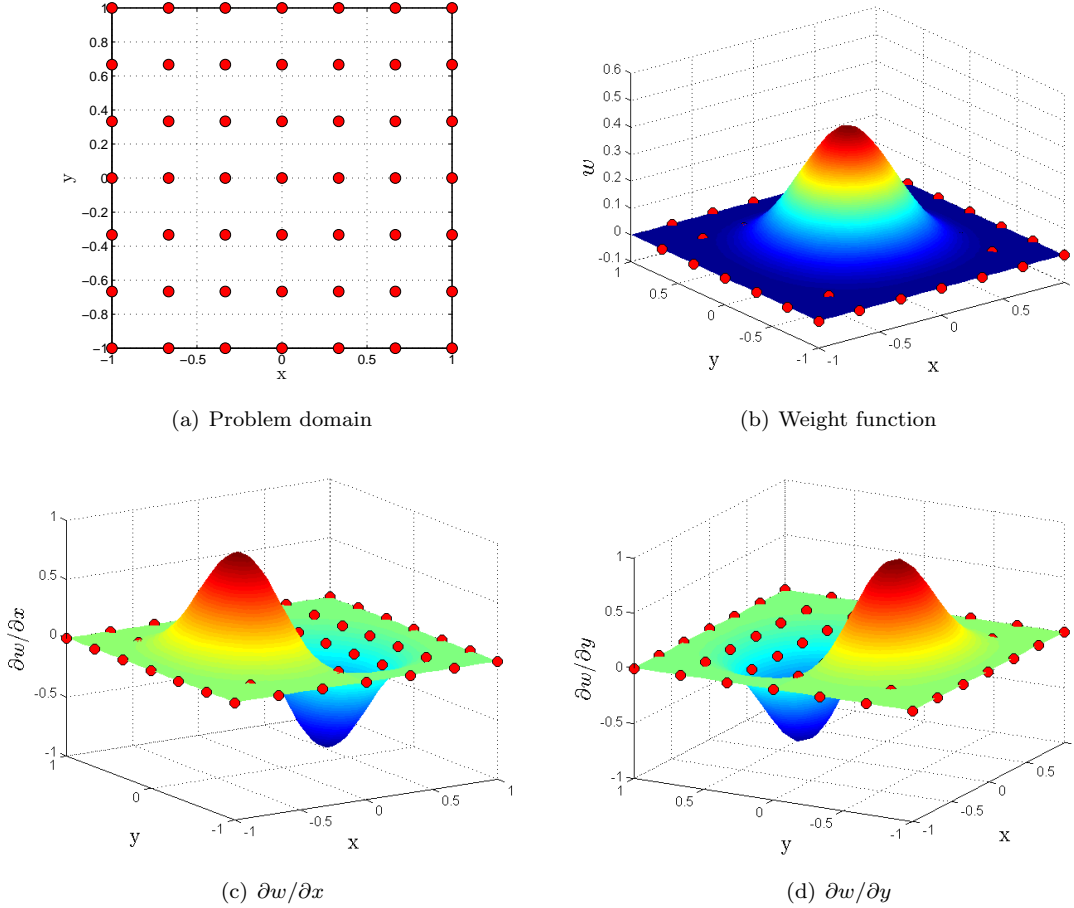
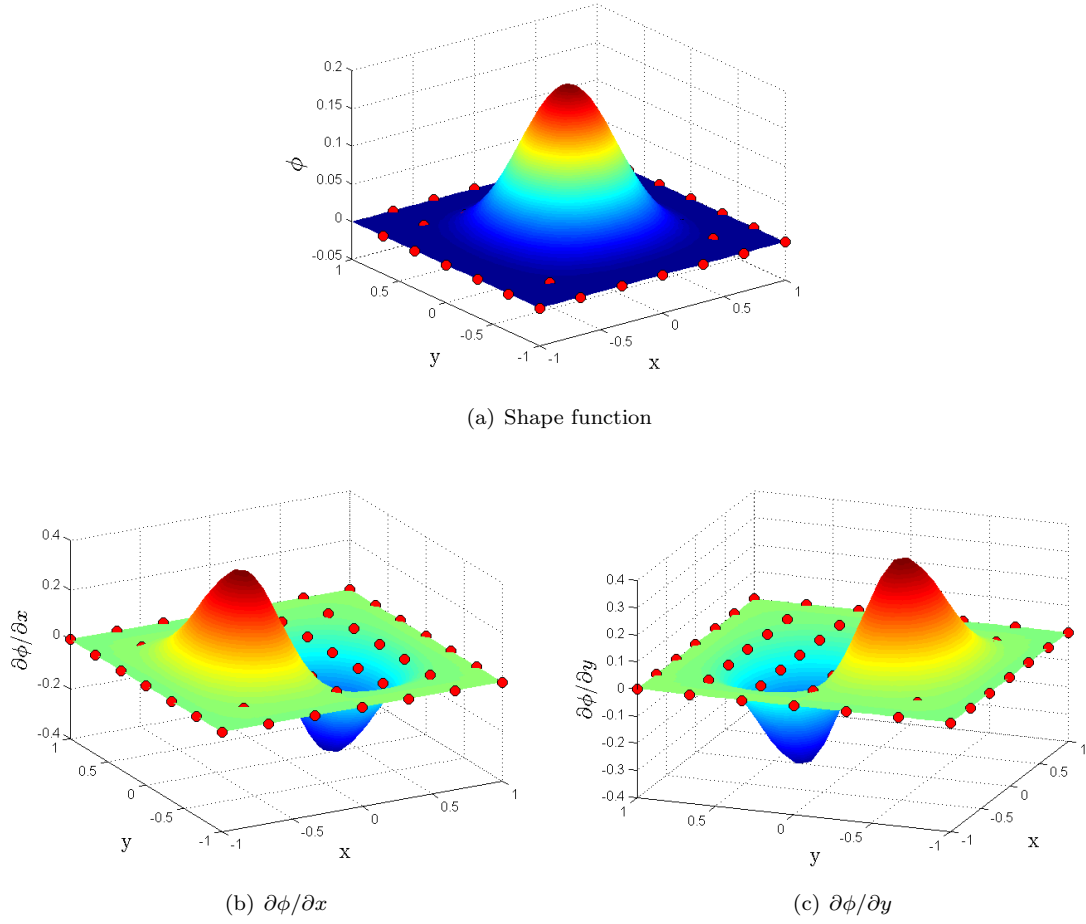


Figure 2.8: 2D problem domain, cubic spline weight function and derivatives for a node at $\mathbf{x} = [0, 0]$

2.5 Element-free Galerkin method

The EFGM is one of the most widely used meshless methods for the solution of boundary value problems. A two-dimensional linear elasto-static formulation of the conventional EFGM with the MLS shape functions is given here, which is straightforward to modify for one and three-dimensional problems. The EFGM was first proposed in [38] as an improvement over its predecessor the diffuse element method [231] and was successfully used in the simulation of linear elasticity and heat conduction problems. As compared to FEM, only nodes were used for the problem discretization, and the MLS shape functions were used as test and trial functions in the weak form of the equilibrium equations. As compared to the diffuse element method a larger number of quadrature points were used, new terms were added in the derivatives of the MLS shape functions and Lagrange multipliers were used for the imposition of the essential boundary conditions. It was also pointed out that as compared to the FEM, the EFGM is more accurate, having a high rate of convergence and no issue of volumetric locking. Nodal stresses were calculated by using the smoothness of the MLS shape function derivatives throughout the problem domain

Figure 2.9: 2D MLS shape function and derivatives for a node at $\mathbf{x} = [0, 0]$

without the standard recovery procedure used in the FEM. The potential advantage of the EFGM in crack growth problems, and in the adaptive analysis were also highlighted. It was also highlighted that the EFGM is computationally expensive as compared to the FEM for the same number of degrees of freedoms and due to the use of the Lagrange multipliers, the dimensions of the final system of equations is increased, and the stiffness matrix is no longer positive definite.

Consider a two-dimensional problem defined in the domain Ω bounded by Γ , as shown in Figure 2.10. The equilibrium equation is written as

$$\nabla^T \boldsymbol{\sigma} + \mathbf{b}_f = 0 \quad \text{in } \Omega, \quad (2.34)$$

where the Cauchy stress vector $\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \tau_{xy} \end{bmatrix}^T$, the body force vector is $\mathbf{b}_f = \begin{bmatrix} b_x & b_y \end{bmatrix}^T$, where b_x and b_y are the body forces in x and y directions respectively and the differential operator

$$\nabla = \begin{Bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{Bmatrix}.$$

The boundary conditions associated with Equation (2.34) are

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_u, \quad (2.35a)$$

$$\boldsymbol{\sigma} \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_t, \quad (2.35b)$$

where Equation (2.35a) is known as displacement, essential or Dirichlet boundary condition and Equation (2.35b) is known as traction, natural, derivative or Neumann boundary condition. Here $\bar{\mathbf{t}}$ is the prescribed traction on the traction boundary Γ_t and $\bar{\mathbf{u}}$ is the prescribed displacement on the essential boundaries Γ_u and \mathbf{n} is a unit normal. The constrained Galerkin weak form of the equilibrium Equation (2.34) subject

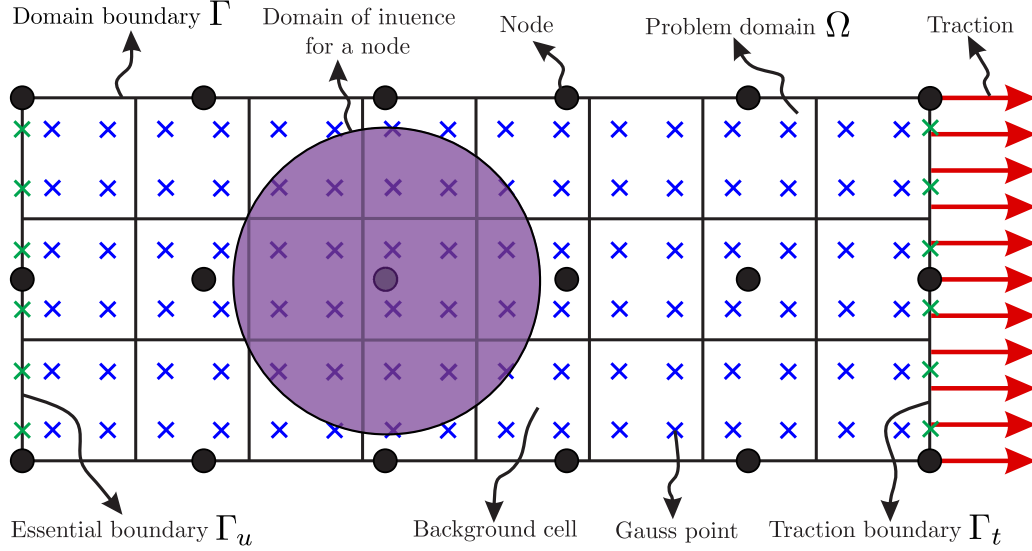


Figure 2.10: The EFGM problem discretization

to boundary conditions 2.35a and 2.35b defined over the global domain is written as

$$\int_{\Omega} \delta (\nabla \mathbf{u})^T \mathbf{D} (\nabla \mathbf{u}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b}_f d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma - \int_{\Gamma_u} \delta \boldsymbol{\lambda}^T (\mathbf{u} - \bar{\mathbf{u}}) d\Gamma - \int_{\Gamma_u} \delta \mathbf{u}^T \boldsymbol{\lambda} d\Gamma = 0, \quad (2.36)$$

all the terms in Equation (2.36) are defined in §2.3. The MLS shape functions are used for the approximation of the field variable and Lagrange multipliers are used to impose the essential boundary conditions. After discretizing the problem with a set of nodes, displacement at a point of interest \mathbf{x} is written as

$$\mathbf{u}^h(\mathbf{x}) = \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} = \sum_{i=1}^n \begin{bmatrix} \phi_i & 0 \\ 0 & \phi_i \end{bmatrix} \begin{Bmatrix} u_{xi} \\ u_{yi} \end{Bmatrix} = \sum_{i=1}^n \phi_i \mathbf{u}_i, \quad (2.37)$$

where $\mathbf{u}^h(\mathbf{x})$ is an approximation of the displacements at a point \mathbf{x} , n is the number of nodes in the support of point \mathbf{x} , ϕ_i is a matrix of the MLS shape functions for node i at a point \mathbf{x} and \mathbf{u}_i are known as fictitious nodal values or nodal parameters. The approximation for the vector of Lagrange multipliers $\boldsymbol{\lambda}$ at a point of interest \mathbf{x} on the essential boundary is written as

$$\boldsymbol{\lambda}(\mathbf{x}) = \begin{Bmatrix} \lambda_x \\ \lambda_y \end{Bmatrix} = \sum_{i=1}^{n_\lambda} \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} \begin{Bmatrix} \lambda_{xi} \\ \lambda_{yi} \end{Bmatrix} = \sum_{i=1}^{n_\lambda} \mathbf{N}_i \boldsymbol{\lambda}_i, \quad (2.38)$$

Where n_λ are the number of nodes on the essential boundary used in this interpolation and \mathbf{N}_i is a matrix of shape functions for nodes i on the essential boundaries at a point of interest \mathbf{x} and $\boldsymbol{\lambda}_i$ are the nodal values of the Lagrange multipliers. The conventional FEM, Lagrangian interpolation is an option to be used for N in Equation (2.38). Using Equations (2.37) and (2.38) in (2.36) and after simplification, the

final discrete system of linear equation is written as

$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{q} \end{Bmatrix}, \quad (2.39)$$

where

$$\mathbf{K}_{ij} = \int_{\Omega} \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j d\Omega, \quad (2.40)$$

$$\mathbf{G}_{ik} = - \int_{\Gamma_u} \phi_i \mathbf{N}_k d\Gamma, \quad (2.41)$$

$$\mathbf{f}_i = \int_{\Gamma_t} \phi_i \bar{t} d\Gamma + \int_{\Omega} \phi_i \mathbf{b}_f d\Omega, \quad (2.42)$$

$$\mathbf{q}_k = - \int_{\Gamma_u} \mathbf{N}_k \bar{u} d\Gamma, \quad (2.43)$$

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial \phi_i}{\partial x} & 0 \\ 0 & \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial y} & \frac{\partial \phi_i}{\partial x} \end{bmatrix}, \quad (2.44)$$

$$\mathbf{N}_k = \begin{bmatrix} N_k & 0 \\ 0 & N_k \end{bmatrix}. \quad (2.45)$$

To perform the integrations in Equations (2.40) to (2.43) numerically, the problem domain Ω , traction boundary Γ_t and essential boundary Γ_u are divided into a number of non-overlapping background cells as shown in Figure 2.10. Conventional Gauss quadrature is used in the background cells for the numerical integration, e.g. the integration of any integrand \mathbf{G} over the global domain Ω is written as

$$\int_{\Omega} \mathbf{G} d\Omega = \sum_{j=1}^{n_c} \int_{\Omega_j} \mathbf{G} d\Omega_j = \sum_{j=1}^{n_c} \sum_{k=1}^{n_g} w_k \mathbf{G}(\mathbf{x}_k) |\mathbf{J}_{kj}|, \quad (2.46)$$

where n_c is the total number of background cells, Ω_j is the domain associated with each background cell, i.e. $\Omega = \sum_{j=1}^{n_c} \Omega_j$, n_g is the number of Gauss points in each background cell, w_k and \mathbf{J}_{kj} are the weight and Jacobian associated with each Gauss point respectively while $\mathbf{G}(\mathbf{x}_k)$ is the value of the integrand at gauss point \mathbf{x}_k . The number of Gauss points in each background cell should be more than that used in the conventional FEM due to the non-polynomial nature of the MLS shape functions. In [38] it is suggested that $n_Q \times n_Q$ Gauss points should be used in each background cell, where

$$n_Q = \sqrt{m} + 2 \quad (2.47)$$

and m is the number of nodes in a background cell. For a two-dimensional problem, it is also suggested in [196] to use $\widetilde{n}_Q > \frac{2}{3}\widetilde{n}$ and $\widetilde{n}_Q = (3 \sim 9)\widetilde{n}$ where \widetilde{n}_Q and \widetilde{n} are the total number of Gauss points and unfixed nodes in the global problem domain respectively.

2.6 Maximum entropy shape functions

Like the MLS shape functions, max-ent shape functions are an alternative way to calculate the shape functions for meshless methods using the concept of informational entropy [157, 269] and the principle of

max-ent [142, 143]. These shape functions were first introduced in [287] for the construction of polygonal interpolants. In [269] informational entropy was referred to as a lack of knowledge or uncertainty, while in [142, 143], the principle of max-ent was introduced, according to which maximizing the informational entropy, will result in a least biased or feasible solution for the probability distribution in the case of insufficient data. This idea was used in [287] to solve the under-determined system of equations that appear in the case of the polygonal interpolants for the case ($n > 3$) with constant and linear reproducing constraints, where n is the number of sides of the polygon. Shape functions derived for the polygonal interpolation in [287] are not ideal for meshless methods as they extend to the global problem domain. The non-local and non-interpolating characteristics of these shape functions are highlighted in [12] where the (more useful) local max-ent formulations are introduced and incorporated into meshless modelling of linear and nonlinear elasticity. The weak Kronecker delta property of these shape functions was also observed, and its correspondence with the MLS was also highlighted. Compact support shape functions are derived using Gaussian weight functions (or priors) in [12], work which is extended in [289] for any weight function (or generalized prior). First-order consistent max-ent shape functions [289] are then extended to second order in [77] and higher order in [110] and max-ent is used in [263] for the automatic calculation of the nodal domain of influence within a meshless method. Other recent examples of the use of max-ent in meshless methods can be found in [219, 244, 245]. Below, the background to the theory of max-ent is outlined followed by an explanation of how useful shape functions are derived.

2.6.1 Information and informational entropy

Information theory was first developed in the context of communication engineering [10]. Information has a special meaning in probability theory, which is different from the word's ordinary usage and is related to the element of surprise. In [157] the concept of Information is termed as "uncertainty". The information is inversely proportional to the probability. Consider an experiment, the outcome of which is events A_1 and A_2 with known probabilities. Take two alternatives:

$$\begin{pmatrix} A_1 & A_2 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} A_1 & A_2 \\ 0.99 & 0.01 \end{pmatrix}.$$

1. The first is more uncertain and has more information, e.g. when event A_1 is realized.
2. The second is almost sure and has therefore, less information.

Consider mutually independent events x_1, x_2, \dots, x_n within a sample space Ω with probabilities p_1, p_2, \dots, p_n respectively. The quantity for measuring the amount of information or uncertainty or informational entropy for a finite scheme is given as [157]

$$H(p_1, p_2, \dots, p_n) = - \sum_{i=1}^n p_i \log p_i, \quad (2.48)$$

with the assumption of $p_i \log p_i = 0$ if $p_i = 0$. The characteristics of informational entropy H are clear from Equation (2.48), which are given in [157] and also summarized in [12] and [289] as follows:

1. $H(p) \geq 0$.
2. $H(p_1, p_2, \dots, p_n) = 0$ for a complete certainty, i.e. if only one of the p_i is one and all other zeros.
3. H is maximum for the most uncertain or equally likely outcomes, i.e. if $p_i = 1/n$ for ($i = 1, 2, \dots, n$).
4. $H(p_1, p_2, \dots, p_n) = H(p_1, p_2, \dots, p_n, 0)$, i.e. adding an event of zero probability has no effect on entropy.

5. $0 \leq H \leq \log(n)$.
6. For two independent schemes A and B , $H(AB) = H(A) + H(B)$.
7. For two dependent schemes A and B , $H(AB) = H(A) + H_A(B)$ where $H_A(B)$ is the entropy of B after realization of A and $H_A(B) \leq H(B)$.

2.6.2 Polygonal interpolants or global max-ent shape functions

Consider mutually independent events x_1, x_2, \dots, x_n within a sample space Ω with unknown probabilities p_1, p_2, \dots, p_n respectively. As $P(\Omega) = 1$ and $p_i > 0$, i.e. the nonnegative probabilities should satisfy the condition of $\sum_{i=1}^n p_i = 1$ and assume m observations, with $m < (n - 1)$ where the expectation $\langle g_r(x) \rangle$ of a function $g_r(x)$ is known, this will make the system under-determined. In this scenario, the most likely probability distribution is obtained by using the Jaynes' principle of max-ent [142, 143]

$$\text{maximize} \left([H(p_1, p_2, \dots, p_n)] = - \sum_{i=1}^n p_i \log p_i \right), \quad (2.49)$$

subject to constraints

$$\sum_{i=1}^n p_i = 1, \quad \sum_{i=1}^n p_i g_r(x_i) = \langle g_r(x) \rangle, \quad (2.50)$$

where $\langle g_r(x) \rangle$ is the expectation of $g_r(x)$. The idea for using the principle of max-ent is that the informational entropy of the probability distribution which will satisfy the given constraints will exist in the vicinity of the max-ent and due to this reason the max-ent distribution is the one which is least biased and has the maximum chance to occur [287].

Making an analogy between the probabilities and two-dimensional polygonal interpolants, consider a polygon with nodes located at the corners with coordinates (x_i, y_i) , while the coordinates of any point in the interior is (x, y) . Consider a shape function associated with each node $\phi_i \geq 0$ with the assumption that the ϕ_i will reproduce a constant and linear function exactly as

$$\sum_{i=1}^n \phi_i = 1, \quad \sum_{i=1}^n \phi_i x_i = x, \quad \sum_{i=1}^n \phi_i y_i = y. \quad (2.51)$$

Equation (2.51) can also be written as

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{Bmatrix} = \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix}. \quad (2.52)$$

For a triangular case with $(n = 3)$ Equation (2.52) has a unique solution but for a case when $(n > 3)$, Equation (2.52) is under-determined, i.e. there are more unknown ϕ_i than the available constraints (2.51). In this case the principle of max-ent is used to find the unknown shape functions ϕ_i , so Equation (2.49) are written in terms of the unknown shape functions as

$$\text{maximize} \left([H(\phi_1, \phi_2, \dots, \phi_n)] = - \sum_{i=1}^n \phi_i \log \phi_i \right), \quad (2.53)$$

subject to constant and linearly reproducing constraints (2.51).

Equations (2.53) and (2.51) constitute an optimization problem which can be solved using the method of Lagrange multipliers. Consider the Lagrange multipliers $\lambda_r (r = 0, 1, 2)$ ¹ associated with the three constraints in Equation (2.51) and using the variation equal to zero

$$\delta \left[\sum_{i=1}^n -\phi_i \log \phi_i + \lambda_0 \left(1 - \sum_{i=1}^n \phi_i \right) + \lambda_1 \left(x - \sum_{i=1}^n \phi_i x_i \right) + \lambda_2 \left(y - \sum_{i=1}^n \phi_i y_i \right) \right] = 0. \quad (2.54)$$

After simplification and by using $\lambda_0 = \log(Z) - 1$, where Z is known as the partition function in statistical mechanics, the equation for the shape functions is written as

$$\phi_i = \frac{e^{-\lambda_1 x_i - \lambda_2 y_i}}{Z}, \quad Z(\lambda_1, \lambda_2) = \sum_{i=1}^n e^{-\lambda_1 x_i - \lambda_2 y_i}. \quad (2.55)$$

The constraints in Equation (2.51) are written using ϕ_i from Equation (2.55)

$$f_1(\lambda_1, \lambda_2) = \frac{\sum_{i=1}^n e^{-\lambda_1 x_i - \lambda_2 y_i} x_i}{Z} - x, \quad f_2(\lambda_1, \lambda_2) = \frac{\sum_{i=1}^n e^{-\lambda_1 x_i - \lambda_2 y_i} y_i}{Z} - y. \quad (2.56)$$

Equation (2.56) can also be written as

$$-\frac{\partial \lambda_0}{\partial \lambda_r} = -\frac{\partial(\log Z)}{\partial \lambda_r} = \langle g_r(x) \rangle, \quad (r = 1, 2), \quad (2.57)$$

where $\langle g_1(x, y) \rangle = x$ and $\langle g_2(x, y) \rangle = y$. By solving the nonlinear set of Equations (2.56) for λ_1 and λ_2 , the shape functions can be obtained from Equation (2.55). Now using Equations (2.53) and (2.55) the maximum informational entropy is $H_{max} = \log(Z) + \lambda_1 x + \lambda_2 y$. Practically, due to numerical complications [287] the primal problem of maximization is recast into a dual problem of minimization. By letting $\tilde{x}_i = x_i - x$ and $\tilde{y}_i = y_i - y$, Equations (2.51), (2.55) and 2.56 are written as

$$\sum_{i=1}^n \phi_i \tilde{x}_i = 0, \quad \sum_{i=1}^n \phi_i \tilde{y}_i = 0, \quad (2.58)$$

$$\phi_i = \frac{e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i}}{\tilde{Z}}, \quad \tilde{Z}(\lambda_1, \lambda_2) = \sum_{i=1}^n e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i}, \quad (2.59)$$

and

$$f_r(\lambda_1, \lambda_2) = \frac{\partial(\log \tilde{Z})}{\partial \lambda_r} = 0, \quad (r = 1, 2), \quad (2.60a)$$

$$f_1(\lambda_1, \lambda_2) = -\frac{\sum_{i=1}^n e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i} \tilde{x}_i}{\tilde{Z}}, \quad (2.60b)$$

$$f_2(\lambda_1, \lambda_2) = -\frac{\sum_{i=1}^n e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i} \tilde{y}_i}{\tilde{Z}}. \quad (2.60c)$$

In this case $H_{max} = \log(\tilde{Z})$. In this dual problem the Lagrange multipliers are determined as a set that minimizes a convex potential function $F(\lambda_1, \lambda_2) = H_{max}$. This problem is solved using an iterative solution method such as Newton's method, the solution of which includes an initial guess for λ_1 and λ_2 .

¹Not the same as used for the imposition of the essential boundary conditions

The solution at the $(k+1)^{th}$ iteration is given as

$$\lambda_r^{k+1} = \lambda_r^k + \Delta \lambda_r^k, \quad \Delta \lambda^k = -\mathbf{A}^{-1} \nabla F, \quad (2.61)$$

where ∇F is the gradient of F and is given as

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial \lambda_1} \\ \frac{\partial F}{\partial \lambda_2} \end{bmatrix} = \begin{bmatrix} -\frac{\sum_{i=1}^n e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i} \tilde{x}_i}{\tilde{Z}} \\ -\frac{\sum_{i=1}^n e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i} \tilde{y}_i}{\tilde{Z}} \end{bmatrix}. \quad (2.62)$$

and \mathbf{A} is a Hessian matrix and is given as

$$\mathbf{A} = \nabla^2 F = \begin{bmatrix} \langle \tilde{x}^2 \rangle - \langle \tilde{x} \rangle^2 & \langle \tilde{x} \tilde{y} \rangle - \langle \tilde{x} \rangle \langle \tilde{y} \rangle \\ \langle \tilde{x} \tilde{y} \rangle - \langle \tilde{x} \rangle \langle \tilde{y} \rangle & \langle \tilde{y}^2 \rangle - \langle \tilde{y} \rangle^2 \end{bmatrix}, \quad (2.63)$$

where $\langle \tilde{x} \rangle = \sum_{i=1}^n \phi_i \tilde{x}_i$. The convergence criterion is $\|\nabla F\|^{(k)} < \varepsilon$, where ε is a very small number $10^{-6} \sim 10^{-10}$.

2.6.3 Local maximum entropy shape functions

Local max-ent approximation formulations are summarized in [289] using the Shannon-Jaynes general form of information entropy or relative entropy functional. The general form of entropy or the relative entropy, is written as

$$H(p, m) = - \sum_{i=1}^n p_i \log \left(\frac{p_i}{m_i} \right), \quad (2.64)$$

where m_i is a prior distribution that will be used in the calculation of unknown probabilities p_i . In the case of shape functions the following applies

$$\text{maximize} \left(H(\phi, w) = - \sum_{i=1}^n \phi_i \log \left(\frac{\phi_i}{w_i} \right) \right), \quad (2.65)$$

subject to constraints

$$\sum_{i=1}^n \phi_i = 1, \quad \sum_{i=1}^n \phi_i \tilde{x}_i = 0, \quad \sum_{i=1}^n \phi_i \tilde{y}_i = 0. \quad (2.66)$$

The shape functions are written as

$$\phi_i = \frac{\tilde{Z}_i}{\tilde{Z}}, \quad \tilde{Z}_i = w_i e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i}, \quad \tilde{Z} = \sum_{j=1}^n \tilde{Z}_j, \quad (2.67)$$

where w_i is the prior distribution, i.e. any weight function. The uniform prior is used in [287], Gaussian prior is used in [12] and generalized prior is used in [288, 289]. The expressions for the shape function derivatives are [288]

$$\nabla \phi_i = \phi_i \left(\nabla f_i - \sum_{i=1}^n \phi_i \nabla f_i \right), \quad (2.68)$$

where

$$\nabla f_i = \frac{\nabla w_i}{w_i} + \lambda + \tilde{x}^i [\mathbf{A}^{-1} - \mathbf{A}^{-1} \cdot B], \quad B = \sum_{k=1}^n \phi_k \tilde{x}^k \otimes \frac{\nabla w_k}{w_k}, \quad (2.69)$$

where \otimes is the dyadic product of two vectors, i.e. for any two vectors \mathbf{A} and \mathbf{B} , $\mathbf{A} \otimes \mathbf{B} = \mathbf{A}\mathbf{B}^T$

2.6.4 Numerical demonstration

One- and two-dimensional numerical implementation of the determination of max-ent shape functions is now presented. Cubic spline weight functions with $d_{max} = 3.0$ is used in the calculation of max-ent shape function.

One dimension

Consider the same 1D domain and discretization as given in §2.4.2. The convergence path of the Newton method to determine the Lagrange multipliers for three points $x = 0.4, 0.93, 0.02$ are shown in Figures 2.11(a), 2.11(b) and 2.11(c) respectively, while the step by step convergence of the Lagrange multipliers with $\varepsilon = 10^{-6}$ are given in Table 2.1. It is obvious from Figures 2.11(a), 2.11(b) and 2.11(c) and Table 2.1 that if the point of interest lies in the centre of the nodes in support, e.g. $x = 0.4$ then the Newton method will converge in one iteration but when the point approaches the boundary, then the number of iterations increases.

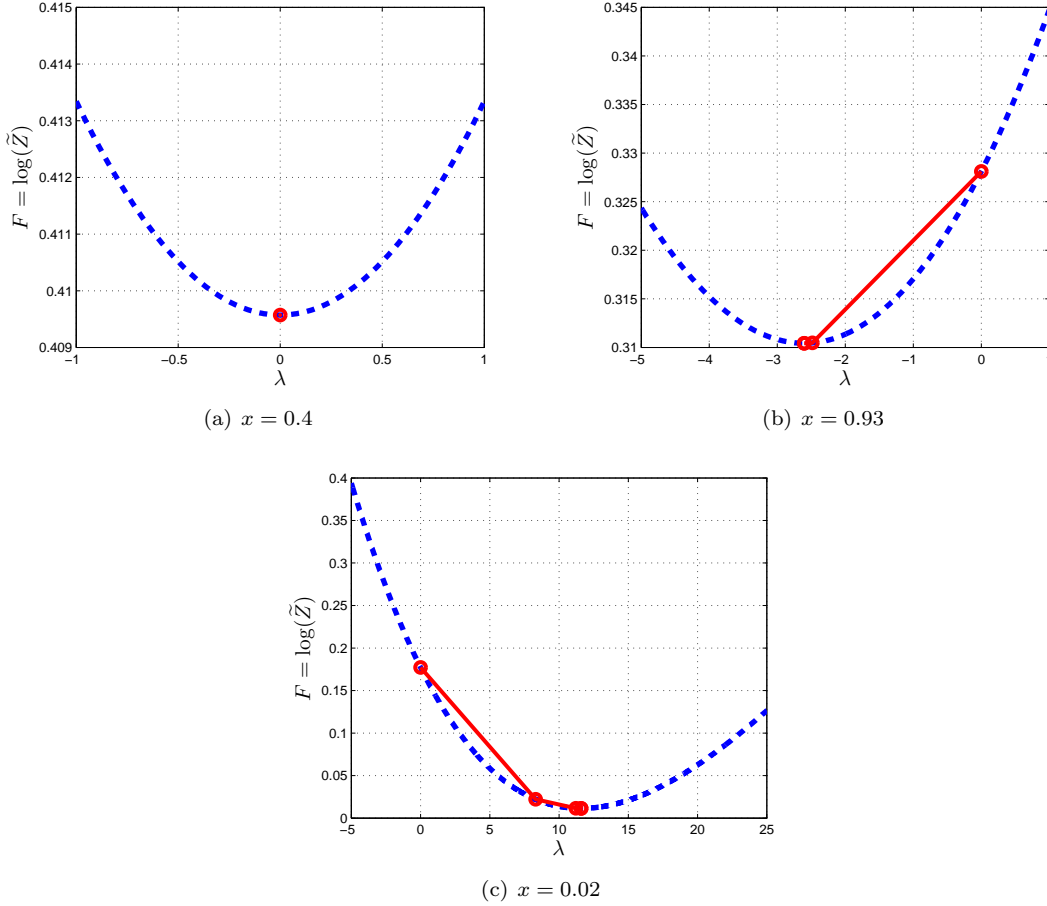


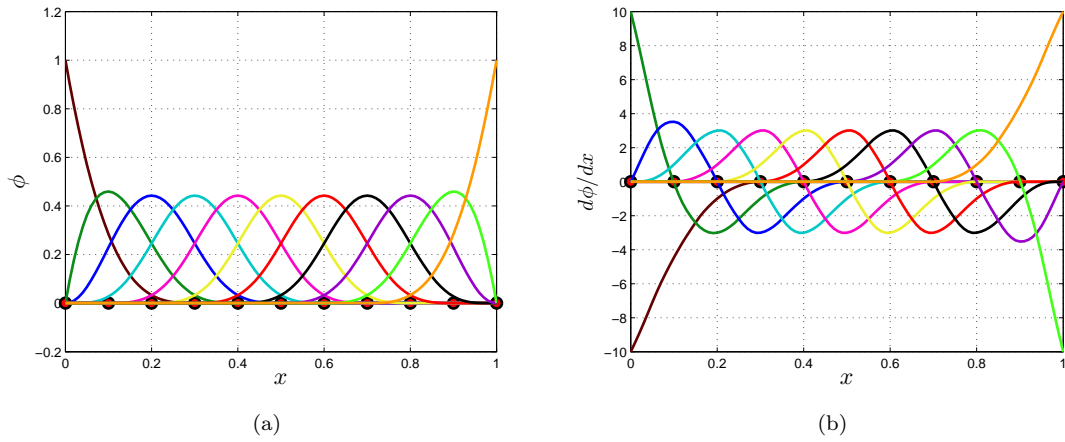
Figure 2.11: Convergence path of the Newton's method for different points in one-dimensional domain for max-ent shape functions

Figure 2.12(a) shows the local max-ent shape functions, while Figure 2.12(b) shows the corresponding shape function derivatives. In Figure 2.12(a) the shape functions of all the internal nodes vanish at the boundary, which is different from the MLS shape functions, where the shape functions for the internal nodes are non zero at the boundary. Due to this weak Kronecker delta property of the max-ent shape

Iteration	$x = 0.4$		$x = 0.93$		$x = 0.02$	
	λ	ε	λ	ε	λ	ε
1	0	7.1869e-17	0	0.0138	0	0.0328
2	-	-	-2.4830	6.1102e-4	8.3118	0.0068
3	-	-	-2.6039	1.6048e-6	11.2180	7.0147e-4
4	-	-	-2.6042	1.1229e-11	11.5938	1.0503e-5
5	-	-	-	-	11.5996	2.4644e-9

Table 2.1: Convergence of Newton's method for different points in 1D problem domain

functions the imposition of the essential boundary condition is as straightforward in a meshless method as in the case of the FEM. The smoothness of the first derivative of max-ent shape functions can be seen in 2.12(b).

Figure 2.12: 1D max-ent shape functions and derivatives for $d_{max} = 3.0$

Two dimensions

Consider the same two-dimensional square domain and discretization as shown in Figure 2.8(a) given in §2.4.2. The convergence of the Newton method for four different points of interest $\mathbf{x} = (0.0, 0.0)$, $\mathbf{x} = (0.7, -0.7)$, $\mathbf{x} = (-0.85, 0.9)$ and $\mathbf{x} = (-0.99, -0.8)$ are given in the Figures 2.13(a), 2.13(b), 2.13(c) and 2.13(d) respectively, while the step by step convergence for all these points are also shown in Table 2.2. It is also clear in this case that if the point of interest is in the centre of the nodes in support as in the case $\mathbf{x} = (0.0, 0.0)$ then only one Newton iteration is required but the number of Newton iteration increases when the point approaching the boundary. The max-ent shape function for a node at $\mathbf{x} = [0, 0]$

Itr	$\mathbf{x}=(0.0, 0.0)$			$\mathbf{x}=(0.7, -0.7)$			$\mathbf{x}=(-0.85, 0.9)$			$\mathbf{x}=(-0.99, -0.8)$		
	λ_1	λ_2	ε	λ_1	λ_2	ε	λ_1	λ_2	ε	λ_1	λ_2	ε
1	0	0	4.44e-18	0	0	0.04	0	0	0.11	0	0	0.14
2	-	-	-	-0.43	0.43	8.28e-4	1.38	-1.98	0.01	3.61	0.96	0.04
3	-	-	-	-0.44	0.44	3.94e-7	1.56	-2.44	7.76e-4	6.26	1.02	0.01
4	-	-	-	-	-	-	1.57	-2.47	2.42e-6	8.01	1.03	0.003
5	-	-	-	-	-	-	1.57	-2.47	2.47e-11	8.72	1.03	3.15e-4
6	-	-	-	-	-	-	-	-	-	8.81	1.03	4.70e-6
7	-	-	-	-	-	-	-	-	-	8.81	1.03	1.08e-9

Table 2.2: Convergence of Newton's method for different points in 2D problem domain

is shown in Figure 2.14(a) and its derivatives w.r.t. x and y are shown in Figure 2.14(b) and 2.14(c)

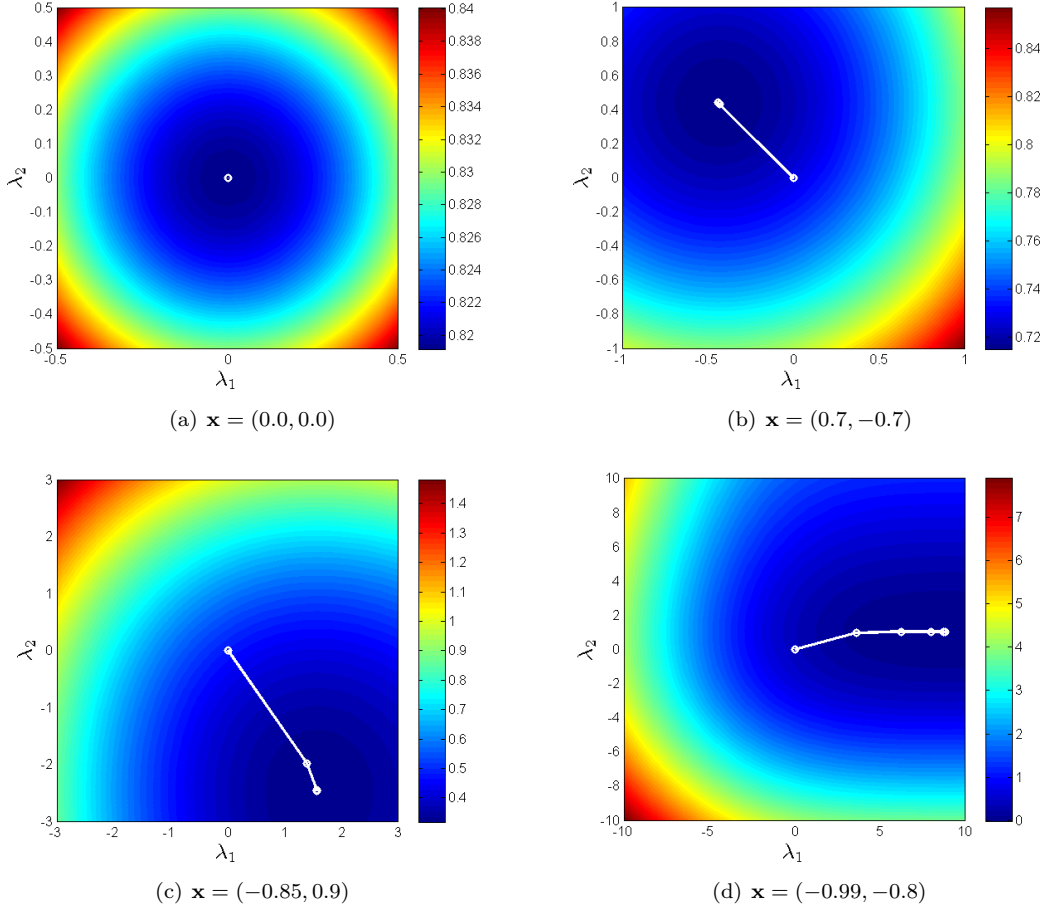


Figure 2.13: Convergence path of the Newton's method for different points in two-dimensional domain for max-ent shape functions

respectively. The weak Kronecker delta property and smoothness of the first spatial derivatives of the max-ent shape functions can also be observed in these figures.

2.7 Element-free Galerkin method with max-ent shape functions

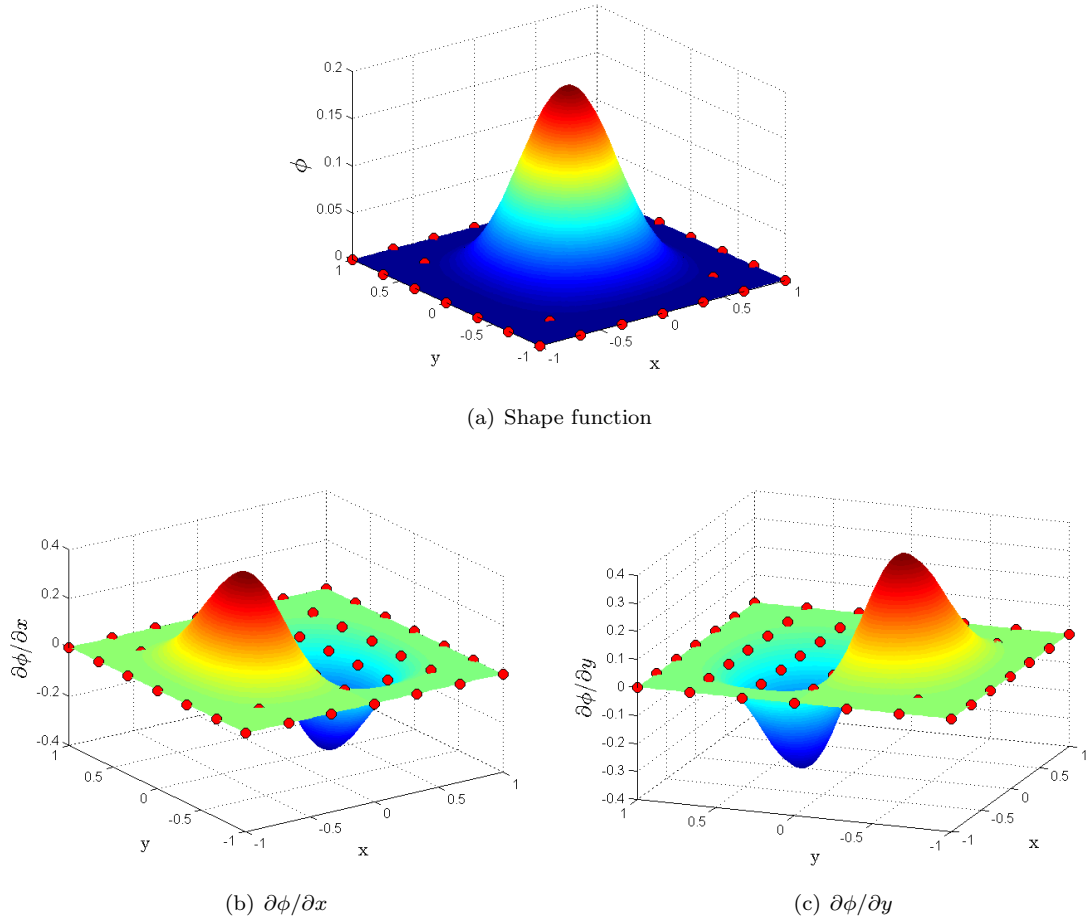
By using max-ent shape functions in §2.5, Equation (2.36) is written as

$$\int_{\Omega} \delta (\nabla \mathbf{u})^T \mathbf{D} (\nabla \mathbf{u}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b}_f d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma = 0. \quad (2.70)$$

The max-ent shape functions possess the weak Kronecker delta property and in Equation (2.70) there is no need to use Lagrange multipliers, i.e. the last two terms in Equation (2.36) to enforce the essential boundary conditions. The essential boundary conditions are implemented directly as in the case of the FEM. The final discretized system of equations is

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (2.71)$$

where \mathbf{K} and \mathbf{f} are defined in Equations (2.40) and (2.42) respectively.

Figure 2.14: 2D max-ent shape function and derivatives for a node at $\mathbf{x} = [0, 0]$

2.8 Numerical examples

One- two- and three-dimensional linear elastic problems, most with analytical solutions are now presented to demonstrate the implementation and performance of the EFGM with both the MLS and max-ent shape functions. In all problems, cubic spline weight functions are used in the calculation of the both MLS and max-ent shape functions while a linear basis function is used in the calculation of MLS shape functions. For convergence studies the L_2 norm of error in displacement $\|e_u\|$ and energy $\|e\|$, suggested in [332] are used, i.e.

$$\|e_u\| = \left(\int_{\Omega} \mathbf{e}_u^T \mathbf{e}_u d\Omega \right)^{\frac{1}{2}}, \quad (2.72a)$$

$$\|e\| = \left(\int_{\Omega} \mathbf{e}_\varepsilon^T \mathbf{D} \mathbf{e}_\varepsilon d\Omega \right)^{\frac{1}{2}} = \left(\int_{\Omega} \mathbf{e}_\sigma^T \mathbf{D}^{-1} \mathbf{e}_\sigma d\Omega \right)^{\frac{1}{2}}. \quad (2.72b)$$

where \mathbf{e}_u , \mathbf{e}_ε and \mathbf{e}_σ are errors in displacements, strains and stresses at a point \mathbf{x} and are written as

$$\mathbf{e}_u = \mathbf{u}_{exact} - \mathbf{u}_{num}, \quad (2.73a)$$

$$\mathbf{e}_\varepsilon = \varepsilon_{exact} - \varepsilon_{num}, \quad (2.73b)$$

$$\mathbf{e}_\sigma = \sigma_{exact} - \sigma_{num}, \quad (2.73c)$$

where \mathbf{u}_{exact} , ε_{exact} and σ_{exact} are the exact displacements, strains and stresses at a point \mathbf{x} , while \mathbf{u}_{num} , ε_{num} and σ_{num} are numerical displacements, strains and stresses at the same point respectively.

One-dimensional bar

Consider a one-dimensional bar of unit length subjected to body force and fixed at a point $x = 0$. The geometry, coordinate system, loading and boundary conditions are shown in Figure 2.15. For a

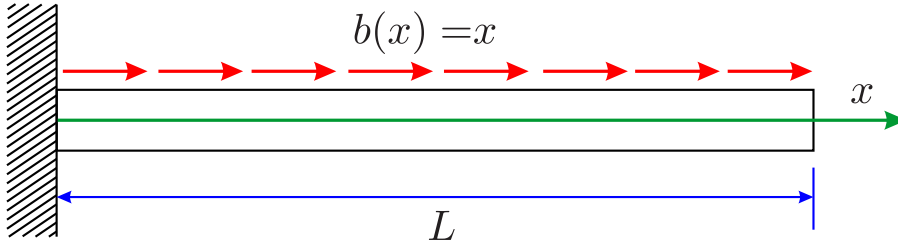


Figure 2.15: Geometry and loading for 1D bar problem

one-dimensional problem, Equations (2.40) to (2.43) are written as

$$K_{ij} = \int_0^L \phi_{i,x} E \phi_{j,x} dx, \quad (2.74)$$

$$G_{ik} = -\phi_k|_{\Gamma_{ui}}, \quad (2.75)$$

$$f_i = \phi_i|_{\Gamma_t} + \int_0^L \phi_i b dx, \quad (2.76)$$

$$q_k = -\bar{u}_k, \quad (2.77)$$

where E is the modulus of elasticity. In 1D case, the essential and traction boundaries are consisting of one node, each and the boundary integrations are replaced by the corresponding nodal values in Equations (2.75) to (2.77).

The analytical solutions for the displacement and stress field for this problem are given as

$$u(x) = \frac{1}{E} \left(\frac{1}{2}x - \frac{x^3}{6} \right), \quad (2.78a)$$

$$\sigma(x) = \left(\frac{1-x^2}{2} \right). \quad (2.78b)$$

The problem is discretized with eleven equally distributed nodes, ten background cells, four Gauss points in each background cell and one Gauss point on the essential boundary with $E = 1.0$ and $d_{max} = 2.5$. The problem is solved using both MLS and max-ent shape functions, in which Lagrange multipliers are used to impose the essential boundary conditions with MLS shape functions, while the direct method is used in the case of max-ent shape functions. Due to the direct imposition of the essential boundary

conditions in the EFGM with max-ent shape functions, there is no need to use Gauss points on the essential boundary. Figures 2.16(a) and 2.16(b) shows the comparison between the EFGM with the MLS and max-ent shape functions and analytical solutions for displacement and stress along the length of the bar, which are in excellent agreement. For convergence studies the same problem is solved with the same

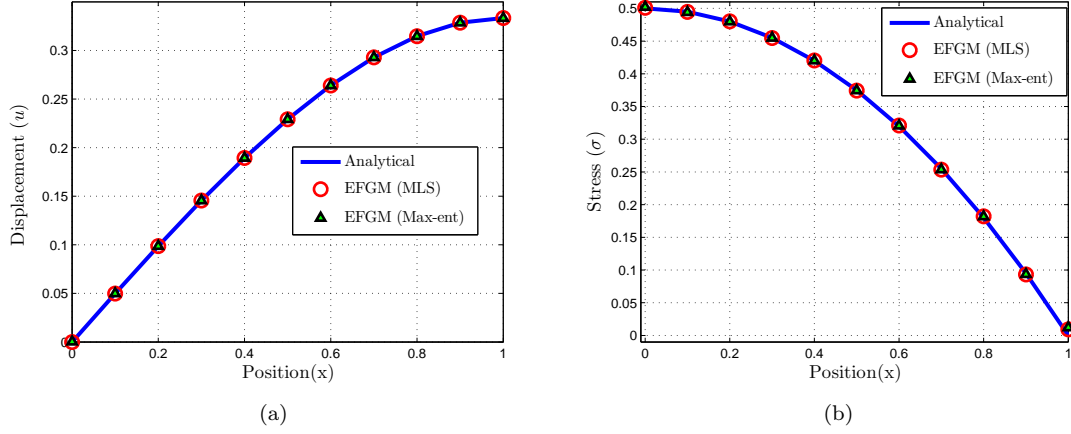


Figure 2.16: Displacement and stress for 1D bar problem

parameters, but the number of equally distributed nodes used are 11, 13, 15, 17, 19, 21 and the number of background cells used are 10, 12, 14, 16, 18, 20. For comparison with finite elements the same problem is also solved with the same number of nodes and two noded linear elements each having two Gauss points in each element. Figures 2.17(a) and 2.17(b) show the convergence of $\|e_u\|$ and $\|e\|$ for both the EFGM and the FEM. In these figures 2.17(a) and 2.17(b) h , R_u and R_e are distance between nodes and the rates of convergence as measured from these plots in the case of $\|e_u\|$ and $\|e\|$ respectively. From Figures 2.17(a) and 2.17(b) it is clear that the EFGM is more accurate with high rate of convergence as compared to the equivalent FEM and the accuracy and rate of convergence for the EFGM with the MLS shape functions are the same as that with max-ent shape functions.

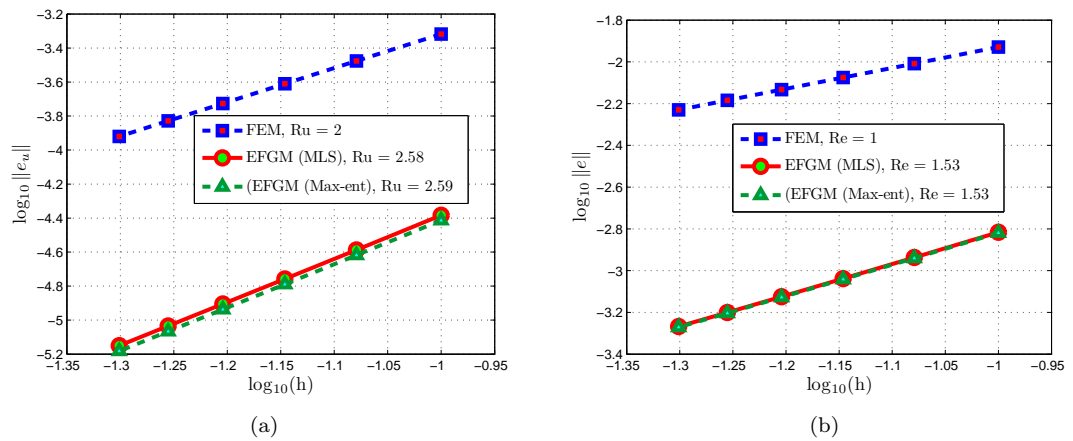


Figure 2.17: Rate of convergence for $\|e_u\|$ and $\|e\|$ for 1D bar problem

Two-dimensional cantilever beam

The behaviour of the 2D cantilever beam subjected to parabolic traction at the free end [293] is examined. The geometry, coordinate system, loading and boundary conditions for the problem, which are more

complicated than is often appreciated [20], are given in Figure 2.18. The analytical solution for the displacement field is given as [330]

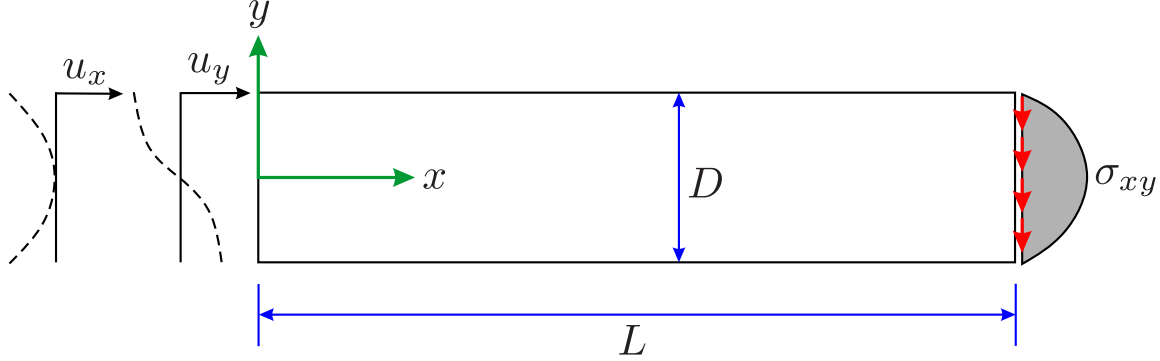


Figure 2.18: Geometry, boundary condition and loading for 2D beam problem

$$u(x, y) = \frac{Py}{6EI} \left[(6L - 3x)x + (2 + \nu)y^2 - \frac{3D^2}{2}(1 + \nu) \right], \quad (2.79a)$$

$$v(x, y) = -\frac{P}{6EI} [3\nu y^2 (L - x) + (3L - x)x^2], \quad (2.79b)$$

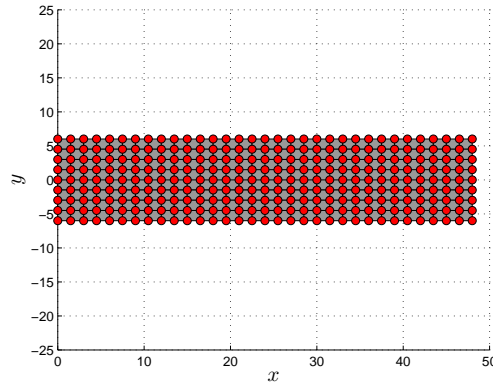
while the analytical solution for the stress fields is

$$\sigma_{xx} = \frac{P(L - x)y}{I}, \quad (2.80a)$$

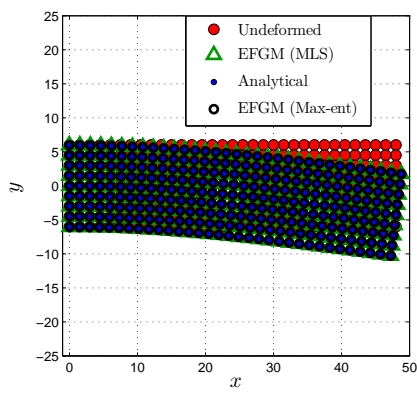
$$\sigma_{yy} = 0, \quad (2.80b)$$

$$\sigma_{xy} = -\frac{P}{2I} \left[\frac{D^2}{4} - y^2 \right], \quad (2.80c)$$

where E is modulus of elasticity, ν is Poisson's ratio and I is the second moment of area. The problem is solved with plane stress condition with $P = 1000$, $\nu = 0.3$, $E = 30 \times 10^6$, $D = 12$, $L = 48$ and unit thickness, all in compatible units. The problem is discretized with 175 (25×7) nodes, 144 (24×6) background cells with $d_{max} = 3.5$ and 4×4 Gauss quadrature per cell are used, while 4 Gauss quadrature points per line cell are used for integrations on the essential and traction boundaries. As mentioned in the previous example, there is no need to use Gauss points on the essential boundary in the EFGM with max-ent shape functions. The problem discretization with background cells and nodes is shown in Figure 2.19(a), while a comparison between the analytical and the EFGM with both the MLS and max-ent shape functions deflections at nodes and deflections at $y = 0$ are shown in Figures 2.19(b) and 2.19(c) respectively, which are in excellent agreement. Figures 2.20(a) and 2.20(c) show the contours of the normal stress σ_{xx} and shear stress σ_{xy} over the deflected beam, while Figures 2.20(b) and 2.20(d) compare the analytical and the EFGM normal and shear stresses at $x = L/2$ with both the MLS and max-ent, which are again in excellent agreement. For the convergence studies the same problem is solved with the same parameters, using $d_{max} = 2.5$ with 10 (5×2), 27 (9×3), and 52 (13×4) nodes and with 4 (4×1), 16 (8×2) and 36 (12×3) background cells. For comparison, the same problem is also solved with the FEM with four node iso-parametric quadrilateral elements, and with 2×2 gauss points in each element. Figures 2.21(a) and 2.21(b) show the convergence of $\|e_u\|$ and $\|e\|$ for the FEM and the EFGM with both the MLS and max-ent shape functions. From the Figures 2.21(a) and 2.21(b) it is clear that the EFGM while using either MLS or max-ent shape functions is more accurate, and the rate of convergence is higher than the equivalent FEM.



(a) Background cells and nodes



(b) Deflected nodes

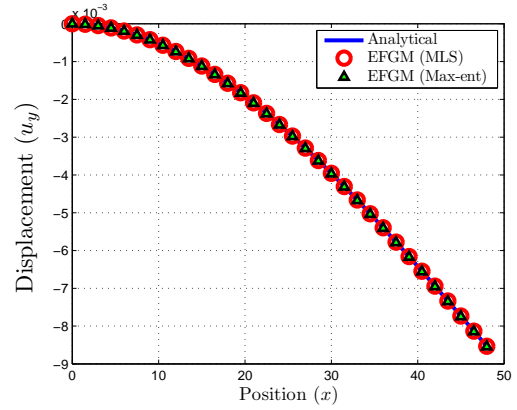
(c) Deflection at $y = 0$

Figure 2.19: Discretization and deflections for 2D beam problem

Hole in an infinite plate

The third example is an infinite plate with a circular hole of radius a in the centre, subjected to unidirectional in-plane pressure as shown in Figure 2.22. Due to symmetry only a portion of the upper-right quadrant is modelled with the pressures from the analytical expression, the geometry, coordinate system and boundary conditions are also shown in Figure 2.22. The analytical solution for the displacement field is given as [293]

$$u_x = \frac{1+\nu}{E} p \left(\frac{1}{1+\nu} r \cos \theta + \frac{2a^2}{(1+\nu)r} \cos \theta + \frac{a^2}{2r} \cos 3\theta - \frac{a^4}{2r^3} \cos 3\theta \right), \quad (2.81a)$$

$$u_y = \frac{1+\nu}{E} p \left(\frac{-\nu}{1+\nu} r \sin \theta - \frac{(1-\nu)a^2}{(1+\nu)r} \sin \theta + \frac{a^2}{2r} \sin 3\theta - \frac{a^4}{2r^3} \sin 3\theta \right), \quad (2.81b)$$

while the analytical solutions for the stress fields are given as

$$\sigma_{xx} = p \left[1 - \frac{a^2}{r^2} \left(\frac{3}{2} \cos 2\theta + \cos 4\theta \right) + \frac{3a^4}{2r^4} \cos 4\theta \right], \quad (2.82a)$$

$$\sigma_{xy} = p \left[-\frac{a^2}{r^2} \left(\frac{1}{2} \sin 2\theta + \sin 4\theta \right) + \frac{3a^4}{2r^4} \sin 4\theta \right], \quad (2.82b)$$

$$\sigma_{yy} = p \left[-\frac{a^2}{r^2} \left(\frac{1}{2} \cos 2\theta - \cos 4\theta \right) - \frac{3a^4}{2r^4} \cos 4\theta \right], \quad (2.82c)$$

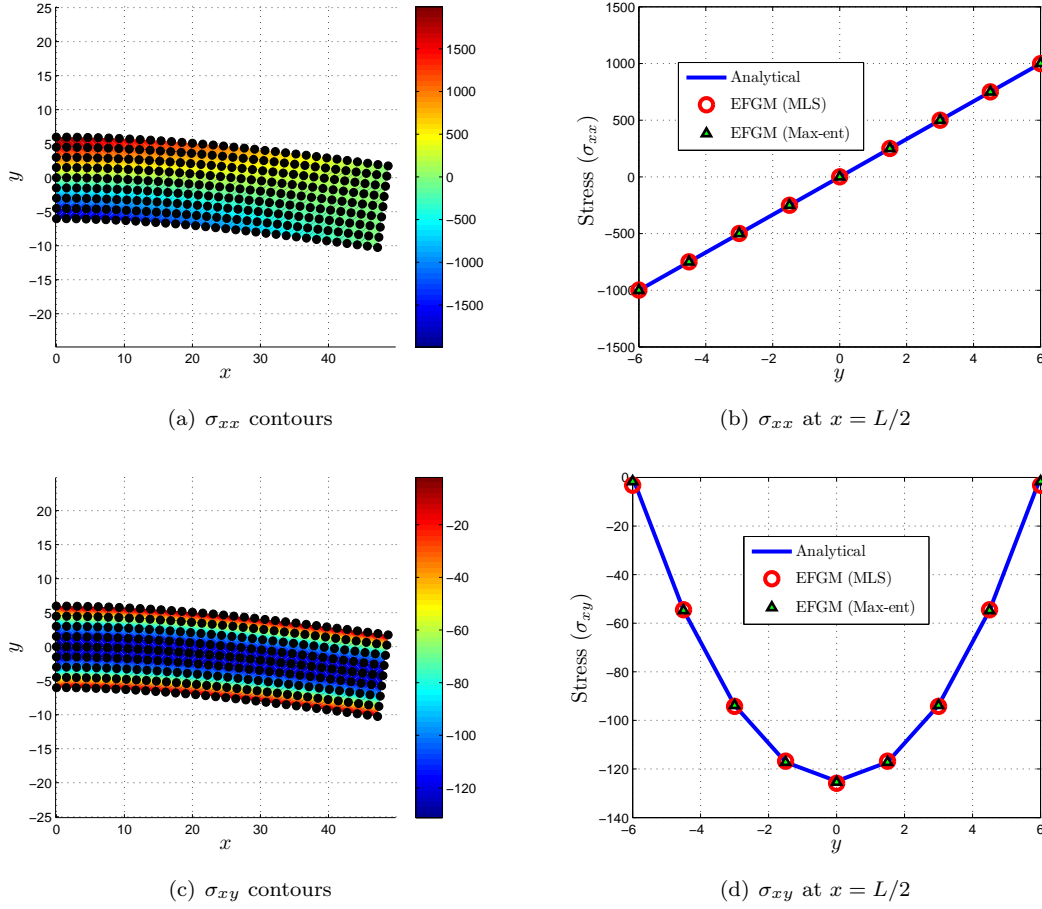
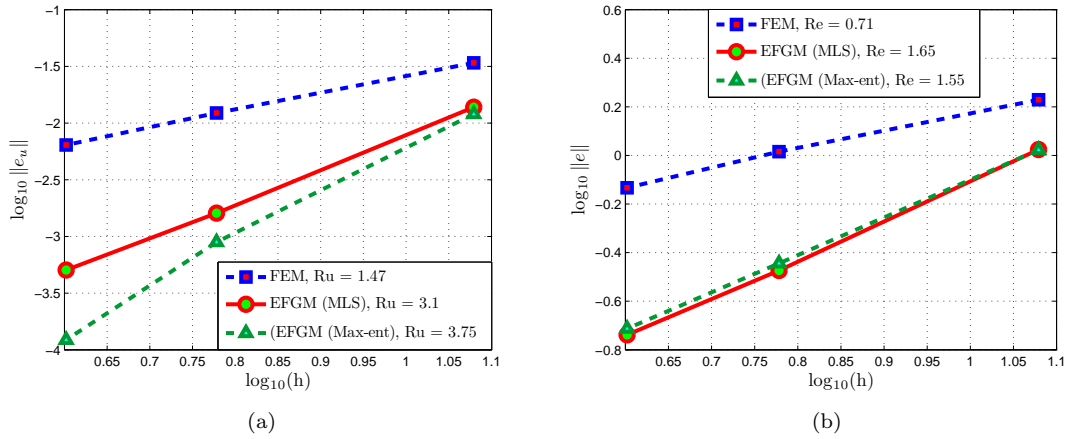


Figure 2.20: Normal and shear stresses for 2D beam problem

Figure 2.21: Rate of convergence for $\|e_u\|$ and $\|e\|$ for 2D beam problem

where E is the modulus of elasticity, ν is Poisson's ratio and r and θ are the polar coordinates. The problem is solved with a plane stress condition and with $a = 1$, $b = 4$, $E = 1000$, $\nu = 0.25$, $p = 1$. The discretization parameters used are 121 (11×11) nodes, i.e. 11 in the r and 11 in the θ direction, 100 (10×10) background cells, (4×4) Gauss quadrature per cell and 4 Gauss quadrature points per line cell on the essential and traction boundaries. Nodes and background cells are shown in Figure 2.23(a);

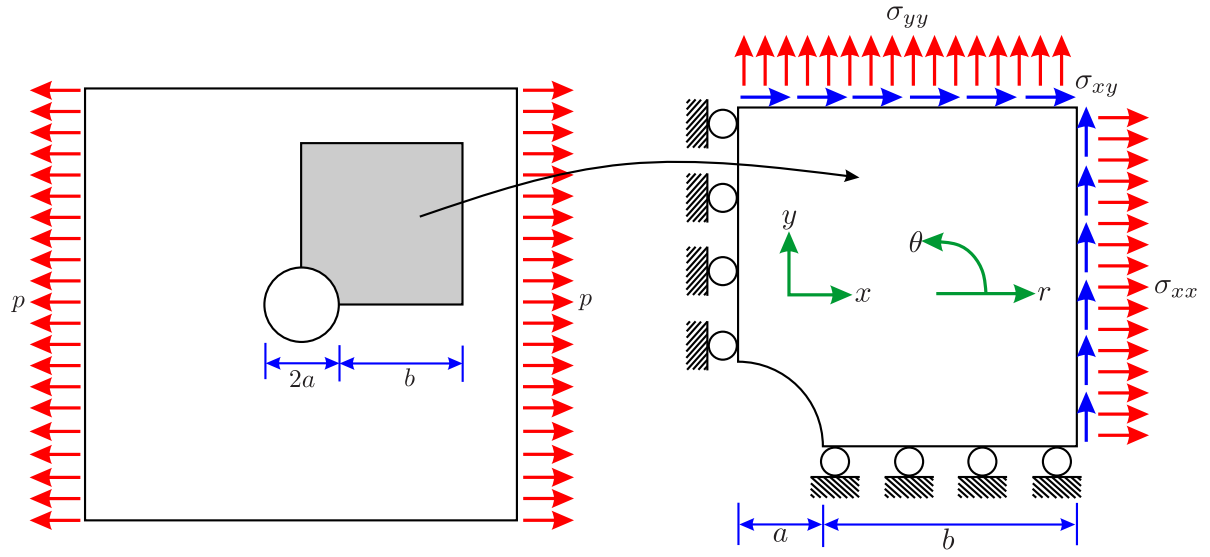


Figure 2.22: Geometry, boundary condition and loading for a 2D hole in an infinite plate problem

domain, essential and traction boundaries and Gauss points are shown in Figure 2.23(b). The comparison of the deflection at nodes predicted by the EFGM with both MLS and max-ent shape functions with analytical solution is shown in Figure 2.23(c). The contours of σ_{xx} and σ_{yy} over the deformed plate are shown in Figures 2.24(a), 2.24(c) respectively, while the comparison of σ_{xx} vs y at $x = 0$ and σ_{yy} vs x at $y = 0$ with the analytical solution are given in Figures 2.24(b) and 2.24(d) respectively. Figures 2.23(c), 2.24(b) and 2.24(d) shows an excellent agreement for the displacements and stresses between the EFGM with both MLS and max-ent shape functions and analytical solutions.

For the convergence studies the same problem is solved with $d_{max} = 3.5$ with 81 (9×9), 121 (11×11) and 169 (13×13) nodes, and with 64 (8×8), 100 (10×10) and 144 (12×12) background cells. For comparison, the same problem is also solved with the FEM with four node iso-parametric quadrilateral element with 2×2 gauss points in each element. Figures 2.25(a) and 2.25(b) shows the convergence of $\|e_u\|$ and $\|e\|$ for both the EFGM and the FEM, and it is clear from these figures that the accuracy and rate of convergence of the EFGM while using either the MLS or max-ent shape functions is higher than the equivalent FEM.

Three-dimensional cantilever beam with end load

The final example to demonstrate the EFGM implemented in this research is a 3D problem. The two-dimensional EFGM formulation derived in the §2.5 and §2.7 is straightforward to extend to three-dimensional problems. For three-dimensional problems Equations (2.44) and (2.45) are written as

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial \phi_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial \phi_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial \phi_i}{\partial z} \\ \frac{\partial \phi_i}{\partial y} & \frac{\partial \phi_i}{\partial x} & 0 \\ \frac{\partial \phi_i}{\partial z} & 0 & \frac{\partial \phi_i}{\partial x} \\ 0 & \frac{\partial \phi_i}{\partial z} & \frac{\partial \phi_i}{\partial y} \end{bmatrix}, \quad (2.83)$$

$$\mathbf{N}_k = \begin{bmatrix} N_k & 0 & 0 \\ 0 & N_k & 0 \\ 0 & 0 & N_k \end{bmatrix}. \quad (2.84)$$

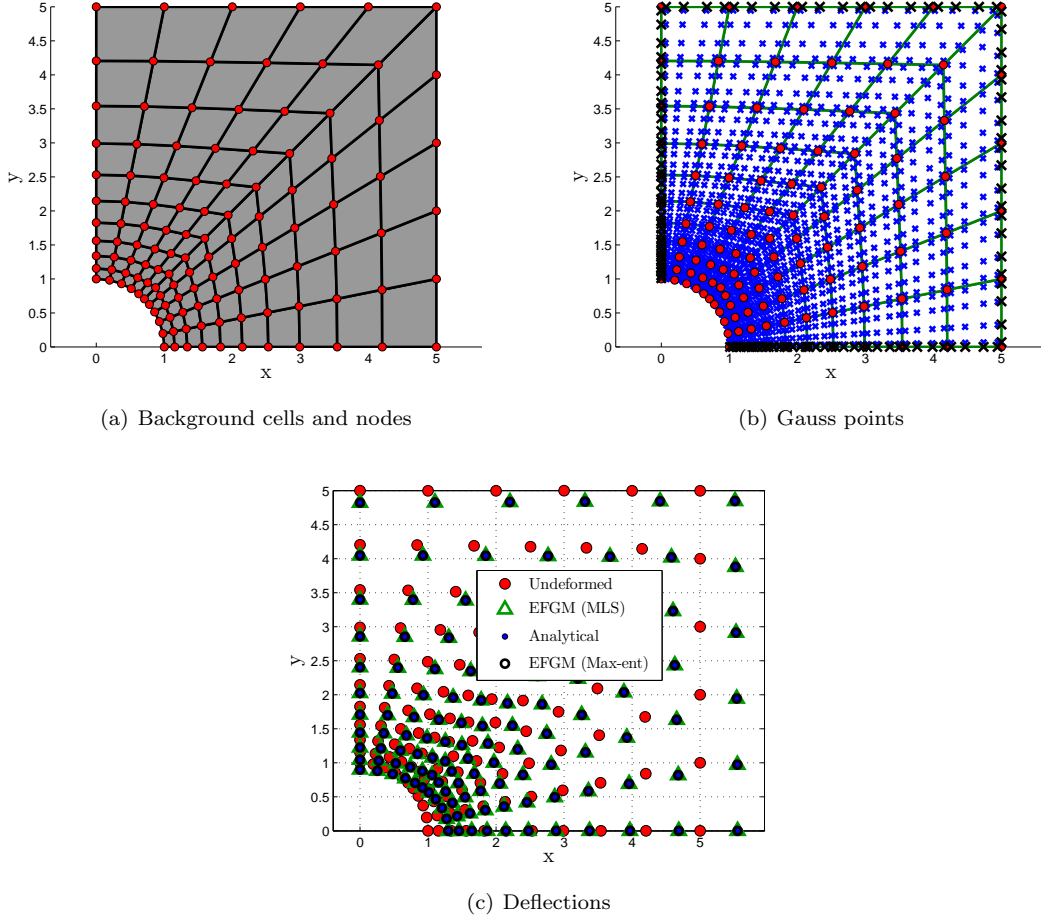


Figure 2.23: Problem discretization with background cells, nodes, Gauss points and deflected nodes for the hole in an infinite plate problem

In the case of max-ent shape functions, Equations (2.67), (2.62) and (2.63) are written as

$$\phi_i = \frac{\tilde{Z}_i}{\tilde{Z}}, \quad \tilde{Z}_i = w_i e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i - \lambda_3 \tilde{z}_i}, \quad \tilde{Z} = \sum_{j=1}^n \tilde{Z}_j, \quad (2.85)$$

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial \lambda_1} \\ \frac{\partial F}{\partial \lambda_2} \\ \frac{\partial F}{\partial \lambda_3} \end{bmatrix} = \begin{bmatrix} -\frac{\sum_{i=1}^n w_i e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i - \lambda_3 \tilde{z}_i} \tilde{x}_i}{\tilde{Z}} \\ -\frac{\sum_{i=1}^n w_i e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i - \lambda_3 \tilde{z}_i} \tilde{y}_i}{\tilde{Z}} \\ -\frac{\sum_{i=1}^n w_i e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i - \lambda_3 \tilde{z}_i} \tilde{z}_i}{\tilde{Z}} \end{bmatrix}, \quad (2.86)$$

$$\mathbf{A} = \nabla^2 F = \begin{bmatrix} \langle \tilde{x}^2 \rangle - \langle \tilde{x} \rangle^2 & \langle \tilde{x}\tilde{y} \rangle - \langle \tilde{x} \rangle \langle \tilde{y} \rangle & \langle \tilde{x}\tilde{z} \rangle - \langle \tilde{x} \rangle \langle \tilde{z} \rangle \\ \langle \tilde{x}\tilde{y} \rangle - \langle \tilde{x} \rangle \langle \tilde{y} \rangle & \langle \tilde{y}^2 \rangle - \langle \tilde{y} \rangle^2 & \langle \tilde{y}\tilde{z} \rangle - \langle \tilde{y} \rangle \langle \tilde{z} \rangle \\ \langle \tilde{x}\tilde{z} \rangle - \langle \tilde{x} \rangle \langle \tilde{z} \rangle & \langle \tilde{y}\tilde{z} \rangle - \langle \tilde{y} \rangle \langle \tilde{z} \rangle & \langle \tilde{z}^2 \rangle - \langle \tilde{z} \rangle^2 \end{bmatrix}, \quad (2.87)$$

where $\tilde{z}_i = z_i - z$.

Consider a three-dimensional cantilever beam loaded by uniformly distributed shearing traction at its free end with a total equivalent load of P in the positive y direction, and which is fixed at one end as shown in Figure 2.26(a). The same problem is also given in [31] to validate the three-dimensional

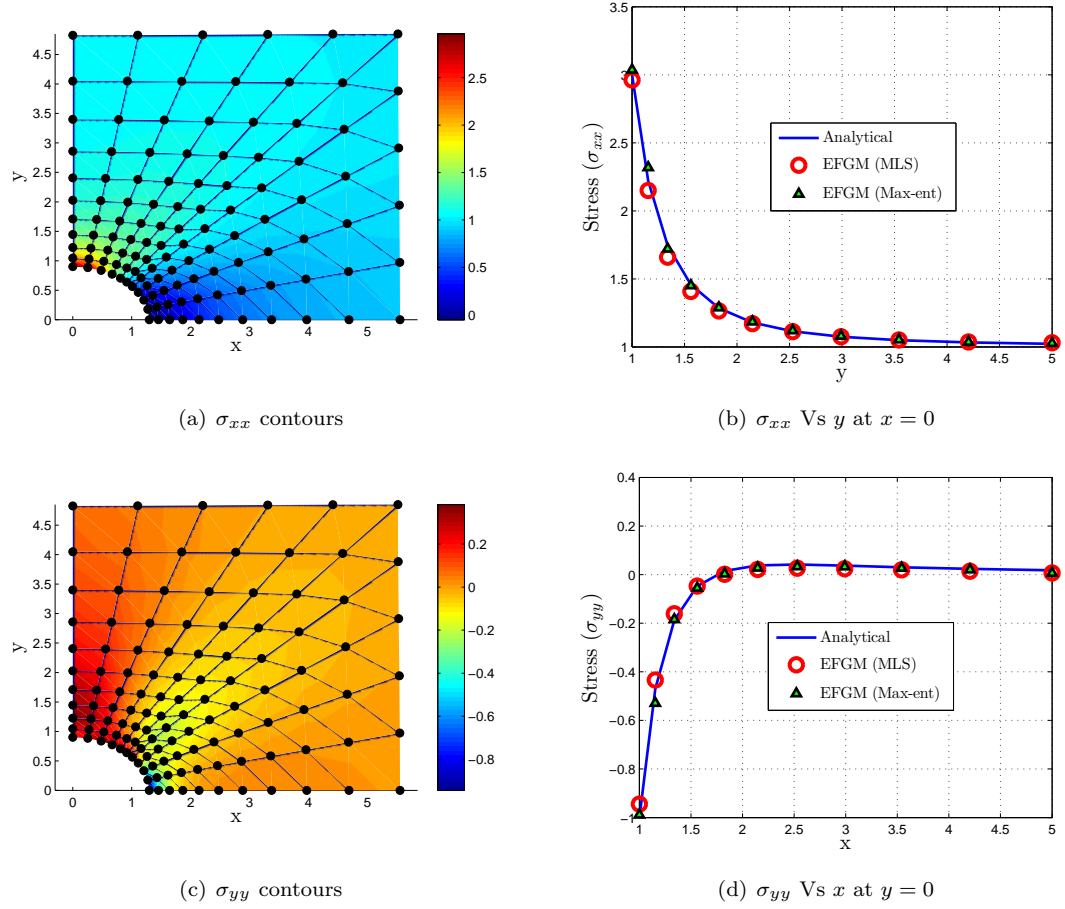
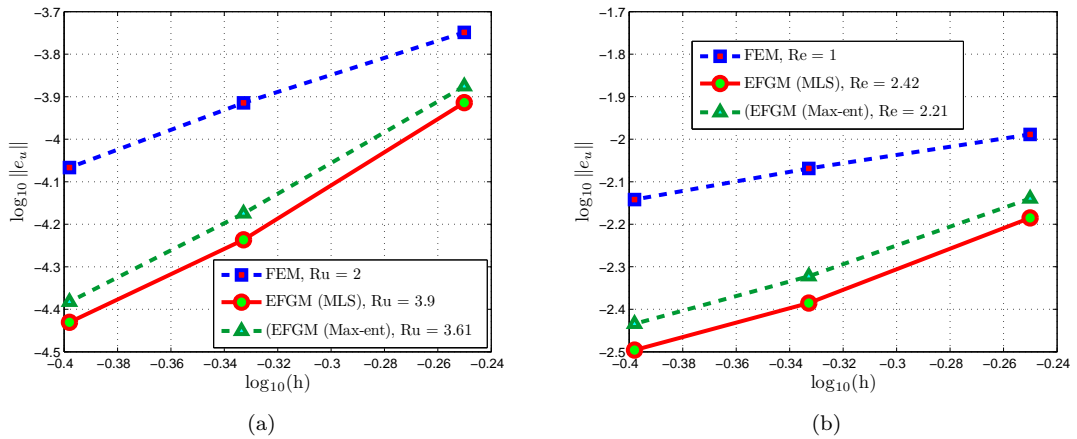


Figure 2.24: Stresses for the hole in an infinite plate problem

Figure 2.25: Rate of convergence for $\|e_u\|$ and $\|e\|$ for the hole in an infinite plate problem

application of the EFGM. The problem is solved with $E = 21 \times 10^7$, $\nu = 0.25$, $P = 10$, $L = 12$, $H = 1$ and $D = 1$, where L , H and D are the lengths in x , y , z directions. The problem is discretized using 625 ($25 \times 5 \times 5$) nodes, 384 ($24 \times 4 \times 4$) background cells, ($4 \times 4 \times 4$) Gauss quadrature per background cell and (2×2) Gauss quadrature on the essential and traction boundaries with $d_{max} = 2.5$. Analytical solutions for displacement and stresses are not available for this problem, so comparisons are made with

the analytical solution for the two-dimensional cantilever beam in the previous section. Background cells and nodes of the problem are shown in Figure 2.26(b), while the comparison between analytical and the EFGM solution for the displacement with both the MLS and max-ent are shown in Figure 2.26(c), which are in excellent agreement. Figures 2.27(a) and 2.27(b) shows the comparison of normal σ_{xx} and shear σ_{xy} stresses at $x = L/2$ and $z = 0$ recovered using EFGM with both the MLS and max-ent shape functions with the analytical solution. As expected the linear distribution of the normal σ_{xx} is recovered very accurately as compared to the parabolic distribution of the shear stress, which can be accurately recovered using a fine discretization.

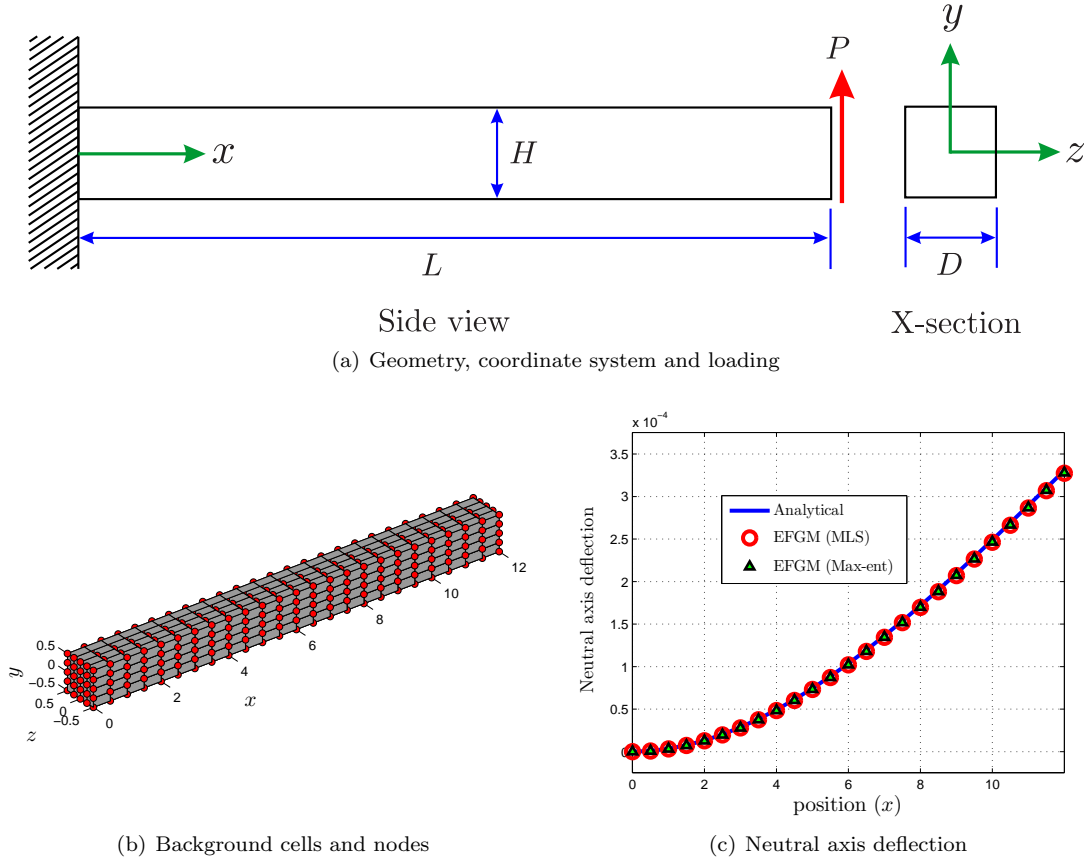


Figure 2.26: 3D beam problem geometry, discretization and neutral axis deflection

2.9 Run time comparison between max-ent and MLS shape functions

To compare the run time of max-ent and MLS shape functions and corresponding shape function derivatives, a cube is considered with 10 unit dimensions in the x , y and z directions. Seven different discretizations are used in this case with 125, 343, 729, 1331, 2197, 3375 and 4913 uniformly distributed nodes in the consecutive discretizations, while the number of background cells used in the consecutive discretizations are 64, 216, 512, 1000, 1728, 2744 and 4096. Three sample discretizations with 125, 729 and 2197 nodes are shown in Figures 2.28(a-c) respectively. 64 ($4 \times 4 \times 4$) Gauss points are used in each background cell and the scaling parameter used for the domain of influence is $d_{max} = 1.5$. The run time comparison between the max-ent and MLS shape functions for all the discretizations are shown in Table 2.3, while run time versus no of Gauss points are also shown in Figure 2.29. The timings reported for

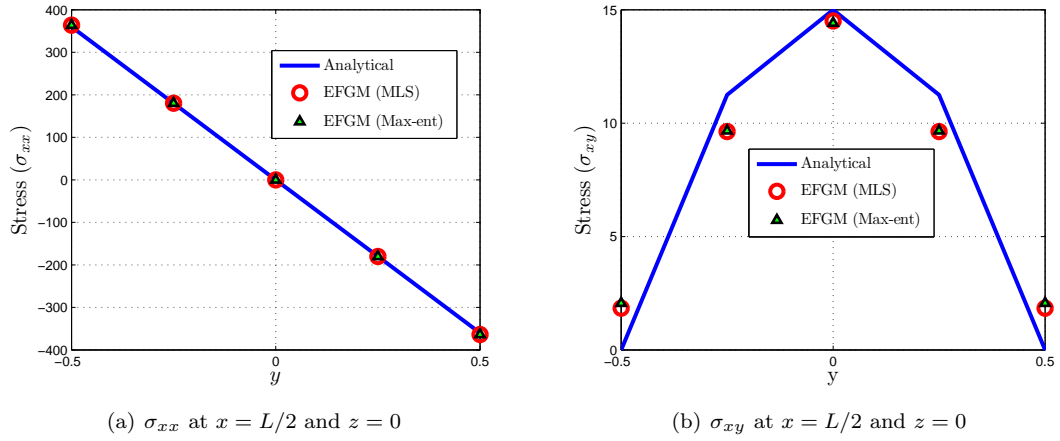


Figure 2.27: Normal and shear stresses for 3D cantilever beam problem

each discretization is the total time required to calculate the shape functions and corresponding shape function derivatives at all Gauss points in the problem domain. It is clear from Table 2.3 and Figure 2.29 that the calculation of max-ent shape functions and corresponding shape function derivatives are slightly more efficient as compared to the MLS shape functions and corresponding shape function derivatives.

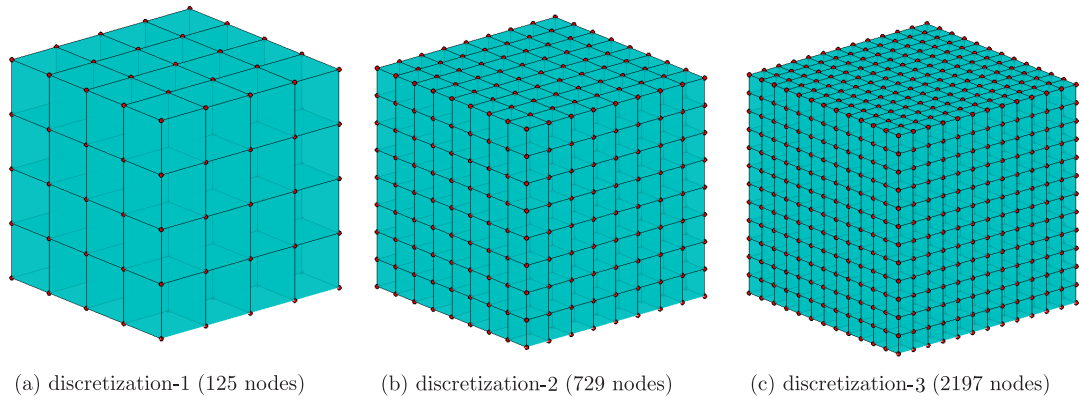


Figure 2.28: Sample discretizations used for the run time comparison of max-ent and MLS shape functions

Nodes	Background Cells	Number of Gauss points	max-ent (sec)	MLS (sec)
125	64	4096	0.83	0.83
343	216	13824	3.15	3.34
729	512	32768	8.21	8.71
1331	1000	64000	14.5	16.84
2197	1728	110592	26.15	28.02
3375	2744	175616	47.06	53.49
4913	4096	262144	72.25	78.18

Table 2.3: Run time comparison of max-ent and MLS shape functions

2.10 Concluding remarks

In this chapter numerical implementation of the EFGM is described in detail with the MLS and max-ent shape functions. As the MLS shape functions do not possess the Kronecker delta property, Lagrange multipliers are used to enforce the essential boundary conditions in an approximate sense. The recently

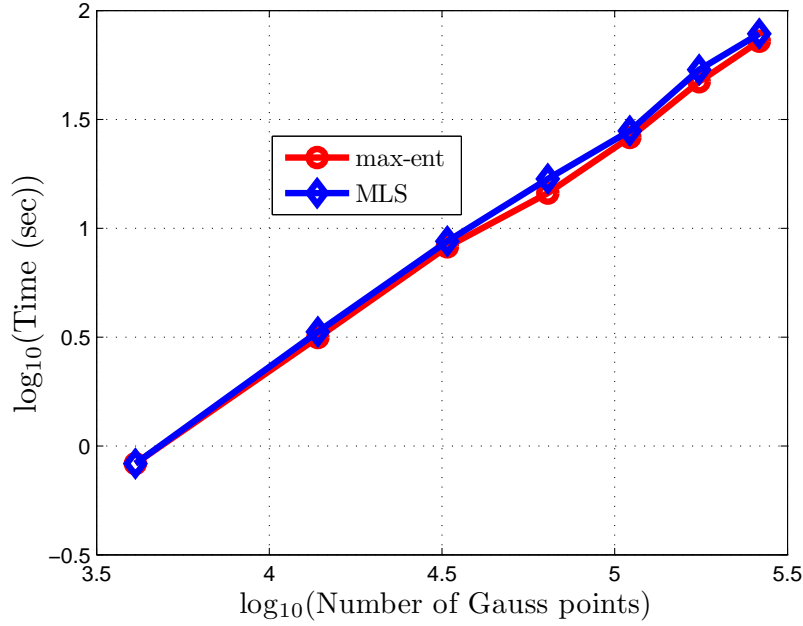


Figure 2.29: Run time comparison of max-ent and MLS shape functions

developed local max-ent shape functions are also implemented in the EFGM. These shape functions possess a weak Kronecker delta property at the boundaries, which allows the imposition of essential boundary conditions directly as in the FEM. It is also shown that while calculating the max-ent shape functions the number of iterations required for the convergence of the Newton method increases as the point approaches the problem boundary.

All methods implemented and demonstrated in this Chapter have been developed from scratch in MATLAB and used to run benchmark numerical examples from one-, two- and three-dimensional linear elasticity to show the performance and implementation of the EFGM. The numerical results of displacements and stresses, in all of the problems show very good agreement with the analytical solutions. It is also shown that the accuracy and rate of convergence of the EFGM for all the benchmark numerical examples, while using either the MLS or max-ent is faster (or the rate is higher) than the corresponding FEM. It is also mentioned that the calculation of max-ent shape functions and corresponding shape function derivatives are slightly efficient as compared with MLS counterpart.

Chapter 3

Modelling nonlinearities with element-free Galerkin method

3.1 Introduction

Problems with finite deformation and elasto-plasticity are very important in a variety of engineering applications, including geomechanics, biomechanics and metal forming. Generally, these problems can be further divided into subcategories, combining small strain, large strain, small displacement and large displacement hypothesis. A linear relationship exists between strains and displacements in small strain and small displacement problems, while a nonlinear relationship exists between strains and displacements in all other situations. In case of elasto-plastic deformations, bodies deform permanently under the application of external loads, as opposed to the fully reversible elastic deformations. The relationship between stresses and strains are linear for elasticity problems and nonlinear in case of problems involving plasticity. Theory of plasticity is concerned with finding the relationship between the stresses and strains in the case of plastic deformation. According to [246] an elasto-plastic model should have the following three components:

1. Relationship between stresses and strains during elastic deformations.
2. A yield function or criterion, which identify the difference between elastic and plastic deformations.
3. Relationship between stresses and strains in the case of plastic deformations.

Different material models exist in the literature, which can accurately simulate the response of different materials, e.g. metals, concrete, rocks and soils.

The FEM is a conventional way to solve these problems but there are always issues of mesh distortion and remeshing is required at different solution steps, which is computationally very expensive. Meshless methods are an ideal choice for these problems, as the problem can be discretized only based on a set of nodes, so there is no issues of mesh distortion and the expensive remeshing can be avoided. A number of relevant references can be found throughout the literature on the use of EFGM for modelling problems with finite deformations and elasto-plasticity. In [31] three-dimensional small strain, elastic and elasto-plastic formulations were developed for the EFGM with singular weight functions to assist the direct imposition of essential boundary conditions. Linear hardening and consistent tangent were used for the elasto-plastic formulations. EFGM was used for the modelling of constant volume (isochoric) plasticity in [14] and it was shown that the EFGM can overcome the very stiff response (volumetric

locking) of linear finite elements. It was also shown that the selection of the proper influence domain is an important parameter for the EFGM to get the limit load, e.g. for cantilever beam and Prandtl's punch test problems. Furthermore, it was also mentioned that the linear FEM shape functions can be retrieved from the MLS shape functions by using a very small domain of influence. A perfect-plasticity von Mises yield surface with the associated flow was used for the plane-strain analysis. In [187] the EFGM was used for large deformations with contact problems encounter in impact and metal forming, e.g. upsetting and extrusion processes. A cubic spline weight function with linear basis were used in the calculation of the MLS shape functions, while a linear isotropic hardening von Mises yield surface represented the material behavior. A computer aided design/computer aided engineering (CAD/CAE) system was developed for the EFGM for two dimensional small strains elasto-plastic problems in [124], which can use an input geometry from CAD or digital system (scanners, CT, etc.). Von Mises yield functions was used with associative flow and an explicit stress integration. The same conclusion was given to the relationship between the influence domain and volumetric locking, as in [14]. The issue of volumetric locking was further discussed in [56, 86, 328]. The EFGM with elasto-plastic formulation based on the von Mises yield criterion and power law hardening were used for the stress analysis of plate with a crack problem in [149]. The standard visibility criterion [39, 41, 243, 290] was used in modelling crack while the J-integral was used as a criterion for the benchmark comparison. Modified EFGM [7] was extended in [264] to finite strain problems, including axisymmetric and plane strain. Special shape functions were used in the modified EFGM on the essential boundaries derived from the extended partition of unity finite element method (PUFEM), which allows the direct imposition of essential boundary conditions. Total Lagrangian formulation was used with associative flow, isotropic hardening and von Mises yield surface. The EFGM was used in a number of references for metal forming simulation, including plane strain rolling in [319], bulk metal forming such as an equal channel angular pressing and cubic billet upsetting in [206], 3D-upsetting process in [203], forming of nail-shaped parts in [321] and 3D two-stage forging process in [207]. The EFGM was used to simulate a cantilever beam subjected to uniformly distributed load with small strains and large displacements response in [313] and large deformations response in the pile in [323].

The alternative meshless methods were also used for modeling problems with material and geometrical nonlinearities, i.e. RKPM with hyper-elastic and elasto-plastic formulations was used to simulate large deformations of rubber parts, cylinder under internal pressure, impact of aluminum bar and necking of circular cylinder [53]. In [52, 54] RKPM with large strains elasto-plastic deformations was used to model metal forming problems, including sheet metal, ring compression, upsetting, Taylor bar impact, forging and backward extrusion. 2D RKPM with a slightly compressible rigid-plastic formulation was used for the simulation of plane strain rolling problems in [268], while the same model was used in [318] to simulate compression of cylinders performs and rectangular bars and plane strain backward extrusion, for steady and non-steady bulk metal forming simulation, including flat rolling, compression of rods and heading of cylindrical billets in [317]. The rigid-plastic simulation of the metal forming was simulated using RKPM in [248] and was used for ring compression and forging simulations. In [125] the hybrid MLPG method originally proposed in [82] was extended to problems with small-strain elasto-plasticity and was used to simulate footing problems. Hybrid MLPG [82] is an ideal method for geomechanics problems, where the MLPG is used to model the near field, while the meshless scale boundary method is used to model the far field. In [322] a meshless method based on either MLS or RKPM with stabilized conforming nodal integration was used to simulate the elasto-plastic contact with the metal forming simulation and was used for sheet metal, necking, spring-back and 3D extrusion problems.

A meshless method based on the local weak form and MLS shape functions was developed for elasto-plastic formulation in [120] and was used for uniaxial bar problem. In [314] the MLPG was used for

the numerical simulation of the elasto-plastic plates with moderate thickness, while the radial basis functions (RBF) [198] were used, which facilitates the direct imposition of essential boundary conditions. Corrected smooth particle hydrodynamics (CSPH) methods with corrected kernel and integration were used for metal forming simulation and was used for upsetting and forging simulation in [44]. In [123] rigid-plastic point collocation method was used in 2D metal forming simulation of the plane stain upsetting and the results were validated by comparing with FE results. The NEM was used to model the large deformations in the biomechanics problems in [84, 87]. In metal forming simulation, NEM was used in a number of references, e.g. in large deformation extrusion process and simulation of moving interfaces in casting process in [6], in the extrusion process in [96] and in the backward extrusion process in [141]. The linear elastic EFGM formulation developed in Chapter 2 is extended here to include both material and geometrical nonlinearities, which is although not a novel contribution but is an essential constituent of Chapter 4 to 7. Varieties of benchmark numerical examples are solved to demonstrate its correct implementation.

The outline of this chapter is as follows. The classical theory of plasticity is explained in §3.2, which consists of elasto-plastic tangent matrix derivation in §3.2.1 followed by a full procedure of stress integration in §3.2.2, derivation of the consistent or algorithmic tangent in §3.2.3 and the numerical implementation algorithms in §3.2.4. The formulations of the theory of plasticity in case of finite deformations is explained in §3.3, which is further divided into an update Lagrangian in §3.3.1 and total Lagrangian in §3.3.2. A number of numerical examples are given in §3.4 to demonstrate the implementation and performance of the use of the EFGM in case of problems with elasto-plasticity and finite deformation. Concluding remarks are given in §3.5.

3.2 Classical theory of plasticity

The following are the main components of the elasto-plasticity and can be found in any classical book on computational plasticity, e.g. [81, 271] or thesis on computational plasticity, e.g. [64, 67, 190]. In this research, we are only using perfect plasticity, i.e. without hardening/softening and with the associative flow

Hypoelastic law The rate form of the Cauchy stresses and elastic strains is related as

$$\dot{\boldsymbol{\sigma}} = \mathbf{D}\dot{\boldsymbol{\varepsilon}}^e, \quad (3.1)$$

where the Cauchy stress vector $\boldsymbol{\sigma}$ and elastic strain vector $\boldsymbol{\varepsilon}^e$ are written as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{zz} & \sigma_{xy} & \sigma_{yz} & \sigma_{xz} \end{bmatrix}^T, \quad \boldsymbol{\varepsilon}^e = \begin{bmatrix} \varepsilon_{xx}^e & \varepsilon_{yy}^e & \varepsilon_{zz}^e & 2\varepsilon_{xy}^e & 2\varepsilon_{yz}^e & 2\varepsilon_{xz}^e \end{bmatrix}^T \quad (3.2)$$

\mathbf{D} is a matrix of elastic moduli and is already defined in Chapter 2.

Additive decomposition of strains The rate of straining, $\dot{\boldsymbol{\varepsilon}}$, can be decomposed into elastic parts $\dot{\boldsymbol{\varepsilon}}^e$ and plastic parts $\dot{\boldsymbol{\varepsilon}}^p$ and is written as

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^e + \dot{\boldsymbol{\varepsilon}}^p. \quad (3.3)$$

Yield function It defines a surface in stress space \mathbb{S} , which separates the admissible and inadmissible regions. The admissible region \mathbb{E}_σ is written as

$$\mathbb{E}_\sigma = \{\boldsymbol{\sigma} \in \mathbb{S} \mid f(\boldsymbol{\sigma}) \leq 0\}, \quad (3.4)$$

where $f(\boldsymbol{\sigma})$ is yield function. The interior or elastic region of \mathbb{E}_σ is represented by $\text{int}(\mathbb{E}_\sigma)$ for

which $f(\boldsymbol{\sigma}) < 0$ and the boundary of \mathbb{E}_σ or elasto-plastic region is represented by $\partial\mathbb{E}_\sigma$ for which $f(\boldsymbol{\sigma}) = 0$.

Flow rule In this case it is assumed that the plastic strain is proportional to the stress gradient of the plastic potential function g and is written as

$$\dot{\boldsymbol{\epsilon}}^p = \dot{\gamma} \frac{\partial g(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} = \dot{\gamma} g_{,\boldsymbol{\sigma}}, \quad (3.5)$$

where $\dot{\gamma} \geq 0$ is a non-negative constant of proportionality and is known as plastic multiplier, while $\frac{\partial g(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}}$ define the direction of plastic flow. If f and g coincide the flow rule is associated, otherwise, the flow rule is said to be non-associated.

Kuhn-Tucker loading/unloading and consistency conditions Kuhn-Tucker loading/unloading and consistency conditions are

$$\dot{\gamma} \geq 0, \quad f(\boldsymbol{\sigma}) \leq 0, \quad \dot{\gamma} f(\boldsymbol{\sigma}) = 0 \quad (3.6)$$

and

$$\dot{\gamma} \dot{f}(\boldsymbol{\sigma}) = 0, \quad (\text{for } f(\boldsymbol{\sigma}) = 0). \quad (3.7)$$

1. If $\boldsymbol{\sigma} \in \text{int}(\mathbb{E}_\sigma)$, i.e. $f(\boldsymbol{\sigma}) < 0$ then from Equation (3.6) $\dot{\gamma} = 0$ and from Equation (3.5) $\dot{\boldsymbol{\epsilon}}^p = 0$ and from Equation (3.3) $\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}}^e$ and Equation (3.1) gives $\dot{\boldsymbol{\sigma}} = \mathbf{D}\dot{\boldsymbol{\epsilon}}^e$ and the response is known as instantaneously elastic.
2. If $\boldsymbol{\sigma} \in \partial\mathbb{E}_\sigma$, i.e. $f(\boldsymbol{\sigma}) = 0$, then Equation (3.6) is satisfied for both of $\dot{\gamma} = 0$ and $\dot{\gamma} > 0$. For this case the exact value of $\dot{\gamma}$ can be determined using Equation (3.7)
 - (a) If $\dot{f}(\boldsymbol{\sigma}) < 0$, from Equation (3.7) $\dot{\gamma} = 0$ and from Equation (3.5) $\dot{\boldsymbol{\epsilon}}^p = 0$ and the response is known as elastic unloading.
 - (b) If $\dot{f}(\boldsymbol{\sigma}) = 0$ then Equation (3.7) is satisfied for both of $\dot{\gamma} = 0$ and $\dot{\gamma} > 0$.
 - i. If $\dot{\gamma} > 0$ so from Equation (3.5) $\dot{\boldsymbol{\epsilon}}^p \neq 0$ and the response is known as plastic loading.
 - ii. If $\dot{\gamma} = 0$ the response is known as neutral loading.

3.2.1 Elasto-plastic tangent matrix

To derive the expression for infinitesimal elasto-plastic tangent matrix, which provide a relationship between stress and strain rate. After combining Equations (3.5) and (3.3) and using associative flow, i.e. $f = g$

$$\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}}^e + \dot{\gamma} f_{,\boldsymbol{\sigma}}. \quad (3.8)$$

Equation (3.1) can be written, after using $\dot{\boldsymbol{\epsilon}}^e$ from 3.8

$$\dot{\boldsymbol{\sigma}} = \mathbf{D}(\dot{\boldsymbol{\epsilon}} - \dot{\gamma} f_{,\boldsymbol{\sigma}}). \quad (3.9)$$

The equation for the consistency condition is written as

$$\dot{f} = f_{,\boldsymbol{\sigma}}^T \dot{\boldsymbol{\sigma}} = f_{,\boldsymbol{\sigma}}^T \mathbf{D}(\dot{\boldsymbol{\epsilon}} - \dot{\gamma} f_{,\boldsymbol{\sigma}}) = 0 \quad (3.10)$$

Equation (3.10) can be rearranged to get $\dot{\gamma}$ as

$$\dot{\gamma} = \frac{f_{,\boldsymbol{\sigma}}^T \mathbf{D} \dot{\boldsymbol{\epsilon}}}{f_{,\boldsymbol{\sigma}}^T \mathbf{D} f_{,\boldsymbol{\sigma}}}. \quad (3.11)$$

Putting $\dot{\gamma}$ from Equation (3.11) back in Equation (3.9)

$$\dot{\sigma} = \mathbf{D} \left(\dot{\epsilon} - \left(\frac{f_{,\sigma}^T \mathbf{D} \dot{\epsilon}}{f_{,\sigma}^T \mathbf{D} f_{,\sigma}} \right) f_{,\sigma} \right) = \mathbf{D} \left(\mathbf{I}_6 - \frac{f_{,\sigma} f_{,\sigma}^T \mathbf{D}}{f_{,\sigma}^T \mathbf{D} f_{,\sigma}} \right) \dot{\epsilon} = \mathbf{D}^{ep} \dot{\epsilon}, \quad (3.12)$$

where \mathbf{I}_6 is (6×6) identity matrix and \mathbf{D}^{ep} is only a function of the yield function f , the elastic stiffness matrix \mathbf{D} and the current stress state σ and is known as an infinitesimal, instantaneous elasto-plastic tangent matrix and is written as

$$\mathbf{D}^{ep} = \mathbf{D} \left(\mathbf{I}_6 - \frac{f_{,\sigma} f_{,\sigma}^T \mathbf{D}}{f_{,\sigma}^T \mathbf{D} f_{,\sigma}} \right). \quad (3.13)$$

3.2.2 Principle of stress integration

In this section detail is given for the determination of the stress state for a given strain increment. Due to the nonlinearities of the constitutive equation in elasto-plasticity, it is generally not possible to solve these equations analytically and fully Implicit backward Euler stress integration is used here for this purpose. To explain the procedure, consider a point A at the end of solution step n in the stress space inside a yield surface with stress state σ_n corresponds to an elastic strain state ϵ_n as shown in Figure 3.1. If $\Delta\epsilon_{n+1}$ is strain increment within solution step $n + 1$, then the goal of the stress integration is to determine the stress state σ_{n+1} . This process is divided into two parts, i.e. an elastic predictor AB followed by a plastic corrector BC . During an elastic prediction assuming linear-elasticity the stress state is found using

$$\sigma^t = \mathbf{D}(\epsilon_n^e + \Delta\epsilon_{n+1}) = \mathbf{D}\epsilon_t^e, \quad (3.14)$$

where σ^t and ϵ_t^e are known as trial stress state and trial elastic strain state respectively. If σ^t lies inside the yield surface, then there is no need of correction but in case when σ^t lies outside the yield surface a plastic correction is required, which is given as

$$\sigma_{n+1} = \mathbf{D}(\epsilon_t^e - \Delta\epsilon^p). \quad (3.15)$$

The plastic correction in Equation (3.15) is performed iteratively to return the stress back onto the yield surface. The goal of the fully Implicit backward Euler stress integration is the convergence of the residual \mathbf{b}_r within specified tolerance

$$\mathbf{b}_r = \begin{bmatrix} \epsilon^e - \epsilon_t^e + \Delta\gamma f_{,\sigma} \\ f \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{r1} \\ b_{r2} \end{bmatrix}, \quad (3.16)$$

where $\Delta\gamma$ is incremental plastic multipliers. The backward Euler increment is written as

$$\delta\mathbf{x} = -\mathbf{A}^{-1}\mathbf{b}_r \quad (3.17)$$

here $\mathbf{x} = \begin{bmatrix} \epsilon^e & \Delta\gamma \end{bmatrix}^T$ is a vector of unknown and \mathbf{A} is Hessian matrix and is obtained from the derivative of the residual \mathbf{b}_r with respect to the unknowns and written as

$$\mathbf{A} = \begin{bmatrix} \mathbf{b}_{r1,\epsilon^e} & \mathbf{b}_{r1,\Delta\gamma} \\ b_{r2,\epsilon^e} & b_{r2,\Delta\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_6 + \Delta\gamma f_{,\sigma\sigma} \mathbf{D} & f_{,\sigma} \\ f_{,\sigma}^T \mathbf{D} & 0 \end{bmatrix}. \quad (3.18)$$

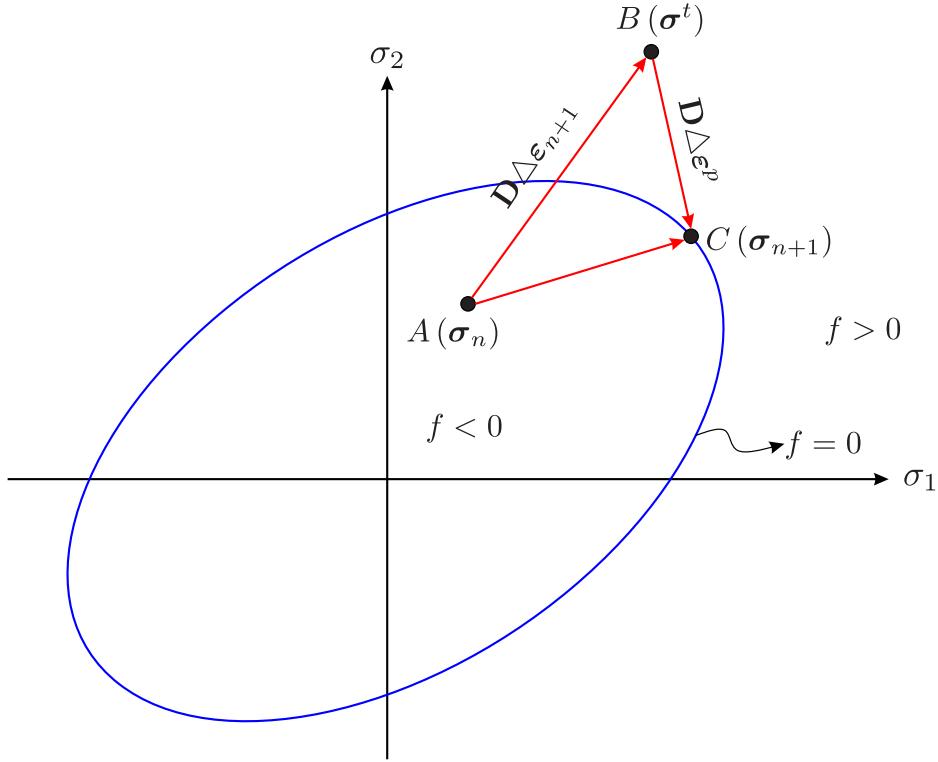


Figure 3.1: Stress return procedure

The first backward Euler iteration is started with

$${}^0\boldsymbol{\varepsilon}^e = \boldsymbol{\varepsilon}_t^e, \quad {}^0\Delta\gamma = 0, \quad {}^0\mathbf{b}_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & f \end{bmatrix}^T, \quad (3.19)$$

where the pre-superscript is the backward Euler iteration number. The step by step backward Euler procedure is explained graphically in [67], which is summarized as:

1. Calculate ${}^i\mathbf{A}$ from Equation (3.18).
2. Calculate the increment using ${}^i\delta\mathbf{x} = -{}^i\mathbf{A}^{-1}{}^i\mathbf{b}_r$.
3. Update the unknown using ${}^{i+1}\mathbf{x} = {}^i\mathbf{x} + {}^i\delta\mathbf{x}$.
4. Calculate new ${}^{i+1}f_{,\sigma}$ and ${}^{i+1}f_{,\sigma\sigma}$.
5. Calculate ${}^{i+1}\mathbf{b}_r$ from Equation (3.16).
6. If ${}^{i+1}\mathbf{b}_r$ is not within tolerance repeat the steps, otherwise calculate \mathbf{D}^{ep} from Equation (3.13).

3.2.3 Consistent or algorithmic tangent

If the tangent \mathbf{D}^{ep} is used in the Newton-Raphson iteration for the incremental-iterative solution of the equilibrium equations, the inherent quadratic rate of asymptotic convergence of the global Newton-Raphson procedure will be lost. A consistent or algorithmic tangent \mathbf{D}^{alg} , which is consistent with the integration of constitutive equations or stress updating algorithm allows the quadratic rate of asymptotic convergence of the global Newton-Raphson algorithm [274]. Mathematically, the consistent tangent is the change in stress w.r.t to the trial elastic strain and is written as

$$\mathbf{D}^{alg} = \boldsymbol{\sigma}_{,\boldsymbol{\varepsilon}_t^e} \quad (3.20)$$

The formulations derived for fully Implicit backward Euler stress integration can be used to derive the expression for the consistent tangent. Linearization of the first equation of the residual (3.16) [67]

$$\mathbf{C}^e d\boldsymbol{\sigma} + \Delta\gamma f_{,\sigma\sigma} d\boldsymbol{\sigma} + d\Delta\gamma f_{,\sigma} = d\boldsymbol{\varepsilon}_t^e, \quad (3.21)$$

where $\mathbf{C}^e = \mathbf{D}^{-1}$ and is known as the elastic compliance matrix. while linearization of the second equation of the residual (3.16) is written as

$$f_{,\sigma}^T d\boldsymbol{\sigma} = 0. \quad (3.22)$$

Combining Equations (3.21) and (3.22)

$$\underbrace{\begin{bmatrix} \mathbf{C}^e + \Delta\gamma f_{,\sigma\sigma} & f_{,\sigma} \\ f_{,\sigma}^T & 0 \end{bmatrix}}_{(\mathbf{A}^{alg})^{-1}} \begin{Bmatrix} d\boldsymbol{\sigma} \\ d\Delta\gamma \end{Bmatrix} = \begin{Bmatrix} d\boldsymbol{\varepsilon}_t^e \\ 0 \end{Bmatrix}. \quad (3.23)$$

Equation (3.23) can also be written as

$$\begin{Bmatrix} d\boldsymbol{\sigma} \\ d\Delta\gamma \end{Bmatrix} = \begin{bmatrix} \mathbf{D}^{alg} & \mathbf{A}_{12}^{alg} \\ \mathbf{A}_{21}^{alg} & \mathbf{A}_{22}^{alg} \end{bmatrix} \begin{Bmatrix} d\boldsymbol{\varepsilon}_t^e \\ 0 \end{Bmatrix} \quad (3.24)$$

where \mathbf{D}^{alg} is the required consistent or algorithmic tangent.

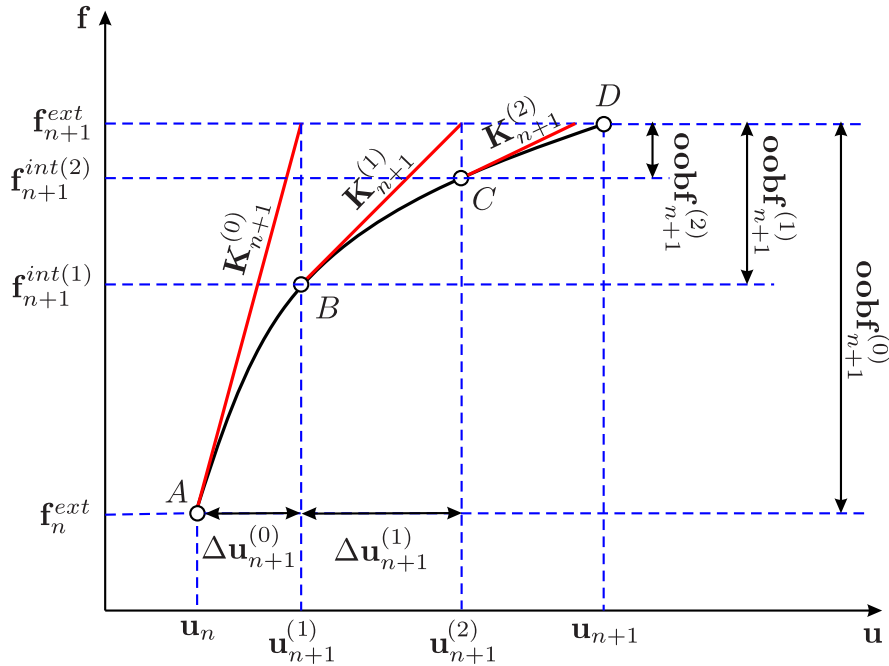


Figure 3.2: Convergence of the Newton-Raphson method for an iteration $n + 1$

3.2.4 Numerical implementation algorithm

To solve a non-linear structural mechanics problem, the internal stresses should be balanced with the external applied forces. In this research, an incremental-iterative procedure is used, i.e. the load or displacement (in load or displacement control problems respectively) is applied incrementally in steps and an iterative method, e.g. Newton-Raphson is used to sought convergence for each increment. Newton-

Raphson method is a natural choice in these types of situations, due to its robustness, efficiency and asymptotic quadratic rate of convergence. The equilibrium equation is discretized incrementally in each Newton-Raphson iteration, and the discretized version of the incremental equilibrium equation is written as

$$\mathbf{f}_{n+1}^{int}(\mathbf{u}_{n+1}) - \mathbf{f}_{n+1}^{ext} = \mathbf{oobf}_{n+1} = 0, \quad (3.25)$$

where $n + 1$ is the global Newton-Raphson iteration counter, \mathbf{f}_{n+1}^{int} and \mathbf{f}_{n+1}^{ext} are the global internal and external force vectors respectively, \mathbf{oobf}_{n+1} is the residual or out-of-balance force and \mathbf{u}_{n+1} is a vector of nodal parameters or fictitious nodal values. The expression for the internal loads for an increment $n + 1$ are given as

$$\mathbf{f}_{n+1}^{int} = \int_{\Omega} \mathbf{B}^T \boldsymbol{\sigma} d\Omega = \sum_{i=1}^{n_g} \mathbf{B}_i^T \boldsymbol{\sigma}_i |\mathbf{J}_i| w_i, \quad (3.26)$$

where n_g are the total number of Gauss points within the problem domain, \mathbf{B}_i is the strain displacement matrix at Gauss point i , consisting of the shape function derivatives calculated with reference to the original nodal coordinates and the original position of the Gauss points for small strain plasticity and $|\mathbf{J}_i|$ and w_i are the Jacobian and weights associated with each Gauss points respectively. The external load \mathbf{f}_{n+1}^{ext} for increment $n + 1$ is a fraction of the total external load \mathbf{f}^{ext} , which is found only once at the start of the simulation using Equation (2.42). The step by step convergence of the Newton-Raphson method for a load increment $\mathbf{f}_{n+1}^{ext} - \mathbf{f}_n^{ext}$ is shown in Figure 3.2, where \mathbf{u}_{n+1} is the final converged nodal parameters. Each individual local Newton-Raphson iteration is the solution of the following linear system of equations

$$\mathbf{K}_{n+1}^{iNR} \Delta \mathbf{u}_{n+1}^{iNR} = \mathbf{oobf}_{n+1}^{iNR}, \quad (3.27)$$

where iNR is the local Newton-Raphson counter and $\Delta \mathbf{u}_{n+1}^{iNR}$ is the incremental nodal displacements or fictitious nodal values and \mathbf{K}_{n+1}^{iNR} is the global stiffness matrix and is written as

$$\mathbf{K}_{n+1} = \int_{\Omega} \mathbf{B}^T \mathbf{D}^{alg} \mathbf{B} d\Omega \quad (3.28)$$

where \mathbf{D}^{alg} is a small strain consistent or algorithmic tangent. The nodal parameters at the end of iteration $n + 1$ is written as

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \sum_{i=1}^{n_{NR}} \Delta \mathbf{u}_{n+1}^{iNR}, \quad (3.29)$$

where n_{NR} is the total number of local Newton-Raphson iteration required for convergence. The initial global stiffness matrix $\mathbf{K}_{n+1}^{(0)}$ at point A is also changing in the each local Newton-Raphson iteration until convergence is achieved. The convergence criterion of the Newton-Raphson method is [67, 81]

$$\frac{|\mathbf{oobf}_{n+1}^{n_{NR}}|}{|\mathbf{f}_{n+1}^{ext}|} < \varepsilon_{tol}, \quad (3.30)$$

where ε_{tol} is a small number and in this thesis, a value 1×10^{-6} is used. $\mathbf{u}_{n+1} = \mathbf{u}_{n+1}^{(m)}$ is considered to be the solution for the global increment $n + 1$. In this study the Prandtl-Reuss constitutive model is used, which comprises the von Mises yield function with perfect plasticity and associated flow. The von Mises yield function is defined as $f = \sqrt{3J_2} - \sigma_y$, where J_2 is the second invariant of the deviatoric stress and σ_y is the yield strength of the material. Effective plastic strain is used as one of the measures to evaluate

the performance of the proposed model and is given as

$$\overline{\varepsilon^p} = \sqrt{\frac{2}{3} (\varepsilon^p)^T (\varepsilon^p)}, \quad (3.31)$$

3.3 Theory of plasticity with finite deformation

As opposed to the infinitesimal formulation derived in §3.2 where a linear relation is assumed to exist between the strains and displacements, a finite deformation (large deformation) elasto-plastic formulation is derived in this section, in which the relationship between strains and displacements is nonlinear. Two different types of formulations i.e. updated Lagrangian and total Lagrangian are given here. In the updated Lagrangian, all the kinematical variables are referred to the current configuration, while in the total Lagrangian, all these variables are referred back to the original configuration. Consider a point \mathbf{P} in the original or the reference configuration of a solid body Ω_0 at time $t = 0$ as shown in Figure 3.3 with coordinates \mathbf{X} . After deformation \mathbf{u} at time t the same point is represented by \mathbf{p} with coordinates \mathbf{x} in the current or deformed configuration Ω . Here the deformation gradient is used, which is the fundamental

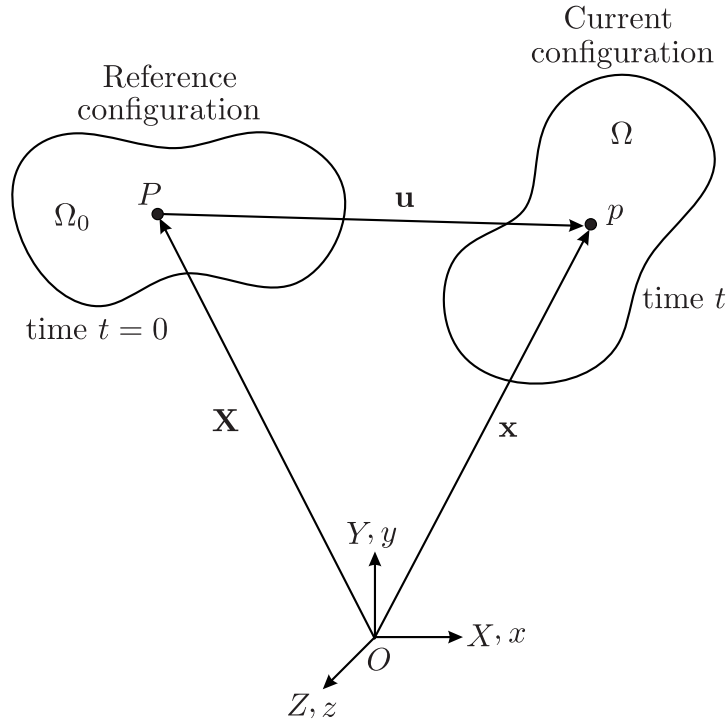


Figure 3.3: Relationship between the current and reference configuration

measure of deformation providing the relationship between the current and reference configurations, i.e.

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I}_3 + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}, \quad (3.32)$$

where \mathbf{I}_3 is a (3×3) unit matrix. The polar decomposition of the deformation gradient is written as [130]

$$\mathbf{F} = \mathbf{R}\mathbf{U} + \mathbf{v}\mathbf{R}, \quad (3.33)$$

where \mathbf{U} and \mathbf{v} are the right and left stretch matrices respectively, while \mathbf{R} is known as the rotation matrix. \mathbf{U} and \mathbf{v} can also be written as

$$\mathbf{U} = \sqrt{\mathbf{C}} = \sqrt{\mathbf{F}^T \mathbf{F}}, \quad (3.34a)$$

$$\mathbf{v} = \sqrt{\mathbf{b}} = \sqrt{\mathbf{F} \mathbf{F}^T}, \quad (3.34b)$$

where \mathbf{C} and \mathbf{b} are the right and left Cauchy-Green strain matrices respectively. The work-conjugate stress and strain measures used here are logarithmic strain $\boldsymbol{\varepsilon}$ and Kirchhoff stress $\boldsymbol{\tau}$ [144], which are given as [71]

$$\boldsymbol{\varepsilon} = \frac{1}{2} \ln \mathbf{b}, \quad \boldsymbol{\tau} = J \boldsymbol{\sigma}, \quad (3.35)$$

where J is the determinant of the deformation gradient \mathbf{F} and $\boldsymbol{\sigma}$ is the Cauchy stress. The Numerical implementations are now given separately for the finite deformation in the case of both updated Lagrangian and total Lagrangian.

3.3.1 Updated Lagrangian formulation

In updated Lagrangian, the geometry is updated at each solution step, which include the update of nodal coordinates, Gauss points and domain of influences. The shape function and the corresponding derivative are also calculated at each solution step. In updated Lagrangian formulations, the updated deformation gradient \mathbf{F}_{n+1} at the end of increment $n + 1$ is written as

$$\mathbf{F}_{n+1} = \Delta \mathbf{F} \mathbf{F}_n, \quad \Delta \mathbf{F} = [\mathbf{I}_3 - \delta \mathbf{F}]^{-1}, \quad \delta \mathbf{F} = \sum_{i=1}^L \Delta \mathbf{u}_i \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{bmatrix}_i^T, \quad (3.36)$$

where $\Delta \mathbf{F}$ is an increment in the deformation gradient, \mathbf{F}_n is the value of the deformation gradient at the end of previous increment, \mathbf{I}_3 is a 3×3 unit matrix in this case, L is the number of nodes in support, $\Delta \mathbf{u}_i$ is the vector of incremental nodal parameters (or fictitious nodal values) and the shape function derivatives are calculated with reference to the updated geometry. The trial elastic left Cauchy-Green strain matrix \mathbf{b}_{tr}^e is written as

$$\mathbf{b}_{tr}^e = \Delta \mathbf{F} \mathbf{b}_n^e \Delta \mathbf{F}^T, \quad (3.37)$$

where \mathbf{b}_n^e is the value of the elastic left Cauchy-Green strain matrix at the end of previous increment and is obtained by rearranging Equation (3.35) in terms of \mathbf{b} and using $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_n^e$. Equation (3.37) can be used in Equation (3.35) to calculate the trial elastic strain $\boldsymbol{\varepsilon}_t^e$, which is input to the constitutive model. The internal global nodal forces in this case can be found using Equation (3.26) but now the \mathbf{B}_j is the strain displacement matrix calculated with reference to the updated nodal coordinates and Gauss points. In this case, the equation for the global stiffness matrix is written as

$$K_{ij} = \int_{\Omega} \mathbf{G}_i^T \mathbf{a} \mathbf{G}_j d\Omega, \quad (3.38)$$

where \mathbf{G} is a full strain-displacement matrix, calculated with reference to the updated nodal coordinates and Gauss points and is written as

$$\mathbf{G}_i = \begin{bmatrix} \frac{\partial \phi_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial \phi_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial \phi_i}{\partial z} \\ \frac{\partial \phi_i}{\partial y} & 0 & 0 \\ 0 & \frac{\partial \phi_i}{\partial x} & 0 \\ 0 & \frac{\partial \phi_i}{\partial z} & 0 \\ 0 & 0 & \frac{\partial \phi_i}{\partial y} \\ 0 & 0 & \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial z} & 0 & 0 \end{bmatrix} \quad (3.39)$$

and \mathbf{a} is the isotropic spatial consistent or algorithmic tangent and is written as

$$\mathbf{a} = \frac{1}{2J} \mathbf{D}^{alg} \mathbf{L} \mathbf{B}^a - \mathbf{S} \quad (3.40)$$

where

$$\mathbf{L} = \frac{\partial \ln(\mathbf{b}_{tr}^e)}{\partial \mathbf{b}_{tr}^e}, \quad (\mathbf{B}^a)_{ijkl} = \delta_{ik} (\mathbf{b}_{tr}^e)_{jl} + \delta_{jk} (\mathbf{b}_{tr}^e)_{il}, \quad (\mathbf{S})_{ijkl} = (\boldsymbol{\sigma})_{il} \delta_{jk}. \quad (3.41)$$

Here \mathbf{D}^{alg} is the small strain consistent or algorithmic tangent, \mathbf{L} is the derivative of the logarithm of \mathbf{b}_{tr}^e with respect to its component, the detail of which is given in [218], while \mathbf{S} is known as the non-symmetric stress corrector and δ_{ij} is the Kronecker delta.

3.3.2 Total Lagrangian formulation

The basic Newton-Raphson algorithm in the same for both updated and total Lagrangian formulations but there are some differences when calculating some of parameters, which are given in this section. In total Lagrangian formulation, there is no need to update the geometry as the variables are referred back to the original configuration. The shape functions and the corresponding derivatives are calculated and stored at the start of the simulation and are used in every solution step. In this case, the deformation gradient \mathbf{F}_{n+1} for increment $n+1$ is calculated from the total nodal parameters or fictitious nodal values \mathbf{u}_i as compared to the updated Lagrangian formulations, where \mathbf{F}_{n+1} is calculated from incremental nodal parameters or fictitious nodal values $\Delta \mathbf{u}_i$. The increment in the deformation gradient $\Delta \mathbf{F}$ is then calculated using the deformation gradient \mathbf{F}_n of the previous converged iteration.

$$\Delta \mathbf{F} = \mathbf{F}_{n+1} \mathbf{F}_n^{-1}, \quad \mathbf{F}_{n+1} = \mathbf{I}_3 - \sum_{i=1}^L \mathbf{u}_i \left[\begin{array}{c} \frac{\partial \phi}{\partial X} \\ \frac{\partial \phi}{\partial Y} \\ \frac{\partial \phi}{\partial Z} \end{array} \right]_i^T, \quad (3.42)$$

where the shape function derivatives are calculated with reference to the original configuration. In this case equation for the global internal force vector Equation (3.26) is written as

$$\mathbf{f}_{n+1}^{int} = \int_{\Omega} \mathbf{G}^T \mathbf{P} d\Omega = \sum_{i=1}^{n_g} \mathbf{G}_i^T \mathbf{P}_i |\mathbf{J}_i| w_i, \quad (3.43)$$

where \mathbf{G} is the full strain-displacement matrix, consisting of the shape function derivatives with respect to the original configuration and \mathbf{P} is the nine component non-symmetric first Piola-Kirchhoff stress and

is given as

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T} = \boldsymbol{\tau}\mathbf{F}^{-T}, \quad (3.44)$$

$\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ is the Cauchy and Kirchhoff stress respectively. The equation for the global stiffness matrix is written as

$$\mathbf{K} = \int_{\Omega} \mathbf{G}^T \tilde{\mathbf{A}} \mathbf{G} d\Omega, \quad (3.45)$$

where $\tilde{\mathbf{A}}$ is the isotropic material stiffness tangent and is written as

$$\tilde{\mathbf{A}} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}} = \frac{\partial \boldsymbol{\tau} \mathbf{F}^{-T}}{\partial \mathbf{F}}, \quad (3.46)$$

The partial derivative is expressed as

$$\tilde{A}_{ijkl} = \frac{\partial \tau_{ip} F_{jp}^{-1}}{\partial F_{kl}} = \frac{\partial \tau_{ip}}{\partial F_{kl}} F_{jp}^{-1} - P_{ik} F_{jk}^{-1}. \quad (3.47)$$

After using the chain rule The partial derivative $\frac{\partial \tau_{ip}}{\partial F_{kl}}$ is written as

$$\frac{\partial \tau_{ip}}{\partial F_{kl}} = \frac{\partial \tau_{ip}}{\partial (\varepsilon_t^e)_{ab}} \frac{\partial (\varepsilon_t^e)_{ab}}{\partial (b_t^e)_{cd}} \frac{\partial (b_t^e)_{cd}}{\partial F_{kl}}, \quad (3.48)$$

where $\frac{\partial \tau_{ip}}{\partial (\varepsilon_t^e)_{ab}} = D_{ipab}^{alg}$ is infinitesimal consistent or algorithmic tangent and $\frac{\partial (\varepsilon_t^e)_{ab}}{\partial (b_t^e)_{cd}} = L_{abcd}$ is the the partial derivative of logarithm of b_t^e with respect to its components and is written as

$$\frac{\partial (b_t^e)_{cd}}{\partial F_{kl}} = \delta_{ck} ((F_n^{-1})_{lw} (b_n^e)_{wv} \Delta F_{dv}) + \delta_{ck} (\Delta F_{cw} (b_n^e)_{wv} (F_n^{-1})_{lv}). \quad (3.49)$$

3.4 Numerical examples

Numerical examples are now presented to demonstrate the use of EFGM with max-ent shape function in modelling of problems with material and geometrical nonlinearities using the updated Lagrangian formulation. The first two examples, i.e. the infinite plate strip and the elastic cantilever beam are elastic (no plasticity) problems with finite deformation, while the third example of the double notched tensile specimen is a small strain plasticity problem. The final examples of thick walled cylinder expansion include both finite deformation and elasto-plasticity.

3.4.1 Infinite plate strip

Two different infinite plate strips subjected to uniformly distributed load q and both with different types of boundary conditions, i.e. simply supported and fully fixed on both sides are analysed in this section, as shown in Figure 3.4. High membrane stresses are the main source of nonlinearities in this problem, as the deflection is small as compared to the plate thickness. The dimensions of the plate are, length L in x direction, thickness h in z direction and with an infinite width in y or out of plane direction. The analytical solution for this problem is given in [294], where it is given that the solution of the finite length plate approaches the one with infinite length when $\frac{L}{h} > 3$. The same problem is also solved in [221] to validate the FEA finite deformation solution. The solution for the deflection w at a distance x is given as [221, 294]

$$w = \frac{qL^4}{16u^4D} \left[\frac{\cosh u \left(1 - \frac{2x}{L}\right)}{\cosh u} - 1 \right] + \frac{qL^2x}{8u^2D} (L - x) \quad (3.50)$$

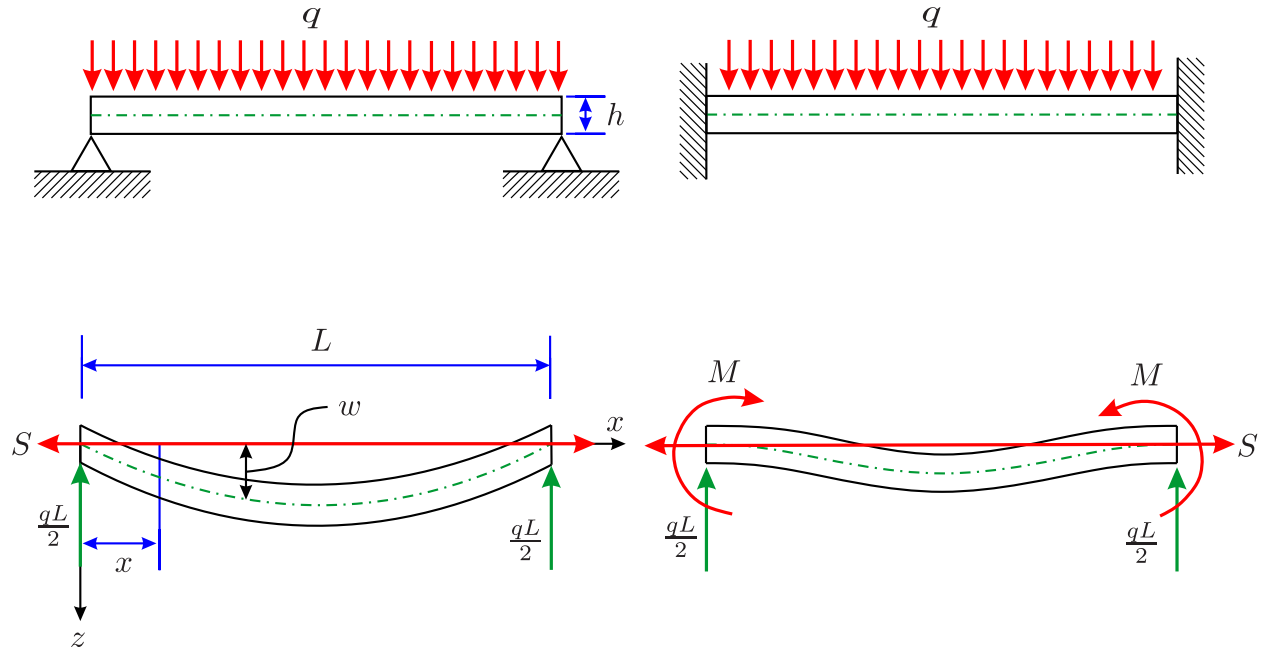


Figure 3.4: Infinite plate strip subjected to uniformly distributed load

where $D = \frac{Eh^3}{12(1-\nu^2)}$ is the flexural rigidity of the plate, E is the Young's modulus, ν is the poisson's ratio and $u^2 = \frac{SL^2}{4D}$, is found using

$$\frac{135}{16u^9} \left[\tanh u + \frac{u \tanh^2 u}{5} - u + \frac{2u^3}{15} \right] = \frac{E^2 h^8}{(1-\nu^2)^2 q^2 L^8}. \quad (3.51)$$

where S is the in-plane force. The maximum neutral axis deflection at $x = \frac{L}{2}$ is also given as

$$w_{max} = \frac{5qL^4}{384D} \left[\frac{\text{sech } u - 1 + \frac{u^2}{2}}{\frac{5u^4}{24}} \right], \quad (3.52)$$

For the plate with fully fixed boundary conditions, Equation (3.50) is given as

$$w = \frac{qL^4}{16u^3 D \tanh u} \left[\frac{\cosh \left(u \left(1 - \frac{2x}{L} \right) \right)}{\cosh u} - 1 \right] + \frac{qL^2 x}{8u^2 D} (L - x). \quad (3.53)$$

where u in this case can be found from

$$\frac{135}{16u^9} \left[\frac{2u^3}{15} - \frac{u^3}{5 \sinh^2 u} + \frac{4u}{5} - \frac{3u^2}{5 \tanh u} \right] = \frac{E^2 h^8}{(1-\nu^2)^2 q^2 L^8}, \quad (3.54)$$

and maximum neutral axis deflection at $x = \frac{L}{2}$, i.e. Equation (3.52) is written as

$$w_{max} = \frac{5qL^4}{384D} \left[\frac{24}{u^3} \left(\frac{u}{2} + \text{csch } u - \coth u \right) \right] \quad (3.55)$$

The problem is solved using an updated Lagrangian formulation with max-ent shape functions with $L = 5$, $h = 0.2$, $E = 1 \times 10^7$, $\nu = 0.25$ and $q = 40$, all in compatible units. The total pressure is applied in 20 equal step. Due to symmetry only half of the plate is analysed with 93 (31×3) nodes and 60 (30×2) background integration cells as shown in Figures 3.5(a) and 3.5(b). The way to impose the

essential boundary conditions and the final deformed and undeformed configuration with contours of u_y for both the cases are given in Figures 3.5(a) and 3.5(b). Comparison for the pressure versus displacement with the analytical solution for both cases are given in Figure 3.6(a) and 3.6(b), which are in excellent agreement. The deflected profile of the plate is also compared with the analytical solution for both cases and are shown in Figures 3.7(a) and 3.7(b) and are in excellent agreement. It is also clear from Figures 3.6(a) and 3.6(b) that the response of the plate with simply supported boundary conditions is highly nonlinear compared to the one with fully fixed boundary conditions.

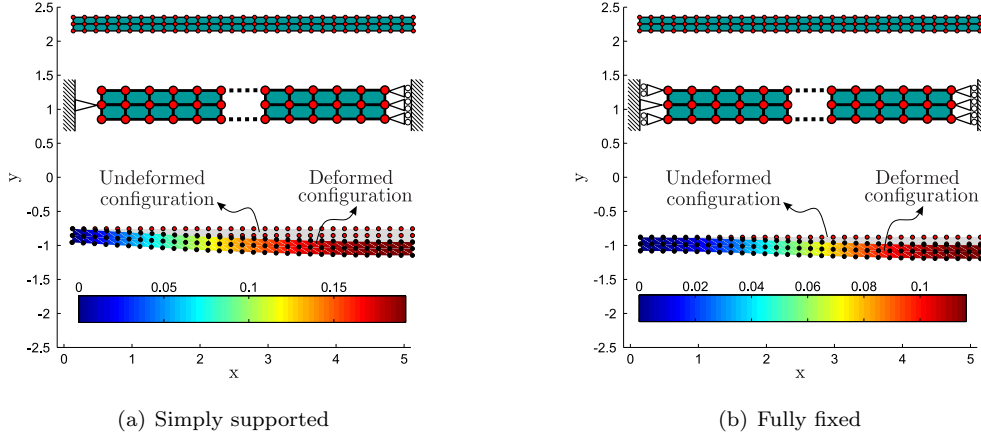


Figure 3.5: Discretization, imposition of essential boundary conditions and undeformed and deformed configuration for the infinite plate strip problem

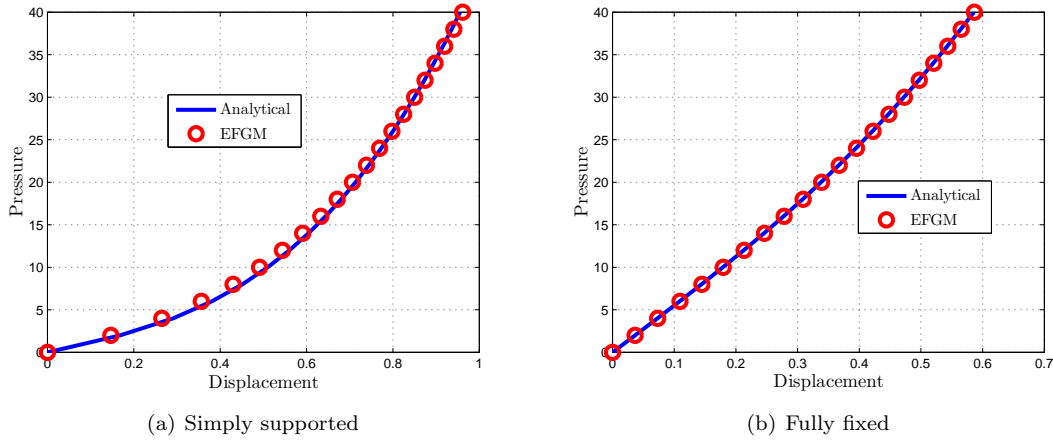
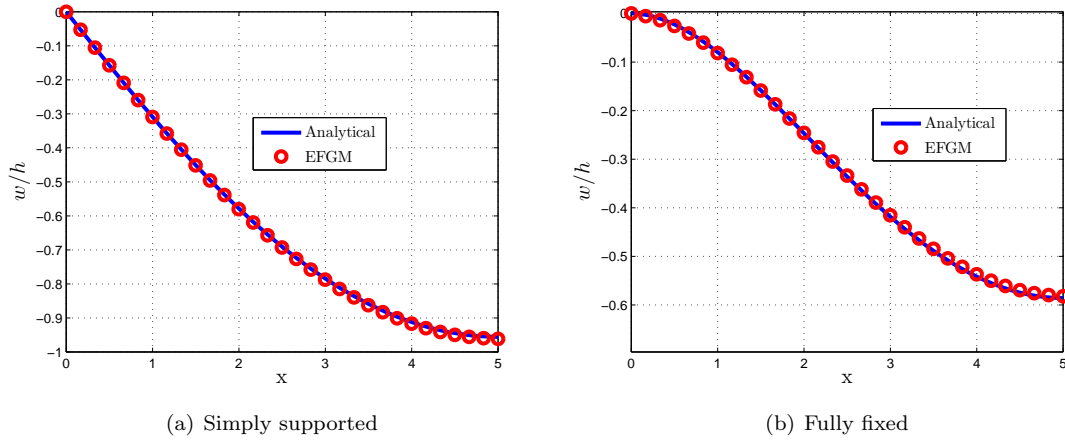


Figure 3.6: Pressure versus displacement for the infinite plate strip problem

3.4.2 Elastic cantilever Beam

The second problem to demonstrate the implementation of the finite deformation is an elastic cantilever beam. Two different cantilever beams are analysed in this section, one with a uniformly distributed load (UDL) and the other with tip load. The geometry, boundary condition and loading with deformed and undeformed configuration for the the cantilever beam with UDL is shown in Figure 3.8(a), where L , H , B are the length, height and breadth in the x , y and z direction, u_x and u_y are the deformations in x and y direction respectively. The beam is subjected to UDL of q . The problem is solved with $L = 10$, $H = 1$, $B = 1$, $E = 1.2 \times 10^7$, $\nu = 0.2$ and $q = 10$. The discretization parameters used are 63 (21×3)

Figure 3.7: Deflected profile or w/h versus x for the infinite plate strip problem

nodes and $40 (20 \times 2)$ background integration cells. The pressure is applied to the neutral axis in 20 equal increments and 1×10^6 is used as convergence criterion for the Newton's algorithm. The undeformed and final deformed configuration is also shown in Figure 3.8(b), which is clearly a very large deformation. Figures 3.9(a) and 3.9(b) shows the comparison of pressure versus u_x and u_y with the reference solution from [129], which are in good agreement with the reference solution. The same problem is used in [221] to validate the application of FEA in finite deformation problems. The geometry, boundary condition

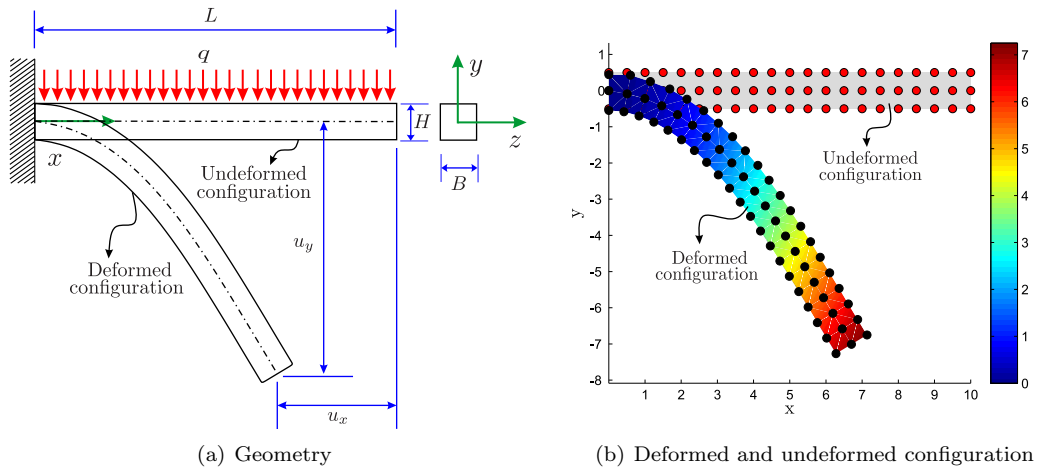


Figure 3.8: Geometry with undeformed and deformed configuration of the elastic cantilever beam problem with UDL

and loading for the cantilever beam subjected to tip load are shown in Figure 3.10(a), where all the parameters are the same as for the beam with UDL but P is the tip load and θ is the angle of rotation or the angle which the neutral axis makes with the horizontal after deformation. For this problem, all the material, geometry and discretization parameter are also the same as the beam with UDL. In this case load $P = 100$ is applied at the tip of neutral axis with 20 equal load steps. Figure 3.10(b) shows the undeformed and final deformed configuration for this problem, which is clearly a finite deformation problem. Figures 3.11(a), 3.11(b) and 3.11(c) shows the comparison of P versus u_x , u_y and θ with the reference solution [221], which shows excellent agreement. The same problem is also solved in [67] using FEA.

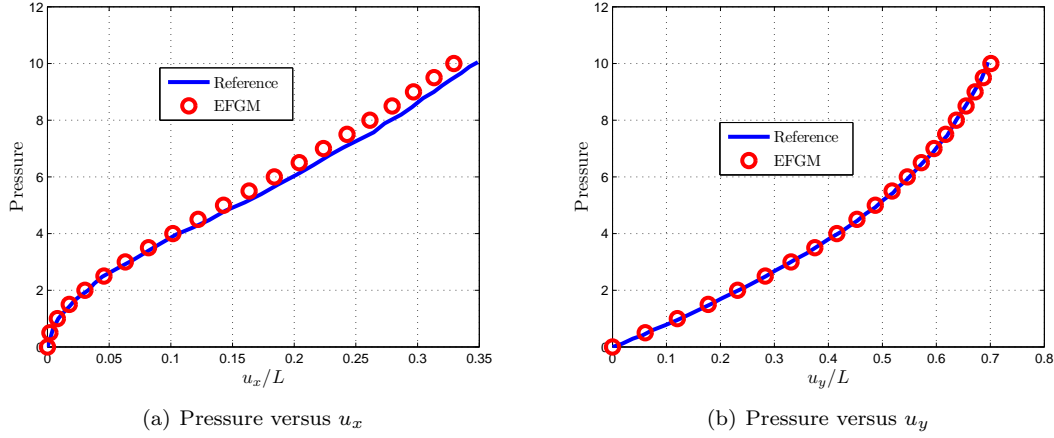
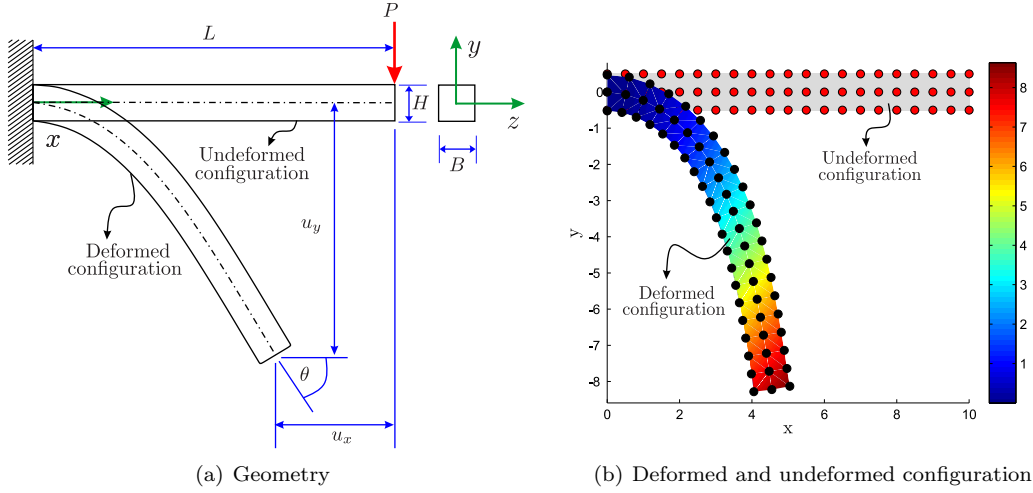
Figure 3.9: Pressure versus u_x and u_y for the elastic cantilever beam problem with UDL

Figure 3.10: Geometry with undeformed and deformed configuration of the elastic cantilever beam problem with tip load

3.4.3 Double notched tensile specimen

The third problem is the deep double notched tensile specimen, which is the standard benchmark for the incompressible small strain plasticity and can be found in a number of reference, e.g. [67, 81, 230]. The geometry with boundary conditions and loading for this problem is shown in Figure 3.12. The dimension of the tensile specimen used in this case are, height $h = 30$ and width $w = 10$. The specimen consists of two very deep notches in the center, while the lower and upper portions are connected by a ligament with dimension $b = 2$. A displacement u is applied at the upper and lower edges of the specimen. Due to symmetry only a quarter of the upper right shown in gray in Figure 3.12 is analysed. The material properties used in this case are $E = 206.9$, $\nu = 0.29$, $\sigma_y = 0.45$. The problem is discretized with 96 (6×16) nodes and 75 (5×15) background cells as shown in the right of Figure 3.12, while (4×4) gauss points are used in each background cell. A total deflection of $u = 0.17$ is applied in 20 equal increments. The plastic gauss points at the end of increments 2, 5, 10, 15, 20 are shown in red in Figure 3.13. The plastic region increases incrementally starting from 2nd iteration and the full formation of the shear band can be seen at the 20th iteration. The final contours of u_x and u_y is also shown in Figure 3.14(a). In this problem, if R represents the total reaction on the upper edge, then the net axial stress $\bar{\sigma}$ on the ligament

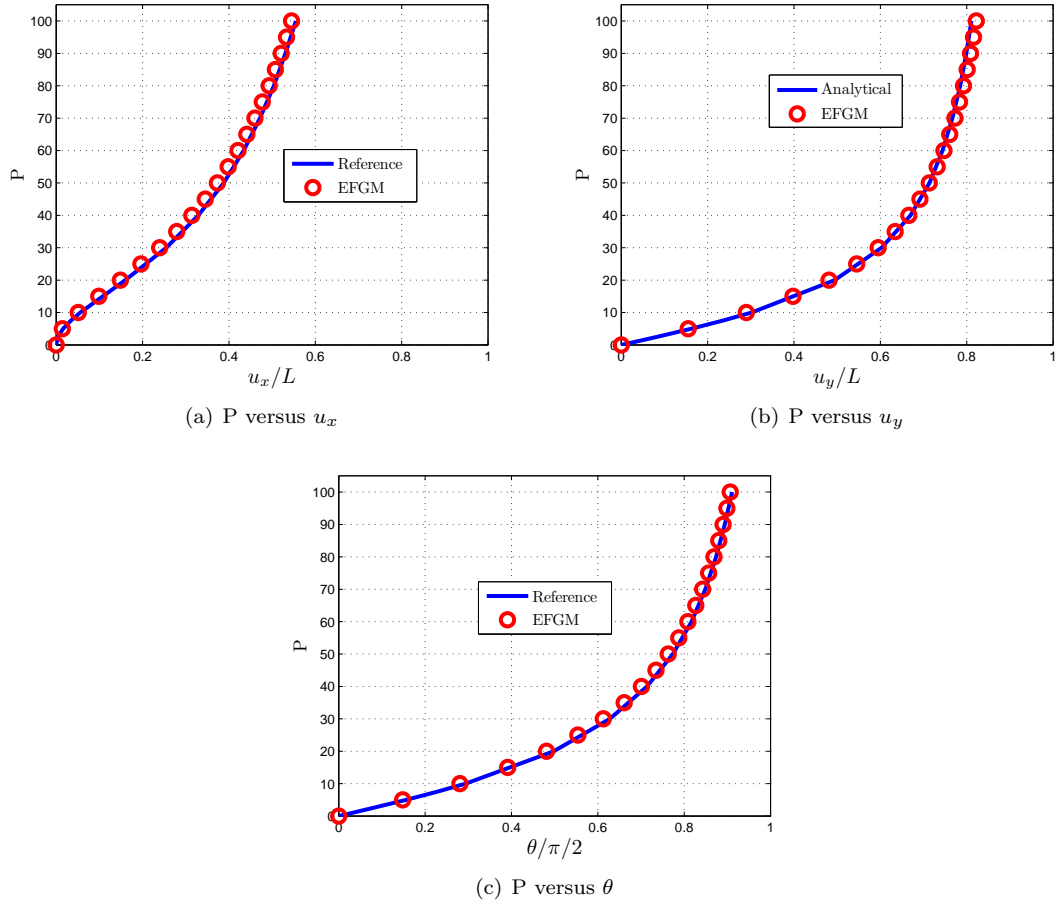


Figure 3.11: P versus u_x , u_y and θ for the elastic cantilever beam problem with tip load

is written as [81]

$$\bar{\sigma} = \frac{R}{b}. \quad (3.56)$$

The limiting net axial stress on the ligament $\bar{\sigma}_{lim}$ in the case of perfect plasticity is determined by Prandtl and is given as [81, 230]

$$\bar{\sigma}_{lim} = \frac{(2 + \pi)}{\sqrt{3}} \sigma_y \approx 2.97 \sigma_y. \quad (3.57)$$

The final ligament net stress normalized over the yield strength, $\bar{\sigma}/\sigma_y$ versus edge deflection is shown in Figure 3.14(b). For comparison the Prandtl limiting load and plot is also given from [81] and a very good agreement can be seen, and the plot converges to the Prandtl limiting net axial stress.

3.4.4 Thick wall cylinder expansion

The final example is the expansion of the thick walled cylinder and is the standard benchmark for finite deformation with plasticity and can be found in a number of references, e.g. [67, 271–273]. A thick walled cylinder with internal radius of $r_i = 10$ and outer radius $r_o = 20m$ is expended to the internal radius $r_o = 85m$. The material properties chosen in this case are $E = 11.05 GPa$, $\nu = 0.454$ corresponding to shear modulus $\mu = 3.8 GPa$ and bulk modulus $k = 40 GPa$ and $\sigma_y = 0.5 MPa$, so that comparison can

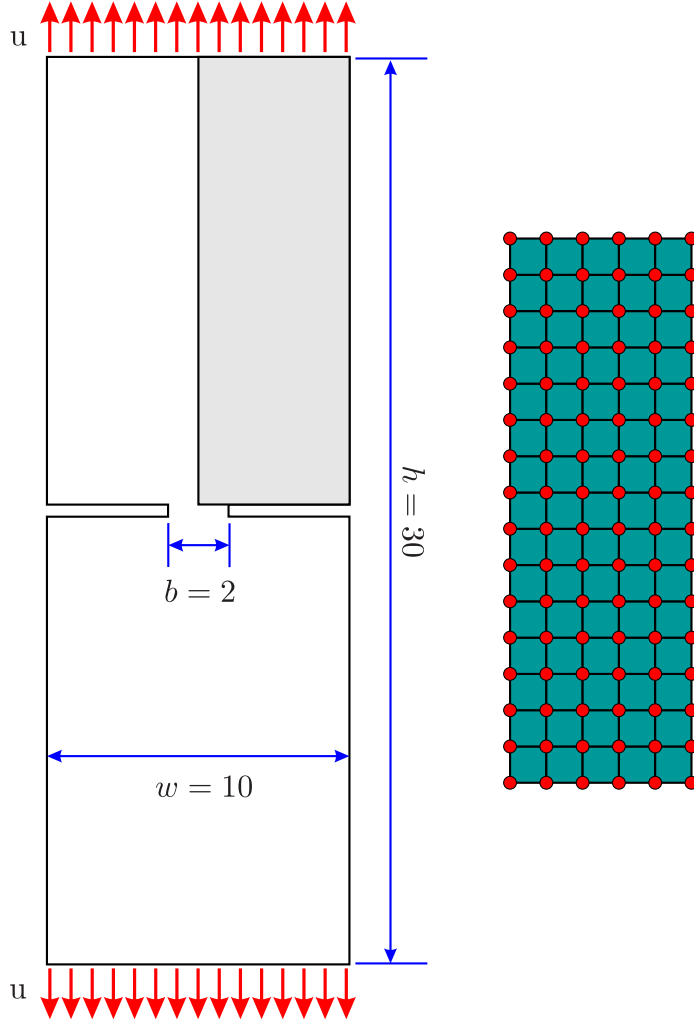


Figure 3.12: Geometry, boundary conditions, loading and discretization for double notched tensile specimen

be made with the rigid plasticity analytical solution given in [127] as

$$p = \frac{2\sigma_y}{\sqrt{3}} \left(\frac{\sqrt{r_o^2 - r_i^2} + r^2}{r} \right) \quad (3.58)$$

where p is the internal pressure and r is the current internal radius. The problem is solved using the standard axisymmetric approach with polar coordinates [283, 336] with x and y in this case are the radial and axial coordinates for the cylinder. In this case, the following modification needs to be made to the standard formulations.

1. The strain displacement matrices \mathbf{B} and \mathbf{G} in Equations, 2.44 and 3.39 are written as

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial \phi_i}{\partial x} & 0 \\ 0 & \frac{\partial \phi_i}{\partial y} \\ \frac{\phi_i}{x} & 0 \\ \frac{\partial \phi_i}{\partial y} & \frac{\partial \phi_i}{\partial x} \end{bmatrix}, \quad \mathbf{G}_i = \begin{bmatrix} \frac{\partial \phi_i}{\partial x} & 0 \\ 0 & \frac{\partial \phi_i}{\partial y} \\ \frac{\phi_i}{x} & 0 \\ 0 & \frac{\partial \phi_i}{\partial x} \end{bmatrix}, \quad (3.59)$$

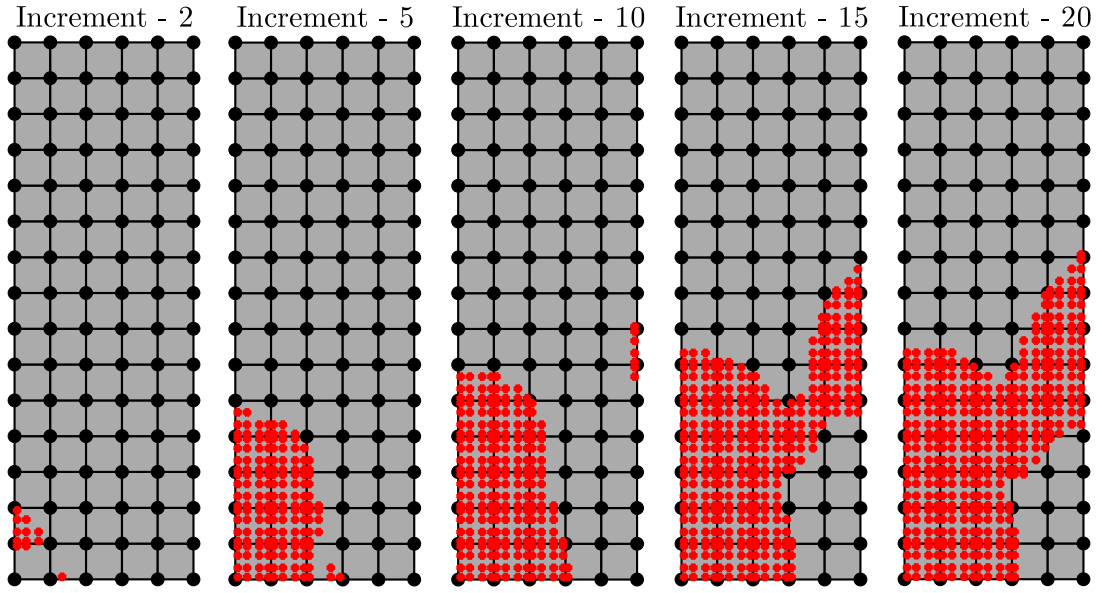
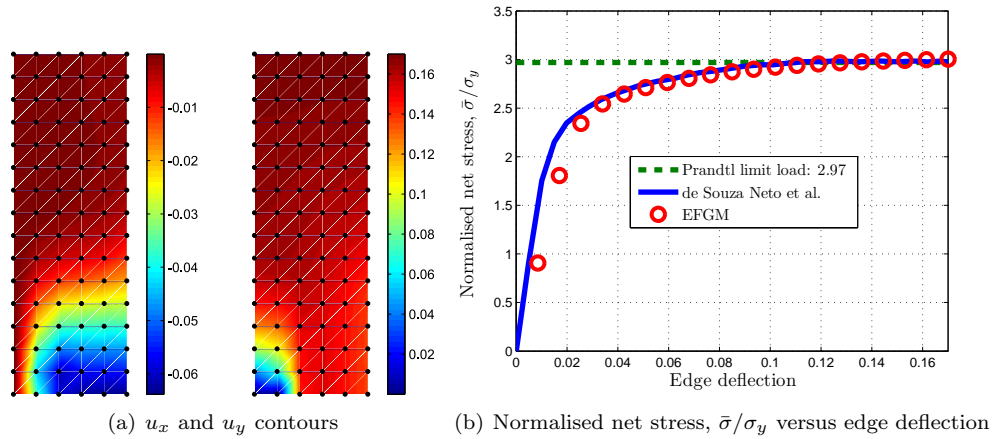


Figure 3.13: Plastic gauss points at different increments for double notched tensile specimen

Figure 3.14: u_x , u_y contours and normalized net stress, $\bar{\sigma}/\sigma_y$ versus edge deflection for double notched tensile specimen

where x is the updated radial position of the Gauss point.

2. The Equations for internal force (3.26) and stiffness matrix (3.38) are written as

$$\mathbf{f}_{n+1}^{int} = \sum_{i=1}^{n_g} \mathbf{B}_i^T \boldsymbol{\sigma}_i |J_i| w_i x_i, \quad (3.60)$$

$$\mathbf{K}_{n+1} = \sum_{i=1}^{n_g} \mathbf{G}_i^T \mathbf{a} \mathbf{G}_i |J_i| w_i x_i. \quad (3.61)$$

The problem geometry with the axisymmetric discretization and the corresponding boundary condition are shown in Figure 3.15, i.e. the problem is discretized with 22 nodes and 10 background cells. The total internal displacement of 75 m is applied in 30 equal steps and the comparison of internal pressure versus current internal radius with the analytical solution is shown in Figure 3.16, which are in excellent

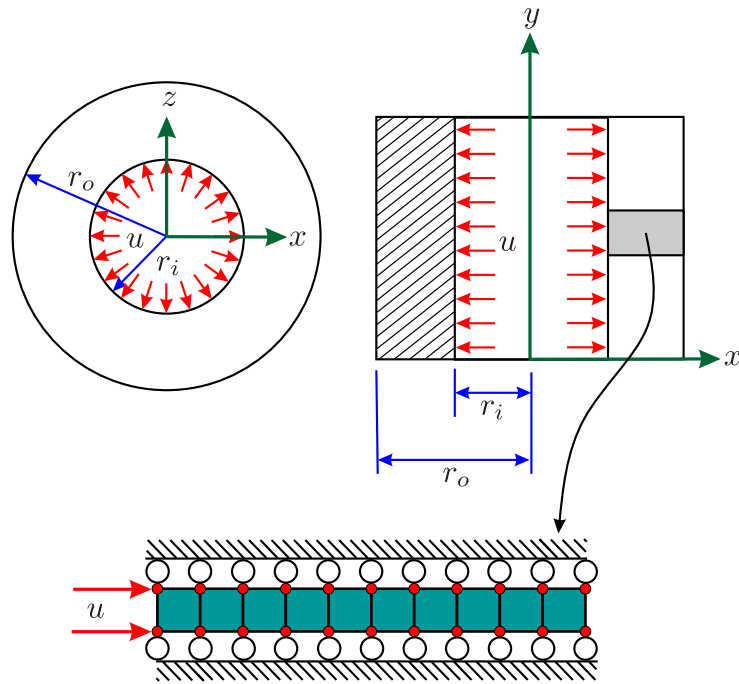


Figure 3.15: Geometry, discretization and boundary conditions for the thick walled cylinder problem

agreement.

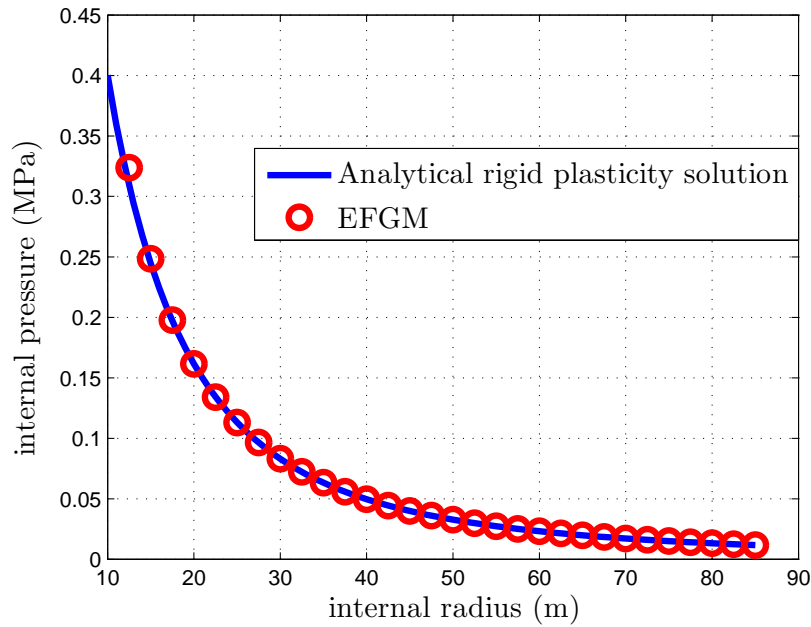


Figure 3.16: Internal pressure versus current internal radius for the thick walled cylinder problem

3.5 Concluding remarks

In this chapter EFGM with max-ent shape functions has been used for modelling problems subjected to elasto-plasticity and finite deformation. Detailed formulations of the elasto-plasticity was given followed by its implementation in the finite deformation framework, including updated and total Lagrangian.

Different benchmark numerical examples subjected to elasto-plasticity and finite deformations are solved to demonstrate the implementation and performance of the current approach. Excellent agreement has been shown between the analytical or reference results and the corresponding EFGM results in case of all the attempted benchmark problems. The formulation developed throughout this chapter will be extended to nonlinear adaptive analysis in the next chapter, which include error estimation, refinement and the projection of path dependent variables from one discretization to the next one.

Chapter 4

Error estimation and adaptivity in element-free Galerkin method

4.1 Introduction

Efficient computational modelling of problems, including both material and geometric nonlinearities remains challenging. Often these problems are modelled with an adaptive FEM due to their complicated nature and high computational cost. For nonlinear problems, adaptive analysis consists of estimation of the discretization error, which is the prime source of error in computational mechanics, with a suitable refinement strategy and projection of the path dependent variable from the old discretization to the new one. Meshless methods offer the attractive possibility of simpler adaptive procedures involving no remeshing, simply insertion or deletion of nodes. In this chapter, an adaptive meshless approach for nonlinear solid mechanics is developed based on the element-free Galerkin method with updated Lagrangian formulations for two- and three-dimensional problems. An existing error estimation procedure for two-dimensional linear elasto-static problems, is extended here for two- and three-dimensional nonlinear problems, including finite deformation and elasto-plasticity, and a new adaptive procedure is described and demonstrated.

It is clearly recognized in FE modelling that adaptive procedures based on robust error estimates are necessary to remove user-bias and to automate analyses, and mature procedures are now available, many based on recovery type measures such as the famous Zienkiewicz-Zhu approach [334, 335]. Error estimation and adaptivity have yet to become widely used in meshless methods partly because of the current lack of a firm mathematical basis for error estimation as has been developed for FEs. However, there is evidence of a strong interest as indicated in the following. Chung and Belytschko [62] described possibly the first error estimator for the EFGM using the difference between raw EFGM results and projected stresses from the same nodal distribution but with reduced domains of influence. In [137] error estimation and adaptivity are performed using stress gradients, thus requiring second derivatives of the meshless shape functions. In [105] a computationally efficient method for error estimation based on tessellation is proposed for the EFGM, i.e. error at a Gauss point is calculated as the difference between the EFGM stress or strain and that calculated at the nearest node. A slight change is made in [104] where comparison is instead made with a field calculated using a first-order Taylor series expansion with a four quadrant criterion, and the two approaches are compared by the same authors in [103]. [256] describes an error estimate for the EFGM based on a Taylor series with a higher-order derivative, and a structured grid is used instead of a cloud of points, which makes the implementation very straightforward.

Error estimation based on the gradient of the strain energy density is proposed in [211], and an adaptive analysis for EFGM is proposed in [197] based on background cells, the error being estimated based on two different integration orders, and a refinement algorithm based on local Delaunay triangulation is also proposed. In [184], the approach from Chung & Belytschko [62] is used for error estimation and adaptive analysis in crack propagation problems. A Zienkiewicz-Zhu recovery type error estimator is proposed in [182]; two methods are used for stress recovery, including that due to Chung & Belytschko [62] with the discrete MLS over a stationary least square fitting, and it is found that the approach of [62] is more effective, and is used further for adaptive analysis in [183].

The survey of the current literature reveals that to date the adaptive meshless methods have been limited in their use to elasto-static problems with infinitesimal strains. In this chapter, we incorporate a new adaptive procedure into the max-ent EFGM for two- and three-dimensional material and geometrically nonlinear problems with updated Lagrangian formulations, including a robust means of transferring data between discretizations.

This chapter is structured as follows. Adaptive procedures for the linear elastic EFGM are discussed in §4.2, including error estimation formulation with one- and two-dimensional numerical examples given in §4.2.1. The refinement strategy for two-dimensional problems is discussed in §4.2.2 and the automatic way to calculate the two-dimensional nodal domains of influence based on Voronoi diagram is given in §4.2.3. A number of two-dimensional adaptive linear elastic numerical examples are given in §4.2.4 to show the performance of the current approach. The adaptive procedure developed is extended to two-dimensional nonlinear problems in §4.3, including error estimation in §4.3.1, transfer of the path dependent variables between the consecutive discretization is also explained in §4.3.2 and a number of numerical examples are given in §4.3.3. The two-dimensional formulation is extended to three-dimensional nonlinear problems in §4.4 and the required changes to the two-dimensional formulation and implementation are also explained. A number of nonlinear three-dimensional numerical examples are also given in §4.4.1 to show the implementation and performance of the proposed approach in case of three-dimensional problems. Concluding remarks are given in §4.5.

4.2 Adaptivity in linear-elastic problems

The error estimation algorithms, when applied to FEs, work element-by-element. For the EFGM, no elements are present so instead we assess error quadrature cell by quadrature cell. The main components required in an adaptive algorithm for linear analysis are as follows:

1. estimation of discretization error;
2. refinement strategy;

Each are discussed with reference to the meshless modelling used here in turn below.

4.2.1 Error estimation

Local and global recovery type error estimation procedures are given in this section, based on the procedure proposed in [62], which is confined to linear elasto-static problems. The basic idea of this error estimation procedure is to use the difference between the projected and the direct EFGM solutions and is the same as the conventional recovery type error estimation in the FEM [335]. In the FEM, strain and the corresponding stress fields are discontinuous at the inter-element boundaries, and different stress smoothing techniques have been used to estimate continuous strain and stress fields. This continuous strain and stress field is then used as a projected solution for the recovery type error estimation in the FEM. As there are no elements in the EFGM, so there is no issue of strain and stress discontinuity, and

the procedure used for the error estimation in case of the FEM is not directly applicable to the EFGM, because the derived stress and strain fields are already smooth. The error estimation procedure proposed for the EFGM in [62] is used here for the error estimation, in which projected stresses are calculated from the nodal stresses based on reduced domains of influence. It was also shown in [62] that the error estimator performs well, when the domain of influence for stress projection is as small as possible. A conventional [332] recovery type error estimation procedure is used in this case, i.e. the error is initially calculated locally at the Gauss (or integration) points, and then the global error is calculated integrating over the problem domain using an appropriate norm.

The exact error in stress or strain fields at a point \mathbf{x} is the difference between the exact and numerical (EFGM) values, i.e.

$$\boldsymbol{\sigma}^e(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{x}) - \boldsymbol{\sigma}^h(\mathbf{x}), \quad \boldsymbol{\varepsilon}^e(\mathbf{x}) = \boldsymbol{\varepsilon}(\mathbf{x}) - \boldsymbol{\varepsilon}^h(\mathbf{x}) \quad (4.1)$$

where $\boldsymbol{\sigma}^e(\mathbf{x})$ and $\boldsymbol{\varepsilon}^e(\mathbf{x})$ are the exact error in the stress and strain at a point \mathbf{x} respectively, $\boldsymbol{\sigma}(\mathbf{x})$ and $\boldsymbol{\varepsilon}(\mathbf{x})$ are the exact stress and strain at a point \mathbf{x} while $\boldsymbol{\sigma}^h(\mathbf{x})$ and $\boldsymbol{\varepsilon}^h(\mathbf{x})$ are the EFGM stress and strain at a point \mathbf{x} . Error for the individual cell and for the whole domain can then be found using an appropriate norm, error in energy norm is used in this research and is written as

$$\|e\| = \left[\int_{\Omega} \boldsymbol{\sigma}^e(\mathbf{x})^T \mathbf{D}^{-1} \boldsymbol{\sigma}^e(\mathbf{x}) d\Omega \right]^{\frac{1}{2}} = \left[\int_{\Omega} \boldsymbol{\varepsilon}^e(\mathbf{x})^T \mathbf{D} \boldsymbol{\varepsilon}^e(\mathbf{x}) d\Omega \right]^{\frac{1}{2}}. \quad (4.2)$$

As the exact stresses are not available for real-life problems, so the projected stress $\boldsymbol{\sigma}^p(\mathbf{x})$ and projected strain $\boldsymbol{\varepsilon}^p(\mathbf{x})$ are used in Equation (4.1) instead of the exact stress and exact strain respectively, i.e.

$$\boldsymbol{\sigma}^e(\mathbf{x}) \approx \boldsymbol{\sigma}_p^e(\mathbf{x}) = \boldsymbol{\sigma}^p(\mathbf{x}) - \boldsymbol{\sigma}^h(\mathbf{x}), \quad \boldsymbol{\varepsilon}^e(\mathbf{x}) \approx \boldsymbol{\varepsilon}_p^e(\mathbf{x}) = \boldsymbol{\varepsilon}^p(\mathbf{x}) - \boldsymbol{\varepsilon}^h(\mathbf{x}) \quad (4.3)$$

In the case of the EFGM, the stress and strain vector returned at any point \mathbf{x} is written as

$$\boldsymbol{\sigma}^h(\mathbf{x}) = \sum_{i=1}^{na} \mathbf{D} \mathbf{B}_i \mathbf{u}_i, \quad \boldsymbol{\varepsilon}^h(\mathbf{x}) = \sum_{i=1}^{na} \mathbf{B}_i \mathbf{u}_i \quad (4.4)$$

where na is the number of nodes in the support of point \mathbf{x} based on the domain of influence for analysis d_m^a and \mathbf{u}_i are fictitious nodal values for node i . The projected stress and strain vector at \mathbf{x} are recovered using

$$\boldsymbol{\sigma}^p(\mathbf{x}) = \sum_{j=1}^{np} \psi_j(\mathbf{x}) \boldsymbol{\sigma}^h(\mathbf{x}_j), \quad \boldsymbol{\varepsilon}^p(\mathbf{x}) = \sum_{j=1}^{np} \psi_j(\mathbf{x}) \boldsymbol{\varepsilon}^h(\mathbf{x}_j). \quad (4.5)$$

Here $\psi_j(\mathbf{x})$ is the shape function of a node j at a point \mathbf{x} based on a reduced domain of influence d_m^p and np is the number of nodes in the support of point \mathbf{x} based on that reduced domain of influence. Equation (4.2) in this case is written as

$$\|e\| \approx \|\bar{e}\| = \left[\int_{\Omega} \boldsymbol{\sigma}_p^e(\mathbf{x})^T \mathbf{D}^{-1} \boldsymbol{\sigma}_p^e(\mathbf{x}) d\Omega \right]^{\frac{1}{2}} = \left[\int_{\Omega} \boldsymbol{\varepsilon}_p^e(\mathbf{x})^T \mathbf{D} \boldsymbol{\varepsilon}_p^e(\mathbf{x}) d\Omega \right]^{\frac{1}{2}}, \quad (4.6)$$

where $\|\bar{e}\|$ is the estimated error in energy norm. To measure the quality of an error estimator, effectivity index is used, which is the ratio of estimated error to the exact error and is given as [62, 335]

$$\theta = \frac{\|\bar{e}\|}{\|e\|}. \quad (4.7)$$

The quality of an error estimator is mainly dependent upon the procedure used to get the projected strains and stresses. For an asymptotically exact error estimator, the discretization error approaches the exact error or θ approaches 1, when $h \rightarrow 0$, where h is the distance between the neighbouring nodes. The relative error in energy norm is often used for adaptive analysis and is written as

$$\eta = \frac{\|\bar{e}\|}{\|U\|} \times 100, \quad (4.8)$$

where $\|U\|$ is the energy norm and is written as

$$\|U\| = \left[\int_{\Omega} \left| \boldsymbol{\sigma}^p(\mathbf{x})^T \mathbf{D}^{-1} \boldsymbol{\sigma}^p(\mathbf{x}) \right| d\Omega \right]^{\frac{1}{2}}. \quad (4.9)$$

Another way to estimate error is point-wise error in energy, i.e. to calculate the error in energy at each Gauss point. The equations for the exact and estimated point-wise error in energy are written as

$$\|e^p\| = \left[\boldsymbol{\sigma}^e(\mathbf{x})^T \mathbf{D}^{-1} \boldsymbol{\sigma}^e(\mathbf{x}) \right]^{\frac{1}{2}} \quad (4.10)$$

$$\|e^p\| \approx \|\bar{e}^p\| = \left[\boldsymbol{\sigma}_p^e(\mathbf{x})^T \mathbf{D}^{-1} \boldsymbol{\sigma}_p^e(\mathbf{x}) \right]^{\frac{1}{2}} \quad (4.11)$$

Numerical Examples

Two linear elastic numerical examples in one- and two-dimensions are now given to get an insight into the error estimation. Cubic spline weight functions are used in all the examples.

One-dimensional bar

The first example is the one-dimensional bar problem where the structure is subjected to body load with the same boundary conditions as shown in Figure 2.15. The problem is solved for two different loading conditions, i.e. $b(x) = 100$ and $b(x) = 100x$ with $E = 1$ and $L = 1$. The problem is discretized with 11 equidistant nodes and 10 background integration cells. The analytical solutions for the stress field for both loading conditions are

$$\sigma = \begin{cases} 100(1-x), & \text{for } b(x) = 100, \\ 100(1-x^2), & \text{for } b(x) = 100x. \end{cases} \quad (4.12)$$

In this case, the domain of influence for analysis and projection is determined using $d_m^a = d_{max}^a h$ and $d_m^p = d_{max}^p h$ respectively, where h is the distance between the two neighbouring nodes and d_{max}^a and d_{max}^p are the scaling parameters for the nodal domain of influence for analysis and projection respectively. Comparison between the numerical (EFGM), projected and analytical stress distribution along the length of the bar, calculated at Gauss points are shown in Figures 4.1(a) and 4.1(b) for each of the loading conditions, i.e. $b(x) = 100$ and $b(x) = 100x$. The scaling parameters used in this case are $d_{max}^a = 1.4$ and $d_{max}^p = 1.2$. From Figures 4.1(a) and 4.1(b) it is clear that the projected stress distribution is closer to the analytical stress distribution than the corresponding EFGM stress distribution. In Figure 4.1(a) the profile of the stress distribution is linear, and the oscillation in the EFGM stress is constant throughout the profile, while in Figure 4.1(b) the profile of stress distribution is parabolic and the oscillation in the EFGM stress increases with an increase in stress gradient.

To determine the optimal combination of d_{max}^a and d_{max}^p for the best performance of the error estimation, the effectivity index (θ) for each of the loading conditions is given in Figure 4.2(a) and 4.2(b).

From these plots, clearly the optimal value of the effectivity index is obtained using a smaller value of d_{max}^p . It is also clear from these plots, that for smaller values of d_{max}^a the value of effectivity index is close to one for all values of d_{max}^p but this may not be an optimal option and is further analysed in Figures 4.3(a) and 4.3(b). In Figure 4.3(a) and 4.3(b), relative error is given for the same combination of d_{max}^a and d_{max}^p , where it is shown that for smaller value of d_{max}^a , the effectivity index is close to one although the relative error is higher.

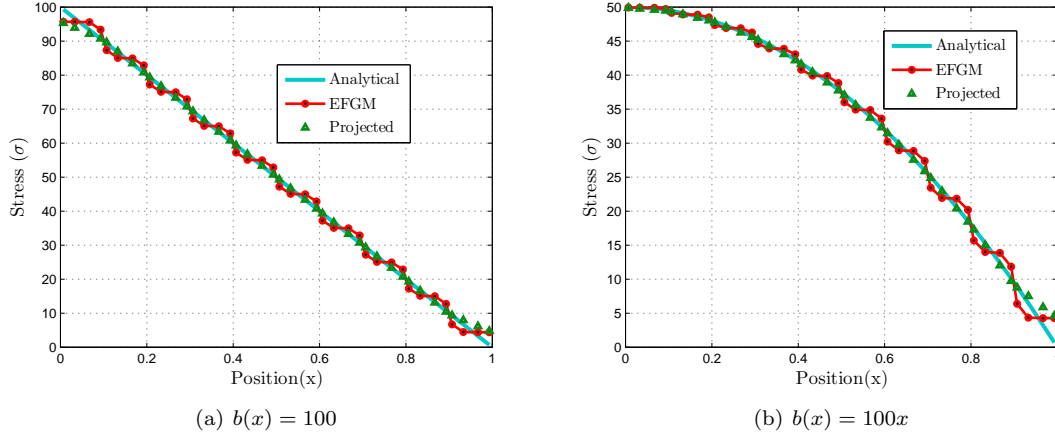


Figure 4.1: Comparison of EFGM, projected and analytical stress at Gauss points using $d_{max}^a = 1.4$ and $d_{max}^p = 1.2$ for one-dimensional bar problem

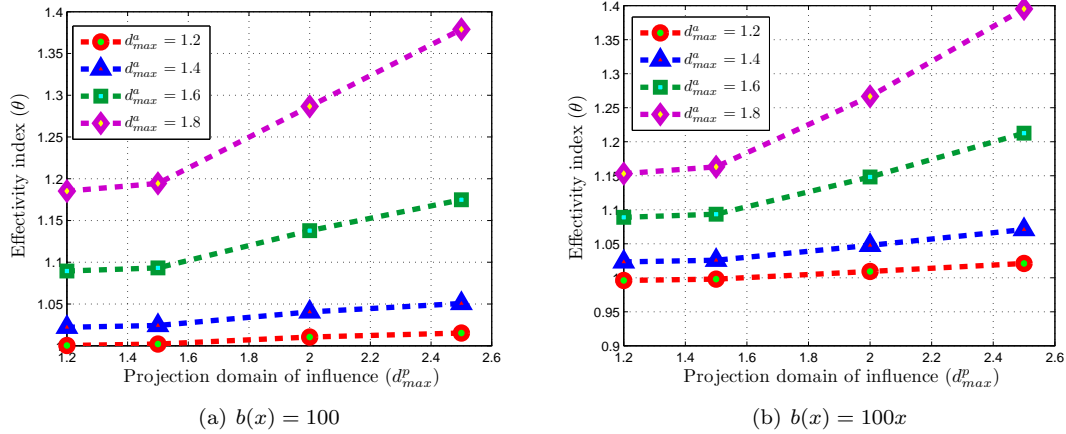


Figure 4.2: Effectivity index (θ) for different d_{max}^a and d_{max}^p values for one-dimensional bar problem

Hole in an infinite plate

The second numerical example is the same hole in an infinite plate problem, with the same geometry, loading and boundary condition as shown in Figure 2.22. The material and geometrical parameters used in this case are the same as given in §2.8. The analytical solution for the displacement and stress fields are given in Equations (2.81) and (2.82) respectively. Two different discretizations are considered for this problem to demonstrate the performance of the error estimator. The first discretization uses 121 nodes and is shown in Figure 4.4(a), while the second uses 441 nodes and is shown in Figure 4.4(b). The problem with 121 nodes is analysed with $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ and the contours of cell-wise error in energy norm are shown in Figure 4.5(a) with relative error in energy norm of 2.16%. The error is higher

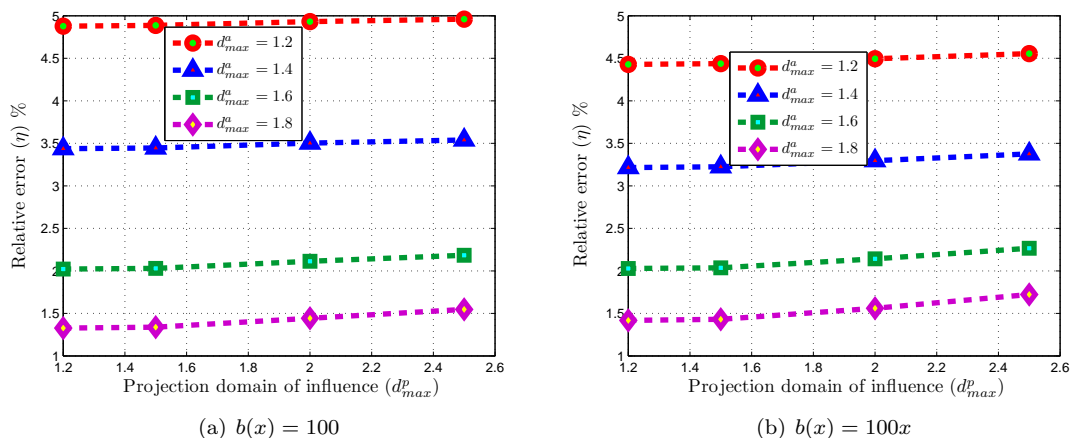


Figure 4.3: Relative error (η) for different d_{max}^a and d_{max}^p values for the one-dimensional bar problem

near the centre of the hole as expected. For comparison, the contours of exact cell-wise error in energy norm are also shown in Figure 4.5(b) with relative error in energy norm of 1.72 %. The contours of the cell-wise error in energy norm and relative error are in close agreement with each other. A comparison is also made between the estimated and exact error for the second discretization with 441 nodes and the contours of the cell-wise error in energy norm are shown in Figures 4.6(a) and 4.6(b), which are in a very good agreement. For this case, corresponding values of the estimated and exact relative error in energy norm are 0.70 % and 0.56 % respectively, which are very close to each other. The relative error in energy norm also decreases from the coarse discretization to the fine discretization.

To find out the optimal combination of parameters for d_{max}^a and d_{max}^p , plots of the effectivity index for a discretization of 121 nodes for different combinations of d_{max}^a and d_{max}^p is shown in Figure 4.7(a). Once again the error estimator gives very good results for smaller value of d_{max}^p , i.e. the effectivity index approaches 1 and increasing the value of d_{max}^p spoils the performance of the error estimator. Similar to the one-dimensional bar problem, for this problem for smaller value of d_{max}^a , the effectivity index is also very close to 1 for all values of d_{max}^p but again is not an optimal combination to use in the analysis and is further analysed in Figure 4.7(b). In Figure 4.7(b) plots for the relative error in energy norm for the same combination of d_{max}^a and d_{max}^p are shown, where clearly for a smaller value of d_{max}^a the relative error is relatively high.

4.2.2 Refinement strategy

Having devised a means of estimating error, we now move to the refinement strategy. The approach is the same as that used in the original Zienkiewicz-Zhu paper [332], where the target is the reduction of the relative error for the whole problem domain below a set value $\bar{\eta}$, i.e.

$$\eta < \bar{\eta}. \quad (4.13)$$

The goal of the adaptive analysis is to distribute the error equally into each background cell, so the permissible error in each background cell $\|e\|_k$ is written as

$$\|e\|_k = \frac{\bar{\eta}}{100} \left(\frac{\|U\|^2}{nc} \right)^{\frac{1}{2}}, \quad (4.14)$$

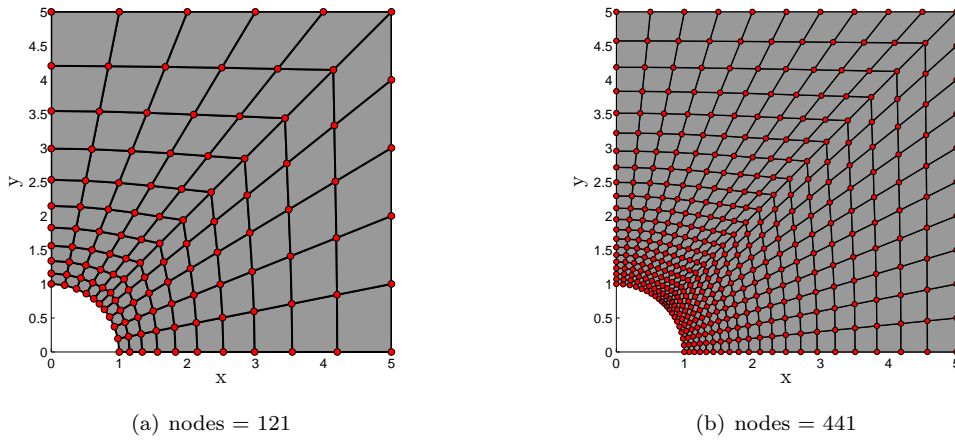


Figure 4.4: Problem discretizations with background cells and nodes for the hole in an infinite plate problem

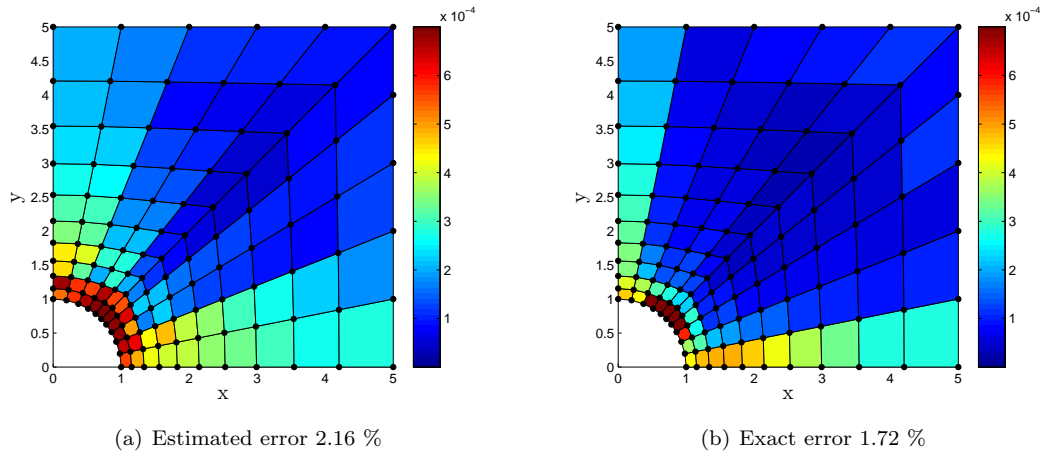


Figure 4.5: Comparison of estimated and exact cells-wise error in energy norm for 121 nodes for $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ for the hole in an infinite plate problem

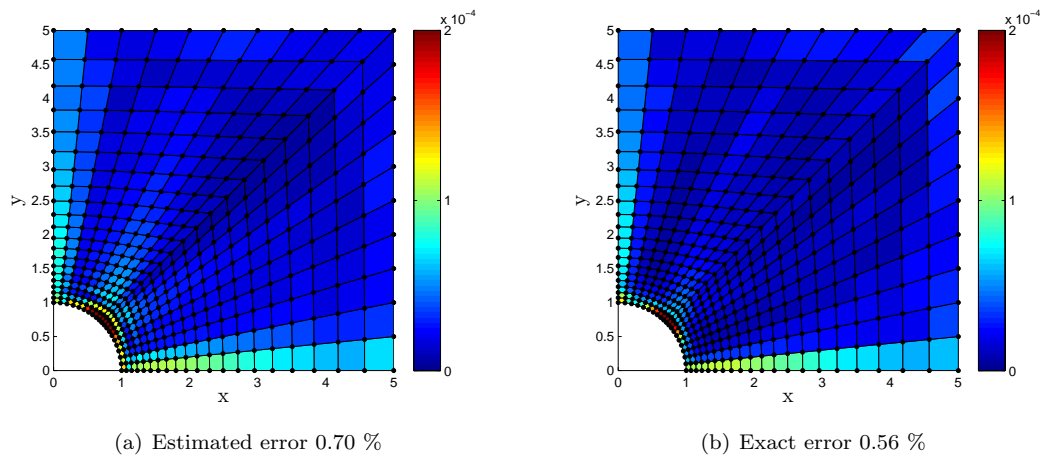


Figure 4.6: Comparison of estimated and exact cells-wise error in energy norm for 441 nodes for $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ for the hole in an infinite plate problem

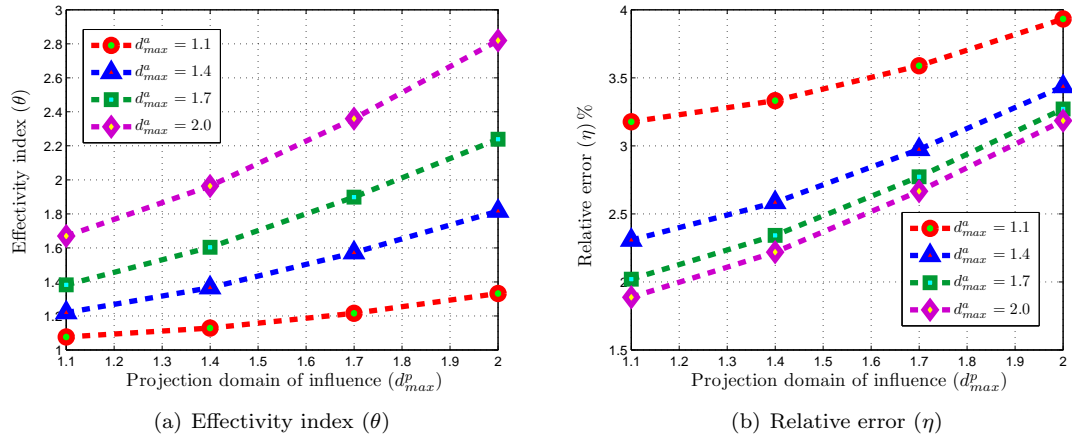


Figure 4.7: Effectivity index (θ) and relative error (η) for different d_{max}^a and d_{max}^p values for 121 nodes for the hole in an infinite plate problem

where nc is the total number of background cells. Integration cells need to be refined for which $\frac{\|e^p\|_k}{\|e\|_k} > 1$.

The step-by-step refinement of the cell is illustrated in Figure 4.8, as suggested in [256]. It is clear from the figure that in the cells which need refinement, five new nodes are inserted, one on each side and one in the middle of the cell, and then the cell is further divided into four new cells. The refinement strategy used here is selected due to the use of background mesh in our case, where nodes are attached to the background cells. Further division of the cells retains the background mesh, which can assist in the implementation of the kd-tree (k-dimensional tree) with the background mesh algorithm [29]. Although the kd-tree with background mesh is not used in this chapter, it is a way to search for nodes in the support of Gauss points, i.e. to take advantage of the computational efficiency of background mesh and kd-tree algorithm simultaneously and will be implemented in the coming chapters. Furthermore, refinement strategy used here is simple to implement and straightforward to extend to three-dimensional case.

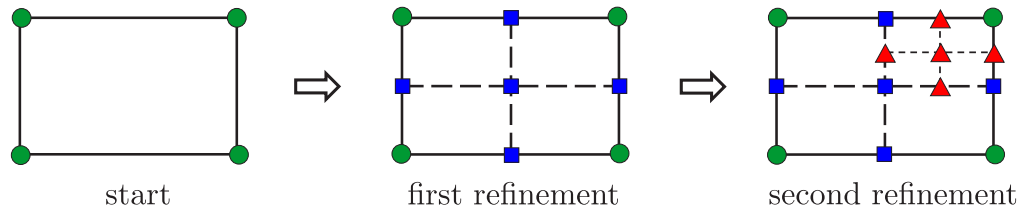


Figure 4.8: Step by step refinement of the integration cells

4.2.3 Domain of influence based on Voronoi diagram

Calculation of proper nodal influence domains for nonuniform nodal distributions, results in our case from adaptive refinement, is not a straightforward task. The nodal influence domains should be smaller in a region with fine discretization as compared to a region with coarse discretization. This size distribution of the influence domains maintains the computational efficiency as well as the local characteristics of the meshless solution without unnecessary over-smoothing in a region with fine discretization, while at the same time making sure that there are sufficient nodes in the support of Gauss points necessary for the shape function calculations in a region with coarse discretization. In this research, the method of Voronoi diagrams [180, 181] are used to calculate the nodal domain of influence. As the size of domain

of influence for node i is $d_{mi} = d_{max}c_i$, where d_{max} is a scaling parameter and c_i is maximum distance to a node in a set of surrounding nodes, which are determined in this case using the information from Voronoi diagrams, i.e.

$$c_i = \max_{j \in S_j} \|\mathbf{x}_j - \mathbf{x}_i\|. \quad (4.15)$$

S_j is a set of surrounding nodes and is given as

$$S_j = [j : V(j) \cap V(i) \neq \emptyset] \quad (4.16)$$

where $V(i)$ is the Voronoi cell of a node i . For illustrative purposes, a sample 2D grid of nodes is shown in Figure 4.9(a), its Voronoi diagram is shown in Figure 4.9(b) and the corresponding rectangular nodal domains of influence (reduced in size for clarity) are shown in Figure 4.9(c). For two-dimensional prob-

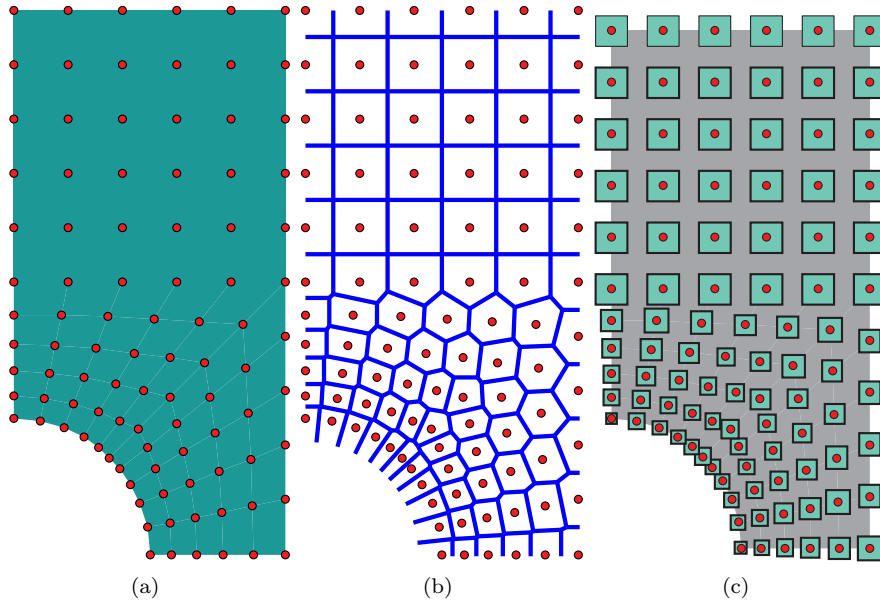


Figure 4.9: Problem discretization, Voronoi diagram and corresponding nodal domains of influence

lems, built-in MATLAB functions “voronoin” and “voronoi” are used to calculate the nodal domains of influence and plotting the Voronoi diagram respectively. The Voronoi diagram generated by MATLAB is unbounded, so proper care is needed to truncate the Voronoi diagram according to the problem boundaries.

4.2.4 Numerical examples

To demonstrate the implementation and performance of the error estimation in the adaptive analysis for linear elastic problems, numerical examples are now presented. The results from the adaptive refinement are also compared with the corresponding uniform refinement with the same number of degrees of freedom.

Cantilever beam

The first example is a two-dimensional cantilever beam subjected to parabolic traction at the free end and fully fixed at the other. The coordinate system, boundary conditions, geometry and loading for the beam are shown in Figure 4.10. The expression for the traction applied on the right-hand side of the beam is

$$p = -\frac{P}{2I} \left[\frac{D^2}{4} - y^2 \right], \quad (4.17)$$

where P is the total load. The problem is solved with $L = 48$, $D = 12$, $P = 1000$, $\nu = 0.3$, $E = 30 \times 10^6$, $d_{max}^a = 1.5$, $d_{max}^p = 1.1$ and $\bar{\eta} = 2\%$, all in compatible units. The step by step adaptive refinements are shown in Figure 4.11 with 175, 319 and 598 nodes in the first, second and third discretizations respectively. Nodes are automatically inserted in a high-stress region, i.e. near the fixed boundary. The corresponding step by step Voronoi diagrams for the adaptively refined discretizations are also shown in Figure 4.12, used for the calculation of the influence domains are shown in Figure 4.13. The influence domains are smaller in a region with fine discretization as compared to the region with coarse discretization. This size distribution of the influence domain maintains the local characteristics of the meshless solution without unnecessary over-smoothing in a region with fine discretization, while at the same time making sure that there are sufficient nodes in the support of Gauss points necessary for the shape function calculations in a region with coarse discretization. The same problem is also solved with uniformly refined discretizations

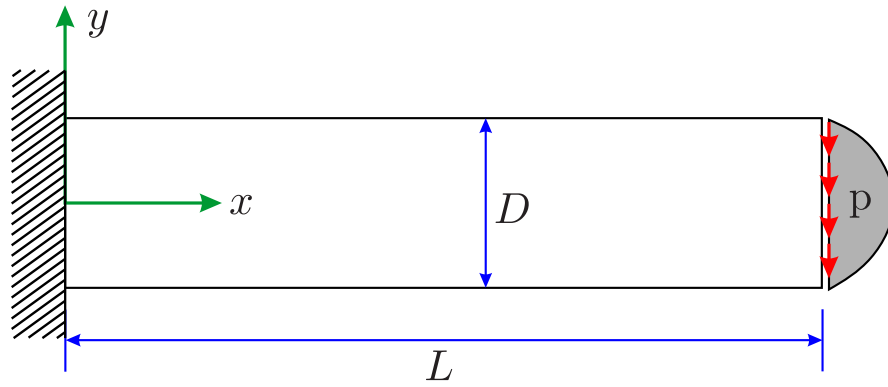


Figure 4.10: Geometry, boundary condition and loading for the cantilever beam problem

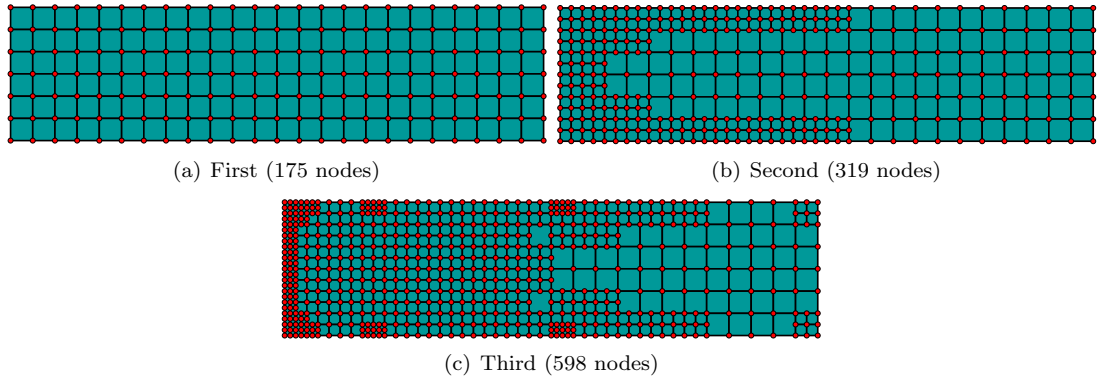


Figure 4.11: Step by step adaptive refinement for the cantilever beam problem

shown in Figure 4.14 with approximately the same number of degrees of freedom as in corresponding, adaptive discretizations. The corresponding step by step Voronoi diagrams and the domains of influence are also shown in Figures 4.15 and 4.16 respectively. The sizes of the influence domains are the same throughout the problem domain because of the uniformly distributed nodes. The comparison of the relative error (η) versus degrees of freedom for the two cases, i.e. adaptively refined and uniformly refined, is shown in Figure 4.17 where it is clear that for the same number of degrees of freedom, the adaptively refined case performs better and the decrease in relative error is greater as compared to the uniformly refined case. The reason is that, in the adaptively refined case, more nodes are concentrated in high-stress regions with high discretization error.

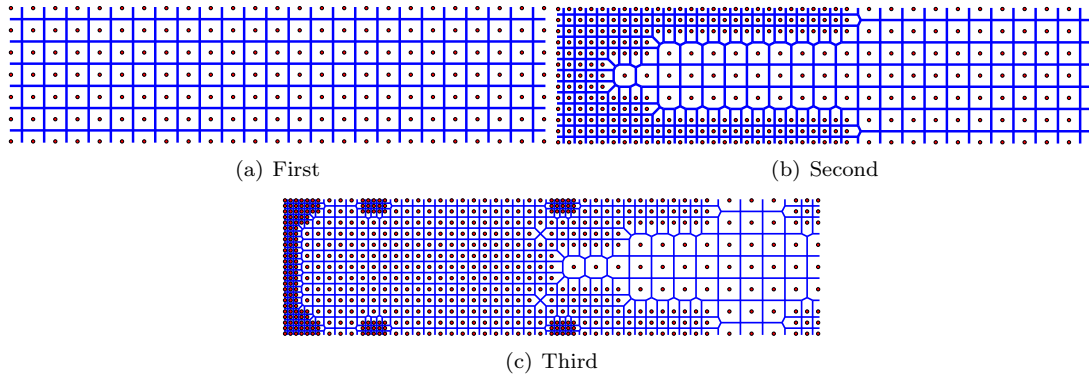


Figure 4.12: Step by step Voronoi diagram for the adaptively refined cantilever beam problem

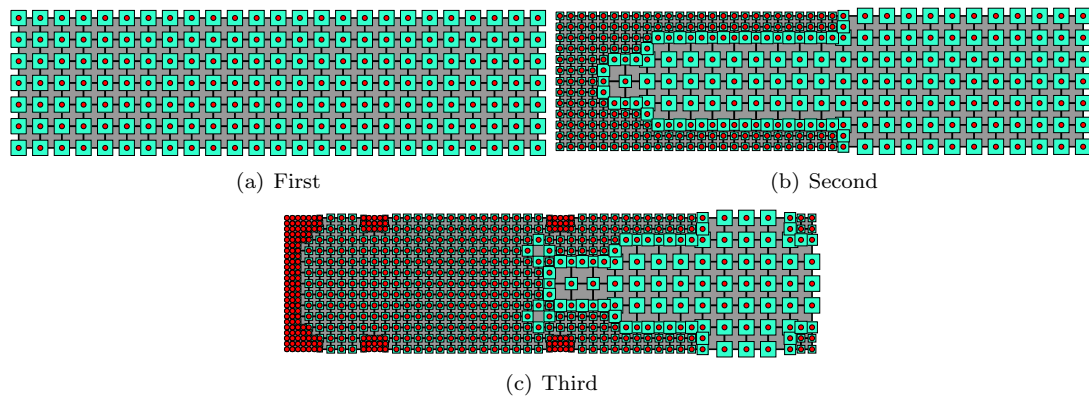


Figure 4.13: Step by step influence domains for the adaptively refined cantilever beam problem

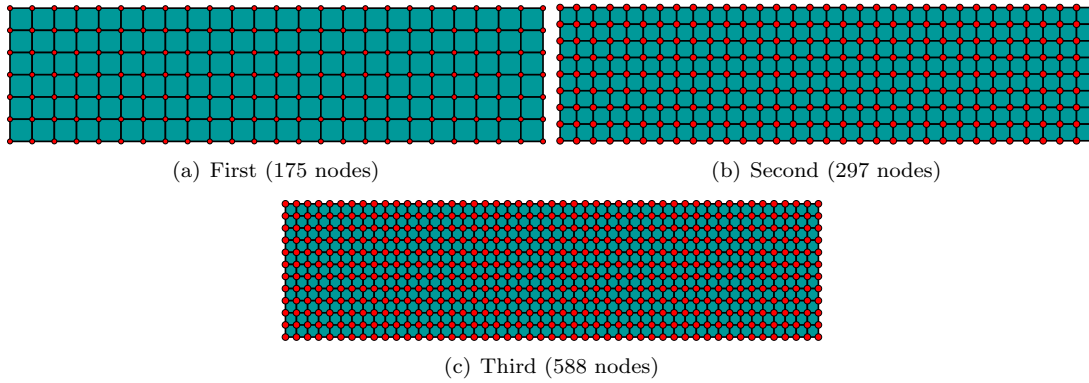


Figure 4.14: Step by step uniform refinement for the cantilever beam problem

Square block problem

The second example is a square block subjected to uniform unit traction on the left edge and fully fixed on the bottom edge. The geometry, boundary conditions and loading for this problem are shown in Figure 4.18. The problem is solved with $E = 1 \times 10^3$, $\nu = 0.3$, $d_{max}^a = 1.5$, $d_{max}^p = 1.1$ and $\bar{\eta} = 3.0\%$, all in compatible units. The step by step adaptive refinement for this problem is shown in Figure 4.19 with 121, 221 and 333 nodes in consecutive discretizations. During the adaptive analysis, nodes are automatically added to the high-stress regions. The step by step Voronoi diagrams for the three consecutive discretizations are shown in Figure 4.20, which are used here for calculation of the nodal domain of influence. The domains of influence for all the nodes in the problem domain in all the three

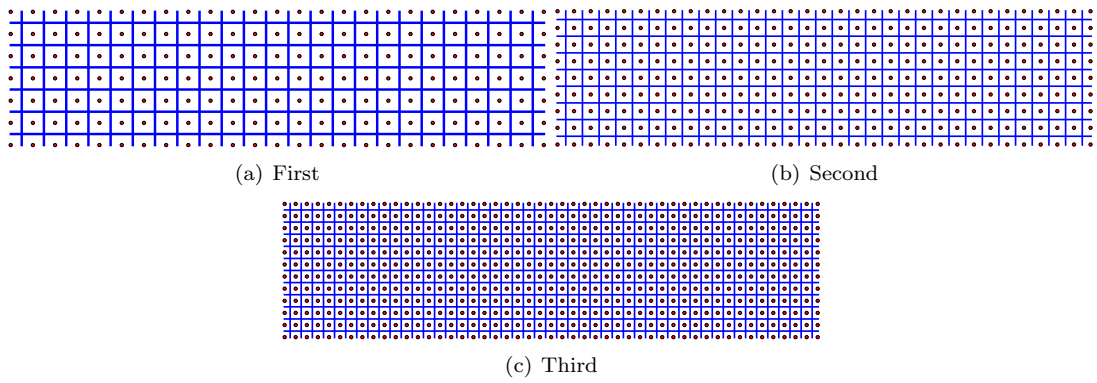


Figure 4.15: Step by step Voronoi diagram for the uniformly refined cantilever beam problem

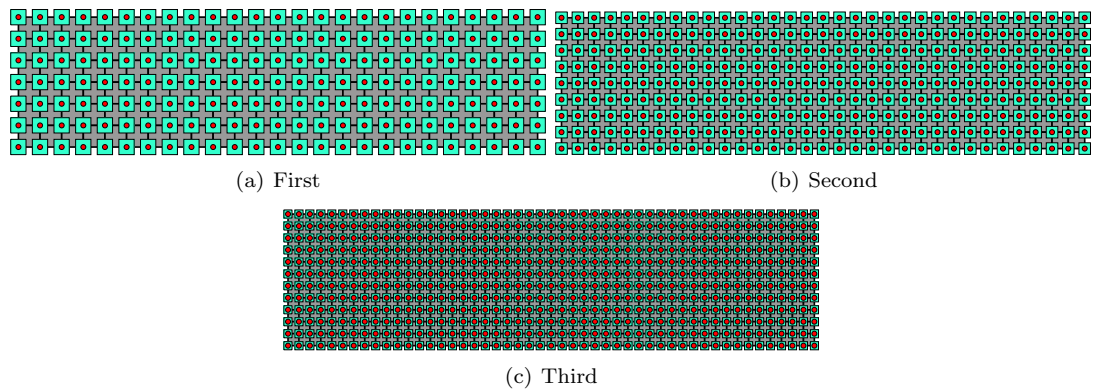
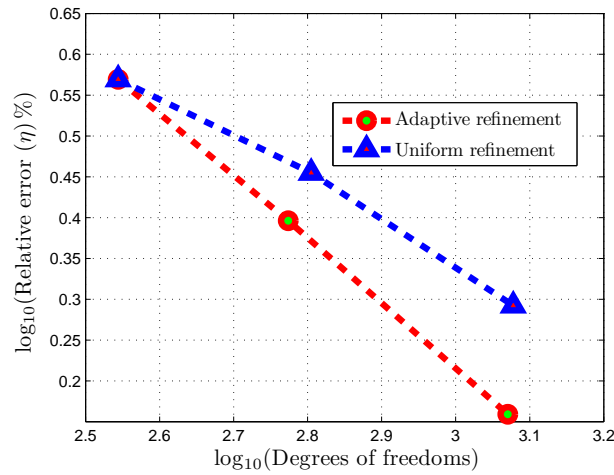


Figure 4.16: Step by step influence domains for the uniformly refined cantilever beam problem

Figure 4.17: Relative error $(\eta) \%$ versus degrees of freedom for the cantilever beam problem

discretizations are shown in Figure 4.21. It is clear from Figure 4.21, that the influence domains are adjusted automatically with reference to the nodal distribution.

The same problem is also solved for uniform discretization with approximately the same number of degrees of freedom as in Figure 4.19 and are shown in Figure 4.22 with 121, 225 and 361 in the first, second and third discretizations respectively. The relative errors (η) versus the degrees of freedom in the adaptive case and uniform case are compared in Figure 4.23. The adaptive case clearly performs better

than the uniform refinement case.

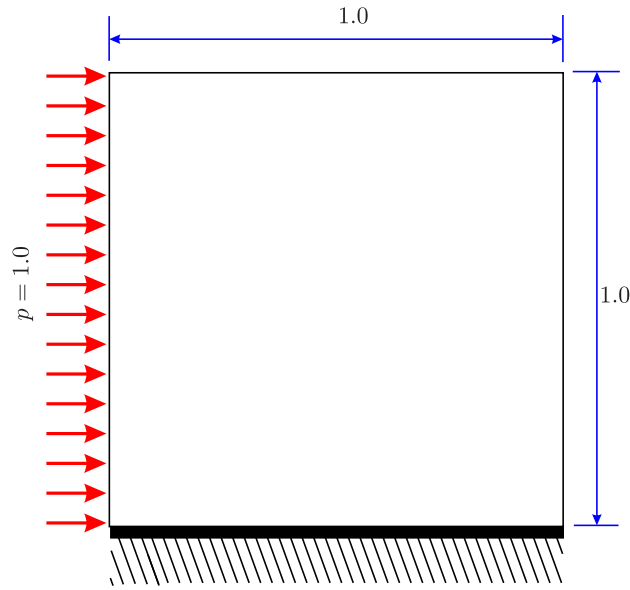


Figure 4.18: Geometry, boundary condition and loading for the square block problem

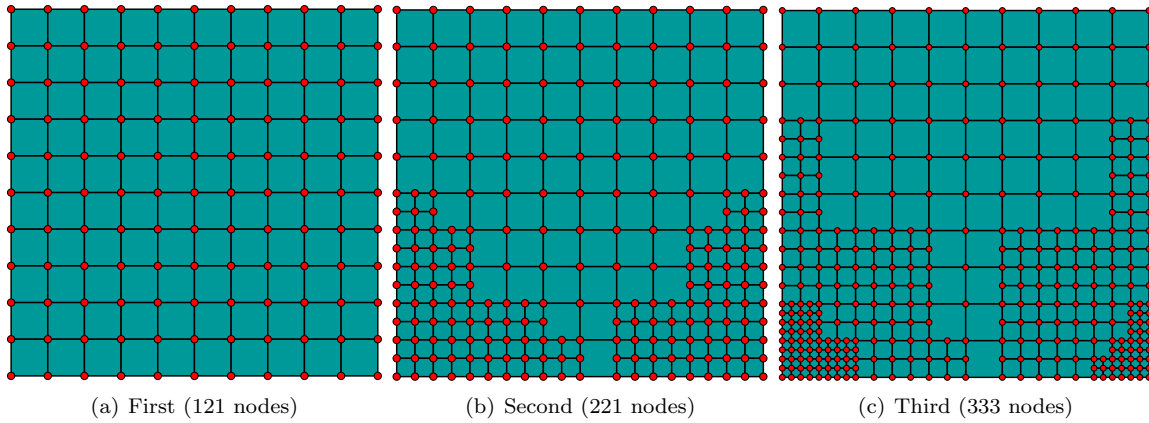


Figure 4.19: Step by step adaptive refinement for the square block problem

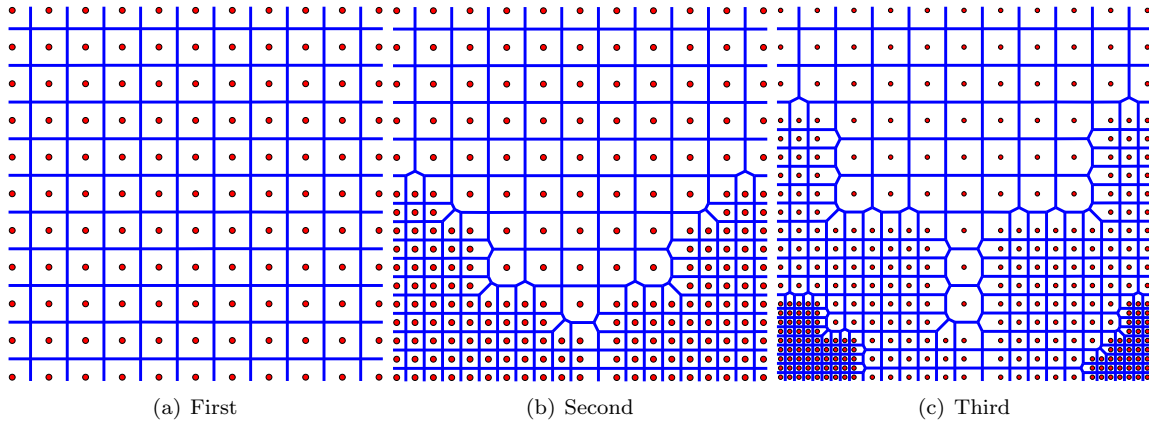


Figure 4.20: Step by step Voronoi diagram for the adaptively refined square block problem

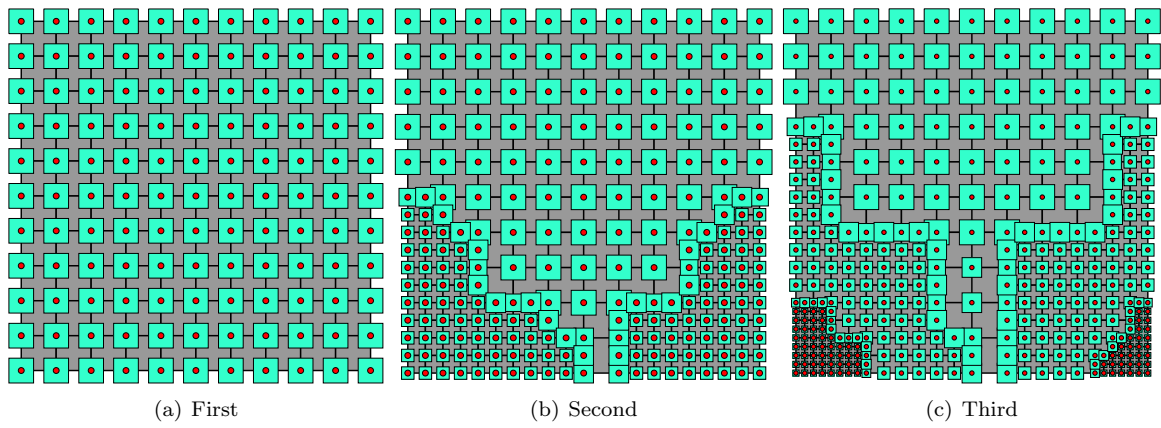


Figure 4.21: Step by step influence domains for the adaptively refined square block problem

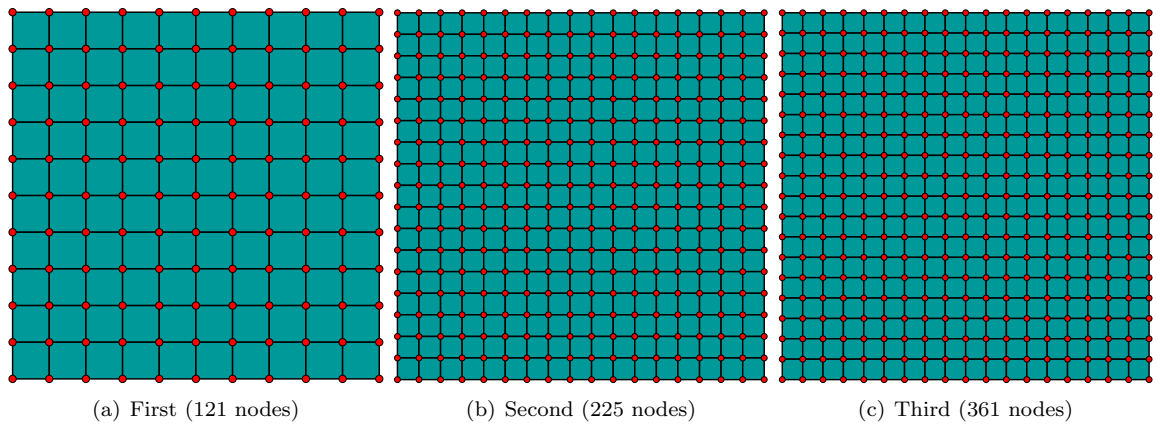
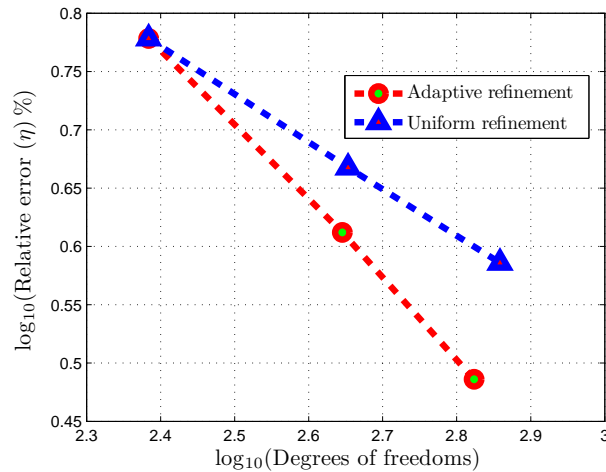


Figure 4.22: Step by step uniform refinement for the square block problem

Figure 4.23: Relative error (η) % versus degrees of freedom for the square block problem

L-shaped plate

The final example to show the performance of the linear elastic adaptive analysis is the L-shaped plate problem, the geometry, boundary conditions and loading for which are shown in Figure 4.24. Here point A in Figure 4.24 is a point of high stress concentration due to the geometry of a re-entrant corner. This

problem is solved with $E = 1 \times 10^3$, $\nu = 0.25$, $d_{max}^a = 1.5$, $d_{max}^p = 1.1$ and $\bar{\eta} = 5.0\%$, all in compatible units. The step by step adaptive refinements for this problem are shown in Figure 4.25 with 225, 281, 343 and 421 nodes in first, second, third and fourth discretizations. As expected the adaptive algorithm automatically adds nodes in the vicinity of point A, which is a point of high stress concentration and needs more nodes to accurately capture the solution. The step by step Voronoi diagrams and the corresponding influence domains for the adaptive discretization are shown in Figures 4.26 and 4.27 respectively. The automatic adjustment of the size of the influence domain with reference to the nodal distribution is also clear.

The same problem is also solved with uniform discretizations with the approximately the same number of degrees of freedom as shown in Figure 4.28 with 225, 280, 341 and 408 nodes in the first, second and third discretizations. A comparison between the relative error (η) versus the degrees of freedom for the adaptive and uniform refinements is shown in Figure 4.29, in which the adaptive case once again performs better than the corresponding uniform discretization case for the same number of degrees of freedom.

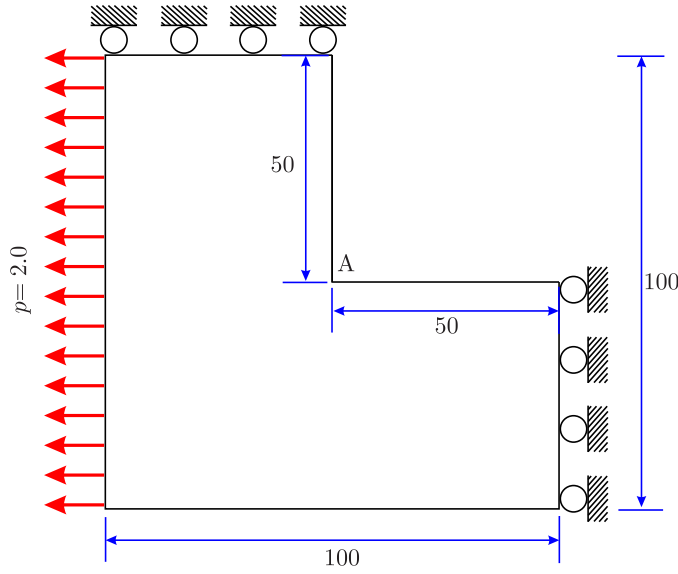


Figure 4.24: Geometry, boundary condition and loading for the L-shaped plate problem

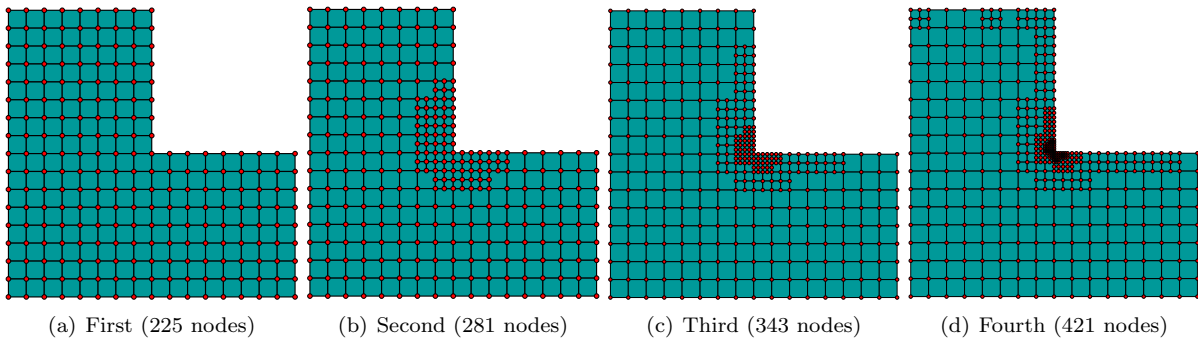


Figure 4.25: Step by step adaptive refinement for the L-shaped plate problem

4.3 Adaptivity in nonlinear problems

Conceptually adaptive procedures for nonlinear problems are the same as for linear problems but with additional challenges. In nonlinear problems, an incremental error is estimated and compared with a

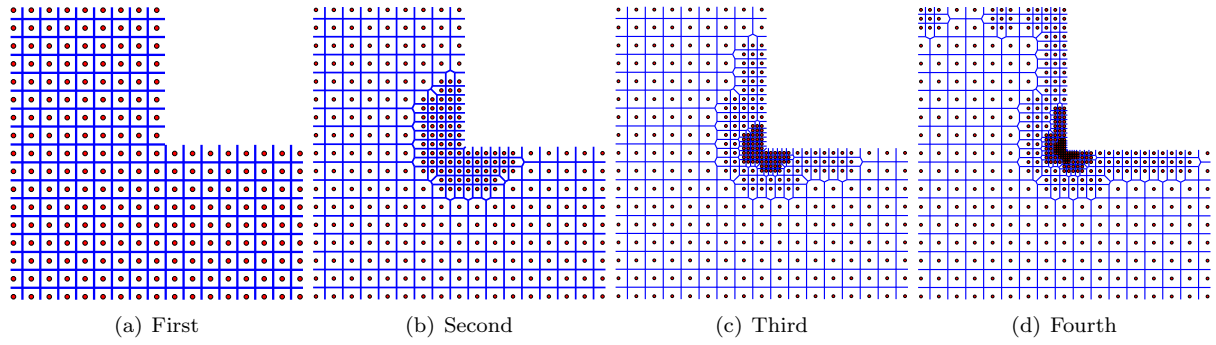


Figure 4.26: Step by step Voronoi diagram for the adaptively refined L-shaped plate problem

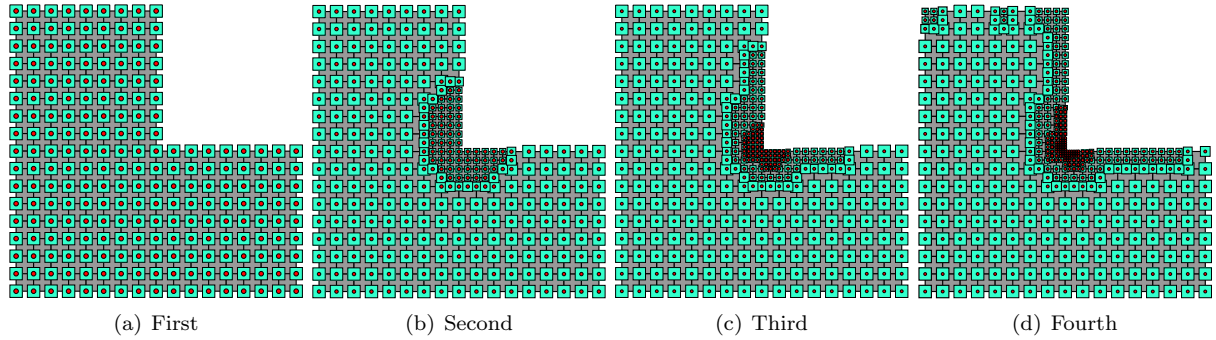


Figure 4.27: Step by step influence domains for the adaptively refined L-shaped plate problem

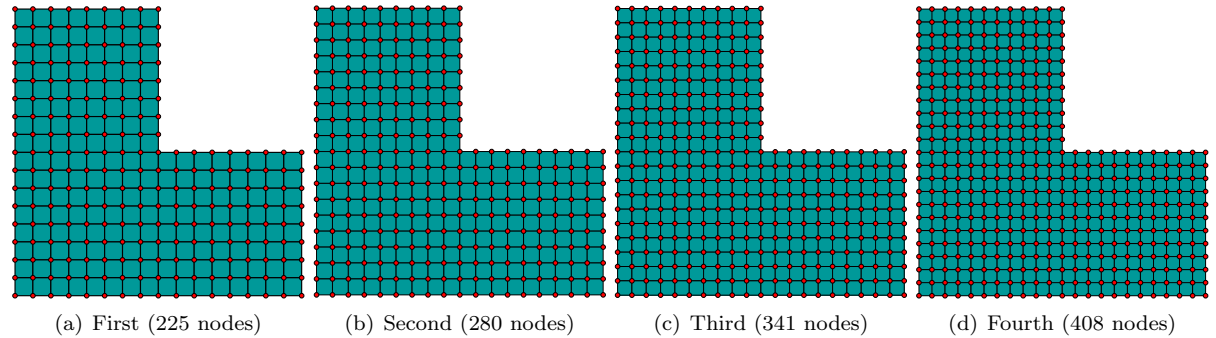


Figure 4.28: Step by step uniform refinement for the L-shaped plate problem

predefined value at the end of each solution step. Refinement is performed if required and path dependent variables are projected from the old discretization to the new. Figure 4.30 shows a flowchart of the overall adaptive algorithm used in this study. Components required for the nonlinear adaptive algorithm are the same as for linear problems with the addition of data transfer between consecutive discretizations. Each component is now discussed for nonlinear problems with reference to the meshless modelling used here in turn below.

4.3.1 Error estimation

For finite element modelling the Zienkiewicz-Zhu error estimator (based on the SPR method), has been widely used for materially and geometrically nonlinear FE problems. An example can be found in [250] for industrial metal forming problems with elasto-plasticity and finite strains, with estimators based on energy norms, plastic dissipation and rate of plastic work. In [163] the SPR method is used in large deformation problems with hyperelasticity as a material model. An adaptive strategy with error estimation based

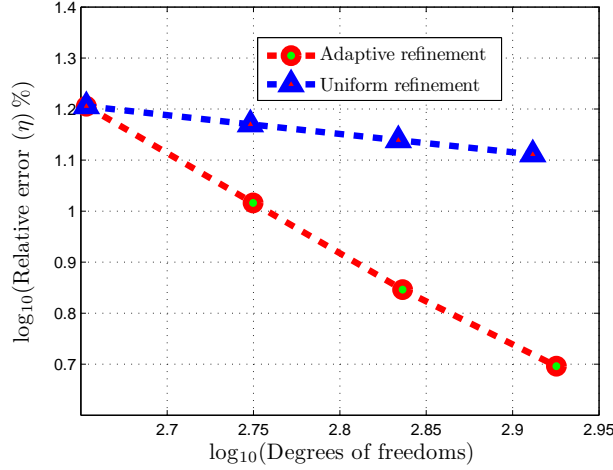


Figure 4.29: Relative error (η) % versus degrees of freedom for the L-shaped plate problem

on an L_2 norm of strain is used in 3D analysis in [316] with large deformation to model liquefaction phenomena. The Zienkiewicz-Zhu error estimator is used in the case of viscoplasticity problems, with application to metal forming processes in [169]. In this case the deviatoric stress is used to calculate the error in energy norm due to the incompressible nature of the materials. The work in [169] is further extended in [46] for 3D complex forging simulations and in [134] the SPR method with error based on an L_2 norm in strain is used in analyses involving non-homogeneous soil with large deformation. Here we extend the use of the recovery type error estimation for the EFGM proposed in [62] and as described in §4.2.1, to nonlinear problems modelled with the EFGM, including finite deformation and elasto-plasticity. The error in energy norm is used, as first proposed in [332] as a practical error estimator for engineering analysis for linear elasto-static problems. This also follows the approach in [250] also mentioned above. For the nonlinear case, the exact incremental error in energy norm for solution step n for the problem domain Ω is equal to [45]

$$\|e\| = \left[\int_{\Omega} \left| (\boldsymbol{\tau}_n(\mathbf{x}) - \boldsymbol{\tau}_n^h(\mathbf{x}))^T (\Delta \boldsymbol{\varepsilon}_n(\mathbf{x}) - \Delta \boldsymbol{\varepsilon}_n^h(\mathbf{x})) \right| d\Omega \right]^{\frac{1}{2}}, \quad (4.18)$$

where $\boldsymbol{\tau}_n(\mathbf{x})$ and $\boldsymbol{\tau}_n^h(\mathbf{x})$ are the exact and numerical Kirchhoff stresses respectively at point \mathbf{x} for solution step n , while $\Delta \boldsymbol{\varepsilon}_n(\mathbf{x})$ and $\Delta \boldsymbol{\varepsilon}_n^h(\mathbf{x})$ are the exact and the numerical incremental logarithmic strains at a point \mathbf{x} for the solution step n , i.e.

$$\Delta \boldsymbol{\varepsilon}_n(\mathbf{x}) = \boldsymbol{\varepsilon}_n(\mathbf{x}) - \boldsymbol{\varepsilon}_{n-1}(\mathbf{x}), \quad \Delta \boldsymbol{\varepsilon}_n^h(\mathbf{x}) = \boldsymbol{\varepsilon}_n^h(\mathbf{x}) - \boldsymbol{\varepsilon}_{n-1}^h(\mathbf{x}). \quad (4.19)$$

For practical engineering problems, exact stresses and strain increments are not available, therefore, as in other recovery methods, projected Kirchhoff stresses $\boldsymbol{\tau}_n^p(\mathbf{x})$ and projected incremental logarithmic strains $\Delta \boldsymbol{\varepsilon}_n^p(\mathbf{x})$ are used to obtain an error estimator $\|e^p\|$, i.e. replacing the exact quantities in Equation (4.18). In the case of the EFGM the projected Kirchhoff stresses and the projected logarithmic strain increments for the solution step n at point \mathbf{x} are determined from the following equation

$$\boldsymbol{\tau}_n^p(\mathbf{x}) = \sum_{i=1}^{np} \psi_i(\mathbf{x}) \boldsymbol{\tau}_n^h(\mathbf{x}_i), \quad \Delta \boldsymbol{\varepsilon}_n^p(\mathbf{x}) = \sum_{i=1}^{np} \psi_i(\mathbf{x}) \Delta \boldsymbol{\varepsilon}_n^h(\mathbf{x}_i), \quad (4.20)$$

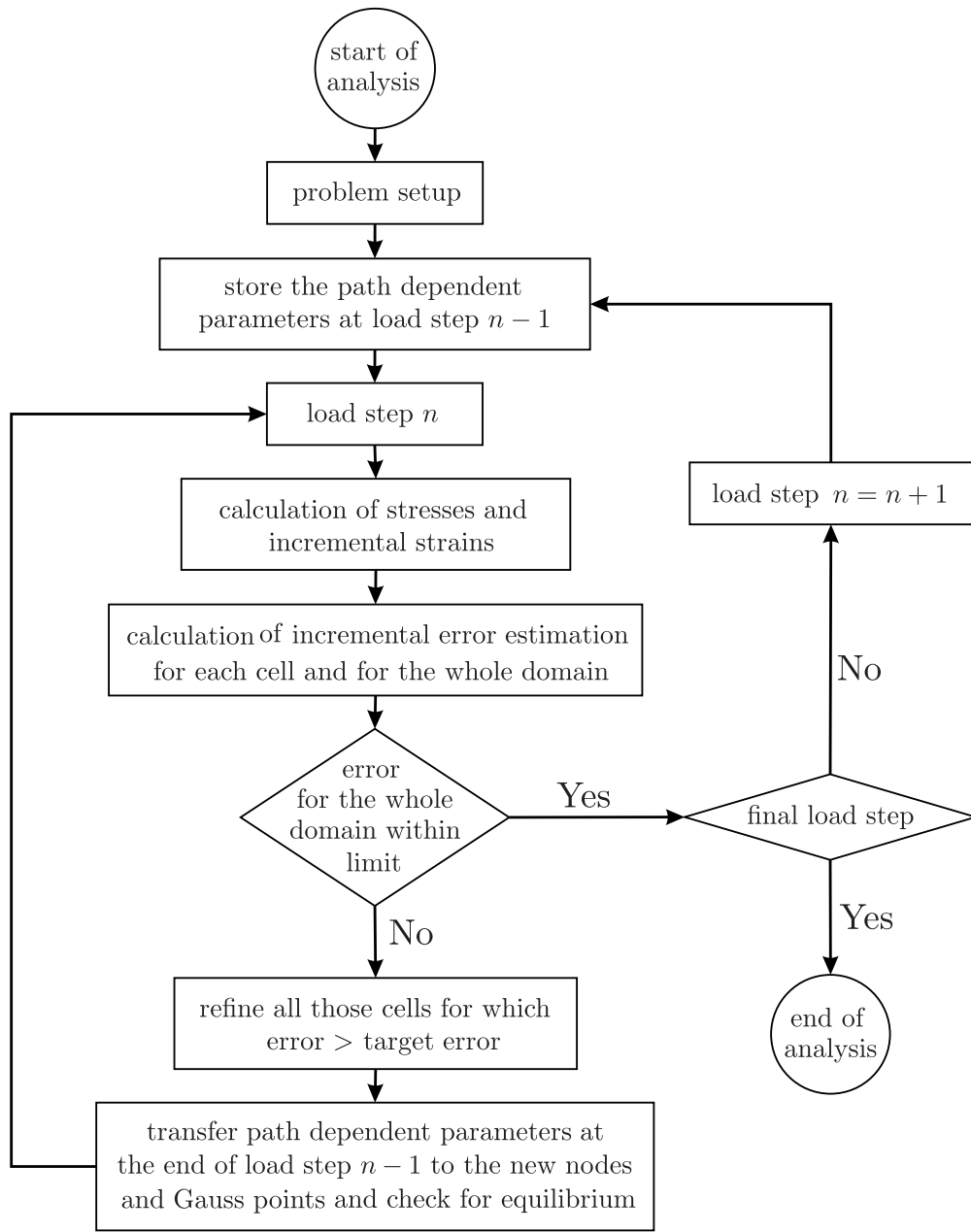


Figure 4.30: Nonlinear adaptive algorithm

where $\psi_i(\mathbf{x})$ is the shape function of a node i evaluated at a point \mathbf{x} with reduced nodal support as compared to the support used to obtain the original approximate solution. The approach is therefore, similar to that originally proposed for meshless linear elasto-statics in [62] in obtaining projected values but uses the incremental error in energy norm as the error measure, in addition to the use of finite deformation stress and strain measures. The incremental global percentage error for the whole domain is written as

$$\eta = \frac{\|e^p\|}{\|U\|} \times 100, \quad (4.21)$$

where $\|U\|$ is the incremental energy norm for solution step n

$$\|U\| = \left[\int_{\Omega} \left| (\boldsymbol{\tau}_n^p(\mathbf{x}))^T (\Delta \boldsymbol{\varepsilon}_n^p(\mathbf{x})) \right| d\Omega \right]^{\frac{1}{2}}. \quad (4.22)$$

Equations (4.18) and (4.22) can also be integrated over the each individual integration cell Ω_k instead of the whole problem domain Ω to calculate the incremental error in energy norm and incremental energy norm for each cell. For the whole domain Ω we can write squares of the error and energy norms as

$$\|e^p\|^2 = \sum_{k=1}^{nc} \|e^p\|_k^2, \quad \|U\|^2 = \sum_{k=1}^{nc} \|U\|_k^2, \quad (4.23)$$

where $\|e^p\|_k$ is the incremental error estimate in energy norm for background cell k and $\|U\|_k$ is the incremental energy norm for background cell k .

4.3.2 Refinement strategy & data transfer

The refinement strategy is the same as that used for linear elastic problems in Equations (4.13) and (4.14) but with corresponding, incremental values. The step by step refinement of the background cells in this case is also the same as shown in Figure 4.8. A major task in the nonlinear adaptive analysis is an accurate transfer of path dependent variables between the old and new discretizations. In this chapter, the moving least squares (MLS) approximation is used to transfer data from the old nodes and Gauss points to their new counterparts, a strategy suggested in [256]. The reasons for the choice of MLS rather than max-ent are that the Newton process for determining the Lagrange multipliers for the max-ent formulation becomes poorly convergent when mapping points are close to a boundary and secondly max-ent cannot deal with a new node position which falls outside the original domain, as might commonly occur with finite deformation. The path dependent parameters for the nodes are displacement \mathbf{u} , elastic strain and plastic strain. For the Gauss points the variables are the same except for the omission of displacement. Writing any of these path dependent variables as ξ , transfer occurs using

$$\xi(\mathbf{x}_{new}) = \sum_{i=1}^{n_{old}} \psi_i(\mathbf{x}_{new}) \xi(\mathbf{x}_{old})_i, \quad (4.24)$$

where \mathbf{x}_{new} is the position of the new node or Gauss point, \mathbf{x}_{old} is the position of old nodes or Gauss points, n_{old} is the number of old nodes or Gauss points in the support of \mathbf{x}_{new} and $\psi_j(\mathbf{x}_{new})$ are MLS shape functions.

4.3.3 Numerical examples

A number of numerical examples are now presented to demonstrate the correct implementation and performance of the full nonlinear meshless modelling approach whose components have been described above. All are taken from or are closely related to problems presented in other publications using finite element analysis. We begin with a simple small deformation problem in 2D to demonstrate the robustness of the basic code, before moving on to examples, which demonstrate the full features of the modelling. In all examples, cubic spline weight functions are used for the calculation of the MLS and max-ent shape function and the Prandtl-Reuss constitutive model is used to capture the elasto-plastic response.

Strip footing collapse

The same problem is also analyzed in [81] to determine a limiting load with finite element analysis. The geometry and boundary conditions of the problem are given in Figure 4.31. The analytical solution for the limiting load for the same problems is also given in [81] as

$$P_{lim} = (2 + \pi) c \approx 5.14c \approx 2.97\sigma_y. \quad (4.25)$$

where c is the cohesion or shear strength (and for von Mises plasticity $c = \sigma_y/\sqrt{3}$). Only one-half of the model shown in grey in Figure 4.31 is analyzed due to symmetry with plane strain conditions and displacement control with $E = 1 \times 10^7$ kPa, $\nu = 0.48$ and $\sigma_y = 848.7$ kPa. In this case 0.002 m of vertical displacement is applied to the footing in 10 equal steps, which is clearly a small strain problem. The scaling parameters for the domain of influence for analysis and projection used in this case are $d_{max}^a = 1.5$, $d_{max}^p = 1.1$ respectively. This problem is solved with adaptive analysis with a target error of 20% and the evaluation of adaptive discretizations shown in Figure 4.32 with the number of nodes in the consecutive discretizations being 121, 191, 305 and 543 respectively. The Voronoi diagrams for the different discretizations in Figure 4.32 are shown in Figure 4.33, which are used for the calculation of the nodal influence domains given in Figure 4.34. Contour plots of the effective plastic strain are shown in Figure 4.35(a). For comparison the same problem is also solved with an initial discretization without adaptivity and the contours of effective plastic strain are shown in Figure 4.35(b), which is clearly a very rigid response as compared to the case of the adaptive analysis. The normalized pressure vs displacement curves are given in Figure 4.36. For comparison the same problem is also solved with different initial adaptive discretizations shown in Figure 4.32 without adaptivity and the normalized pressure vs displacement curves are also given in Figure 4.36. The convergence of the solution for the adaptive analysis as well as for different adaptive initial discretization without adaptivity to the analytical solution is clear from the plot. The solution of this problem with small deformations and matching to a known analytical solution gives confidence to extend the proposed method to problems involving large strains. It is obvious from the plot that the adaptive solution as shown has a number of jumps in the curve. The jumps represent points where rediscrretization is taking place and mapping has been carried out. The jumps are due to changes in the equilibrium state of the domain due to the altered discretization. They should not be confused with jumps in displacements. These plots show results from successive analysis not a single calculation.

Perforated tensile specimen

The second problems which begins to challenge the modelling, is a perforated thin rectangular plate subject to uniaxial tensile stress applied by prescribed displacements of one set of edges as used in [45]. The same geometry is used here and is shown in Figure 4.37. One-quarter of the plate, shown in grey in Figure 4.37, is analyzed due to symmetry, using plane strain conditions with displacement control. The material properties used in this case are $E = 1 \times 10^5$, $\nu = 0.3$ and $\sigma_y = 1 \times 10^3$. The scaling parameters for the domain of influence for analysis and projection used in this case are $d_{max}^a = 1.5$, $d_{max}^p = 1.2$ respectively. A total vertical displacement of 0.3 units is applied to the upper horizontal edge of the plate over 50 equal steps, clearly making this a finite deformation problem given the original dimensions of the plate. This example is solved using the adaptive procedure with a target error of 20%. During analysis, two adaptive refinements are performed. The initial nodal discretization is shown in Figure 4.38(b), while the first and second refinements (at increasing displacements) are given in Figures 4.38(c) and 4.38(d) respectively. The final deformed configuration (at the final displacement of 0.3 units) is also shown, in Figure 4.38(e). The numbers of nodes in the consecutive discretizations are 65, 121 and 229. The

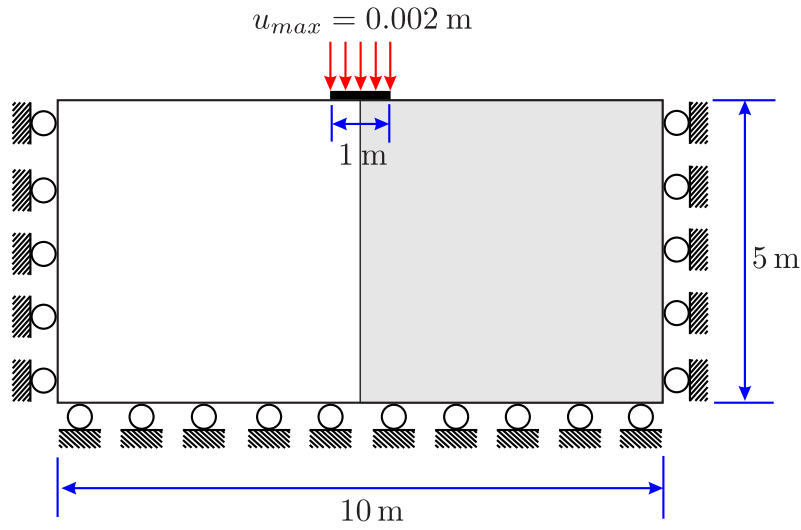


Figure 4.31: Geometry, boundary condition and loading for the strip footing problem

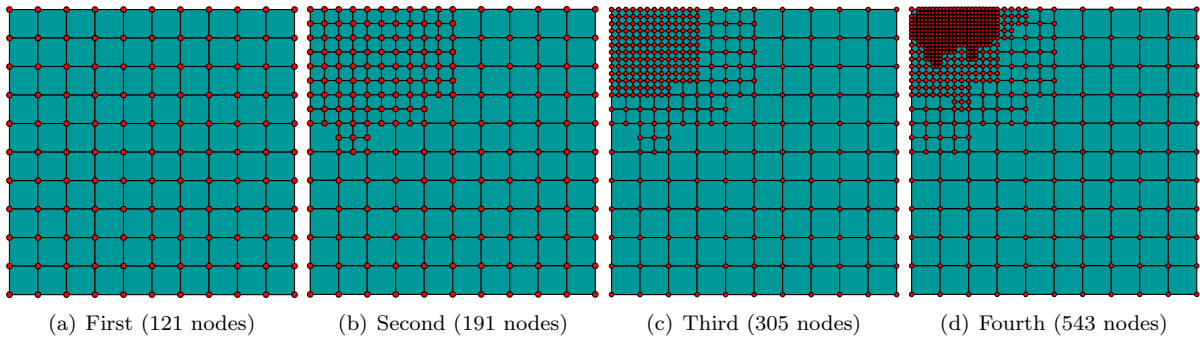


Figure 4.32: Step by step discretizations for the strip footing problem

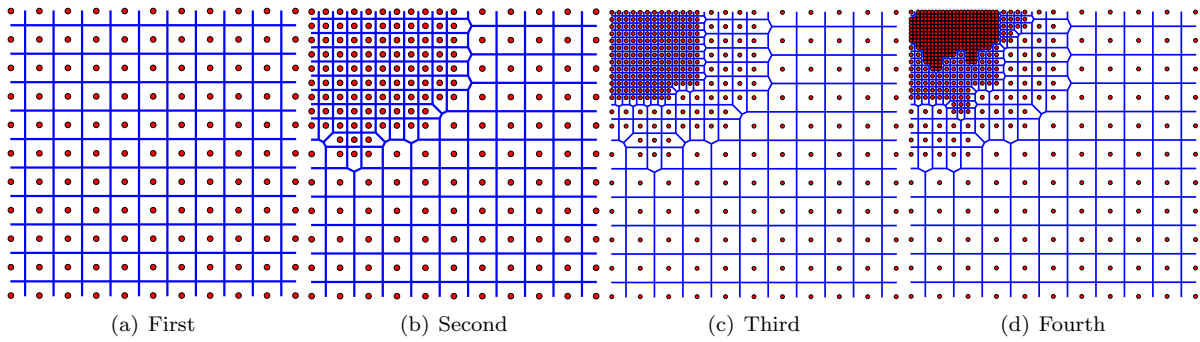


Figure 4.33: Step by step Voronoi diagrams for the strip footing problem

refinement procedure subdivides integration cells placing new nodes to define new smaller cells, leading to a roughly structured pattern of refinement. This is clearly unlikely to be the optimal refinement for the load increment as a more unstructured arrangement would be expected to be optimal. However, the simplicity of the refinement algorithm provides a competing advantage.

Contours of effective plastic strain over the deformed configuration at the end of the adaptive process are shown in Figure 4.39(a), in which a developing shear band of finite thickness is clearly evident in the thinning section of the plate adjacent to the hole. The same problem is also solved with an initial coarse discretization without refinement and the contour plot of effective plastic strain at the end of the analysis

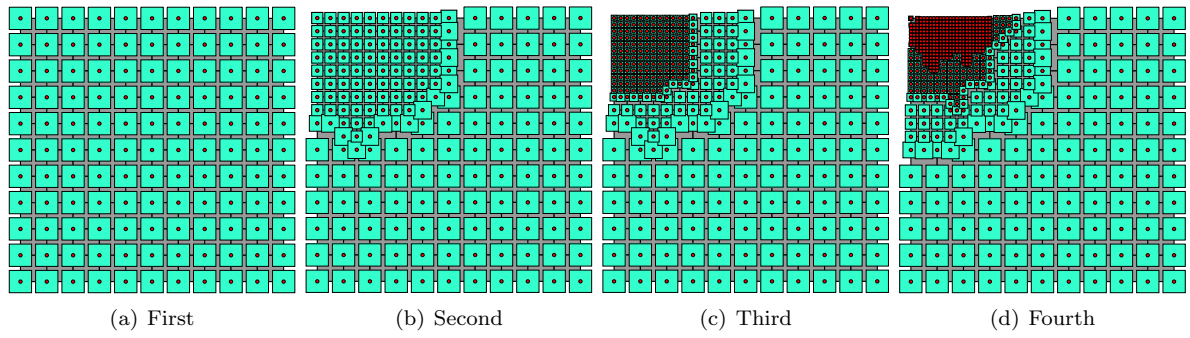


Figure 4.34: Step by step influence domains for the strip footing problem

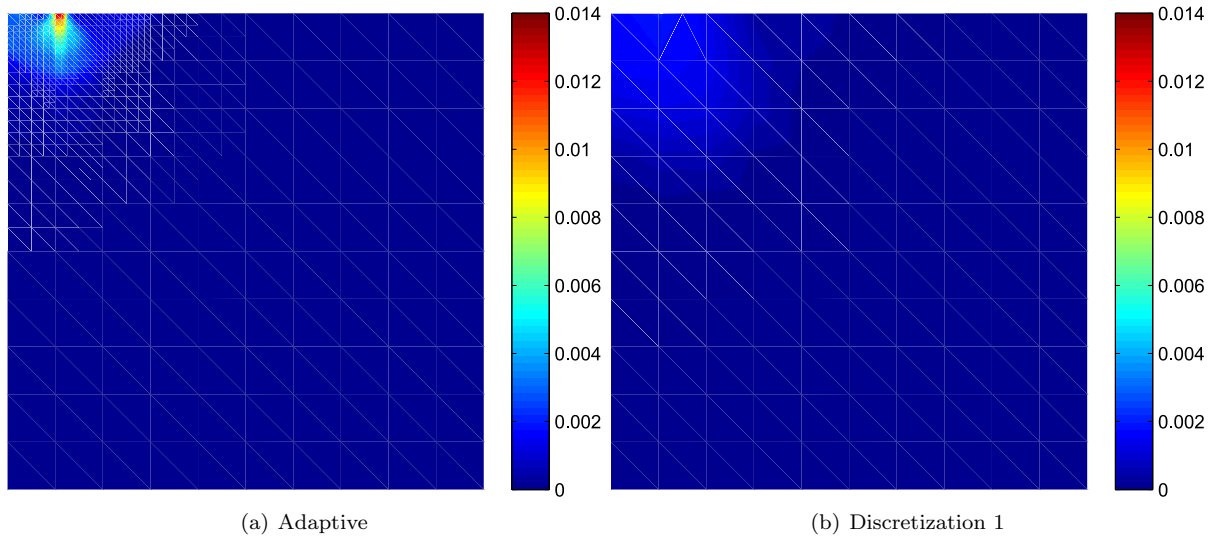


Figure 4.35: Effective plastic strain contours for the strip footing problem

is shown in Figure 4.39(b). Comparing these two figures the sensitivity of the level of refinement of the discretization when conducting finite deformation analysis is clear. Reaction/displacement plots are given in Figure 4.40, the reaction being the integrated nodal loads along the displaced edge and the displacement the prescribed displacement. The same problem is also solved with different initial adaptive discretizations shown in Figures 4.38(b), 4.38(c) and 4.38(d) as well as with very fine or reference discretization shown in Figure 4.38(a) without adaptivity for the whole problem domain. The convergence of the curve for the adaptive analysis as well as for different adaptive initial discretization without adaptivity to the reference solution is clear from the plot. It is also clear from Figure 4.40 that the adaptive analysis captures the geometric softening behavior much better than the coarse discretization. The steps in the adaptive analysis curve occur at points of refinement and are not physically meaningful, although they provide a useful reminder of the adaptive process and follow the presentation in [205]. The effect of re-discretization at refinement is to change the net equilibrium reaction for the given displacement (as one would expect between different refinements in FEA, for instance).

Footing loaded on a slope

The third example is footing on a block of elasto-plastic material resembling a natural slope. This problem also appears in a number of references, e.g. [45, 158, 162, 205, 249] and the modelling of problems, which are dominated by shear bands are covered using meshless methods in [254, 255, 258]. The geometry, boundary condition and loading for this problem are taken from [45] and are shown in Figure 4.41. The

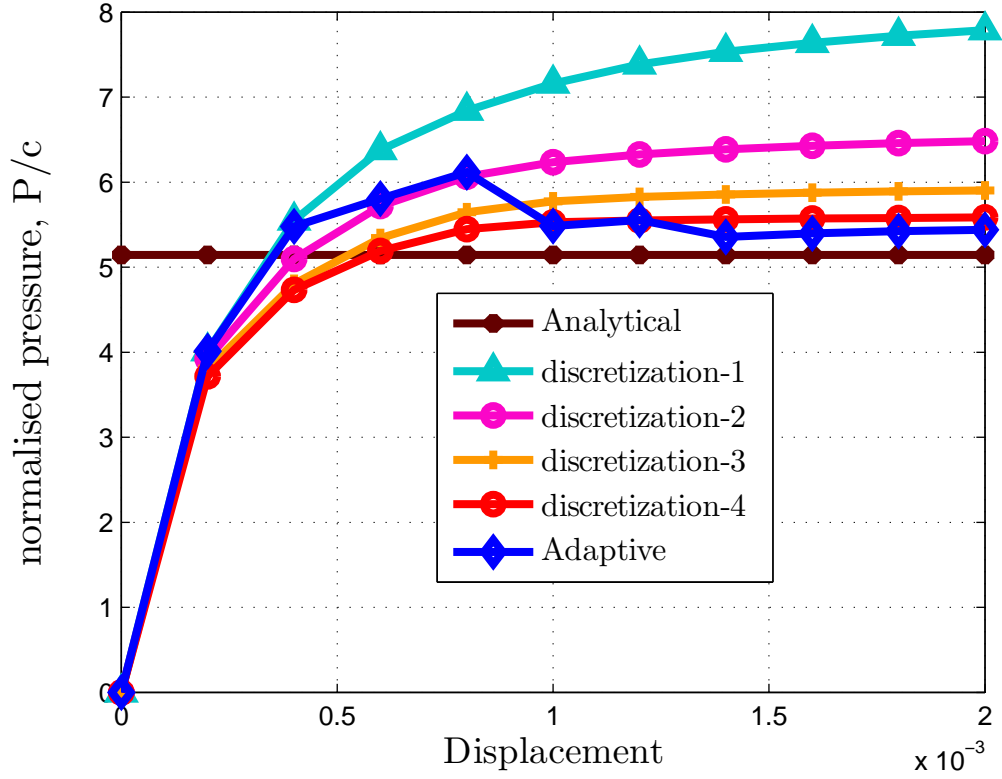


Figure 4.36: Normalized pressure vs displacement for different discretizations for the strip footing problem

material properties used in this case are $E = 2.0 \times 10^4$, $\nu = 0.3$ and $\sigma_y = 30$, while the scaling parameters for the domain of influence for analysis and projection used in this case are $d_{max}^a = 1.5$, $d_{max}^p = 1.25$ respectively. The problem is solved using displacement control applied to a single point offset from the centre of the footing, under plane strain conditions. A rough and rigid footing is used in this case with a Young's modulus of 100 times that of the slope material. A total displacement of 3.0 units is applied over 50 equal steps. Two adaptive refinements are used in this case with a target error of 15%.

The initial discretization is shown in Figure 4.42(a), while the first and second refinements are given in Figure 4.42(b), and 4.42(c) with 127, 282 and 570 nodes respectively. The final deformed configuration for this problem is shown in Figure 4.42(d). The step by step adaptive procedure concentrates nodes along the shear band, which in this case is curved, as would be predicted by simple limit equilibrium theory. The contour plots of the effective plastic strain over the deformed configuration at the end of adaptive analysis is shown in Figure 4.43(a), which clearly show a gradually extending shear band of finite thickness. Once again, the same problem has been solved with an initial coarse discretization without refinement and the contours of effective plastic strain over the deformed configuration at the end of analysis are shown in Figure 4.43(b). The convergence of the load displacement curve to the clear limit load for the adaptive analysis as well as for different adaptive initial discretization shown in Figures 4.42(a), 4.42(b) and 4.42(c) without adaptivity is clear from the Figure 4.44. For the coarse analysis however a small hardening response is evident (for the whole structure, recalling that the elasto-plastic model at the material point is itself only perfectly plastic). For the adaptive case though the response is closer to perfectly plastic.

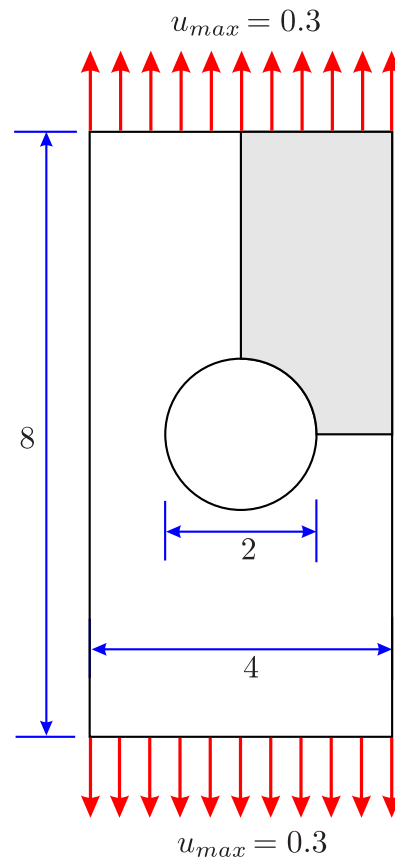


Figure 4.37: Geometry, boundary condition and loading for the perforated tensile specimen problem

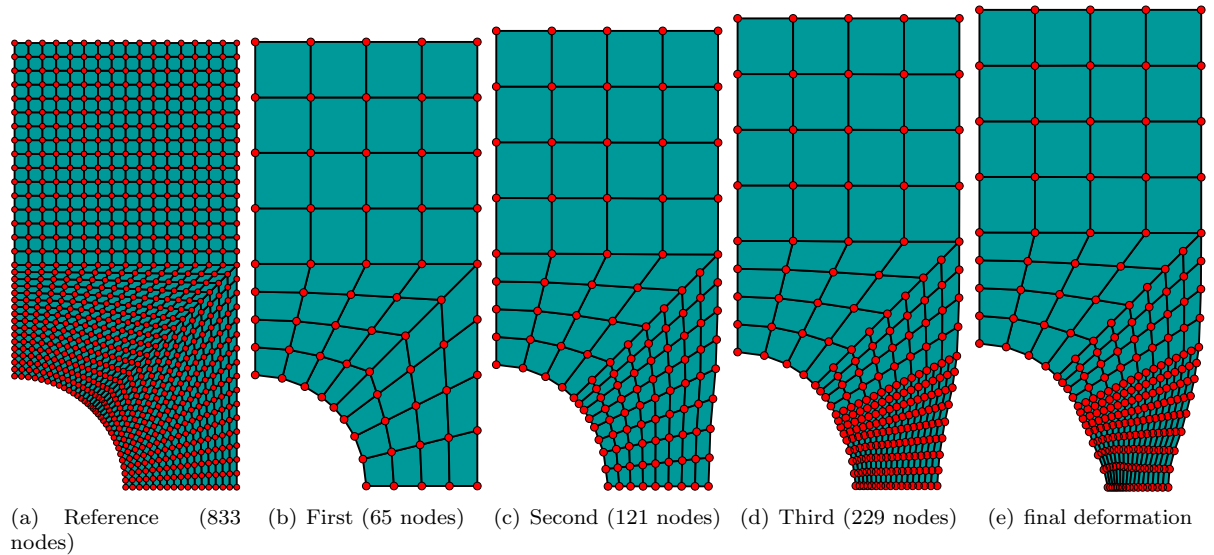


Figure 4.38: Step by step discretizations for the perforated tensile specimen problem

Footing loaded on a vertical cut

The fourth example is again a footing, but this time placed adjacent to vertical unsupported face on a block of elasto-plastic material. The problem matches that found in a number of references, such as

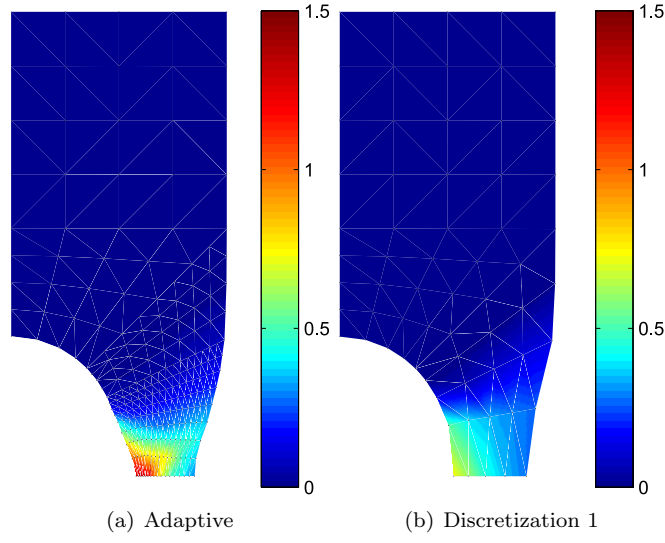


Figure 4.39: Effective plastic strain contours for the perforated tensile specimen problem

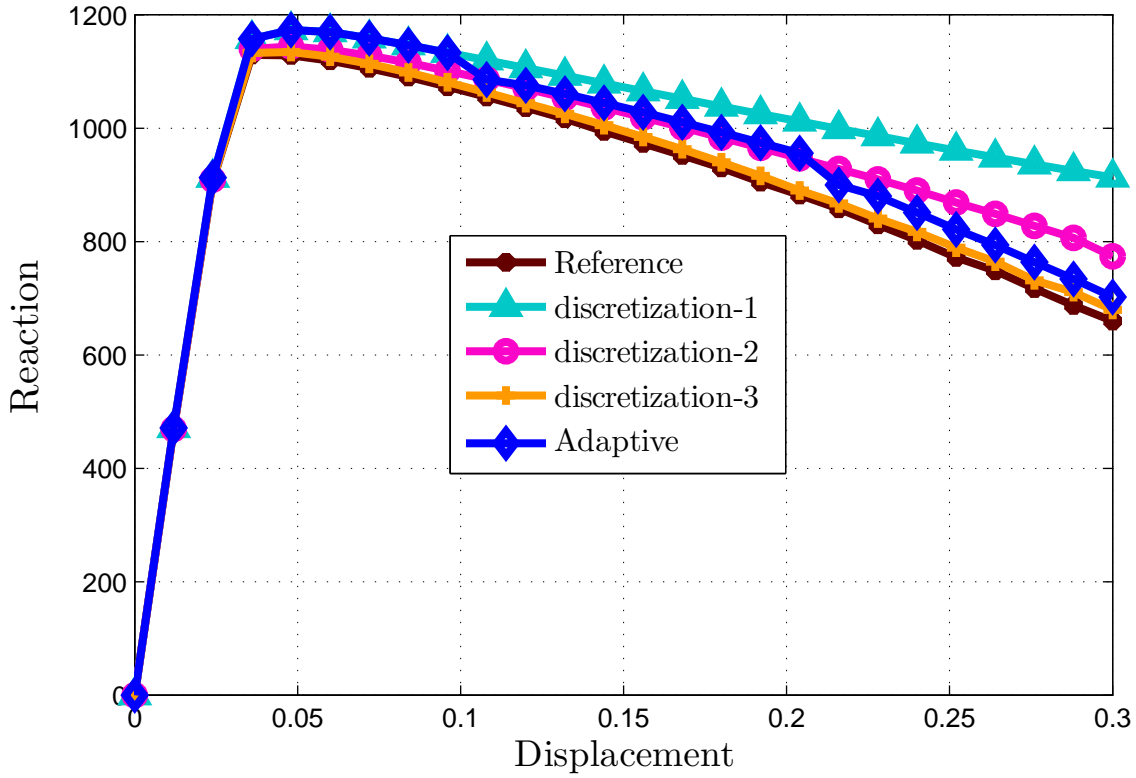


Figure 4.40: Normalized pressure vs displacement for different discretizations for the perforated tensile specimen problem

[158, 159, 162]. The geometry and boundary conditions for this problem are given in Figure 4.45 and the problem is analysed with displacement control under plane strain conditions where a total displacement of 500 mm is applied over 100 equal steps. The material properties used in this case are $E = 2.0 \times 10^4$ Pa, $\nu = 0.3$ and $\sigma_y = 30$ Pa, while the scaling parameters for the domain of influence for analysis and projection used in this case are $d_{max}^a = 1.5$, $d_{max}^p = 1.2$ respectively. A rough and rigid footing is used in

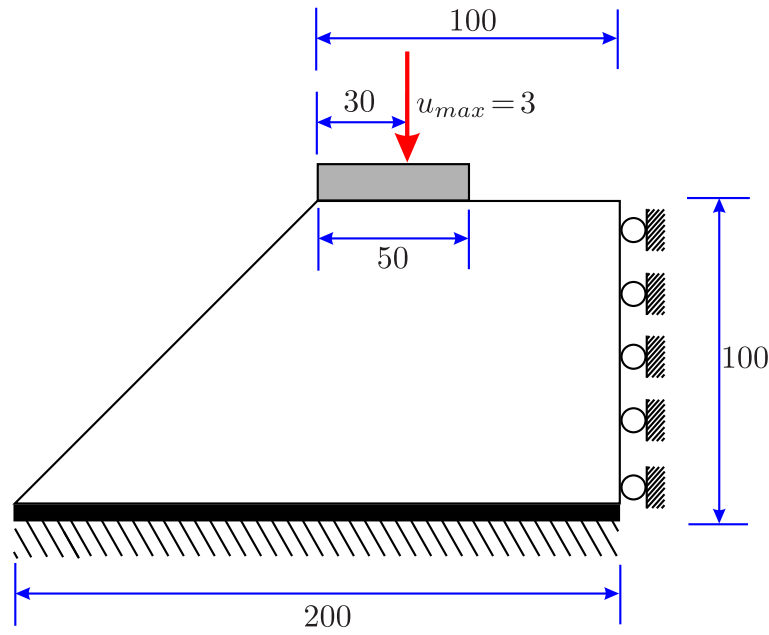


Figure 4.41: Geometry, boundary condition and loading for the footing loaded on a slope problem

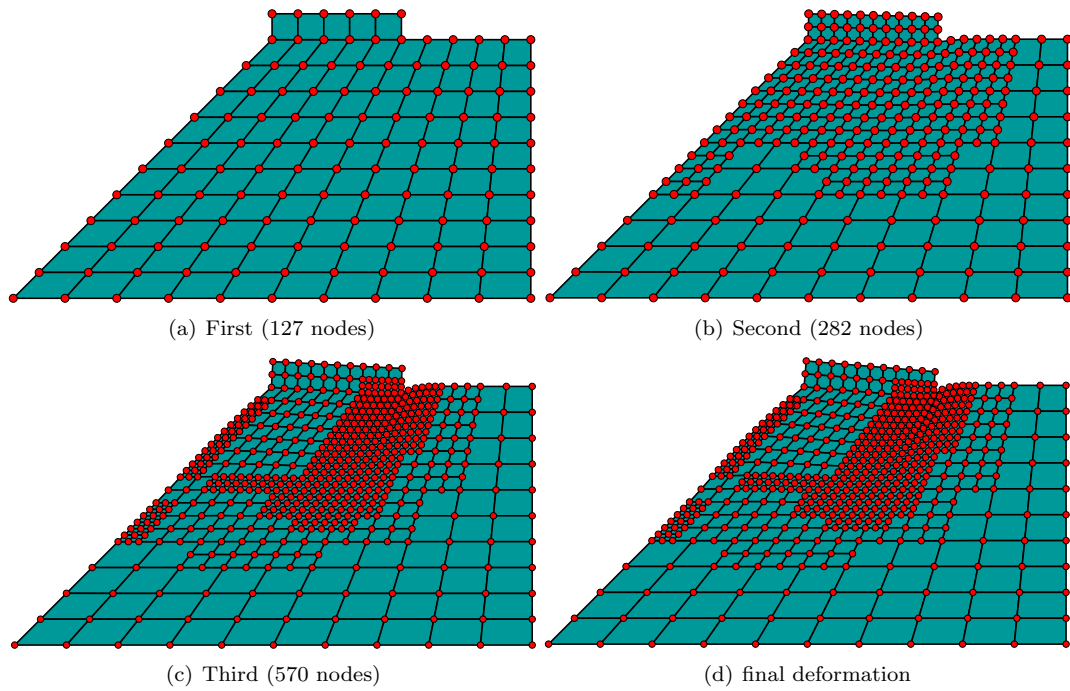


Figure 4.42: Step by step discretizations for the footing loaded on a slope problem

this case with a Young's modulus 100 times that of the block material.

Two refinements are used in this case with a target error of 20%. The initial discretization is shown in Figure 4.46(a), while the first and second refinements appear in Figures 4.46(b) and 4.46(c) respectively. The final deformed configuration for this problem is shown in Figure 4.46(d). The step by step adaptive procedure concentrates nodes along an emerging shear band as expected and once again, the structured nature of the added nodes is clear. The numbers of nodes for the 1st, 2nd and 3rd discretization are 127, 260 and 544 respectively. Contour plots of effective plastic strain over the deformed configuration are

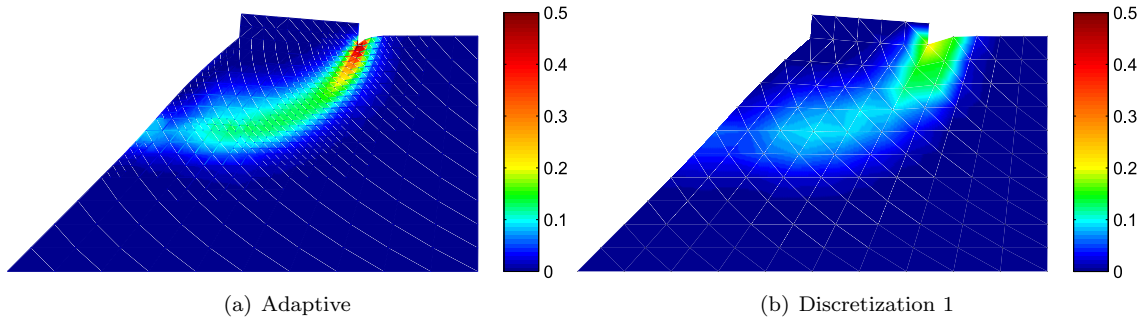


Figure 4.43: Effective plastic strain contours for the footing loaded on a slope problem

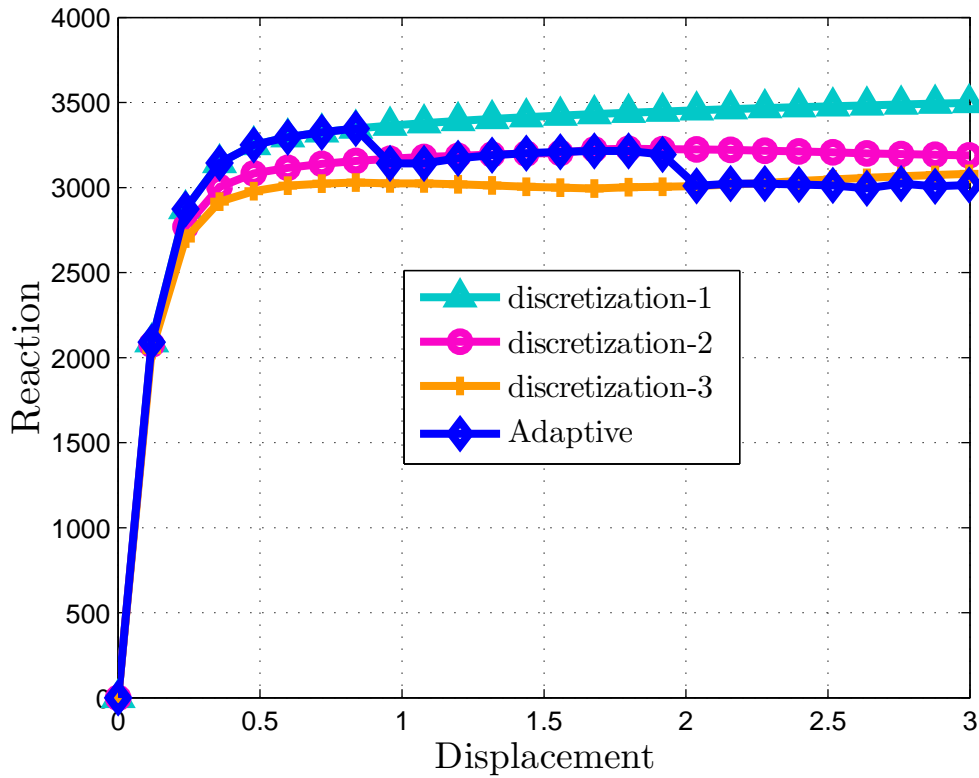


Figure 4.44: Normalized pressure vs displacement for different discretizations for the footing loaded on a slope problem

shown in Figure 4.47(a), which clearly shows a shear band of finite thickness. The same problem is also solved with an initial discretization without refinement, and the contour plot of effective plastic strain is shown in Figure 4.47(b). The comparison between the load displacement curves for the adaptive and coarse discretization is shown in Figure 4.48. Here we can see a perfectly plastic form of failure occurring as the wedge of soil slides as a rigid body along the shear plane. The roughness of the footing and the displacement control serve to control the displacement despite, there being near-zero stiffness. It is clear here that the problem requires finite deformation modelling and it is also clear that finite elements along the shear band would be severely distorted.

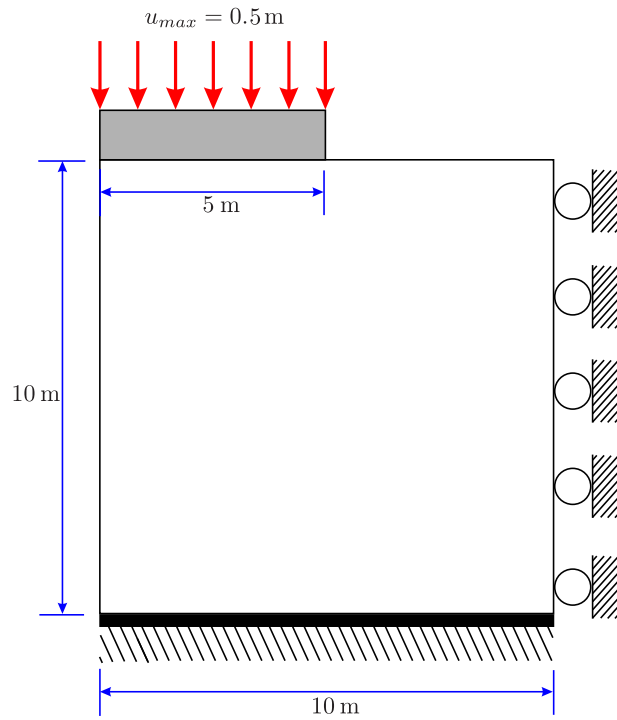


Figure 4.45: Geometry, boundary condition and loading for the footing loaded on a vertical cut problem

4.4 Extension to three-dimensional nonlinear problems

Extension of the proposed adaptive algorithm to the three-dimensional nonlinear problems is straightforward but with additional challenges. Intel® Visual FORTRAN is used as a programming language for three-dimensional problems because of the high computational cost associated with programming using MATLAB. IMSL FORTRAN numerical libraries have been used for the solution of the linear system of equations, calculation of the matrix inverse, eigenvalues and eigenvectors. For three-dimensional problems with unstructured nodes, results from an adaptive analysis, calculation of the proper nodal domains of influence is very challenging. For these problems, three-dimensional Voronoi diagrams are required to calculate the influence domains as explained in §4.2.3 for a two-dimensional case. Luckily, Voro++ [265, 266] is available and has been used here. This is an open-source software library for the calculation of three-dimensional Voronoi diagrams for a set of particles in space, written in C++. The Voro++ library also provides a list of neighbouring nodes for all the nodes in the problem domain based on the nodes which share the same face of the Voronoi cells. This neighbouring list can be used for the calculation of the influence domain based on the maximum distance between the nodes of interest and the other neighbouring nodes. Since it is required for the updated Lagrangian formulation to calculate the nodal influence domains at the end of each solution step and for each new discretization after refinement, it is necessary to link the Voro++ library with to the FORTRAN code to make the process automatic. The Voro++ library is written in C++, so the ISO.C.BINDING module is used to call and linked the library from FORTRAN code. A sample 3D discretization is shown in Figure 4.49(a), the corresponding Voronoi diagram generated with the Voro++ library is shown in Figure 4.49(b) and the resultant hexahedral nodal influence domains (reduced in size for clarity) are shown in Figure 4.49(c).

For three-dimensional problems, the formulations for the error estimation and projection of the path dependent variables between the old and new discretizations are the same as in the two-dimensional case and as explained in the previous section. The refinement strategy for a three-dimensional hexahedral

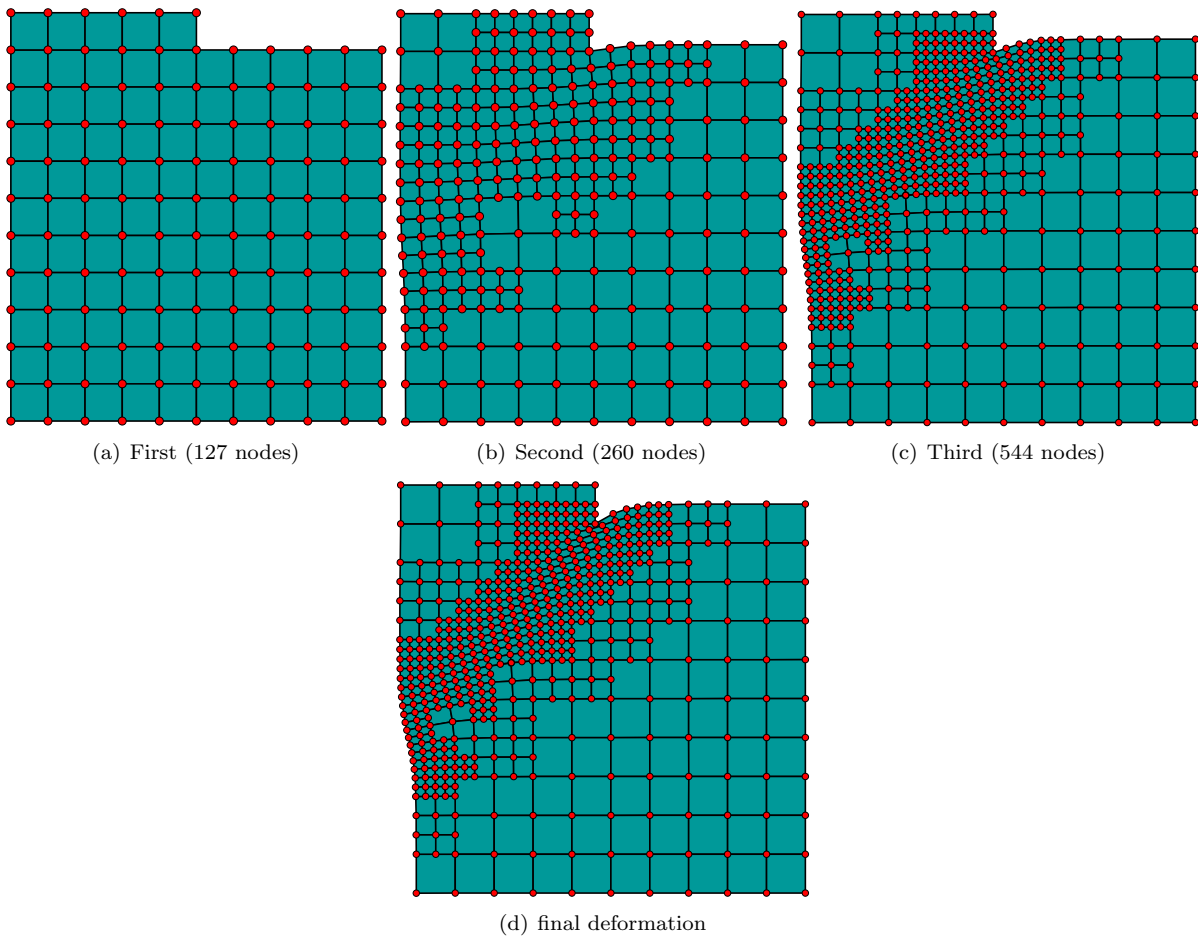


Figure 4.46: Step by step discretizations for the footing loaded on a vertical cut problem

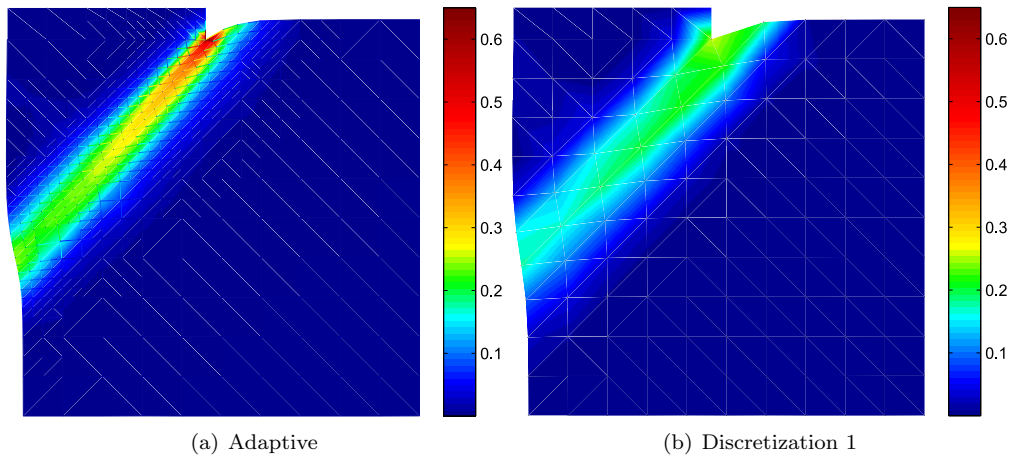


Figure 4.47: Effective plastic strain contours for the footing loaded on a vertical cut problem

background cell is shown in Figure 4.50. Nineteen new nodes are added in this case with one on each side, one on each face and one in the middle of the original cell. Furthermore, the cell is divided into eight new cells.

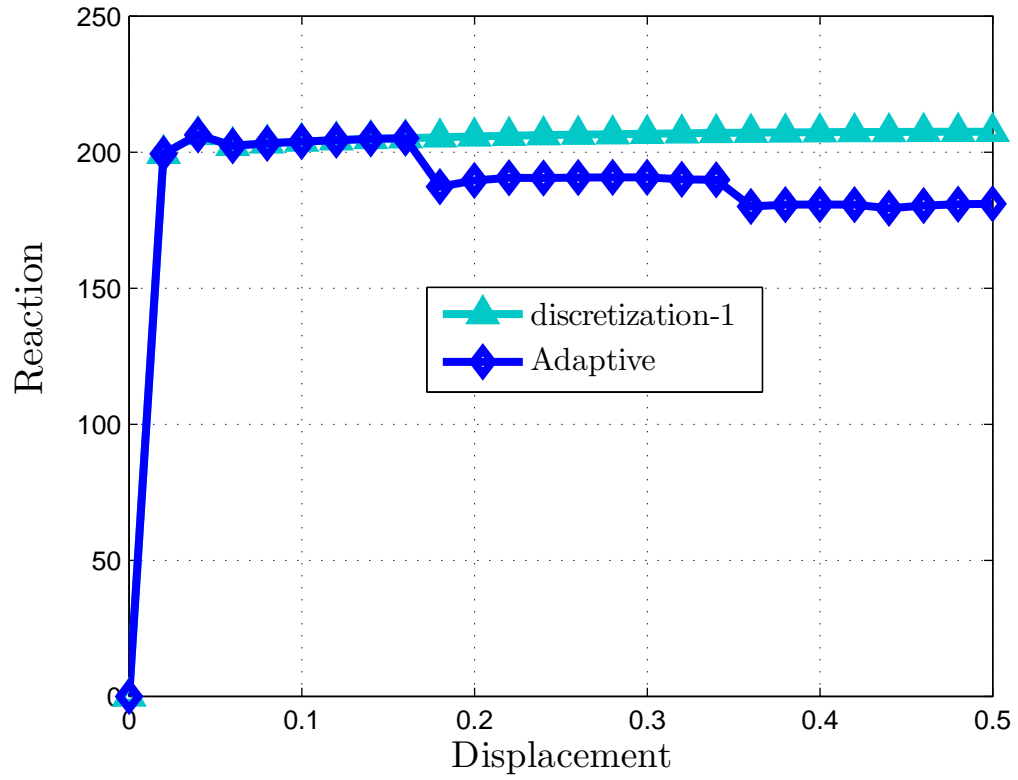


Figure 4.48: Normalized pressure vs displacement for different discretizations for the footing loaded on a vertical cut problem

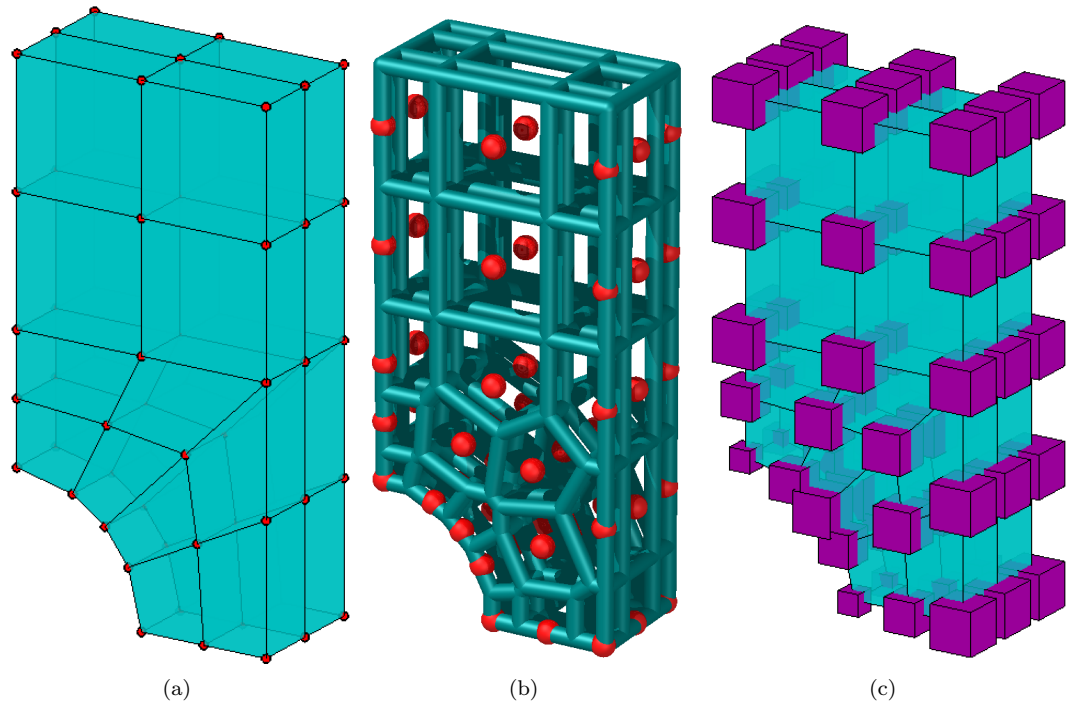


Figure 4.49: Three-dimensional problem discretization, Voronoi diagram and corresponding nodal domains of influence

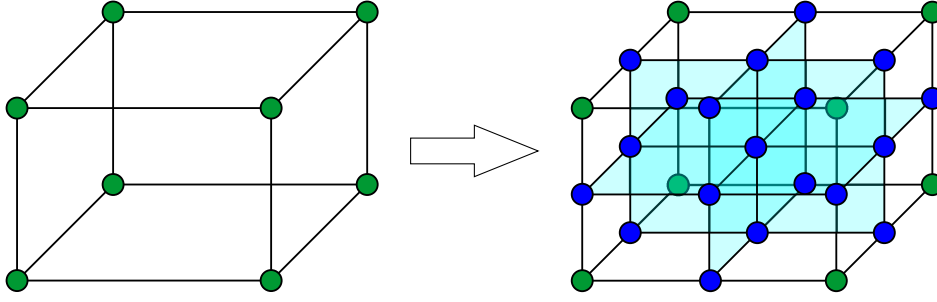


Figure 4.50: Three dimensional refinement strategy

4.4.1 Numerical examples

Numerical examples are now presented to demonstrate the implementation and performance of the current approach in solving three-dimensional nonlinear problems. Cubic spline weight functions are used for the calculation of the three-dimensional MLS and max-ent shape functions while the Prandtl-Reuss constitutive model is used in elasto-plasticity modelling in all the numerical examples in this section.

Three-dimensional plate with a hole

The first three-dimensional numerical example is a plate with a central hole subjected to unidirectional tension; the same numerical example is also given in [93]. Geometry and loading for this problem are shown in Figure 4.51. Due to symmetry only the one-eighth of the problem shown in gray in Figure 4.51 is modelled with the material properties $E = 1.0 \times 10^5$, $\nu = 0.3$ and $\sigma_y = 1.0 \times 10^3$, all in compatible units. The scaling parameters used for the domains of influence for analysis, and projection are $d_{max}^a = 1.5$, $d_{max}^p = 1.1$ respectively. The total displacement of 0.1 units is applied to the top face with 10 equal steps and a relative error used for adaptive analysis is 20%

The step by step adaptive refinements are shown in Figure 4.52 with 63, 243 and 847 nodes in the consecutive discretizations. The refinement algorithm automatically adds nodes in the thinning section of the plate, which needs more nodes to accurately model the solution. The step by step plots for the Voronoi diagrams, which are used here for the calculation of the influence domains are shown in Figure 4.53 and the corresponding domains of influence are shown in Figure 4.54. From Figure 4.54 it is clear that the nodal domains of influence are smaller in a region, where nodes are relatively close to each other. The contours of the effective plastic strain at the end of adaptive analysis are shown in Figure 4.55(a), where the formation of the shear band starting from the hole of finite thickness is clear. The same problem is also solved with initial discretization without adaptivity and the contours of the effective plastic strain with the same scale as in Figure 4.55(a) are shown in Figure 4.55(b) with no clear shear band. This shows the effectiveness of the proposed adaptive strategy for three-dimensional problems. The reaction versus displacement curve at the end of analysis for the adaptive and different initial discretizations shown in Figure 4.52 without adaptivity are shown in Figure 4.56. Discretization-1 shows a rigid response without softening as expected due to necking as compared to all other three cases. The region inside the rectangle is magnified and is shown in same Figure 4.56, where jumps in the adaptive curves are clear, which is the point where refinement takes place in the adaptive analysis.

Three-dimensional footing loaded on a vertical cut

The second three-dimensional problem solved by the proposed approach is a three-dimensional footing loaded near the unsupported faces of a vertical cut. The geometry and loading for this problem are shown in Figure 4.57. This problem is solved with material properties $E = 2.0 \times 10^4$, $\nu = 0.3$ and $\sigma_y = 30$ all in compatible units. The scaling parameters used for the nodal domain of influence for analysis, and

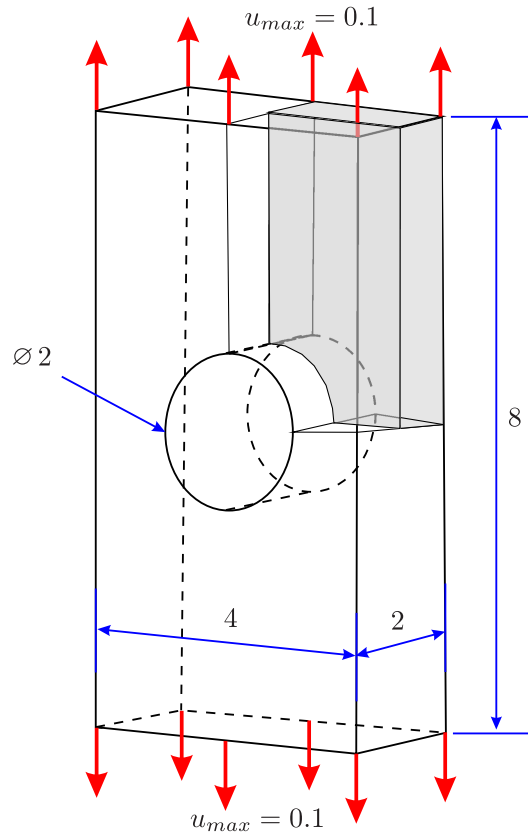


Figure 4.51: Geometry, boundary condition and loading for the 3D plate with a hole problem

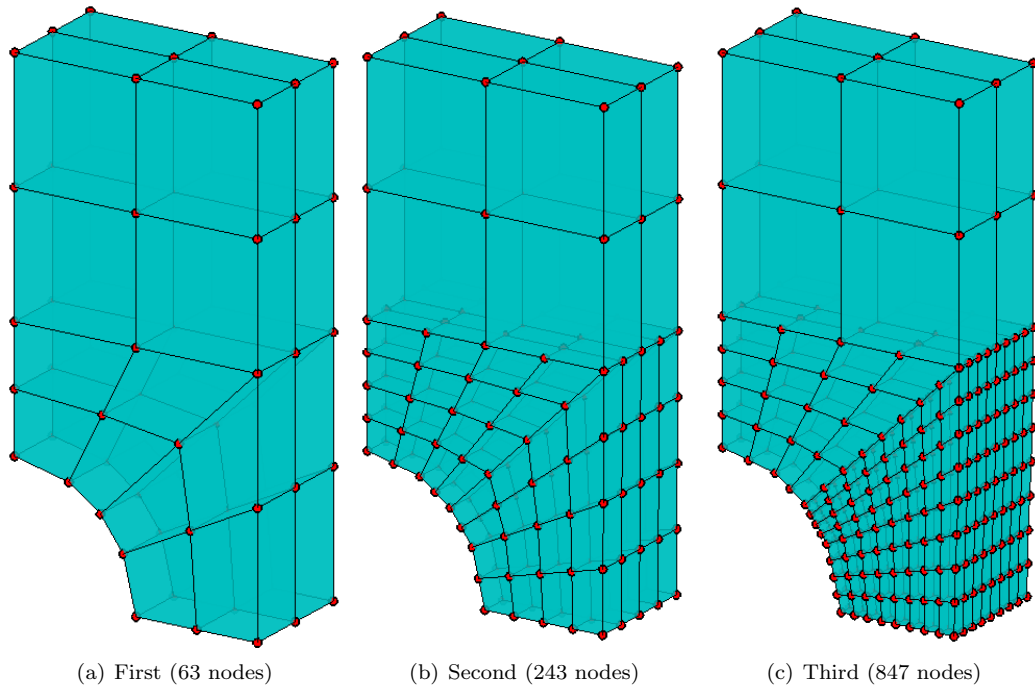


Figure 4.52: Step by step discretizations for the 3D plate with a hole problem

projection are $d_{max}^a = 1.35$, $d_{max}^p = 1.01$ respectively. Only the vertical cut section of the problem is analysed without the footing for the ease of computation, while the rigid footing is modelled by giving

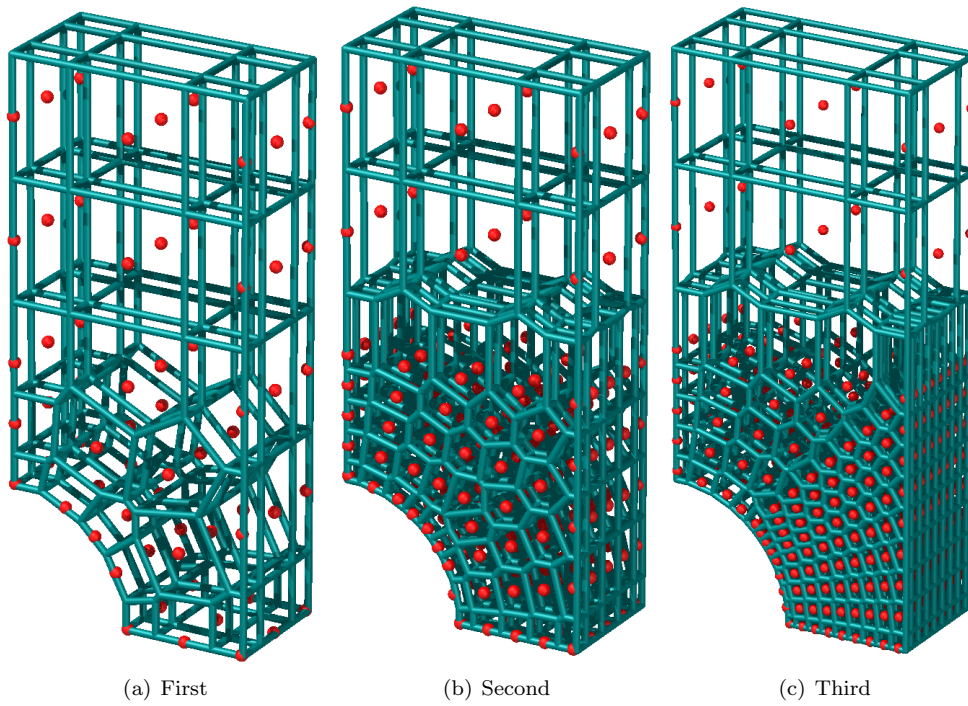


Figure 4.53: Step by step Voronoi diagrams for the 3D plate with a hole problem

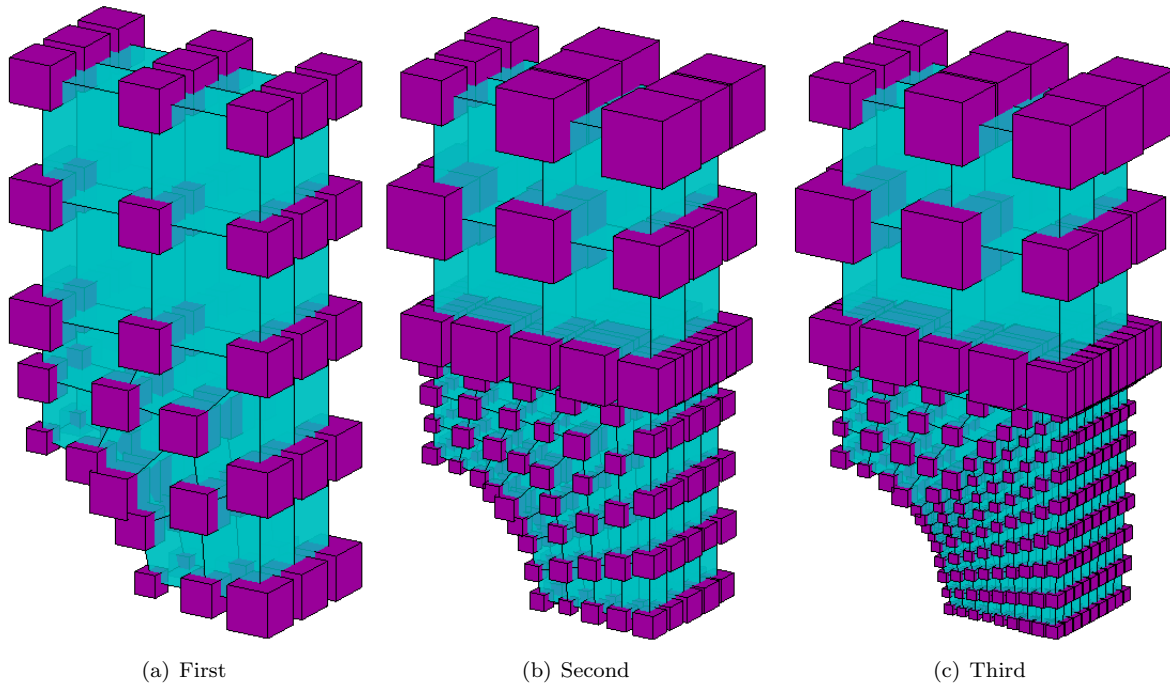


Figure 4.54: Step by step influence domains for the 3D plate with a hole problem

the surface nodes attached to the footing a vertical downward displacement. A total displacement of 0.1 units is applied in 15 equal steps.

A 25% relative error is used for the adaptive analysis, and the resultant step by step adaptive discretizations are shown in Figure 4.58 with 125, 417 and 1579 nodes in the first, second and third discretizations respectively. The adaptive algorithm adds nodes in a region, where failure is expected and

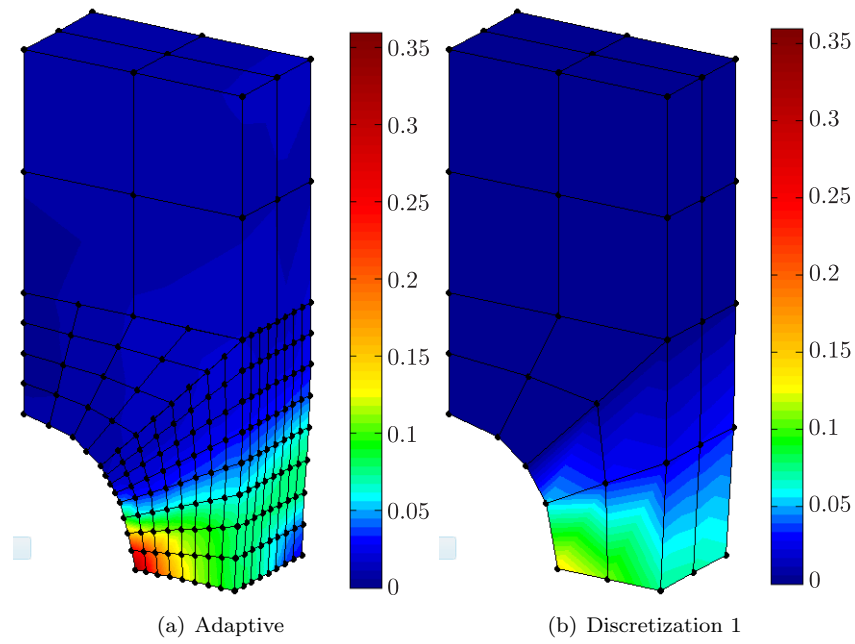


Figure 4.55: Effective plastic strain contours for the 3D plate with a hole problem

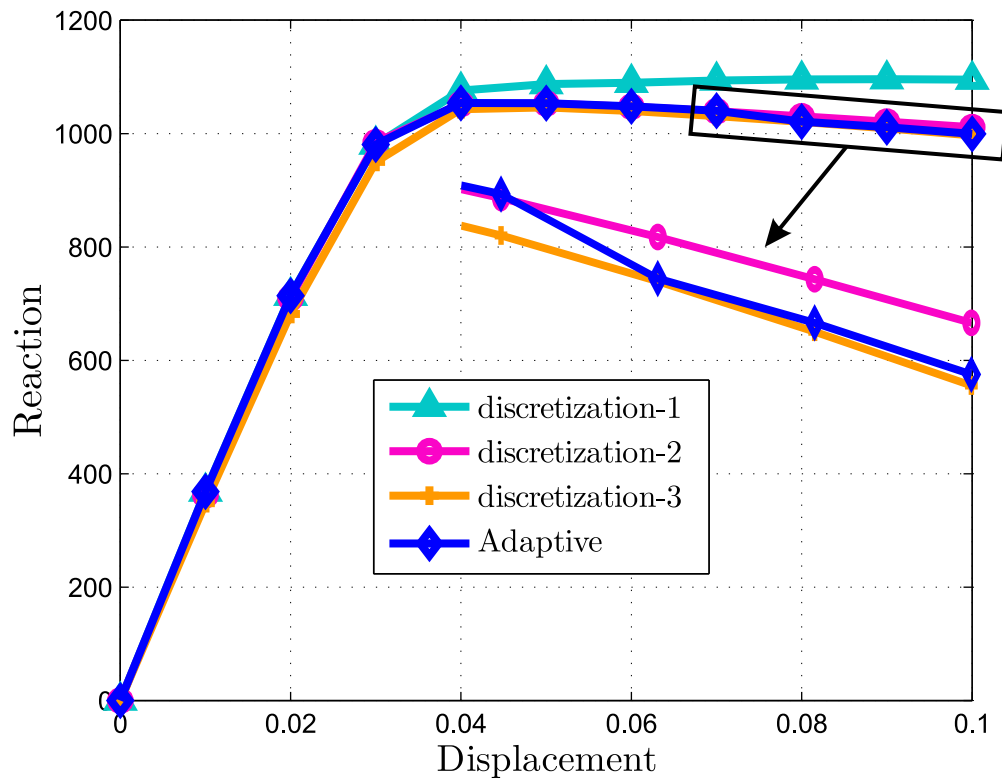


Figure 4.56: Reaction vs displacement for different discretizations for the 3D plate with a hole problem

more nodes are needed for accurate modelling. The corresponding Voronoi diagrams for the step by step discretizations are shown in Figure 4.59. The corresponding step by step influence domains are also shown in Figure 4.60. The contours of the effective plastic strain at the end of adaptive analysis is shown

in Figure 4.61(a). The formation of the shear band with finite thickness is very clear in this case. The same problem is also solved by the initial discretization and the contours of the effective plastic strain at the end of analysis on the same scale as Figure 4.61(a) are shown in Figure 4.61(b) with no clear shear band. This shows the effectiveness of our proposed algorithm in case of three-dimensional problems. The load versus displacement curve for the adaptive and different initial discretization shown in Figure 4.58 without adaptivity are also given in Figure 4.62. Steps in the curve for the adaptive case are points where refinement takes place.

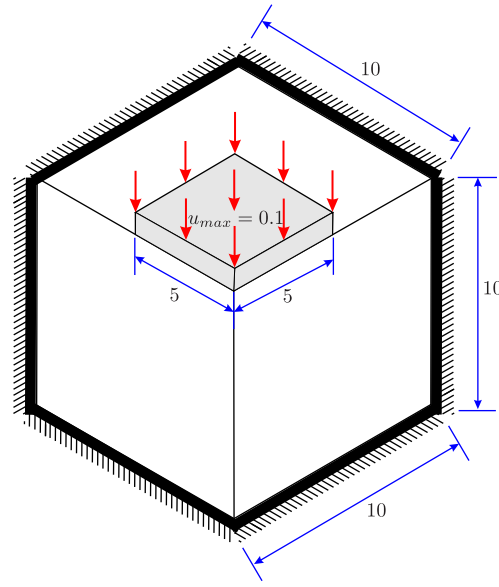


Figure 4.57: Geometry, boundary condition and loading for the 3D vertical cut problem

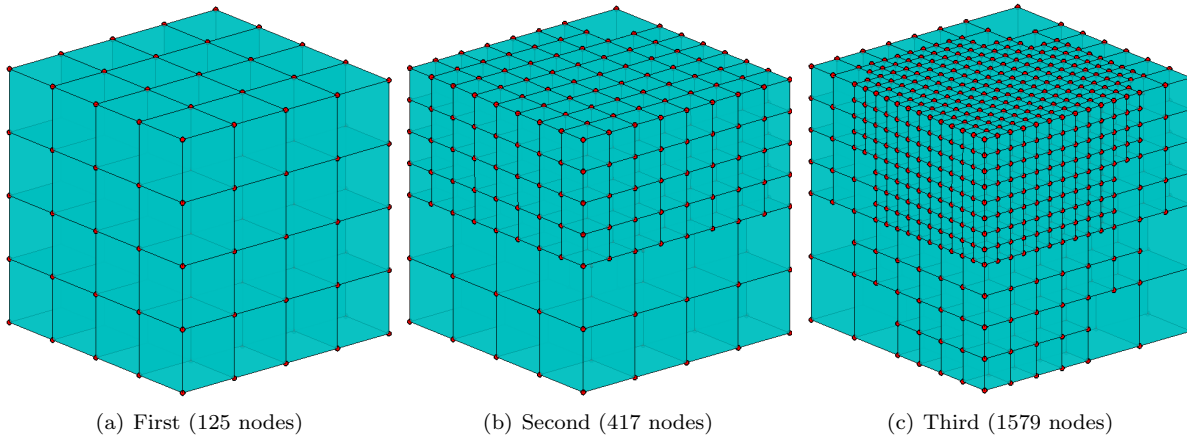


Figure 4.58: Step by step discretizations for the 3D vertical cut problem

4.5 Concluding remarks

In this chapter, a new numerical model for use in two- and three-dimensional nonlinear solid mechanics has been described using meshless methods based on an updated Lagrangian formulation. MATLAB is used as a programming language for two-dimensional problems, while due to the high computational cost of the later FORTRAN is used as a programming language for three-dimensional problems. The proposed model can accommodate both geometrical and material nonlinearity and includes automatic adaptive refinement

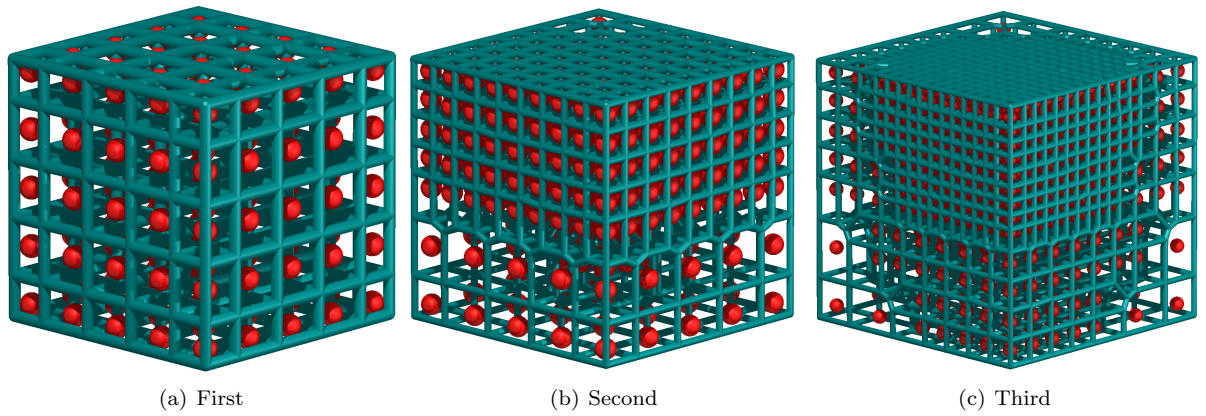


Figure 4.59: Step by step Voronoi diagrams for the 3D vertical cut problem

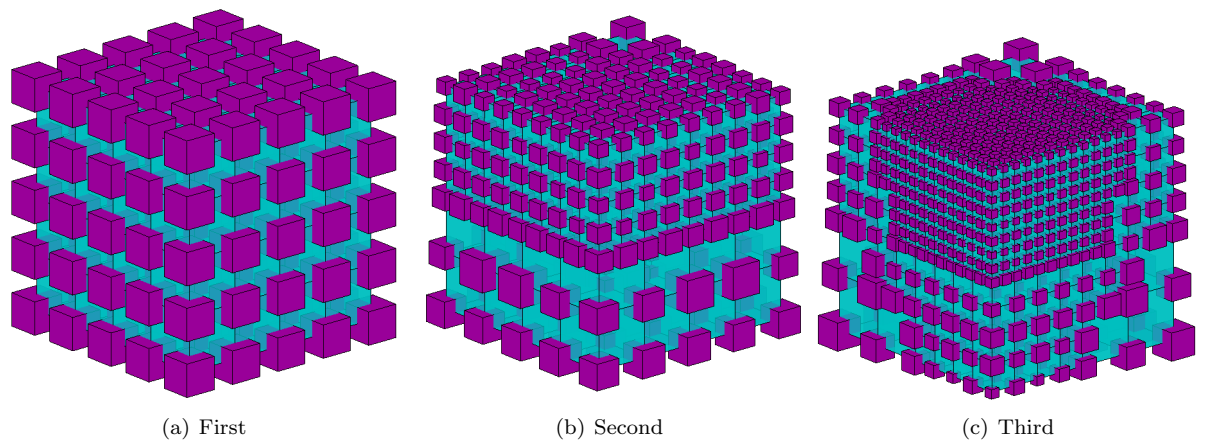


Figure 4.60: Step by step influence domains for the 3D vertical cut problem

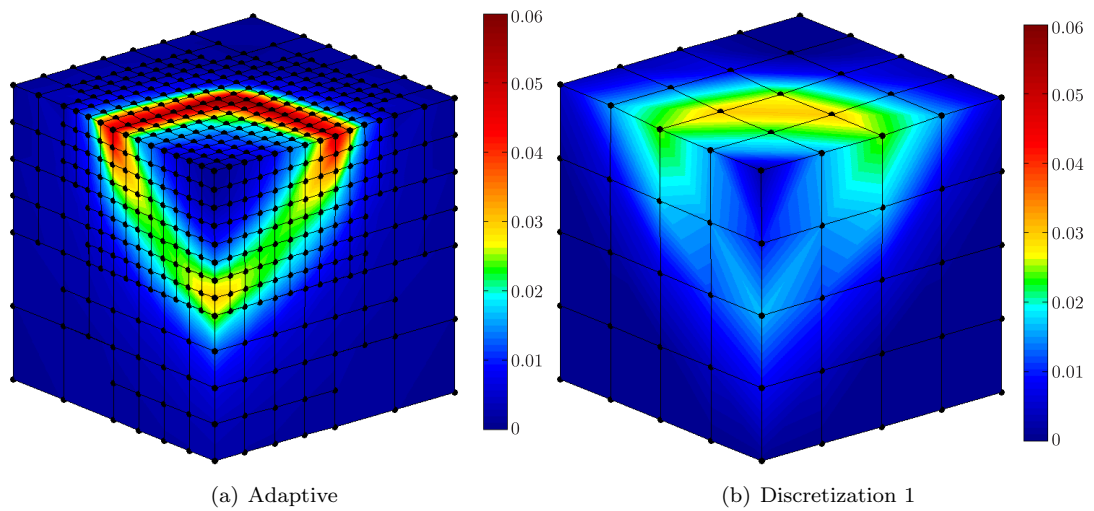


Figure 4.61: Effective plastic strain contours for the 3D plate with a hole problem

as well as a number of other innovative features (such as max-ent shape functions). The error estimation procedure proposed by Chung and Belytschko [62] for two-dimensional linear elasto-static problems for the EFGM is extended here to two- and three-dimensional nonlinear problems. The strategy for the calculation of nodal domains of influence based on Voronoi diagrams is used, which provides enough

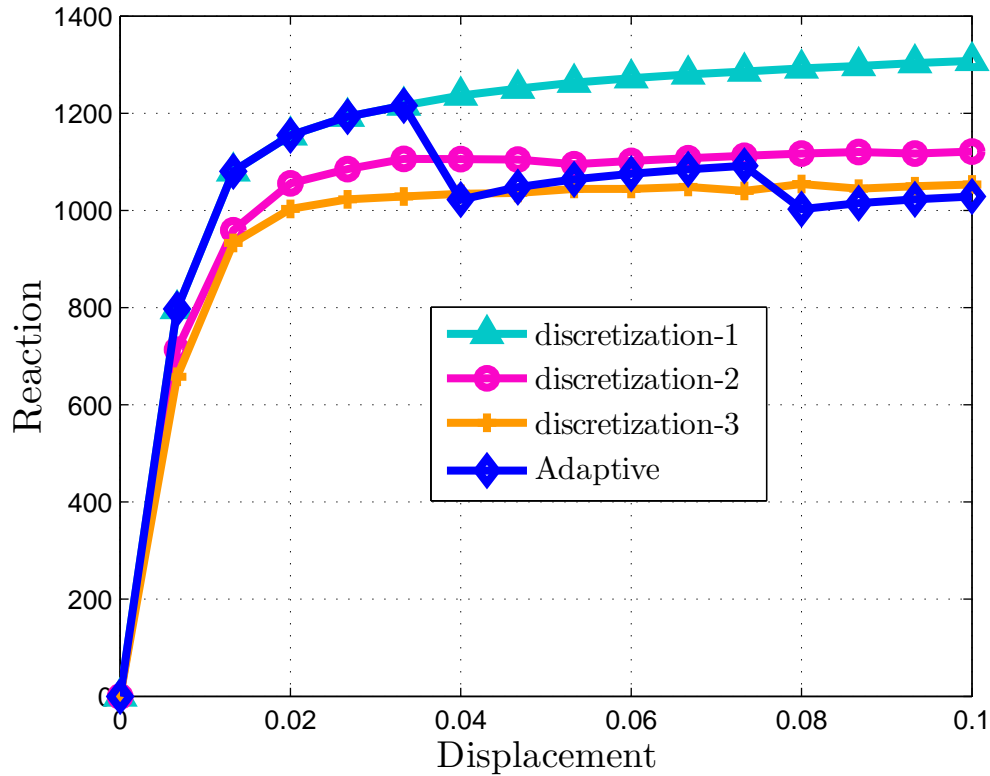


Figure 4.62: Reaction vs displacement for different discretizations for the 3D vertical cut problem

nodes for the stable calculation of the shape function and can also capture the local material behaviour accurately without over-smoothing the results. For the generation of the Voronoi diagrams for a set of nodes, the MATLAB built-in function is used for two-dimensional problems, while for three-dimensional problems, the Voro++ software library is used in the FORTRAN code. An incremental error is calculated for each solution step and is used as the criterion to identify regions for refinement, which can take place very simply both due to the meshless approach and the chosen structured grids of nodes. MLS shape functions are used to transfer the path dependent variables from the old discretization to the new. A number of two-dimensional linear elastic and two- and three-dimensional nonlinear numerical examples have been presented to demonstrate the capabilities of the new modelling procedure as a whole.

Chapter 5

Finite element-element-free Galerkin method coupling

5.1 Introduction

The EFGM is superior to the FEM in terms of accuracy per degree of freedom and convergence but is computationally more expensive. Therefore, it is more practical to use the EFGM only in a region, which is difficult to model using the FEM, while the FEM can be used in the remaining part of the problem domain. The proper coupling between the FEM and the EFGM is then of prime importance for accurate results.

Interface elements have been used between the EFG and the FE regions of a problem in the past [40, 243, 312] due to the incompatibility between the use of the MLS shape functions within the EFG region for the approximation of the field variables and standard shape functions in the FE region. Hybrid shape functions of these elements consisting of both the FEM, and the EFGM shape functions were formed using ramp functions [40]. In [166], to facilitate the direct imposition of essential boundary conditions; a strip of the FE was used on the essential boundary while the EFGM was used in the remaining part of the problem domain. Furthermore, the FE and the EFG regions were coupled using the procedure proposed in [40], i.e. using interface elements between the FE and the EFG regions of the problem domain. The FE-EFGM coupling procedure of [40] was also extended for the case of nodal integration used in the EFG region in [35]. A continuous blending method for the FE-EFGM coupling was introduced in [94, 135], as compared to the FE-EFGM coupling procedure of [40], there is no need for ramp functions and the use of the FE nodes as the EFG nodes are not required therefore, the EFG nodes can also be added within the transition region. Lagrange multipliers were used for the same coupling in [126] and this approach was used for linear-elastic numerical examples. The FE-EFGM coupling with Lagrange multipliers was also extended to nonlinear, reinforced concrete problems in [257], in which reinforcement was modelled with the FEM and concrete with the EFGM. In [119] a transition or bridge region was used for coupling between the the FEM and meshless methods. The transition region was discretized by particles, which were independent of the FE and meshless nodes. A detail review of the coupling between the EFGM and the FEM can be found in [259]. A coupled FE-EFGM was proposed in [306] for the simulation of automotive crash tests, in which areas with very high deformation were modelled with the EFGM. A constraint was used to ensure the continuity of the shape functions across the FE-EFGM interface without using proper interface elements. A slight variation of the FE-EFGM coupling with interface elements [40], in which there was no need for a pre-existing transition region between the FE and EFG

regions was proposed in [260] for the simulation of linear elastic fracture mechanics problems, including mode-I, mode-II and mixed mode problems. The area near the crack was modelled with the EFGM, while the FEM was used in the remaining part of the problem domain. A coupled FE-EFGM procedure based on a collocation approach was proposed in [315], in which at the interface between the FE and the EFG regions, fictitious nodal values were converted to real nodal displacements using the MLS shape functions and were assigned back to the FE nodes. In [91], the FE-EFGM coupling procedure of [40] was used to couple the EFGM with the FEM to conveniently impose the essential boundary conditions in the dynamic soil structure interaction problems.

In this chapter, a new way of coupling the EFGM and the FEM is proposed using local max-ent shape functions in the EFG zone, the detail of max-ent shape function can be found in §2.6. These shape functions possess a weak Kronecker delta property at the boundaries, which provides a natural way to couple the EFG with the FE regions as compared to the MLS basis functions, which need extra care to properly couple the same regions. In this new approach, there is no need for interface elements between the EFG and the FE regions or any other special treatment for the shape function continuity across the FE-EFGM interface as required in most previous research as outlined above. One-, two- and three-dimensional linear and two-dimensional nonlinear benchmark numerical examples are solved by the new approach to show the implementation and performance of the current approach. A convergence study is also performed based on error in energy norm. It should be noted that this approach is entirely new and has not, to the author's knowledge, been suggested before.

The outline of this chapter is as follows. A detailed description of the conventional FE-EFGM coupling procedure in two-dimensions is given in §5.2, followed by the explanation of the shape functions used in the FE, EFG and interface regions in §5.2.1. The conventional way of coupling is also extended to one- and three-dimensional cases in §5.2.2. A detailed description of the new way of coupling is given in §5.3. One-, two- and three-dimensional, linear elastic numerical examples are then given in §5.4 to show the implementation and performance of the coupling procedure for linear elastic problems. The proposed coupled FE-EFGM approach is then extended to nonlinear problems in §5.5, followed by two nonlinear numerical examples in §5.5.1. Concluding remarks are given in §5.6.

5.2 FE-EFGM coupling using interface elements

A two-dimensional FE-EFGM coupling formulation, using the MLS shape function in the EFG zone for the approximation of the field variables is given here, which is straightforward to modify for one- or three-dimensional problems. The MLS shape functions do not possess the Kronecker delta property like the FE shape functions and due to this reason interface elements are introduced between the FE and the EFG zones [40, 243, 312], to properly couple the two regions. The coupling procedure, using interface elements is shown in Figure 5.1 in which Ω_E , Ω_I and Ω_F are the EFG, interface elements and the FE regions respectively and Γ_E and Γ_F are the boundaries of the interface elements on the EFG and the FE sides respectively. In Figure 5.1, interface element A has three EFG nodes and one FE node, interface element B has one EFG node and three FE nodes and interface element C has two FE nodes and two EFG nodes. The shape functions for the interface elements are hybrid shape functions of the FE and the EFG shape functions to make the displacement continuous along Γ_E and Γ_F . Shape functions for the three regions, i.e. the FE, the EFG and interface elements are explained in the next section.

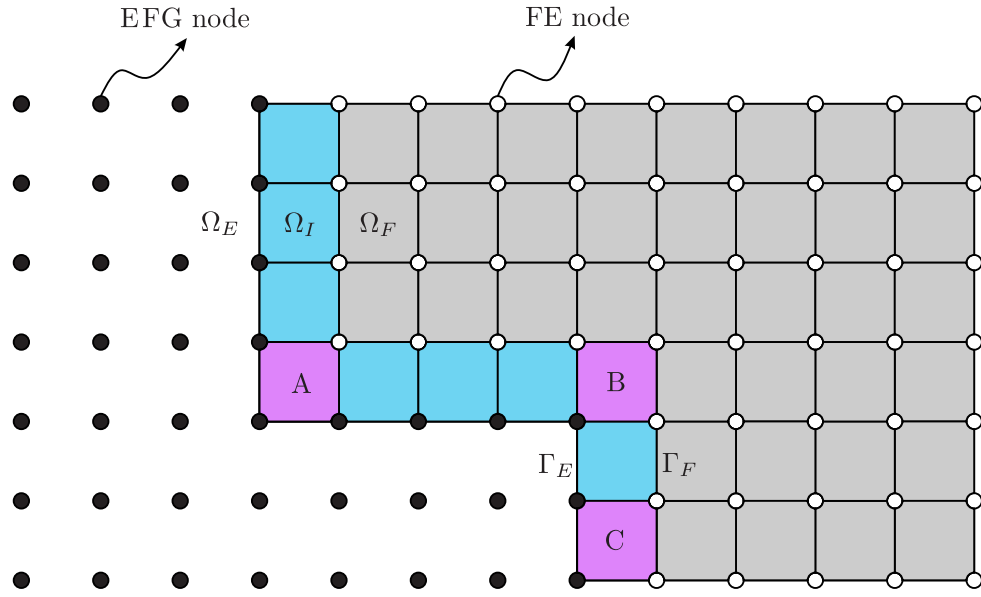


Figure 5.1: FE-EFGM coupling with interface elements

5.2.1 Shape functions

Displacement can be approximated at point \mathbf{x} in a similar way in the three regions, i.e.

$$u^h(\mathbf{x}) = \sum_{i=1}^n \tilde{N}_i(\mathbf{x}) u_i, \quad (5.1)$$

where $u^h(\mathbf{x})$ is the approximate displacement component at point \mathbf{x} , $\tilde{N}_i(\mathbf{x})$ is either the FE, the EFG or interface element shape function for node i evaluated at point \mathbf{x} , n is the number of nodes in support of point \mathbf{x} and u_i are either nodal displacements in the case of the FEM, or nodal parameters in case of the EFGM.

FE shape functions

In the FE region four node iso-parametric quadrilateral elements are used. A single four node iso-parametric quadrilateral element is shown in Figure 5.2 in natural or parent coordinates ξ and η . For

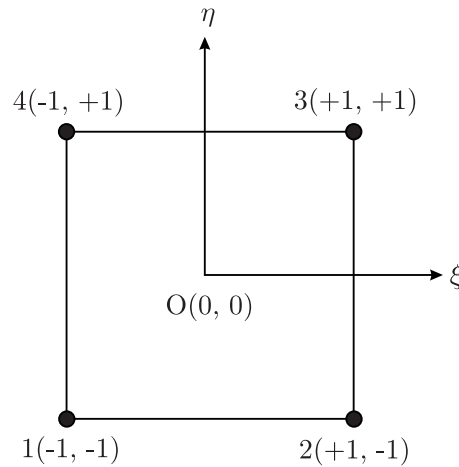


Figure 5.2: Four node iso-parametric quadrilateral element in natural coordinates

this element, shape functions are the standard bilinear shape functions [283], i.e.

$$N_i = \frac{1}{4} (1 + \xi_i \xi) (1 + \eta_i \eta), \quad (5.2)$$

where i is the node numbering, while ξ_i and η_i are the nodal coordinates of the parent element.

EFGM shape functions

In the EFG region MLS shape functions are used for the approximation of displacements, which are already explained in §2.4.

Interface element shape functions

In the interface region displacement is approximated using both the FE and the EFG displacements and is written as

$$u^h(\mathbf{x}) = [1 - R(\mathbf{x})] u^{FE}(\mathbf{x}) + R(\mathbf{x}) u^{EFG}(\mathbf{x}), \quad \mathbf{x} \in \Omega_I, \quad (5.3)$$

where $u^{FE}(\mathbf{x})$ and $u^{EFG}(\mathbf{x})$ are the approximations of displacement at \mathbf{x} of the FEM and the EFGM respectively and $R(\mathbf{x})$ is the ramp function and is written as

$$R(\mathbf{x}) = \sum_j N_j(\mathbf{x}), \quad \mathbf{x}_j \in \Gamma_E, \quad (5.4)$$

i.e. R is the sum of shape functions of nodes of the interface element on the EFG region side, Γ_E . R varies linearly from one on the EFG region boundaries Γ_E to zero on the FE region boundaries Γ_F , i.e.

$$R(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Gamma_E \\ 0 & \mathbf{x} \in \Gamma_F. \end{cases} \quad (5.5)$$

In Equation (5.3), displacement u^h will reduce to u^{FE} and u^{EFG} on Γ_F and Γ_E respectively. Combining Equations (5.3) and (5.1)

$$u^h(\mathbf{x}) = [1 - R(\mathbf{x})] \sum_{i=1}^{ne} N_i(\mathbf{x}) u_i + R(\mathbf{x}) \sum_{i=1}^{ns} \phi_i(\mathbf{x}) u_i \quad \mathbf{x} \in \Omega_I, \quad (5.6)$$

where ne is the number of nodes in an element used in the FE regions and ns is the number of nodes in the support of point \mathbf{x} in the EFG region. Equation (5.6) can also be written as

$$u^h(\mathbf{x}) = \sum_{i=1}^{ns} \bar{\phi}_i(\mathbf{x}) u_i \quad \mathbf{x} \in \Omega_I, \quad (5.7)$$

where $\bar{\phi}_i$ are the shape functions of the interface elements and can be written as

$$\bar{\phi}_i(\mathbf{x}) = \begin{cases} [1 - R(\mathbf{x})] N_i(\mathbf{x}) + R(\mathbf{x}) \phi_i(\mathbf{x}) & \mathbf{x}_i \in \Omega_I^e, \\ R(\mathbf{x}) \phi_i(\mathbf{x}) & \mathbf{x}_i \notin \Omega_I^e, \end{cases} \quad (5.8)$$

where Ω_I^e is an interface element. The equation for the shape function derivative for the interface elements is written as

$$\bar{\phi}_{i,k} = \begin{cases} R_{,k} N_i + [1 - R] N_{i,k} + R_{,k} \phi_i + R \phi_{i,k} & \mathbf{x}_i \in \Omega_I^e, \\ R_{,k} \phi_i + R \phi_{i,k} & \mathbf{x}_i \notin \Omega_I^e, \end{cases} \quad (5.9)$$

where

$$R_{,k} = \sum_j N_{j,k}, \quad \mathbf{x}_j \in \Gamma_E. \quad (5.10)$$

In Equation (5.9), reference to \mathbf{x} is eliminated for clarity. A blending function, $\beta(\mathbf{x})$ is used instead of $R(\mathbf{x})$ in Equation (5.3) in References [243, 312] for displacement approximation, which is given as

$$\beta(\mathbf{x}) = 3R^2(\mathbf{x}) - 2R^3(\mathbf{x}). \quad (5.11)$$

Here $\beta(\mathbf{x})$ will create a smooth blending in the interface elements between the FE and the EFG region and will reduce the discontinuity in the shape function derivatives along Γ_E and Γ_F . After using $\beta(\mathbf{x})$ instead of $R(\mathbf{x})$, Equations (5.3), (5.8) and (5.9) are written as

$$u^h(\mathbf{x}) = [1 - \beta(\mathbf{x})] u^{FE}(\mathbf{x}) + \beta(\mathbf{x}) u^{EFG}(\mathbf{x}) \quad \mathbf{x} \in \Omega_I, \quad (5.12)$$

$$\bar{\phi}_i(\mathbf{x}) = \begin{cases} [1 - \beta(\mathbf{x})] N_i(\mathbf{x}) + \beta(\mathbf{x}) \phi_i(\mathbf{x}) & \mathbf{x}_i \in \Omega_I^e, \\ \beta(\mathbf{x}) \phi_i(\mathbf{x}) & \mathbf{x}_i \notin \Omega_I^e, \end{cases} \quad (5.13)$$

$$\bar{\phi}_{i,k} = \begin{cases} \beta_{,k} N_i + [1 - \beta] N_{i,k} + \beta_{,k} \phi_i + \beta \phi_{i,k} & \mathbf{x}_i \in \Omega_I^e, \\ \beta_{,k} \phi_i + \beta \phi_{i,k} & \mathbf{x}_i \notin \Omega_I^e, \end{cases} \quad (5.14)$$

where $\beta_{,k}$ is written as

$$\beta_{,k} = (6R - 6R^2) R_{,k}. \quad (5.15)$$

A comparison between $R(\mathbf{x})$ and $\beta(\mathbf{x})$ for a one-dimensional interface element is shown in Figure 5.3. $R(\mathbf{x})$ varies from 0 on Γ_F to 1 on Γ_E linearly, while smooth blending can be seen in case of $\beta(\mathbf{x})$, especially at the boundaries Γ_F and Γ_E , which will produce continuous shape function derivatives along Γ_F and Γ_E . The same comparison for $R(\mathbf{x})$ and $\beta(\mathbf{x})$ in the case of two-dimensional interface elements A, B and C, shown in Figure 5.1 are given in Figure 5.4. Figures 5.4(a) and 5.4(b) shows the plots for $R(\mathbf{x})$ and $\beta(\mathbf{x})$ respectively for the element A, with three EFG and one FE nodes. Both $R(\mathbf{x})$ and $\beta(\mathbf{x})$ varies from 0 on Γ_F to 1 on Γ_E and the smoothness of $\beta(\mathbf{x})$ can also be seen as compared to the corresponding $R(\mathbf{x})$. The same comparison is also shown for element B with one EFG and three FE nodes and C with two EFG and two FE nodes in Figure 5.4(c), 5.4(d), 5.4(e) and 5.4(f).

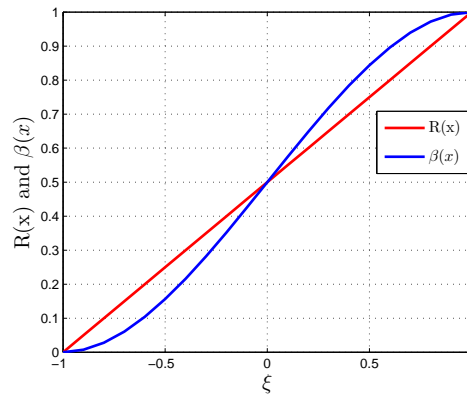


Figure 5.3: Sample one-dimensional ramp $R(\mathbf{x})$ and blending $\beta(\mathbf{x})$ functions

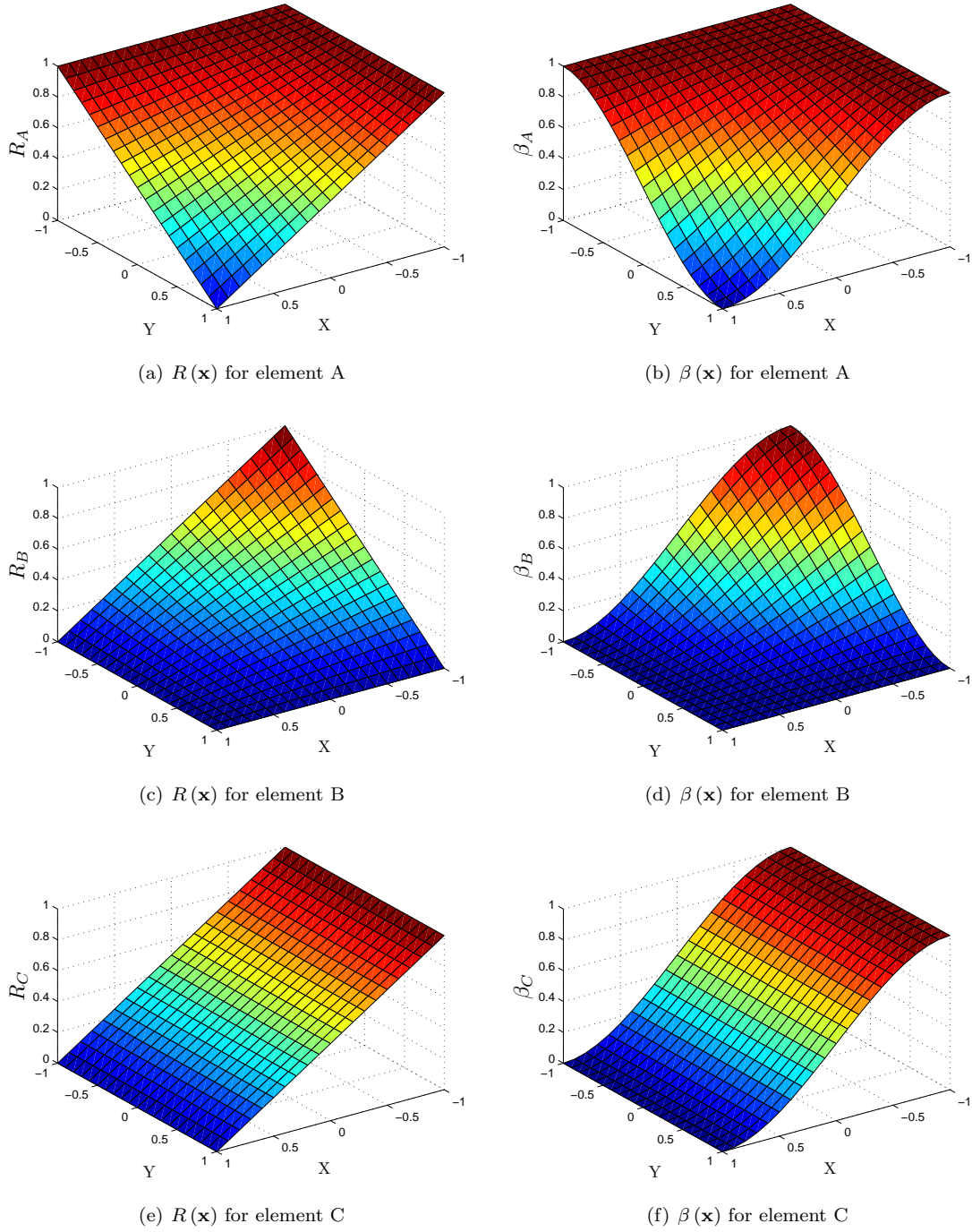


Figure 5.4: Sample two-dimensional ramp and blending functions

5.2.2 FE-EFGM for one- and three-dimensional problems

Reduction of the two-dimensional coupled FE-EFGM formulation to one-dimension or extension to three-dimensions is very straightforward. Appropriate one- and three-dimensional shape functions should be used within the FE and the EFG regions for one- and three-dimensional problems respectively. In this research, in the case of one-dimensional problems, two-node linear elements are used in the FE and interface regions, while a two node background cells are used in the EFG region. The two node FE in the local coordinate ξ is shown in Figure 5.5, for which the shape functions in the local coordinates are

given as [204]

$$N_i = \frac{1}{2} (1 + \xi_i \xi) \quad (5.16)$$

where i is the node number, ξ_i are nodal coordinates of the parent element. In three-dimensional problems eight-node hexahedral elements are used in the FE and interface regions and the same hexahedral background cells are used in the EFG region. An eight node hexahedral element is shown in Figure 5.6 in the natural coordinates (ξ, η, ψ) , the shape functions for which are given as [283]

$$N_i = \frac{1}{8} (1 + \xi_i \xi) (1 + \eta_i \eta) (1 + \psi_i \psi) \quad (5.17)$$

where again i is the node number and ξ_i , η_i and ψ_i are the nodal coordinates of the parent element.

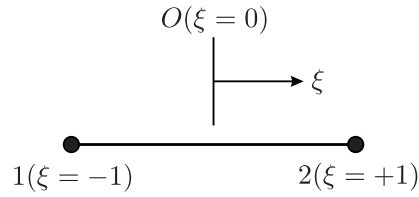


Figure 5.5: Two node linear element in natural coordinates

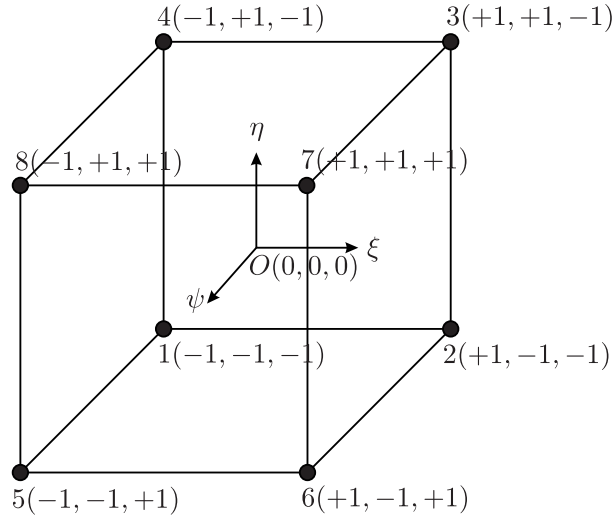


Figure 5.6: Eight node iso-parametric hexahedral element in natural coordinates

5.3 FE-EFGM coupling using max-ent shape functions

Max-ent shape functions provide a natural way to couple the FEM and the EFGM without using interface elements or transition regions between the FE and EFG zones because of the weak Kronecker delta property at the boundaries when using these shape functions. A sample mixed FE and EFG discretization is shown in Figure 5.7, where Ω_E and Ω_F are the EFG and the FE regions and Γ is the boundary between these two regions. The nodes on the boundary Γ between the EFG and FE regions, shown in green in Figure 5.7, are used in the displacement approximation for both the EFG and the FE regions.

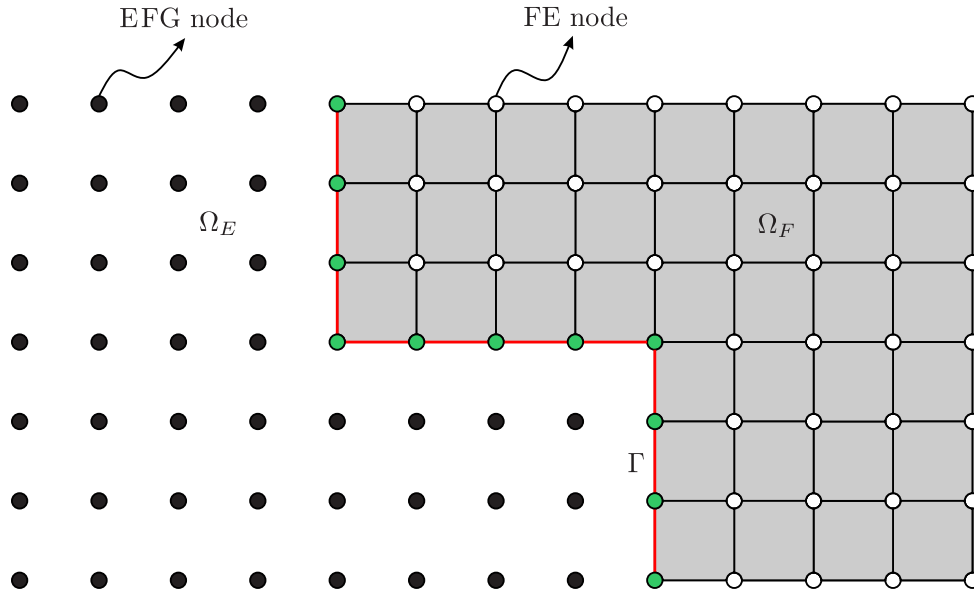


Figure 5.7: FE-EFGM coupling using max-ent shape functions

5.4 Numerical examples

One-, two- and three-dimensional linear elastic numerical examples are now given to demonstrate the implementation and performance of the proposed coupled FE-EFGM formulations. The results from the proposed approach are also compared with the conventional FE-EFGM coupling with the interface elements [40] and with other alternatives.

5.4.1 One-dimensional bar

The same one-dimensional bar is analysed in this section for which the geometry, material properties and analytical solution for the displacement and stress field are given in §2.8. Initially, the problem is solved using the conventional FE-EFGM coupling, i.e. using MLS shape functions within the EFG region and using a transition or interface region between the EFG and the FE regions (MLS with interface). The problem discretizations with four FE, an interface element and five EFG background cells are shown in Figure 5.8(a). The same problem is also solved with the proposed FE-EFGM coupling, i.e. using max-ent shape functions within the EFG region and without the interface element between the FE and the EFG regions (max-ent without interface). The problem discretization for the proposed coupling with four FE and six EFG background cells is shown in Figure 5.8(b). For comparison the same problem is also solved with the same discretization as shown in Figure 5.8(b) but using MLS shape functions within the EFG regions (MLS without interface). In all three cases, the essential boundary conditions are implemented directly because of the use of the FE region on the essential boundary side.

Initially, the problem is solved with $d_{max} = 2.5$ in the EFG region. The comparison between the nodal displacements along the length of the bar in all the three cases and the analytical solution is shown in Figure 5.9(a). The nodal displacements coincide and there is no apparent difference. The same problem is further analysed with the same three types of discretization but this time using $d_{max} = 3.5$. In this case the comparison between the nodal displacements along the length of the bar in all three cases, and the analytical solution are shown in Figure 5.9(b). This time the results for the nodal displacements of the two cases, i.e. MLS with interface and max-ent without interface are in a very good agreement with the analytical solution. The deviation of the nodal displacements from the analytical solution for the

MLS without interface can be clearly seen in Figure 5.9(b). This shows the importance of the interface element, when using MLS in the EFG region of the problem.

The dependence of the results on d_{max} is further analysed using plots of the shape functions and shape function derivatives for all three cases. Plots for the shape functions and shape function derivatives for the MLS with interface elements with $d_{max} = 2.5$ are shown in Figures 5.10(a) and 5.10(c) respectively, and for $d_{max} = 3.5$ are shown in Figures 5.10(b) and 5.10(d) respectively. It is clear from these plots that increasing d_{max} increases the number of nodes influencing the zone between the EFG and FE regions. In this case, the interface element creates a smooth blending between the EFG and the FE regions, i.e. there is no discontinuity in the shape functions and shape functions derivative on Γ_E and Γ_F . A slight jump can be observed in the shape functions and shape function derivatives on Γ_E , which is the point where the shape functions change from the MLS shape functions to the interface element shape functions. For the max-ent without interface, plots for the shape functions and shape function derivatives for $d_{max} = 2.5$ are shown in Figures 5.11(a) and 5.11(c) respectively, and for $d_{max} = 3.5$ in Figures 5.11(b) and 5.11(d) respectively. In this case due to the weak Kronecker delta property of the max-ent shape functions, all the shape functions and shape function derivatives naturally blend together very smoothly on the boundary node between the EFG and the FE region. The boundary node is used for the shape functions' calculations on both the EFG and the FE region and the shape functions of all the internal nodes on the EFG side goes to zero at this node. For the MLS without interface, plots for the shape function and derivatives for $d_{max} = 2.5$ are shown in Figures 5.12(a) and 5.12(c) respectively, and for $d_{max} = 3.5$ in Figures 5.12(b) and 5.12(d) respectively. A slight discontinuity in the shape functions and shape function derivatives on the boundary nodes can be seen for $d_{max} = 2.5$, which may not be sufficient to produce a significant deviation in the nodal displacements from the analytical solution as shown in Figure 5.9(a). A very clear discontinuity in the shape functions and shape function derivatives can be seen on the boundary node between the EFG and the FE region for $d_{max} = 3.5$, which produces a clear deviation in the nodal displacements from the analytical solutions as shown in Figure 5.9(b). This clearly shows the need for interface elements between the FE and the EFG region, when using the MLS shape functions within the EFG regions.

To study the convergence properties of the methods for the same one-dimensional bar problem, the error in energy norm $\|e\|$ [332] is used (as described in §2.8). For this purpose, for the max-ent without interface and the MLS without interface cases, half of the problem is discretized with the FE region, and the other half is discretized with the EFG region. For the MLS with interface case, half of the problem is discretized with the FE region, and the other half is discretized with the combine interface and the EFG regions. The number of nodes used for analysis in the consecutive discretizations are 11, 13, 15, 17, 19 and 21. $d_{max} = 2.5$ is used in this case. The convergence plots for the three types of discretizations are shown in Figure 5.13, in which curves for the MLS with interface and max-ent without interface coincide with each other with almost the same rate of convergence. The MLS without interface performs very poorly with high error and very low rate of convergence as compared to the other two cases. For comparison, curves are also given in the same figure for the cases when the same problem is solved only with the FEM and only with the EFGM with the MLS and the max-ent shape functions. It is clear from Figure 5.13 that the coupled FE-EFGM performs somewhere between the pure FEM and the pure EFGM. The error in energy norm and rate of convergence for the coupled FE-EFGM also lies between the pure FEM and the pure EFGM cases. For the MLS without interface case, although at the start, the error in energy norm lies between the pure FEM and the pure EFGM cases, the rate of convergence is lower than the pure FEM case.

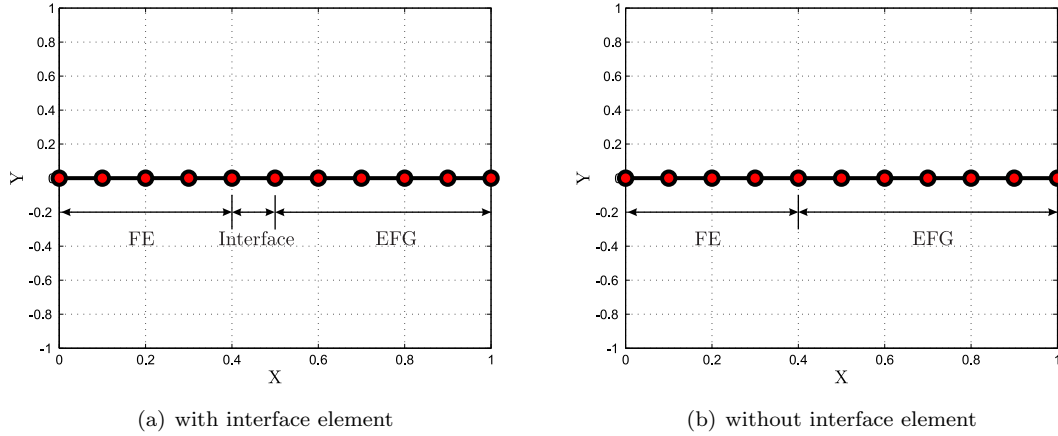


Figure 5.8: Discretizations for the one-dimensional bar problem

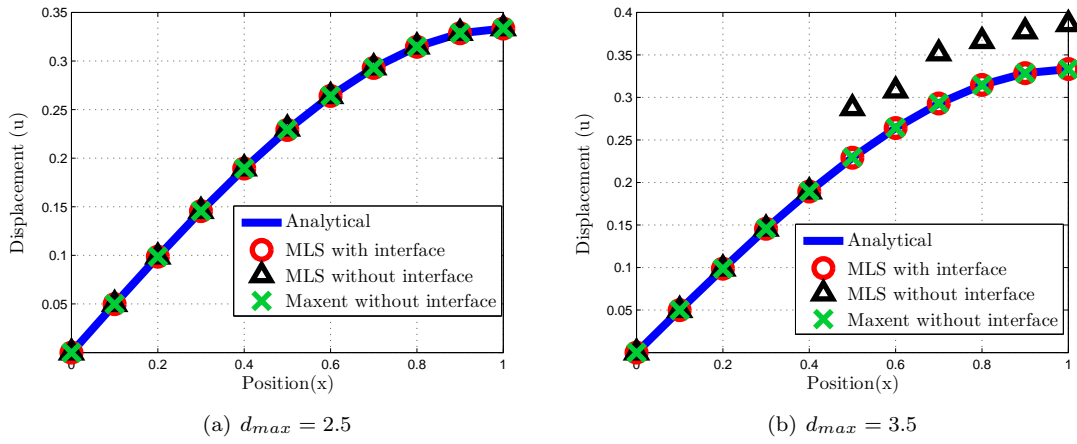


Figure 5.9: Displacements for the one-dimensional bar problem

5.4.2 Two-dimensional cantilever beam

The second problem modelled with the proposed coupled FE-EFGM formulation is the same two-dimensional beam problem as given in §2.8. The geometry, material properties and analytical solutions for displacements and stresses are also given in the same section. The problem is solved with the three different discretizations, i.e. the MLS with interface, the max-ent without interface and the MLS without interface. The problem is discretized with 189 (21×9) nodes and 160 (20×8) background cells. The problem discretizations are shown in Figures 5.14(a) and 5.14(b) with interface elements and without interface elements respectively. In Figure 5.14(a), the first five columns are the FE, and the sixth one consists of interface elements, while the rest are the EFG background cells. In Figure 5.14(b) the first five columns are the FE while the rest are the EFG background cells. In all three cases, essential boundary conditions are implemented directly due to the use of the FE on the essential boundary side.

Initially, the problem is solved with $d_{max} = 2.5$ and comparison of nodal displacement at $y = 0$, and stresses σ_{xx} and σ_{xy} at $x = L/2$ to the analytical solutions are shown in Figures 5.15(a), 5.15(b) and 5.15(c) respectively. In this case all the results are in a very close agreement with the analytical solutions. Next, the same problem is solved with $d_{max} = 3.5$ and the same comparison of the nodal displacement and stresses to the analytical solutions are shown in Figures 5.16(a), 5.16(b) and 5.16(c) respectively. A very clear difference between the numerical and analytical results in the case of the MLS

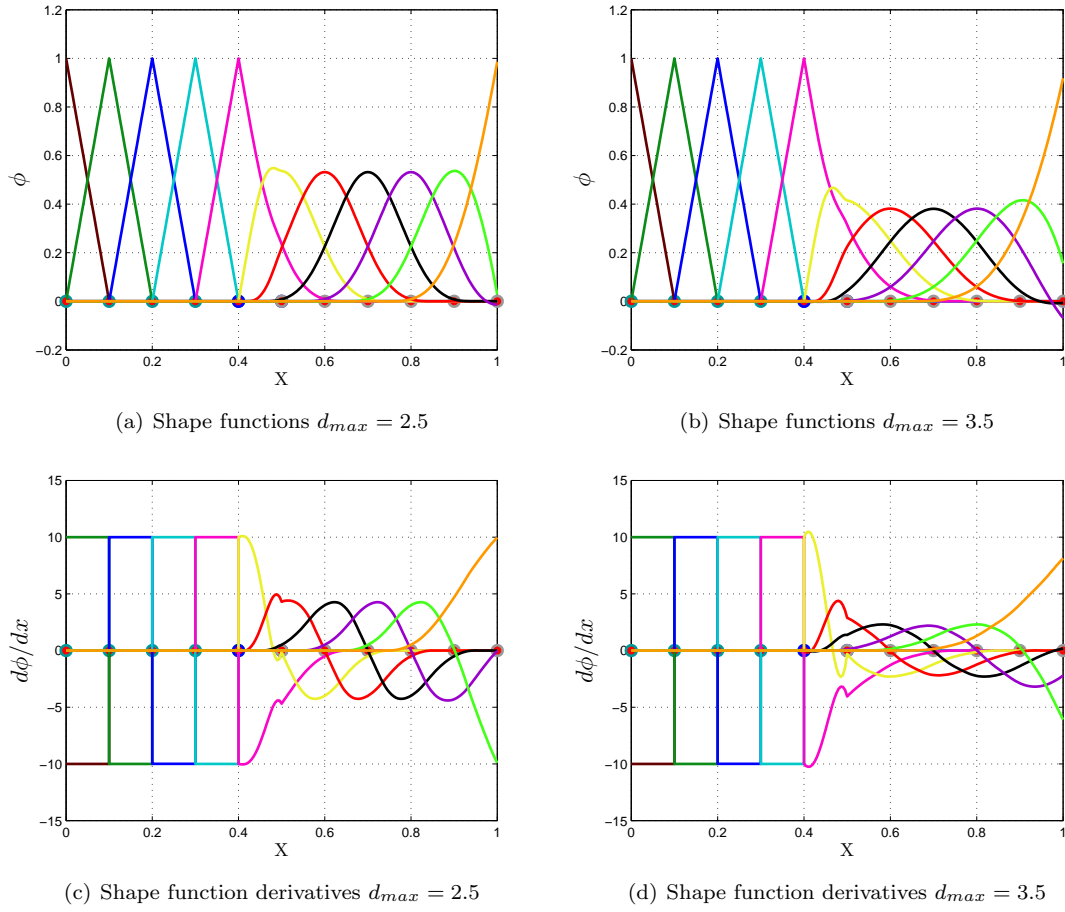


Figure 5.10: Shape functions and shape function derivatives for the one-dimensional bar problem coupling using MLS with interface elements

without interface case can be seen in these plots. The results in the case of the MLS with interface and the max-ent without interface are again in a very close agreement with the analytical solutions. This once again verifies that there is no need for the interface elements in the coupled FE-EFGM formulation while using the max-ent shape functions within the EFG region.

To study convergence for the same problem once again the error in energy norm $\|e\|$ is used. In this case 27 (9×3), 52 (13×4), and 85 (17×5) nodes are used in the consecutive discretizations. For the max-ent without interface and the MLS without interface cases, half of the beam is modelled with the FEM, and the other half is modelled with the EFGM. For the MLS with interface case, half of the problem is modelled with the FEM, and the other half is discretized with the combine interface elements and the EFG background cells. The convergence plots for $d_{max} = 2.5$ are shown in Figure 5.17(a) in which all three curves are in very close agreement with almost the same rate of convergence. The same plots for $d_{max} = 3.5$ are shown in Figure 5.17(b), in which the curves for the MLS with interface, and the max-ent without interface are in very close agreement with almost the same rate of convergence, but the error is very high in the case of the MLS without interface case. This once again proves the importance of the interface elements (or transition regions) in the case when using MLS shape functions within the EFG regions.

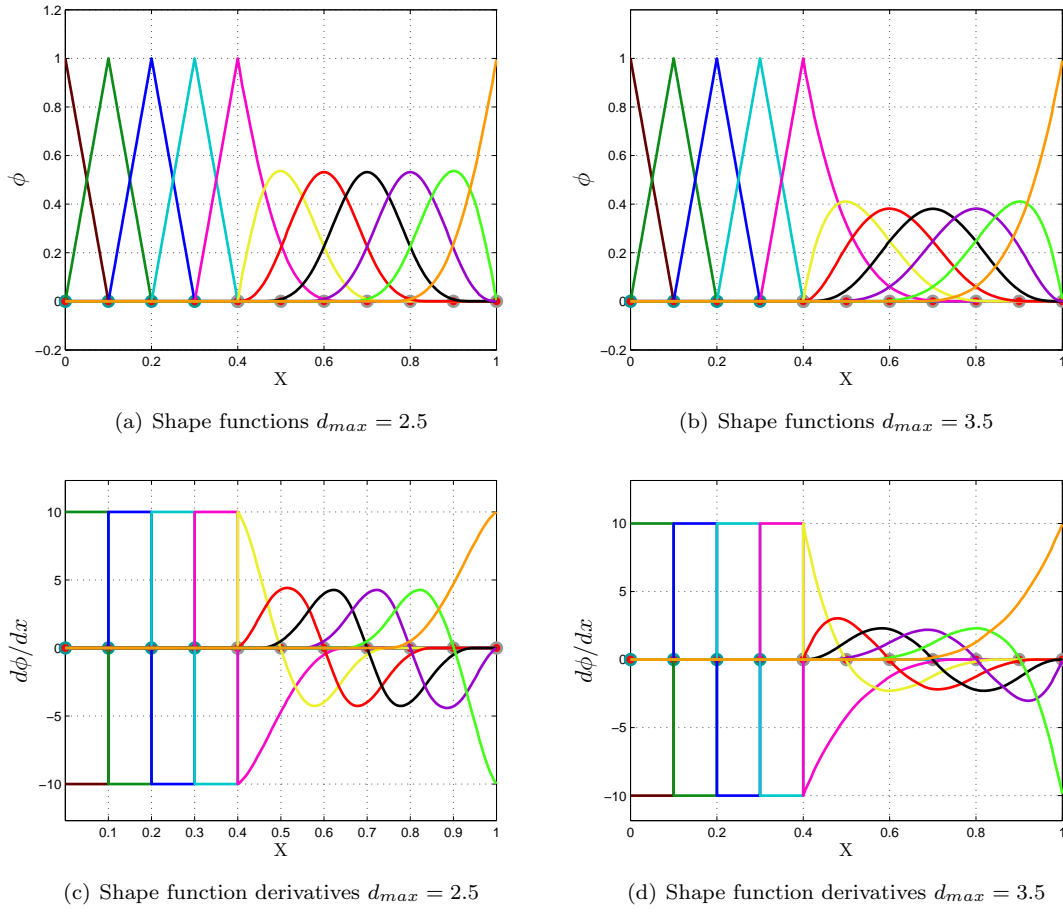


Figure 5.11: Shape functions and shape function derivatives for the one-dimensional bar problem coupling using max-ent without interface elements

5.4.3 Flexible strip footing

The third numerical example is an elastic plane strain flexible strip footing to demonstrate the performance of the proposed coupled FE-EFGM formulation. The same problem is also solved by hybrid MLPG and scaled boundary method in [82], the exact solution for which is given in [252]. The exact solution for the relative displacement between the centre of the footing and any point on the surface a horizontal distance x from the centre is

$$\Delta u_y = \frac{2q(1-\nu^2)}{\pi E} \{ (x-R) \ln |x-R| - (x+R) \ln |x+R| + 2R \ln R \}, \quad (5.18)$$

where $2R$ is the length of footing, q is the uniformly distributed load on footing, E is the modulus of elasticity and ν is Poisson's ratio. In this case, a flexible footing of half length $R = 0.6$ is loaded over soil with an applied pressure $q = 1000$ as shown in Figure 5.18. Due to symmetry only one-half of the problem, shown in grey in Figure 5.18, is modelled with appropriate boundary conditions. The coupled FE-EFGM discretization of the problem with and without an interface region are shown in Figures 5.19(a) and 5.19(b) respectively. In Figure 5.19(a), the region shown in gray from $y = 7.2$ to $y = 8$ and from $x = 0$ to $x = 0.8$ is the EFG region, followed by a strip of interface elements shown in orange and the rest are the FE region. In Figure 5.19(b), the region shown in gray from $y = 7.2$ to $y = 8$ is the EFG region and the rest are the FE region.

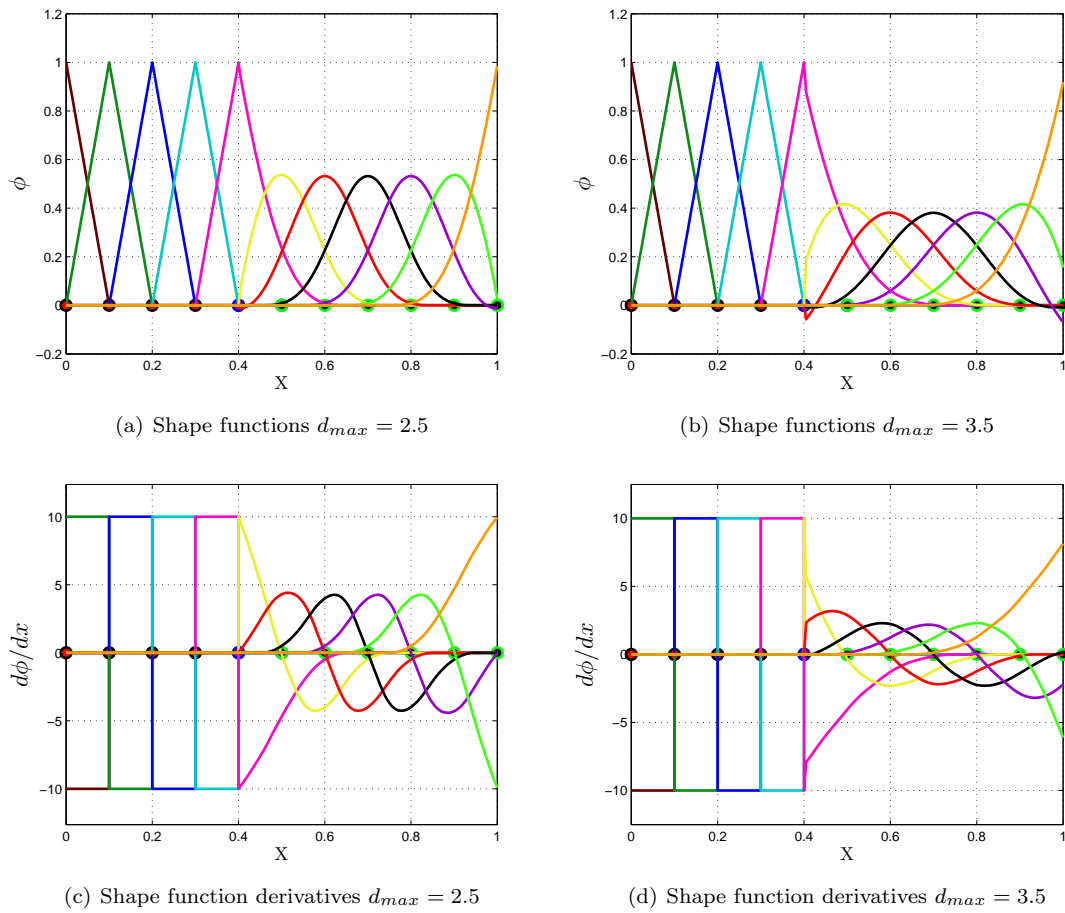


Figure 5.12: Shape functions and shape function derivatives for the one-dimensional bar problem coupling using MLS without interface elements

Initially, the problem is solved with the MLS with interface using the discretization shown in Figure 5.19(a). The imposition of the essential boundary conditions on the right and bottom edge is straightforward because of the use FE on these boundaries, while on the left edge it is complicated due to the presence of both the FE and the EFG regions. On the left edge, Lagrange multipliers are used to impose the essential boundary conditions for the first six nodes from the top, while the direct method is used for the rest of the nodes. The discretization shown in Figure 5.19(b), is used for the MLS without interface and max-ent without interface formulations in which again for the MLS without interface case, Lagrange multipliers are used to impose the essential boundary conditions for the first five nodes from the top on the left edge, while the direct method is used for the rest of the boundary nodes. In the case of max-ent without interface, all the essential boundary conditions are implemented directly. The problem is first solved with $d_{max} = 2.5$ for all three cases and a comparison of the surface deflection with the analytical solution is given in Figure 5.20(a). Here all the numerical results are in a close agreement with the analytical solution. The same problem is again solved with the same parameters but changing d_{max} to 3.8 and a comparison between the numerical surface deflection in all three cases and the analytical solution is given in Figure 5.20(b). Once again, the results from the MLS with interface and max-ent without interface are in a very close agreement with each other and the exact solution, but a very clear difference can be seen in the results for the MLS without interface case, especially near the centre of the footing. Contour plots for the vertical displacements (u_y) at $d_{max} = 3.8$ for a region of the problem

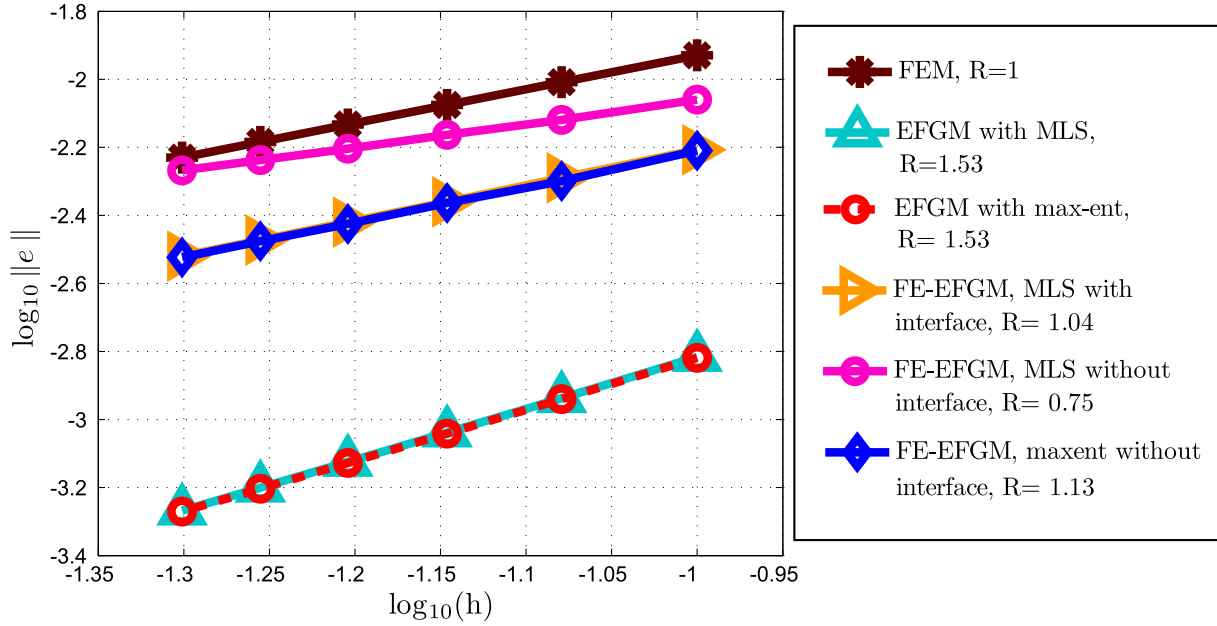


Figure 5.13: Convergence plots for different discretizations for the one-dimensional bar problem

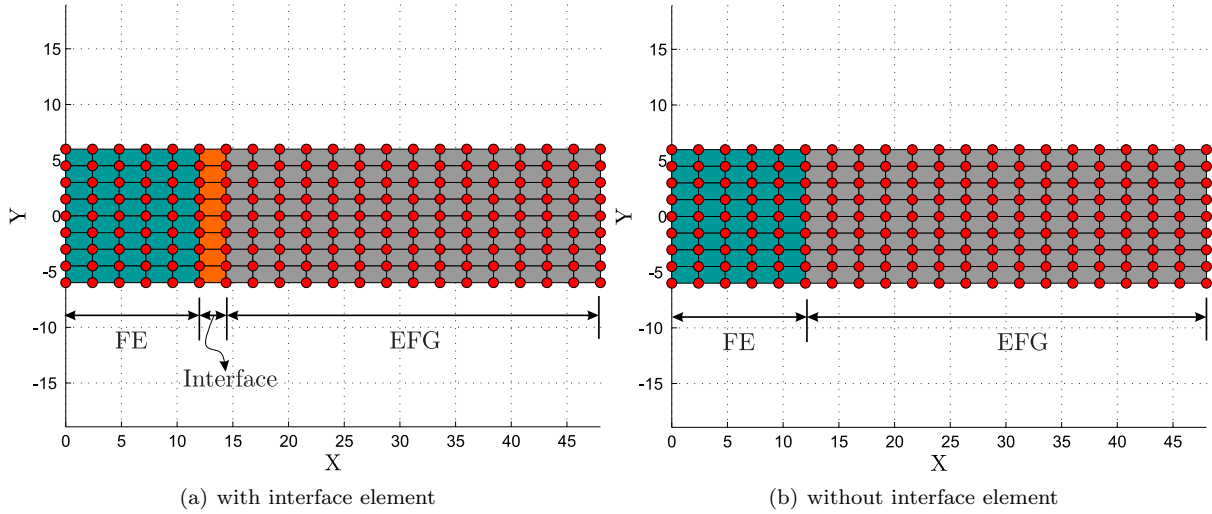
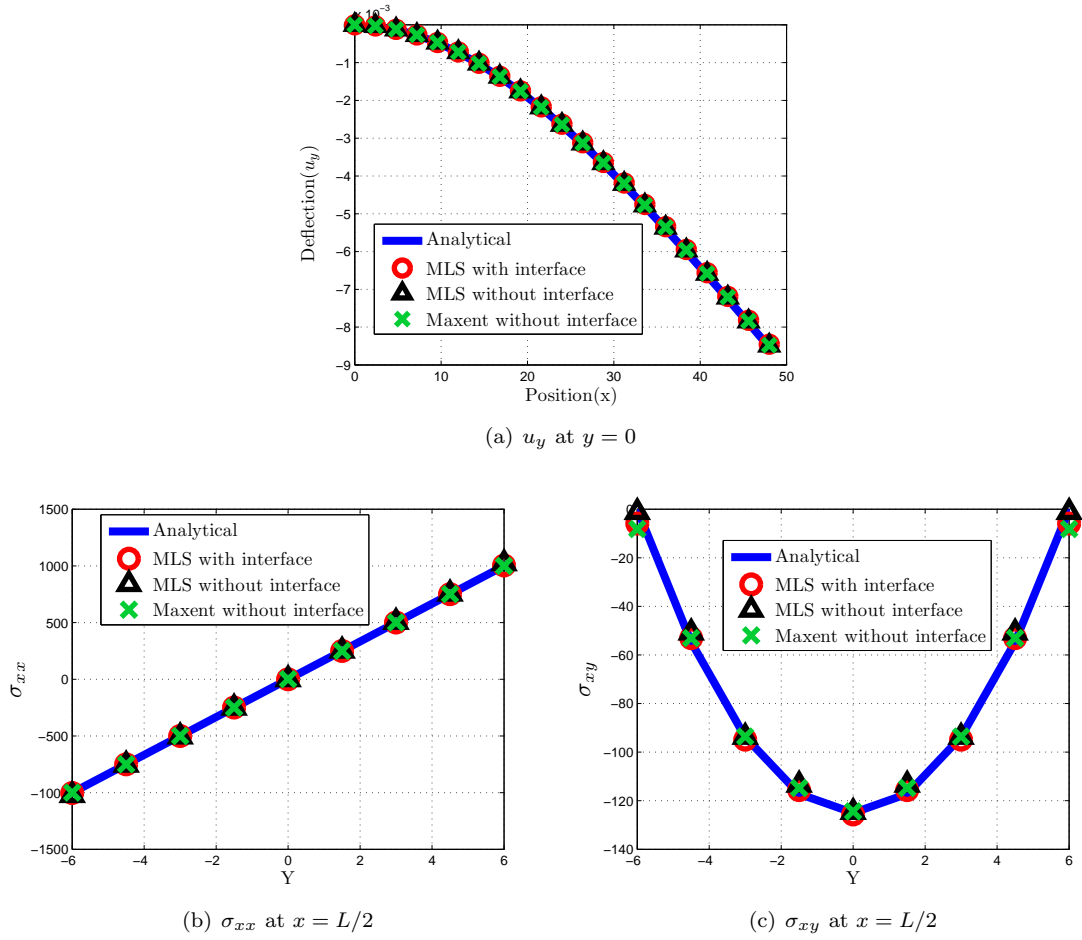


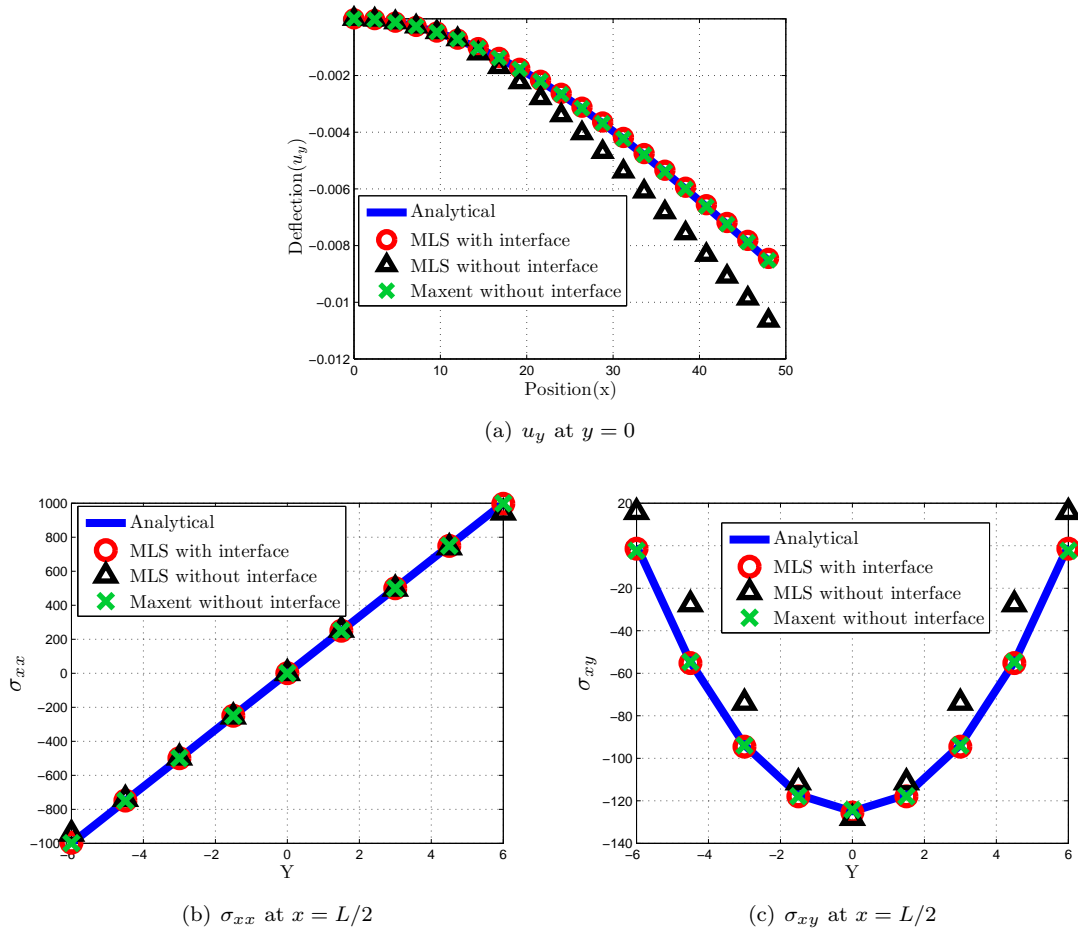
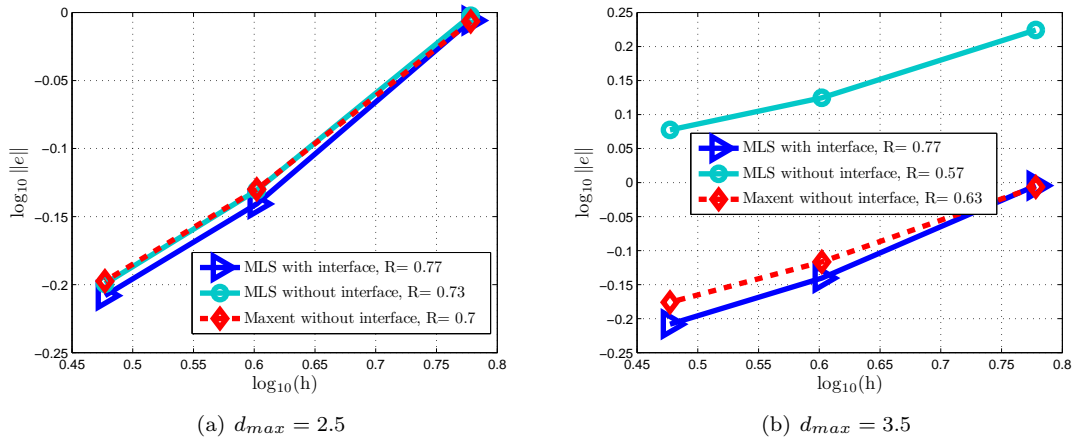
Figure 5.14: Discretizations for the two-dimensional beam problem

domain near the footing with dimensions (2×2) are also given in Figures 5.21(a), 5.21(b) and 5.21(c) for the MLS with interface, the MLS without interface and the max-ent with interface cases respectively. The u_y contours are very smooth throughout the regions in the MLS with interface and the max-ent without interface cases, while the results in the case of the MLS without interface are disappointing. The curves in this case are clearly in error, especially near the centre of the footing. The results once again verify that there is no need for a transition region between the FE and the EFG regions, when using max-ent shape functions within the EFG region, and on the other hand, also verifies the fact that the transition region is unavoidable in cases when using MLS within in the EFG region.

Figure 5.15: Displacements and stresses for the two-dimensional beam problem at $d_{max} = 2.5$

5.4.4 Three-dimensional cantilever beam problem with end load

The fourth numerical example is a three-dimensional cantilever beam subjected to end load, for which the geometry, boundary conditions and material properties are given in §2.8. The analytical solution for the displacement field is not available for this problem, so the comparison is made with the two-dimensional cantilever beam problem, given in the same section. The problem is initially solved with different discretizations as shown in Figure 5.22, in which MLS shape functions are used within the EFG regions. The pure FE and EFG discretizations are shown in Figures 5.22(a) and 5.22(b) respectively, while the coupled EFGM-FE and FE-EFGM discretizations are shown in Figures 5.22(c) and 5.22(d) respectively. In the coupled EFGM-FE and FE-EFGM half of the beam is discretized with the FE, while the interface elements and the EFG background cells are used in the remaining half. In EFGM-FE and FE-EFGM, the EFG and the FE regions are used respectively on the essential boundary side of the beam. Lagrange multipliers are used to impose the essential boundary conditions in the EFGM and the EFGM-FE cases, while in the FEM and the FE-EFGM cases essential boundary conditions are implemented directly due to the use of the FE on the essential boundary side. Comparison of neutral axis displacements between the four cases in Figure 5.22 and the analytical solution is shown in Figure 5.24(a). Excellent agreement with the analytical solution can be seen in the EFGM and the EFGM-FE cases, while in the FEM and the FE-EFGM cases, response are relatively stiff. The excellent agreement in the EFGM and the EFGM-FE cases are due to the use of the EFGM near the essential boundary, while

Figure 5.16: Displacements and stresses for the two-dimensional beam problem at $d_{max} = 3.5$ Figure 5.17: Convergence plots for different discretizations and at different d_{max} for the two-dimensional beam problem

in the FEM and FE-EFGM, the relatively stiff response is due to the use of linear eight node elements on the essential boundary side, which are unable to capture accurately the cubic profile of the neutral axis displacement.

The same problem is next solved by different FEM, EFGM and coupled FE-EFGM discretizations

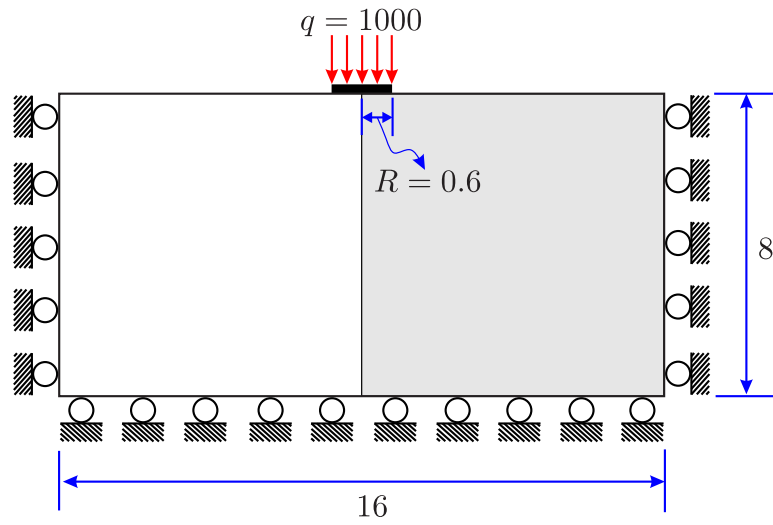


Figure 5.18: Geometry, boundary condition and loading for the flexible strip footing problem

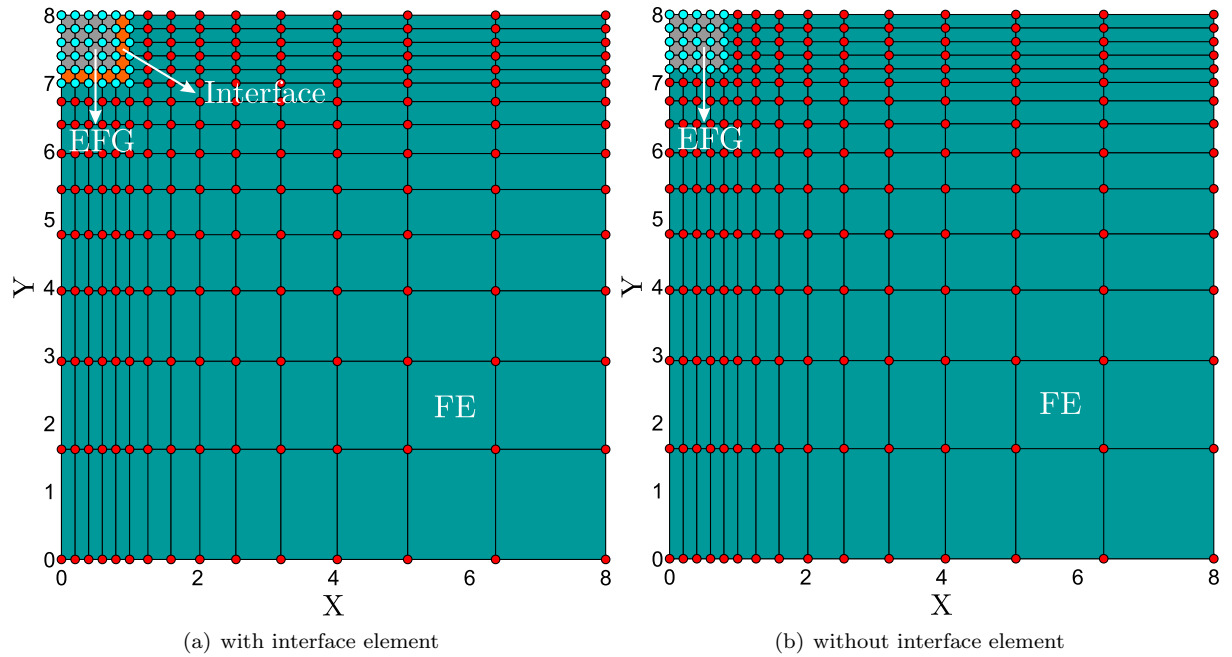


Figure 5.19: Discretizations for the flexible strip footing problem

with max-ent shape functions in the EFG regions. The pure FEM and the EFGM discretizations are the same as shown in Figures 5.22(a) and 5.22(b) respectively, while the coupled EFGM-FE and FE-EFGM discretizations are shown in Figures 5.23(a) and 5.23(b) respectively. In this case, half of the problem is discretized with the FE, and the other half with the EFG background cells. In Figure 5.23(a), the EFG region is used on the essential boundary side, while in Figure 5.23(b) the FE region is used on the essential boundary. This time all the essential boundary conditions are implemented directly due to the use of the max-ent shape functions in the EFG region. Comparisons of the neutral axis displacements with the analytical solution are shown in Figure 5.24(b), in which once again, excellent agreement between the EFGM, EFGM-FE and the analytical solution can be seen. A relatively stiff response is also shown in the case of the pure FE and the FE-EFGM due to the use of the FE on the essential boundary side of the problem.

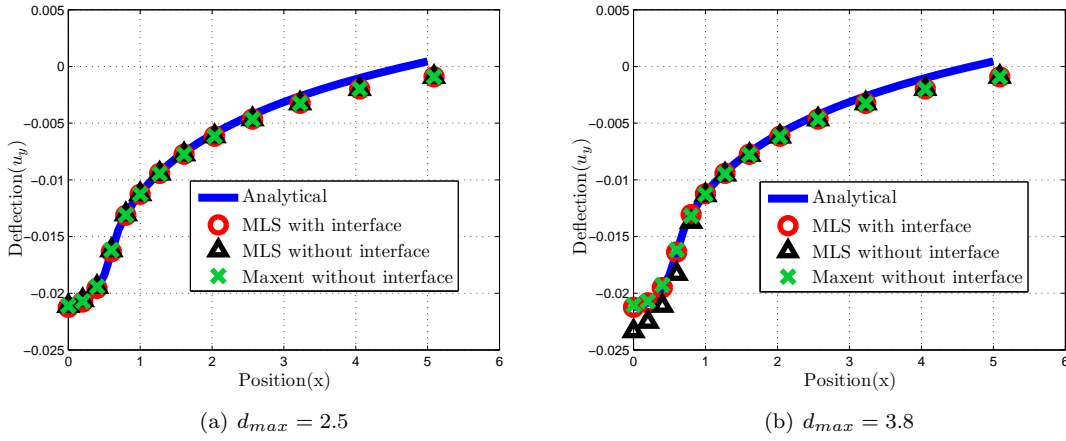
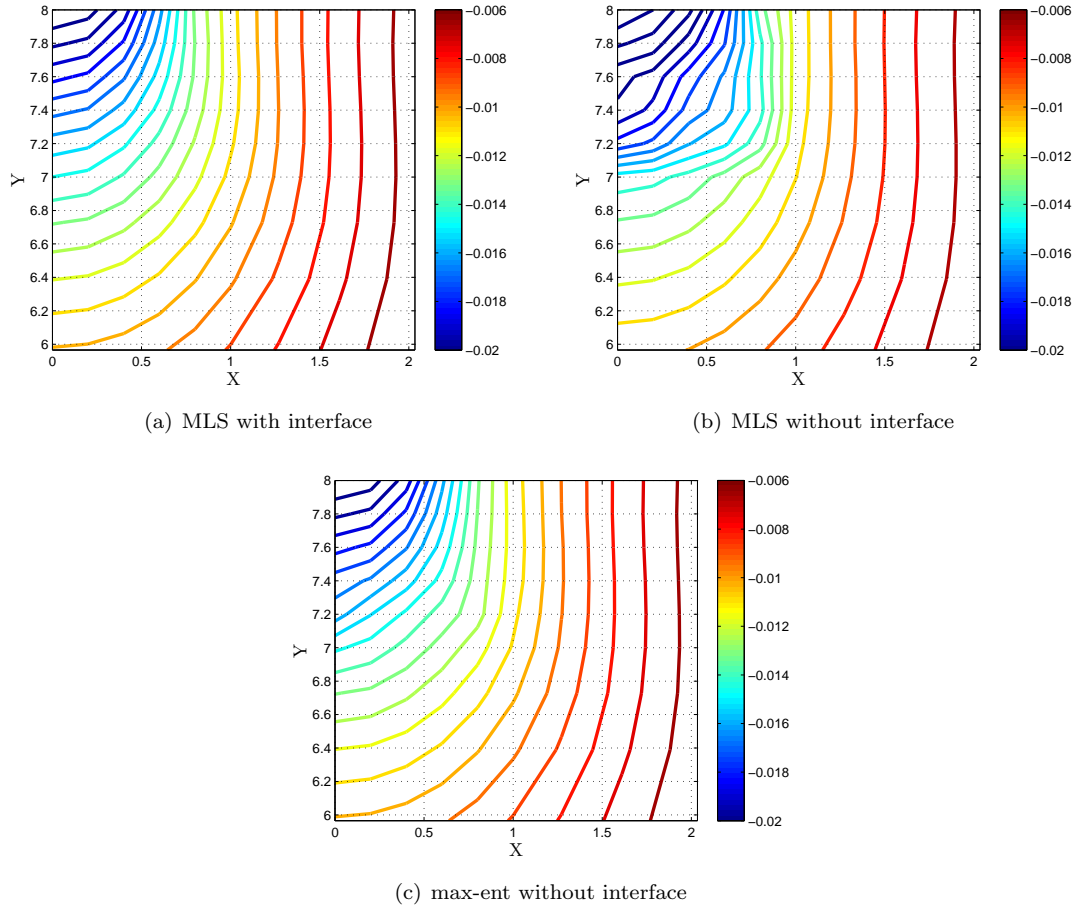


Figure 5.20: Surface deflection for different discretization for the flexible strip footing problem

Figure 5.21: Displacement (u_y) contours at $d_{max} = 3.8$ for the flexible strip footing problem

5.5 Extension to nonlinear problems

Having demonstrated above the accuracy of the proposed approach using coupled FE with max-ent based EFG regions, the next step is to move to nonlinear problems. In this section, the proposed coupled FE-EFGM approach is extended to geometrically nonlinear problems, which is very straightforward. For linear elastic or small strain problems, the FE or the EFG regions can be identified based on the problem

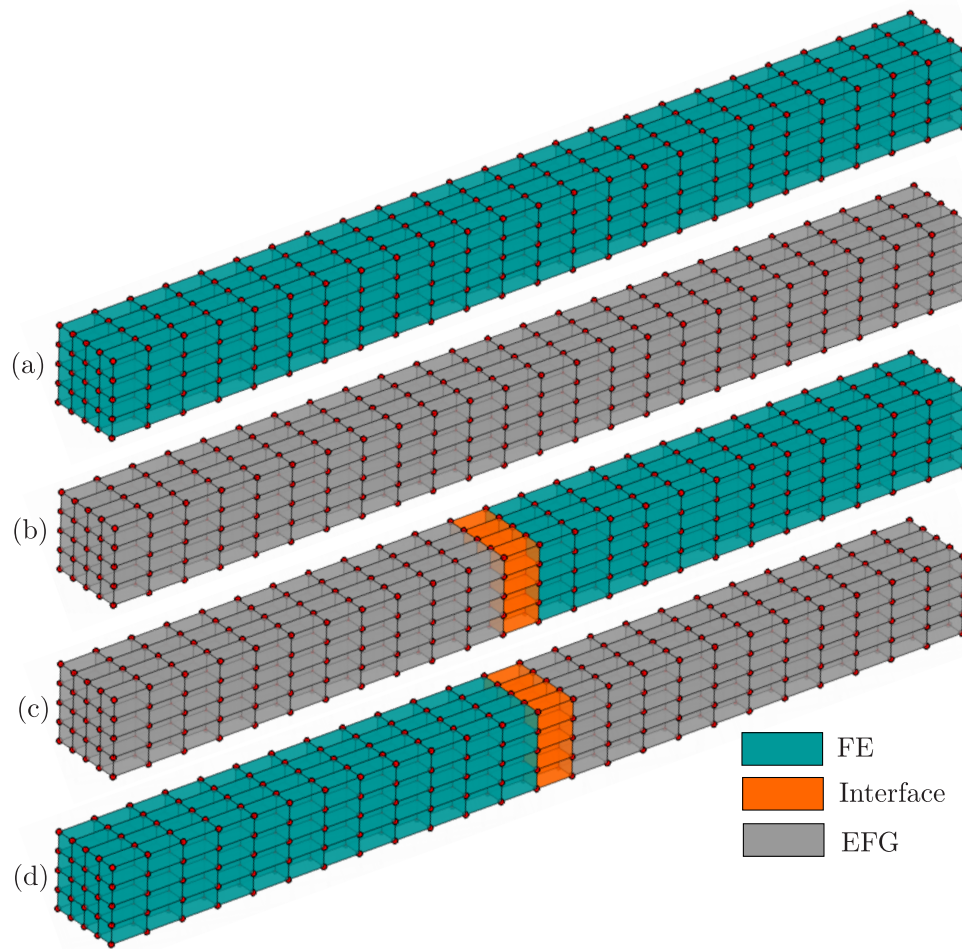


Figure 5.22: Three-dimensional beam problem discretizations with interface region (a) FE; (b) EFGM; (c) EFGM-FE; (d) FE-EFGM

dimensions, which is not applicable in case of large strain or finite deformation problems, as the geometry is changing during the solution. In these problems, it is necessary to attach a tag to each integration (Gauss) point to identify the FE or the EFG regions in the problem domain to which it belongs. In this case, the Updated Lagrangian formulation is used to model finite deformation, and due to the use of the max-ent shape function in the EFG region, all the essential boundary conditions are implemented directly.

5.5.1 Numerical examples

Two numerical examples are now given in this section to demonstrate the implementation and performance of the current coupled FE-EFGM approach with geometrical nonlinearity.

Infinite plate strip

The first geometrically nonlinear problem solved by the coupled FE-EFGM formulation is an infinite plate strip subjected to uniformly distributed load and with simple supported edges. The geometry, boundary conditions and material properties for this problem are given in §3.4. The analytical solution for the neutral axis deflected profile and pressure versus displacement are also given in the same section. Due to symmetry only half of the problem is modelled and the detail of the imposition of essential boundary

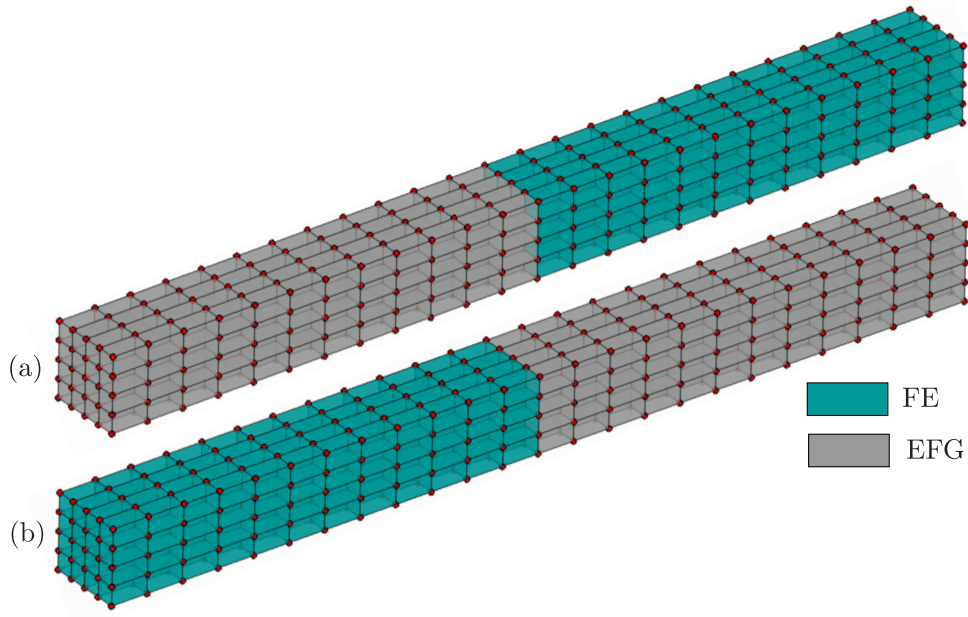


Figure 5.23: Three-dimensional beam problem discretizations without interface region (a) EFGM-FE; (b) FE-EFGM

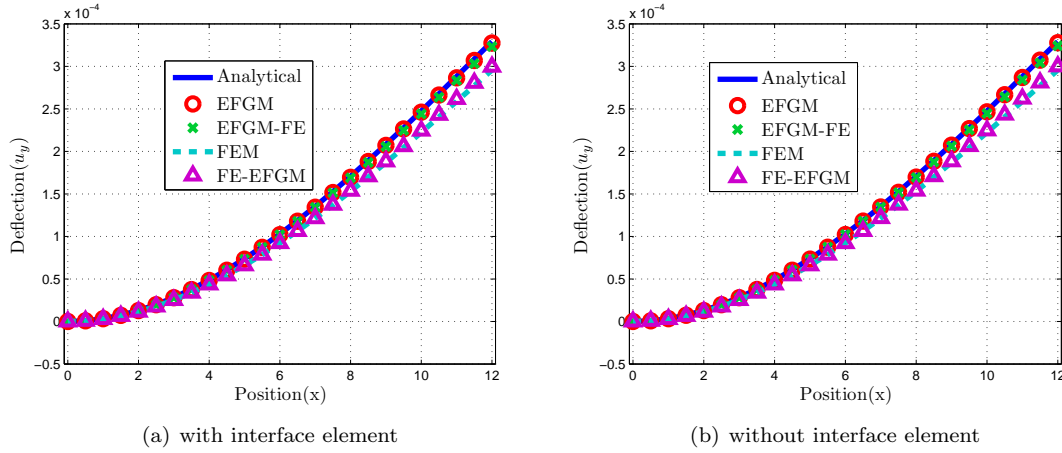


Figure 5.24: Three-dimensional beam problem neutral axis deflection u_y

conditions are shown in §3.4. In this case half of the remaining problem domain is discretized with the 40 (20×2) EFG background cells and the remaining half with the 40 (20×2) FE, while the total number of nodes used is 123 (41×3). The coupled FE-EFGM undeformed configuration is shown in Figure 5.25(a), while the deformed configuration for the same problem at the end of analysis is also shown in the same figure. A comparison between numerical and analytical applied pressure versus the central displacement of the neutral axis normalized over the thickness of the plate is shown in Figure 5.25(b), showing excellent agreement. Furthermore, at the end of analysis a comparison between the numerical and analytical solution for the deflected profile normalized over the thickness of the plate is shown in Figure 5.25(c), again with excellent agreement. These results demonstrates that the proposed FE-EFGM approach can accurately capture the nonlinear behaviour of the plate.

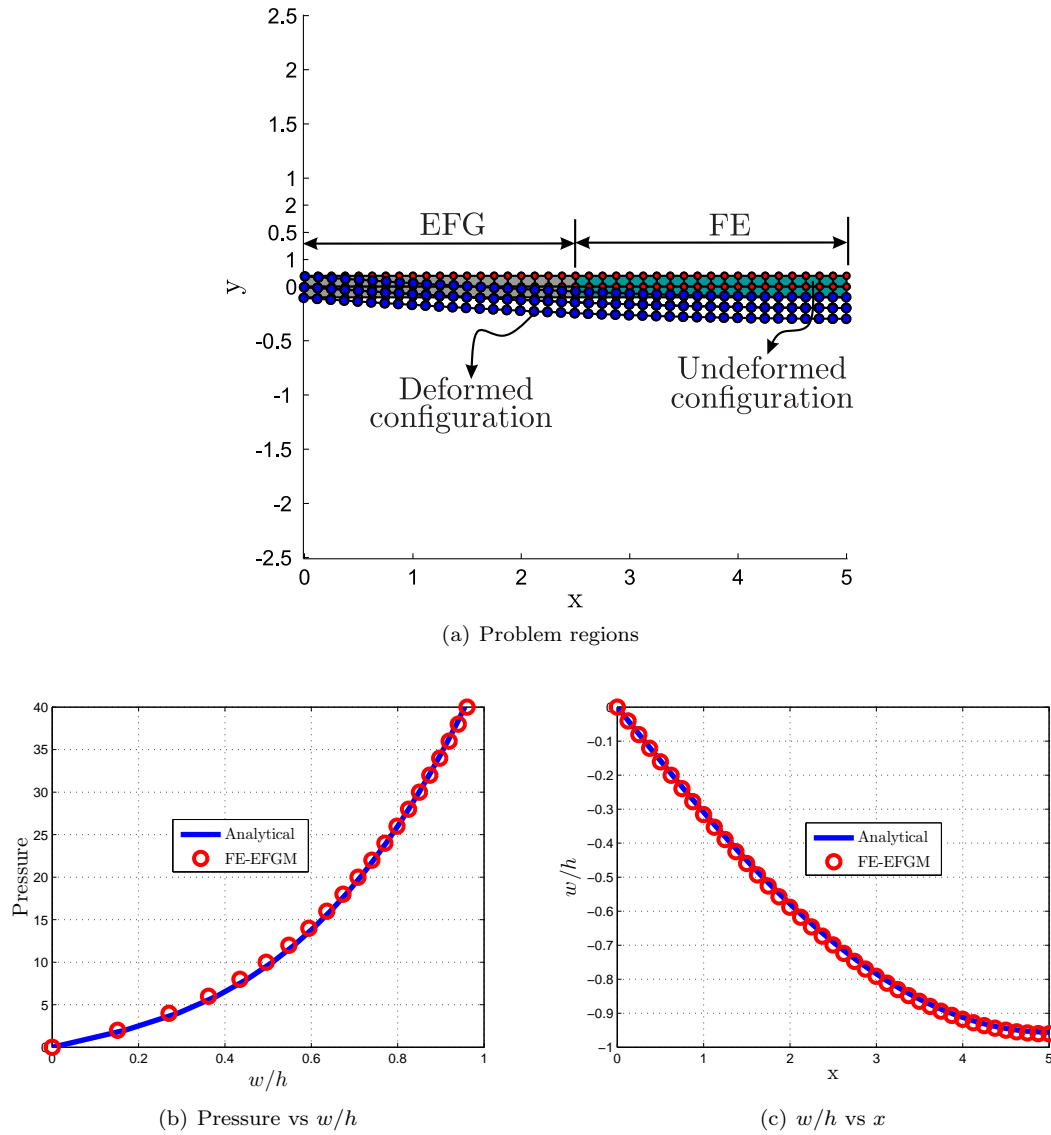


Figure 5.25: Problem regions and results for the nonlinear infinite plate strip

Elastic cantilever beam

The second geometrically nonlinear problem is an elastic cantilever beam, subjected to uniformly distributed load. The geometry, boundary conditions and material properties are the same as given in §3.4. Half of the beam is modelled with the 20 (10×2) EFG background cells, and the other half with 20 (10×2) FE. An undeformed hybrid FE-EFGM discretization is shown in Figure 5.26(a) with 42 (21×2) nodes. The final deformed configuration for this problem is also shown in Figure 5.26(a), which shows very large deformation. A comparison is also shown between the numerical and the reference solution [221] for the pressure versus u_x and u_y normalized over the length of the beam in Figure 5.26(b) and 5.26(c) respectively. Very good agreement can be seen in these plots between the analytical and reference solution, which once again shows that the proposed coupled FE-EFGM approach can be used for nonlinear problems, even with very large deformation.

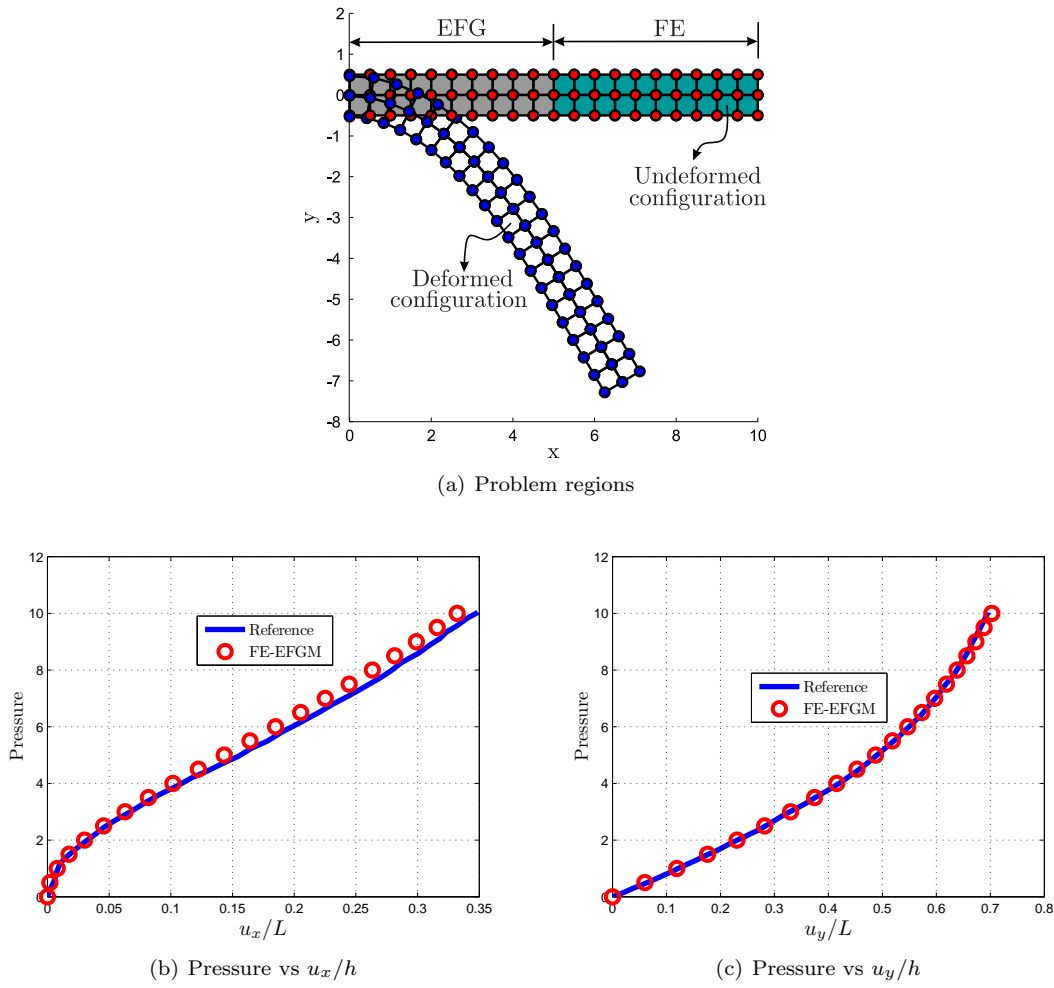


Figure 5.26: Problem regions and results for the nonlinear elastic cantilever beam

5.6 Concluding remarks

A new method of coupling the FEM with the EFGM based on the local max-ent shape functions is proposed in this chapter. These shape functions possess a weak Kronecker delta property at the boundaries, which provide a natural way to couple both the regions without the use of the transition or interface region. Results from the proposed coupled FE-EFGM are also compared with the corresponding results from the conventional FE-EFGM coupling, in which the MLS shape functions are used within the EFG region and interface elements are used between the FE and the EFG regions. A comparison is also made with the case, in which the MLS shape functions are used within the EFG region but with no interface region between the FE and the EFG regions is provided. Very good agreement between the analytical and numerical solution is observed in all three cases at lower domains of influence, as the MLS shape functions approaches the FE shape functions at lower influence domains. However, the performance of the MLS without interface case deteriorates at larger influence domains, as the MLS shape functions will even lose the approximate Kronecker delta property but even though the proposed coupling approach performs satisfactorily. Convergence plots of error in energy norm show that, the proposed coupling approach performs very close to conventional coupling, with almost the same rate of convergence, even at very large influence domains. The proposed coupling procedure is also extended to three-dimensional linear and two-dimensional geometrically nonlinear problems. Once again, in these cases, it is shown from

the solution of benchmark numerical examples that the proposed coupling approach performs satisfactorily, with excellent agreement between the numerical and corresponding analytical or reference results. Although lower-order elements are used in the FE region, it is straightforward to extend to high-order elements. Furthermore, only geometric nonlinearities are used in this chapter but it is straightforward to include material nonlinearities. The FE-EFGM coupling procedure proposed in this chapter, will be extended to adaptive FE-EFGM coupling in the next chapter.

Chapter 6

Adaptive coupling of finite element and element-free Galerkin methods

6.1 Introduction

The main limitation of the FE-EFGM coupling procedure, proposed in Chapter 5, is that it is necessary to specify FE and EFG regions in the problem domain at the start of simulation in the preprocessing stage. These regions are fixed in the problem domain and performance is then highly user dependent. For practical engineering problems, it will be very difficult, especially for an inexperienced user to decide on appropriate FE and EFG regions in a given problem domain. For nonlinear problems with geometric and material nonlinearities, it is even more difficult to specify FE and EFG regions, as geometry is changing during the simulation. In these problems, the exact specifications of FE and EFG regions are obviously variable and change during the evolution of the problem geometry and changes in mesh quality. To overcome these shortcomings, in this chapter the FE-EFGM coupling proposed in Chapter 5 is extended to become **adaptive** FE-EFGM coupling for two-dimensional linear elastic and three-dimensional nonlinear problems with both geometric and material nonlinearities. In the method proposed here, initially the whole of the problem domain is modelled using the FEM. During an analysis those finite elements which violate a predefined error measure are automatically converted to an EFG zone. This zone can be further refined by adding nodes, thus avoiding computationally expensive FE remeshing.

A total Lagrangian formulation is preferred over the updated Lagrangian formulation for modelling finite deformation due to its computational efficiency. The well-established error estimation procedure of Zienkiewicz & Zhu [332] with the SPR method for recovery of the nodal stresses [334, 335] is used in the FE region to determine the elements requiring conversion to the EFG background integration cells. The Chung & Belytschko [62] error estimator is used in the EFG region for further adaptive refinement. For nonlinear problems, an incremental form of the Zienkiewicz & Zhu error estimation based on the SPR method [45, 334] is used to estimate error in the FE region. An incremental form of Chung & Belytschko's error estimation method for nonlinear problems covered in Chapter 4 and also given in reference [301], is used here to estimate error in the EFG zone in the case of nonlinear problems. The FE-EFGM coupling procedure based on the use of max-ent shape functions in the EFG zone, already covered in Chapter 5 and also given in reference [300], is used here to couple the evolving FE and EFG regions in the problem domain. A very limited literature is available for adaptive FE-meshless coupling, for instance references [150, 199] deal only with two-dimensional problems without proper error estimation and with no further adaptivity in the meshless zone.

This chapter is organized as follows. A detailed explanation of error estimation in the case of the FEM is given for two-dimensional linear elastic problems in §6.2. §6.2 is further divided into two parts, the SPR method is explained in §6.2.1, and the detail of error estimation is given in §6.2.2. The two-dimensional linear elastic error estimation procedure, explained in §6.2, is then extended to the three-dimensional nonlinear case in §6.3. The adaptive FE-EFGM coupling procedure for two-dimensional linear elastic problems is given in §6.4, and numerical examples are given in §6.4.1 to show its correct implementation and performance. A comparative study between the updated and total Lagrangian formulation is given in §6.5. The adaptive FE-EFGM coupling procedure, given in §6.4 for two-dimensional linear elastic problems, is further extended to three-dimensional nonlinear problems in §6.6. Two numerical examples are also given in §6.6.1, to show the implementation and performance of the three-dimensional adaptively coupled FE-EFGM algorithm. Concluding remarks follow in §6.7.

6.2 FE error estimation for two-dimensional linear elastic problems

For linear elastic problems, the main components of the adaptive FEM are the same as the corresponding EFGM, given in §4.2, i.e. estimation of the discretization error followed by refinement strategy. The full adaptive FE procedure is not covered here but only the FE error estimation is explained, which will then be used in the adaptively coupled FE-EFGM procedure to determine those FEs which need conversion to the EFG background cells. The details of the discretization error estimation procedure in the case of the EFGM are given in §4.2.1, which can be used directly in the case of the FEM. The only difference here in the case of the FEM is the calculation of stresses and strains. Nodal stresses are recovered here using the SPR method, which are then used to calculate the projected stresses for the Zienkiewicz & Zhu error estimation. The details of the SPR method are given in the following section.

6.2.1 Superconvergent patch recovery method

A general postprocessing procedure was developed by Zienkiewicz & Zhu in [334] to calculate the nodal stresses and strains from the nodal displacements in finite element analysis. In this procedure stresses and strains are initially calculated at the superconvergent points, which are then used to calculate stresses and strain at the nodes. The procedure was applied to one- and two-dimensional linear, quadrilateral and cubic elements. It was shown that, in the cases of linear and cubic elements the recovered nodal stresses and strains are superconvergent (one order higher accurate or $O(h^{P+1})$), i.e. for the linear and cubic elements the accuracy is $O(h^{1+1}) = O(h^2)$ and $O(h^{3+1}) = O(h^4)$ respectively, where h is the element size. It was also shown that an ultraconvergence, i.e. $O(h^{P+2})$ or two orders higher accuracy was obtained in the case of quadratic elements, i.e. $O(h^4)$. The recovered solution can then be used in the FE error estimation. A class of *a posteriori* error estimations, in which recovered solutions are used instead of the analytical solutions are known as Zienkiewicz & Zhu error estimations. The importance of the superconvergence of the recovered stresses of an FE solution for the asymptotic exactness of the Zienkiewicz & Zhu error estimator had already been shown, by the same authors, in their previous work in [326, 333]. It was also reported that, in comparison to other counterparts, the Zienkiewicz & Zhu error estimators were simple to implement and computationally very efficient.

The directly calculated nodal stresses and strains from the nodal displacements using the shape functions derivatives are however less accurate and discontinuous. Different strategies have been suggested in the literature to calculate accurate nodal stresses. One of the commonly used methods is averaging or local projection techniques, in which stresses are calculated at the nodes by extrapolation from the superconvergent sampling points, which are then averaged to get a single value. The accuracy of the

stresses from this method is highly dependent on the existence of the superconvergent points in the FEs. Further detail and theoretical background of the superconvergent points within FEs and the averaging or local projection method can be found in a number of references [9, 30, 128, 165, 186, 212, 220, 337, 338]. Other procedures for nodal stress recovery can also be found in the literature, including global projection [235, 236, 331] and extraction and other alternatives [22, 48, 292]. In all the above-mentioned nodal stress recovery methods, the nodal stresses obtained, are generally not superconvergent. In the SPR method, nodal stresses are calculated by using a discrete or local least square fitting over a set of superconvergent points in a patch of elements. The SPR method is superior to other stress recovery procedures because the recovered stresses are generally superconvergent or even ultraconvergent for the case of quadratic elements as stated above. The FE approximate displacement \mathbf{u}^h is written as

$$\mathbf{u}^h = \sum_{i=1}^n N_i \mathbf{u}_i, \quad (6.1)$$

where \mathbf{u}_i is a vector of nodal displacements associated with a node i , N_i is the FE shape function and n is the number of nodes in an element. The stress field $\boldsymbol{\sigma}^h$, consistent with the FE solution \mathbf{u}^h is written as

$$\boldsymbol{\sigma}^h = \mathbf{D}\boldsymbol{\varepsilon}^h = \mathbf{D}\nabla\mathbf{u}^h \quad (6.2)$$

where \mathbf{D} is the elasticity matrix, $\boldsymbol{\varepsilon}^h$ is the vector of consistent strain field and ∇ is a differential operator. The directly calculated consistent stress and strain fields are not continuous at element boundaries and nodes. The objective of the recovery technique is to find the nodal stresses $\boldsymbol{\sigma}_i^*$ such that a continuous and accurate stress field $\boldsymbol{\sigma}^*$ is obtained, i.e.

$$\boldsymbol{\sigma}^* = \sum_{i=1}^n N_i \boldsymbol{\sigma}_i^*, \quad (6.3)$$

where N_i is the same elemental shape function used in the displacement interpolation. Here the stress field $\boldsymbol{\sigma}^*$ is smooth throughout the problem domain and is more accurate than the corresponding consistent stress field $\boldsymbol{\sigma}^h$. In the SPR method, stresses are initially calculated at superconvergent points, from which nodal stresses $\boldsymbol{\sigma}_i^*$ are obtained by the least squares fitting of the complete basis as used in the displacement approximation over a patch of elements surrounding an assembly node. The polynomial expansion of the stress component σ_p^* is written as

$$\sigma_p^* = \mathbf{P}(\mathbf{x}) \mathbf{a} \quad (6.4)$$

where $\mathbf{P}(\mathbf{x})$ is the monomial basis function, $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}$ are spatial coordinates and \mathbf{a} is a vector of unknown coefficients. For one-dimensional linear and quadratic elements $\mathbf{P}(\mathbf{x})$ and corresponding \mathbf{a} are given as

$$\mathbf{P}(\mathbf{x}) = \begin{bmatrix} 1 & x \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T, \quad (6.5a)$$

$$\mathbf{P}(\mathbf{x}) = \begin{bmatrix} 1 & x & x^2 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T, \quad (6.5b)$$

for two-dimensional linear and quadratic elements $\mathbf{P}(\mathbf{x})$ and corresponding \mathbf{a} are given as

$$\mathbf{P}(\mathbf{x}) = \begin{bmatrix} 1, & x, & y \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1, & \dots, & a_3 \end{bmatrix}^T, \quad (6.6a)$$

$$\mathbf{P} = \begin{bmatrix} 1, & x, & y, & x^2, & xy, & y^2 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1, & \dots, & a_6 \end{bmatrix}^T, \quad (6.6b)$$

and for three-dimensional linear and quadratic elements $\mathbf{P}(\mathbf{x})$ and corresponding \mathbf{a} are given as

$$\mathbf{P}(\mathbf{x}) = \begin{bmatrix} 1, & x, & y, & z \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1, & \dots, & a_4 \end{bmatrix}^T, \quad (6.7a)$$

$$\mathbf{P}(\mathbf{x}) = \begin{bmatrix} 1, & x, & y, & z, & xy, & yz, & zx, & x^2, & y^2, & z^2 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_1, & \dots, & a_{10} \end{bmatrix}^T. \quad (6.7b)$$

In Equation (6.4), unknown coefficients \mathbf{a} are determined separately for each patch by least squares fitting to the stresses at the superconvergent points. For the least squares fitting, the following equation is minimized w.r.t. \mathbf{a}

$$\mathbf{F}(\mathbf{a}) = \sum_{i=1}^n (\sigma^h(\mathbf{x}_i) - \sigma_p^*(\mathbf{x}_i))^2 = \sum_{i=1}^n (\sigma^h(\mathbf{x}_i) - \mathbf{P}(\mathbf{x}_i)\mathbf{a})^2, \quad (6.8)$$

where n is the total number of superconvergent points in a single patch of elements, $\mathbf{x}_i = \begin{bmatrix} x_i, & y_i, & z_i \end{bmatrix}$ are the spatial coordinates of superconvergent points, $\sigma^h(\mathbf{x}_i)$ are the stresses calculated directly by the FEM at the superconvergent points and $\sigma_p^*(\mathbf{x}_i)$ are the recovered stresses at the same points using least square fitting. Minimization of Equation (6.8) w.r.t. \mathbf{a} gives

$$\frac{\partial \mathbf{F}(\mathbf{a})}{\partial \mathbf{a}} = 0. \quad (6.9)$$

This leads to the following set of equations

$$\mathbf{A}\mathbf{a} = \mathbf{b}, \quad \text{or} \quad \mathbf{a} = \mathbf{A}^{-1}\mathbf{b}, \quad (6.10)$$

where

$$\mathbf{A} = \sum_{i=1}^n \mathbf{P}^T(\mathbf{x}_i) \mathbf{P}(\mathbf{x}_i), \quad \mathbf{b} = \sum_{i=1}^n \mathbf{P}^T(\mathbf{x}_i) \sigma^h(\mathbf{x}_i). \quad (6.11)$$

After calculating \mathbf{a} , nodal stresses are calculated from Equation (6.4), i.e. using the nodal coordinates. The size of matrices \mathbf{A} and \mathbf{b} are relatively small and depend on \mathbf{P} . \mathbf{b} should be determined separately for each stress component, while \mathbf{A}^{-1} is the same for all stress components and is determined only once. The recovered nodal stresses in this case are superconvergent as these are very closely fitted to the stresses at the superconvergent points.

A typical patch for one-dimensional linear elements is shown in Figure 6.1(a), in which only two elements are assembled at a common node known here as a patch assembly node, while there is only one superconvergent point per element. The patch will be used to calculate the stress only at the patch assembly node. In Figure 6.1(b), a patch is shown for one-dimensional 3-node quadratic elements, which also consists of only two elements assembled at a common node, i.e. the patch assembly node, but in this case there are two superconvergent points per element. Also in this case, the patch will be used to recover the stresses at the patch assembly node as well as at all other nodes inside the patch, shown as solid circle in Figure 6.1(b). For one-dimensional 3-node quadratic elements; every internal element belongs to two patches, and for internal nodes, a value of stress is recovered from each patch, and the final

stress value is determined by averaging. For two-dimensional problems, typical patches for quadrilateral elements with four, eight, nine and sixteen nodes are shown in Figures 6.2(a), 6.2(b), 6.2(c) and 6.2(d) respectively. In all these patches, four elements are joined at a common patch assembly node. In Figures 6.2(a) there is only one superconvergent point per element, and the patch is used to recover the stresses only at patch assembly nodes. In Figure 6.2(b) and 6.2(c) there are four superconvergent points per element, while in Figure 6.2(d) there are nine superconvergent points per element. In Figures 6.2(b), 6.2(c) and 6.2(d), patches are used to recover stresses at the patch assembly nodes as well as internal nodes, which are inside these patches, shown as solid circles. In Figure 6.2(b), 6.2(c) and 6.2(d), all internal nodes normally belong to more than one patch, and stresses are recovered at these nodes from each patch, and the final stresses are then calculated by averaging. Finally, in the three-dimensional case, patches are shown in Figures 6.3(a) and 6.3(b) for 8-node and 20-node hexahedral elements respectively and the process works in a similar way to that described for one- and two-dimensional.

To recover stresses at boundary nodes, internal or boundary patches can be used. Accuracy of these stresses are the same, if recovered either from the boundary or internal patches [334]. In some situations, however sufficient elements and superconvergent points are not available to construct patches at the boundary. Even if sufficient elements and superconvergent points are available, the construction of these boundary patches involves extra unnecessary work, as the stresses at boundary nodes can be recovered from the already constructed internal patches. To avoid these complications, in this research, here internal patches are used to recover stresses at the boundary nodes. Typical internal patches for 16-node quadrilateral elements near the boundary, which can be used to recover the stresses at the boundary nodes are shown in Figure 6.4. A sample corner patch is shown in Figure 6.4(a), while a sample edge patch is shown in Figure 6.4(b).

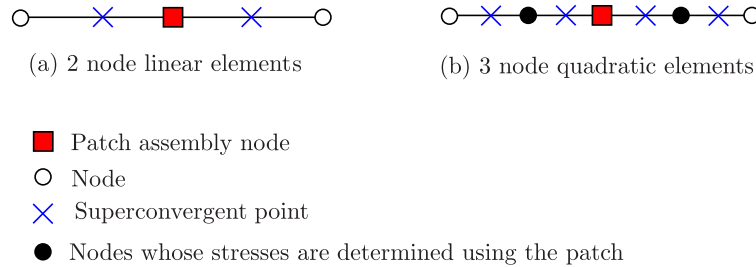


Figure 6.1: Sample one-dimensional linear and quadratic elements patches [334]

6.2.2 Error estimation

The accuracy of the Zienkiewicz & Zhu error estimation procedure is highly dependent on the accuracy of the stress recovery procedure. The SPR method is an ideal choice for the stress recovery in the Zienkiewicz & Zhu error estimation procedure due to its high accuracy and lower computational cost [335] as compared to other alternative. The FE error estimation procedure is the same as that used in the case of the EFGM, described in detail in §4.2.1.

6.3 Extension of FE error estimation to three-dimensional and nonlinear problems

The Zienkiewicz & Zhu error estimation procedure with the SPR method for stress recovery has already been used by other researchers in the literature for three-dimensional and nonlinear problems for adaptive analysis. The procedure for error estimation for nonlinear problems in the case of the FEM is the same,

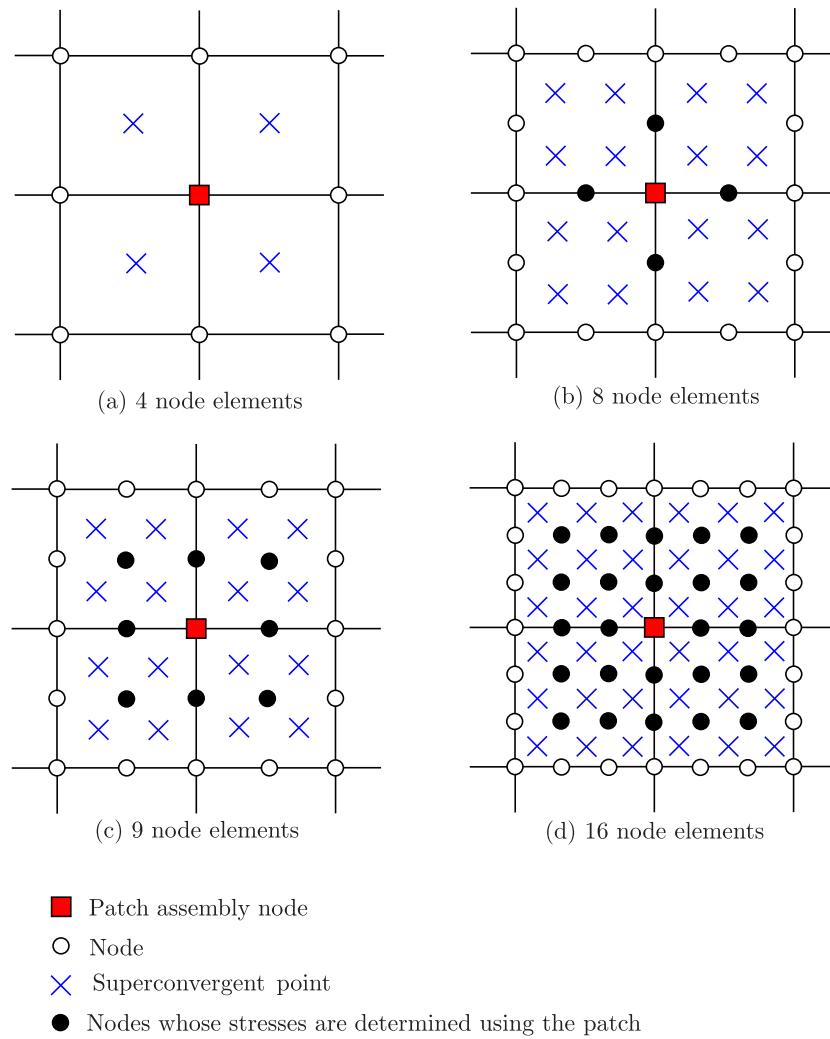


Figure 6.2: Sample two-dimensional elements patches [334]

as given in detail in §4.3 for the EFGM. Although, the full implementation and performance of the Zienkiewicz & Zhu error estimation for three-dimensional and nonlinear problems are not given in this section as we done previously for the linear elastic problem in §6.2 but the relevant literature is now surveyed.

The Zienkiewicz & Zhu error estimation procedure was extended to nonlinear problems for the first time in [45] and was used for two-dimensional problems subjected to small strain elasto-plasticity. For each solution step, incremental error in energy norm was calculated from nodal stresses and nodal incremental strains, recovered using the SPR method. Further applications of the Zienkiewicz & Zhu error estimation procedure in two-dimensional nonlinear problems can be found in a number of references, e.g. in [134], the SPR method was used to estimate error for two-dimensional footings on soil problems, subjected to large deformation with elasto-plasticity. 6-node triangular elements with three Gauss points per element were used to discretize the problems and the L_2 norm of error in strain was used for the adaptive analysis. In [163], recovery of the Cauchy stress was performed by the SPR method for two-dimensional problems with large deformation using hyperelasticity as a material model. The improved SPR method was introduced for two-dimensional large deformation problems in [117], in which integration points were used as sampling points and bilinear 4-node quadrilateral elements with (2×2) Gauss points were used in the analysis.

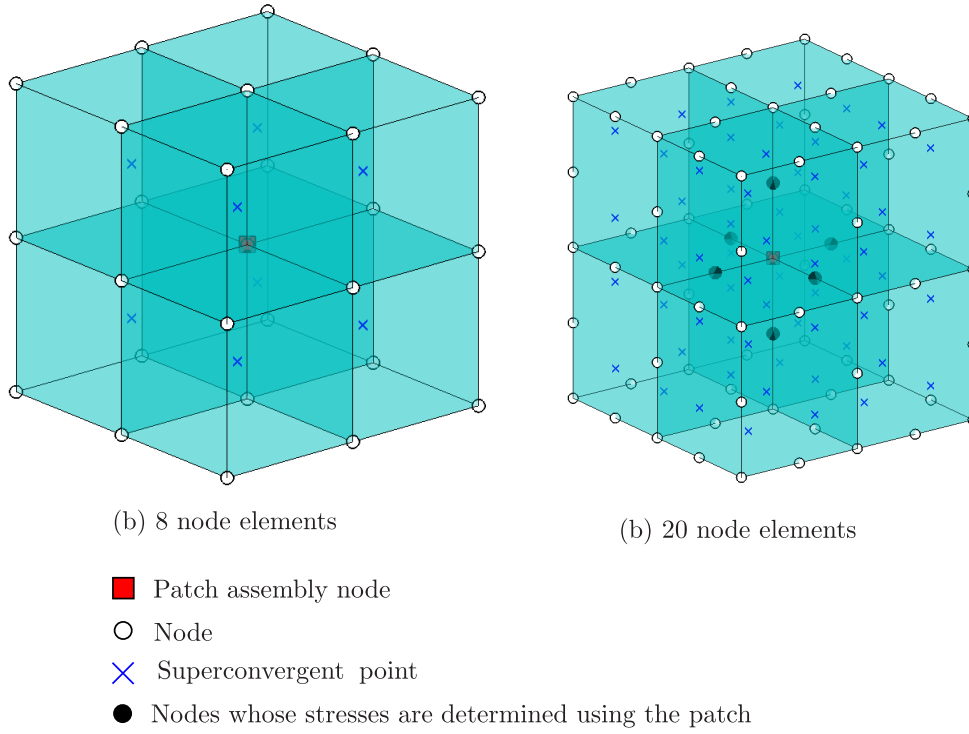


Figure 6.3: Sample three-dimensional elements patches

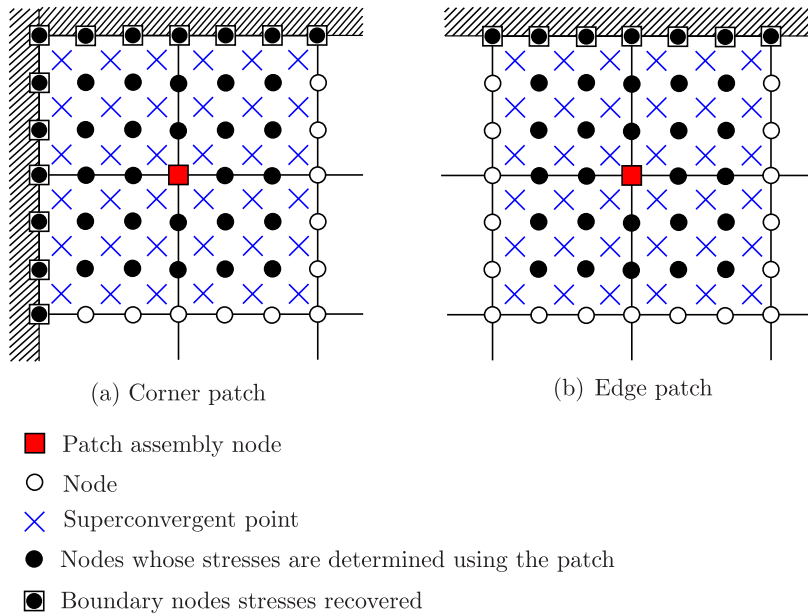


Figure 6.4: Stress calculation for boundary nodes

It was shown in [5, 26] that the SPR method can perform well, even if the sampling points used for stress recovery are not the superconvergent points. This information is very helpful to extend the SPR method originally proposed for linear problems to nonlinear problems, where most of the path dependent variables are available at the integration points, which are generally not superconvergent. The use of integration points as sampling points also allows the inclusion of more terms in the polynomial of the least squares fitting over the patch of elements, which is normally impossible with the superconvergent

points, because they are few in number. The nonexistence of superconvergent points in the case of nonlinear problems was also mentioned in [163], where it was concluded that using (2×2) integration points in linear 4-node quadrilateral elements as sampling points led to better performance than using one superconvergent Gauss point. Gauss points were also used as sampling points in [117] for problems subjected to large deformation.

The Zienkiewicz & Zhu error estimation procedure with the SPR method for stress recovery has been used in a number of linear and nonlinear three-dimensional problems. In [61], the SPR method was used to calculate error in three-dimensional h-adaptive analysis for linear-elastic problems. Adaptive FEA based on the modified SPR method was used to model curved cracks in three-dimensional problems in [226]. In [327], a new algorithm was proposed for mesh generation for the arbitrary geometries for hexahedral elements, which can be used in fully automatic adaptive FEA for practical engineering problems. In [316], the SPR method was used for error estimation in three-dimensional nonlinear problems, involving liquefaction of soil due to seismic effects. The incremental L_2 norm of error in strain for each solution step was used here in h-adaptive analysis, while linear 8-node hexahedral elements were used to discretize the problems' domains. The SPR method was extended to three-dimensional nonlinear problems, and its applications were explored in the transferring of the path dependent variables in [108, 160, 161]. Tetrahedral elements were used in the analysis, while first, second and third degree polynomials were used in the least squares fitting for the SPR method. In [47], an improvement of the SPR method, i.e. minimal patch recovery (MPR), was introduced in which there was no need to calculate the nodal stresses. The recovered solutions at Gauss points were calculated directly from the least squares projection of the Gauss points belong to the neighbouring elements. The method was applied to three-dimensional elasticity and metal forming problems with tetrahedral elements.

6.4 Adaptive FE-EFGM coupling for two-dimensional linear-elastic problems

The adaptive FE-EFGM coupling procedure described in this section is for two-dimensional linear elastic problems, which will be extended next to three-dimensional problems with both material and geometrical nonlinearities. In the adaptive FE-EFGM coupling procedure, as already mentioned in §6.1, initially, the whole of the problem domain is modeled using the FEM. During the analysis those elements which violate a predefined error measure are automatically converted to an EFG zone. This EFG zone can then be further refined by adding nodes, overcoming computationally expensive FE remeshing. In this case, local error in energy norm and the corresponding local energy norm for either an individual FE or EFG background cell Ω_e are written as

$$\|e_e^p\| = \left(\int_{\Omega_e} (\sigma_p^e(\mathbf{x}))^T \mathbf{D}^{-1} (\sigma_p^e(\mathbf{x})) d\Omega_e \right)^{\frac{1}{2}}, \quad \|U_e\| = \left(\int_{\Omega_e} (\sigma^p(\mathbf{x}))^T \mathbf{D}^{-1} (\sigma^p(\mathbf{x})) d\Omega_e \right)^{\frac{1}{2}}, \quad (6.12)$$

where subscript e shows an individual FE or EFG background cell with domain Ω_e , and $\sigma_p^e(\mathbf{x})$ is error in stress at a Gauss point \mathbf{x} , which is already given in §4.2.1 and $\sigma^p(\mathbf{x})$ is a vector of projected stresses at a Gauss point located at \mathbf{x} . Projected stress in the FE and the EFG zones of the problem domain are calculated using the procedures of Zienkiewicz & Zhu and Chung & Belytschko respectively. Equations for the global error in energy norm and the corresponding global energy norm for the problem domain Ω are then written as

$$\|e^p\|^2 = \sum_{i=1}^{n_e} \|e_i^p\|^2, \quad \|U\|^2 = \sum_{i=1}^{n_e} \|U_i\|^2, \quad (6.13)$$

where for linear elastic problems n_e is the number of FEs in the first iteration and the number of EFG background cells in the consecutive iterations. Global relative percentage error η and permissible local error in an individual FE or EFG background cells $\overline{\|e_e\|}$ are then written as

$$\eta = \frac{\|e^p\|}{\|U\|} \times 100, \quad \overline{\|e_e\|} = \frac{\bar{\eta}}{100} \left(\frac{\|U\|^2}{n_e} \right)^{\frac{1}{2}}, \quad (6.14)$$

where $\bar{\eta}$ is the global permissible relative error for the whole problem domain. The adaptive procedure is triggered by the global relative error, i.e. when $\eta > \bar{\eta}$ and the conversion of FEs to EFG background cells or further refinement of EFG background cells is performed when $\frac{\|e^p\|}{\|e_e\|} > 1$.

The full adaptively coupled FE-EFGM algorithm for linear elastic problems is shown in Figure 6.5, which has been implemented from scratch using MATLAB as a programming language. At the start of analysis, the whole of the problem domain is modelled with the FEM, which includes specification of geometry, material properties and meshing. Here four-node iso-parametric quadrilateral elements with (2×2) integration (or Gauss) points per element are used in the analysis. In the algorithm, shown in Figure 6.5, itr is the current iteration and max itr is the maximum number of iterations specified by the user at the start of analysis. During each iteration, stiffness matrix is assembled, essential and traction boundary conditions are implemented and after solution of the final system of equations nodal displacements or fictitious nodal values are calculated for the FE and the EFG regions respectively. During the first iteration, nodal stresses are calculated using the standard SPR method, already explained in §6.2.1, and local and global relative errors are then calculated using the Zienkiewicz & Zhu error estimation procedure, already explained in §6.2.2. This global relative error is then used to trigger the algorithm, which converts FEs to EFG background cells.

From the second iteration onward, the problem domain consists of combined FE and EFG discretizations with (2×2) Gauss points per element in the FE region and (4×4) Gauss points in each background cell in the EFG region. After this point, error is only calculated for the EFG region, as FEs with high errors are already converted into EFG background cells in the first iteration. After solution of the final system of equations, nodal stresses, local and global relative errors are calculated for the EFG region of the problem domain using the approach of Chung & Belytschko [62]. The global relative error is then used to decide further adaptive refinement in the EFG region, and those background cells, where error is more than the permissible error are automatically refined.

The refinement strategy used here is almost the same as used for the adaptive EFGM given in §4.2.2 and is shown in Figure 4.8. The direct use of the adaptive EFGM refinement strategy here in the EFG region of the problem domain creates complications in the FE region of the problem domain, which requires further consideration. The step by step improved refinement strategy, suitable for the adaptively coupled FE-EFGM is shown in Figure 6.6. In this new strategy, refinement of the EFG background cells is the same as used in the original adaptive EFGM case, i.e. five new nodes are added in each background cell, which is then divided into four new cells. A sample coupled FE-EFG discretization is shown in Figure 6.6(a), in which columns A, B and C are the EFG background integration cells and columns D, E and F are FEs. The refinement of background cells in column A, B and C creates hanging nodes in FEs, i.e. columns D in the FE region of the problem domain as shown in Figure 6.6(b). To avoid the complication of dealing with FEs with hanging nodes, the strategy used here is to convert all those FEs to the corresponding EFG background integration cells as shown in Figure 6.6(c).

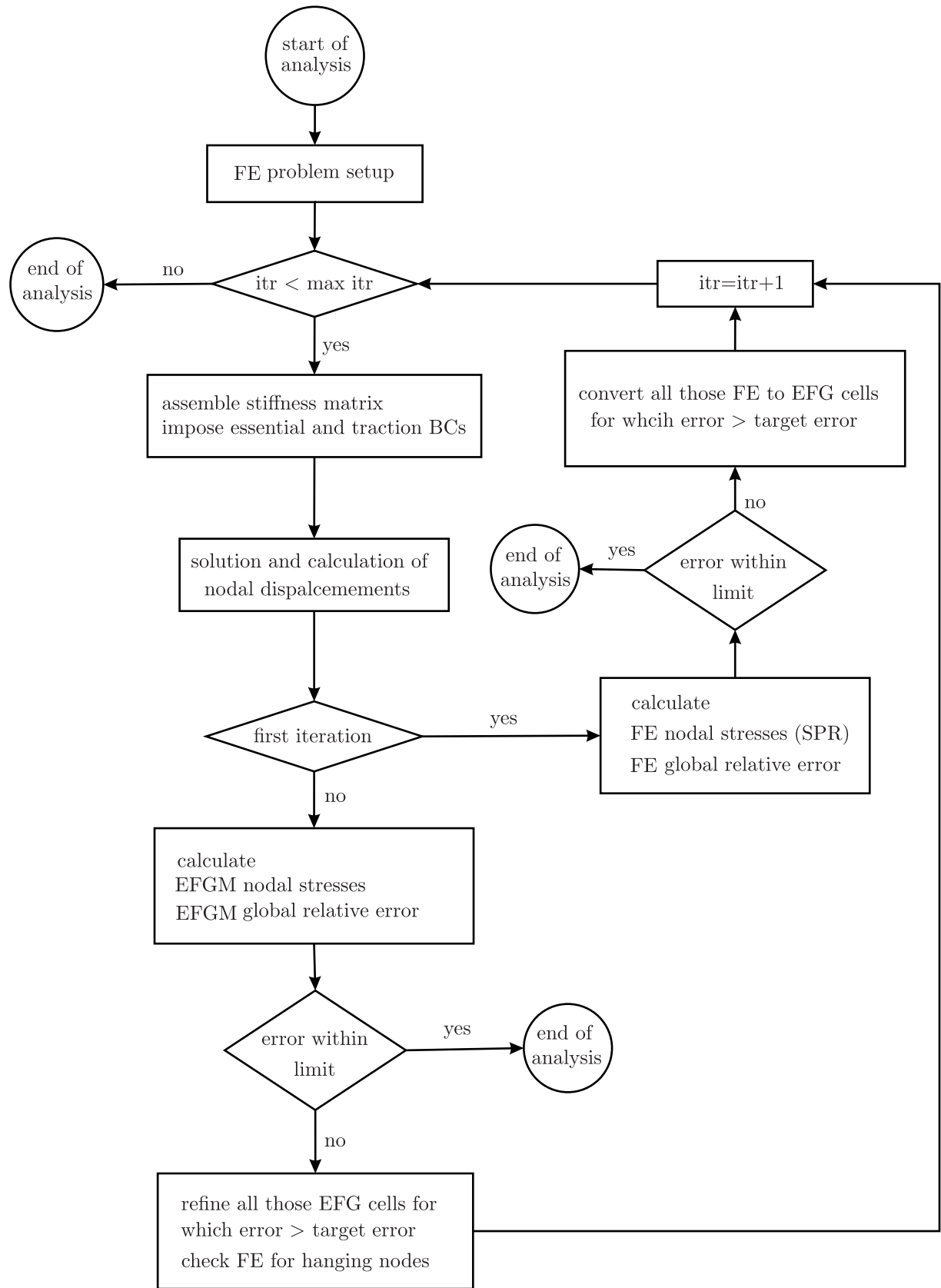


Figure 6.5: Adaptively coupled FE-EFGM algorithm for linear elastic problems

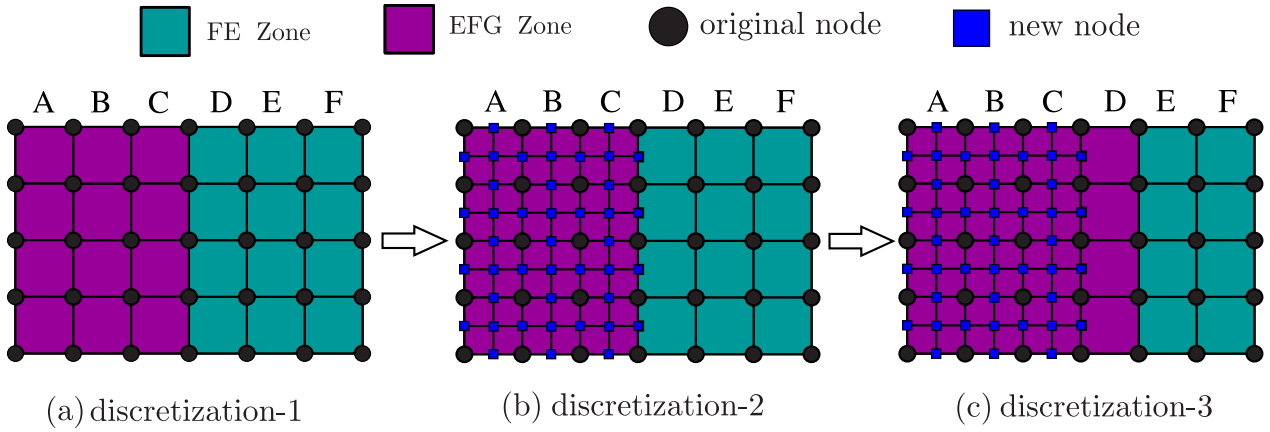


Figure 6.6: Step by step refinement strategy for the adaptive coupled FE-EFGM

6.4.1 Numerical examples

In the following two-dimensional linear elastic numerical examples are given to demonstrate the correct implementation and performance of the full adaptively coupled FE-EFGM modelling procedure.

Square block problem

The first numerical example is a two-dimensional square block subjected to uniform pressure on the left edge. The same problem is also solved using the adaptive EFGM in §4.2.4. The same geometry, boundary conditions, loading and material properties are used here as given in §4.2.4. The global permissible error $\bar{\eta}$ used here is 6%, and the scaling parameters used for the domains of influence for analysis and projection in the EFG zone are $d_{max}^a = 1.5$ and $d_{max}^p = 1.2$ respectively. The evolving step by step adaptively coupled FE-EFGM discretizations in this case are shown in Figure 6.7. The FE mesh at the start of analysis with 225 (15×15) elements and 256 (16×16) nodes is shown in Figure 6.7(a). The converted FEs to corresponding EFG background cells based on the Zienkiewicz & Zhu error estimation at the end of first iteration are shown in Figure 6.7(b). During this first FEs to EFG background cells conversion the number of nodes is constant. Subsequent refinements in the EFG regions of the problem domain based on the Chung & Belytschko error estimation procedure are shown in Figure 6.7(c). It can also be seen in Figure 6.7(c) that in the EFG zone on the right-hand side, all new nodes are added within the EFG region and there is no issue of hanging nodes in the FE region of the problem domain, but in the EFG zone on the left-hand side, four more FEs at the top of EFG zone are converted to EFG background cells, to avoid hanging nodes in the FE region of the problem domain. As expected, the adaptively coupled FE-EFG algorithm, initially converts the high-stress zones in the problem domain to the corresponding EFG zones and then adds nodes to the high-stress EFG zones.

The corresponding integration (or Gauss) points in the evolving adaptively coupled FE-EFGM discretizations are shown in Figure 6.8, where (2×2) Gauss points per element are used in the FE region of the problem domain and (4×4) Gauss points are used in each background cell in the EFG region of the problem domain. The contours of displacements in the x-direction, i.e. u_x over the 100X magnified deformed geometry for the three consecutive discretizations are shown in Figures 6.9(a), 6.9(b) and 6.9(c) respectively, which are in a very good agreement with each other demonstrating a correct implementation of the adaptively coupled FE-EFGM algorithm. The contours of the von Mises stress over the problem domain after the first FE iteration are also shown in Figure 6.10, which are obtained from the SPR

method's recovered nodal stresses using the following equation

$$\sigma_{VM} = [\sigma_{xx}^2 + \sigma_{yy}^2 - \sigma_{xx}\sigma_{yy} + 3\sigma_{xy}^2]^{\frac{1}{2}}, \quad (6.15)$$

where σ_{xx} and σ_{yy} are the normal stresses in x and y directions and σ_{xy} is the shear stress. For comparison, the same problem is also solved with uniformly refined FE meshes with almost the same number of nodes as shown in Figures 6.7(a) and 6.7(c). The uniformly refined meshes in this case with 256 (16×16) and 400 (20×20) nodes are shown in Figures 6.11(a) and 6.11(b) respectively. Comparison of the relative error (η) for the adaptively coupled FE-EFGM case and uniformly refined FEM case are shown in Figure 6.12, in which three data points are available for the adaptively coupled FE-EFGM case and only two data points are available in the uniformly refined FEM case. In Figure 6.12, it is clear that the decrease in the relative error is greater in the case of the adaptively coupled FE-EFGM as compared to the uniformly refined FEM case.

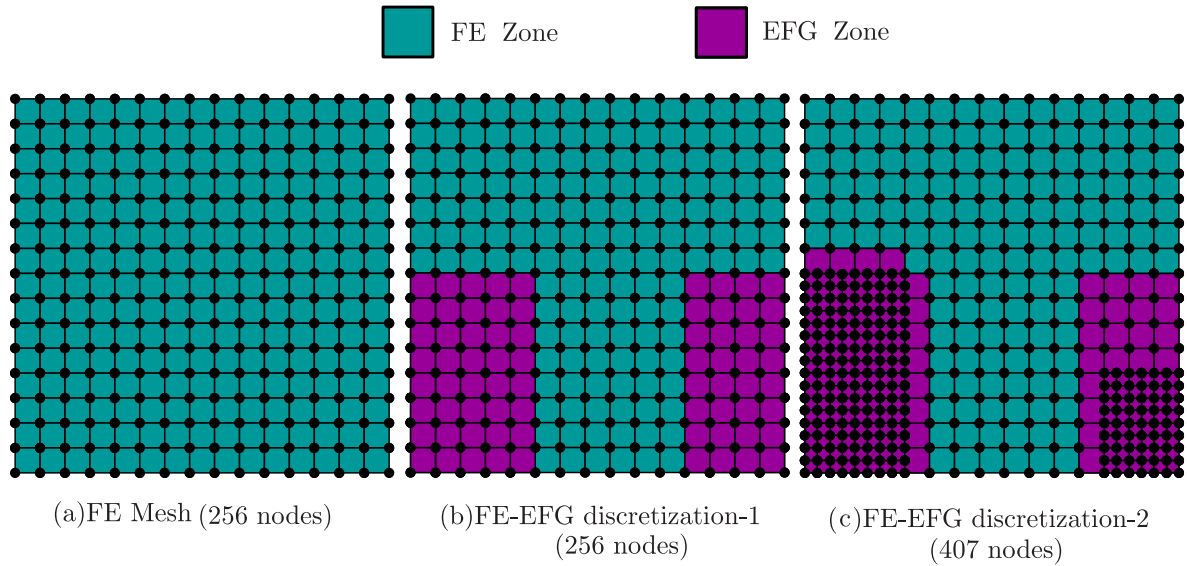


Figure 6.7: Step by step discretizations for the square block problem

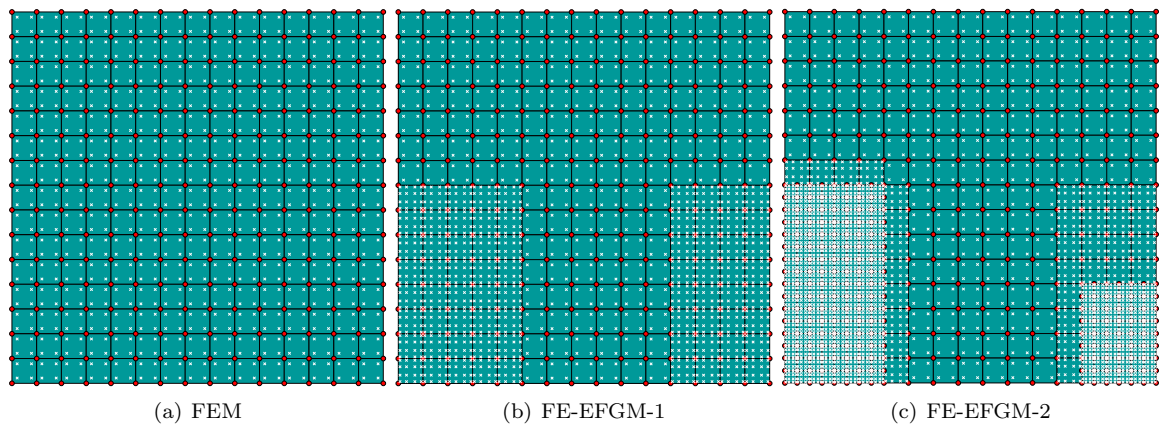


Figure 6.8: Step by step Gauss points for the square block problem

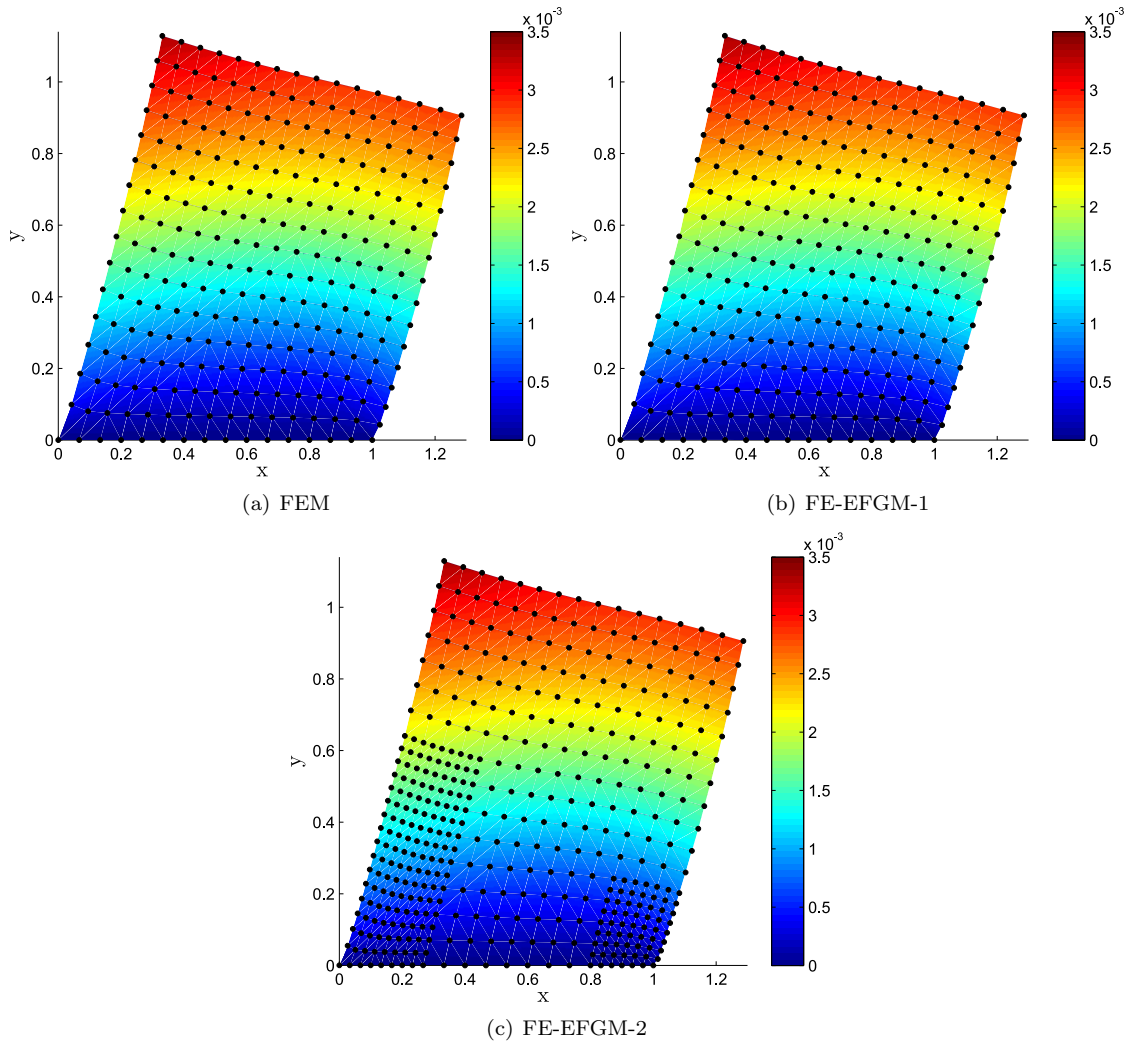
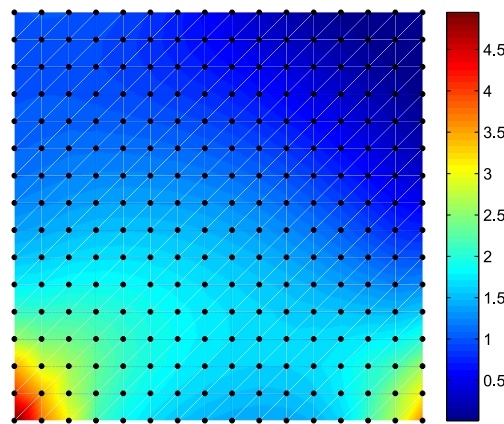
Figure 6.9: u_x contours for the square block problem

Figure 6.10: Von Misses stress contours for the square block problem

L-shaped plate

The second two-dimensional linear elastic numerical example is an L-shaped plate subjected to uniform pressure on the left edge. The same problem is also solved using the adaptive EFGM in §4.2.4 and is

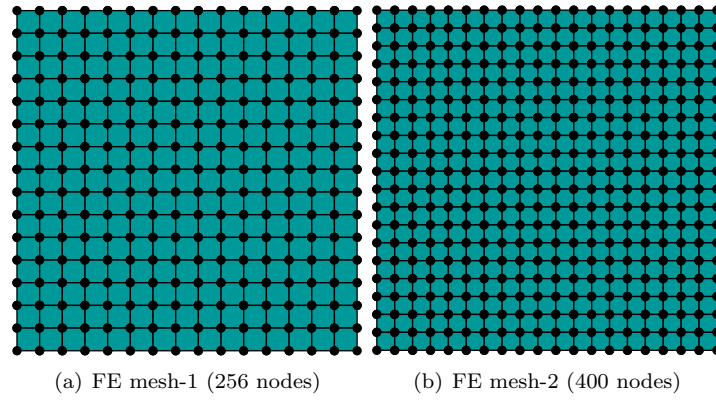
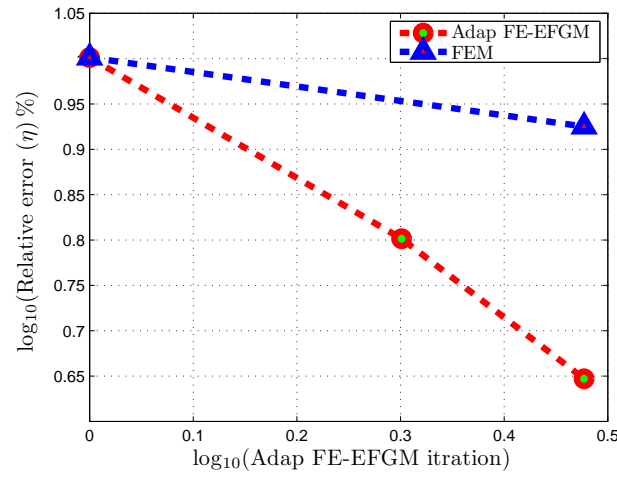


Figure 6.11: Step by step FE uniform refinement for the square block problem

Figure 6.12: Relative error $(\eta) \%$ for the square block problem

a standard numerical example to test the performances of adaptive algorithms. The same geometry, boundary conditions, loading and material properties are used in this case as shown in Figure 4.24 and given in §4.2.4. The permissible relative error $\bar{\eta}$ used here is 7%, while the scaling parameters used for the domains of influence for analysis and projection in the EFG region are $d_{max}^a = 1.4$ and $d_{max}^p = 1.1$ respectively. The step by step evolving discretizations in this case are shown in Figure 6.13. The FE mesh at the start of analysis with 341 nodes is shown in Figure 6.13(a) and the first and second adaptively coupled FE-EFG discretization with 341 and 406 nodes are shown in Figure 6.13(b) and 6.13(c) respectively. The number of nodes is the same in the first two discretizations. In 6.13(c) two more FEs are converted to the corresponding EFG background cells, to avoid hanging nodes in these elements. As expected, near the corner A in Figure 6.13, the adaptively coupled FE-EFG algorithm converts the FE elements to EFG background cells and adds more nodes in the EFG region.

The Gauss points for these evolving discretizations are shown in Figures 6.14(a), 6.14(b) and 6.14(c) respectively, i.e. (2×2) Gauss points in each FE and (4×4) Gauss points in each background cell. The contours of the displacements in the x-direction, i.e. u_x for the three consecutive adaptively coupled FE-EFG discretizations are shown in Figures 6.15(a), 6.15(b) and 6.15(c) respectively. These displacements contours are in excellent agreement with each other, which again shows the correct implementations of the adaptively coupled FE-EFG algorithm. The contours of von Mises stress after the first FE iteration over the problem geometry are also shown in Figure 6.16, which shows maximum stress near corner A.

For comparison the same problem is also solved with the FEM with uniformly refined meshes with almost the same number of nodes as shown in Figure 6.13. The first and second uniformly refined meshes with 341 and 408 nodes are shown in Figure 6.17(a) and 6.17(b) respectively. Comparison between the relative error (η) for the adaptively coupled FE-EFGM case and uniformly refined FEM case is shown in Figure 6.18, in which the adaptively coupled FE-EFGM case clearly performs better than the uniformly refined FEM.

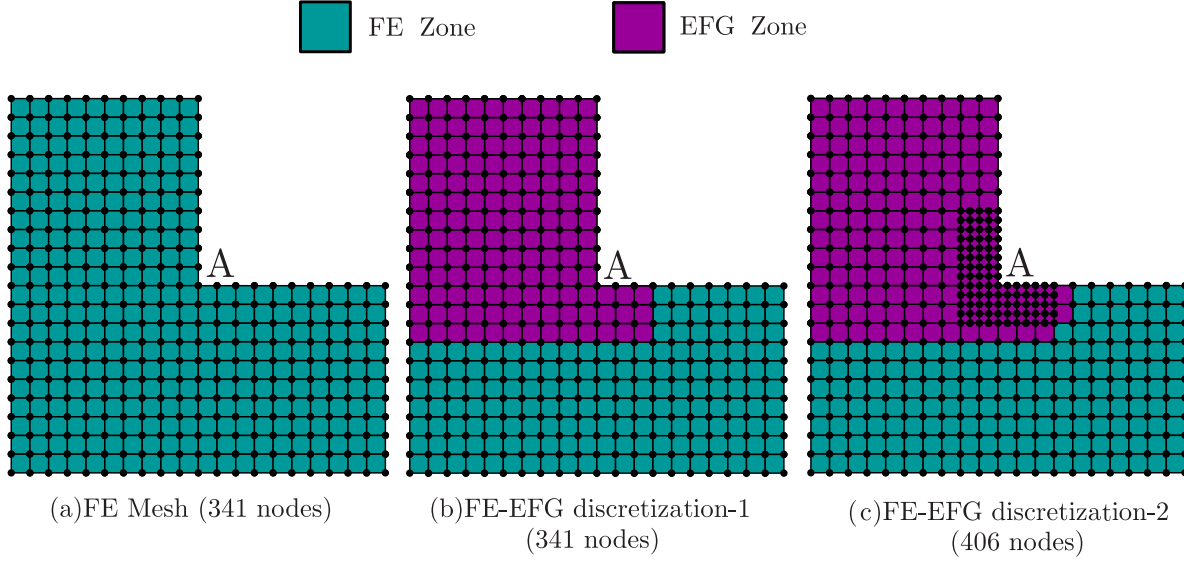


Figure 6.13: Step by step discretizations for the L-shaped plate problem

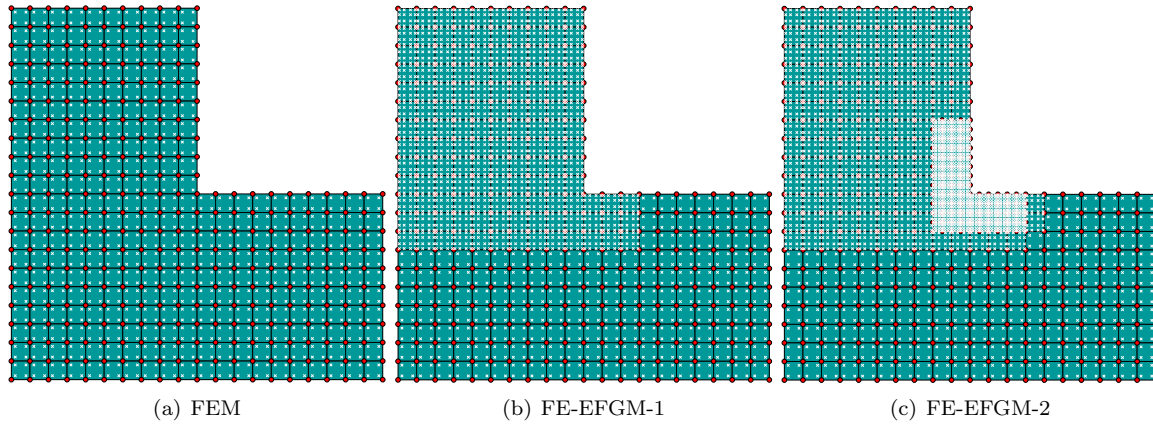


Figure 6.14: Step by step Gauss points for the L-shaped plate problem

6.5 Implementation of the total Lagrangian formulation

So far, in this thesis updated Lagrangian formulation has been used to model problems subjected to finite deformation, as describe in detail in §3.3.1. In the updated Lagrangian formulation, geometry is updated after each solution step, which includes the update of nodal coordinates, Gauss points and domains of influence. Shape functions and corresponding derivatives are also calculated during each solution step. On the other hand, in total Lagrangian formulation, there is no need to update the geometry as the variables are referred back to the original configuration. Shape functions and corresponding derivatives are calculated and stored at the start of the analysis and are used in every solution step. The detailed total

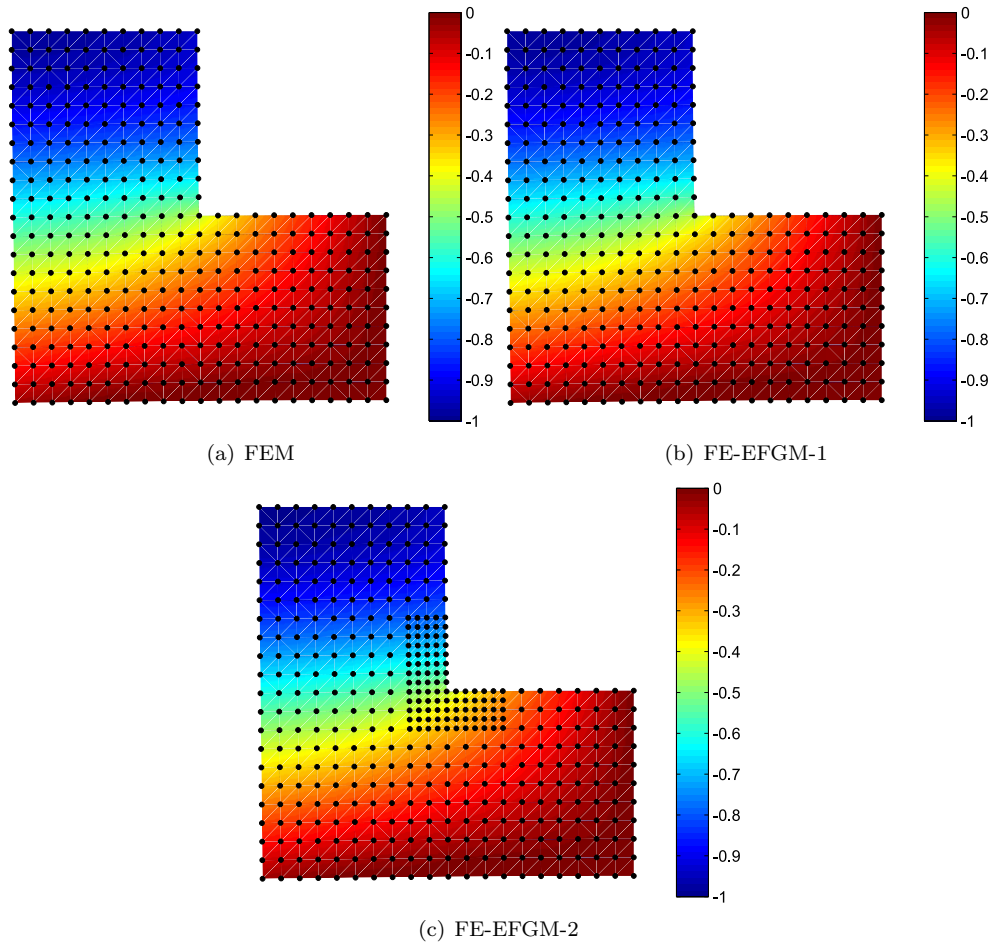
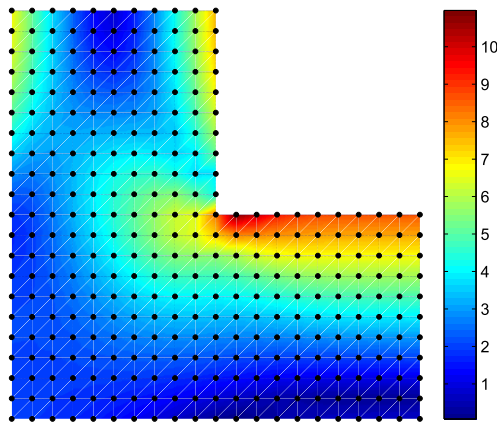
Figure 6.15: u_x contours for the L-shaped plate problem

Figure 6.16: Von Misses stress contours for the L-shaped plate problem

Lagrangian formulation is given in §3.3.2. Due to all these involved computations, updated Lagrangian formulations is not suitable to use for meshless methods, as the calculation of influence domains and shape functions and corresponding shape function derivatives are computationally expensive tasks in meshless methods.

In this section, the performance of a total Lagrangian formulations, while modelling problems with large deformation and elasto-plasticity in the case of meshless methods is compared with the updated

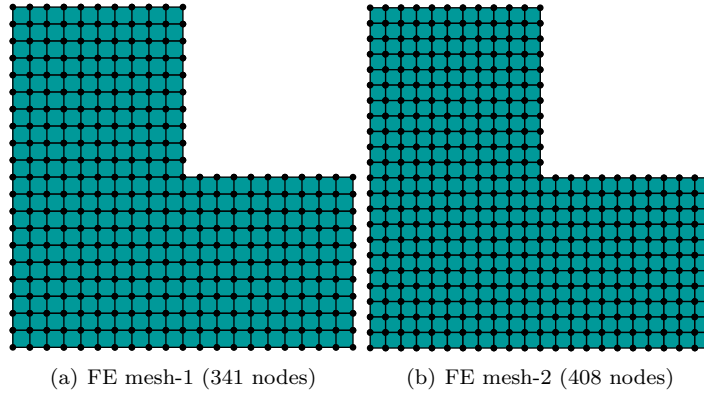
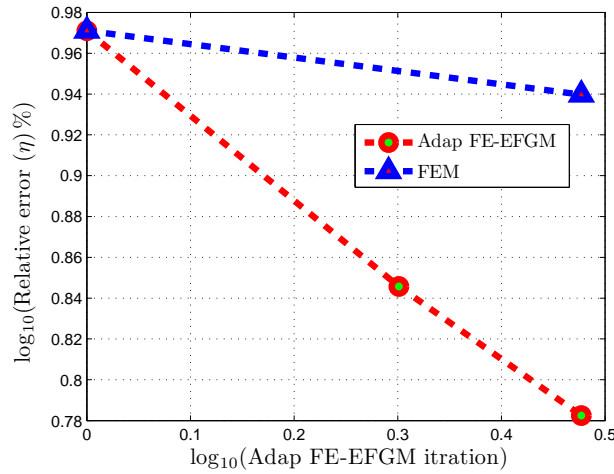


Figure 6.17: Step by step FE uniform refinement for the L-shaped plate problem

Figure 6.18: Relative error (η) % for the L-shaped plate problem

Lagrangian. For this purpose, the three-dimensional plate with a hole problem is considered, for which the geometry, boundary conditions and material properties are given in §4.4.1. In the analysis a displacement of 0.2 units is applied to the top face in 20 equal steps. The scaling parameter used for the domain of influence in this case is $d_{max} = 1.5$. For comparison of the simulation time, the problem is solved using five different discretizations with 189, 480, 975, 1728 and 2793 degrees of freedom. Comparisons between the simulation time versus degrees of freedom for both the updated and total Lagrangian formulations are shown in Table 6.1 and in Figure 6.19. The times reported here, are the times required for full simulation. It is clear from Table 6.1 and Figure 6.19 that the total Lagrangian formulation is computationally efficient as compared to the updated Lagrangian formulation.

A comparison of the reaction versus displacement between the total and updated Lagrangian formulations is also shown in Figure 6.20. Three different curves, obtained using 189, 975, and 2793 degrees of freedom in the analysis are considered in this case. In Figure 6.20, UL and TL refers to updated Lagrangian formulation and total Lagrangian formulation respectively. The corresponding curves for the total and updated Lagrangian results are in a very good agreement with each other. The geometric softening with the increase in degrees of freedom is also clear in both the updated and total Lagrangian cases. Unlike the FEM modelling, where the corresponding reaction versus displacement curves for the updated and total Lagrangian cases for the same number of degrees of freedoms matches exactly, here a small difference can be observed in these curves. The difference in these corresponding curves may be

due to the update of influence domains at the end of each solution step in the updated Lagrangian case, due to which nodes in the support of each Gauss points changes from one load step to the next, while in the total Lagrangian case these nodes remain constant throughout the simulation.

DOFs	Updated Lagrangian (sec)	Total Lagrangian (sec)
189	39.99	17.01
480	281.55	115.57
975	1133.48	448.11
1728	2992.42	1041.46
2793	7288.38	2905.86

Table 6.1: Simulation times versus degrees of freedom for the updated and total Lagrangian formulations

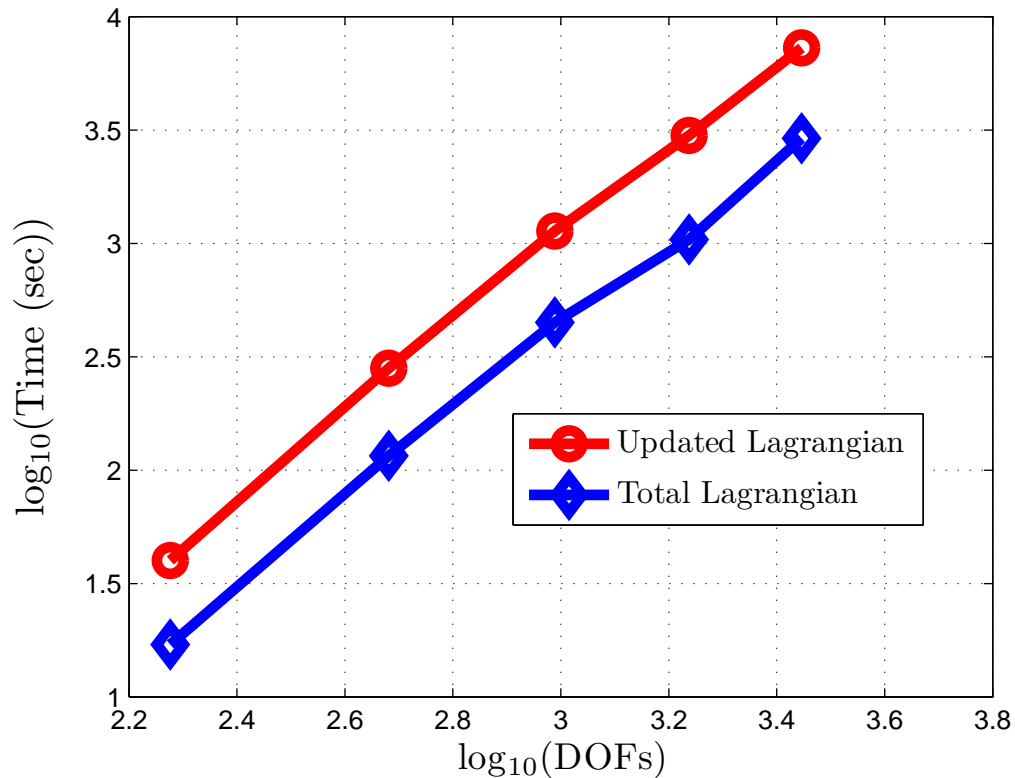


Figure 6.19: Simulation times versus degrees of freedom for the updated and total Lagrangian formulations

6.6 Adaptive FE-EFGM coupling for three-dimensional nonlinear problems

In this section, the adaptive FE-EFGM coupling procedure for two-dimensional linear elastic problems is extended to three-dimensional nonlinear problems with both material and geometrical nonlinearities. The total Lagrangian formulation is used here to model finite deformation and for modelling elasto-plasticity, the Prandtl-Reuss constitutive model is used, which comprises the von Mises yield function with perfect plasticity and associated flow. As compared to the two-dimensional linear elastic counterpart, in which error in the FE region of the problem domain is calculated only in the first discretization, here the error

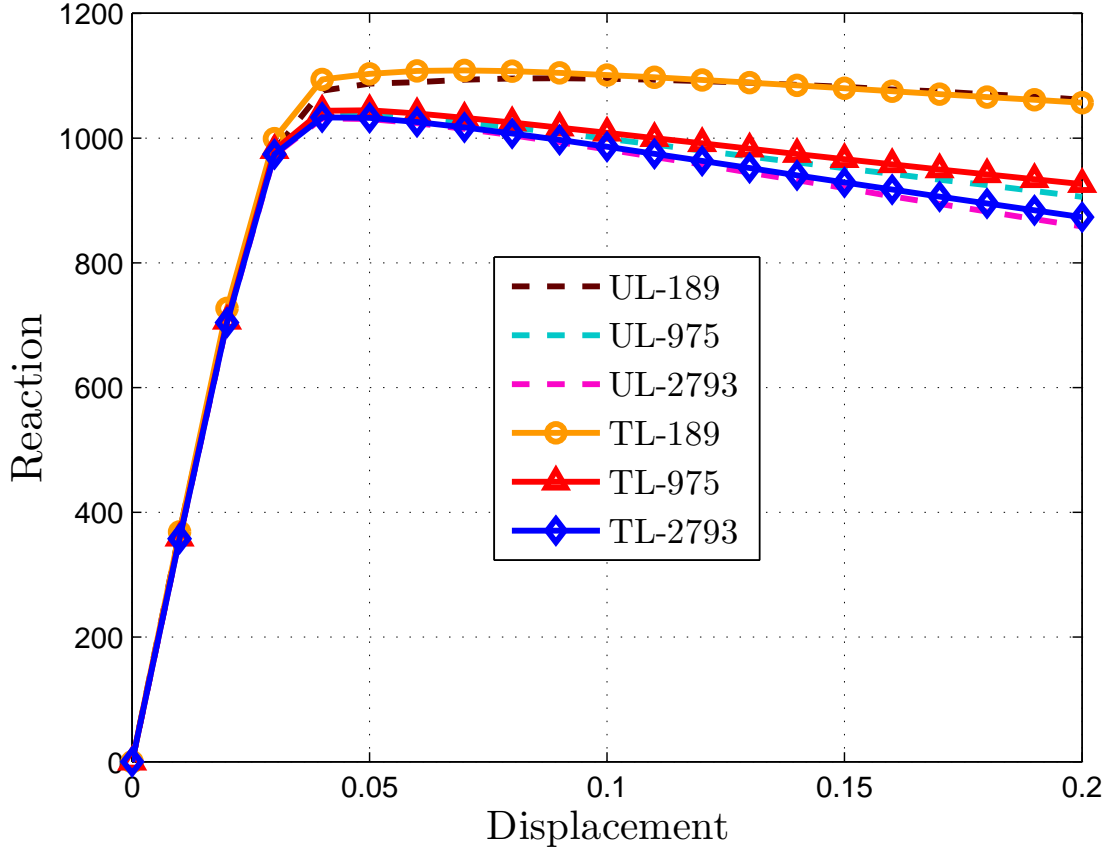


Figure 6.20: Reaction versus displacement for the updated and total Lagrangian formulations

in the FE region of the problem domain is calculated during each evolving discretization. Due to the incremental loading, it is expected in the subsequent discretizations that more elements will be converted to corresponding EFG background integration cells. The three-dimensional nonlinear adaptively coupled FE-EFG algorithm has been implemented from scratch by the author using FORTRAN. The same incremental procedure, used for the adaptive FEM in [45] and for the adaptive EFGM in [301] is used here. In this case, incremental global relative error in energy norm is calculated for each solution step, and checked against a permissible value. Here due to the total Lagrangian formulation, undeformed discretizations are refined as compared to the current or deformed discretizations refinements in the updated Lagrangian formulation.

For solution step n , equations for the incremental error in energy norm and the corresponding energy norm for either an individual FE or EFG background cell Ω_e are written as

$$\|e_e^p\| = \left[\int_{\Omega_e} \left| (\boldsymbol{\tau}_n^p(\mathbf{x}) - \boldsymbol{\tau}_n^h(\mathbf{x}))^T (\Delta \boldsymbol{\varepsilon}_n^p(\mathbf{x}) - \Delta \boldsymbol{\varepsilon}_n^h(\mathbf{x})) \right| d\Omega_e \right]^{\frac{1}{2}}, \quad \|U_e\| = \left[\int_{\Omega_e} \left| (\boldsymbol{\tau}_n^p(\mathbf{x}))^T (\Delta \boldsymbol{\varepsilon}_n^p(\mathbf{x})) \right| d\Omega_e \right]^{\frac{1}{2}}, \quad (6.16)$$

where the subscript e shows an individual FE or EFG background cell, $\boldsymbol{\tau}_n^p(\mathbf{x})$ and $\boldsymbol{\tau}_n^h(\mathbf{x})$ are the projected and numerical Kirchhoff stresses respectively at a Gauss point \mathbf{x} for solution step n , while $\Delta \boldsymbol{\varepsilon}_n^p(\mathbf{x})$ and $\Delta \boldsymbol{\varepsilon}_n^h(\mathbf{x})$ are the projected and numerical incremental logarithmic strains at a Gauss point \mathbf{x} for the solution step n . Projected Kirchhoff stresses and the projected logarithmic strains in this case are

calculated using the Zienkiewicz & Zhu and Chung & Belytschko procedures in the FE and the EFG region of the problem domain respectively.

Equations for the error in energy norm and the corresponding energy norm for the problem domain Ω are written as

$$\|e^p\|^2 = \sum_{i=1}^{n_{FE}} \|e_{FE}^p\|_i^2 + \sum_{j=1}^{n_{EFG}} \|e_{EFG}^p\|_j^2, \quad \|U\|^2 = \sum_{i=1}^{n_{FE}} \|U_{FE}\|_i^2 + \sum_{j=1}^{n_{EFG}} \|U_{EFG}\|_j^2, \quad (6.17)$$

where n_{FE} and n_{EFG} are the total numbers of FEs and EFG background cells respectively in the problem domain. Incremental global relative percentage error and incremental permissible error in an individual FE or EFG background cell are written as

$$\eta = \frac{\|e^p\|}{\|U\|} \times 100, \quad \overline{\|e_e\|} = \frac{\bar{\eta}}{100} \left(\frac{\|U\|^2}{n} \right)^{\frac{1}{2}}, \quad (6.18)$$

where $n = n_{FE} + n_{EFG}$ and $\bar{\eta}$ is a permissible relative error for the whole problem domain. The remaining procedure of conversion of the FEs to the EFG background cells and further refinement of the EFG background cells are the same as given for the two-dimensional linear elastic case. The same refinement strategy suggested originally for the two-dimensional linear elastic case, is extended here to three-dimensions. The details of dealing with other complications associated with the nonlinear case, including data transfer between the evolving discretization are the same as already covered in Chapter 4.

6.6.1 Numerical examples

Numerical examples are now given to demonstrate the correct implementation and performance of the full three-dimensional nonlinear adaptively coupled FE-EFG algorithm. The scaling parameters used here for the domains of influence for analysis and projection in the EFG zone are $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ respectively. In the following numerical examples 25% permissible relative error is used. Kd-tree with background mesh is used for the neighbour search in the following numerical examples the details of which is given in Appendix-B.

Three-dimensional plate with a hole

The first numerical example is a three-dimensional plate with a central hole subjected to unidirectional tension, the detail of which can be found in §4.4.1. Only one-eighth of the plate is modelled here due to symmetry. A total displacement of 0.15 units is applied to the top face in 15 equal steps. At the start of analysis, the whole of the problem domain is modelled using the FEM with 325 nodes as shown in Figure 6.21(a). For this first discretization, the standard SPR method is used to recover the nodal stresses and strains, which are then used in the Zienkiewicz & Zhu error estimation procedure. After first conversion, the second discretization consists of both FEs and EFG background cells as shown in Figure 6.21(b), in which the number of nodes remains the same. From the second discretization onward, both Zienkiewicz & Zhu and Chung & Belytschko error estimation procedures are used in the FE and the EFG regions of the problem domain respectively. Subsequent discretizations after further conversion of the FEs to the EFG background cells and refinement in the EFG background cells are shown in Figures 6.21(c) and 6.21(d). The number of nodes in the third and fourth discretizations are 929 and 4119 respectively. The adaptively coupled FE-EFG algorithm based on the combined Zienkiewicz & Zhu and Chung & Belytschko error estimation procedures add nodes automatically in the thinning section of the plate as expected.

The contours of the displacements u_x and u_y are shown in Figures 6.22(a) and 6.22(b) respectively,

which shows the correct implementation of the adaptively coupled FE-EFG algorithm. The contours of the effective plastic strain are also shown in Figure 6.22(c), in which a shear band of finite thickness next to the hole is clearly evident. Reaction versus displacement plot for this problem is given in Figure 6.23. For comparison, the same problem is also solved without adaptivity, using different initial FE and coupled FE-EFG discretizations shown in Figure 6.21 for which the reaction versus displacement plots are shown in Figure 6.23 for comparison. In this case, the solution obtained with final adaptively coupled FE-EFG discretization shown in Figure 6.21(d) is considered as a reference. As compared to the coupled FE-EFGM cases, the FEM response is relatively rigid without obvious geometric softening behaviour. The convergence of the reaction versus displacement curves to the reference solution, with an increasing the number of nodes can also be observed in Figure 6.23, which shows the correct implementation and effectiveness of the proposed adaptively coupled FE-EFG algorithm.

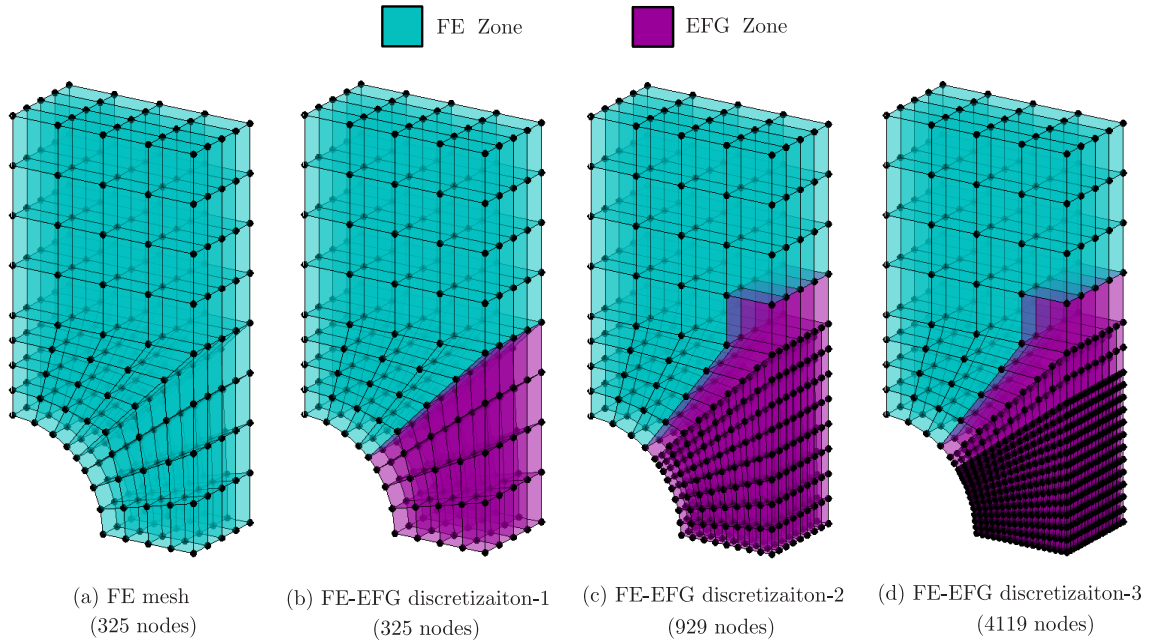


Figure 6.21: Step by step discretizations for the three-dimensional plate with a hole problem

Three-dimensional footing loaded on a vertical cut

The second numerical example is a three-dimensional footing loaded next to unsupported faces of soil, the detail of which is given in §4.4.1. The geometry used in this case is shown in Figure 6.24(a). Here only the soil section of the problem is modelled with the adaptively coupled FE-EFGM with total vertical displacement of 0.1 units applied to nodes beneath the footing in 15 equal steps. The FE mesh at the start of the analysis is shown in Figure 6.24(b) and the step by step conversion of the FEs to the EFG background cells and subsequent refinements of the EFG background cells are shown in Figures 6.24(c), 6.24(d) and 6.24(e). The number of nodes used in first two discretizations is 729, while in the subsequent discretizations, the number of nodes increases to 1215 and 2950. For this problem, failure is expected in the soil below the footing, where FEs are automatically converted to the EFG background cells, which are refined in the subsequent discretizations. The contours of u_y and effective plastic strain are also shown in Figures 6.25(a) and 6.25(b) respectively. It can be seen in Figure 6.25(a), that the total displacement is concentrated below the footing and a very clear shear band of finite thickness can also be seen in Figure 6.25(b). A reaction versus displacement plot for this problem is shown in Figure 6.26. For comparison, the same problem is also solved with different starting discretizations shown in Figure 6.24

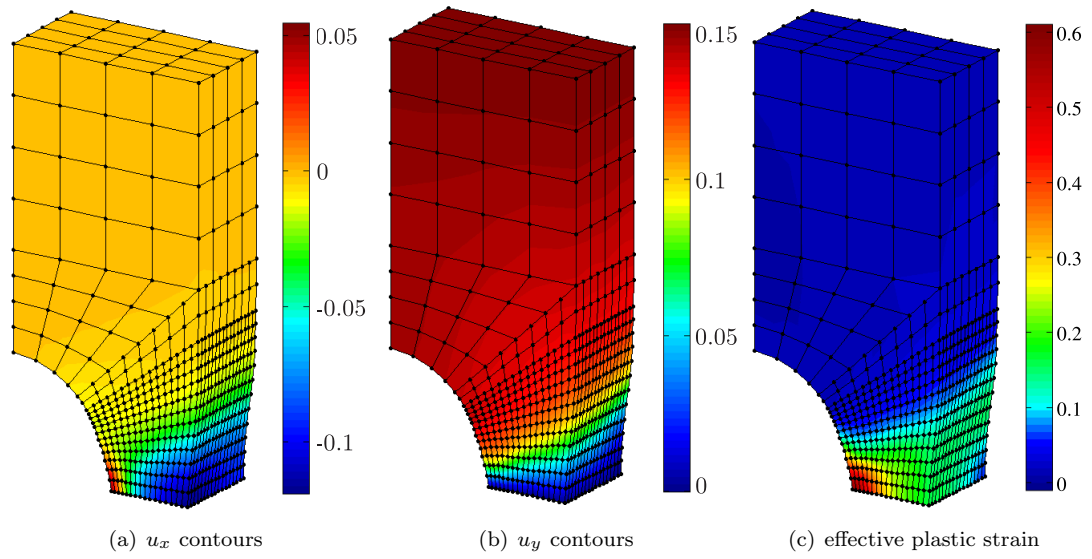


Figure 6.22: Displacements and effective plastic strain contours for the three-dimensional plate with a hole problem

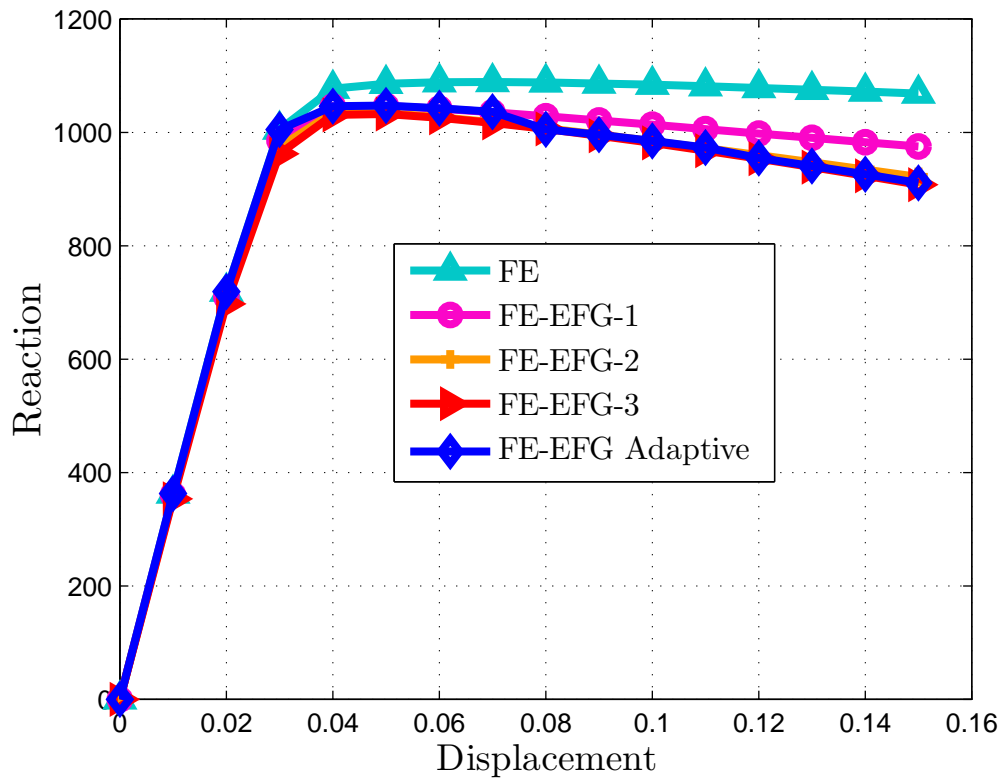


Figure 6.23: Reaction versus displacement for the three-dimensional plate with a hole problem

without adaptivity and plots for the reaction versus displacement are also shown in Figure 6.26. The solution with discretization 6.24(e) is considered as a reference solution in this case. Convergence of the adaptively coupled FE-EFGM case and all other cases to the reference solution is clear in Figure 6.26.

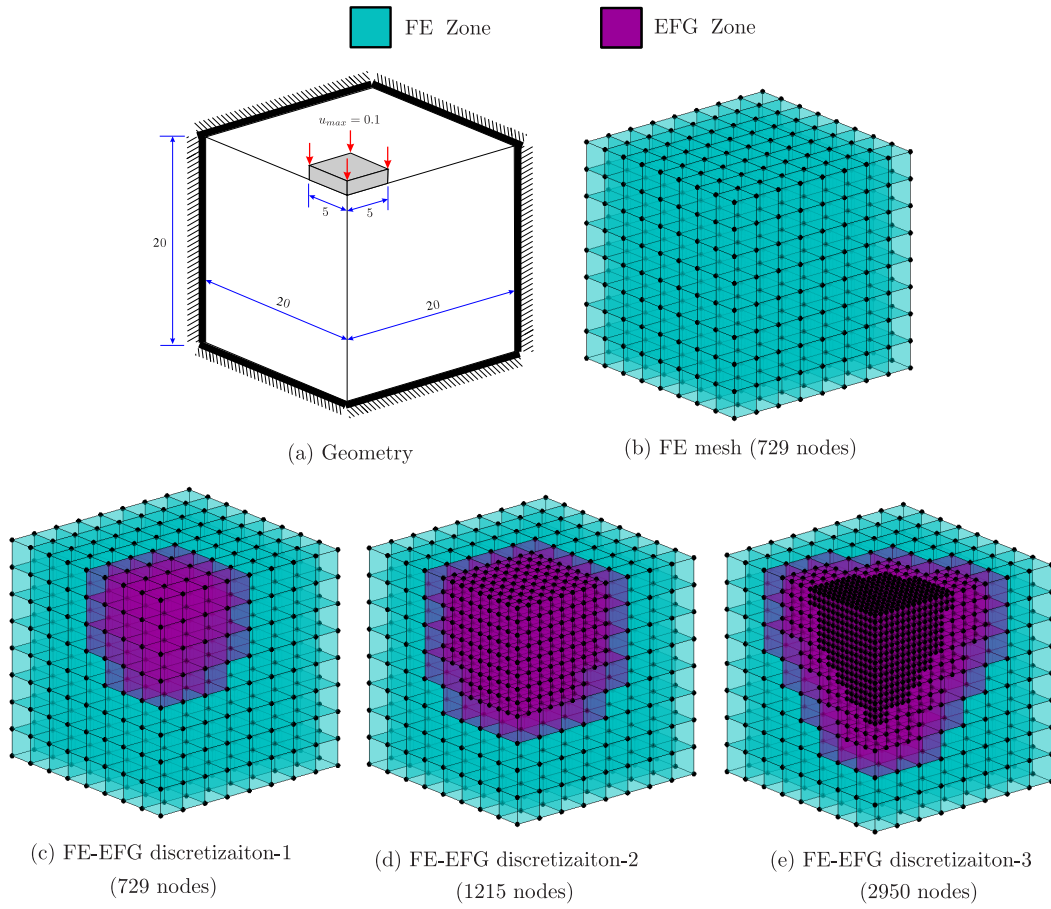
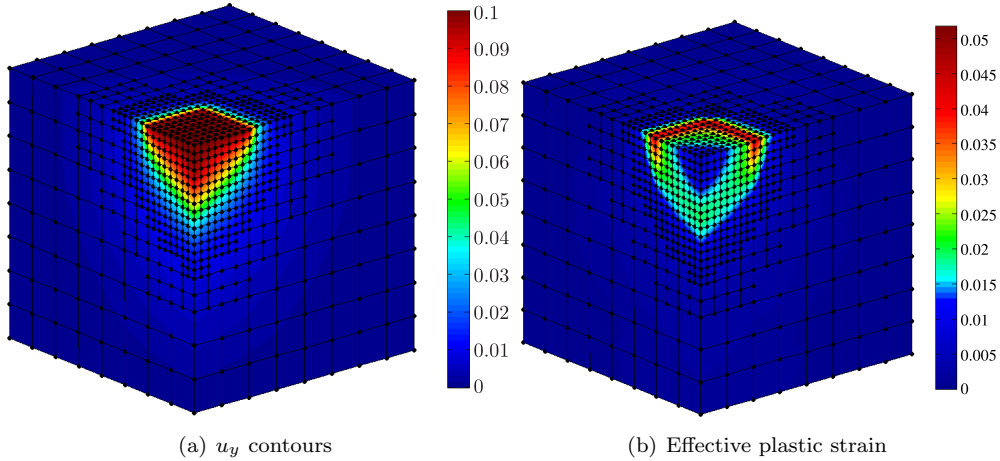


Figure 6.24: Step by step discretizations for the three-dimensional vertical cut problem

Figure 6.25: Displacement u_y and effective plastic strain contours for the three-dimensional vertical cut problem

6.7 Concluding remarks

In this chapter, a new numerical method is developed for two-dimensional linear elastic problems, in which initially, the whole of the problem domain is modelled using the FEM and during the analysis those elements, which violate a predefined error measure are automatically converted to an EFG zone.

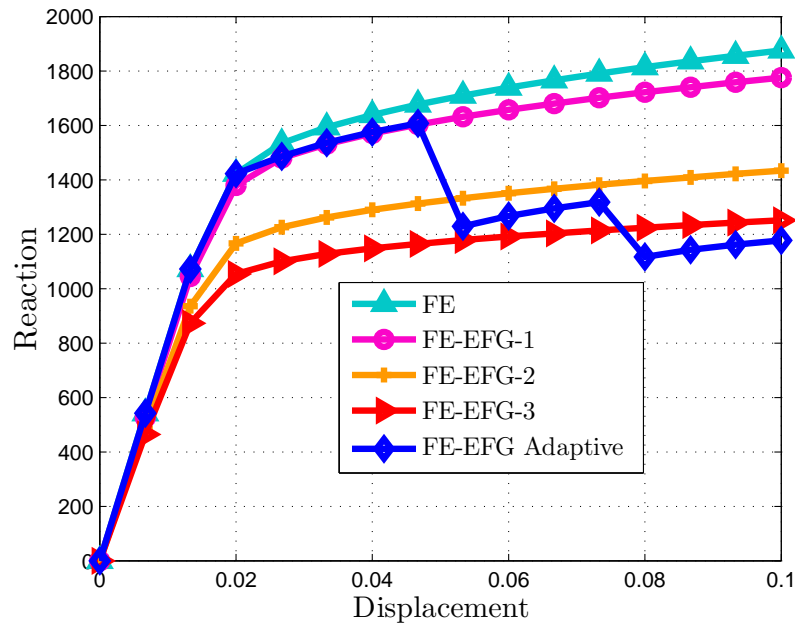


Figure 6.26: Reaction versus displacement for the three-dimensional vertical cut problem

The Zienkiewicz & Zhu error estimation procedure with the SPR method for stress recovery is used in the FE region to determine FEs requiring conversion to EFG background cells and Chung & Belytschko's [62] error estimation procedure is used in the EFG region for further adaptive refinement. A new refinement strategy is also suggested for the proposed adaptively coupled FE-EFGM. The full implementation and performance of the two-dimensional linear elastic adaptively coupled FE-EFGM algorithm is demonstrated with the help of two numerical examples. Results are also compared with the FEM uniform refinement case, and it is shown that the proposed method performs better. The adaptively coupled FE-EFG algorithm is also extended to the challenging three-dimensional nonlinear case with both material and geometrical nonlinearities. The total Lagrangian formulation is used instead of the updated Lagrangian to model finite deformation due to its computational efficiency. Incremental forms of the Zienkiewicz & Zhu error estimation procedure and the Chung & Belytschko's error estimation procedures are used in this case in the FE and EFG region of the problem domain. The implementation and performance of the three-dimensional nonlinear adaptively coupled FE-EFG algorithm is also demonstrated with two numerical examples. The performance of total and updated Lagrangian approaches is compared, and it is shown that total Lagrangian is computationally efficient.

Chapter 7

Parallel computations

7.1 Introduction

In almost every scientific field including academia and industry, complexity and size of the problems increases with time. For solution of these problems, computers with very large memory and very high computational power are required. Conventional sequential computers cannot handle these large and complicated problems due to their limited memory and limited computational power. Parallel computations have been used to solve these problems very conveniently, working on the “divide and conquer” strategy [302]. Using this strategy a very large and computationally demanding problem is divided into small manageable subproblems and each is then assigned to a different computer. Parallel computations are now commonly used in almost every scientific field, e.g. engineering, physics, geology, metrology, chemistry, biology, etc. Different types of parallel computers are now commonly available in both academia and industry, mostly based on either shared memory or distributed memory computer architectures.

The FEM is the most commonly used numerical technique for the solution of practical problems in different scientific fields. In the FEM, the memory requirement and corresponding computational demand increase with the increase in degrees of freedom, therefore, parallel computations have been used extensively in the FEM, the details of which can be found in many references, including [32, 49, 59, 116, 210, 320]. The domain decomposition method is the traditional way to divide a large FEM problem into small subproblems, in which load balancing and inter-processor communication are two important parameters. For the excellent performance of parallel programs, equal computational load must be assigned to each processor to minimize the waiting or idle time and at the same time, inter-processor communication should be minimized. Different strategies have been used in the literature for domain decomposition, including graph based techniques [152, 185, 304] and geometry based techniques [78, 146]. Graph based techniques are commonly used in the FEM for accuracy and efficiency.

As compared to the FEM, meshless methods are computationally expensive and are currently limited to two-dimensional and simple three-dimensional problems. Although, as compared to the FEM, meshless methods are ideal in modelling certain types of problems but their computational inefficiency restricts their use in practice to date to simple academic problems. In this chapter, a parallel algorithm is presented based on the distributed memory computer architecture model for linear and adaptive nonlinear EFGM, which is then extended to nonlinear adaptively coupled FE-EFGM. Total Lagrangian formulation is used to model finite deformation and the Prandtl-Reuss constitutive model is used to model elasto-plasticity. Codes have been developed from scratch by the author based on these algorithms in FORTRAN with other supporting libraries, including NAG, Voro++ and kd-tree. The MPI library¹ is used for inter-processor

¹Explained later in this chapter.

communication. Furthermore, open-source software packages, METIS¹ and MUMPS¹ have been used to automatically divide the problem domain and for the solution of the final system of linear equations in parallel, respectively. The codes have been run successfully on the Durham University high performance cluster Hamilton. Performance parameters used for the parallel programs are simulation time, speedup and efficiency. A final demonstration problem is also presented in this chapter to demonstrate the capabilities of the methods developed in this thesis so far to handle problems with very fine discretizations subjected to very large deformation and elasto-plasticity.

Parallel computing has been attempted in a number of references for the EFGM. In [302], parallel three-dimensional EFGM code was developed for linear-elastic problems. In this study Parallel computer was constructed by joining several low cost computers with a high-speed network and the MPI library was used for the inter-processor communication. Accuracy, run time, speedup and efficiency were used to measure the performance of the parallel program for benchmark numerical problems. A parallel code was developed for the sparse linear solver based on data decomposition strategy and was used for the EFGM in [281]. FORTRAN was used as a programming language and the MPI library was used for the inter-processor communication. The code was validated on three-dimensional heat transfer problems. In this case, the total simulation time, communication time, speedup and efficiency were used as measures of performance. One of the most commonly used domain decomposition methods in the FEM, the dual finite element tearing and interconnecting (FETI) [92], was used for the EFGM in [217].

In the RKPM, parallel computation has also been used in a number of references. In [305], a parallel code was developed for the RKPM to simulate three-dimensional bulk metal forming using the MPI library. Integration cells were divided among the processors using the domain decomposition method and nodes/particles were then duplicated accordingly. As compared to the FEM, the high communication cost in the case of meshless methods was also mentioned in this study. von-Mises constitutive model with linear isotropic hardening was used to model elasto-plasticity. Taylor bar impact and backward extrusion were used as test problems to show the performance of parallel algorithm and results were compared with the reference FEM approach using 8-node hex elements. For explicit dynamics analysis, a parallel code based on the MPI library was developed in [78]. Integration points were used for partitioning instead of the background cells because of the different number of nodes in the support of each Gauss point, which results in different computational load associated with each Gauss point. Nodes/particles were then duplicated accordingly. It was shown that the METIS partitions were almost perfectly balanced. It was also mentioned that the communication cost in the case of the parallel RKPM algorithm is larger than the corresponding parallel FEM algorithm. Three-point bending of a notched beam and three-dimensional shear band simulation in a tensile specimen were used as numerical examples to show the performance of this parallel code. In [106], a parallel RKPM code based on the distributed memory computer architecture was developed to solve viscous compressible flow problems. The code was run on the IBM SP supercomputer and results were presented for 1 to 16 processors. A new algorithm based on D'Alembert's principle was also presented to efficiently enforce the essential boundary conditions in parallel with minimum communication cost. In this new algorithm shape functions were modified in the preprocessing stage to satisfy the Kronecker delta property on the essential boundary nodes. ParMETIS, a parallel implementation of METIS, was used for the domain decomposition. The partitioning performed was based on Gauss points. Measures of relative speedup and efficiency were used due to memory problems, i.e. the simulation time of the two processors was taken as a reference. The results reported were based on the average of five runs. In this case, flow over a NACA 7012 airfoil was used as a test problem.

¹Explained later in this chapter.

Parallel computing has also been used for other meshless methods, e.g in [270] for a free mesh method to simulate the flow problems and in [214] for the SPH method for impact simulation. A parallel code was also developed based on the shared memory architecture using OpenMP for the point interpolation method (PIM) for three-dimensional metal forming simulation in [133].

This chapter is organized as follows. Introduction to basic parallel computer architecture is given in §7.2. Inter-processor communication for the distributed memory computer architecture using the MPI library is briefly described in §7.3. Performance parameters of the parallel computer program, including speedup and efficiency are introduced in §7.4. Open-source software packages for the automatic domain decomposition (METIS) and solution of the final system of linear equations in parallel (MUMPS) are briefly explained in §7.5 and §7.6 respectively. Parallel linear elastic EFGM algorithm and its implementation are described in §7.7, and then validated with two numerical examples in §7.7.1. The parallel linear elastic EFGM algorithm is extended to parallel adaptive nonlinear EFGM case in §7.8, and is validated with two three-dimensional numerical examples, one with adaptivity and another without adaptivity in §7.8.1. A parallel nonlinear adaptively coupled FE-EFGM code is also developed and is validated using a problem with very fine discretizations subjected to finite deformation and elasto-plasticity in §7.9. Final concluding remarks are given in §7.10.

7.2 Parallel computer architecture

For efficient parallel programming, some knowledge of computer architecture is essential. In this section, computer architecture is explained very briefly, the detail of which can be found in a number of specialized references, e.g. [32, 247, 261]. Parallel computer architecture is based on the conventional sequential von Neumann architecture, which mainly consists of three main parts, i.e. central processing unit (CPU), main memory and an interconnection network to connect the CPU and memory. A block diagram of the von Neumann architecture is shown in Figure 7.1. The CPU is further divided into two main parts, i.e. a control unit and an arithmetic and logical unit (ALU). The control unit determines the sequence of instructions to be executed, and the ALU is used to actually execute these instructions. The function of the main memory is to store instructions and data, while the interconnect is used to transfer data and instructions between memory and CPU. The von Neumann architecture can execute only a single instruction on a single set of data. The commonly used classification of parallel computer architecture

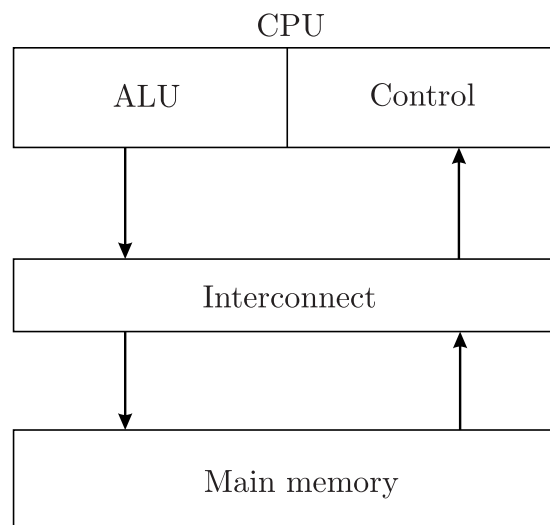


Figure 7.1: The von Neumann conventional sequential computer architecture [247]

is based on Flynn's taxonomy [97], which depends on the data and instruction stream, a computer can simultaneously handle. In this classification, four letter mnemonics are used, in which the first two refer to the instruction stream, and the last two refer to the data stream. The following are four types of systems based on Flynn's taxonomy.

Single instruction single data (SISD) systems

In these systems, a computer can execute a single instruction at a time and can read or write one data item from or to the memory at a time. The conventional sequential von Neumann architecture belongs to this type of system.

Single instruction multiple data (SIMD) systems

In these systems, the same instruction is executed on multiple data sets. Theoretically a SIMD system consists of a CPU with one control unit and multiple ALUs. The control unit will broadcast the instruction to each ALU, which will be executed on different sets of data. In the conventional SIMD systems, all the ALUs should perform the same function at the same time, as these have no memory to store the instructions. SIMD systems are very good for data-parallelism, in which a large array is divided between the processors and approximately the same instructions are applied to each data subset.

Multiple instruction single data (MISD) systems

In these systems, a single data stream is subjected to different instructions on different processors. This is useful in situations, where different operations need to perform on a single set of data. These systems require a special processor for each operation, and are not available commercially at the moment [32].

Multiple instruction multiple data (MIMD) systems

In these systems, different instructions can be executed at the same time on different streams of data. The MIMD architecture thus consists of separate independent processors, each with its own control unit and ALU. Each of the MIMD processors can operate independently at its own pace and can execute different instructions at the same time. Almost all commercial parallel computers belong to this class of systems. MIMD systems are further classified into two types depending on the memory usage, i.e. shared memory systems, which are also called multiprocessors, and distributed memory systems, which are also called as multi-computers. The details of these two types of systems are given as follows.

Shared memory systems

In the shared or global memory system architecture, independent processors are attached to a common memory by an interconnection network. In these systems, each processor has direct access to each memory location in the shared memory. As compared to the distributed memory systems, in these systems, there is no need for data duplication. Communication cost in this case highly depends on the number of processors, i.e. in the case of a small number of processors the communication is very fast but increasing the number of processors will slow down the communication. The block diagram of the shared memory system is shown in Figure 7.2.

Distributed memory systems

In the distributed memory system architecture, each processor, which is also called a node, has its own memory. For each node, the data required for its local processing is stored in its local memory. In these systems, message passings are used to transfer data from the local memory of one node to the local memory of another node, in which one computer sends the message, and the other receives it. The communication in this case is performed through the interconnecting network.

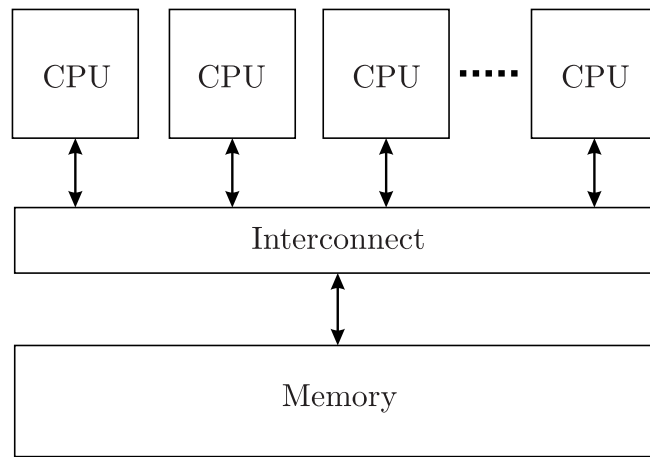


Figure 7.2: Block diagram of the shared memory system architecture [247]

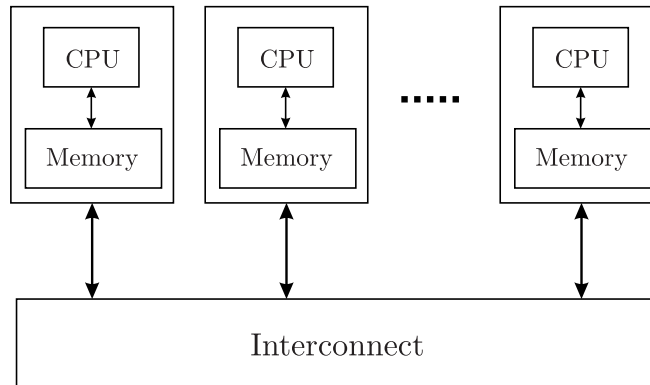


Figure 7.3: Block diagram of the distributed memory system architecture [247]

7.3 Message passing interface (MPI)

In distributed memory parallel computer systems, each processor has its own local memory. Each processor can only access its own local memory directly and if data are required from the local memory of another processor, message passing is used. In message passing one processor sends a message and the other receives it. This message passing is normally provided by the communication libraries, where the processor calls the library's functions or subroutines to start the communication. These communication libraries can be either hardware specific or portable, which can be used on different platforms. The message passing interface or MPI is the most commonly used communication library for distributed memory parallel programming, although there are other alternatives, e.g. parallel virtual machine (PVM) [107].

A very brief introduction to the MPI library and corresponding MPI program structure is given here, detailed descriptions of which can be found in a number of specialized references, e.g. [113–115, 247, 261, 284]. MPI is the standardization of the message passing aspects of the parallel computing, which was first introduced by manufacturers and users in the supercomputing conference in 1992 [302]. As compared to other communication libraries, MPI is used commonly due to its three main characteristics, i.e. portability, efficiency and functionality [114, 282]. The commonly used programming languages for scientific computations, e.g. FORTRAN, C and C++ can support MPI, in which the MPI program is first compiled with a standard compiler and is then linked with the MPI library. The MPI library is only responsible for the communication part of the code and has nothing to do with the rest of the

computation. In transferring data between the processors, a complete specification of the data is required, i.e. its location within the memory, its data type and number of data items.

Communication between the processors can be broadly classified as “point to point” and “collective”. Only two processors take part in the point to point communication, in which one processor sends a message, and the other receives it. The `"mpi_send"` and `"mpi_recv"` are the two main subroutines in the MPI library used for point to point communication. In collective communication, more than two processors take part, which is normally used to distribute the input data from the host to all other processors involved in the computations. It can also be used to collect the final results from these processors. The `"mpi_bcast"` and `"mpi_reduce"` are the two MPI subroutines used for collective communications. While using the MPI library, each processor is identified by its rank, which is a positive integer value, ranging from 0 to $n - 1$, where n is the total number of processors involved in the computation. Furthermore, the tag used to identify a message and a group of processors involved in message passing is identified with the communicator. In the MPI library, all the processors involved in the computation belong to the default communicator `"mpi_comm_world"`. Header file `"mpif.h"` must be included at the start of the MPI program, which provides links to all the relevant MPI subroutines and data types. The MPI program must be started and terminated with `"mpi_init"` and `"mpi_finalize"` respectively. Finally, the MPI program is executed using the command

```
mpirun -n 20 exe_name
```

which will execute the program `"exe_name"` on 20 processors, for instance.

7.4 Performance of parallel programs

Computational efficiency is one of the main purposes of writing a parallel program. Therefore, run time of a parallel program must be compared with the corresponding run time of a sequential equivalent. Speedup and efficiency are the two measures, which are commonly used to evaluate the performance of a parallel program, the detail of which can also be found in [114, 247, 261, 282]. Run time or simulation time and cost of parallel programming are also important parameters required for the proper understanding of speedup and efficiency. In the following, before giving proper definitions of speedup and efficiency, run time and cost of parallel programming are introduced.

Run time or simulation time

For a sequential program, run time is the time between the start and end of simulation, while for a parallel program, it is the time between the start of simulation and end of computation on the last processor.

Cost of the parallel computation

The cost of parallel computation C_n is the total amount of work done by all processors, involved in the computation, i.e.

$$C_n = nT_n, \quad (7.1)$$

where n is the number of processors and T_n is the time the program spends on n processors (parallel run time). Cost of parallel computing is also known as work or processor-runtime product [261].

Speedup

Speedup S_n is the ratio of time a parallel program spends on one processor T_1 to the time the same program spends on n processors T_n , i.e.

$$S_n = \frac{T_1}{T_n}. \quad (7.2)$$

Speedup shows a reduction in the simulation time of a parallel program, when it runs on n processors. In an ideal situation $S_n = n$ but generally $S_n \leq n$ due to additional work done by a parallel program, known as overhead. This overhead consists of communication between the processors, synchronization, idle time for some processors due to unbalanced load and redundant calculation to avoid data transfer. Furthermore, it is impossible to 100% parallelize a code due to its sequential parts, especially the input and output part.

Efficiency

Efficiency E_n is the ratio of the total amount of work done by all the processors to the time or work done by a single processor or alternatively it is defined as the ratio of the number of processors to speedup, i.e.

$$E_n = \frac{nT_n}{T_1} = \frac{n}{S_n}. \quad (7.3)$$

In an ideal situation $E_n = 1$ but generally $E_n \leq 1$.

7.5 Automatic partitioning of the problem domain with METIS

Intelligent mesh partitioning between processors is one of the most important components of parallel programming in numerical simulation and the goals are load balancing and minimization of inter-processor communication. In the case of the EFM, for perfect load balancing, an equal number of elements should be assigned to each processor, and the purpose is to reduce the idle or waiting time for some processors, results from the unbalanced load. Minimizing adjacent elements of the mesh partitions leads to minimum inter-processor communication.

METIS [151], which is an open-source software package for partitioning of unstructured graphs, partitioning meshes and computing fill-reducing ordering of sparse matrices is used here to uniquely divide the background cells among the processors involved in the computations. As compared to other alternatives, METIS provides high-quality partitions with high efficiency. METIS partitioning are based on graph partitioning algorithms, in which an FE mesh (background cells in our case) is initially converted to a graph, in which each vertex of the graph represents a FE. After partitioning, the graph is then converted back to the FE mesh. METIS partitioning is based on multilevel graph partitioning [153, 154], which is an efficient way to intelligently divide very large graphs as compared to conventional partitioning, in which the partitioning is performed on the original graph. A multilevel partitioning algorithm with a sample graph is shown in Figure 7.4, which consists of three-phases, i.e. coarsening, partitioning and refinement or uncoarsening. During the coarsening phase, the size of graph reduces in steps by collapsing its vertices, and METIS uses algorithms that make it easier to find high-quality partitions. After the coarsening phase, partitioning is then performed on the reduced graph and due to its small size, this step is normally very fast. In the refinement or uncoarsening phase, vertices are uncollapsed step by step until we get the original graph. In the uncoarsening phase, METIS concentrates on the portions of the graph, which are close to the partitions' boundaries and the goal is to reduce inter-processor communication. METIS provides a command line stand alone program as well as an interface to other programming languages, e.g. FORTRAN, C.

7.6 MULTifrontal Massively Parallel Solver

The MULTifrontal Massively Parallel Solver (MUMPS) [2] is an open-source direct solver for the solution of the general sparse system of linear equations, i.e. $\mathbf{K}\mathbf{u} = \mathbf{f}$. The input matrix \mathbf{K} can be symmetric positive definite, general symmetric or unsymmetric. MUMPS supports a variety of ways to enter the input matrix \mathbf{K} , i.e. it can be in assembled format or in unassembled or elemental format. Furthermore,

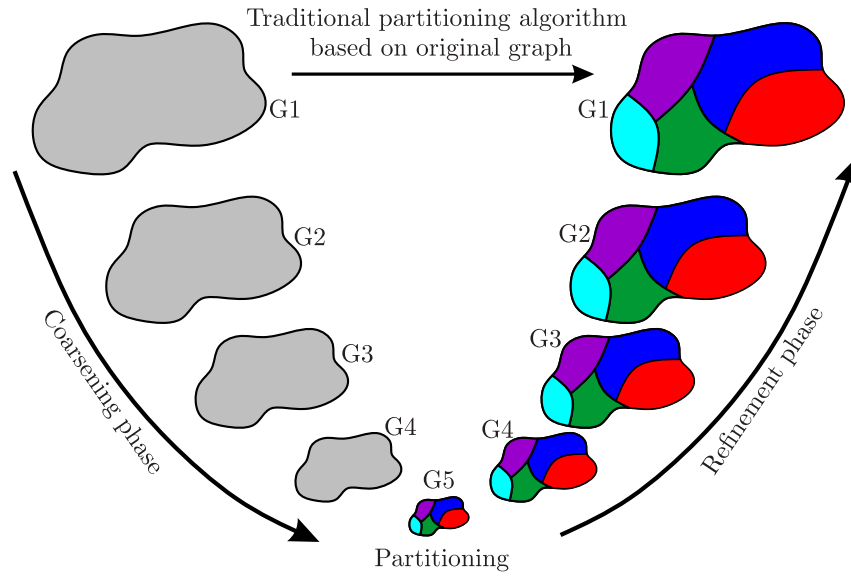


Figure 7.4: Traditional and multilevel partitioning algorithms [151]

using the assembled format, MUMPS can accept the \mathbf{K} matrix in a centralized or distributed format. In the centralized matrix format, only one global \mathbf{K} is assembled on the host, while in the distributed format, individual \mathbf{K} matrices are assembled on each processor. In the centralized matrix input format, MUMPS itself divides and distributes the matrix to other processors. Distributed \mathbf{K} matrix input format is used in this research due to a very large memory requirement on the host in the case of the centralized matrix input format to store one global \mathbf{K} matrix. Finally, MUMPS can support different arithmetic, e.g. complex, single and double precision. MUMPS is written in FORTRAN but can also be used with other common scientific languages, e.g. C, MATLAB and Octave. MUMPS is available in both parallel and sequential format. For the sequential version only the BLAS library is required but for the parallel version, BLAS, BLACS, MPI and ScaLAPACK libraries are required.

7.7 Parallel algorithm for linear elastic problems

The parallel algorithm for the EFGM for linear elastic problems is described here, which will be extended to the adaptive nonlinear case in the next section. Before explaining the parallel EFGM algorithm itself, first a comparison is given of the computation involved in the parallel EFGM as compared to the parallel FEM algorithm. Sample partitions for two processors for the FEM and the EFGM are shown in Figures 7.5(a) and 7.5(b) respectively. In the FEM case, the boundary nodes between the two processors are required on both processors for the calculation of the shape functions and corresponding shape function derivatives. These nodes are shown as solid circles in Figure 7.5(a) and should be duplicated on both processors. These shared nodes represent the communication cost, i.e. the information required to be transferred between the two processors. As compared to the FEM, in the EFGM case, more nodes are required for the calculation of shape functions and corresponding shape function derivatives as shown in Figure 7.5(b), therefore more nodes need to be shared between the processors in the EFGM case. As Figure 7.5(b) shows, the influence domains of nodes shown in red covers Gauss points belong to the other processors, therefore, these nodes must be duplicated. This increase in the shared nodes as compared to the FEM case, increases the communication cost in the EFGM case.

The flow chart for the parallel algorithm for the EFGM for linear elastic problems is shown in Figure 7.6, all of which has been implemented from scratch by the author in FORTRAN with NAG, MPI, METIS,

MUMPS, KDTree and Voro++ supporting libraries. The two computationally expensive parts, i.e. assembly of the stiffness matrix and the solution of the final system of linear equations, are performed in parallel. At the start of the analysis, the MPI library is initialized and the problem is redundantly set up on each processor involved in the computation, to reduce the inter-processor communication. The problem setup includes the definition of nodal coordinates, background cells, Gauss points and influence domains. Input data is then prepared for METIS for domain decomposition, the detail of which can be found in [151]. METIS library subroutine "METIS_PartMeshDual" is used here for this purpose, which directly divides the background integration cells into the user defined number of subdomains. The output of "METIS_PartMeshDual" is "epart", which is a vector of length equal to the number of background cells where each entry shows the processor number to which the background cell belongs. Kd-tree with background mesh algorithm (the details of which can be found in Appendix-B) is used next to search nodes in the support of each Gauss point. The next step is the assembly of stiffness matrices on each processor using the MUMPS distributed assembled matrix format. The size of the stiffness matrix on each processor depends on the nodes, which are in support of its Gauss points, e.g. in Figure 7.5(b), shared nodes must be included in these nodes for both processors. The stiffness matrix on each processor is then calculated and assembled. Force vector \mathbf{f} is assembled only on the host or master processor. Input data are then prepared for MUMPS for the solution of the final system of linear equations, the detail of which can be found in [2]. Finally, output data, e.g. displacements and stresses, are calculated before finalizing the MPI library and end of analysis.

7.7.1 Numerical examples

Two two-dimensional linear-elastic numerical examples are now given to demonstrate the implementation and performance of the linear elastic parallel algorithm.

Two-dimensional beam problem

The first numerical example is a two-dimensional cantilever beam problem, the geometry, loading, boundary conditions and material properties for which are given in §2.8. This problem is solved with two different discretizations, i.e. one coarse one and one fine. In the fine discretization, 6601 nodes (DOFs=13202) and 6400 background cells are used, while in the coarse discretization, nodes and background cells are decreased to 1701 (DOFs=3402) and 1600 respectively. (4×4) Gauss points are used in each background cell and $d_{max} = 3.0$ is used as scaling parameter for the domain of influence. The problem was run on the Hamilton cluster [1] with 1-25 processors. Sample METIS partitions for the discretization with 13202 DOFs for 2, 5, 10, 16, 20 and 25 processors are shown in Figures 7.7(a-f) respectively. In these figures, each color represents a separate METIS partition, for which the calculations are performed on a different processor. A comparison between the selective deformed nodes and their analytical solution are shown in Figure 7.8(a), while a comparison between the numerical and analytical neutral axis deflections are shown in Figure 7.8(b). The results obtained in this case are in excellent agreement with the analytical solutions, which shows the correct implementation of the parallel linear elastic algorithm.

The performances of this problem on the Hamilton cluster are shown in Figure 7.9. The timing results for the 13202 DOFs case are shown in Figure 7.9(a). The timing reported here for the stiffness matrix includes calculation of the shape functions and corresponding shape function derivatives and calculation and assembly of the stiffness matrix. The time spent on the solution of the final system of linear equations using MUMPS is tagged as "MUMPS" in Figure 7.9(a). The total time reported is the sum of stiffness matrix and MUMPS times. Here the time spent in the solution of the system of linear equations is very small as compared with the time spent in the assembly of the stiffness matrices. It is clear from Figure 7.9(a), that increasing the number of processors reduces all the three reported times. The timing

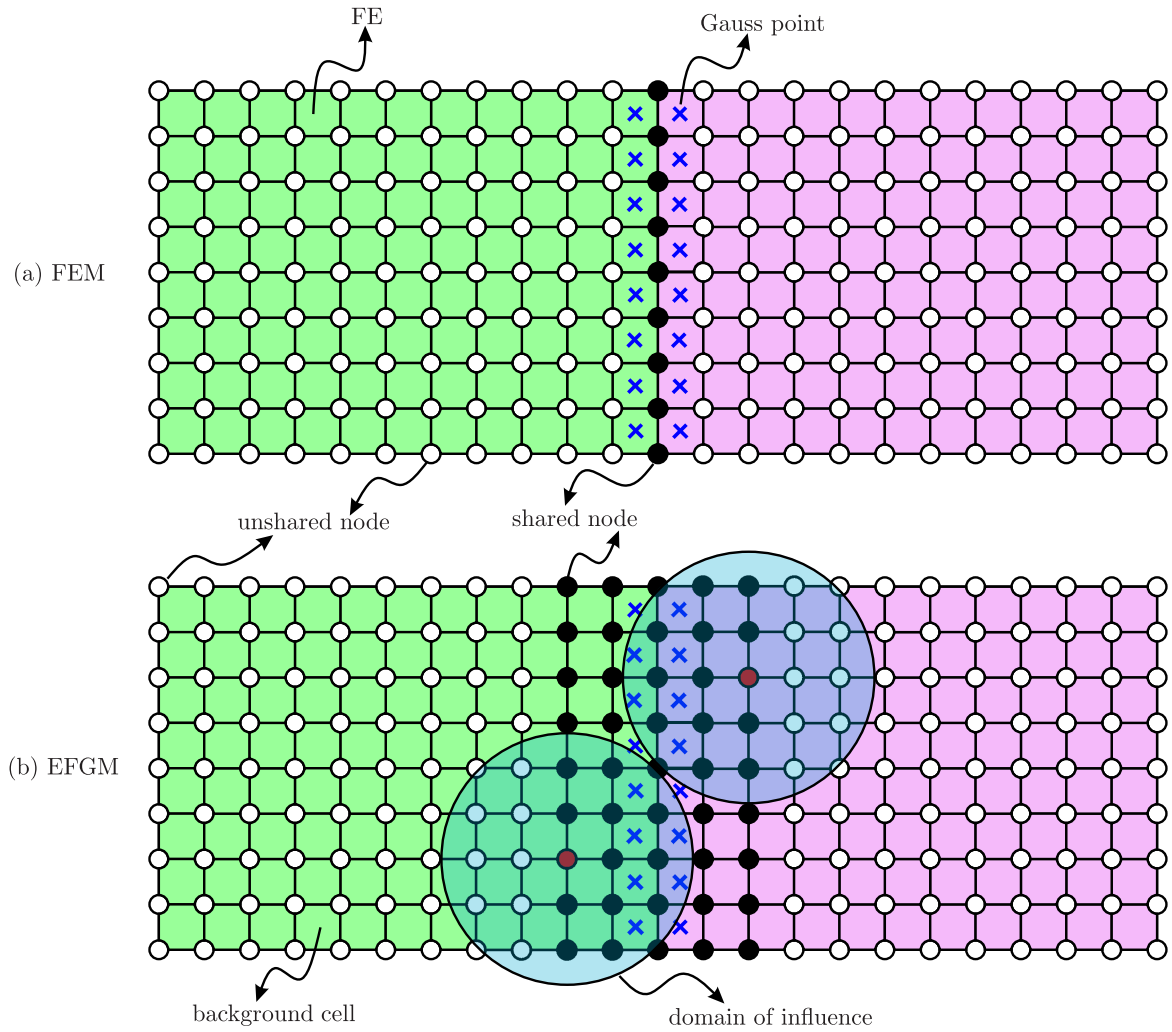


Figure 7.5: Sample partitions for two processors

results for the 3402 DOFs case are also shown in Figure 7.9(b) showing the same decreasing trend with increasing the number of processors as shown in Figure 7.9(a). A comparison between the actual speedup and ideal speedup for both the discretizations versus the number of processors are shown in 7.9(c). For this problem the fine discretization performs relatively better than the coarse discretization. For 25 processors, a speedup of 12 is obtained for the fine discretization, while a speedup of 9.4 is obtained for the coarse discretization. A comparison between the actual and ideal efficiencies for both the discretizations versus the number of processors is shown in Figure 7.9(d). For 25 processors, 48.0% and 37.6% efficiencies are recorded for fine and coarse discretizations respectively.

Hole in an infinite plate problem

The second linear-elastic numerical example is a two-dimensional infinite plate with a central hole, the geometry, loading, boundary conditions and material properties for which are already given in §2.8. In this case again the problem is solved with two different discretizations. The first is fine one with 7381 nodes (14762 DOFs) while the second discretization is coarse one with 3691 nodes (3782 DOFs). (4×4) Gauss points are again used in this problem and the scaling parameter for the domain of influence used is $d_{max} = 3.5$. The problem was run on the Hamilton cluster using 1-30 processors. Sample METIS partitions of the fine discretization for 2, 6, 12, 20 and 30 processors are shown in Figures 7.10(a-e).

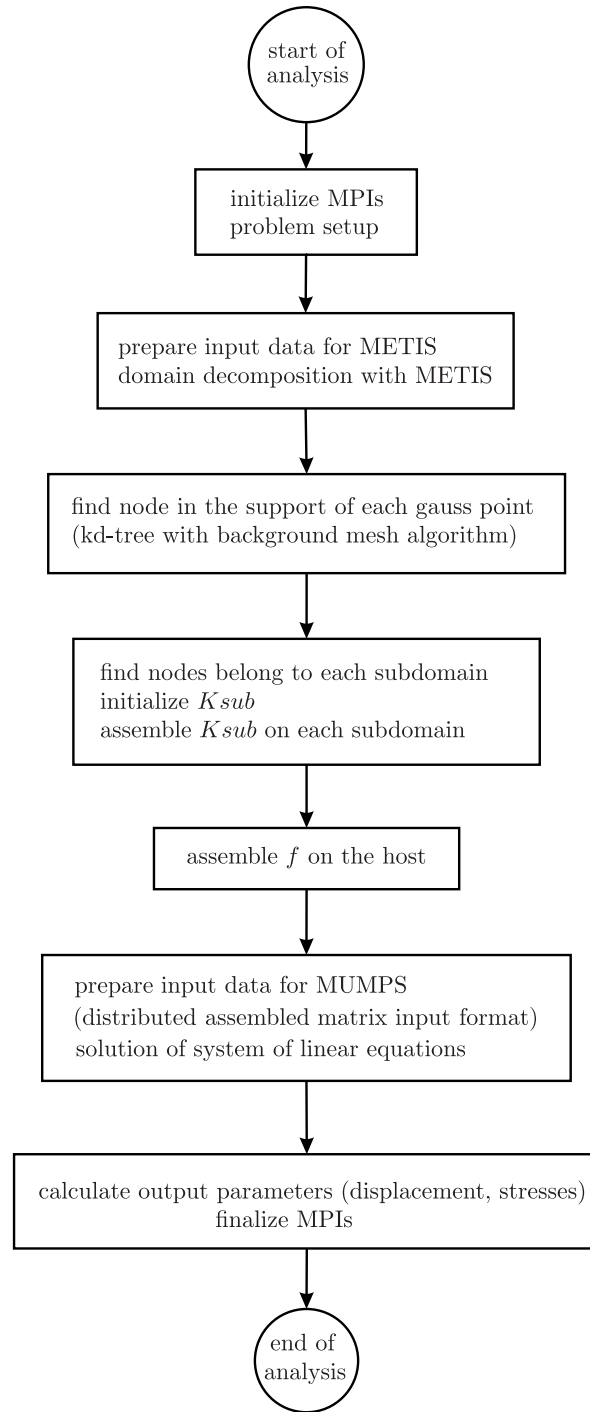


Figure 7.6: Linear elastic parallel algorithm

Comparison of the deformed nodes with the numerical displacements, and their corresponding analytical solutions are shown in Figure 7.11, which are in excellent agreement with each other and show again the correct implementation of the parallel linear elastic algorithm.

The performances of this problem on the Hamilton cluster are shown in Figure 7.12. The timing results for the fine discretization with 14762 DOFs are shown in Figure 7.12(b) and for the coarse discretization with 3402 DOFs are shown in Figure 7.12(b). Here again, in both cases all the three reported times, i.e. time spent in the calculation and assembly of stiffness matrices, time spent in the solution of the

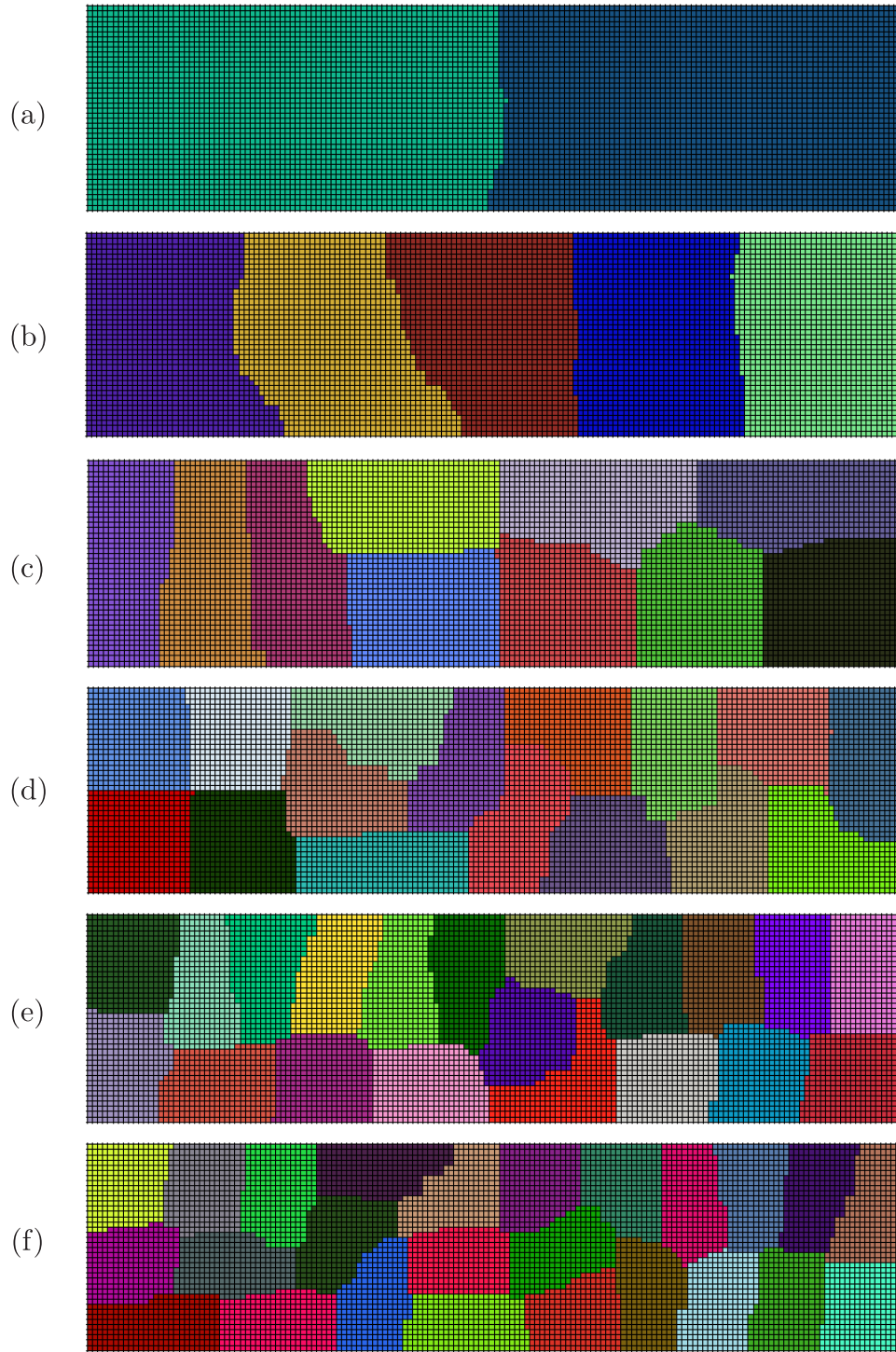


Figure 7.7: Selective Metis partitions for the discretization with 13202 DOFs for the two-dimensional beam problem

final system of equations with MUMPS, and total time decreases with an increase in the number of processors. Comparison between the actual speedup for both fine and coarse discretizations and ideal speedup are shown in Figure 7.12(c). Similarly, comparison between the actual efficiencies for fine and coarse discretizations and ideal efficiency are shown in Figure 7.12(d). For the fine discretization, a speedup and efficiency of 8.16 and 27.2% respectively are recorded for 30 processors, while for the same

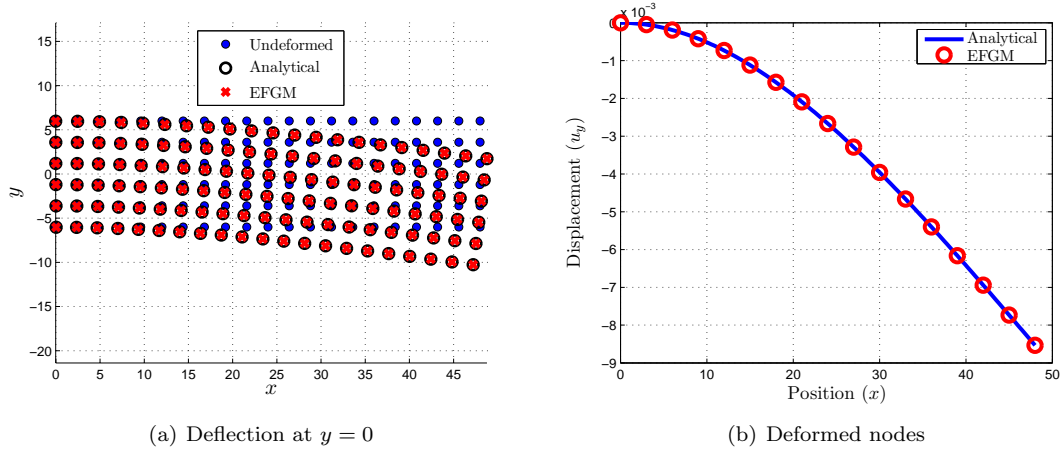


Figure 7.8: Deflections for the selective nodes for the two-dimensional beam problem

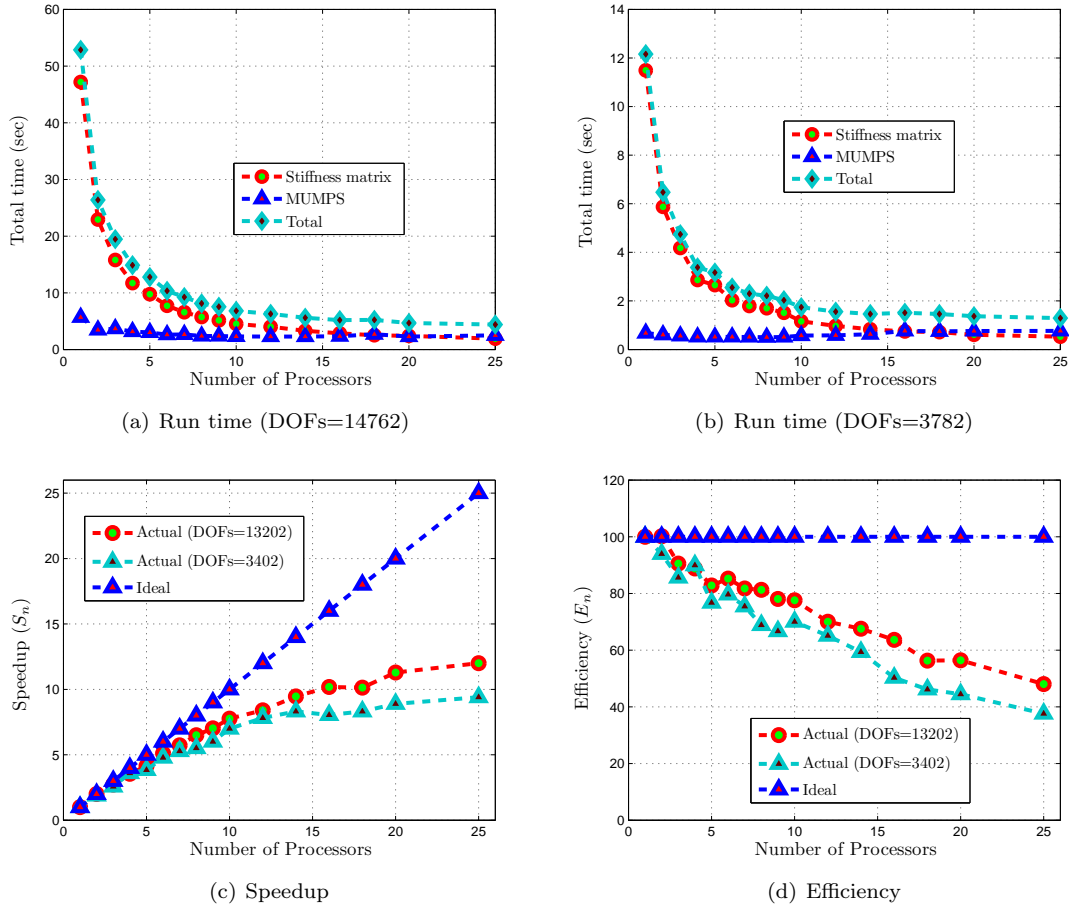


Figure 7.9: Performance on the Hamilton cluster [1] using 1-25 processors for the two-dimensional beam problem

number of processors, for the coarse discretization, speedup and efficiency recorded are 10.0 and 30.47% respectively.

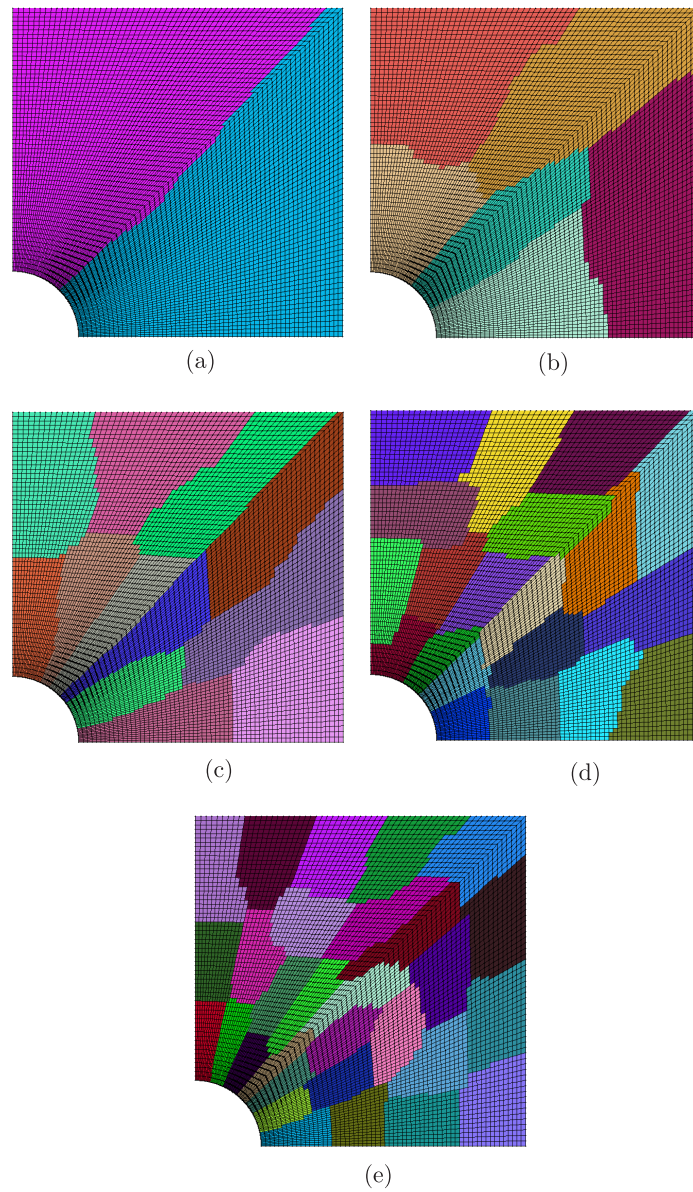


Figure 7.10: Selective Metis partitions for the discretization with 14762 DOFs for the hole in an infinite plate problem

7.8 Parallel algorithm for adaptive nonlinear problems

In this section, the parallel algorithm described previously for the EFGM for linear elastic problems is extended to adaptive nonlinear problems with both material and geometrical nonlinearities. The flow chart of the parallel algorithm in this case is shown in Figure 7.13. A total Lagrangian formulation is used here to model finite deformation and the Prandtl-Reuss constitutive model is used to model elastoplasticity, the details of which are already given in §3.3.2 and §3.2.4 respectively. Here the start of the algorithm is almost the same as given for the linear elastic case, i.e. the definition of the problem and METIS partitioning. Nodal influence domains for analysis and projection are calculated on the host and are broadcast to all other processors involved in the computation. Kd-tree with background mesh algorithm is then used to calculate nodes in support of each Gauss point for analysis and projection and is also used to find nodes in support of each node for the calculation of nodal strains and stresses

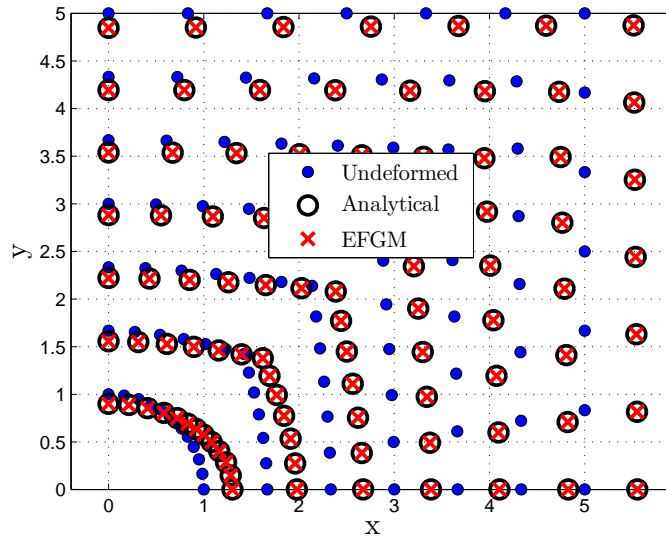
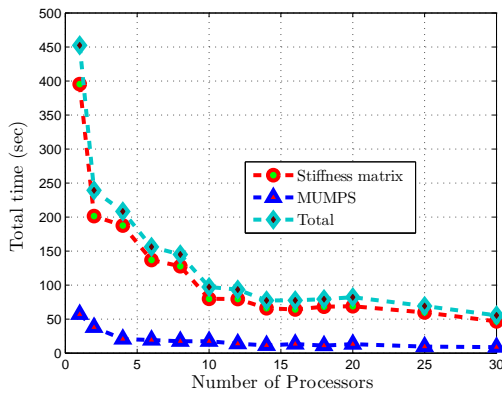
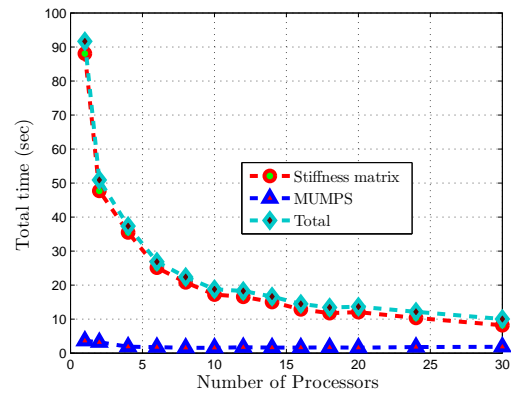


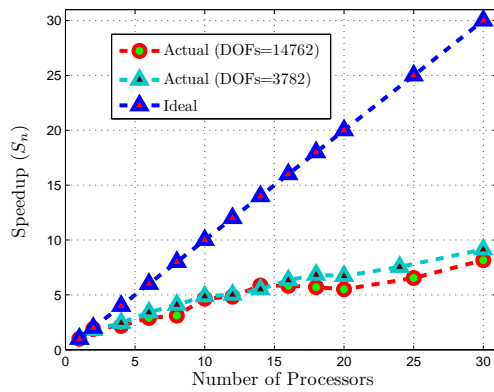
Figure 7.11: Displacements for selective nodes for the hole in an infinite plate problem



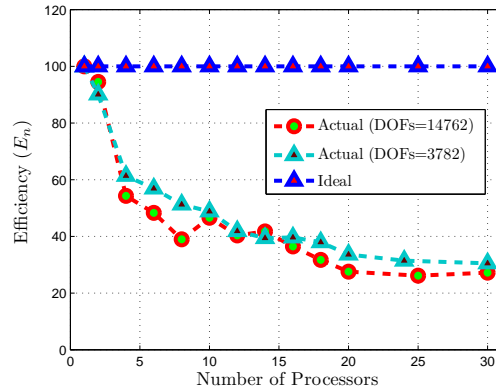
(a) Run time (DOFs=13202)



(b) Run time (DOFs=3402)



(c) Speedup



(d) Efficiency

Figure 7.12: Performance on the Hamilton cluster [1] using 1-30 processors for the hole in an infinite plate problem

for the calculation of the Chung & Belytschko error estimates. For each processor, shape functions and corresponding shape function derivatives are calculated for analysis at each Gauss point belonging to it and are stored in separate files, i.e. each file belongs to one processor. Similarly, the same procedure is repeated for the calculation of shape functions at each Gauss point for projection, and the calculation of shape functions and corresponding shape function derivatives at each node for the calculation of nodal strains and stresses. Determination of the size of the stiffness matrix on each processor and calculation and assembly of the stiffness matrix is then performed in parallel in the same way as explained previously for the linear elastic case.

For load step n , systems of linear equations are solved in parallel with MUMPS, using the distributed assembled matrix input format. Internal and reaction forces are calculated next in parallel separately on each processor and are combined on the host to calculate the out-of-balance force, which is then used to control the Newton-Raphson iteration. After convergence of the Newton-Raphson iterations, the next step is the calculation of nodal strains and stresses and error estimation. METIS also gives nodal partitions, which are used here to calculate the nodal strains and stresses in parallel. The error is also estimated in parallel here because the Chung & Belytschko error estimation procedure works cell-wise. Therefore, error is calculated separately on each processor and they are then combined to calculate the global error. The global error is then used to control the adaptive process, as already explained in §4.3. Transfer of the path dependent variables both for the nodes and for the Gauss points are also performed here in parallel. After each refinement, METIS is used for automatic domain decomposition of the problem domain.

7.8.1 Numerical examples

A three-dimensional nonlinear problem with both material and geometrical nonlinearities is now given, to demonstrate the implementation and performance of the parallel nonlinear adaptive algorithm. The problem solved here is a three-dimensional plate with a hole problem subjected to unidirectional loading. The same problem is also solved in §4.4.1, where its geometry, boundary conditions and material properties are also given. Due to symmetry, only one-eighth of the problem is solved. The problem is solved here with the same code using two different strategies, i.e. initially it is solved without adaptivity and then with adaptivity. In the “without adaptive” case, all the subroutines related with adaptivity are switched off, which includes calculation of nodal stresses, error estimation, refinement and data transfer between the consecutive discretizations. In the following these two cases are described separately.

Without adaptivity

A total displacement of 0.15 units is applied to the top face of the plate in 15 equal steps. The scaling parameter used here for the domain of influence is $d_{max} = 1.5$ and the problem is solved with two different discretizations. The first discretization is fine with 6075 DOFs and the second is relatively coarse with 2793 DOFs. The problem was run on the Hamilton cluster with 1-8 processors. Sample METIS partitions for 2, 4, 6 and 8 processors for the 6075 DOFs case are shown in Figures 7.14(a-d) respectively, while for the 2793 DOFs case are shown in Figures 7.15(a-d) respectively. The same problem is also solved with the FEM, using a relatively fine eight-node hexahedral mesh with 18759 DOFs to serve as a reference solution, for which the mesh is shown in Figure 7.16(a). A comparison between the reaction versus displacement for both the fine and coarse discretizations and the reference FEM is shown in Figure 7.16(b), which are in a very good agreement and demonstrates the correct implementation of the parallel adaptive nonlinear algorithm.

The performance on the Hamilton cluster is shown in Figure 7.17. Simulation time versus the number of processors for both the discretizations are shown in Figure 7.17(a). The timings reported here are the

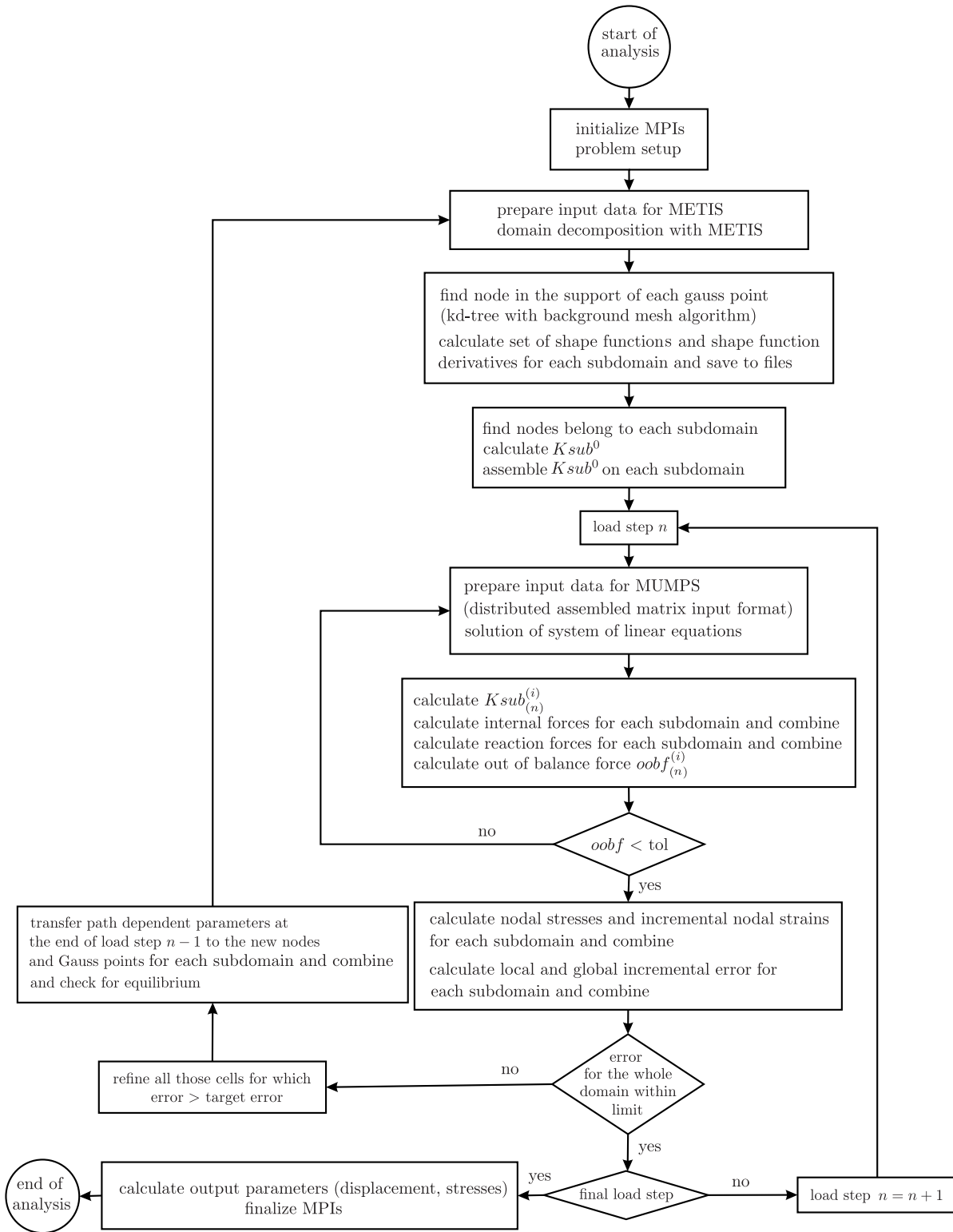


Figure 7.13: Nonlinear adaptive parallel algorithm

total simulation times. It is clear from Figure 7.17(a) that for this problem with a fine discretization, the code spends 1008 seconds on one processor but on eight processors, it spends only 298 seconds and in the case of the coarse discretization, it spends 277 and 91 seconds on one and eight processors respectively, which are a significant reduction in simulation time. A comparison between the actual speedup versus

the number of processors for both discretizations and the ideal speedup is shown in Figure 7.17(b). For eight processors a speedup of 3.38 and 3.04 are obtained for fine and coarse discretization respectively. A comparison between the efficiencies for both discretizations and the ideal is shown in Figure 7.17(c). For eight processors 42.26% and 37.95% efficiencies are obtained for fine and coarse discretizations respectively. For this problem, the fine discretization performs better than the corresponding coarse one.

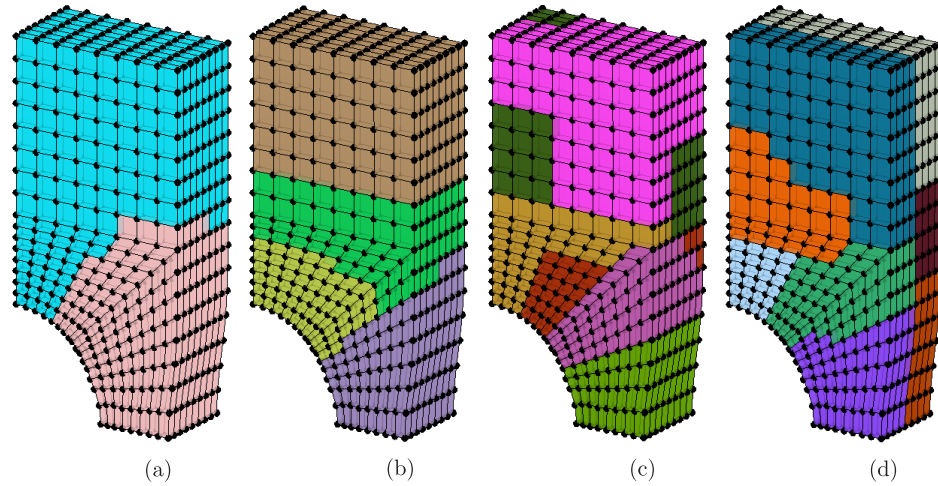


Figure 7.14: Selective Metis partitions for 6075 degrees of freedom for the 3D plate with a hole problem

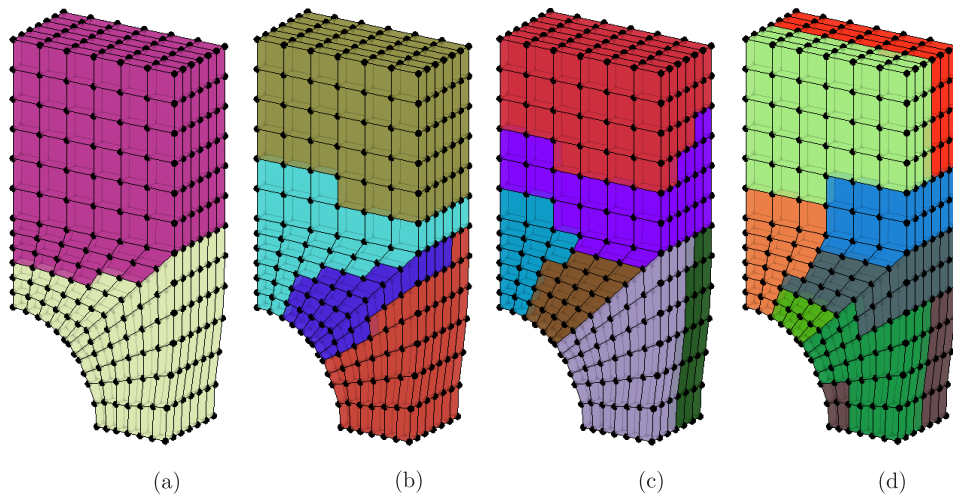


Figure 7.15: Selective Metis partitions for 2793 degrees of freedom for the 3D plate with a hole problem

With adaptivity

In the adaptive case, a total displacement of 0.5 units is applied to the top face of the plate in 20 equal load steps. The scaling parameters used here for the domains of influence of analysis and projections are $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ respectively. The value used here for the permissible relative error is 15%. A very coarse discretization is used at the start of analysis with only 189 DOFs, which is subsequently refined adaptively and the number of DOFs in the second and third discretizations are 852 and 3486 respectively. This problem was run on the Hamilton cluster using 1-6 processors. The maximum number of processors is restricted here by the coarse initial discretization, as METIS cannot divide it

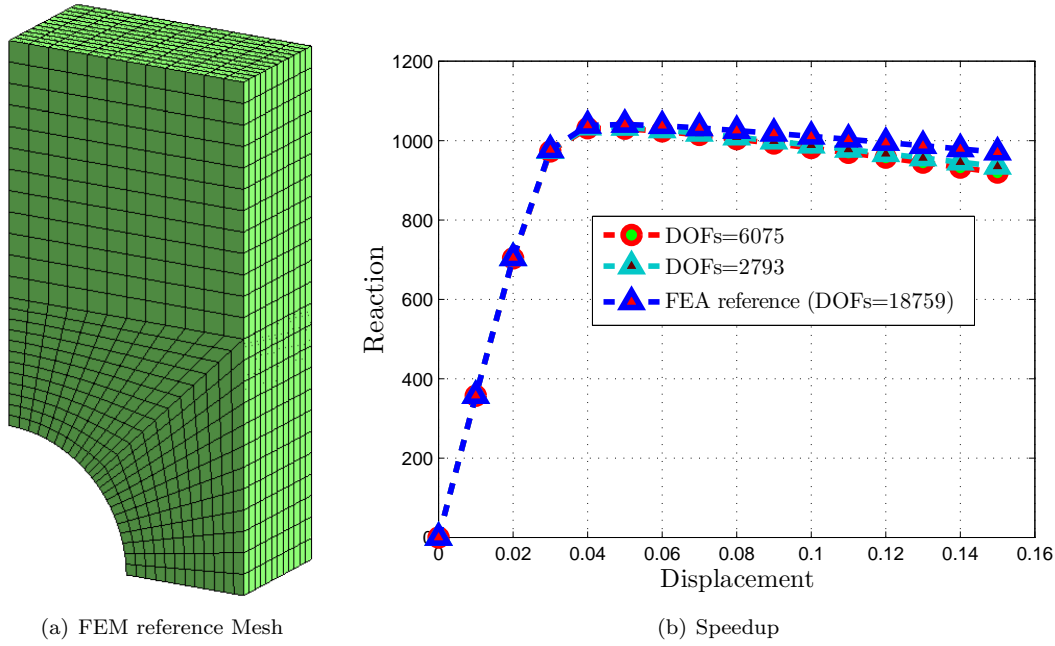


Figure 7.16: FEM reference mesh and reaction versus displacement for the 3D plate with a hole problem

into more than six subdomains. Sample METIS partitions of the adaptively refined discretizations for two processors are shown in Figures 7.18(a-c), while the final deformed configuration is shown in Figure 7.18(d). The same METIS partition and the final deformed configuration in the case of five processors are shown in Figures 7.19(a-d) respectively. The number of processors is constant during the analysis and are specified at the start of simulation and METIS repartitions the problem domain into the same number of subdomains as processors after each refinement. The final displacement contours in the x and y directions over the full plate are shown in Figures 7.20(a) and 7.20(b) respectively. As only one-eighth of this problem is analyzed, the full plate solution is given here for better presentation obtained by mirroring the one-eighth solution across the symmetry planes. Formation of the necking in the centre of the plate is obvious in both figures. The same problem is also solved with a relatively fine FEM mesh using eight-node hexahedral elements with 18759 DOFs to serve as a reference, for which the mesh is already shown in Figure 7.16(a). A comparison of the reaction versus displacement for this problem with the reference FEM is shown in Figure 7.20(c), which are in a very good agreement and shows the correct implementation of the parallel adaptive nonlinear algorithm.

Performances on the Hamilton cluster are shown in Figure 7.21. Simulation time versus the number of processors are shown in Figure 7.21(a). The timing reported here are again full simulation times. On one processor, it takes 614 seconds, while for six processors, it takes only 190 seconds. Furthermore, comparisons of speedup and efficiency with the corresponding ideal values are given in Figures 7.21(b) and 7.21(c) respectively. In the case of six processors speedup and efficiency achieved are 3.23 and 53.76% respectively.

7.9 Final demonstration problem

So far, in this thesis, all the demonstration problems have been solved with relatively small deformations using relatively coarse discretizations. The rationale is that the methods need proper testing and demonstration and small problems are best suited for this. The use of the EFGM, which is computationally very expensive as compared to the corresponding FEM has meant that thus problems involving very

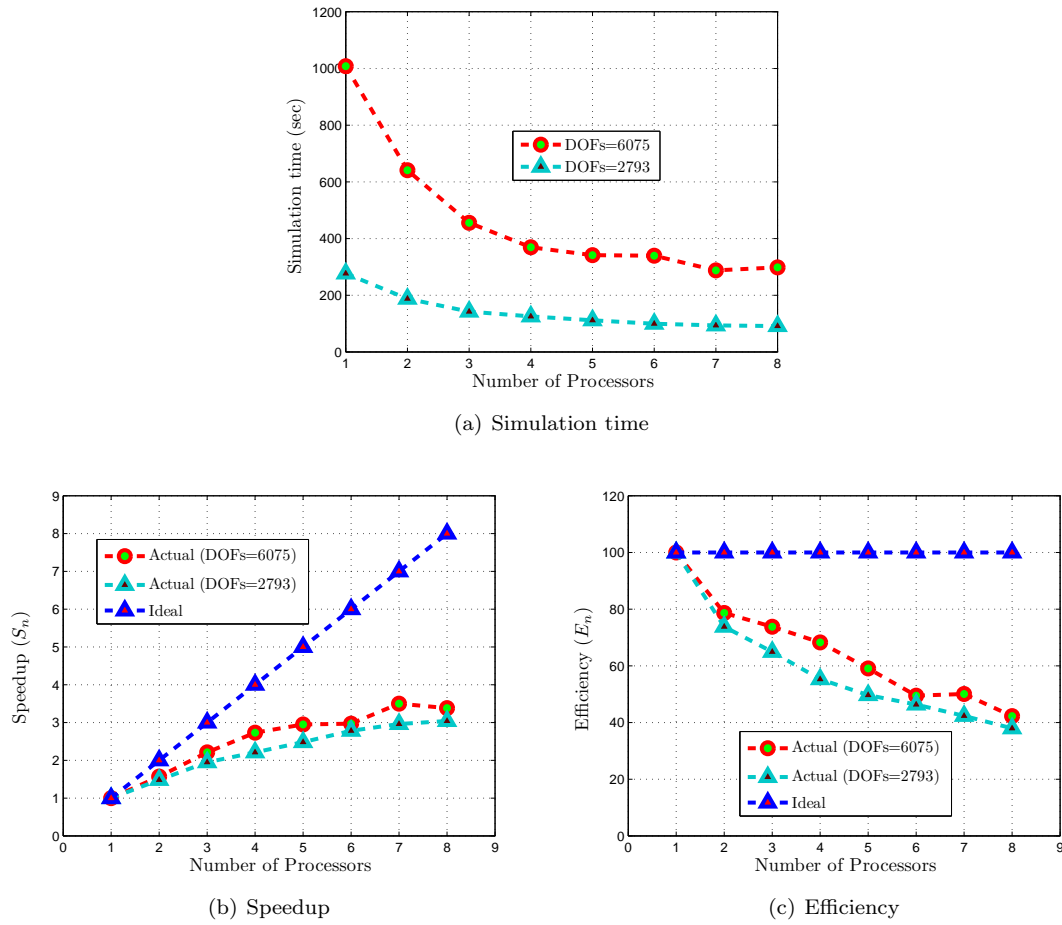


Figure 7.17: Performance on the Hamilton cluster [1] using 1-8 processors for the the 3D plate with a hole problem

large deformation problems with very fine discretizations have not been attempted. However, many new techniques have been developed so far in this thesis, including adaptive EFGM, FE-EFGM coupling, adaptive FE-EFGM coupling and parallel adaptive nonlinear EFGM. The purpose of this final demonstration problem is to show the capabilities of the methods developed so far, to handle problems with very fine discretizations subjected to very large deformation and elasto-plasticity. The same three-dimensional plate with a hole problem subjected to unidirectional tension as used in the previous section is selected again as a test problem.

The parallel code developed in the previous section is restricted to adaptive nonlinear EFGM, which is extended here to also include adaptive FE-EFGM coupling. The parallel nonlinear adaptively coupled FE-EFGM code developed here is based on the parallel nonlinear adaptive EFGM algorithm described in §7.8 and the sequential nonlinear adaptively coupled FE-EFGM algorithm described in §6.6. In this new algorithm, a coupled FE-EFGM discretization consisting of both FEs and EFG background cells is partitioned with METIS, where each METIS partition can consist of both FEs and EFG background cells. Kd-tree with background mesh algorithm is then used to find nodes in the support of those Gauss points, which belong to the EFG region of the problem domain. To calculate error for the coupled FE-EFGM discretization, consisting of both the Zienkiewicz & Zhu and the Chung & Belytschko error estimations for the FE and EFG region of the problem domain respectively, nodal incremental strains and stresses are calculated in both regions. Nodal incremental strains and stresses are calculated here sequentially in

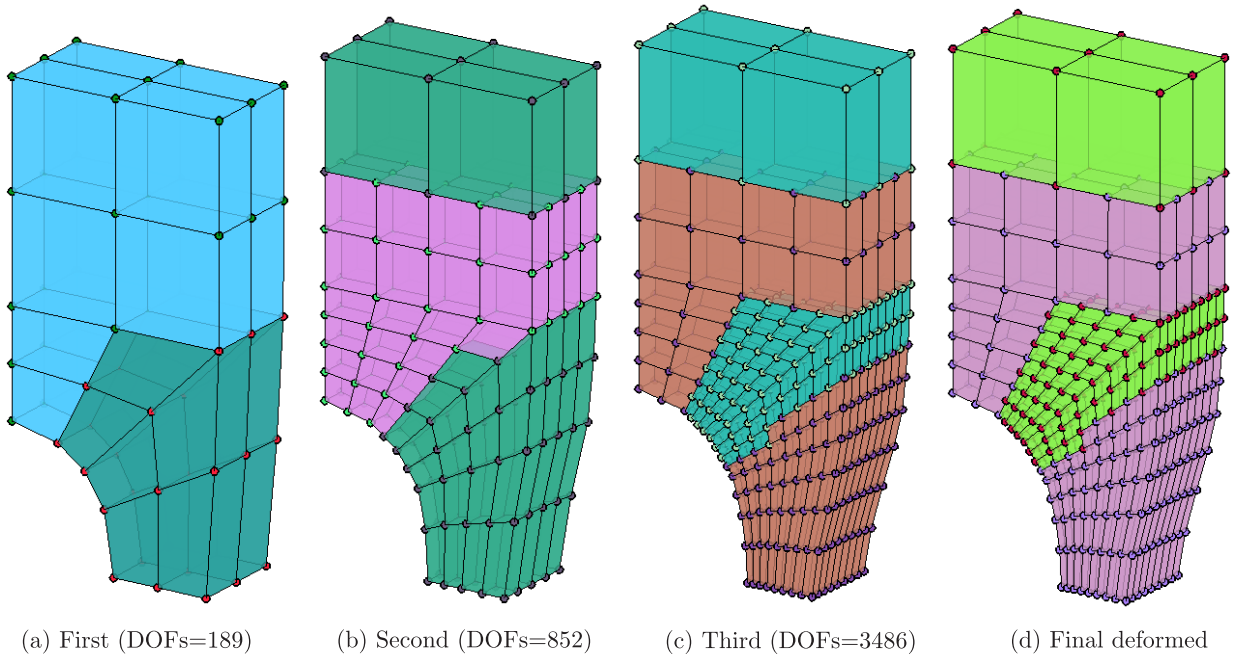


Figure 7.18: Step by step metis partitions for two processors for the adaptive 3D plate with a hole problem

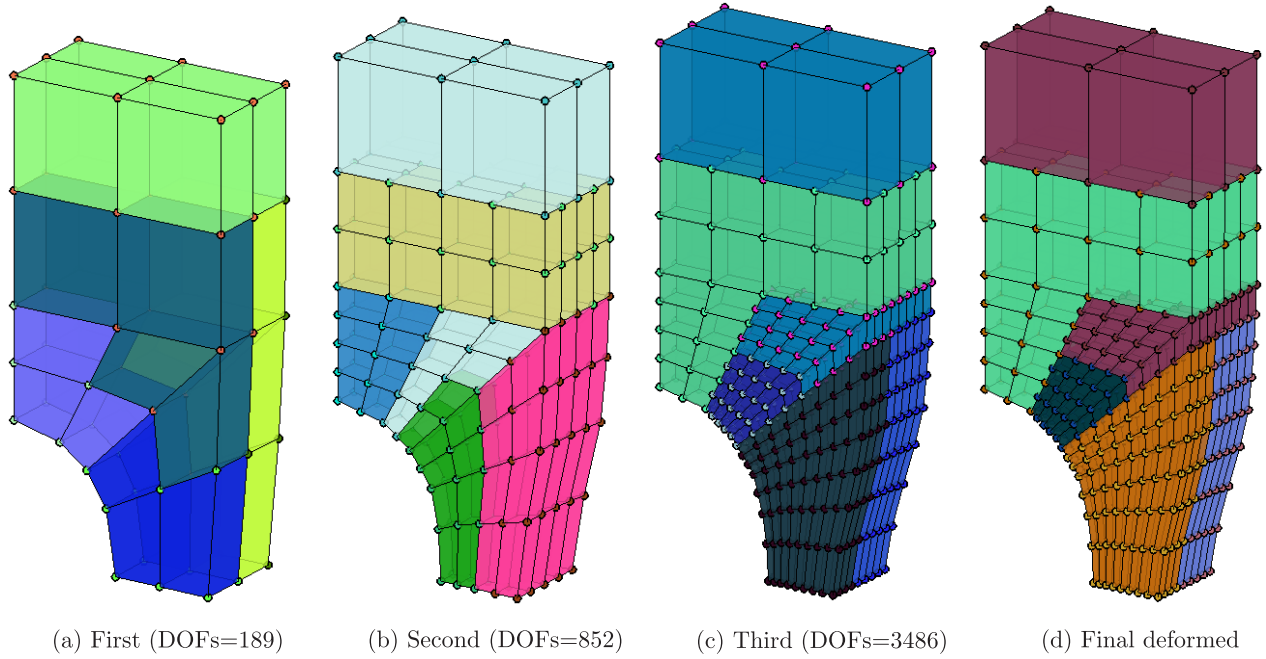


Figure 7.19: Step by step metis partitions for five processors for the adaptive 3D plate with a hole problem

contrast to the previous parallel nonlinear adaptive EFGM algorithm, where nodal incremental strains and stresses are calculated in parallel. The SPR method is used in the FE region of the problem domain to calculate the nodal incremental strains and stresses, which is difficult to parallelize. After sequential calculation of the nodal incremental strains and stresses, the error is then calculated in parallel. The rest of the algorithm is the same as described in §7.8 for the parallel nonlinear adaptive EFGM with all the relevant changes associated with the adaptively coupled FE-EFGM algorithm described in §6.6.

The problem is solved with the parallel nonlinear adaptively coupled FE-EFGM code. Again only

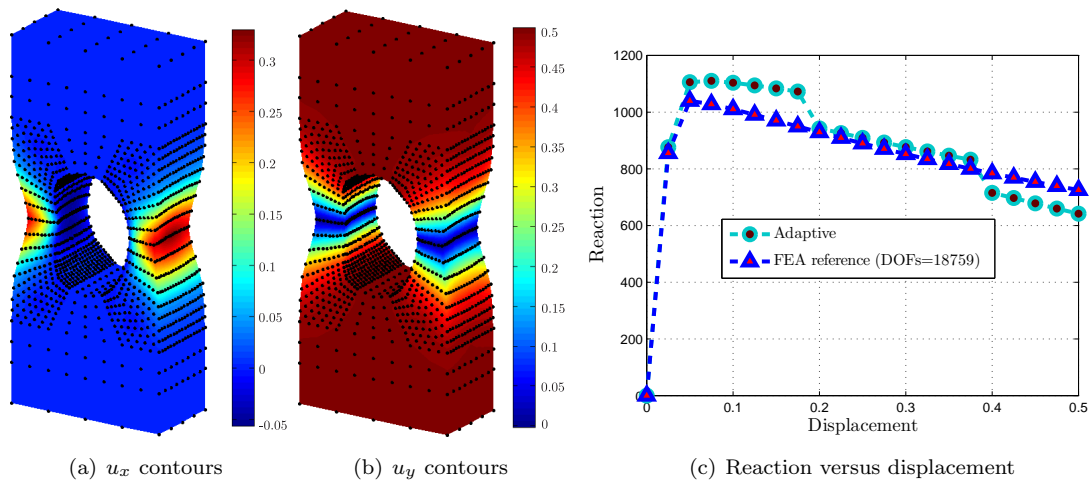


Figure 7.20: Final displacements and reaction versus displacement for the adaptive 3D plate with a hole problem

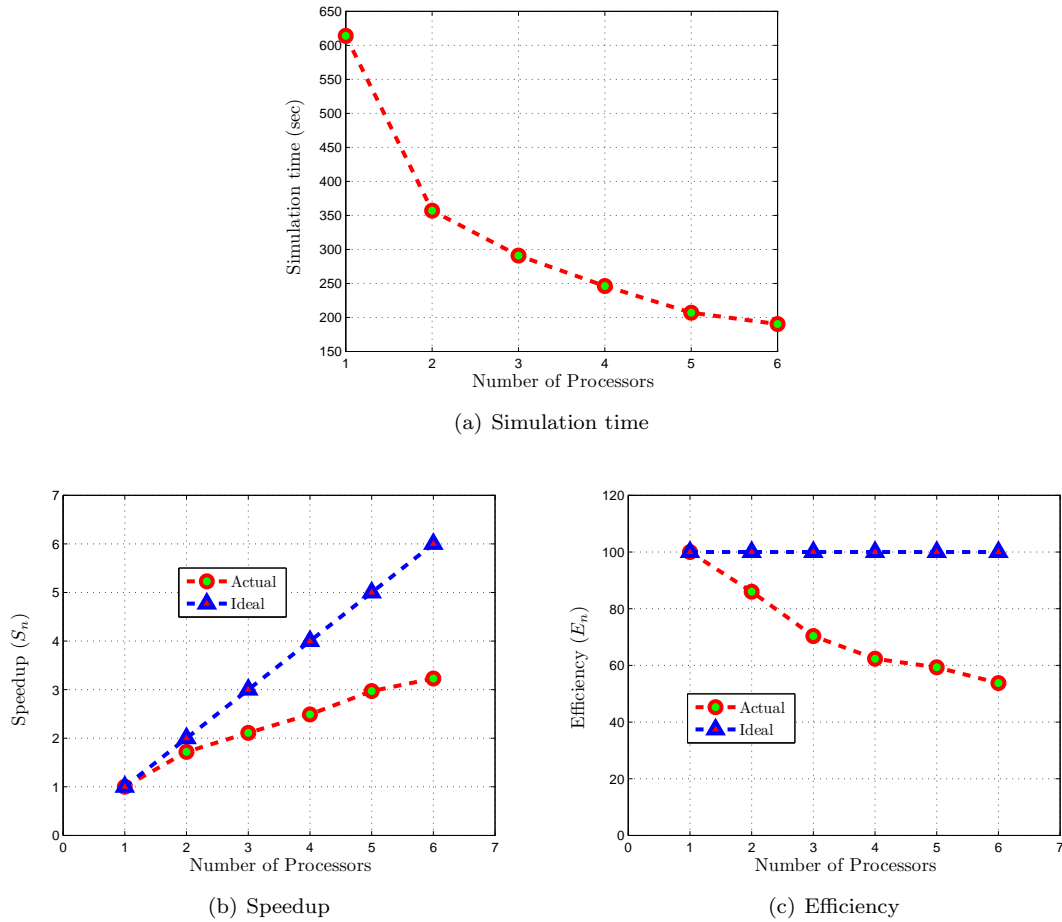


Figure 7.21: Performances on the Hamilton cluster [1] using 1-6 processors for the adaptive 3D plate with a hole problem

one-eighth of the problem is solved and a total displacement of 1.5 units is applied to the top face of the plate in equal 100 load steps. Scaling parameters used for the domains of influence of analysis and projections are $d_{max}^a = 1.5$ and $d_{max}^p = 1.1$ respectively and the permissible relative error used is 15%.

The problem is first solved using a starting FE mesh with 975 DOFs as shown in Figure 7.22(a). During the analysis, those FEs which violate a specified error measure based on the Zienkiewicz & Zhu error estimation procedure are converted adaptively into an EFG zone. The second discretization, or the first adaptively coupled FE-EFG discretization, is shown in Figure 7.22(b). During this first conversion, the number of degrees of freedom remains the same. The third and fourth discretizations are shown in Figures 7.22(c) and 7.22(d) respectively, which are obtained from the adaptive refinement based on the combined Zienkiewicz & Zhu and Chung & Belytschko error estimations. The degrees of freedom in the third and fourth discretizations are 3,198 and 11,277 respectively. Contours of the u_x and u_y over the final deformed configurations are given in Figures 7.23(b) and 7.23(c) respectively. Again full plate solutions are given in these figures, which are obtained by mirroring the one-eighth solution across the symmetry planes. For comparison, the original undeformed configuration is also given in Figure 7.23(a). Reaction versus displacement plot is shown in Figure 7.29. The curve for this case is tagged as “adaptive”. This problem is solved on Hamilton cluster using 10 processors with a total simulation time of 2,654 seconds. It is clear from Figures 7.23 and 7.29, that the developed code, can efficiently handle very large deformation problems.

The same problem with the same input parameters is next solved with the same code but this time the starting FE mesh is relatively fine as compared to the starting mesh used in the previous case, shown in Figure 7.22(a). The starting FE mesh used here is shown in Figure 7.25(a), which is then adaptively refined in the same way as explained in the previous case. The second, third and fourth adaptive discretizations are shown in Figures 7.25(b-d) respectively. The number of degrees of freedom in first, second, third and fourth discretizations are 6075, 6075, 21609 and 52386 respectively. The contours of the u_x and u_y over the deformed configurations are shown in Figures 7.26(b) and 7.26(c) respectively, while the undeformed configuration is also shown in Figure 7.26(a) for comparison. In this case, necking near the centre of the hole is more prominent as compared to the previous case. The reaction versus displacement plot for this case is also shown in Figure 7.29, this time the curve is tagged as “Adaptive-1”. As expected, the geometric softening in this case is more obvious as compared to the previous case. The analysis in this case was run on the Hamilton cluster on 50 processors with 60 load steps with a total simulation of 9391 seconds. This second nonlinear adaptively coupled FE-EFGM case further demonstrates the capabilities of the code to handle problems with fine discretizations subjected to very large deformation and elasto-plasticity. For this problem, percentage of the total simulation time consumed in each portion of the code is shown in Figure 7.24. The time spend to set up the problem and in domain decomposition using METIS is negligible. Due to the total Lagrangian formulation, the time spend in the calculation of the influence domains using Voro++, neighbour search using kd-tree with background mesh and the calculation of the shape functions and corresponding shape function derivatives are relatively small, as these are calculated and saved at the start of simulation and after each refinement and then used in each Newton-Raphson iteration. Calculation of the error estimation and refinement & transfer of path dependent variable from one discretization to the next takes 1.85% and 0.98% of the total simulation time respectively. The calculation and assembly of the stiffness matrix at the start and during each Newton-Raphson iteration consumed 24.12% of the total simulation time, while the solution of the final system of linear equations with MUMPS is the most computationally expensive part of the simulation and takes 71.73% of the total simulation time.

For comparison, the same problem is also solved with the FEM, with eight-node hexahedral elements using three different meshes as shown in Figures 7.27(a-c). The first mesh is very coarse with only 975 DOFs, the second one is relatively fine with 42,483 DOFs and the third one is very fine with 154,128 DOFs. For the 975 DOFs case, the analysis is run on the Hamilton cluster using 10 processors, and the

total simulation time is only 242 seconds. The final deformed configuration for the full plate with the u_y contours are shown in Figure 7.28(a). In this case, due to volumetric locking, material on both sides of the hole move rigidly toward the centre of the hole. Although, some necking can be seen near the centre of the hole, it is less obvious as compared to the previous two adaptively coupled FE-EFGM cases. Furthermore, the reaction versus displacement plot for this case is shown in Figure 7.29 and is tagged as “FEA (DOFs=975)”. Although, some geometric softening can be seen in this case, the response is very rigid as compared to the adaptively coupled FE-EFGM cases, and is unrealistic.

Subsequently, for the second and third FE mesh, the final deformed configurations with u_y contours are shown in Figures 7.28(b) and 7.28(c) respectively. In the second FEM case, i.e. the one with 42,483 DOFs the analysis was run on the Hamilton cluster using 40 processors with 100 load steps, with a total simulation time of 14,579 seconds. In the third FEM case, i.e. the one with 154,128 DOFs the analysis was run on the Hamilton cluster on 100 processors with 50 load steps, with a total simulation time of 48,680 seconds. In Figure 7.28(b), necking is more prominent and rigid material movement toward the centre of the hole is less obvious as compared to the response of coarse FEM shown in Figure 7.28(a). For the fine FEM case shown in Figure 7.28(c), the response is even better than the response shown in Figure 7.28(b) with very prominent necking as one would expect physically and almost no rigid material movement near the centre of the hole. The reaction versus displacement curve for the second and third FEM cases are also shown in Figure 7.29, which are tagged as “FEA (DOFs=42,483)” and “FEA (DOFs=154,128)” respectively. These reaction versus displacement curves shows more geometric softening as compared to the coarse FEM discretization. Furthermore, it is also clear from Figure 7.29 that increasing the number of degrees of freedom, leads to an increase in geometric softening.

Overall summary of the results for this final demonstration problem, with five different discretizations is given in Table 7.1. Due to the use of different load steps and number of processors for each analysis, work done per load step is calculated for each analysis using

$$\text{Work done per load step} = \frac{\text{Run time} \times \text{No of processors}}{\text{Load steps}}, \quad (7.4)$$

and is also given in the same table for comparison. As compared to the fine FEA case, work done per load step associated with the two adaptive cases are very small. Although, work done per load step associated with the first two FEA cases are very small, but it is shown above that in these cases the results are not realistic. To summarize, as compared to the corresponding FEA, the parallel adaptively coupled FE-EFGM algorithm can efficiently and accurately model problems subjected to both finite deformation and elasto-plasticity.

Cases	DoFs				Load steps	Number of processors used	Run time (sec)	Work per load step
Adaptive	975	975	3,198	11,277	100	10	2,654	265.4
Adaptive-1	6,075	6,075	21,609	52,386	60	50	9,391	7825.8
FEA (DOFs=975)	975				100	10	242	24.2
FEA (DOFs=42,483)	42,483				100	40	14,579	5831.6
FEA (DOFs=154,128)	154,128				50	100	48,680	97360.0

Table 7.1: Final demonstration problem results summary

7.10 Concluding remarks

Parallel algorithms are developed in this chapter for linear and adaptive nonlinear EFGM for distributed memory computer architecture. The parallel adaptive nonlinear EFGM algorithm is then extended to the

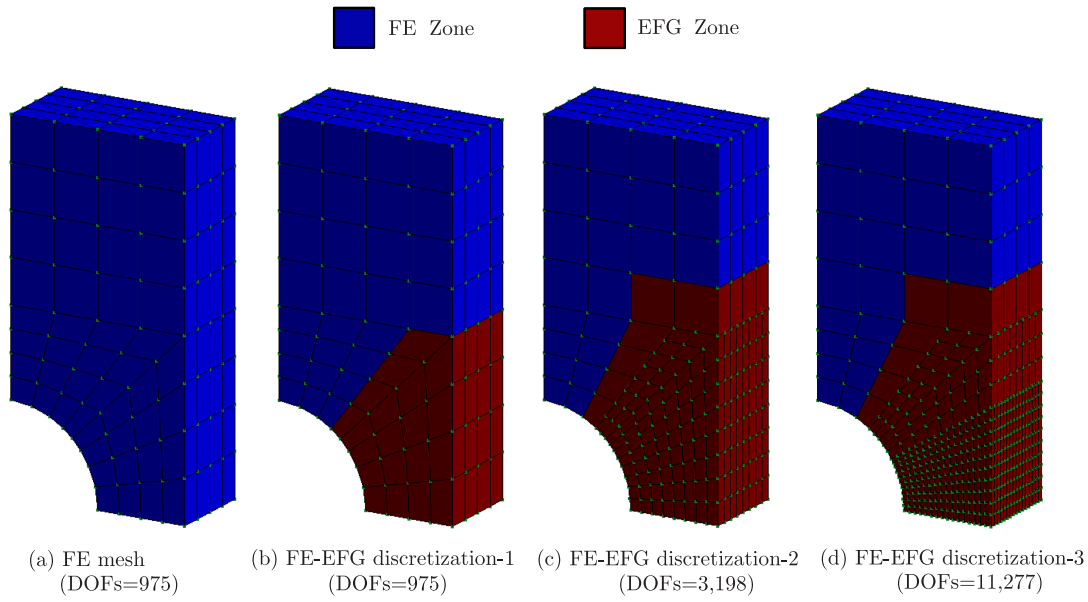


Figure 7.22: Step by step discretizations for the “adaptive” case for the final demonstration problem

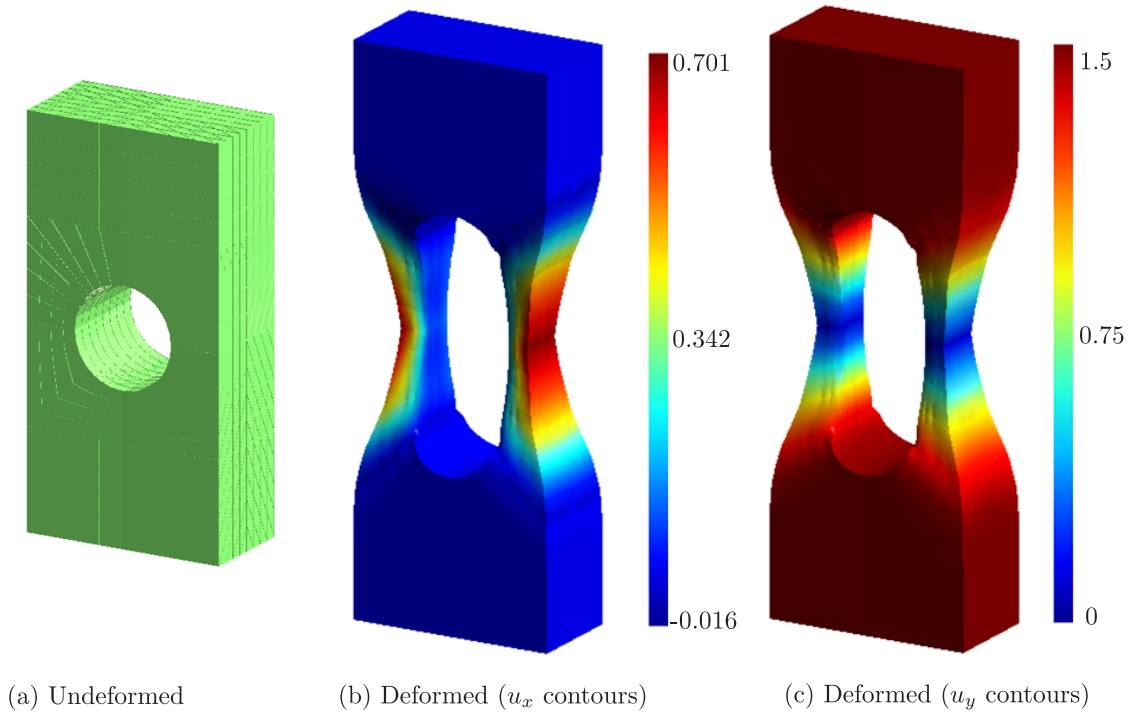


Figure 7.23: Contours of u_x and u_y over the final deformed geometry for the “adaptive” case for the final demonstration problem

parallel nonlinear adaptively coupled FE-EFGM case. In nonlinear cases, total Lagrangian formulation is used to model finite deformation and the Prandtl-Reuss constitutive model is used to model elastoplasticity. Parallel codes are developed based on these algorithms in FORTRAN with other supporting libraries, i.e. NAG, Voro++ and kd-tree. The MPI library is used for the inter-processor communication and an open-source software package, METIS is used for the automatic domain decomposition of the problem domain. An open-source parallel solver, MUMPS is used for the solution of the final system of

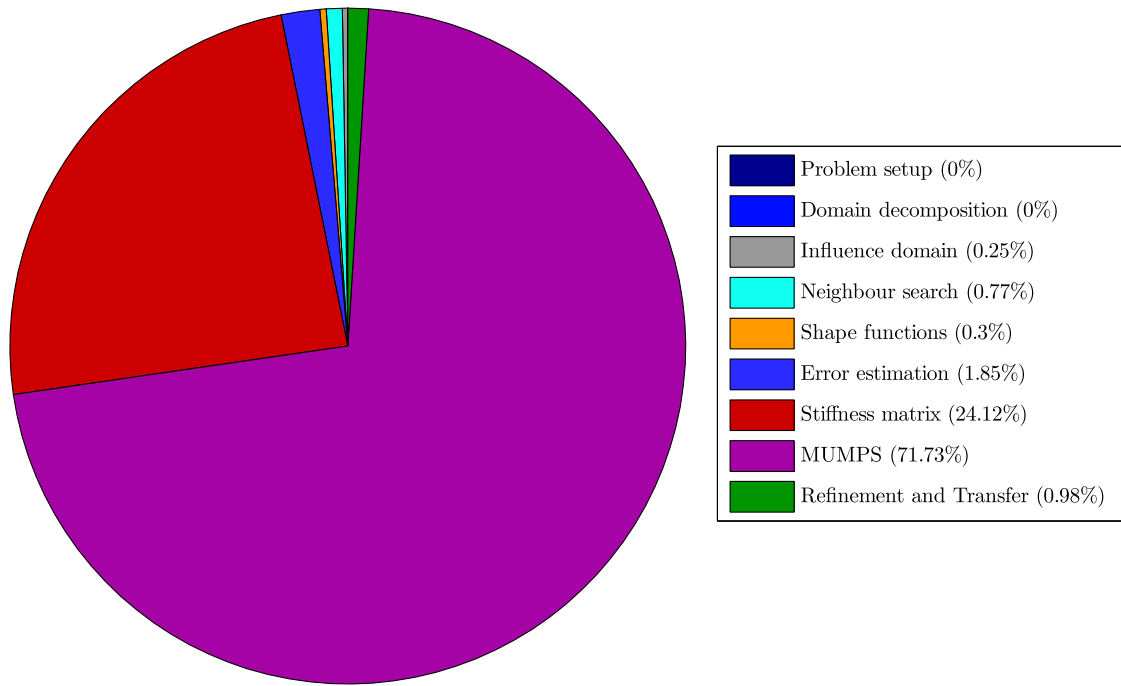


Figure 7.24: Time distribution in different part of the code

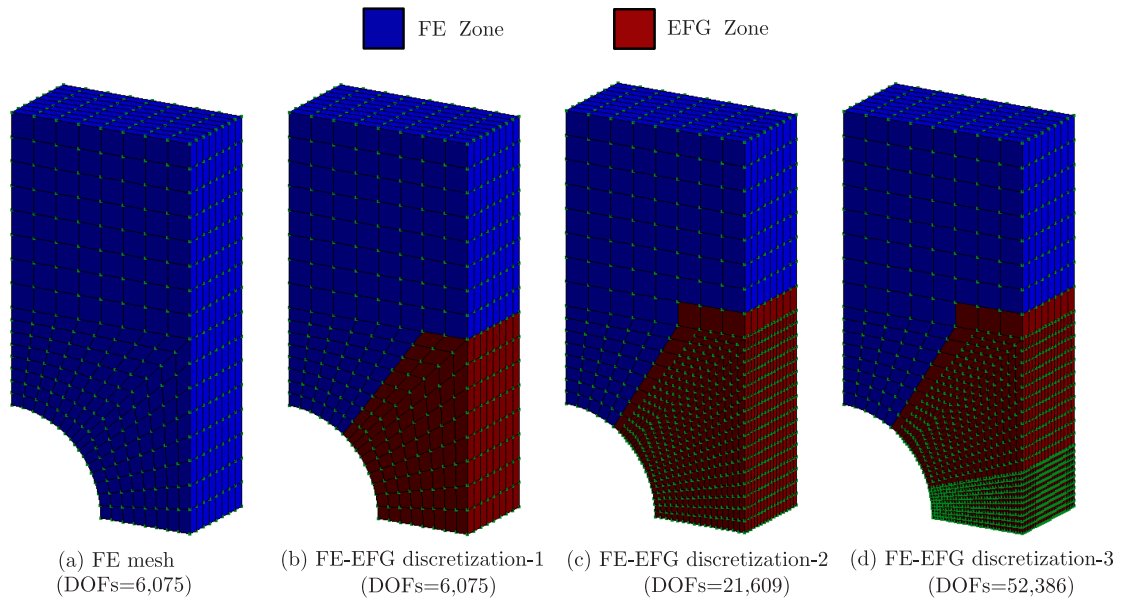


Figure 7.25: Step by step discretizations for the “adaptive-1” case for the final demonstration problem

linear equations. MUMPS supports a variety of ways to enter the input matrix, distributed assembly matrix input format is used in this research, in which there is no need to assemble the full stiffness matrix. Parallel codes are run successfully on the Durham University’s high-performance cluster Hamilton with very good speedup and efficiencies in all the cases. Two two-dimensional numerical examples, i.e. a cantilever beam problem and an infinite plate with a hole problem are solved to show the implementation and performance of the parallel linear-elastic algorithm. In this case, the displacement results are also compared with the corresponding analytical results, which are in a very good agreement. Furthermore, three-dimensional plate with a hole problem is solved with two different strategies, i.e. without and

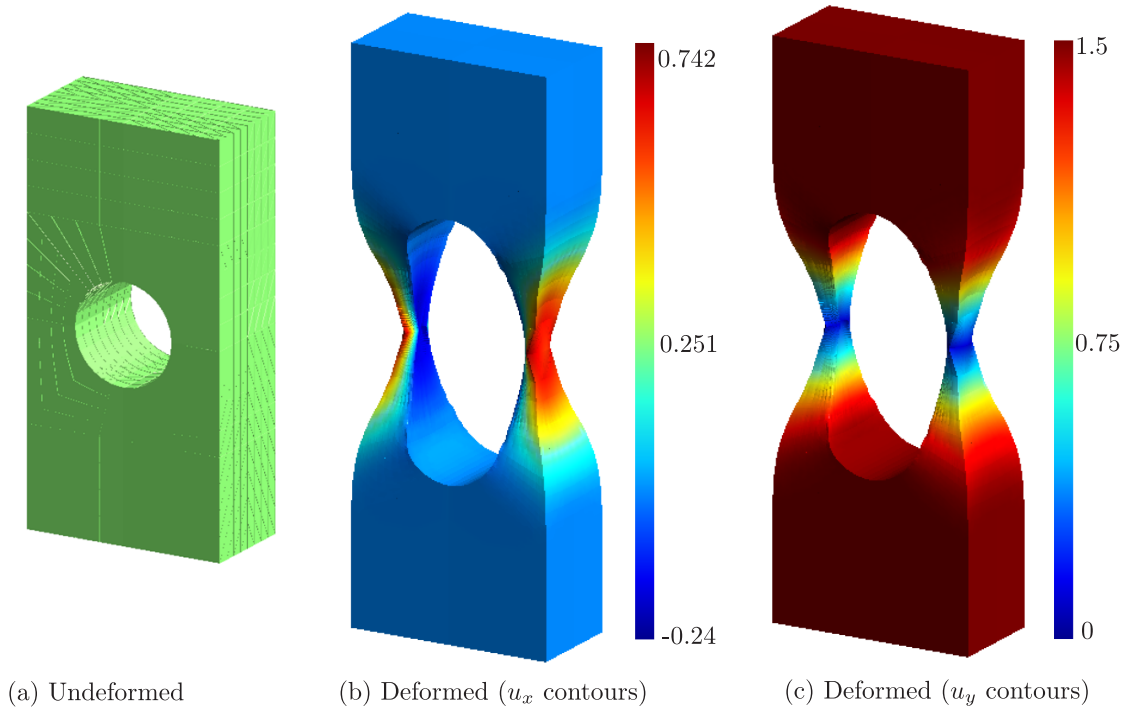


Figure 7.26: Contours of u_x and u_y over the final deformed geometry for the “adaptive-1” case for the final demonstration problem

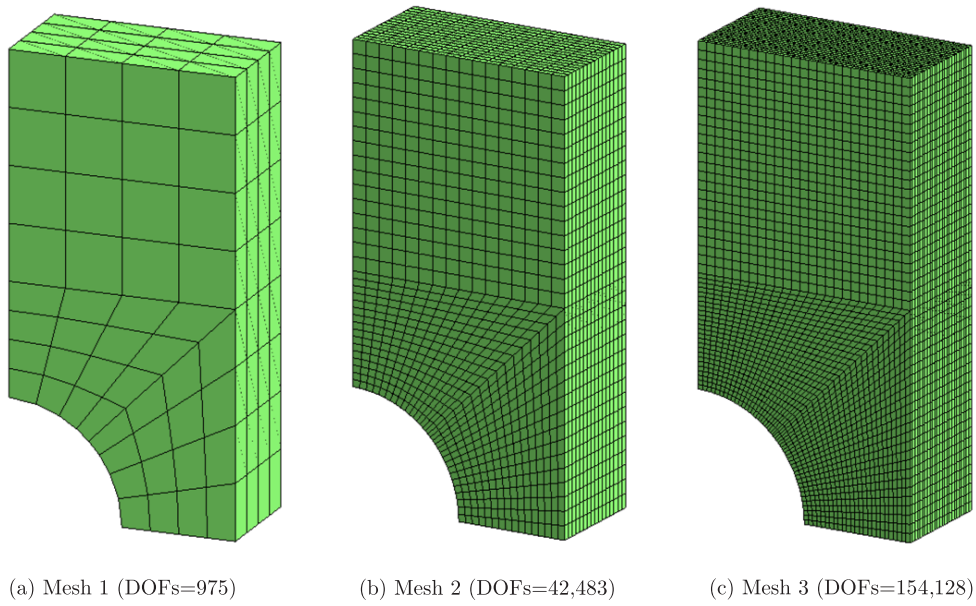


Figure 7.27: Reference FEM meshes for the final demonstration problem

with adaptivity to show the implementation and performance of the parallel adaptive nonlinear EFGM code. In this case, reaction versus displacement plots are compared with the corresponding reference FEM reaction versus displacement plots, which are obtained using a relatively fine FE mesh, and both are in a very good agreement. The final demonstration problem is also given to show the capabilities of the code to handle problems with very fine discretizations subjected to very large deformation and elasto-plasticity. Results in this case are compared with the corresponding FEM results, and it is shown

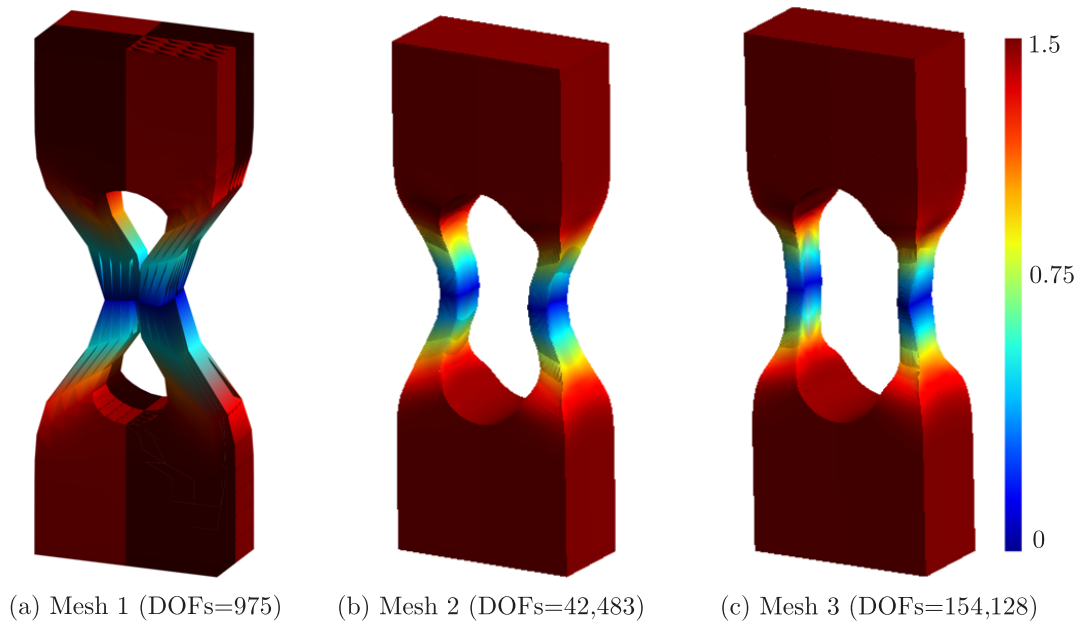


Figure 7.28: Contours of u_y over the final deformed geometry for the reference FEM meshes for the final demonstration problem

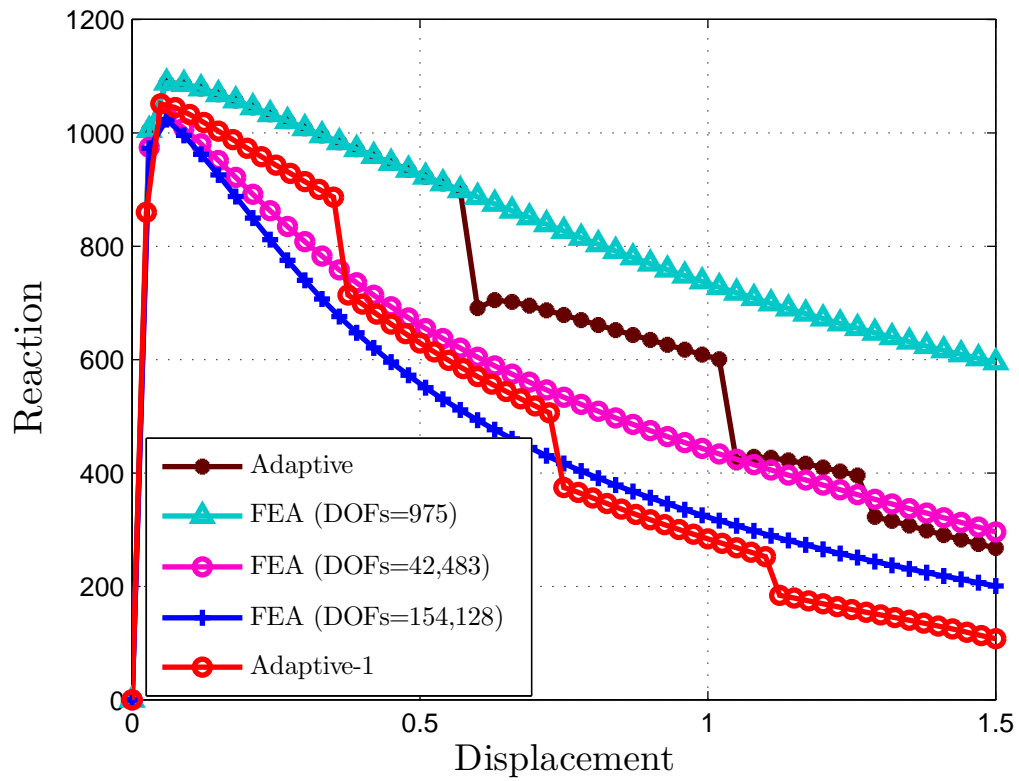


Figure 7.29: Reaction versus displacement for the final demonstration problem

that the parallel nonlinear adaptively coupled FE-EFGM algorithm can simulate these problems very efficiently with excellent results.

Chapter 8

Conclusions and recommendations for future work

8.1 Overall assessment

Modelling of problems with both material and geometrical nonlinearities remains challenging due to their complicated nature and high computational cost. Different strategies have been used throughout the literature for the solution of these problems but the challenge remains the same, i.e. to find numerical methods to solve these problems accurately and efficiently. Meshless methods are ideal in some ways to solve these problems, but their high computational cost restricts their use to small academic problems. In this thesis, a variety of new numerical strategies have been proposed for meshless methods, where the goal is to increase their computational efficiency.

Chapter 1 provides a general introduction and motivation to this thesis. All the background associated with numerical implementations of the EFGM for linear-elastic problems are covered in Chapter 2 using both MLS and max-ent shape functions. In this thesis, linear basis functions and cubic spline weight functions are used in the MLS shape functions. In the conventional EFGM, MLS shape functions are used for the approximation of the field variables, background cells are used for numerical integration and Lagrange multipliers are used for the imposition of the essential boundary conditions as MLS shape functions do not satisfy the Kronecker delta property. The recently developed, max-ent shape functions are also used in the EFGM. As compared to MLS, these shape functions satisfy a weak Kronecker delta property at the domain boundaries, which allows the imposition of essential boundary conditions directly. One-, two- and three-dimensional linear-elastic benchmark numerical examples are presented, and it is shown that EFGM with both MLS and max-ent shape functions perform satisfactorily. Furthermore, it is also shown that EFGM is more accurate with higher rate of convergence as compared to the corresponding FEM. In Chapter 3, the linear-elastic EFGM is extended to nonlinear case with both material and geometrical nonlinearities. Complete formulations are first described for small strain elasto-plasticity, which is then extended to finite deformation case. The Prandtl-Reuss constitutive model is used throughout this thesis to model elasto-plasticity, while in this chapter, updated Lagrangian formulation is used to model finite deformation. A variety of numerical examples subjected to small strain, large strain and elasto-plasticity are presented to demonstrate the correct implementation.

The main contributions of this research work are contained in Chapters 4 to 7. In the following, important points and main conclusions of each contributing chapter are summarized one by one.

Error estimation and adaptivity in the EFGM

In Chapter 4, a new adaptive meshless approach is developed for the EFGM for nonlinear solid mechanics problems subjected to both elasto-plasticity and finite deformation. This chapter starts with two-dimensional linear elastic adaptive analysis, in which the Chung & Belytschko error estimation procedure is used, in which projected stresses are calculated from the nodal stresses based on the reduced domain of influence. Effectivity index, Cell-wise error in energy norm and relative error are used as performance parameters. A refinement strategy based on the background integration cells and calculation of the nodal domains of influence based on the Voronoi diagram are also given. In the numerical examples, relative error versus DOFs is used as a performance parameter, and it is shown that adaptive refinement performs better than the uniform refinement.

The two-dimensional linear elastic adaptive procedure is then extended to two-dimensional nonlinear problems. The Chung & Belytschko error estimation procedure, which was originally proposed for linear elastic problems is extended here to nonlinear problems. The same refinement strategy, used for the linear elastic case is used here, and MLS shape functions are used for transferring of the path dependent variables between the consecutive discretizations. In the numerical examples, pressure/reaction versus displacement and effective plastic strain are used as performance parameters. The first numerical example presented is a small strain elasto-plasticity strip footing collapse with known analytical solution of limit load, to demonstrate the robustness of the basic code, before moving on to examples with both finite deformation and elasto-plasticity. Reaction versus displacement plots are compared with analytical and reference solutions. The two-dimensional nonlinear adaptive EFGM algorithm is also extended to three-dimensional case. The two-dimensional linear and nonlinear adaptive EFGM algorithms are implemented in MATLAB, while the three-dimensional nonlinear adaptive EFGM algorithm is implemented in FORTRAN. For the two-dimensional problems, MATLAB built-in functions are used for generation of Voronoi diagrams but for the three-dimensional problems an open-source software library, Voro++ is used. The two-dimensional refinement strategy is also extended to three-dimensional case. Two numerical examples are also presented, to show the implementation and performance of the three-dimensional nonlinear adaptive algorithm.

FE-EFGM coupling

In Chapter 5, a new FE-EFGM coupling procedure based on max-ent shape functions is proposed for two-dimensional linear elastic problems. In this coupling procedure, max-ent shape functions are used in the EFG region of the problem domain, which possesses a weak Kronecker delta property at the domain boundaries and provides a natural way to couple the FEM and the EFGM. As compared to the conventional FE-EFGM coupling, in which MLS shape functions are used in the EFG region of the problem domain, there is no need to use an interface or transition region between FE and EFG regions. This chapter starts with the conventional FE-EFGM coupling and full formulations of the interface elements' shape functions are given, which are based on blending functions. The FE-EFGM coupling procedure based on the max-ent shape functions is then described. It is demonstrated through one-, two-, and three-dimensional linear elastic benchmark numerical examples that this new FE-EFGM coupling procedure performs satisfactorily as compared with other alternatives, even at very large nodal influence domains. The linear elastic FE-EFGM coupling procedure is also extended to the geometrically nonlinear problems. As compared to the linear elastic case, in which FE and EFG regions in the problem domain are specified based on the problem dimensions, in the nonlinear case, these regions are identified by a tag associated with each integration point. In this case, an updated Lagrangian formulation is used to model the finite deformation. Finally, two-dimensional geometrically nonlinear benchmark numerical examples

are presented to demonstrate the implementation and performance of geometrically nonlinear FE-EFGM coupling procedure.

Adaptive FE-EFGM coupling

In Chapter 6, the FE-EFGM coupling procedure based on the max-ent shape functions proposed in Chapter 5, is extended to become **adaptive** FE-EFGM coupling. In this new coupling procedure, initially the whole of the problem domain is modelled using the FEM. During an analysis those finite elements which violate a predefined error measure are automatically converted to an EFG zone. This EFG zone can be further refined by adding nodes, thus avoiding computationally expensive FE remeshing. The Zienkiewicz & Zhu error estimation procedure with the SPR method for strains and stresses recovery are used in the FE region of the problem domain, while the Chung & Belytschko error estimation procedure is used in the EFG region of the problem domain. In the SPR method, internal patches are used to recover the nodal strains and stresses at the boundary nodes. The adaptively coupled FE-EFGM procedure for linear elastic problems are implemented in MATLAB. In the linear elastic case, error in FE region is calculated only once during the first iteration and in the subsequent iterations, the error is then calculated in EFG region. A new refinement strategy is also proposed for the adaptive FE-EFGM coupling. It is demonstrated with numerical examples, that the proposed two-dimensional linear elastic adaptively coupled FE-EFGM perform better than the corresponding uniform refinement.

In this chapter, the performance of the updated and total Lagrangian formulations in the case of the EFGM are also compared, and it is shown that total Lagrangian is computationally more efficient. The two-dimensional linear elastic adaptively coupled FE-EFGM is also extended to three-dimensional nonlinear case, in which total Lagrangian formulation is used for modelling finite deformation and Kd-tree with background mesh is used for the neighbouring search. As compared to the linear elastic case, here error in the FE region of the problem domain is calculated during each evolving discretization. Incremental error for both FE and EFG region of the problem domain are calculated for each solution step and is then used in the adaptive procedure. The three-dimensional nonlinear adaptively coupled FE-EFGM algorithm is implemented in FORTRAN, which is validated with two numerical examples.

Parallel computations

In Chapter 7, a parallel linear elastic computer code is developed for the EFGM in FORTRAN based on the distributed memory computer architecture, in which assembly of the stiffness matrix and the solution of the final system of linear equations are performed in parallel. The MPI library is used for inter-processor communication and open-source software packages, METIS and MUMPS are used for the automatic domain decomposition and solution of the final system of linear equations respectively. Simulation time, speedup and efficiency are used as performance parameters for the parallel program. The parallel linear elastic EFGM code is run successfully on the Durham University high performance cluster Hamilton. The code is validated with two two-dimensional numerical examples, in which it is shown that significant reduction in simulation time can be achieved by increasing the number of processors. Furthermore, it is also shown that increasing degrees of freedoms leads to better performance on the Hamilton cluster.

The parallel linear elastic EFGM algorithm is then extended to adaptive nonlinear EFGM case, which is validated with two three-dimensional nonlinear problems, one without adaptivity and the other with adaptivity. In this case, reaction versus displacement results are also compared with the corresponding reference results obtained from very fine FEM meshes, and it is shown that both are in a very good agreement. Performance on the Hamilton cluster are also given, and it is shown once again in this case that a significant reduction in the simulation time can be achieved by increasing the number of processors.

Finally, the parallel adaptive nonlinear EFGM algorithm is extended to parallel nonlinear adaptively coupled FE-EFGM case, and is used to solve the final demonstration problem. This final demonstration problem is one with very fine discretization subjected to very large deformation and elasto-plasticity. Results obtained from the final demonstration problem is also compared with results obtained from a very fine FEM mesh and it is shown that both are in a very good agreement. Furthermore, it is also shown that coarse FEM meshes cannot accurately simulate the problem due to the volumetric locking behaviour.

8.2 Recommendations for future work

The work presented in this thesis can be extended in the future in a number of ways:

- In this thesis, EFGM is used to model problems subjected to finite deformation and elasto-plasticity, in which background cells are used for numerical integration, which is not an ideal integration scheme for such problems. However the use of background integration cells in the EFGM simplifies its numerical implementation. Alternative integration schemes are also available in the literature for meshless methods, which may be more suitable for finite deformation problems and could be used in our case, e.g. floating stress-point integration [241, 242], nodal integration [33], stabilized conforming nodal integration [55], integration based on partition of unity [89], etc. On the other hand, fully meshless methods can be used to avoid the use of background integration cells, e.g. the MLPG method [18, 19], meshless local Kriging method [122], etc.
- In this thesis, a simple constitutive model, i.e. Prandtl-Reuss, is used to model elasto-plasticity, which comprises the von Mises yield function with perfect plasticity and associated flow. This could clearly be extended in the future, to include the effect of hardening/softening in the plasticity model. This work could also be extended to use pressure dependent constitutive models, e.g. Mohr-Coulomb and Drucker-Prager. Furthermore, the ongoing work elsewhere in the Mechanics Group, Durham University (e.g. Will Coombs' anisotropic plasticity models [67–70, 72]) can be studied and be incorporated in this work.
- Adaptive analysis in the case of nonlinear problems is a very challenging topic. Although, in the literature, varieties of error estimators with a solid mathematical background are available for the FEM in the case of problems with both material and geometrical nonlinearities but for meshless methods, this topic is relatively unexplored. In this thesis, recovery type error estimation procedures, i.e. Zienkiewicz & Zhu and Chung & Belytschko, are used in the FEM and the EFGM respectively. This work could be extended in the future, to study and incorporate other types of error estimation procedures, which are commonly used in the FEM, e.g. residual based error estimation procedures [5, 24–27, 155], model based error estimation procedures [60, 179, 237, 238] and error estimation based on the constitutive relation [171–177]. Recently developed, goal-oriented error estimation procedures [112, 253, 262, 286] could also be studied and used in this research work.
- Calculation of the proper nodal domains of influence is very important for the accuracy of meshless methods and is very challenging for three-dimensional problems with unstructured nodes. Although, in this research work, an open-source software library Voro++ is used, it should be noted that Voro++ is limited to simple geometries and cannot handle problems with arbitrary boundaries. Further investigations are required to devise a robust strategy for automatic calculation of the nodal domains of influence for three-dimensional problems with unstructured nodes.
- In this thesis, background integration cells are uniquely divided among the processors in the parallel

computations, and nodes/particle are then duplicated accordingly. This strategy will lead to equal numbers of Gauss points on each processor, which is an ideal partitioning strategy in the case of the FEM but in meshless methods, it produces load imbalance. As compared to the FEM, in meshless methods, the number of nodes belonging to the support of each Gauss point varies, which leads to a different computational load associated with each Gauss point. The research work of this thesis could be extended in the future to uniquely divide Gauss points among the processor with weight assigned to each Gauss point represent its computational load [78, 106].

- Graphic processing units (GPUs), which were designed initially for the computer graphics, are now commonly used for a modern version of parallel computing due to their tremendous computational power and cost as compared to standard high-performance computer clusters. GPUs have already been used in a variety of scientific fields, e.g. in chemistry [303], general purpose molecular dynamics [8] and flow simulation [90]. The branch of parallel computing, which uses the GPUs for general-purpose computation is known as GPGPU. Compute Unified Device Architecture (CUDA) is a language used to program the GPUs for the general-purpose computing. GPUs are now commonly used to accelerate the FE simulation and the number of references can be found in the literature, e.g. [50, 145, 164, 291] but in meshless methods, the uses of GPUs are relatively unexplored. The research work of this thesis can be further extended by using GPUs for parallel computation.

References

- [1] The Hamilton cluster, Durham University. <https://www.dur.ac.uk/cis/local/hpc/hamilton/>. (Accessed date: 31 Dec 2012).
- [2] *MUltifrontal Massively Parallel Solver - (MUMPS 4.10.0) Users' Guide*, May 2011. Available at: <http://graal.ens-lyon.fr/MUMPS/> (Accessed date: 26 Jan 2013).
- [3] k-d tree. Wikipedia, the free encyclopedia (Available at: http://en.wikipedia.org/wiki/K-d_tree), 11 2012. (Accessed date: 12 Dec 2012).
- [4] Finite element method. Wikipedia, the free encyclopedia (Available at: http://en.wikipedia.org/wiki/Finite_element_method), 2 2013. (Accessed date: 16 Feb 2013).
- [5] M. Ainsworth and J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. John Wiley & Sons, NewYork, 2000.
- [6] I. Alfaro, J. Yvonnet, E. Cueto, F. Chinesta, and M. Doblare. Meshless methods with application to metal forming. *Computer Methods in Applied Mechanics and Engineering*, 195:6661 – 6675, 2006.
- [7] M. K. Alves and R. Rossi. A modified element-free Galerkin method with essential boundary conditions enforced by an extended partition of unity finite element weight function. *International Journal for Numerical Methods in Engineering*, 57(11):1523–1552, 2003.
- [8] J. A. Anderson, C. D. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics*, 227:5342 – 5359, 2008.
- [9] A. B. Andreev and R. D. Lazarov. Superconvergence of the gradient for quadratic triangular finite elements. *Numerical Methods for Partial Differential Equations*, 4(1):15–32, 1988.
- [10] D. Applebaum. *Probability and Information, An Integrated Approach*. Cambridge University Press, 2008.
- [11] J. H. Argyris. Energy theorems and structural analysis: A generalized discourse with applications on energy principles of structural analysis including the effects of temperature and non-linear stress-strain relations. *Aircraft Engineering and Aerospace Technology*, 26 (10):347 – 356, 1954.
- [12] M. Arroyo and M. Ortiz. Local *maximum-entropy* approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering*, 65:2167–2202, 2006.
- [13] A.K. Ashmawy, B. Sukumaran, and A.V. Hoang. Evaluating the influence of particle shape on liquefaction behavior using discrete element method. In *Proceedings of the thirteenth international offshore and polar engineering conference (ISOPE 2003), Honolulu, Hawaii*, 2003.

-
- [14] H. Askes, R. de Borst, and O. Heeres. Conditions for locking-free elasto-plastic analyses in the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 173(1-2):99–109, 1999.
 - [15] H. Askes and T. Pannachet. Negative penalty functions in the element-free Galerkin method. *Communications in Numerical Methods in Engineering*, 21(2):97–98, 2005.
 - [16] S. N. Atluri. *The Meshless Method (MLPG) For Domain & BIE Discretizations*. Tech Science Press, USA.
 - [17] S. N. Atluri and S. Shen. *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, USA, 2002.
 - [18] S. N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22:117–127, 1998.
 - [19] S. N. Atluri and T. L. Zhu. The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics. *Computational Mechanics*, 25:169–179, 2000.
 - [20] C. E. Augarde and A. J. Deeks. The use of Timoshenko’s exact solution for a cantilever beam in adaptive analysis. *Finite Elements in Analysis and Design*, 44:595 – 601, 2008.
 - [21] O. Axelsson. *Finite Difference Methods, Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2004.
 - [22] I. Babuška and A. Miller . The post-processing approach in the finite element method — Part I: Calculation of displacements, stresses and other higher derivatives of the displacements. *International Journal for Numerical Methods in Engineering*, 20(6):1085–1109, 1984.
 - [23] I. Babuška and J. M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40(4):727–758, 1997.
 - [24] I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, 1978.
 - [25] I. Babuška, T. Strouboulis, and C. S. Upadhyay. A model study of the quality of a posteriori error estimators for finite element solutions of linear elliptic problems, with particular reference to the behavior near the boundary. *International Journal for Numerical Methods in Engineering*, 40(14):2521–2577, 1997.
 - [26] I. Babuška, T. Strouboulis, C. S. Upadhyay, S. K. Gangaraj, and K. Copps. Validation of a posteriori error estimators by numerical approach. *International Journal for Numerical Methods in Engineering*, 37(7):1073–1123, 1994.
 - [27] I. Babuška, T. Strouboulis, and C.S. Upadhyay. A model study of the quality of a posteriori error estimators for linear elliptic problems. error estimation in the interior of patchwise uniform grids of triangles. *Computer Methods in Applied Mechanics and Engineering*, 114(34):307 – 378, 1994.
 - [28] G. S. Bala. *Discrete finite element method for continuum-discontinuum analysis*. PhD thesis, University of Alberta (Canada), 2008.
 - [29] E. Barbieri and M. Meo. A fast object-oriented Matlab implementation of the Reproducing Kernel Particle Method. *Computational Mechanics*, 49:581–602, 2012.
-

-
- [30] J. Barlow. Optimal stress locations in finite element models. *International Journal for Numerical Methods in Engineering*, 10(2):243–251, 1976.
- [31] W. Barry and S. Saigal. A three-dimensional element-free Galerkin elastic and elastoplastic formulation. *International Journal for Numerical Methods in Engineering*, 46(5):671–693, 1999.
- [32] A. T. Becene. *Parallel processing of finite strain, materially nonlinear and incompressible finite element analysis problems*. PhD thesis, University of Rochester, 2003.
- [33] S. Beissel and T. Belytschko. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139(14):49 – 74, 1996.
- [34] T. Belytschko. A survey of numerical methods and computer programs for dynamic structural analysis. *Nuclear Engineering and Design*, 37(1):23–34, 1976.
- [35] T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach. On the completeness of meshfree particle methods. *International Journal for Numerical Methods in Engineering*, 43(5):785–819, 1998.
- [36] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3 – 47, 1996.
- [37] T. Belytschko, P. Krysl, and Y. Krongauz. A three-dimensional explicit element-free Galerkin method. *International Journal for Numerical Methods in Fluids*, 24(12):1253–1270, 1997.
- [38] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.
- [39] T. Belytschko, Y.Y. Lu, L. Gu, and M. Tabbara. Element-free Galerkin methods for static and dynamic fracture. *International Journal of Solids and Structures*, 32:2547 – 2570, 1995.
- [40] T. Belytschko, D. Organ, and Y. Krongauz. A coupled finite element-element-free Galerkin method. *Computational Mechanics*, 17:186–195, 1995.
- [41] T. Belytschko and M. Tabbara. Dynamic fracture using element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 39(6):923–938, 1996.
- [42] W. Benz. Smooth particle hydrodynamics - a review. In *Proceedings of the NATO Advanced Research Workshop on The Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects*, Kluwer Academic, Boston, 1990.
- [43] N. Bićanić. *Discrete Element Methods, Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2007.
- [44] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, 47(6):1189–1214, 2000.
- [45] B. Boroomand and O.C. Zienkiewicz. Recovery procedures in error estimation and adaptivity. Part II: Adaptivity in nonlinear problems of elasto-plasticity behaviour. *Computer Methods in Applied Mechanics and Engineering*, 176(1-4):127 – 146, 1999.
-

-
- [46] R. Boussetta, T. Coupez, and L. Fourment. Adaptive remeshing based on a posteriori error estimation for forging simulation. *Computer Methods in Applied Mechanics and Engineering*, 195(48-49):6626 – 6645, 2006.
 - [47] R. Boussetta and L. Fourment. A posteriori error estimation and three-dimensional adaptive remeshing: Application to error control of non-steady metal forming simulations. *AIP Conference Proceedings*, 712(1):2246–2251, 2004.
 - [48] J. H. Bramble and A. H. Schatz. Higher order local accuracy by averaging in the finite element method. *Mathematics of Computation*, 31:94–111, 1977.
 - [49] W. T. Carter, T.-L. Sham, and K. H. Law. A parallel finite element method and its prototype implementation on a hypercube. *Computers & Structures*, 31(6):921 – 934, 1989.
 - [50] C. Cecka, A. J. Lew, and E. Darve. Assembly of finite element methods on graphics processors. *International Journal for Numerical Methods in Engineering*, 85:640–669, 2011.
 - [51] E. Cervera. *Evolutionary Structural Optimisation Based on Boundary Element Representation of B-spline Geometry*. PhD thesis, The University of Durham, 2003.
 - [52] J. S. Chen, C. Pan, C. M. O. L. Roque, and H. P. Wang. A lagrangian reproducing kernel particle method for metal forming analysis. *Computational Mechanics*, 22(3):289–307, 1998.
 - [53] J. S. Chen, C. Pan, C. T. Wu, and W. K. Liu. Reproducing Kernel Particle Methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):195–227, 1996.
 - [54] J. S. Chen, C. M. O. L. Roque, C. Pan, and S. T. Button. Analysis of metal forming process based on meshless method. *Journal of Materials Processing Technology*, 80-81:642–646, 1998.
 - [55] J. S. Chen, C. T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466, 2001.
 - [56] J. S. Chen, S. Yoon, H. P. Wang, and W. K. Liu. An improved reproducing kernel particle method for nearly incompressible finite elasticity. *Computer Methods in Applied Mechanics and Engineering*, 181(13):117 – 145, 2000.
 - [57] X. L. Chen, G. R. Liu, and S. P. Lim. An element free Galerkin method for the free vibration analysis of composite laminates of complicated shape. *Composite Structures*, 59(2):279 – 289, 2003.
 - [58] Y. Chen, J. D. Lee, and A. Eskandarian. *Meshless Methods in Solid Mechanics*. Springer, 2006.
 - [59] K.N. Chiang and R.E. Fulton. Concepts and implementation of parallel finite element analysis. *Computers & Structures*, 36(6):1039 – 1046, 1990.
 - [60] J.R. Cho and J.T. Oden. A priori modeling error estimates of hierarchical models for elasticity problems for plate- and shell-like structures. *Mathematical and Computer Modelling*, 23(10):117 – 133, 1996.
 - [61] C. K. Choi and N. H. Lee. A 3D adaptive mesh refinement using variable-node solid transition elements. *International Journal for Numerical Methods in Engineering*, 39(9):1585–1606, 1996.
-

-
- [62] H.-J. Chung and T. Belytschko. An error estimate in the EFG method. *Computational Mechanics*, 21:91–100, 1998.
- [63] V. Cingoski, N. Miyamoto, and H. Yamashita. Element-free Galerkin method for electromagnetic field computations. *Magnetics, IEEE Transactions on DOI - 10.1109/20.717759*, 34(5):3236–3239, 1998.
- [64] J. Clausen. *Efficient non-linear finite element implementation of elasto-plasticity for geotechnical problems*. PhD thesis, Aalborg University, Denmark, 2007.
- [65] R. W. Clough. The finite element method in plane stress analysis. In *Proceedings, American Society of Civil Engineers, 2nd Conference on Electronic Computation, Pittsburgh, PA*, pages 345–378, 1960.
- [66] R. W. Clough and Y. Rashid. Finite element analysis of axisymmetric solids. *Journal of the Engineering Mechanics Division, Proceedings of the American Society of Civil Engineers*, 91:71–85, 1965.
- [67] W. M. Coombs. *Finite deformation of particulate geomaterials: frictional and anisotropic Critical State elasto-plasticity*. PhD thesis, School of Engineering & Computing Sciences, Durham University, UK, 2011.
- [68] W. M. Coombs and R. S. Crouch. Algorithmic issues for three-invariant hyperplastic critical state models. *Computer Methods in Applied Mechanics and Engineering*, 200:2297 – 2318, 2011.
- [69] W. M. Coombs and R. S. Crouch. Non-associated reuleaux plasticity: Analytical stress integration and consistent tangent for finite deformation mechanics. *Computer Methods in Applied Mechanics and Engineering*, 200(912):1021 – 1037, 2011.
- [70] W. M. Coombs, R. S. Crouch, and C. E. Augarde. Reuleaux plasticity: Analytical backward Euler stress integration and consistent tangent. *Computer Methods in Applied Mechanics and Engineering*, 199:1733 – 1743, 2010.
- [71] W. M. Coombs, R. S. Crouch, and C. E. Augarde. 70-line 3D finite deformation elastoplastic finite-element code. In *Proc. Numerical Methods in Geotechnical Engineering (NUMGE), Trondheim, Norway*, pages 151–156, June 3-5 2010.
- [72] W.M. Coombs, R.S. Crouch, and C.E. Augarde. A unique critical state two-surface hyperplasticity model for fine-grained particulate media. *Journal of the Mechanics and Physics of Solids*, 61(1):175 – 189, 2013.
- [73] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, 3rd edition, 2009.
- [74] P. A. Cundall. A computer model for simulating progressive, large-scale movements in blocky rock systems. *Proceedings, international symposium on rock fracture, Nancy, france*, II-8, 1971.
- [75] P. A. Cundall and R. D. Hart. Numerical modeling of discontinua. *Comprehensive Rock Engineering, Ed. J. A. Hudson*, 2:231–243, 1993.
- [76] P.A. Cundall and O.D.L. Strack. Discrete numerical model for granular assemblies. *Geotechnique*, 29 (1):47–65, 1979.
-

-
- [77] C. J. Cyron, M. Arroyo, and M. Ortiz. Smooth, second order, non-negative meshfree approximants selected by maximum entropy. *International Journal for Numerical Methods in Engineering*, 79(13):1605–1632, 2009.
 - [78] K. T. Danielson, S. Hao, W. K. Liu, R. A. Uras, and S. Li. Parallel computation of meshless methods for explicit dynamic analysis. *International Journal for Numerical Methods in Engineering*, 47(7):1323–1341, 2000.
 - [79] N. Das. *Modeling Three-Dimensional Shape of Sand Grains Using Discrete Element Method*. PhD thesis, University of South Florida, 2007.
 - [80] R. de Borst. *Solids and Structures: Introduction and Survey, Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2004.
 - [81] E. A. de Souza Neto, D. Perić, and D. R. J. Owen. *Computational Methods for Plasticity: Theory and Applications*. John Wiley & Sons, Ltd, 2008.
 - [82] A. J. Deeks and C. E. Augarde. A hybrid meshless local Petro-Galerkin method for unbounded domains. *Computer Methods in Applied Mechanics and Engineering*, 196(46):843 – 852, 2007.
 - [83] G. R. Liu Y. L. Wu H. Ding. Meshfree weak-strong (MWS) form method and its application to incompressible flow problems. *International Journal for Numerical methods fluids*, 46 (10):1025–1047, 2004.
 - [84] M. Doblaré, E. Cueto, B. Calvo, M.A. Martínez, J.M. Garcia, and J. Cego nino. On the employ of meshless methods in biomechanics. *Computer Methods in Applied Mechanics and Engineering*, 194(68):801 – 821, 2005.
 - [85] J. Dolbow and T. Belytschko. An introduction to programming the meshless element-free Galerkin method. *Archives of Computational Methods in Engineering*, 5:207–241, 1998.
 - [86] J. Dolbow and T. Belytschko. Volumetric locking in the element free Galerkin method. *International Journal for Numerical Methods in Engineering*, 46(6):925–942, 1999.
 - [87] M.H. Doweidar, B. Calvo, I. Alfaro, P. Groenenboom, and M. Doblaré. A comparison of implicit and explicit natural element methods in large strains problems: Application to soft biological tissues modeling. *Computer Methods in Applied Mechanics and Engineering*, 199(2528):1691 – 1700, 2010.
 - [88] C. Armando Duarte and J. Tinsley Oden. An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):237 – 262, 1996.
 - [89] M. Dufloot and H. Nguyen-Dang. A truly meshless Galerkin method based on a moving least squares quadrature. *Communications in Numerical Methods in Engineering*, 18(6):441–449, 2002.
 - [90] E. Elsen, P. LeGresley, and E. Darve. Large calculation of the flow over a hypersonic vehicle using a GPU. *Journal of Computational Physics*, 227(24):10148 – 10161, 2008.
 - [91] O. V. Estorff and J. Quan. Direct coupling of EFGM-FEM and EFGM-BEM for dynamic soilstructure interactions. *International Journal of Computational Methods*, 02(04):627–644, 2005.
 - [92] C. Farhat and F. X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 1991.
-

-
- [93] Y. T. Feng and D. Perić. Coarse mesh evolution strategies in the Galerkin multigrid method with adaptive remeshing for geometrically non-linear problems. *International Journal for Numerical Methods in Engineering*, 49(4):547–571, 2000.
- [94] S. Fernández-Méndez and A. Huerta. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48:1615–1636, 2000.
- [95] S. Fernández-Méndez and A. Huerta. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193:1257–1275, 2004.
- [96] L. Filice, I. Alfaro, F. Gagliardi, E. Cueto, F. Micari, and F. Chinesta. A preliminary comparison between finite element and meshless simulations of extrusion. *Journal of Materials Processing Technology*, 209(6):3039 – 3049, 2009.
- [97] M.J. Flynn. Very high-speed computing systems. *Proceedings of the IEEE*, 54(12):1901 – 1909, 1966.
- [98] T.M. Foster, M.S. Mohamed, J. Trevelyan, and G. Coates. Rapid re-meshing and re-solution of three-dimensional boundary element problems for interactive stress analysis. *Engineering Analysis with Boundary Elements*, 36 (9):1331–1343, 2012.
- [99] T. P. Fries and H. G. Matthies. Classification and overview of meshfree methods. *Tech. Rep. Informatikbericht-Nr. 2003-03, Institute of Scientific Computing, Technical University Braunschweig*, 2004.
- [100] M. B. Liu G. R. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Publishing Co. Pte. Ltd. Singapore, 2003.
- [101] R. H. Gallagher, J. Padlog, and P. P. Bijlaard. Stress analysis of heated complex shapes. *Journal of the American Rocket Society*, 32:700–707, 1962.
- [102] R. R. Ganagher and J. Padlog. Discrete element approach to structural stability analysis. *Journol of the American Institute of Aeronautics and Astronautics*, 1 (6):1437–1439, 1963.
- [103] L. Gavete, J. L. Cuesta, and A. Ruiz. A numerical comparison of two different approximations of the error in a meshless method. *European Journal of Mechanics - A/Solids*, 21(6):1037 – 1054, 2002.
- [104] L. Gavete, J. L. Cuesta, and A. Ruiz. A procedure for approximation of the error in the EFG method. *International Journal for Numerical Methods in Engineering*, 53(3):677–690, 2002.
- [105] L. Gavete, S. Falcu, and A. Ruiz. An error indicator for the element free Galerkin method. *European Journal of Mechanics - A/Solids*, 20(2):327 – 341, 2001.
- [106] F. Günther, W. K. Liu, D. Diachin, and M. A. Christon. Multi-scale meshfree parallel computations for viscous, compressible flows. *Computer Methods in Applied Mechanics and Engineering*, 190:279 – 303, 2000.
- [107] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine: A Users’ Guide and Tutorial for Network Parallel Computing*. MIT Press, Cambridge, 1996.
-

-
- [108] S.A. Gharehbaghi and A.R. Khoei. Three-dimensional superconvergent patch recovery method and its application to data transferring in small-strain plasticity. *Computational Mechanics*, 41:293–312, 2008.
- [109] R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375389, 1977.
- [110] D. González, E. Cueto, and M. Doblaré. A higher order method based on local maximum entropy approximation. *International Journal for Numerical Methods in Engineering*, 83(6):741–764, 2010.
- [111] P. E. Grafton and D. R. Strome. Analysis of axisymmetric shells by the direct stiffness method. *Journal of the American Institute of Aeronautics and Astronautics*, 1 (10):2342–2347, 1963.
- [112] T. Grätsch and K. J. Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & Structures*, 83:235 – 265, 2005.
- [113] W. Gropp, S. Huss-Lederman, A. Lumsdaine, E. Lusk, B. Nitzberg, W. Saphir, and M. Snir. *MPI: The Complete Reference-Volume 2: The MPI2 Extensions*. MIT, Cambridge, MA, USA, 1998.
- [114] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, Cambridge, MA, USA, 2nd edition, 1999.
- [115] W. Gropp, E. Lusk, and R. Thakur. *Using MPI-2: Advanced Features of the Message-Passing Interface*. MIT Press, Cambridge, MA, USA, 1999.
- [116] A. Del Grosso and G. Righetti. Finite element techniques and artificial intelligence on parallel machines. *Computers & Structures*, 30(4):999 – 1007, 1988.
- [117] H. Gu, Z. Zong, and K.C. Hung. A modified superconvergent patch recovery method and its application to large deformation problems. *Finite Elements in Analysis and Design*, 40:665 – 687, 2004.
- [118] Y. T. Gu and G. R. Liu. A boundary point interpolation method for stress analysis of solids. *Computational Mechanics*, 28(1):47–54, 2002.
- [119] Y. T. Gu and L. C. Zhang. Coupling of the meshfree and finite element methods for determination of the crack tip fields. *Engineering Fracture Mechanics*, 75(5):986 – 1004, 2008.
- [120] Y.T. Gu. An elasto-plastic analysis of solids by the local meshless method based on MLS. *International Journal of Modern Physics B*, 22(31/32):5780–5786, December 2008.
- [121] Y.T. Gu and G.R. Liu. A local point interpolation method for static and dynamic analysis of thin beams. *Computer Methods in Applied Mechanics and Engineering*, 190(42):5515 – 5528, 2001.
- [122] Y.T. Gu, Q.X. Wang, and K.Y. Lam. A meshless local Kriging method for large deformation analyses. *Computer Methods in Applied Mechanics and Engineering*, 196:1673 – 1684, 2007.
- [123] Y. M. Guo, K. Nakanishi, and Y. Yokouchi. A nonlinear rigid-plastic analysis for metal forming problem using the rigid-plastic point collocation method. *Advances in Engineering Software*, 36(4):234 – 242, 2005.
- [124] O. Hazama, H. Okuda, and K. Wakatsuchi. A digital systematization of meshfree method and its applications to elasto-plastic infinitesimal deformation analysis. *Advances in Engineering Software*, 32(8):647–664, 2001.
-

-
- [125] C. E. Heaney, C. E. Augarde, and A. J. Deeks. Modelling elasto-plasticity using the hybrid MLPG method. *Computer Modeling in Engineering and Sciences*, 56(2):153–178, 2010.
- [126] D. Hegen. Element-free Galerkin methods in combination with finite element approaches. *Computer Methods in Applied Mechanics and Engineering*, 135(1-2):143 – 166, 1996.
- [127] R. Hill. *The mathematical theory of plasticity*. Oxford University Press, London, 1950.
- [128] E. Hinton and J. S. Campbell. Local and global smoothing of discontinuous finite element functions using a least squares method. *International Journal for Numerical Methods in Engineering*, 8(3):461–480, 1974.
- [129] J. T. Holden. On the finite deflections of thin beams. *International Journal of Solids and Structures*, 8(8):1051 – 1055, 1972.
- [130] G. A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. John Wiley & Sons, Ltd, 2000.
- [131] A. Horton, A. Wittek, and K. Miller. Towards meshless methods for surgical simulation. *Computational Biomechanics for Medicine Workshop, Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2006, Copenhagen, Denmark*, 2006.
- [132] A. Hrennikoff. Solution of problems in elasticity by the frame work method. *Journal of Applied Mechanics*, 8 (4):169–175, 1941.
- [133] W. Hu, L. G. Yao, and Z. Z. Hua. Parallel point interpolation method for three-dimensional metal forming simulations. *Engineering Analysis with Boundary Elements*, 31(4):326 – 342, 2007.
- [134] Y. Hu and M. F. Randolph. H-adaptive FE analysis of elasto-plastic non-homogeneous soil with large deformation. *Computers and Geotechnics*, 23(1-2):61 – 83, 1998.
- [135] A. Huerta, S. Fernández-Méndez, and W. K. Liu. A comparison of two formulations to blend finite elements and mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193:1105 – 1117, 2004.
- [136] R. Huiskies and E. Y. S. Chao. A survey of finite element analysis in orthopedic biomechanics: The first decade. *Journal of Biomechanics*, 16(6):385–409, 1983.
- [137] U. Hussler-Combe and C. Korn. An adaptive approach with the element-free-Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 162(1-4):203 – 222, 1998.
- [138] D. V. Hutton. *Fundamental of finite element analysis*. McGraw-Hill, New York, 2004.
- [139] S. R. Idelsohn and E. O nate. To mesh or not to mesh. that is the question ... *Computer Methods in Applied Mechanics and Engineering*, 195:4681 – 4696, 2006.
- [140] M.A. Jaswon. Integral equation methods in potential theory - I. *Proc. Roy. Soc. Lond., A*, 275:2332, 1963.
- [141] S.A.S. Javanmard, F. Daneshmand, M.M. Moshksar, and R. Ebrahimi. Meshless analysis of backward extrusion by natural element method. *Iranian Journal of Science and Technology, Transaction B. Engineering*, 35 (M2):167180, 2011.
- [142] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.
-

-
- [143] E. T. Jaynes. Information theory and statistical mechanics-II. *Physical Review*, 108:171–190, 1957.
- [144] W. Ji, A. M. Waas, and Z. P. Bazant. Errors caused by non-work-conjugate stress and strain measures and necessary corrections in finite element programs. *Journal of Applied Mechanics*, 77(4):044504, 2010.
- [145] G. R. Joldes, A. Wittek, and K. Miller. Real-time nonlinear finite element computations on GPU - application to neurosurgical simulation. *Computer Methods in Applied Mechanics and Engineering*, 199(4952):3305 – 3314, 2010.
- [146] M. T. Jones and P. E. Plassmann. Computational results for parallel unstructured mesh computations. *Computing Systems in Engineering*, 5:297 – 309, 1994.
- [147] I. Kaljević and S. Saigal. An improved element free Galerkin formulation. *International Journal for Numerical Methods in Engineering*, 40(16):2953–2974, 1997.
- [148] E.J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics-I surface approximations and partial derivative estimates. *Computers & Mathematics with Applications*, 19:127 – 145, 1990.
- [149] M. H. Kargarnovin, H. E. Toussi, and S. J. Fariborz. Elasto-plastic element-free Galerkin method. *Computational Mechanics*, 33(3):206–214, February 2004.
- [150] H. Karutz, R. Chudoba, and W.B. Krätzig. Automatic adaptive generation of a coupled finite element/element-free Galerkin discretization. *Finite Elements in Analysis and Design*, 38(11):1075 – 1091, 2002.
- [151] G. Karypis. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices Version 5.0*. Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455, August 2011. Available at:<http://glaros.dtc.umn.edu/gkhome/metis/metis/overview> (Accessed date: 26 Jan 2013).
- [152] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.
- [153] G. Karypis and V. Kumar. Multilevel algorithms for multi-constraint graph partitioning. In *Supercomputing*, 1998.
- [154] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48(1):96–129, 1998.
- [155] D. W. Kelly, De S. R., J. P. Gago, O. C. Zienkiewicz, and I. Babuška. A posteriori error analysis and adaptive processes in the finite element method: Part I-error analysis. *International Journal for Numerical Methods in Engineering*, 19(11):1593–1619, 1983.
- [156] M. B. Kennel. KDTREE 2: Fortran 95 and C++ software to efficiently search for near neighbors in a multi-dimensional Euclidean space, 2004.
- [157] A. I. Khinchin. *Mathematical foundation of information theory*. Dover Publications, Inc. , New York, 1957.
-

-
- [158] A. R. Khoei, A. R. Tabarraie, and S. A. Gharehbaghi. H-adaptive mesh refinement for shear band localization in elasto-plasticity cosserat continuum. *Communications in Nonlinear Science and Numerical Simulation*, 10(3):253 – 286, 2005.
- [159] A.R. Khoei and A. Bakhshiani. A hypoelasto-viscoplastic endochronic model for numerical simulation of shear band localization. *Finite Elements in Analysis and Design*, 41(14):1384 – 1400, 2005.
- [160] A.R. Khoei and S.A. Gharehbaghi. The superconvergence patch recovery technique and data transfer operators in 3D plasticity problems. *Finite Elements in Analysis and Design*, 43(8):630 – 648, 2007.
- [161] A.R. Khoei and S.A. Gharehbaghi. Three-dimensional data transfer operators in large plasticity deformations using modified-SPR technique. *Applied Mathematical Modelling*, 33(7):3269 – 3285, 2009.
- [162] A.R. Khoei, S.A. Gharehbaghi, A.R. Tabarraie, and A. Riahi. Error estimation, adaptivity and data transfer in enriched plasticity continua to analysis of shear band localization. *Applied Mathematical Modelling*, 31(6):983 – 1000, 2007.
- [163] M. Kitamra, H. Gu, and H. Nobukawa. A study of applying the superconvergent patch recovery (SPR) method to large deformation problem. *Journal of the Society of Naval Architects of Japan*, (187):201–208, 2000.
- [164] D. Komatitsch, D. Michéa, and G. Erlebacher. Porting a high-order finite-element earthquake modeling application to nvidia graphics cards using cuda. *Journal of Parallel and Distributed Computing*, 69(5):451 – 460, 2009.
- [165] M. Krizek and P. Neittaanmaki. On superconvergence techniques. *Acta Applicandae Mathematicae*, 9:175–198, 1987.
- [166] Y. Krongauz and T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131:133 – 145, 1996.
- [167] P. Krysl and T. Belytschko. Analysis of thin plates by the element-free Galerkin method. *Computational Mechanics*, 17:26–35, 1995.
- [168] P. Krysl and T. Belytschko. Analysis of thin shells by the element-free Galerkin method. *International Journal of Solids and Structures*, 33(20-22):3057 – 3080, 1996.
- [169] J.L. Chenot L. Fourment. Error estimators for viscoplastic materials: application to forming processes. *Engineering Computations*, 12:469 – 490, 1995.
- [170] J.C. Lachat and J.O. Watson. Effective numerical treatment of boundary integral equations: a formulation for three dimensional elastostatics. *International Journal for Numerical Methods in Engineering*, 10:991–1005, 1976.
- [171] P. Ladevèze. Constitutive relation errors for F.E. analysis considering (visco-) plasticity and damage. *International Journal for Numerical Methods in Engineering*, 52(5-6):527–542, 2001.
- [172] P. Ladevèze and D. Leguillon. Error estimate procedure in the finite element method and applications. *SIAM Journal on Numerical Analysis*, 20(3):485–509, 1983.
-

-
- [173] P. Ladevèze and N. Moës. A new a posteriori error estimation for nonlinear time-dependent finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 157(12):45 – 68, 1998.
- [174] P. Ladevèze and N. Moës. A posteriori constitutive relation error estimators for nonlinear finite element analysis and adaptive control. In P. Ladevèze and J.T. Oden, editors, *Advances in Adaptive Computational Methods in Mechanics*, volume 47 of *Studies in Applied Mechanics*, pages 231 – 256. Elsevier, 1998.
- [175] P. Ladevèze, N. Moës, and B. Douchin. Constitutive relation error estimators for (visco)plastic finite element analysis with softening. *Computer Methods in Applied Mechanics and Engineering*, 176:247 – 264, 1999.
- [176] P. Ladevèze, J.P. Pelle, and Ph. Rougeot. Error estimation and mesh optimization for classical finite elements. *Engineering Computations*, 8(3):69–80, 1991.
- [177] P. Ladevèze and Ph. Rougeot. New advances on a posteriori error on constitutive relation in F.E. analysis. *Computer Methods in Applied Mechanics and Engineering*, 150(14):239 – 249, 1997.
- [178] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.
- [179] F. Larsson and K. Runesson. Modeling and discretization errors in hyperelasto-(visco-)plasticity with a view to hierarchical modeling. *Computer Methods in Applied Mechanics and Engineering*, 193:5283 – 5300, 2004.
- [180] C. V. Le. *Novel numerical procedures for limit analysis of structures - meshfree methods and mathematical programming*. PhD thesis, University of Sheffield, 2010.
- [181] C. V. Le, H. Askes, and M. Gilbert. Adaptive element-free Galerkin method applied to the limit analysis of plates. *Computer Methods in Applied Mechanics and Engineering*, 199(37-40):2487 – 2496, 2010.
- [182] C. K. Lee and C. E. Zhou. On error estimation and adaptive refinement for element free Galerkin method: Part I: stress recovery and a posteriori error estimation. *Computers & Structures*, 82(4-5):413 – 428, 2004.
- [183] C. K. Lee and C. E. Zhou. On error estimation and adaptive refinement for element free Galerkin method: Part II: adaptive refinement. *Computers & Structures*, 82(4-5):429 – 443, 2004.
- [184] G.H. Lee, H.J. Chung, and C.K. Choi. Adaptive crack propagation analysis with the element-free Galerkin method. *International Journal for Numerical Methods in Engineering*, 56(3):331–350, 2003.
- [185] R. Leland and B. Hendrickson. *The Chaco user's guide: version 2.0*. Sandia National Labs, Albuquerque, NM, 1995. Technical Report SAND94-2692.
- [186] N. Levine. Superconvergent recovery of the gradient from piecewise linear finite-element approximations. *IMA Journal of Numerical Analysis*, 5(4):407–427, 1985.
- [187] G. Y. Li and T. Belytschko. Element-free Galerkin method for contact problems in metal forming analysis. *Engineering Computations*, 18:62–78, 2001.
-

-
- [188] S. Li and W. K. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1):1–34, 2002.
- [189] S. Li and W. K. Liu. *Meshfree Particle Methods*. Springer, 2004.
- [190] T. Li. *Stress integration strategies for a new hardening/softening elasto-plasticity model for structural concrete*. PhD thesis, University of Sheffield, UK, 2005.
- [191] L. D. Libersky, A. G. Petschek, T. C. Carney, J. R. Hipp, and F. A. Allahdadi. High strain lagrangian hydrodynamics: A three-dimensional SPH code for dynamic material response. *Journal of Computational Physics*, 109(1):67 – 75, 1993.
- [192] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its applications in applied mechanics. *Computers & Structures*, 11:83–95, 1980.
- [193] G. R. Liu. *Mesh Free Methods: Moving beyond the Finite Element Method*. CRC press, 2nd edition, 2010.
- [194] G. R. Liu and Y. T. Gu. A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering*, 50(4):937–951, 2001.
- [195] G. R. Liu and Y. T. Gu. A truly meshless method based on the strong-weak form. In *Advances in Meshfree and X-FEM Methods, Proceeding of the 1st Asian Workshop in Meshfree Methods, Singapore, (Ed. Liu GR)*, pages 259–261, 2002.
- [196] G. R. Liu and Y. T. Gu. *An Introduction to Meshfree Methods and their Programming*. Springer, 2005.
- [197] G. R. Liu and Z. H. Tu. An adaptive procedure based on background cells for meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 191(17-18):1923 – 1943, 2002.
- [198] G.R. Liu and Y.T. Gu. A local radial point interpolation method (LRPIM) for free vibration analyses of 2-D solids. *Journal of Sound and Vibration*, 246(1):29 – 46, 2001.
- [199] L. Liu, X. Dong, and C. Li. Adaptive finite element-element-free Galerkin coupling method for bulk metal forming processes. *Journal of Zhejiang University - Science A*, 10:353–360, 2009.
- [200] L. Liu, G. R. Liu, and V. B. C. Tan. Element free method for static and free vibration analysis of spatial thin shell structures. *Computer Methods in Applied Mechanics and Engineering*, 191(51-52):5923 – 5942, 2002.
- [201] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995.
- [202] X. Liu, G.R. Liu, K. Tai, and K.Y. Lam. Radial point interpolation collocation method (RPICM) for partial differential equations. *Computers & Mathematics with Applications*, 50:1425 – 1442, 2005.
- [203] Y. Liu, J. Chen, S. Yu, and C. Li. Numerical simulation of three-dimensional bulk forming processes by the element-free Galerkin method. *The International Journal of Advanced Manufacturing Technology*, 36(5):442–450, March 2008.
- [204] D. L. Logan. *A First Course in the Finite Element Method*. Thomson Canada Limited, 4th edition, 2007.
-

-
- [205] H. Lu. *Progressive adaptivity formulation in Galerkin meshfree method*. PhD thesis, The University of Iowa, United States – Iowa, 2001.
 - [206] P. Lu, G. Zhao, Y. Guan, and X. Wu. Bulk metal forming process simulation based on rigid-plastic/viscoplastic element free Galerkin method. *Materials Science and Engineering: A*, 479(12):197 – 212, 2008.
 - [207] P. Lu, G. Zhao, Y. Guan, and X. Wu. Research on rigid/visco-plastic element-free Galerkin method and key simulation techniques for three-dimensional bulk metal forming processes. *The International Journal of Advanced Manufacturing Technology*, 53:485–503, 2011.
 - [208] Y. Y. Lu, T. Belytschko, and L. Gu. A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 113(3-4):397 – 414, 1994.
 - [209] L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:10131024, 1977.
 - [210] J. C. Luo and M. B. Friedman. A parallel computational model for the finite element method on a memory-sharing multiprocessor computer. *Computer Methods in Applied Mechanics and Engineering*, 84(2):193 – 209, 1990.
 - [211] Y. Luo and U. Häussler-Combe. A gradient-based adaptation procedure and its implementation in the element-free Galerkin method. *International Journal for Numerical Methods in Engineering*, 56(9):1335–1354, 2003.
 - [212] R. J. Mackinnon and G. F. Carey. Superconvergent derivatives: A taylor series analysis. *International Journal for Numerical Methods in Engineering*, 28(3):489–509, 1989.
 - [213] H. C. Martin. Plane elasticity problems and the direct stiffness method. *The Trend in Engineering*, 13:5–19, 1961.
 - [214] D.F. Medina and J.K. Chen. Three-dimensional simulations of impact induced damage in composite structures using the parallelized SPH method. *Composites Part A: Applied Science and Manufacturing*, 31(8):853 – 860, 2000.
 - [215] R. J. Melosh. A stiffness matrix for the analysis of thin plates in bending. *Journal of the Aerospace Sciences*, 28 (1):34–42, 1961.
 - [216] R. J. Melosh. Structural analysis of solids. In *Journal of the Structural Division, Proceedings of the American Society of Civil Engineers*, pages 205–223, 1963.
 - [217] P. Metsis and M. Papadrakakis. Overlapping and non-overlapping domain decomposition methods for large-scale meshless EFG simulations. *Computer Methods in Applied Mechanics and Engineering*, 229 - 232(0):128 – 141, 2012.
 - [218] C. Miehe. Comparison of two algorithms for the computation of fourth-order isotropic tensor functions. *Computers & Structures*, 66(1):37 – 43, 1998.
 - [219] D. Millán, A. Rosolen, and M. Arroyo. Thin shell analysis from scattered points with maximum-entropy approximants. *International Journal for Numerical Methods in Engineering*, 85(6):723–751, 2011.
-

-
- [220] T. Moan. Experiences with orthogonal polynomials and "best" numerical integration formulas on a triangle; with particular reference to finite element approximations. *ZAMM - Journal of Applied Mathematics and Mechanics*, 54(7):501–508, 1974.
- [221] T. K. Molstad. Finite deformation analysis using the finite element method. Master's thesis, The University of British Columbia, 1977.
- [222] J. Monaghan. Why particle methods work. *SIAM Journal on Scientific and Statistical Computing*, 3(4):422–433, 1982.
- [223] J. J. Monaghan. An introduction to SPH. *Computer Physics Communications*, 48(1):89–96, 1988.
- [224] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomical and Astrophysics*, 30:543–574, 1992.
- [225] A. Moore. A tutorial on kd-trees. Technical Report 209, University of Cambridge Computer Laboratory. Extract from PhD thesis., 1991.
- [226] H. Moslemi and A.R. Khoei. 3D adaptive finite element modeling of non-planar curved crack growth using the weighted superconvergent patch recovery method. *Engineering Fracture Mechanics*, 76(11):1703 – 1728, 2009.
- [227] S. Mukherjee, M. K. Chati, and X. Shi. Evaluation of nearly singular integrals in boundary element contour and node methods for three-dimensional linear elasticity. *International Journal of Solids and Structures*, 37(51):7633 – 7654, 2000.
- [228] Y. X. Mukherjee and S. Mukherjee. The boundary node method for potential problems. *International Journal for Numerical Methods in Engineering*, 40(5):797–815, 1997.
- [229] A. Munjiza and K.R.F. Andrews. NBS contact detection algorithm for bodies of similar size. *International Journal for Numerical Methods in Engineering*, 43:131–149, 1998.
- [230] J.C. Nagtegaal, D.M. Parks, and J.R. Rice. On numerically accurate finite element solutions in the fully plastic range. *Computer Methods in Applied Mechanics and Engineering*, 4(2):153 – 177, 1974.
- [231] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
- [232] H. Netuzhylov. Enforcement of boundary conditions in meshfree methods using interpolating moving least squares. *Engineering Analysis with Boundary Elements*, 32(6):512 – 516, 2008.
- [233] V. P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3):763 – 813, 2008.
- [234] E. Oñate, S. Idelsohn, O. C. Zienkiewicz, and R. L. Taylor. A finite point method in computational mechanics. applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, 39(22):3839–3866, 1996.
- [235] J. T. Oden and H. J. Brauchli. On the calculation of consistent stress distributions in finite element approximations. *International Journal for Numerical Methods in Engineering*, 3(3):317–325, 1971.
-

-
- [236] J. T. Oden and J. N. Reddy. Note on an approximate method for computing consistent conjugate stresses in elastic finite elements. *International Journal for Numerical Methods in Engineering*, 6(1):55–61, 1973.
 - [237] J.T. Oden and S. Prudhomme. Estimation of modeling error in computational mechanics. *Journal of Computational Physics*, 182(2):496 – 515, 2002.
 - [238] J.T. Oden, S. Prudhomme, D. C. Hammerand, and M. S. Kuczma. Modeling error and adaptivity in nonlinear continuum mechanics. *Computer Methods in Applied Mechanics and Engineering*, 190(4950):6663 – 6684, 2001.
 - [239] K. Onishi, T. Kuroki, and M. Tanaka. An application of a boundary element method to natural convection. *Applied Mathematical Modelling*, 8(6):383 – 390, 1984.
 - [240] K. Onishi, T. Kuroki, and M. Tanaka. An application of boundary element method to incompressible laminar viscous flows. *Engineering Analysis*, 1(3):122 – 127, 1984.
 - [241] Y. Onishi and K. Amaya. A novel meshfree method for large deformation analysis of elastic and viscoelastic bodies without using background cells. *Journal of Solid Mechanics and Materials Engineering*, 4(11):1673–1686, 2010.
 - [242] Y. Onishi and K. Amaya. Floating stress-point integration meshfree method for large deformation analysis of elastic and elastoplastic materials. *International Journal for Numerical Methods in Engineering*, 92(1):36–55, 2012.
 - [243] D. J. Organ. *Numerical solution of dynamic fracture problems using the element-free Galerkin method*. PhD thesis, Northwestern Univeristy, 1996.
 - [244] A. Ortiz, M.A. Puso, and N. Sukumar. Maximum-entropy meshfree method for compressible and near-incompressible elasticity. *Computer Methods in Applied Mechanics and Engineering*, 199(25-28):1859 – 1871, 2010.
 - [245] A. Ortiz, M.A. Puso, and N. Sukumar. Maximum-entropy meshfree method for incompressible media problems. *Finite Elements in Analysis and Design*, 47(6):572 – 585, 2011.
 - [246] D. R. J. Owen and E. Hinton. *Finite elements in plasticity: theory and practice*. Pineridge Press, Swansea, UK., 1980.
 - [247] P. Pacheco. *An Introduction to Parallel Programming*. Elsevier Inc, 2011.
 - [248] Y. H. Park. Rigid-plastic analysis for metal forming processes using a reproducing kernel particle method. *Journal of Materials Processing Technology*, 183(23):256 – 263, 2007.
 - [249] D. Perić, C. Hochard, M. Dutko, and D. R. J. Owen. Transfer operators for evolving meshes in small strain elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 137(3-4):331 – 344, 1996.
 - [250] D. Perić, M. Vaz Jr., and D.R.J. Owen. On adaptive strategies for large deformations of elasto-plastic solids at finite strains: computational issues and industrial applications. *Computer Methods in Applied Mechanics and Engineering*, 176(1-4):279 – 312, 1999.
 - [251] A. Portela, M.H. Aliabadi, and D.P. Rooke. The dual boundary element method - effective implementation for crack problems. *International Journal for Numerical Methods in Engineering*, 33(6):1269–1287, 1992.
-

-
- [252] H. G. Poulos and E. H. Davis. *Elastic solutions for soil and rock mechanics*. John Wiley and Sons, New York, 1974.
- [253] S. Prudhomme and J.T. Oden. On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors. *Computer Methods in Applied Mechanics and Engineering*, 176:313 – 331, 1999.
- [254] T. Rabczuk, P. M. A. Areias, and T. Belytschko. A simplified mesh-free method for shear bands with cohesive surfaces. *International Journal for Numerical Methods in Engineering*, 69(5):993–1021, 2007.
- [255] T. Rabczuk and P.M.A. Areias. A new approach for modelling slip lines in geological materials with cohesive models. *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(11):1159–1172, 2006.
- [256] T. Rabczuk and T. Belytschko. Adaptivity for structured meshfree particle methods in 2D and 3D. *International Journal for Numerical Methods in Engineering*, 63(11):1559–1582, 2005.
- [257] T. Rabczuk and T. Belytschko. Application of particle methods to static fracture of reinforced concrete structures. *International Journal of Fracture*, 137:19–49, 2006.
- [258] T. Rabczuk and E. Samaniego. Discontinuous modelling of shear bands using adaptive meshfree methods. *Computer Methods in Applied Mechanics and Engineering*, 197(6-8):641–658, 2008.
- [259] T. Rabczuk, S. P. Xiao, and M. Sauer. Coupling of mesh-free methods with finite elements: basic concepts and test results. *Communications in Numerical Methods in Engineering*, 22(10):1031–1065, 2006.
- [260] B.N. Rao and S. Rahman. A coupled meshless-finite element method for fracture analysis of cracks. *International Journal of Pressure Vessels and Piping*, 78(9):647 – 657, 2001.
- [261] T. Rauber and G. Rünger. *Parallel Programming for Multicore and Cluster Systems*. Springer, 2010.
- [262] M. Rüter and E. Stein. Goal-oriented a posteriori error estimates in linear elastic fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, 195:251 – 278, 2006.
- [263] A. Rosolen, D. Mill e n, and M. Arroyo. On the optimum support size in meshfree methods: A variational adaptivity approach with maximum-entropy approximants. *International Journal for Numerical Methods in Engineering*, 82(7):868–895, 2010.
- [264] R. Rossi and M. Alves. On the analysis of an EFG method under large deformations and volumetric locking. *Computational Mechanics*, 39:381–399, 2007.
- [265] C. H. Rycroft. *Multiscale modeling in granular flow*. PhD thesis, Massachusetts Institute of Technology, 2007.
- [266] C. H. Rycroft, G. S. Grest, J. W. Landry, and M. Z. Bazant. Analysis of granular flow in a pebble-bed nuclear reactor. *Physical Review E*, 74 (2):021306, 2006.
- [267] A.M. Sallam. *Studies on Modeling Angular Soil Particles Using the Discrete Element Method*. PhD thesis, University of South Florida, 2004.
-

-
- [268] X. Shangwu, W.K. Liu, J. Cao, J.M.C. Rodrigues, and P.A.F. Martins. On the utilization of the reproducing kernel particle method for the numerical simulation of plane strain rolling. *International Journal of Machine Tools and Manufacture*, 43(1):89 – 102, 2003.
- [269] C. E. Shannon. A mathematical theory of communication. *The Bell Systems Technical Journal*, 27:379–423, 1948.
- [270] M. Shirazaki and G. Yagawa. Large-scale parallel flow analysis based on free mesh method: a virtually meshless method. *Computer Methods in Applied Mechanics and Engineering*, 174:419 – 431, 1999.
- [271] J. C. Simo and T. R. Hughes. *Computational inelasticity*. Springer Verlag, NY, 1998.
- [272] J.C. Simo. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. Part II: Computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 68(1):1 – 31, 1988.
- [273] J.C. Simo and M. Ortiz. A unified approach to finite deformation elastoplastic analysis based on the use of hyperelastic constitutive equations. *Computer Methods in Applied Mechanics and Engineering*, 49(2):221 – 245, 1985.
- [274] J.C. Simo and R.L. Taylor. Consistent tangent operators for rate-independent elastoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 48(1):101 – 118, 1985.
- [275] R. Simpson. *Enrichment of the Boundary Element Method through the Partition of Unity Method for Fracture Analysis using Local and Global Formulations*. PhD thesis, Durham University, 2010.
- [276] R. Simpson and J. Trevelyan. A partition of unity enriched dual boundary element method for accurate computations in fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, 200:1 – 10, 2011.
- [277] A. Singh, I. V. Singh, and R. Prakash. Meshless element free Galerkin method for unsteady nonlinear heat transfer problems. *International Journal of Heat and Mass Transfer*, 50(5-6):1212 – 1219, 2007.
- [278] I. Singh. Heat transfer analysis of composite slabs using meshless element free Galerkin method. *Computational Mechanics*, 38:521–532, 2006.
- [279] I. V. Singh. Meshless EFG method in three-dimensional heat transfer problems: A numerical comparison, cost and error analysis. *Numerical Heat Transfer, Part A: Applications: An International Journal of Computation and Methodology*, 46(2):199–220, 2004.
- [280] I. V. Singh, K. Sandeep, and Ravi Prakash. Heat transfer analysis of two-dimensional fins using meshless element free Galerkin method. *Numerical Heat Transfer, Part A: Applications: An International Journal of Computation and Methodology*, 44(1):73–84, 2003.
- [281] I.V. Singh and P.K. Jain. Parallel EFG algorithm for heat transfer problems. *Advances in Engineering Software*, 36(8):554 – 560, 2005.
- [282] H. Slim. An introduction to parallel programming, Guide 48, Version: 2.0. Durham University (Available at: <https://www.dur.ac.uk/resources/its/info/guides/48ParallelProg.pdf>), 10 2010. (Accessed date: 2 Jan 2013).
-

-
- [283] I. M. Smith and D. V. Griffith. *Programming the Finite Element Method*. Wiley, Chichester, fourth edition, 2006.
- [284] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra. *MPI-The Complete Reference, Volume 1: The MPI Core*. MIT Press, Cambridge, MA, USA, 2nd. (revised) edition, 1998.
- [285] E. Stein. *Fundamentals: Introduction and Survey, Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2004.
- [286] E. Stein, M. Rüter, and S. Ohnimus. Error-controlled adaptive goal-oriented modeling and finite element approximations in elasticity. *Computer Methods in Applied Mechanics and Engineering*, 196:3598 – 3613, 2007.
- [287] N. Sukumar. Construction of polygonal interpolants: a maximum entropy approach. *International Journal for Numerical Methods in Engineering*, 61:2159–2181, 2004.
- [288] N. Sukumar. *Fortran 90 Library for Maximum-Entropy Basis Functions. User’s Reference Manual Version 1.4*. Code available at <http://www.imechanica.org/node/3424>, 2008.
- [289] N. Sukumar and R. W. Wright. Overview and construction of meshfree basis functions: from moving least squares to entropy approximants. *International Journal for Numerical Methods in Engineering*, 70:181–205, 2007.
- [290] L. Gu T. Belytschko and Y. Y. Lu. Fracture and crack growth by element free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering*, 2:519–534, 1994.
- [291] Z. A. Taylor, M. Cheng, and S. Ourselin. Real-time nonlinear finite element analysis for surgical simulation using graphics processing units. In Nicholas Ayache, Sbastien Ourselin, and Anthony Maeder, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2007*, volume 4791 of *Lecture Notes in Computer Science*, pages 701–708. Springer Berlin Heidelberg, 2007.
- [292] V. Thomée. High order local approximations to derivatives in the finite element method. *Mathematics of Computation*, 31:652–660, 1977.
- [293] S. P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. McGraw-Hill, New York, 1970.
- [294] S. P. Timoshenko and S. Woinkowsky-Krieger. *Theory of plates and shells*. McGraw-Hill Book Company, NY, 1959.
- [295] E. Cervera & J. Trevelyan. Evolutionary structural optimisation based on boundary representation of NURBS Part I: 2D algorithms. *Computers & Structures*, 83(23-24):1902–1916, 2005.
- [296] E. Cervera & J. Trevelyan. Evolutionary structural optimisation based on boundary representation of NURBS Part II: 3D algorithms. *Computers & Structures*, 83(23-24):1917–1929, 2005.
- [297] J. Trevelyan and G. Coates. On adaptive definition of the plane wave basis for wave boundary elements in acoustic scattering: the 2D case. *Computer Modeling in Engineering and Sciences*, 55(2):147–170, 2010.
- [298] M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp. Stiffness and deflection analysis of complex structures. *Journal of Aeronautical Science*, 23(9):805–823, 1956.
-

-
- [299] M. J. Turner, E. H. Dill, H. C. Martin, and R. J. Melosh. Large deflections of structures subjected to heating and external loads. *Journal of Aeronautical Sciences*, 27(2):97–107, 1960.
 - [300] Z. Ullah, C. E. Augarde, R. S. Crouch, and W. M. Coombs. FE-EFGM coupling using maximum entropy shape functions and its application to small and finite deformation. In O Laghrouche, A EL Kacimi, P Woodward, and G Medero, editors, *19th UK Conference on Computational Mechanics (ACME-UK)*, pages 277–280. Heriot-Watt University, 5-6 April 2011.
 - [301] Z. Ullah and C.E. Augarde. Finite deformation elasto-plastic modelling using an adaptive meshless method. *Computers & Structures*, (in press, available online: <http://dx.doi.org/10.1016/j.compstruc.2012.04.001>), 2012.
 - [302] T. Vacharasintopchai. A parallel implementation of the element-free Galerkin method on a network of PCs. Master’s thesis, School of Civil Engineering, Asian Institute of Technology, Bangkok, Thailand, 2000.
 - [303] L. Vogt, R. O. Amaya, S. Kermes, Y. Shao, C. A. Bedolla, and A. A. Guzik. Accelerating resolution-of-the-identity second-order mollerplesset quantum chemistry calculations with graphical processing units. *The Journal of Physical Chemistry A*, 112:2049–2057, 2008.
 - [304] C. H. Walshaw, M. Cross, and M. G. Everett. A localized algorithm for optimizing unstructured mesh partitions. *International Journal of High Performance Computing Applications*, 9(4):280–295, 1995.
 - [305] H. Wang, G. Li, X. Han, and Z. H. Zhong. Development of parallel 3D RKPM meshless bulk forming simulation system. *Advances in Engineering Software*, 38(2):87 – 101, 2007.
 - [306] H. P. Wang, C. T. Wu, Y. Guo, and M. E. Botkin. A coupled meshfree/finite element method for automotive crashworthiness simulations. *International Journal of Impact Engineering*, 36:1210 – 1222, 2009.
 - [307] J. G. Wang and G. R. Liu. Radial point interpolation method for elastoplastic problems. In *1st Int. Conf. On Structural Stability and Dynamics, Taipei, Taiwan*, pages 703–708, 2000.
 - [308] Y. Wang, Z. Wang, and M. Ruan. Element-free Galerkin method for free vibration of rectangular plates with interior elastic point supports and elastically restrained edges. *Journal of Shanghai University (English Edition)*, 14:187–195, 2010.
 - [309] J.O. Watson. Singular boundary elements for the analysis of cracks in plane-strain. *International Journal for Numerical Methods in Engineering*, 38 (14):2389–2411, 1995.
 - [310] J. R. Williams. Contact analysis of large numbers of interacting bodies using discrete modal methods for simulating material failure on the microscopic scale. *Engineering Computations*, 5:198–209, 1988.
 - [311] K. C. L. Wong, L. Wang, H. Zhang, H. Liu, and P. Shi. Meshfree implementation of individualized active cardiac dynamics. *Computerized Medical Imaging and Graphics*, 34(1):91 – 103, 2010. Image-Guided Surgical Planning and Therapy.
 - [312] C. C. Wu. *Meshless method for computer analysis of structure and media*. PhD thesis, University of Wisconsin-Madison, 2001.
 - [313] J. M. Xia, D. M. Wei, and R. H. Jin. Meshless analysis of geometrically nonlinear beams. *Journal of Physics: Conference Series*, 96(1):012005, 2008.
-

-
- [314] P. Xia, S.Y. Long, and K.X. Wei. An analysis for the elasto-plastic problem of the moderately thick plate using the meshless local Petrov-Galerkin method. *Engineering Analysis with Boundary Elements*, 35(7):908 – 914, 2011.
- [315] Q.Z. Xiao and M. Dhanasekar. Coupling of FE and EFG using collocation approach. *Advances in Engineering Software*, 33:507 – 515, 2002.
- [316] T. Xiaowei and S. Tadanobu. Adaptive mesh refinement and error estimate for 3-D seismic analysis of liquefiable soil considering large deformation. *Journal of natural disaster science*, 26(1):37–48, 2004.
- [317] S. Xiong, C.S. Li, J.M.C. Rodrigues, and P.A.F. Martins. Steady and non-steady state analysis of bulk forming processes by the reproducing kernel particle method. *Finite Elements in Analysis and Design*, 41(6):599 – 614, 2005.
- [318] S. Xiong, W. K. Liu, J. Cao, C.S. Li, J.M.C. Rodrigues, and P.A.F. Martins. Simulation of bulk metal forming processes using the reproducing kernel particle method. *Computers & Structures*, 83(89):574 – 587, 2005.
- [319] S. Xiong, J.M.C. Rodrigues, and P. A. F. Martins. Application of the element free Galerkin method to the simulation of plane strain rolling. *European Journal of Mechanics - A/Solids*, 23(1):77–93, 2004.
- [320] G. Yagawa, N. Soneda, and S. Yoshimura. A large scale finite element analysis using domain decomposition method on a parallel computer. *Computers & Structures*, 38:615 – 625, 1991.
- [321] G. Yanjin, W. Xin, Z. Guoqun, and L. Ping. A nonlinear numerical analysis for metal-forming process using the rigid-(visco)plastic element-free Galerkin method. *The International Journal of Advanced Manufacturing Technology*, 42(1):83–92, May 2009.
- [322] S. Yoon and J.S. Chen. Accelerated meshfree method for metal forming simulation. *Finite Elements in Analysis and Design*, 38(10):937 – 948, 2002.
- [323] W. You, H. Liang, and W. Xing-hua. Large deformation analysis of pile foundation using meshless method. *World Conference on Earthquake Engineering*, 2008.
- [324] J. Zhang, Z. Yao, and M. Tanaka. The meshless regular hybrid boundary node method for 2D linear elasticity. *Engineering Analysis with Boundary Elements*, 27(3):259 – 268, 2003.
- [325] X. Zhang, K. Z. Song, M. W. Lu, and X. Liu. Meshless methods based on collocation with radial basis functions. *Computational Mechanics*, 26(4):333–343, 2000.
- [326] J. Z. Zhu and O. C. Zienkiewicz. Superconvergence recovery technique and a posteriori error estimators. *International Journal for Numerical Methods in Engineering*, 30(7):1321–1339, 1990.
- [327] J.Z. Zhu and O.C. Zienkiewicz. A posteriori error estimation and three-dimensional automatic mesh generation. *Finite Elements in Analysis and Design*, 25:167 – 184, 1997.
- [328] T. Zhu and S. N. Atluri. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. *Computational Mechanics*, 21:211–222, 1998.
-

-
- [329] X. Zhuang. *Meshless methods: theory and application in 3D fracture modelling with level sets*. PhD thesis, School of Engineering and Computing Sciences, University of Durham, UK, 2010.
- [330] X. Zhuang, C. Heaney, and C.E. Augarde. On error control in the element-free galerkin method. *Engineering Analysis with Boundary Elements*, 36(3):351 – 360, 2012.
- [331] O. C. Zienkiewicz, X. K. Li, and S. Nakazawa. Iterative solution of mixed problems and the stress recovery procedures. *Communications in Applied Numerical Methods*, 1(1):3–9, 1985.
- [332] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24:337–357, 1987.
- [333] O. C. Zienkiewicz and J. Z. Zhu. Adaptivity and mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):783–810, 1991.
- [334] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992.
- [335] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 1992.
- [336] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method for Solid and Structural Mechanics*. Sixth edition, 2005.
- [337] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method Set*. Elsevier Science, 2005.
- [338] M. Zlamal. Superconvergence and reduced integration in the finite element method. *Mathematics of Computation*, 32:663–685, 1978.
- [339] W. Zongmin. Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications*, 8(2):1–10, 1992.
-

Appendix A

Isoparametric formulations and Jacobian

The regular FEs, e.g. rectangular in two-dimensions are not suitable to discretize irregular geometries [138]. The two-dimensional quadrilateral elements, which are originated from the four nodes rectangular elements, are an ideal choice for both regular and irregular geometries. Figure A.1 shows the original or parent four nodes rectangular element in the natural (ξ, η) coordinate system and corresponding quadrilateral element in the global Cartesian (x, y) coordinate system. The geometry of the mapped

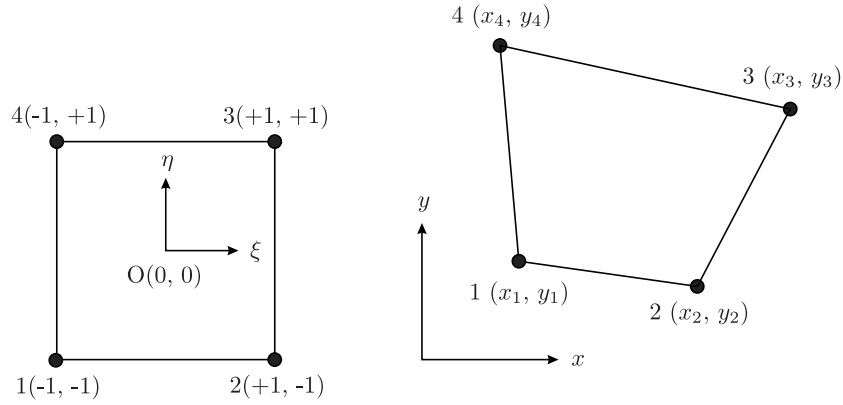


Figure A.1: Mapping of a rectangular element into an isoparametric quadrilateral element

quadrilateral element can be described in terms of shape functions $N_i(\xi, \eta)$ of the original rectangular element, i.e.

$$\begin{aligned} x &= \sum_{i=1}^4 N_i(\xi, \eta) x_i, \\ y &= \sum_{i=1}^4 N_i(\xi, \eta) y_i. \end{aligned} \quad (\text{A.1})$$

In computational mechanics, we often need to calculate the shape function derivatives w.r.t. the global coordinates, i.e. $\frac{\partial N_i}{\partial x}$, $\frac{\partial N_i}{\partial y}$, which are given as

$$\begin{aligned} \frac{\partial N_i}{\partial \xi} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi}, \\ \frac{\partial N_i}{\partial \eta} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \eta}. \end{aligned} \quad (\text{A.2})$$

Equation A.2 can also be written in the matrix form as

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix}. \quad (\text{A.3})$$

In Equation A.3, the left-hand side, i.e. the derivative of the shape functions w.r.t. to the natural coordinates are known, the 2×2 matrix on the right-hand side is known as Jacobian matrix and is written as

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}. \quad (\text{A.4})$$

Equation A.3 is written as

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix}. \quad (\text{A.5})$$

The Jacobian matrix in three-dimensions, while using (ξ, η, ψ) as natural coordinates and (x, y, z) as Cartesian coordinates, is written as

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \psi} & \frac{\partial y}{\partial \psi} & \frac{\partial z}{\partial \psi} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} z_i \\ \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} z_i \\ \sum_{i=1}^4 \frac{\partial N_i}{\partial \psi} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \psi} y_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \psi} z_i \end{bmatrix}. \quad (\text{A.6})$$

Appendix B

Neighbouring search with kd-tree

The neighbour search, which is one of the most important issues in meshless methods, is considered in this appendix. In meshless methods, a set of neighbouring nodes is required for each Gauss point for the calculation of the shape functions and corresponding shape function derivatives. This neighbour search is one of the most computationally expensive tasks in meshless methods. In this thesis from Chapters 2 to 5, the full neighbour search, which is also known as brute force or exhaustive search algorithm, has been used. In the full search algorithm, all nodes in the problem domain are searched for each Gauss point and if the Gauss point lies inside the influence domain of a node, it is attached to its neighbouring list. The main advantage of the full search algorithm is its simple implementation. The full search algorithm gives satisfactory performance, when the number of nodes is small but is not suitable for use with large number nodes due to its high computational cost. The full search algorithm has been used in many references, e.g. [36, 38, 85, 196, 233] and is also mentioned in [139]. Kd-tree and kd-tree with background mesh [29] are two other alternatives, which can be used in the neighbouring search and which may yield computational advantage. In this appendix, a comparative study is performed between the neighbouring search algorithms which include full search, kd-tree and kd-tree with background mesh.

Kd-tree is a generalization of the binary search algorithm to k dimensions and is commonly used in computer science to organize data in k -dimensional space. Kd-tree is briefly described here in computer science terms, the detail of which can be found in [73, 225]. Kd-tree is further classified into binary-tree, quad-tree and oct-tree, depending on the number of data points, each data point is connected to. In binary-tree, quad-tree and oct-tree each data point is connected to a maximum of two, four and eight data points respectively. A step by step partitioning of the space and construction of the kd-tree are shown in Figure B.1. A sample XY space with six data points is considered as shown in Figure B.1(a), an example also given in [3]. At the start of partitioning, a data point (7, 2) is picked, which is known as a root and a line is drawn through it perpendicular to the X-axis. This initial partition divides the space into two parts, where each half consists of approximately the same number of data points. Next, two more data points, i.e. (9, 6) and (5, 4) are considered, one in each half and lines are drawn through these, but this time in different directions, i.e. perpendicular to the Y-axis. These data points are called children. The procedure is now repeated for children producing more children, until lines are drawn through all the data points in the space. The children produced at the end, i.e. data points (2, 3), (4, 6) and (8, 1) in Figure B.1(a) are called leaves. The resulting kd-tree for the partitioned space in Figure B.1(a) is shown in Figure B.1(b), in which X and Y show the direction of the splitting lines through these data points. The use of the kd-tree algorithm in meshless methods consists of two parts, i.e. construction of the kd-tree and its use in the neighbouring search.

KDTREE2 [156] is used here, which is an open-source implementation of the kd-tree algorithm. KDTREE2 is available as a FORTRAN-95 module, which is very straightforward to integrate with the FORTRAN code developed in this thesis. In the kd-tree implementation, Gauss points are used to construct the kd-tree, using the KDTREE2 function “kdtree2.create”. This kd-tree is then searched for each node in the problem domain using the KDTREE2 functions “kdtree2.r_count” and “kdtree2.n_nearest” and is attached to the neighbouring list of those Gauss points, which are inside its influence domain. Another alternative for the neighbouring search is to use the kd-tree with a background mesh [29], i.e. to take advantage of the computational efficiency of having a background mesh and the kd-tree algorithm simultaneously. In this research the background mesh is used for numerical integration, in which nodes are attached to the background integration cells, as shown in Figure B.2. In the kd-tree with background mesh algorithm, nodes are used to construct the kd-tree instead of the Gauss point, which are used in the kd-tree algorithm, mentioned before. This kd-tree is then used to search neighbouring nodes for each node in the problem domain, e.g. for the node shown in red in Figure B.2, nodes shown in green are neighbouring nodes. These neighbouring nodes are then used to find neighbouring background cells, which are all those background cells to which these nodes belong. In Figure B.2, cells in which Gauss points are shown are these neighbouring background cells. The node shown in red in Figure B.2 is then included in the neighboring list of all the Gauss points belonging to these background cells. A second check can then be performed to exclude nodes from the neighbouring list of the Gauss point, when it is outside their influence domains.

To compare the implementation and performance of these three alternative neighbouring search algorithms, i.e. full search, kd-tree and kd-tree with background mesh, a three-dimensional plate with a hole problem is considered. The geometry, boundary conditions, and material properties for this problem are given in §4.4.1. A comparison between the simulation time versus degrees of freedom in the three cases is given in Table B.1 and also shown in Figure B.3. The timings reported here are the time required to assemble a stiffness matrix, which includes the neighbour search, calculation of shape functions and corresponding shape function derivatives and assembly procedure. In all the three cases, the only difference in the codes are the neighbour search. The other two parts, i.e. calculation of the shape functions and shape function derivatives, and assembly procedure are the same. It is clear from Table B.1 and Figure B.3, that the kd-tree with background mesh algorithm is computationally efficient as compared to the two other alternatives. Furthermore, there is no clear difference in the simulation times between the three alternatives when using a small number of degrees of freedom in the analysis but the difference increases with increasing numbers of degrees of freedom.

DOFs	Full search (sec)	Kd-tree (sec)	Kd-tree with backgorud search (sec)
189	0.46	0.36	0.34
975	7.93	4.38	3.75
1728	16.35	9.68	8.68
2793	32.72	19.54	16.42
6075	98.46	57.52	47.91
11253	329.97	173.43	124.44
35328	3027.47	1512.98	977.66

Table B.1: Simulation times versus degrees of freedom for the full search, kd-tree and kd-tree with background mesh algorithms

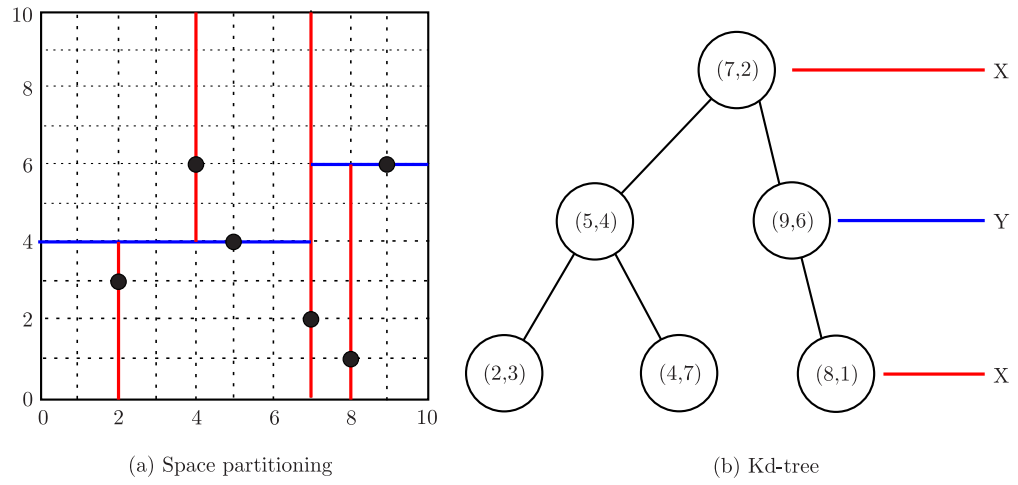


Figure B.1: Sample space partitioning and corresponding kd-tree [3]

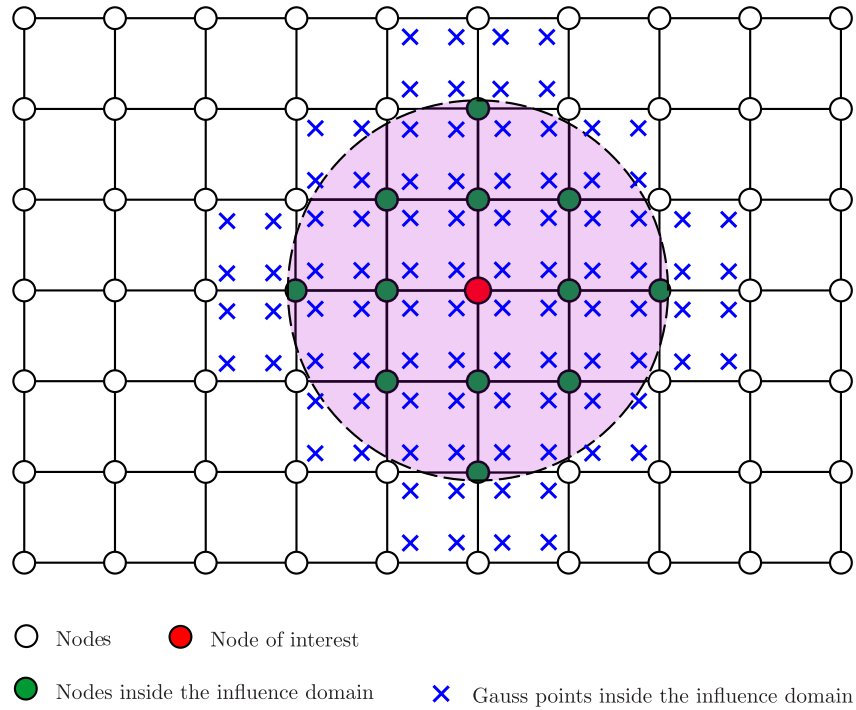


Figure B.2: Implementation of kd-tree with background mesh algorithm

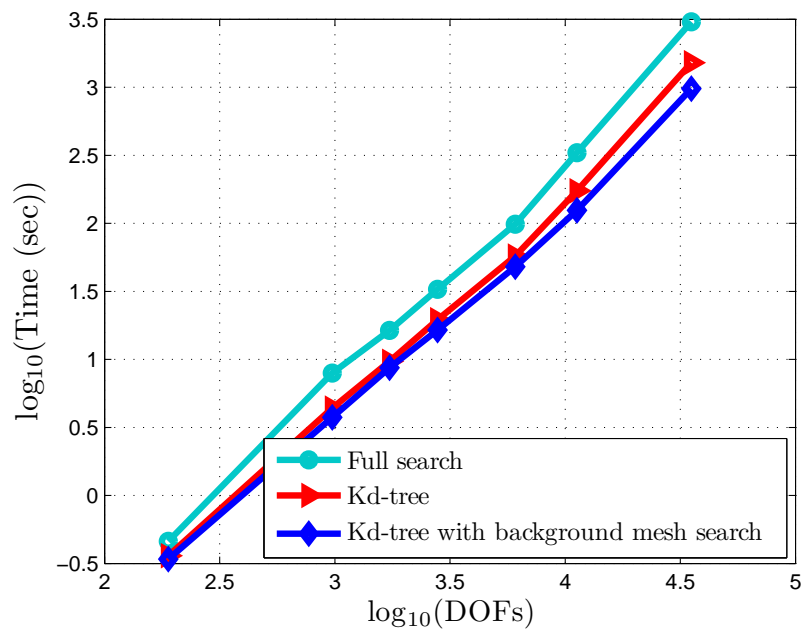


Figure B.3: Simulation times versus degrees of freedom for the full search, kd-tree and kd-tree with background mesh algorithms