Development of a novel strong-form meshless technique : random differential quadrature (RDQ) method with applications for 2-D multiphysics simulation of pH-sensitive hydrogel

Shantanu, Shashikant Mulay

2011

Shantanu, S. M. (2011). Development of a novel strong-form meshless technique : random differential quadrature (RDQ) method with applications for 2-D multiphysics simulation of pH-sensitive hydrogel. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/43540>

<https://doi.org/10.32657/10356/43540>

**DEVELOPMENT OF A NOVEL STRONG-FORM MESHLESS
TECHNIQUE – RANDOM DIFFERENTIAL QUADRATURE (RDQ)
METHOD WITH APPLICATIONS FOR 2-D MULTIPHYSICS
SIMULATION OF pH-SENSITIVE HYDROGEL**

SHANTANU SHASHIKANT MULAY

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2011

**DEVELOPMENT OF A NOVEL STRONG-FORM MESHLESS
TECHNIQUE – RANDOM DIFFERENTIAL QUADRATURE (RDQ)
METHOD WITH APPLICATIONS FOR 2-D MULTIPHYSICS
SIMULATION OF pH-SENSITIVE HYDROGEL**

SHANTANU SHASHIKANT MULAY

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2011

ACKNOWLEDGEMENTS

This research work has been completed possibly by the help of many individuals. I gratefully thank my supervisor, Dr. Li Hua, for giving me an opportunity to work in the area of meshless methods. His constant guidance, suggestions and encouragements were instrumental for me to understand this research area better and gain a level of maturity.

I wish to sincerely express my gratitude to Prof. Tom Hou from the California Institute of Technology (CalTech), whose short course on the computational fluid dynamics enabled me to work on the stability analysis. He was also helpful in providing me guidance and different references on the various aspects of the stability analysis.

I sincerely thank Prof. Sellakkutti Rajendran from the school of Mechanical and Aerospace Engineering (MAE) in clarifying my several doubts and always offering a different perspective about various topics in the finite element method (FEM).

A special thank goes to Dr. Dominic for validating my results of the elasticity problems with his FEM code. I cherish all the memories about the various technical discussions we had during his stay in the NTU.

I wish to thank all the administrative staffs of the School of MAE for their help in all the administrative matters, also a special thank goes to Mr. Teo Hai Beng of the CANES lab for his persistent efforts while addressing my needs of the software, hardware and the Linux server.

I would also like to thank my colleagues and friends, Dominic, Raman, Fukun, Zou Hua, Ye Ting, Suparna, Shashwat, Ajay, Pushkar and Prathmesh for making my stay more enjoyable and memorable.

I also sincerely acknowledge the constant support, love and encouragement of my parents, Mr. Shashikant and Mrs. Mangala, brother Kaustubh, and sister Mrs. Janki. They sacrificed their wishes to make my dream come true. I feel very fortunate to have such a wonderful family.

I owe special thanks to my wife, Supriti who has been my source of inspiration during the stressful period of my research. She was patient enough to discuss the various topics about my research while taking many long walks. I specially admire her quick way of thinking, which helped me a lot during the confusing situations. She was kind enough to check the manuscripts of my several papers as well as this thesis.

I also thank God for his grace, which helped me a lot to go past the stage when research seemed difficult.

I sincerely acknowledge the financial support provided by the NTU through the scholarship.

Shantanu Shashikant Mulay

Singapore, August 2010

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
SUMMARY	viii
LIST OF FIGURES	x
LIST OF TABLES	xxiv
Chapter 1	1
Introduction	1
1.1 Introduction.....	1
1.2 Fundamentals of meshless methods.....	2
1.3 Common steps in a meshless method	3
1.3.1 Geometry creation.....	4
1.3.2 Approximation of the field variable.....	4
1.3.3 Discretization of the governing differential equation	5
1.3.4 Assembly of the system of equations.....	6
1.3.5 Solving the assembled system of equations.....	6
1.4 Motivation of the present research work	7
1.5 Objectives of the present research work	7
1.6 Literature survey	9
1.6.1 Smooth particle hydrodynamics	11
1.6.2 Diffuse element method.....	13
1.6.3 Element-free Galerkin method.....	15
1.6.4 Natural element method.....	16
1.6.5 Reproducing kernel particle method.....	19
1.6.6 Partition of unity finite element method	24
1.6.7 Finite point method	25
1.6.8 Meshless local Petrov-Galerkin method	28
1.6.9 Local boundary integral equation method	33
1.6.10 Point interpolation method.....	36
1.6.11 Local Kriging method	38
1.6.12 Hermite cloud method.....	39
1.6.13 Gradient smoothing method.....	40
1.6.14 Radial point interpolation based finite difference method.....	42
1.7 Outline of the thesis	47
1.8 Summary	48

Chapter 2	49
Development of novel strong-form meshless random differential quadrature (RDQ) method	49
2.1 Introduction.....	49
2.2 Formulation of the fixed reproducing kernel particle method	51
2.3 Formulation of the differential quadrature method.....	55
2.4 Development of the RDQ method	57
2.4.1 Bridge to link the discretization of governing differential equation with the approximation of function value at the field nodes	58
2.4.2 Imposing the Dirichlet boundary conditions.....	60
2.4.3 Imposing the Neumann boundary conditions	61
2.4.4 Solving the final system of equations and computing the approximate function values	62
2.4.5 Computing the approximate derivatives of field variables	63
2.5 Summary	66
Chapter 3	75
Convergence and consistency analyses of RDQ method and its application for solving the microelectromechanical systems	75
3.1 Introduction to the convergence analysis.....	75
3.2 Development of the superconvergence condition.....	76
3.3 The convergence analysis	78
3.3.1 Computation of the convergence rate for the distribution of random field nodes	81
3.3.2 Remarks on the effect of random nodes on the convergence rate	81
3.3.3 1-D test problems.....	83
3.3.4 2-D test problems.....	87
3.3.5 Elasticity problems.....	93
3.4 Application of RDQ method for solving the fixed-fixed and cantilever microswitches under the nonlinear electrostatic loading	105
3.5 Introduction to the consistency analysis of RDQ method	112
3.6 Consistency analysis of the locally applied DQ method	113
3.6.1 Consistency analysis of the 1-D wave equation by the uniform distribution of virtual nodes	114

3.6.2	Consistency analysis of 1-D wave equation by the cosine distribution of virtual nodes.....	121
3.6.3	Consistency analysis of the 1-D Laplace equation by the uniform distribution of virtual nodes.....	124
3.7	Effect of the uniform and cosine distributions of virtual nodes on the convergence of RDQ method.....	126
3.7.1	1-D test problems.....	127
3.7.2	2-D test problems.....	135
3.7.3	Elasticity problems.....	138
3.8	Summary.....	142
Chapter 4		145
Stability analysis of RDQ method using the location of zeroes of polynomials.....		145
4.1	Introduction.....	145
4.2	Stability analysis of the 1 st -order wave equation by RDQ method.....	149
4.2.1	Stability analysis of the 1 st -order wave equation by different schemes for discretization of domains.....	150
4.2.2	Consistency analysis of the stable schemes and their verification by numerically implementing the 1 st -order wave equation by the locally applied DQ method.....	159
4.2.3	Implementation of the RDQ method for the 1 st -order wave equation by the forward time and central space scheme.....	163
4.2.4	Remarks on the solution of the 1 st -order wave equation by the RDQ method.....	166
4.3	Stability analysis of transient heat conduction equation.....	169
4.3.1	Forward time and forward space scheme.....	169
4.3.2	Forward time and central space scheme.....	170
4.4	Stability analysis of the transverse beam deflection equation.....	177
4.4.1	Explicit approach to solve the transverse beam deflection equation by the RDQ method.....	177
4.4.2	Implicit approach to solve the transverse beam deflection equation by the RDQ method.....	184
4.5	Summary.....	187
Chapter 5		190
Adaptive analysis of the RDQ method and the development of adaptive random differential quadrature (ARDQ) method with application in the locally high gradient problems.....		190

5.1	Introduction.....	190
5.2	Error recovery technique in the ARDQ method	195
5.3	Adaptive RDQ method	197
5.3.1	Computation of error in the ARDQ method	197
5.3.2	Procedure of adaptive refinement in the ARDQ method.....	199
5.4	Convergence analysis of ARDQ method.....	204
5.4.1	1-D test problems.....	204
5.4.2	2-D test problems.....	210
5.4.3	Semi-infinite plate with a central hole	224
5.5	Summary	226
Chapter 6		230
2-D simulation of pH-sensitive hydrogels by the RDQ method		230
6.1	Introduction.....	230
6.2	Model development of the 2-D pH-sensitive hydrogel.....	235
6.2.1	Model of 2-D pH-sensitive hydrogel	235
6.2.2	Classification of the PNP equations.....	240
6.2.3	Derivation of the analytical displacements along the moving boundaries 3 and 4 of the hydrogel	241
6.2.4	Discretization of PNP equations and the formulation of Jacobian	243
6.2.5	Exact imposition of the Neumann boundary conditions.....	247
6.2.6	Implementation of the RDQ method for the 2-D simulation of pH-sensitive hydrogels.....	248
6.3	2-D simulation of pH-sensitive hydrogels by the RDQ method.....	251
6.4	Effects of the solution pH and initial fixed-charge concentration on the swelling of 2-D hydrogel.....	260
6.5	Effects of the Young's moduli and geometrical shape of hydrogel at the dry state on the swelling of hydrogel	264
6.6	Summary	280
Chapter 7		283
Conclusions and the recommendations for future work.....		283
7.1	Conclusions from the present research	283
7.1.1	Convergence analysis of the RDQ method.....	283
7.1.2	Consistency analysis of the RDQ method	285
7.1.3	Stability analysis of the RDQ method	285
7.1.4	Adaptive analysis of the RDQ method	285

7.1.5	Study of the pull-in instability in the fixed-fixed and cantilever microswitches under the nonlinear electrostatic load	286
7.1.6	2-D simulation of pH-sensitive hydrogel by the RDQ method	287
7.2	Recommendations for the possible future work	287
7.2.1	Criterion to decide the number of total nodes in the local domain of interpolation	287
7.2.2	Development of a priori or posteriori error estimators for the meshless methods	288
7.2.3	2-D Transient simulation of pH-sensitive hydrogel.....	288
7.2.4	Incorporation of the large deformation theory in the 2-D simulation of pH sensitive hydrogels.....	288
7.2.5	Application of the RDQ method in the fluid-structure interaction problems	288
7.2.6	Development of a novel numerical strong-form meshless method	289
7.3	Summary	292
REFERENCES.....		293
Appendix A: Derivation of characteristic polynomial $\phi(z)$		312
Appendix B: Definition of reduced polynomial $\phi_1(z)$		314
Appendix C: Derivation of discretization equation by Taylor series ..		315
Appendix D: Derivation of the ratio of successive amplitude reduction values for the fixed-fixed beam using the explicit and implicit approaches.....		317
List of publications arising from the thesis		319

SUMMARY

Differential Quadrature (DQ) is one of the efficient techniques for derivative approximation, but it always requires a regular domain discretized with all the points distributed in a fixed pattern only along the straight lines. This severely restricts the DQ while solving problems with the irregular domain discretized by the random field nodes. This limitation of the DQ method is overcome by the presently proposed novel strong-form meshless method, called the random differential quadrature (RDQ) method. This method extends the applicability of the DQ technique over the irregular or regular domain discretized by the field nodes distributed randomly. In the RDQ method, the governing differential equation is discretized with the locally applied DQ method, and the value of function is interpolated approximately by the fixed reproducing kernel particle method. A superconvergence condition is developed first for the RDQ method, which gives more than $O(h^{p+1})$ convergence rate of the function for the uniform or random field nodes scattered in the domain, where p is the highest order of the monomials used in the approximation of function. Approximate derivatives of the function, computed by the RDQ method, are then evaluated by the novel approaches, called the weighted derivative and improved weighted derivative.

The convergence analysis of the RDQ method is then performed by solving several 1-D, 2-D, and the elasticity problems with locally high gradients of the field variable distributions. It is observed from the results that the approaches termed the weighted derivative and improved weighted derivative provide satisfactory rates of the derivative convergence for the field nodes distributed either in the uniform or random manner.

The consistency analysis of the RDQ method is also performed via the 1-D wave and Laplace equations. It is found that the consistency equation obtained by discretizing the governing equation with the virtual nodes of cosine distribution converges to the equation obtained by the virtual nodes distributed uniformly for the sufficiently large number of cosine virtual nodes.

The stability analysis of the RDQ method is further performed for the 1st-order wave, transient heat conduction and transverse beam deflection equations by the Schur and

Neumann polynomials, and it is shown that the RDQ method can be well adjusted for the stability analysis.

The adaptive RDQ (ARDQ) method is finally developed by implementing the adaptive algorithm of h -refinement with the RDQ method. A novel approach with the cross product of vectors is developed to ensure the newly created field nodes to be always within the convex hull of the computational domain. An error recovery technique is also proposed to ensure the error in the solution to be always minimized irrespective of the number of field nodes used in the local interpolation domain of the field variables. Several 1-D and 2-D test problems of the locally high gradients are solved successfully for the convergence analysis of the ARDQ method, and it is observed that the convergence rates obtained by the ARDQ method are indeed improved as compared with those by the RDQ method.

In addition, two configurations of the microswitches, based on the fixed-fixed and cantilever beams, are simulated successfully by the RDQ method to study their behaviour for the pull-in instability caused by the applied voltage. It is found that the pull-in voltages computed by the RDQ method match closely with the experimental results published in open literature.

Finally, 2-D simulation of the pH-sensitive hydrogel is performed to examine the capability of the RDQ method in solving the complex multiphysics problems characterized by the coupled nonlinear partial differential equations. It is validated that the RDQ method smoothly captures the jump of the field variables across the interface between the hydrogel and solution. The analysis and discussion demonstrate that the simulation results are consistent with the nature of the problem.

LIST OF FIGURES

Fig. 1.1. General flowchart of a meshless method.....	3
Fig. 1.2. Local domain for interpolation.....	4
Fig. 1.3. Computation of natural-neighbour coordinates in the NEM.	17
Fig. 1.4. Schematic of different domains in the MLPG method.....	29
Fig. 1.5. Schematic of GSM working principle.	41
Fig. 1.6. Schematic of the RFDM working principle.....	46
Fig. 2.1. Working principle of the RDQ method for the irregular computational domain discretized by the random field nodes coupled with virtual nodes distributed in a fixed pattern.....	50
Fig. 2.2. Relative positioning of field nodes i and j	65
Fig. 3.1. Convergence plots by the uniform and random distributions of the field nodes combined with the cosine virtual nodes for the 1 st 1-D problem of Poisson equation.	84
Fig. 3.2. Comparison between the numerical and analytical values of function by uniform (a) and random (b) field nodes for the 1 st 1-D problem of Poisson equation.....	84
Fig. 3.3. Convergence curves by uniform (a) and random (b) field nodes for the 2 nd 1-D problem of Poisson equation, where the curves are plotted with the field nodes obtained by uniformly decreasing h and the superconvergence condition.....	86
Fig. 3.4. Reduction in the complete error norm with the field nodes obtained by superconvergence condition (a), and convergence plots by the uniform and random field nodes with the 4 th -order monomials (b) for the 2 nd 1-D problem.	86
Fig. 3.5. Comparison between the numerical and analytical values of function by uniform (a) and random (b) field nodes for the 2 nd 1-D problem.	87
Fig. 3.6. Convergence curves by uniform (a) and random (b) field nodes, and the reduction in the Sobolev error norm (c) for the 3 rd 1-D problem of local high gradient.....	88
Fig. 3.7. Convergence curves by uniform (a) and random (b) field nodes by including up to the 2 nd -order monomials for the 1 st 2-D problem of Laplace equation.....	89

Fig. 3.8. Comparison between the numerical and analytical values of function by 33×33 uniform (a) and random (b) field nodes for the 1 st 2-D problem of Laplace equation.	89
Fig. 3.9. Convergence plots by uniform (a) and random (b) field nodes for the 2 nd 2-D problem of Laplace equation.	90
Fig. 3.10. Convergence plots by uniform (a) and random (b) field nodes, and the numerical and analytical distributions of the field variable by uniform field nodes (c) for the 3 rd 2-D problem.	91
Fig. 3.11. Convergence plots by uniform field nodes (a), reduction in Sobolev norm (b), and the numerical distribution of temperature within the domain (c) for the 4 th 2-D problem of steady-state heat conduction.	92
Fig. 3.12. Schematic of a cantilever beam under pure bending load (a), and the convergence plots of displacements u and v by the uniform and random field nodes coupled with the cosine virtual nodes (b).	95
Fig. 3.13. Analytical displacement u (a), numerical displacement u (b), analytical displacement v (c), and numerical displacement v (d), where the domain is discretized by 9×9 uniform field and 41×41 cosine virtual nodes and the functions are approximated by the 2 nd -order monomials in the polynomial basis for a cantilever beam under pure bending load.	96
Fig. 3.14. Schematic of a cantilever beam under a pure shear load (a), and the convergence plots of the displacements u and v by the uniform and random field nodes coupled with the cosine virtual nodes (b).	97
Fig. 3.15. Comparison of the displacements u (a) and v (b) by the 3 rd -order monomials in the polynomial basis of function approximation for a cantilever beam under pure shear load.	99
Fig. 3.16. Analytical σ_{xx} (a), numerical σ_{xx} (b), analytical σ_{xy} (c) and numerical σ_{xy} (d) stresses, where the domain is discretized by 9×9 uniform field and 41×41 cosine virtual nodes for a cantilever beam under a pure shear load.	99
Fig. 3.17. Schematic of a beam under the loading of constant normal stresses.	101

Fig. 3.18. The displacements u (a) and v (b), and stresses σ_{xx} (c) and σ_{yy} (d) when the beam under the normal stresses is solved by uniform field nodes coupled with cosine virtual nodes.....	102
Fig. 3.19. The displacements u (a) and v (b), and stresses σ_{xx} (c) and σ_{yy} (d) when the beam under the normal stresses is solved by random field nodes coupled with cosine virtual nodes.....	103
Fig. 3.20. Schematic of the semi-infinite plate with a central hole (a), and actual computational domain with the boundary and loading conditions (b).	104
Fig. 3.21. Convergence plots of the displacements u and v by uniform field nodes (a), and comparison between the numerical and analytical values of σ_{xx} along the boundary 1 (b) for a problem of semi-infinite plate with a central hole.	104
Fig. 3.22. Schematics of the fixed-fixed (a) and cantilever (b) microswitches.	106
Fig. 3.23. Comparison between the numerical and analytical values of deflection and slope by the uniform (a) and random (b) field nodes for a fixed-fixed beam under the UDL.	107
Fig. 3.24. Peak deflection of beam for different applied voltages (a), full deflection of beam for different applied voltages (b), and convergence of peak deflection with the inner iterations at $V = 15.05$ volt (c) for a fixed-fixed microswitch under the influence of nonlinear electrostatic force.	108
Fig. 3.25. Comparison between the numerical and analytical values of deflection and slope by the uniform (a) and random (b) field nodes for a cantilever beam under the UDL.	109
Fig. 3.26. Deflection of beam under different applied voltages (a), deflection of beam at $\tilde{v} = 2.64453$ volt by modified boundary conditions (b) and the convergence of peak deflection during the inner iterations by Newton method (c).	111
Fig. 3.27. The DQ local domains at the concerned virtual node (x_m, t_n) when the virtual node is assumed in internal domain (a), at left end (b) and right end (c) of the domain for the 1-D wave equation.	119

- Fig. 3.28. The DQ local domains at the virtual node (x_m) when the virtual node is assumed inside the internal domain (a) and right end of the domain (b) for the 1-D Laplace equation. 126
- Fig. 3.29. Convergence curves when the uniform field nodes are combined with the cosine (a) and uniform (b) virtual nodes, and the random field nodes are combined with the cosine (c) and uniform (d) virtual nodes for the 1st 1-D problem. 128
- Fig. 3.30. Convergence curves of the function by five different sets of the random field nodes for the 1st 1-D problem, where the global error varies for the low number of field nodes but converges to almost the constant value as the total random field nodes are increased. 129
- Fig. 3.31. Convergence curves when uniform field nodes are combined with the cosine virtual nodes (a), random field nodes are combined with the cosine virtual nodes (b), uniform field nodes are combined with the uniform virtual nodes (c) and the random field nodes are combined with the uniform virtual nodes (d) for the 2nd 1-D problem. 132
- Fig. 3.32. Convergence curves when the uniform field nodes are combined with the cosine virtual nodes (a), random field nodes are combined with the cosine virtual nodes (b), uniform field nodes are combined with the uniform virtual nodes (c) and the random field nodes are combined with the uniform virtual nodes (d) for the 3rd 1-D problem. 133
- Fig. 3.33. Comparison between the analytical and numerical values of function f (a), the 1st derivative $f_{,x}$ (b) and the 2nd derivative $f_{,xx}$ (c) for the 3rd 1-D problem by 641 uniform field nodes combined with the 641 cosine virtual nodes, and the plots of function value convergence by five different sets of random field nodes combined with the cosine distribution of virtual nodes (d), where the global error varies significantly for the lower number of field nodes but their difference reduces with increase in the number of random field nodes. 136
- Fig. 3.34. Convergence curves when the uniform field nodes are combined with the cosine virtual nodes (a), random field nodes are combined with the cosine virtual nodes (b), uniform field nodes are combined with the uniform virtual

- nodes (c) and the random field nodes are combined with the uniform virtual nodes (d) for the 1st 2-D problem. 137
- Fig. 3.35. Computational domain and boundary conditions for a steady-state heat conduction problem with a heat source at $y = b$ (a), convergence curves for the uniform distribution of field nodes combined with the uniform and cosine distributions of virtual nodes (b), convergence curves for the random distribution of field nodes combined with the uniform and cosine distributions of virtual nodes (c), and the comparison between the numerical and analytical values of temperature along the central x-axis plane (d). 139
- Fig. 3.36. Schematic of a cantilever beam under a pure bending load (a), convergence plots obtained by combining the cosine virtual nodes with the uniform and random field nodes (b), and convergence plots obtained by combining the uniform virtual nodes with the uniform (c) and random (d) field nodes. 141
- Fig. 3.37. One quarter computational domain of a semi-infinite plate (a) and the convergence plots of the displacements u and v by the uniform field nodes combined with the cosine and uniformly distributed virtual nodes (b) for a problem of semi-infinite plate with a central hole. 142
- Fig. 4.1. Local DQ domain around a central virtual node (x_i, y_j) that is inside computational domain (a), and central time and forward space scheme of discretization of the time and space domains (b) for the 1st-order wave equation. 152
- Fig. 4.2. $|\phi_+|^2$ versus (θ/π) plot with r from 0 to 2 (a) and $|\phi_-|^2$ versus (θ/π) plot with r from 0 to 1 (b) by numerically implementing the 2nd stability scheme. 154
- Fig. 4.3. Multi-step discretization scheme of forward time and space domains (a), plot of $|\phi_+|^2$ versus (θ/π) (b) and $|\phi_-|^2$ versus (θ/π) (c) by numerically implementing this scheme with r from 0 to 1 for the 1st-order wave equation. 155
- Fig. 4.4. Single step scheme of forward time and forward space domains (a), and the plot of $|\phi|^2$ versus (θ/π) (b) with r from 0 to 1, in which the root is inside the unit circle when $|r| \leq 0.2$ for the discretization of the 1st-order wave equation. 157

Fig. 4.5. Single step scheme of forward time and central space domains (a) and the plot of amplification factor in the complex plane with a circle having a unit radius (b), where the scheme is found unstable for the discretization of the 1 st -order wave equation.	158
Fig. 4.6. The wave is preserved for $r = 1$ (a) and $r = 0.5$ (b), but the oscillations are observed for $r = 2$ (c), when the 1 st -order wave equation is solved by locally applied DQ method with the 1 st discretization scheme.....	160
Fig. 4.7. Small effect of dispersion is observed for $r = 0.2$ (a) and the oscillations are observed for $r = 0.5$ (b), when the 1 st -order wave equation is solved by the 4 th -scheme.....	162
Fig. 4.8. The wave is preserved for $r = 1$ (a), numerical damping is observed for $r = 0.5$ (b) and the oscillations are observed for $r = 2$ (c) when the 1 st -order wave equation is solved by the locally applied DQ method with the 5 th scheme.....	164
Fig. 4.9. Local interpolation of the function and the creation of DQ local domains for the discretization of governing equation at all virtual nodes by the RDQ method.	165
Fig. 4.10. The comparison between numerical and analytical function values with the spatial domain discretized by uniform (a) and random (b) field nodes coupled with the uniform virtual nodes in both the time and space domains, and the time history plot of comparison between the numerical and analytical function values at the field node $(1, t)$ when the spatial domain is discretized by the uniform and random field nodes coupled with the cosine virtual nodes (c).	168
Fig. 4.11. The $ \phi ^2$ versus (θ/π) plot by $0 \leq m \leq 0.5$ with the forward time and central space scheme (a), and the time history plot of temperature at the field node $(1, t)$ and its comparison with analytical and FEM solutions (b) for the transient heat conduction.	175
Fig. 4.12. The diffusion of temperature from the time $t = 0$ to 1 sec. by uniform field and uniform virtual nodes (a), and random field and uniform virtual nodes (b). ...	176
Fig. 4.13. The diffusion of temperature from the time $t = 0$ to 1 sec. by uniform field and cosine virtual nodes (a), and random field and cosine virtual nodes (b).	176

Fig. 4.14. Discretization stencils of the time and space domains for the transverse beam deflection problem by the explicit approach.	178
Fig. 4.15. Plots of roots $ \phi_+ ^2$ versus (θ/π) (a) and $ \phi_- ^2$ versus (θ/π) (b) for the problem of transverse beam deflection by the explicit approach.	179
Fig. 4.16. Time history plot at the field node $(0.5, t)$ by the locally applied DQ and RDQ methods (a), and the comparison of RDQ method with the FEM and analytical solutions (b) for the problem of transverse beam deflection.	183
Fig. 4.17. Discretization stencils of the time and space domains (a) and the comparison of the numerical solution of the beam deflection at the field node $(0.5, t)$ with the FEM and analytical solutions (b) when the problem of transverse beam deflection is solved by the implicit approach.	188
Fig. 5.1. Mesh refinement for the concerned node when it is located inside the domain or along the boundary (a), or at the corner of the computational domain (b) in the ARDQ method.	200
Fig. 5.2. Working principle of the convex hull approach, when the newly added field node P_n is inside (a) and outside (b) of the computational domain.	203
Fig. 5.3. Distributions of the field nodes during the successive adaptive iterations when 21 randomly distributed field nodes at the beginning of computation are combined with 1001 uniform virtual nodes for the first 1-D problem.	205
Fig. 5.4. Convergence plots when the uniformly (a) and randomly (b) distributed field nodes at the beginning of computation are combined with the uniform and cosine distributed virtual nodes for the first 1-D problem of Poisson equation.	206
Fig. 5.5. Reduction in the true error norm (a), where the lowest value is obtained by combining the random field nodes distributed at the beginning with the uniform virtual nodes, and the comparison between the numerical and analytical values of field variable distribution (b) when the first 1-D problem is solved by the ARDQ method.	207

Fig. 5.6. Distributions of the field nodes during the adaptive iterations of ARDQ method when 21 randomly distributed field nodes at the beginning of computation are combined with the uniform virtual nodes for the second 1-D problem.	208
Fig. 5.7. Convergence curves when the uniformly (a) and randomly (b) distributed field nodes at the beginning of computation are combined with the uniform and cosine virtual nodes, during the solution of second 1-D problem by the ARDQ method.....	209
Fig. 5.8. Reduction in the true error norm with increase in the field nodes when the second 1-D problem is solved by the ARDQ method.....	209
Fig. 5.9. Numerical and analytical distributions of the values of f (a), (df / dx) (b), and $(d^2 f / dx^2)$ (c) when the second 1-D problem is solved by the ARDQ method.	211
Fig. 5.10. Comparison between the values of global error norms obtained by solving the second 1-D problem using the ARDQ method coupled with the error recovery technique and the RDQ method with the uniform refinement of field nodes. .	212
Fig. 5.11. Distributions of the random field nodes at the beginning (a) and end (b) of the solution of first 2-D problem of steady-state heat conduction by the ARDQ method.....	214
Fig. 5.12. Convergence curves when the uniform and random field nodes at the beginning of computation are combined with the cosine and uniform virtual nodes (a) and the reduction in true error norm (b), where the lowest true error is obtained by the uniform field nodes at the start combined with the uniform virtual nodes for the first 2-D problem.	214
Fig. 5.13. Numerical distribution of the temperature (a) and the temperature contours over the uniform grid (b) by 625 field nodes when the first 2-D problem is solved by the ARDQ method.	215
Fig. 5.14. Distribution of temperature at the field nodes along $(0.5, y)$ (a) and the reduction in global error norm (b), where the global error obtained by the ARDQ method is lower than the one obtained by RDQ method for the equal number of field nodes, when the first 2-D problem is solved by the ARDQ method.....	215

Fig. 5.15. Initial total 25 random field nodes (a), and final total 625 field nodes (b) during the solution of the second 2-D problem by the ARDQ method.	216
Fig. 5.16. Convergence curves when the uniform and random field nodes at the beginning are separately combined with the cosine virtual nodes (a) and the reduction in true error norm (b) for the solution of the second 2-D problem by the ARDQ method.....	217
Fig. 5.17. Comparison between the analytical (a) and numerical (b) values of function for the second 2-D problem by the total 876 field nodes obtained at the end of the adaptive refinement by the ARDQ method.....	217
Fig. 5.18. Plots of the 1 st -order derivative of function with respect to x $f_{,x}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 51×51 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.....	218
Fig. 5.19. Plots of the 1 st -order derivative of function with respect to y $f_{,y}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 41×41 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.....	219
Fig. 5.20. Plots of the 2 nd -order derivative of function with respect to x $f_{,xx}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 41×41 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.....	220
Fig. 5.21. Plots of the 2 nd -order derivative of function with respect to y $f_{,yy}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 41×41 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.....	221

Fig. 5.22. Initial number of total 36 field nodes (a) and final number of total 896 field nodes (b) during the solution of the third 2-D problem by the ARDQ method.	221
Fig. 5.23. Convergence plots when the ARDQ method is coupled with the error recovery technique (a), and the convergence rates obtained with and without using the error recovery technique (b) when the third 2-D problem is solved by the ARDQ method.....	222
Fig. 5.24. Reduction in the true error norm with increase in the field nodes when the third 2-D problem is solved by the ARDQ method.	222
Fig. 5.25. Comparison between the analytical (a) and numerical (b) distributions of the temperature for the third 2-D problem.	222
Fig. 5.26. Distributions of the initial number of total 36 uniform field nodes (a) and final number of total 900 field nodes (b) obtained during the solution of the fourth 2-D problem by the ARDQ method.....	223
Fig. 5.27. Convergence curves with and without using the error recovery technique (a) and the reduction in true error norm (b) when the fourth 2-D problem is solved by the ARDQ method.	223
Fig. 5.28. Comparison between the analytical (a) and numerical (b) values of function for the fourth 2-D problem of local high gradient solved by the ARDQ method..	224
Fig. 5.29. The analytical (a) and numerical (b) plots of field variable distribution, and contour plot by 41×41 uniform grid (c) for the fourth 2-D problem.	225
Fig. 5.30. The computational domain for the problem of semi-infinite plate with a central hole.....	225
Fig. 5.31. Distributions of initial total 35 uniform field nodes (a), final total 875 nodes (b), and total 961 cosine virtual nodes (c) for the solution of the problem of semi-infinite plate with a central hole.....	227
Fig. 5.32. Convergence plots of the displacements (a) and the decrease in resultant true error norm (b) by the ARDQ method, where the convergence rates obtained by ARDQ method, based on the adaptive refinement, are higher than the RDQ method with the uniform refinement of nodes.	227

Fig. 5.33. Comparison between the numerical and analytical values of stress σ_{xx} along the boundary $x = 0$ (a), and analytical (b) and numerical (c) values of the stress σ_{xx} , where the relative error is with in $\pm 2\%$ during the final distribution of field nodes.	228
Fig. 5.34. Comparison between the numerical and analytical values of stress σ_{yy} along the boundary $y = 0$ for the problem of semi-infinite plate with a central hole by the ARDQ method.	228
Fig. 6.1. Schematic of the charged hydrogel and solution (a) and schematic of the working principle of the RDQ method by irregular computational domain (b).	231
Fig. 6.2. Schematic of the hydrogel and solution domains (a), schematic of the computational domains of the hydrogel and solution along with the solution boundary conditions (b) and schematic of the hydrogel computational domain along with the boundary conditions (c).....	236
Fig. 6.3. The computation of derivative in the y direction at the last virtual node along the boundary $y = 0$ of the hydrogel.	247
Fig. 6.4. Schematic of the approach to compute the expanded hydrogel area by the field nodes along the hydrogel boundaries 3 and 4, and their projection on the boundary 2	250
Fig. 6.5. The complete flowchart of the 2-D simulation of the pH-responsive hydrogel by the RDQ method.	252
Fig. 6.6. Schematic of the profiles of various field variables after solving the 2-D pH-responsive hydrogel for $\text{pH} = 3$, $E = 29$ and uniform field nodes, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), top view of C_3 (d), bottom view of C_3 (e) and ψ (f).....	255
Fig. 6.7. Cumulative concentration values of the mobile ions ($C_1 + C_2 + C_3$) distributed in the hydrogel and solution domains, where the electroneutrality condition is satisfied in the solution and an average value of 0.35 mM is obtained in the hydrogel domain, which closely matches with the average concentration value of hydrogel fixed-charge of 0.36 mM.....	256

Fig. 6.8. Increase in the hydrogel volume during the iterations of the coupled PNP and mechanical equilibrium equations, where the initial configuration volume was 1.75 mm^2	256
Fig. 6.9. The displacements u (a) and v (b), and the comparison between the original and deformed hydrogel (c) for a curved boundary hydrogel with the uniform distribution of field nodes, where the central node is fully constrained and the Dirichlet boundary conditions are exactly imposed.	257
Fig. 6.10. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for $\text{pH} = 3$ and random field nodes, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), and the distribution of cumulative concentrations of mobile ions (e).	258
Fig. 6.11. The displacements u (a) and v (b), and the comparison between the original and deformed hydrogel (c) for a curved boundary hydrogel with the randomly distributed field nodes.	259
Fig. 6.12. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for different concentration values of initial fixed-charge and $\text{pH} = 3$, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and cumulative mobile ions distribution (f).	261
Fig. 6.13. The displacements u (a) and v (b), and the comparison between the original and deformed hydrogel (c) for the different concentration values of the initial fixed-charge with the solution $\text{pH} = 3$	262
Fig. 6.14. The displacements u (a) and v (b) for the solution $\text{pH} = 7$, and the displacements u (c) and v (d) for the solution $\text{pH} = 11$ at the different values of the initial fixed-charge concentration.	263
Fig. 6.15. Change in the volume of the hydrogel with respect to $\text{pH} = 3, 7$ and 11 as a function of initial fixed-charge concentration (a), and change in the volume of the hydrogel with respect to the values of initial fixed-charge concentration $0.5, 1.05$ and 1.5 mM as a function of solution pH (b).	264
Fig. 6.16. Comparison between the numerical and analytical values of x (a) and y (b) coordinates along the boundaries 3 and 4, respectively, of the deformed square hydrogel, where both the deformations closely match each other.	264

Fig. 6.17. Schematic of the solution and various hydrogel domains, where the three geometrical shapes of the hydrogel are considered with the different values of the ratio $r = W / L$ as $r = 1 : 2$ (a), $r = 1 : 3$ (b) and $r = 1 : 1$ (c).	265
Fig. 6.18. Profiles of different strains ε_{xx} (a), ε_{yy} (b) and ε_{xy} (c) when the 2-D pH-sensitive hydrogel is solved for pH = 3 and $E = 29$	266
Fig. 6.19. Profiles of different stresses σ_{xx} (a), σ_{yy} (b) and σ_{xy} (c), and the surface normals to the principle planes (d) when the 2-D pH-sensitive hydrogel is solved for pH = 3 and $E = 29$	267
Fig. 6.20. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for pH = 3, and $E = 25, 29$ and 36 , where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).	268
Fig. 6.21. The displacements u (a) and v (b), and the original and deformed shapes (c) for a curved boundary hydrogel with pH = 3, and $E = 25, 29$ and 36 , where the hydrogel expansion decreases with an increase in the ratio of the normalized Young's modulus ε	269
Fig. 6.22. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for pH = 7, and $E = 25, 29$ and 36 , where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).	270
Fig. 6.23. The displacements u (a) and v (b), and the original and deformed shapes (c) for a curved boundary hydrogel with pH = 7, and $E = 25, 29$ and 36	271
Fig. 6.24. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for pH = 11, and $E = 25, 29$ and 36 , where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).	272
Fig. 6.25. The displacements u (a) and v (b), and the original and deformed shapes(c) for a curved boundary hydrogel with pH = 11, and $E = 25, 29$ and 36	273

Fig. 6.26. Influence of the solution pH on the hydrogel swelling at different values of normalized Young's modulus ε (a) and the influence of ε on the hydrogel swelling at different values of the solution pH (b).	274
Fig. 6.27. Maximum values of the major and minor principle stresses in the hydrogel at the central fully constrained node (a) and total strain energy in the hydrogel domain (b) for the different values of solution pH.	274
Fig. 6.28. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel with rectangular shape for pH = 3 and $E = 29$, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).....	276
Fig. 6.29. The displacements u (a) and v (b), and the original and deformed shapes (c) for a rectangular hydrogel with pH = 3 and $E = 29$	277
Fig. 6.30. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel with square shape for pH = 3 and $E = 29$, where the different profiles are given as C_f (a), C_1 (b), C_2 (C), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).	278
Fig. 6.31. The displacements u (a) and v (b), and the original and deformed shapes (c) for a square hydrogel domain with pH = 3 and $E = 29$	279
Fig. 6.32. Comparison between the principle stresses (a) and strain energy (b) for the different geometrical shapes of the hydrogel.	280

LIST OF TABLES

Table 3.1. Different possibilities of m_1 , m_2 and λ signs	77
Table 3.2. Convergence rates for the 2^{nd} 1-D problem by the 2^{nd} -order monomials	85
Table 3.3. Decrease in the values of global error with increase in the monomial order .	100
Table 3.4. Convergence rates for the 1^{st} 1-D problem of Poisson equation when the cosine distributed virtual nodes are combined with the uniform and randomly distributed field nodes, respectively	130
Table 3.5. Convergence rates for the 1^{st} 1-D problem of Poisson equation when the uniformly distributed virtual nodes are combined with the uniform and randomly distributed field nodes, respectively.....	131
Table 3.6. Convergence rates for the 2^{nd} 1-D problem of Poisson equation when the cosine distributed virtual nodes are combined with the uniform and randomly distributed field nodes, respectively	131
Table 3.7. Convergence rates for the 2^{nd} 1-D problem of Poisson equation by combining the uniform distribution of virtual nodes with the uniform and random field nodes, respectively.....	131
Table 3.8. Convergence rates for the 3^{rd} 1-D problem by the uniform and random distributions of the field nodes combined respectively with the cosine distribution of virtual nodes	134
Table 3.9. Convergence rates for the 3^{rd} 1-D problem by the uniform and random distributions of the field nodes combined respectively with the uniform distribution of virtual nodes	134
Table 3.10. Convergence rates for the 1^{st} 2-D problem by the cosine distribution of virtual nodes combined with the uniform and random distributions of the field nodes, respectively.....	138
Table 3.11. Convergence rates for the 1^{st} 2-D problem by combining the uniformly distributed virtual nodes with the uniformly and randomly distributed field nodes, respectively.....	138

Table 3.12. Convergence rates for the 2 nd 2-D problem by the cosine distribution of virtual nodes combined with the uniform and random distributions of the field nodes	140
Table 3.13. Convergence rates for the 2 nd 2-D problem by combining the uniform distribution of virtual nodes with the uniform and random distributions of the field nodes	140
Table 5.1. Convergence rates obtained by the ARDQ method for the first 1-D problem of Poisson equation by combining the uniform field nodes at the beginning of computation with the cosine and uniform virtual nodes.....	206
Table 5.2. Convergence rates for the first 1-D problem of Poisson equation by the ARDQ method when the random field nodes at the beginning of computation are combined with the cosine and uniform virtual nodes	208
Table 5.3. Convergence rates for the second 1-D problem of Poisson equation by combining the uniform field nodes at the beginning of computation with the cosine and uniform virtual nodes when solved by the ARDQ method	210
Table 5.4. Convergence rates for the second 1-D problem of Poisson equation by combining the random field nodes at the beginning of computation with the cosine and uniform virtual nodes when solved by the ARDQ method	210
Table 5.5. Reduction in the true error norm for the second 1-D problem when the uniform and random field nodes at the beginning of computation are respectively combined with the cosine virtual nodes.....	211
Table 5.6. Reduction in the true error norm for the second 1-D problem when the uniform and random field nodes at the beginning of computation are separately combined with the uniform virtual nodes	212
Table 5.7. Reduction in the global error norm when the second 1-D problem is solved by the ARDQ method coupled with the error recovery technique, and the RDQ method with the uniform increment in the field nodes	212
Table 5.8. Convergence rates for the first 2-D problem when the uniform and random field nodes at the beginning of computation are combined with the cosine and uniform virtual nodes.....	214

Table 5.9. Reduction in the true error norm for the first 2-D problem when the uniform and random field nodes at the beginning of computation are combined with cosine virtual nodes	216
Table 5.10. Reduction in the true error norm for the first 2-D problem when the uniform and random field nodes at the beginning of computation are combined with uniform virtual nodes.....	216
Table 6.1. List of the parameters used in the 2-D simulation of the pH-responsive hydrogels by the RDQ method.....	253

Chapter 1

Introduction

1.1 Introduction

For a long time, Finite Element Method (FEM) has been a standard tool for numerically solving different engineering problems, especially those from the mechanical engineering. Today's real world problems, such as astrophysics and diffusion phenomena, are highly complex, such that FEM alone is inadequate to solve them. Following are the several limitations of FEM,

- One of the limitations of FEM is a difficulty for generating a good quality mesh. It should be correct according to geometry and the specific requirements of a physical phenomenon, namely it is essential for the mesh to capture the order of continuity required by the underlying geometry. For example, it is not good for a linear triangular element to discretize the circular boundary that has the 2nd-order continuity; otherwise it will introduce an error. However, this problem becomes serious when the time dependent large deformation problem is involved, such as the crack propagation, astrophysics phenomenon, and the extrusion etc. Because the geometry is re-meshed after every time increment in solving these problems, it is possible for the newly generated mesh to contain the distorted elements. These distorted elements may cause numerical problems during further computation.
- The discontinuities in the domain, such as cracks, are difficult to treat by the FEM resulting in the poor quality of the mesh. When the mesh gets distorted, the results are not always correct due to the presence of high discontinuities in the approximation of the field variables.
- In general, the adaptive meshing in the FEM is known to improve the accuracy of the solution when the geometry is adequately simple, and the distribution of field variable is sufficiently smooth and continuous. However, when the geometry becomes highly complex with the distribution of field variable having a local high gradient, the robust adaptive algorithm may not fail itself, but it may compromise the accuracy of the solution by generating few distorted elements. Thus, the adaptive meshing in FEM may

cause the degradation of accuracy and complex program that may lead to the billions of elements, which makes the system of equations solved with great difficulty.

- In mathematics, a degenerated geometry is the limiting case, in which a particular class of object changes its nature so as to belong to a simpler class. For example, a circle with the radius equal to zero unit degenerates into a point, and a line with the length equal to zero unit degenerates into a point. Apart from these simpler cases, there are several other complex cases of the degenerated geometry. If the degenerated geometry is not handled correctly during the generation of mesh, it may easily break the code of the FEM solver. Therefore, a lot of geometry degeneracy cases need to be handled while generating the good quality mesh for the complex geometry. This makes the implementation program of the FEM highly complex and slow during the execution.

Due to these limitations of the FEM, it is necessary to have a method that is able to solve complex problems without generating any mesh. The method should also correctly handle the governing differential equations with the appropriate boundary conditions.

1.2 Fundamentals of meshless methods

Various interpolation functions can be used in the meshless methods, which approximate the values of the field variables at the field nodes. Following are the important properties of a shape function when it is used in the meshless methods.

1. It is required to satisfy the partition of unity condition as

$$\sum_{i=1}^n N_i(x) = 1 \quad (1.1)$$

where $N_i(x)$ is the value of the shape function at the i^{th} field node.

2. It is required to satisfy the reproduction of linear field variable as

$$\sum_{i=1}^n N_i(x) x_i = x \quad (1.2)$$

This is the essential condition to pass a patch test in the FEM, and a preferable condition in the meshless methods.

3. It should possess the Kronecker Delta function property, which is also a preferable condition as

$$N_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \text{ such that } i, j = 1, 2, \dots, n \quad (1.3)$$

If the shape functions satisfy the delta function property, it is easy to impose the essential boundary conditions (BC). However, even if it does not, the essential BC can still be imposed with some special arrangements.

1.3 Common steps in a meshless method

The flowchart given by Fig. 1.1 shows a general sequence of the working procedure of meshless methods, and it is explained below step by step.

Fig. 1.1. General flowchart of a meshless method.

1.3.1 Geometry creation

The geometrical model of the structure is created first. In order to obtain the coordinates in the computational domain, the randomly scattered or uniformly distributed field nodes are created in the domain and boundaries. The values of the field variables are computed at these field nodes. The density of the nodes depends on the accuracy requirement and the distribution of the field variables.

1.3.2 Approximation of the field variable

A local domain is defined around each field node, where the surrounding nodes falling in this local domain are used for an interpolation of the field variable at the interested node, using a suitable interpolation function as

$$f^h(x_I, y_I) = \sum_{i=1}^n N_i(x_I, y_I) u_i \quad (1.4)$$

where the approximate function value $f^h(x_I, y_I)$ at the I^{th} field node is interpolated by the surrounding field nodes $i=1, 2, \dots, n$, and $N_i(x_I, y_I)$ and u_i are the values of shape function and nodal parameter at the i^{th} field node, respectively. The local interpolation domain may be created in any shape, such as a circle or rectangle, as shown in Fig. 1.2.

Fig. 1.2. Local domain for interpolation.

1.3.3 Discretization of the governing differential equation

In general, there are two approaches to handle the governing differential equation, namely the strong- and weak-forms.

1.3.3.1 Strong-form approach

The derivative terms from the governing differential equations are discretized directly by this approach, and the system of equations is obtained in terms of the values of approximate function at the field nodes. It is often difficult to impose the essential BC exactly by the strong-form approach, as the governing equation is satisfied at all the nodes in the internal domain, and the boundary condition is satisfied at every boundary node. This makes the strong-form approach very rigid, and sometimes is not workable during the execution. But, it is very easy to implement and may give accurate results, if proper system equations are formed.

1.3.3.2 Weak-form approach

The governing differential equation is converted to a corresponding weak form, and then solved by the numerical integration technique by this approach. The governing equations and boundary conditions are satisfied averagely over a domain instead of at the individual node. This makes the weak-form approach stable, but it is time consuming to implement, as the numerical integration is involved. In general, the numerical integration is performed by the Gauss quadrature with the Gauss points, and then the results are interpolated at the field nodes. This makes a further approximation of the values of function at the field nodes.

Whether to use a strong- or weak-form approach depends on how the meshless method is formulated. In general, if the problem is static, a system of algebraic equations is obtained after discretizing the governing equations. If the problem is transient, the temporal discretization (explicit or implicit) is required by a suitable numerical method when the meshless method is employed.

1.3.4 Assembly of the system of equations

A linear algebraic equation, which is local in nature, is obtained at every field node by the meshless method based on the strong- or weak-form approach. All the local equations are assembled together to form the global stiffness matrix and force vector ($K U = F$). At this step, the essential boundary conditions are imposed if the strong-form based meshless method is employed. If the interpolation function used in the meshless method does not possess the delta function property, the imposition of essential boundary conditions is difficult sometimes. However, various techniques of imposing the essential boundary conditions are discussed in the work of Krongauz and Belytschko [1].

1.3.5 Solving the assembled system of equations

Final system of equations can be solved by either the direct or iterative methods.

1.3.5.1 Direct method

There are several direct methods to solve the linear system of equations, such as the Gauss elimination, LU decomposition, LDL^T decomposition, and Cholesky factorization.

1.3.5.2 Iterative method

There are also several iterative methods available, such as the Jacobi, Gauss-Seidel, Conjugate Gradient, and krylov subspace methods. These methods involve repetitively solving the system of equations till the desired accuracy is achieved. In other words, the computational error in the solution is minimized by checking the residual value at the end of every iteration.

For the static problems of displacement type, the displacements at every field nodes are obtained, and the strain and stress are computed by different relations. For the free vibration or buckling problems, eigenvalues and eigenvectors are computed by the solver of eigenvalues equation, such as Jacobi, QR, subspace iteration or Lanczos method. For dynamic problems, the modal superposition or direct integration method can be used by the implicit or explicit approaches, where the time histories of the displacement, velocity, and acceleration are obtained.

1.4 Motivation of the present research work

- Although the differential quadrature (DQ) is an effective technique for derivative discretization, it always requires a regular and uniform domain, and all nodes are always required to lie uniformly along a straight line only. Therefore, the main motivation behind the present work is to extend the applicability of the DQ method over an irregular domain with the field nodes distributed either in the uniform or random manner. This novel meshless technique is called the Random Differential Quadrature (RDQ) method. In fact, the word *random* is used to indicate the application of DQ method to the field nodes distributed randomly.
- Another limitation of the DQ method is that it becomes unstable when the large numbers of nodes are used in a derivative discretization. This limitation is overcome in the proposed RDQ method by applying the DQ technique over a local domain of nodes.
- It is relatively easy for the strong form methods to perform the multiphysics analysis, such that the proposed RDQ method should also work well for the multiphysics analysis, such as the 2-D simulation of pH-sensitive hydrogels.
- The RDQ method has less dependence on the various parameters and variables, such that it is easy to implement.
- As the RDQ method is based on the strong form approach, it is interesting and challenging to solve the problems with the locally high gradients.

1.5 Objectives of the present research work

- To develop a novel meshless method, named the RDQ method that can be applied to irregular as well as regular domains discretized by either the random or uniform field nodes.
- In the RDQ method, the general differential quadrature (GDQ) is used to discretize the governing differential equation. It is applied over a local domain, so that it is called as the local differential quadrature (LDQ). One of the main problems with the GDQ is that the computational accuracy is deteriorated with the increase in the number of nodes. This can be avoided if the GDQ is applied over the local domain.

- In the RDQ method, the restriction of applying the GDQ technique over the uniformly distributed nodes is eliminated by the interpolation function of the fixed reproducing kernel particle method (fixed RKPM), and then applying the GDQ over the irregular domains as well.
- The RDQ method uses two types of nodes, namely the field nodes distributed in the uniform or random manner, and the virtual nodes distributed in the uniform or Chebyshev-Gauss-Lobatto pattern. The governing equations are discretized at the virtual nodes, and the distributions of the field variable are captured by the field nodes. In order to locally apply the GDQ method, a local domain is created around each virtual node, in which the virtual nodes falling are considered for the discretization of derivative terms from the governing differential equation at the concerned virtual node. Another local domain is created around each virtual node, in which the field nodes falling are used for an approximation of the function at the concerned virtual node by the fixed RKPM interpolation function. In this way, even if the field nodes are scattered either in the random or uniform manner and the computational domain is either the irregular or regular, the RDQ method can still be employed.
- Several 1-D, 2-D and elasticity problems will be solved for the convergence studies of the RDQ method.
- The consistency, stability, and adaptive analyses of the RDQ method will be conducted in details.
- The RDQ method will be employed for the 2-D multiphysics simulation of pH-sensitive hydrogels with the BioMEMS applications.

In summary, different drawbacks of the classical FEM are discussed in this subsection, and it is seen that how they can be overcome by the meshless methods. The fundamentals and common steps involved in the meshless method are studied. After that, the motivation and objectives of the present research work are stated, and it is understood that the proposed RDQ method can handle the field nodes distributed randomly very well over a non-uniform domain. In the next subsection, the literature survey about various meshless methods is performed.

1.6 Literature survey

Although meshless methods have been in a research area for the past 40 years, the extensive studies are carried out continuously in the last decade when people realized their potential to solve the large deformation and moving boundary problems. They are broadly categorized into two types of groups based on how the governing differential equation is solved, namely the strong- (collocation based) and weak-forms (Galerkin approach based). Some of the earlier developed methods, based on the weak-form approach, include the smooth particle hydrodynamics (SPH) [2-3], the diffuse element method (DEM) [4], the element-free Galerkin method (EFGM) [5-6], the natural element method (NEM) [7], the reproducing kernel particle method (RKPM) [8-10], the partition of unity based FEM (PUFEM) [11], the meshless local Petrov-Galerkin approach (MLPG) [12], the local boundary integral equation (LBIE) method [13-14], the point interpolation method (PIM) [15], and the local Kriging method [16]. Most of these methods use the least square (LS) approximation or the reproducing kernel particle interpolation functions for the approximation of function, and they are combined with the Galerkin or variational weak form of the governing differential equation. Earlier developed methods, based on the strong-form approach, include the finite point method (FPM) [17-18], Hermite cloud method [19], and the gradient smoothing method (GSM) [20]. All these methods were successfully applied to solve different types of engineering problems [16, 18, 21-28]. In general, the meshless methods based on the weak form are considered numerically more stable than those based on the strong form. However, the strong form methods can capture the local high gradients well, and they are easy to implement and cheap in computation.

It is observed that the originally proposed SPH method [2] is unable to reproduce the higher-order terms well, and to satisfy the consistency condition when solving the problems with finite boundaries. Hence, it was enhanced over the last decade to improve its consistency and the stability aspects [8-10, 29]. Liu *et al.* [8-10] modified the SPH window function by introducing a correction function term and called the RKPM, where the SPH window function is called the modified window function. The RKPM is more widely used now than the original SPH method because of the ability of higher-order reproducibility. Wong and Shie [30] performed the large deformation analysis by the

SPH-based Galerkin method with the moving least square (MLS) approximation. Wu *et al.* [31] proposed a differential RKPM, in which the separate sets of the differential reproducing condition were developed to compute the derivative shape functions, instead of conventionally computing them by directly taking the derivative of RKPM approximation. Li *et al.* [32] used the RKPM method to perform the ductile fracture simulations, and achieved very good agreement with the FEM and existing experimental data.

Following the idea of an integral quadrature, Bellman *et al.* [33] proposed the DQ method, in which the derivative at any grid point is approximated by the weighted sum of function values in whole domain provided that all the grid points are always collinear. The important task in the DQ method is to determine the weighting coefficients. Bellman *et al.* [33] suggested two approaches to compute the weighting coefficients. The first is the use of polynomial function as a test function, and the second uses the test function, in which the coordinates of the grid points are chosen as the roots of shifted Legendre polynomials. However, these approaches have a problem when the order of the system of algebraic equations is very high, making the resulting matrix highly ill-conditioned. In order to overcome this problem, Quan and Chang [34-35] proposed another approach, in which the weighting coefficients are computed by Lagrange interpolation polynomials. Shu *et al.* [36] proposed a general approach, in which both Bellman [33] and Quan and Chang [34-35] approaches are combined. Shu [37] proved by the analysis of linear vector space that the polynomials used in Bellman [33] and Quan and Chang [34-35] approaches are nothing but the different sets of the base polynomial vectors of the function approximation. If one of the base vectors satisfies the approximate function equation, the other base vectors do so. This means that all the approaches will lead to the same values of the weighting coefficients, and in turn similar function approximations. Shan *et al.* [38] coupled the local multiquadric-based radial basis functions (RBF) with the DQ method to solve the fluid flow problems with 3-D curved boundary. Liew *et al.* [39] applied the DQ method to model the elastic bonding in the 3-D composite laminates. Liew *et al.* [40] also applied the MLS based DQ (MLSDQ) to solve the moderately thick plate for the shear deformation, and to solve the 4th-order bending differential equation of

thin plate over the irregular boundaries [41]. More work about the DQ method can be found [37, 42-43].

The computation of weighting coefficients by the Shu's general approach [36-37] is discussed in the next chapter. Out of all the meshless methods, several well established methods are selected, and the corresponding formulations and working principles are discussed in details.

1.6.1 Smooth particle hydrodynamics

The SPH method, developed by Lucy [2] and Gingold and Monaghan [3], is considered as an approximation for an integral determined as per the Monte Carlo procedure [3]. As well known, the Monte Carlo method can give the reasonable solution of multiple integrals with fewer field points than the finite difference method (FDM), such that it is feasible to reduce the computational work by employing the statistical smoothening methods like the Monte Carlo method.

Lucy [2] formulated and tested the SPH method for 2-D and 3-D gas dynamical problems of astronomical interest, such as the fission problem for the optically thick protostars. Let us consider the function approximated by the integral approximation as

$$\eta(r) = \int_V w(r-r') \xi(r') \rho(r') dV' \quad (1.5)$$

where $\rho \geq 0$, and $w(r-r')$ is a window function. As per the standard Monte Carlo theory, if a set of J points are randomly distributed in the volume V , and thus the probability of a point found in the volume element dV' at the distance r' is proportional to $\rho(r') dV'$, one can have

$$\tilde{\eta}(r) = \frac{1}{J} \sum_j w(r-r_j) \xi(r_j) \quad (1.6)$$

Here Eq. (1.6) converges to $\eta(r)$ as $J \rightarrow \infty$. If we assume

$$\int_V w(r-r') dV' = 1, \text{ and } w = 0 \text{ for } |r-r_j| > \sigma \quad (1.7)$$

$\eta(r) \rightarrow \xi(r) \rho(r)$ as $\sigma \rightarrow 0$. This is followed by

$$\tilde{\eta}(r) \rightarrow \xi(r) \rho(r) \text{ as } J \rightarrow \infty \text{ and } \sigma \rightarrow 0 \quad (1.8)$$

In the work of Lucy [2], the terms of spatial derivative from the fission equations are represented by the SPH method, as given in Eq. (1.5). The problem is thus solved by applying the Monte Carlo theory to the discrete representation to obtain the continuous representation. If the window function w is approximated as a continuous function in the space, the mass density $\rho(r)$ is obtained from Eqns. (1.6) and (1.8) by setting $\xi(r) = 1$ as

$$\rho(r) = \frac{1}{J} \sum_j w(r - r_j) \quad (1.9)$$

Similarly, a function approximating the entropy density $s = \rho S$ is obtained by setting $\xi(r)$ equal to the specific density as $\xi(r) = S(r)$. Then

$$s(r) = \frac{1}{J} \sum_j S_j w(r - r_j) \quad (1.10)$$

Eqns. (1.9) and (1.10) will converge as $J \rightarrow \infty$ and $\sigma \rightarrow 0$. The window function w is chosen as a function of $|r - r_j|$ with the continuous 2nd-order derivative. For example, a function satisfying these conditions is given as

$$\begin{aligned} w &= \frac{105}{16 \pi \sigma^3} (1 + 3z) (1 - z)^3 \text{ for } z \leq 1 \\ w &= 0 \text{ for } z > 1 \end{aligned} \quad (1.11)$$

where $z = (r / \sigma)$. Here Eq. (1.11) is a bell shaped function. The window function is required to have several properties, such as the normalization property $\int_{\Omega} w(x - x', h) dx' = 1$, namely the summation of window function over a local domain should be equal to 1; $\lim_{h \rightarrow 0} w(x - x', h) = \delta(x - x')$, namely the window function should approach zero when the smoothing length h approaches the zero. The window function should be compact, namely $w(x - x', h) = 0$ when $|x - x'| > k h$, where k is a constant related to the smoothing function, which defines the effective area of smoothing function. This effective area is nothing but the local support domain of point x .

The SPH method became the basic method for the several particle-based methods developed later.

1.6.2 Diffuse element method

The classical FEM has two main drawbacks. The first is that the derivatives are discontinuous across the element boundaries, while the approximated solution is continuous within the element. The second is that the mesh generation with sufficiently good quality is a difficult task, especially for the complex 3-D domains. Nayroles *et al.* [4] proposed the DEM in order to overcome these problems, in which the local polynomial-based interpolation in the FEM is replaced by a local weighted least squares (WLS) fitting that is valid only over the region around the concerned node $\bar{x}$.

The FEM uses the piecewise approximation of the unknown functions over the element e as

$$u^e(x) = \sum_{j=1}^m p_j(x) a_j^e \quad (1.12)$$

where the vector p consists of m independent monomials, and a^e is a vector of m parameters that are constant over the element e . If the element e totally has the n^e nodes, $u^e(x)$ takes the values of u_i , namely

$$\{u_i\} = [P_n] \{a^e\} \quad (1.13)$$

The matrix $[P_n]$ is invertible for $n^e = m$, leading to the standard expression of shape function as

$$\{u_i\} = \langle p_j(x) \rangle [P_n]^{-1} \{a^e\} = \langle N_i(x) \rangle \{u_i\} \quad (1.14)$$

Eq. (1.14) can also be obtained by the LS approach, by minimizing the following expression with respect to a^e as

$$J = \sum_{i=1}^{n^e} [u_i - u^e(x_i)]^2 = \sum_{i=1}^{n^e} [u_i - \langle p_j(x_i) \rangle \{a^e\}]^2 \quad (1.15)$$

The minimization of Eq. (1.15) gives an equation similar to that given in Eq. (1.14), if $n^e = m$. Eq. (1.15) is used in the DEM with an addition of a weighting function (WF) as

$$J = \sum_{i=1}^n w_i^e \left[(u_i - u^e(x_i)) \right]^2 \quad (1.16)$$

where n is the number of total nodes in the local interpolation of the concerned field node $u^e(x)$. The total n^e nodes in an element are separated from the total m monomial terms in an interpolation due to the use of DEM in the FEM. The local nature of the function approximation is preserved as the WF w_i^e vanishes beyond certain distance. The stationarity of J with respect to a^e leads to

$$u^h(x) = \sum_{i=1}^n N_i(x) u_i, \quad \text{where } N_i(x) = \sum_{j=1}^m P_{xj}(x) \left[A^{-1}(x) B(x) \right]_{ji} \quad (1.17)$$

where

$$[A(x)] = \sum_{i=1}^n w_i^e(x) P_{ij} \sum_{k=1}^m P_{ik}, \quad [B(x)] = \sum_{i=1}^n w_i^e(x) P_{ij} \quad (1.18)$$

The necessary condition to get the nonsingular $A^{-1}(x)$ matrix is the existence of m nodes at least in the local interpolation domain of $u^e(x)$. The approximate derivative of the function is computed by differentiating $p(x)$ with respect to x , by considering a^e as a constant, and given as

$$\left(\frac{\partial u}{\partial x} \right)_x = \left\langle \left(\frac{\partial p}{\partial x} \right)_x \right\rangle \{a^x\} \quad (1.19)$$

$$\therefore \left(\frac{\partial u}{\partial x} \right)_x = \left\langle \left(\frac{\partial N}{\partial x} \right)_x \right\rangle \{u_i\}, \quad \text{where } \left\langle \left(\frac{\partial N}{\partial x} \right)_x \right\rangle = \left\langle \frac{\partial p}{\partial x} \right\rangle [A^x]^{-1} [B^x] \quad (1.20)$$

If w_i^e is continuous with respect to x , the shape functions are also continuous with respect to x , and the derivatives of the function up to the m^{th} -order exist. It is noted that, if the WF is constant over the domain, the diffuse approximation is similar to LS fitting of a polynomial function. If the WF is constant only over a fixed sub-domain, the diffuse approximation may become identical with the finite element approximation. The FEM

shape functions given in Eq. (1.14) are replaced by the diffuse approximation shape functions given in Eq. (1.17), while implementing the diffuse approximation in the FEM. The elements are not used for an approximation of the function, but they may still be used for the numerical integration over the domain.

By solving several mechanics problems, it is observed that the gradients of the function computed by the DEM are better than those of the FEM [4].

1.6.3 Element-free Galerkin method

This method was developed by Belytschko *et al.* [5-6, 44-45], by the MLS approximation to construct the trial and test functions for the variational principle in weak form. The several key points of this method are highlighted as

- The EFGM does not exhibit the volumetric locking, even for the linear basis functions;
- the rates of convergence of the EFGM are significantly better than those of the FEM;
- the EFGM is quite effective for the problems of linear elastic fracture.

The notable differences between the EFGM and MLPG method are given as,

- The essential boundary condition is imposed by the penalty method in the MLPG technique, while it is achieved by Lagrange multipliers in the EFGM;
- the variational weak form is used in the EFGM where the trial and test functions are approximated by the MLS approximation, while the Petrov-Galerkin weak form is used in the MLPG method where the trial functions are approximated by the MLS approximation, and the test functions are approximated by the Gaussian weight function that is used in the MLS approximation.

As the weight function $w_i(x)$ plays an important role in the computational accuracy of the final solution obtained by the EFGM, it is chosen in such a way that it should give higher values for the nearer nodes, and progressively lower values away from the concerned node. Therefore, the weight functions in the form of $w_i(x) = w_i(d^{2k})$ are studied, where $d = |x - x_i|$ is the distance between the two nodes x and x_i . The constant k is determined such that the first m derivatives of $w_i(d^{2k})$ with respect to d are continuous. In order to study the effects of different weighting functions in the EFGM,

two weighting functions are considered, namely the exponential and conical functions. The mechanical equilibrium equation over the domain Ω is given as

$$\nabla \cdot \sigma + B = 0 \quad (u = \bar{u} \text{ on } \Gamma_u \text{ and } \sigma \cdot n = \bar{t} \text{ on } \Gamma_t) \quad (1.21)$$

The variational weak form of Eq. (1.21) is given as

$$\int_{\Omega} \delta v (\nabla \cdot v) : \sigma \, d\Omega - \int_{\Omega} \delta v \cdot B \, d\Omega = \int_{\Gamma_t} \delta v \cdot \bar{t} \, d\Gamma + \int_{\Gamma_u} \delta \lambda (u - \bar{u}) \, d\Gamma + \int_{\Gamma_u} \delta v \cdot \lambda \, d\Gamma \quad (1.22)$$

where $u \in H^1$ and $\lambda \in H^0$ are the trial and Lagrange multiplier functions, respectively, and δv and $\delta \lambda$ are the test functions for u and λ , respectively. The H^1 and H^0 are the Sobolev spaces of the degree one and zero, respectively. The discrete form of Eq. (1.22) is obtained by substituting the approximate solution u and the test function δv by Eq. (1.17). The λ is replaced as

$$\lambda(x) = N_I(s) \lambda_I \text{ and } \delta \lambda(x) = N_I(s) \delta \lambda_I \quad (1.23)$$

where $N_I(s)$ and s are Lagrange interpolant and the arc length along the boundary, respectively. In order to solve Eq. (1.22), a cell structure is used and it is independent of the field nodes. Finally, the discrete form of Eq. (1.22) is solved by $n_Q \times n_Q$ Gauss quadrature using the rule $n_Q = \sqrt{m} + 2$, where m is the number of total nodes in a cell.

Several test problems of mechanics, such as the bending, fracture, crack propagation, an infinite plate with a central hole, and steady-state heat conduction, are solved successfully by the EFGM [5-6, 46], and very high rates of the convergence are achieved by the large number of quadrature points in the computational domain. It is shown that, if the EFGM is considered as a continuous form, it is identical to the free Lagrange method if the correct weight function is defined [5].

1.6.4 Natural element method

The NEM proposed by Braun and Sambridge [7] is a general Galerkin form of the weighted residual method, in which the natural-neighbour coordinates are used as the geometrical trial functions. It is required to understand several concepts before proceeding further.

Any set of the arbitrarily distributed nodes in a plane has a unique set of natural neighbours' around each node. Any two nodes are said to be natural neighbours if their voronoi cells [7, 27-28] have a common boundary. The voronoi cell about each node is a part of the plane which is closest to that node. For a given nodal distribution, the natural neighbours are uniquely determined.

Fig. 1.3. Computation of natural-neighbour coordinates in the NEM.

Delaunay triangulation is obtained by connecting all pairs of natural neighbours together. Let us consider a test point x in the Voronoi diagram of the set N , as shown in Fig. 1.3. If the point x is considered as a node along with the set of N nodes, the natural neighbours of x are those nodes, which form an edge of a triangle with the point x in the new tessellation (triangulation). The natural neighbours of x are found to be the 1 to 5 nodes, as shown in Fig. 1.3. The perpendicular bisectors from the point x to its natural neighbours are drawn, and the Voronoi cell T_x , namely the closed polygon of a, b, c, d, e and a , is obtained.

In the Sibson interpolation, the natural-neighbour coordinates are used as the interpolating functions in the natural neighbour interpolation [27-28], and the area is

taken as a measure of the natural-neighbour coordinates for the 2-D domain. As such, the natural neighbour coordinate $\phi_I(x)$ of the point x with respect to its natural neighbour I , where $\forall I : I \in [1 \text{ to } 5]$, is defined as the ratio of the overlap area $A_I(x)$ of the Voronoi cell T_I to the total area $A(x)$ of the Voronoi cell T_x around the point x as given by

$$\phi_I(x) = \frac{A_I(x)}{A(x)} \quad (1.24)$$

where $A(x) = \sum_{I=1}^5 A_I(x)$. The areas A_1 to A_5 , as shown in Fig. 1.3, are called the second-order cells, and the area covered by the closed polygon, a, b, c, d, e and a , is called the first-order Voronoi cell. The ratio of the areas $\phi_I(x)$ is taken as the shape function $N_I(x)$ in the NEM. For example, $N_3(x)$ is computed as

$$N_3(x) = \frac{A_3(x)}{A(x)} \quad (1.25)$$

where $A_3(x)$ is given by the polygon a, f, g, h, e and a . It is noted that, when the point x coincides with the node x_I , the shape function $N_I(x) = 1$, and the other shape functions are equal to zero. As a result, the shape functions constructed by the natural-neighbour coordinates possess the property of Dirac delta function $N_I(x_I) = \delta_{II}$, and the property of partition of unity $\sum_{I=1}^5 N_I(x) = 1$.

The shape functions in the NEM are computed by the natural-neighbour coordinates as explained earlier, and the terms of approximate function from the weak form of the governing equation, as given in Eq. (1.22), are substituted by the trial function $f^h(x) = \sum_i N_i(x - x_i) f_i$. The important difference between the NEM and FEM is that the function in the FEM is approximated within the element and it is discontinuous over the element boundaries, while the function in the NEM is approximated by the natural-neighbour coordinates that are continuous and differentiable everywhere, except at the interested node itself. In order to compute the spatial derivatives in the NEM, a generalized and versatile approach was developed [7] that works with the n dimensions.

There are several aspects to be considered, if the NEM is used. In the classical FEM, the weak-form equation in the integral form is solved by the Gauss quadrature, and sometimes it gives accurate results, as the trial function is approximated by the polynomials. But, the trial functions are not simple polynomial expressions in the NEM, such that the trial and error approach is required when the integral equation is solved by the Gauss quadrature. Furthermore, the NEM method involves several concepts from the computational geometry, and thus the users are required to be knowledgeable enough about them before the NEM is used.

1.6.5 Reproducing kernel particle method

The RKPM method is a discretized form of the reproducing kernel method (RKM), which was developed by Liu *et al.* [8-10] using the wavelet theory and the convolution theorem. They studied the effects of discretization in Fourier analysis. The role of interpolation functions is considered as a low pass filter that reduces the amplitude of signals with frequencies higher than the cutoff frequency. A dilation parameter, as found in the wavelet theory, is introduced in the interpolation function to extend the convolution theory to the broader field.

The convolution of two functions in the function domain (x) is equivalent to the multiplication of two functions in the Fourier transform domain (ξ) . The conventional convolution is defined as

$$u^R(x) = u(x) * \phi(x) = \int_{-\infty}^{+\infty} u(y) \phi(x-y) dy \quad (1.26)$$

where $*$ is the symbol of convolution. The approximate function $u^R(x)$ in the domain Ω is given by the RKM as

$$u^R(x) = \int_{\Omega} u(\tilde{x}) \bar{\phi}_a(x-\tilde{x}) d\tilde{x} \quad (1.27)$$

The given function $u(x)$ is reproduced up to any desired order by choosing a suitable window function $\bar{\phi}_a$. The expansion of $u(x)$ about x by Taylor series is given as

$$u(\tilde{x}) = u(x) - (x - \bar{x}) u'(x) + \frac{1}{2!} (x - \bar{x})^2 u''(x) + \dots + \frac{(-1)^n}{n!} (x - \bar{x})^n u^{(n)}(x) \quad (1.28)$$

Substituting Eq. (1.28) into (1.27) gives

$$\begin{aligned} u^R(x) = & u(x) \int_{\Omega} \bar{\phi}_a(x - \tilde{x}) d\tilde{x} - u'(x) \int_{\Omega} (x - \bar{x}) \bar{\phi}_a(x - \tilde{x}) d\tilde{x} + \frac{u''(x)}{2!} \int_{\Omega} (x - \bar{x})^2 \bar{\phi}_a(x - \tilde{x}) d\tilde{x} + \\ & \dots + \frac{(-1)^n u^{(n)}(x)}{n!} \int_{\Omega} (x - \bar{x})^n \bar{\phi}_a(x - \tilde{x}) d\tilde{x} \end{aligned} \quad (1.29)$$

The integral terms from Eq. (1.29) are used to define the moment as

$$\bar{m}_k(a, x) = \int_{\Omega} (x - \bar{x})^k \bar{\phi}_a(x - \tilde{x}) d\tilde{x}, \quad k = 0, 1, \dots, n \quad (1.30)$$

Rearranging Eq. (1.29) by (1.30) leads to

$$u^R(x) = u(x) \bar{m}_0(a, x) - u'(x) \bar{m}_1(a, x) + \frac{u''(x)}{2!} \bar{m}_2(a, x) + \dots + \frac{(-1)^n u^{(n)}(x)}{n!} \bar{m}_n(a, x) \quad (1.31)$$

In order to reproduce the function up to the n^{th} -order, the reproducing condition needs to be satisfied as

$$\bar{m}_k(a, x) = \delta_{k0}, \quad k = 0, 1, 2, \dots, n \quad (1.32)$$

The window function $\bar{\phi}_a(x - \tilde{x})$ is given as

$$\bar{\phi}_a(x - \tilde{x}) = P^T(x - \tilde{x}) b(a, x) \phi_a(x - \tilde{x}) \quad (1.33)$$

where

$$P(x - \tilde{x}) = \begin{bmatrix} 1 \\ (x - \tilde{x}) \\ \vdots \\ (x - \tilde{x})^n \end{bmatrix} \text{ and } b(a, x) = \begin{bmatrix} b_0(a, x) \\ b_1(a, x) \\ \vdots \\ b_n(a, x) \end{bmatrix} \quad (1.34)$$

The function $\bar{\phi}_a$ is an arbitrarily chosen window function, which does not necessarily satisfy the reproducing conditions. The product of vectors $P^T(x - \tilde{x})$ and $b(a, x)$ is called the correction function $C_a(x; x - \tilde{x})$ to the window function, and the unknown

coefficients $b(a, x)$ are computed by the reproducing condition. Substituting Eq. (1.33) into (1.30) gives

$$\bar{m}_k(a, x) = \int_{\Omega} (x - \bar{x})^k \left[\sum_{k=0}^n (x - \tilde{x})^k b_k(a, x) \phi_a(x - \tilde{x}) \right] d\tilde{x} \quad (1.35)$$

$$\therefore \bar{m}_k(a, x) = b_0(a, x) m_k(a, x) + b_1(a, x) m_{k+1}(a, x) + \dots + b_n(a, x) m_{k+n}(a, x) \quad (1.36)$$

where $m_k(a, x)$ is the k^{th} moment of the original window function $\phi_a(x - \tilde{x})$. Eq. (1.36) can be written in the matrix form as

$$\begin{bmatrix} \bar{m}_0(a, x) \\ \bar{m}_1(a, x) \\ \vdots \\ \bar{m}_n(a, x) \end{bmatrix} = \begin{bmatrix} m_0(a, x) & m_1(a, x) & \dots & m_n(a, x) \\ m_1(a, x) & m_2(a, x) & \dots & m_{n+1}(a, x) \\ \vdots & \dots & \ddots & \vdots \\ m_n(a, x) & m_{n+1}(a, x) & \dots & m_{2n}(a, x) \end{bmatrix} \begin{bmatrix} b_0(a, x) \\ b_1(a, x) \\ \vdots \\ b_n(a, x) \end{bmatrix} \quad (1.37)$$

$$\therefore \bar{\mathbf{m}}(a, x) = \mathbf{M}(a, x) \mathbf{b}(a, x) \quad (1.38)$$

where $\mathbf{M}$ is called the moment matrix and written as

$$\mathbf{M}(a, x) = \int_{\Omega} \mathbf{P}(x - \tilde{x}) \phi_a(x - \tilde{x}) \mathbf{P}^T(x - \tilde{x}) d\tilde{x} \quad (1.39)$$

Eq. (1.32) giving the reproducing condition is written in the vector form as

$$\begin{bmatrix} \bar{m}_0(a, x) \\ \bar{m}_1(a, x) \\ \vdots \\ \bar{m}_n(a, x) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{P}(0) \quad (1.40)$$

Combining Eq. (1.40) with Eq. (1.38) results in

$$\mathbf{M}(a, x) \mathbf{b}(a, x) = \mathbf{P}(0) \quad (1.41)$$

The unknown coefficients $b(a, x)$ are obtained as

$$\mathbf{b}(a, x) = \mathbf{M}^{-1}(a, x) \mathbf{P}(0) \quad (1.42)$$

Substituting Eq. (1.42) into Eq. (1.33) gives the window function $\bar{\phi}_a(x - \tilde{x})$, which leads to the evaluation of approximate function by the RKM, if it is further substituted in Eq. (1.27).

The discrete form of Eq. (1.27) is the RKPM function [10], and given as

$$u^R(x) = \sum_{I=1}^{NP} u(x_I) \bar{\phi}_a(x - x_I) \Delta v_I \quad (1.43)$$

where NP is the total particles in the discrete system, and Δv_I is the nodal volume assigned to each particle. The window function is normalized by the dilation parameter a to reduce the noise in the solution as

$$\bar{\phi}_a(x - x_I) = \frac{1}{a} \bar{\phi}\left(\frac{x - x_I}{a}\right) \quad (1.44)$$

The discrete moment term is given as

$$\tilde{m}_k(a, x) = \sum_{I=1}^{NP} (x - x_I)^k \bar{\phi}_a(x - x_I) \Delta v_I \quad (1.45)$$

Eq. (1.33) gives the originally chosen window function $\phi_a(x - \tilde{x})$. The RKPM shape function is given as

$$N_I(x) = C_a(x; x - x_I) \phi_a(x - x_I) \Delta v_I \quad (1.46)$$

and the approximate function is then given as

$$u^R(x) = \sum_{I=1}^{NP} N_I(x) u_I \quad (1.47)$$

Because of the addition of the correction function term in the RKPM, the field variables are almost accurately reproduced, if the required order of the field variable distribution is included in the vector of polynomial basis. As the RKPM is inherited from the SPH method, it has also inherited the SPH characteristics, such as the smoothening of a function value over a local domain by the approximation of integral function.

Aluru and Li [47] and Aluru [48] discussed several forms of the RKPM, such as the fixed, moving, and multiple fixed RKPM. The main difference between them is the way

kernel or window function is handled. The kernel is fixed at the interested node in case of the fixed RKPM, where the approximate function is given as

$$f^h(x, y) = \int_{\Omega} C(x, y, u, v) K(x_k - u, y_k - v) f(u, v) du dv \quad (1.48)$$

where $C(x, y, u, v)$ is unknown correction function, $f^h(x, y)$ is an approximation of the function $f(x, y)$ at a node (x, y) , and $K(x_k - u, y_k - v)$ is the kernel function fixed at the node (x_k, y_k) . The order of approximate function $f^h(x, y)$ is determined by the order of the monomial basis functions used in the correction function $C(x, y, u, v) = P^T(u, v) c(x, y)$, where $P^T(u, v) = \{b_1(u, v), b_2(u, v), \dots, b_m(u, v)\}$ is the column vector of the m^{th} -order monomial, and $c(x, y)$ is the m^{th} -order unknown row vector. The moment matrix in the fixed RKPM is constant for a local domain due to the fixed kernel [47].

The approximate function in the moving kernel method is given as

$$f^h(x, y) = \int_{\Omega} C(x, y, u, v) K(x - u, y - v) f(u, v) du dv \quad (1.49)$$

As seen from Eq. (1.49), the moment matrix in the moving kernel technique is the function of x and y , and no longer constant.

The approximate function by the multiple fixed kernel method is given as

$$f^h(x, y) = \int_{\Omega} C(x, y, s, t) K_{s,t}(s - x, t - y) f(s, t) ds dt \quad (1.50)$$

It is observed from Eq. (1.50) that the kernel is fixed at the node (s, t) , which is in the local interpolation domain of the central node (x, y) . As such, the kernel is fixed at several nodes, therefore it is called the multiple fixed KPM.

Aluru and Li [47] developed the finite cloud method by combining the fixed RKPM with the collocation method, and performed the convergence analysis. The approximate derivatives in the finite cloud method [47] were computed by the diffuse derivatives approach [4]. Aluru [48] also developed a point collocation method based on the RKPM, and performed the convergence analysis. He showed that the assignment of nodal

volumes in the particle based function approximation is not an issue with the point collocation method, however it is an issue in the Galerkin-approach based RKPM.

1.6.6 Partition of unity finite element method

The classical FEM often encounters some difficulties while solving the problems with the rough or local oscillatory behaviour. In order to overcome this, the partition of unity finite element method (PUFEM) was proposed by Melenk and Babuška [11] and Babuška and Melenk [49]. The prominent features of the PUFEM include

- ability to include *a priori* knowledge about the local behaviour of the field variable in the finite element space,
- ability to construct more suitable finite element spaces for the higher-order governing equations,
- generalization of the classical h , p and hp versions of the FEM.

In the classical FEM, the space of function approximation consists of piecewise polynomials that possess a good property of local approximation, and are conforming and continuous across the boundaries of element. As a result, a function with good property of local approximation may lead to the good quality of solution by the FEM. There are many functions other than the polynomials that have the property of local approximation. If the FEM space is constructed with these functions, the solutions of the differential equation are very well locally approximated. Therefore, the PUFEM offers the means to construct a conforming space out of any given system of local approximation space without sacrificing the approximation properties.

Let $\Omega \subset R^n$ be an open set, $\{\Omega_i\}$ be an open cover of Ω satisfying a pointwise overlap condition, and $\{\phi_i\}$ be a Lipschitz partition of unity subordinate to the cover $\{\Omega_i\}$ satisfying

$$\text{supp } \phi_i \subset \text{closure } \{\Omega_i\} \quad \forall i \quad (1.51)$$

$$\sum_i \phi_i = 1 \quad \text{on } \Omega \quad (1.52)$$

$$\|\phi_i\|_{L^\infty(R^n)} \leq C_\infty \quad (1.53)$$

$$\|\nabla \phi_i\|_{L^\infty(R^n)} \leq \frac{C_G}{\text{diam } \Omega_i} \quad (1.54)$$

where C_∞ and C_G are constants. The $\{\phi_i\}$ is called a (M, C_∞, C_G) partition of unity subordinate to the cover $\{\Omega_i\}$. The partition of unity $\{\phi_i\}$ is said to be of the m degree, such that $m \in N_0$ if $\{\phi_i\} \subset C^m(R^n)$. The covering sets $\{\Omega_i\}$ are called the patches.

The PUFEM method is nothing but the generalized h and p versions, if the spaces of local approximation are chosen as the spaces of polynomials. Several examples are discussed [11] for the construction and various choices of the local approximation spaces with better approximation properties than those of the polynomial of the degree p . However, a proper choice of the partition of unity $\{\phi_i\}$ is required to put the given sets of local approximation spaces together to get a conforming global space. A general choice of the partition of unity is given here. Let $\{\Omega_i\}$ be the collection of overlapping patches that cover Ω , and $\{\psi_i\}$ be the functions that are supported by the patches $\{\Omega_i\}$. Using the normalization, we have

$$\phi_i = \frac{\psi_i}{\sum_j \psi_j} \quad (1.55)$$

This Eq. (1.55) gives the partition of unity subordinate to the cover $\{\Omega_i\}$, in which only those j are considered that satisfy $\Omega_i \cap \Omega_j \neq \emptyset$. The function ϕ_i inherits the smoothness of ψ_i , such that this normalization technique gives one possible construction of the finite element spaces with a higher regularity.

1.6.7 Finite point method

The FPM was proposed by Oñate *et al.* [17-18], in which the WLS interpolation technique is used for an interpolation of the data, and the point collocation method is used to solve the governing differential equations. Several methods for the point data interpolation, such as the LS, WLS, MLS, and RKPM approximations were studied, and the corresponding shape functions were compared with the FEM [17]. It is observed that the methods of point data interpolation are sensitive to the number of points added in the

interpolation. The FPM is finally devised by the WLS interpolation with the Gaussian weighting function, and the point collocation method for the evaluation of governing equations.

In the point data interpolation methods, if Ω_k represents the interpolation domain for a function $u(x)$, and $j = 1, 2, \dots, n$ interpolation points used such that $x_j \in \Omega_k$, the unknown function $u(x)$ is given as

$$u(x) = \sum_{i=1}^m p_i(x) \alpha_i \quad (1.56)$$

where $\alpha_i = [\alpha_1, \alpha_2, \dots, \alpha_m]^T$ is a vector of unknown coefficients, and $p_i(x)$ contain the monomials or the basis functions. Eq. (1.56) represents the model equation for the approximation of function. The function $u(x)$ is sampled at the n points, such that $x_j \in \Omega_k$, then the approximate function is given as

$$u^h = \begin{Bmatrix} u_1^h \\ u_2^h \\ \vdots \\ u_n^h \end{Bmatrix} = \begin{Bmatrix} P_1^T \\ P_2^T \\ \vdots \\ P_n^T \end{Bmatrix} \alpha = C \alpha \quad (1.57)$$

For the FEM, $m = n$, C is thus a square matrix, and the unknown coefficients are obtained by

$$\alpha = C^{-1} u^h \quad (1.58)$$

For the LS method, $m \neq n$, C is thus not a square matrix, and then the function approximation can not fit all the values u^h . This problem can be solved by minimizing the square of difference between the approximate function and its model equation in Eq. (1.56) as

$$J = \sum_{j=1}^n [(u_j - p_j^T \alpha)]^2 \quad (1.59)$$

Minimization of Eq. (1.59) results in

$$\alpha = \bar{C}^{-1} u^h \text{ with } \bar{C}^{-1} = A^{-1} B \quad (1.60)$$

where

$$[A(x)] = \sum_{j=1}^n P(x_j) P^T(x_j), \text{ and } [B(x)] = [p(x_1), p(x_2), \dots, p(x_n)] \quad (1.61)$$

The LS method can be improved in a local region by weighting the squared distances in Eq. (1.59), as shown in Eq. (1.16), and it thus is called the WLS method. The weighting function w_i^e is chosen, such that it takes the value of unity in the vicinity of the interested node, and vanishes beyond the local region Ω_k . In the WLS method with $m \neq n$, the vector α of unknown coefficients should thus be determined by the minimization as given in Eqns. (1.17) and (1.18).

The weighting function w_i^e is translated over the domain in the MLS method, such that it takes the maximum value at the point k with the coordinates x_k , where the unknown function $u(x)$ is evaluated. The following function is thus minimized at the point k as

$$J = \sum_{j=1}^n \phi_k(x_j - x_k) \left[(u_j^h - p_j^T \alpha) \right]^2 \quad (1.62)$$

where ϕ_k changes the shape and span as per the location x_k , which is an arbitrary coordinate position. In the case of a constant grid spacing, ϕ_k becomes

$$\phi_k(x_j - x_k) = \phi(x_j - x) \quad (1.63)$$

It is seen from Eq. (1.62) that the selection of location x_k is difficult as it presents infinite number of possibilities. Therefore, it is convenient to specify the weighting function at the finite number of chosen points only, similar to WLS with the type of $\phi_i(x_j - x_i)$. A good computational procedure with this definition is described as

$$\phi_k(x_j - x_k) = \phi_j(x_k - x_j) \quad (1.64)$$

where x_k is an arbitrary location ($x_k = x$), and j is a fixed point in the domain. With this substitution, the functional to be minimized is rewritten from Eq. (1.62) to the following as

$$J = \sum_{j=1}^n \phi_j(x-x_j) \left[(u_j^h - p_j^T \alpha(x)) \right]^2 \quad (1.65)$$

This equation gives

$$\begin{aligned} [A(x)] &= \sum_{j=1}^n \phi_j(x-x_j) P(x_j) P^T(x_j), \\ [B(x)] &= [\phi_1(x-x_1) p(x_1), \phi_2(x-x_2) p(x_2), \dots, \phi_n(x-x_n) p(x_n)] \end{aligned} \quad (1.66)$$

It is noted in Eq. (1.65) that the unknown parameter α_i is not constant, and it varies with the position x .

In the FPM [17], the governing equations are discretized by the point collocation method, and the LS approximation is used by simply choosing $w_i = \delta_i$, namely the Dirac delta function, resulting in the governing and boundary equations as

$$[A(\hat{u})]_i - b_i = 0 \quad \text{in } \Omega \quad (1.67)$$

$$[B(\hat{u})]_i - t_i = 0 \quad \text{in } \Gamma_t, \text{ and } \hat{u}_i - u_p = 0 \quad \text{in } \Gamma_u \quad (1.68)$$

where Ω , Γ_t and Γ_u are the internal domain, and Neumann and Dirichlet boundaries, respectively. Eqns. (1.67) and (1.68) are simplified by substituting the shape functions as

$$K U^h = f \quad (1.69)$$

Solution of Eq. (1.69) gives the values of approximate solutions u_i .

1.6.8 Meshless local Petrov-Galerkin method

This method was proposed by Atluri and Zhu [12, 50-52], in which a local interpolation domain is created around each field node, and the function is approximated at each field node by the MLS technique with the local interpolation domain. The governing equation is converted to the weak form, but the trial and test functions are chosen from the different spaces, like in the Petrov-Galerkin method. The trial function u is approximated by the MLS technique, and the test function v is approximated by the weighting function, which is used in the MLS approximation of the trial function. The local symmetric weak form (LSWF) is developed by the Petrov-Galerkin approach, and

the final equations are solved by the Gauss quadrature method. The essential BC is imposed by the penalty method.

Fig. 1.4. Schematic of different domains in the MLPG method.

The essential BC is imposed by the penalty method. The formulation of MLS approximation is discussed first. Consider a computational domain, as shown in Fig. 1.4. In general, the global domain Ω is subdivided into several subdomains Ω_s with the local boundary $\partial\Omega_s = L_s \cup \Gamma_s$, where L_s and Γ_s are the parts of the boundary of subdomain Ω_s . Here L_s does not coincide with the global boundary Γ , while Γ_s coincides with Γ .

Let $p_j(x)$ be a vector of monomials in the n -dimensional space, such that $\forall n \in [1, 2, 3]$. Therefore, the monomials for 1-D space are given as

$$P^T(x) = [1, x], \quad m = 2 \text{ for a linear basis} \quad (1.70a)$$

$$P^T(x) = [1, x, x^2], \quad m = 3 \text{ for a quadratic basis} \quad (1.70b)$$

The monomials for 2-D space are given as

$$P^T(x) = [1, x, y, x^2, xy, y^2], \quad m = 6 \text{ for a quadratic basis} \quad (1.70c)$$

Consider a subdomain Ω_x around the node at x with the n field nodes in the interpolation domain, such that the MLS approximation $u^h(x)$ is given as

$$u^h(x) = \sum_{i=1}^m P_{xj}(x) a_j(x) \quad (1.71)$$

where the unknown coefficients $a_j(x)$ are determined by minimizing the weighted discrete L_2 norm as

$$E = \sum_{i=1}^n w(x-x_i) \left[\sum_{j=1}^m P_{ij}(x) a_j(x) - u_i \right]^2 \quad (1.72)$$

where $w(x-x_i) = W_i(x)$ is the weighting function. Eq. (1.72) is minimized with respect to a_j by

$$\frac{\partial E}{\partial a_j} = 2 \sum_{i=1}^n w_i(x) \left[\sum_{j=1}^m P_{ij}(x) a_j(x) - u_i \right] P_{ij} = 0 \quad (1.73)$$

After Eq. (1.73) is simplified, we have

$$A(x) a(x) = B(x) U \quad (1.74)$$

where

$$[A(x)] = \sum_{i=1}^n w_i(x) P_{ij} \sum_{k=1}^m P_{ik}, \quad [B(x)] = \sum_{i=1}^n w_i(x) P_{ij} \quad (1.75)$$

Inverting $A(x)$ from Eq. (1.74) results in

$$a(x) = A^{-1}(x) B(x) U \quad (1.76)$$

Substituting Eq. (1.76) into Eq. (1.71) and then simplifying the result, we have

$$u^h(x) = \sum_{i=1}^n N_i(x) u_i, \quad \text{where } N_i(x) = \sum_{j=1}^m P_{xj}(x) [A^{-1}(x) B(x)]_{ji} \quad (1.77)$$

Eq. (1.77) is the final form for the MLS approximation. It should be noted from Eq. (1.77) that $N_i(x_j) \neq 1$, namely the MLS approximation does not possess the property of

delta function. If A^{-1} exists, the rank of matrix $A(x)$ should be at least m . The Gaussian function is used as the weighting function, as given below

$$w_i(x) = \begin{cases} \frac{e^{[-(d_i/c_i)^{2k}] - e^{[-(r_i/c_i)^{2k}]}}{1 - e^{[-(r_i/c_i)^{2k}]}} , & 0 \leq d_i \leq r_i \\ 0, & d_i \geq r_i \end{cases} \quad (1.78)$$

where $d_i = |x - x_i|$ is the distance between the node located at x and its interpolation node at x_i , c_i is the constant controlling the shape of weight function $w_i(x)$, and r_i is the size of support for the weight function $w_i(x)$ that determines the support of node at x_i .

After the MLS approximations are developed at the field nodes, the LSWF is developed over a local subdomain Ω_s . The shape of the local subdomain is taken as a sphere in 3-D domain, and a circle in 2-D domain. The general Poisson equation over the internal domain Ω with boundary $\Gamma = \Gamma_u \cup \Gamma_q$ is given as

$$\nabla^2 f(x) = \bar{g}(x) \quad x \in \Omega, \text{ such that } f = \bar{f} \text{ on } \Gamma_u \text{ and } f \cdot n = \bar{q} \text{ on } \Gamma_q \quad (1.79)$$

The generalized weak form of Eq. (1.79) is given as

$$\int_{\Omega_s} (\nabla^2 f - \bar{g})v \, d\Omega - \alpha \int_{\Gamma_{su}} (f - \bar{f})v \, d\Gamma = 0 \quad (1.80)$$

where α is a penalty factor, f and v are the trial and test functions, respectively, and Γ_{su} is a part of the boundary $\partial\Omega_s$ of the subdomain Ω_s , over which the Dirichlet boundary condition is specified. It is noted from Fig. 1.4 that, if the subdomain Ω_s is completely inside the global domain Ω , the Γ_{su} term vanishes automatically. As the MLS approximation does not possess the delta function property, the Dirichlet boundary conditions are imposed by the penalty method with $\alpha \gg 1$. Eq. (1.80) is simplified according to Fig. 1.4, by applying the integration by parts and the divergence theorem as

$$\int_{\Omega_s} (f_{,i} v_{,i} + \bar{g}v) \, d\Omega + \alpha \int_{\Gamma_{su}} (f - \bar{f})v \, d\Gamma = \int_{L_s} q v \, d\Gamma + \int_{\Gamma_{su}} q v \, d\Gamma + \int_{\Gamma_{sq}} \bar{q} v \, d\Gamma \quad (1.81)$$

where Γ_{sq} is a part of $\partial\Omega_s$, in which the Neumann boundary condition $\bar{q}$ is specified. If the subdomain Ω_s is completely inside the domain Ω , $L_s = \partial\Omega_s$, such that the integral terms over Γ_{sq} and Γ_{su} boundaries vanish. The test function v is chosen such that it vanishes along L_s . This is accomplished using the Gaussian weight function from the MLS approximation as the test function v . The radius r_i of the support in the Gaussian weight function is replaced by the radius r_o of the subdomain Ω_s , such that v vanishes at r_o . The LSWF given in Eq. (1.81) is then rearranged as

$$\int_{\Omega_s} f_i v_i d\Omega + \alpha \int_{\Gamma_{su}} f v d\Gamma - \int_{\Gamma_{su}} q v d\Gamma = \int_{\Gamma_{sq}} \bar{q} v d\Gamma - \int_{\Omega_s} \bar{g} v d\Omega + \alpha \int_{\Gamma_{su}} \bar{f} v d\Gamma \quad (1.82)$$

Eq. (1.82) is solved at each node to get one algebraic equation in the form of nodal values. The terms f and v are replaced by Eqns. (1.77) and (1.78), respectively, to obtain the discrete form of Eq. (1.82).

The MLPG method is employed to solve several problems of engineering mechanics [12, 21, 50, 52], and achieves the higher rates of convergence, as compared with the FEM. Atluri [52-53] and Atluri and Shen [51] discussed the MLPG method in details with applications in fluid and solid mechanics, respectively. Atluri and Shen [51] discussed different global and local trial and test functions, and provided a broad framework, under which different MLPG forms can be derived by combining it with the different local trial and test functions. Several MLPG mixed schemes are also proposed, such as the finite difference method through MLPG [54], the finite volume method through MLPG [55], and the MLPG mixed collocation method [56]. These MLPG mixed schemes and the MLPG method are used to solve the various engineering problems, such as the large deformation [57], 3-D contact problems [58], heat conduction [59], crack analysis in the 2-D and 3-D domains [60], thermo-piezoelectricity [61], and shell deformation [62]. Cai and Zhu [63] modified the MLPG method to overcome the drawbacks of the Shepard partition of unity (PU) approximation by employing a novel PU-based Shepard and LS interpolations. Liu *et al.* [64] developed a meshless method for the analysis of structural dynamic problems by constructing the trial functions with the concept of natural neighbour, and combining them with the general MLPG method.

Several researchers also modified the MLPG method to solve the specific engineering problems, such as the Navier-Stokes and energy equations [65], limit analysis of plastic collapse [66], microelectromechanical systems [67], topology optimization [68-69], elasto-plastic fracture analysis [70], steady-state and transient heat conduction in the 3-D solid [71], boundary and initial value problems in piezo-electric and magneto-electric-elastic solids [72], thermal bending of Reissner-Mindlin plates [73], and 3-D potential problems [74]. It is observed from these applications that the MLPG method works well to solve these types of engineering problems.

1.6.9 Local boundary integral equation method

If the Galerkin approach based FEM (GFEM) is compared with the boundary element method (BEM), it is observed that the GFEM is more popular due to the local nature of the basis functions. This means that the GFEM basis functions are non-zero over a specific element and zero outside of it, while the BEM reduces the dimension of the problem by one, with the terms of trial function and its derivatives present under the integral sign applied over the global boundary of the domain only. The solution of a problem by the GFEM leads to the banded, sparse, and symmetric matrices, but with the BEM to the full and unsymmetrical matrices. The EFGM was developed by Belytschko *et al.* [5-6, 44-45] to overcome the difficulty of mesh generation in the FEM for complex 3-D domain. However, the EFGM still involves the integrals over the auxiliary elements, and encounters difficulties in imposing the essential boundary condition due to the use of MLS approximation. In order to combine the advantages of the GFEM, BEM and EFGM, Zhu *et al.* [14] and Atluri *et al.* [13] proposed the LBIE method. This method involves the boundary integration only over a local boundary around the concerned node, and uses the MLS approximation for the interpolation of the function values. The final stiffness matrix given by the LBIE method is sparse and banded, and inspite of the use of MLS approximation, the essential boundary conditions are imposed exactly.

A concept of companion solution [14] is introduced in the LBIE method, such that the LBIE for the solution of trial function inside the domain Ω of the given problem involves the trial function in the integral terms, only over the local boundary $\partial\Omega_s$ around the concerned node. This is in contrast with the BEM, where the trial function and its

gradient over the global boundary Γ of the domain Ω are involved under the integral terms. In the LBIE however, only if a source point is positioned on Γ , the integrals over the local boundary $\partial\Omega_s$ involves both the trial function and its gradient. The continuity requirement in the approximation of the trial function can be greatly relaxed in the formulation of LBIE, and no derivative of the shape functions is required while constructing the stiffness matrix of the system, at least for the nodes located in the internal domain of computation. The LBIE approach is understood as follows.

Let us consider a Poisson equation applied over the domain Ω that is enclosed by Γ , where $\Gamma = \Gamma_u \cup \Gamma_t$, and Γ_u and Γ_t are the boundaries involving the essential and natural boundary conditions respectively, namely

$$\nabla^2 u(x) = p(x) \quad x \in \Omega \quad (1.83)$$

$$u = \bar{u} \text{ on } \Gamma_u, \text{ and } \frac{\partial u}{\partial n} = \bar{q} \text{ on } \Gamma_t \quad (1.84)$$

where $\bar{u}$ and $\bar{q}$ are the prescribed values of the potential and normal flux, respectively, on the boundaries Γ_u and Γ_t , respectively, and n is the unit outward direction normal to Γ . A weak form of Eq. (1.83) is constructed as

$$\int_{\Omega} [\nabla^2 u(x) - p(x)] u^* d\Omega = 0 \quad (1.85)$$

where u^* and u are the test and trial functions, respectively. The chosen test function u^* satisfies the equation

$$\nabla^2 u^*(x, y) + \delta(x, y) = 0 \quad (1.86)$$

where $\delta(x, y)$ is the Dirac delta function. Applying an integration by parts twice to Eq. (1.85) leads to

$$u(y) = \int_{\Gamma} u^*(x, y) \frac{\partial u(x)}{\partial n} d\Gamma - \int_{\Gamma} u(x) \frac{\partial u^*(x, y)}{\partial n} d\Gamma - \int_{\Omega} u^*(x, y) p(x) d\Omega \quad (1.87)$$

where x and y are the generic and source points, respectively. Although the source point y is located within the domain Ω , it is known from the potential theory that Eq.

(1.87) holds over the entire domain Ω and the boundaries. As a result, Eq. (1.87) is called the global boundary integral equation (GBIE), and it is used to compute the value of unknown variable u at the source point y . If no entire domain Ω is involved, only a sub-domain Ω_s containing the source point y is considered, such that $\Omega_s \in \Omega$, then Eq. (1.87) should also hold over Ω_s , therefore

$$u(y) = \int_{\partial\Omega_s} u^*(x, y) \frac{\partial u(x)}{\partial n} d\Gamma - \int_{\partial\Omega_s} u(x) \frac{\partial u^*(x, y)}{\partial n} d\Gamma - \int_{\Omega_s} u^*(x, y) p(x) d\Omega \quad (1.88)$$

where $\partial\Omega_s$ is the boundary of Ω_s . Eq. (1.88) evaluates the unknown function u at the source point y by performing the integration over the closed boundary surrounding the concerned point, and over the subdomain that is entirely enclosed within the closed boundary. Eq. (1.87) of GBIE contains the terms of either the potential $\bar{u}$ or normal flux $\bar{q}$ defined at every point on Γ , which makes it a well posed problem. But none of the boundary conditions are known *a priori* while solving Eq. (1.88). Therefore, in order to get rid of the term $u(x)$ of gradient over the subdomain $\partial\Omega_s$ in Eq. (1.88), the concept of companion solution is introduced, which is associated with the fundamental solution. This is defined as the solution of Dirichlet problem over the subdomain Ω_s as

$$\nabla^2 u' = 0 \quad \text{on } \Omega_s \text{ and } u' = u^*(x, y) \quad \text{on } \partial\Omega_s \quad (1.89)$$

It is noted that the fundamental solution u^* is regular everywhere, except the source point y . Using $u^{**} = u^* - u'$ in Eq. (1.85) and performing the integration by parts twice results in

$$\begin{aligned} \int_{\Omega_s} -u(x) \nabla^2 u^*(x, y) d\Omega &= \int_{\partial\Omega_s} -u^{**}(x, y) \frac{\partial u(x)}{\partial n} d\Gamma - \int_{\partial\Omega_s} u(x) \frac{\partial u^{**}(x, y)}{\partial n} d\Gamma - \\ &\quad \int_{\Omega_s} u^{**}(x, y) p(x) d\Omega \end{aligned} \quad (1.90)$$

It is noted that $-\nabla^2 u^{**} = -\nabla^2 u^* + \nabla^2 u' = \delta(x, y)$ in Ω_s , and $u^{**} = 0$ along $\partial\Omega_s$. Eq. (1.90) is further modified by the definition of companion solution as

$$u(y) = - \int_{\partial\Omega_s} u(x) \frac{\partial u^{**}(x, y)}{\partial n} d\Gamma - \int_{\Omega_s} u^{**}(x, y) p(x) d\Omega \quad (1.91)$$

As such, only the unknown variable u appears in the integral form of local boundary. Eq. (1.91) is called the LBIE, and it is applicable to any shape of the subdomain Ω_s . The LBIE method is successfully employed to solve several problems from mechanics [13].

1.6.10 Point interpolation method

The PIM is an interpolation technique proposed by Liu and Gu [15], by which the terms of approximate function from the weak-form governing equation, as given in Eq. (1.82), are replaced by the expression of shape functions computed by the PIM.

Consider a function $u(x)$ defined over the domain Ω and discretized by the randomly distributed field nodes. The value of function $u(x)$ at each field node is interpolated by the surrounding field nodes via the polynomial as

$$u^h(x) = P^T(x) A(x) \quad (1.92)$$

where the function $u(x)$ is interpolated by the field nodes x_i ($i=1, 2, \dots, n$), $P^T(x)$ and $A(x)$ are the vectors of monomials and unknown coefficients, respectively, as given in Eq. (1.70) and

$$A^T(x) = [a_1, a_2, \dots, a_m] \quad (1.93)$$

The coefficients $A(x)$ in Eq. (1.92) are computed by satisfying Eq. (1.92) at all the n field nodes surrounding the concerned node at x . As such, the approximate function at the i^{th} node is given as

$$u_i = P^T(x_i) A \quad (1.94)$$

where u_i is the nodal value of u at the node $x = x_i$. Eq. (1.94) is imposed at all the n field nodes, and then the result is given in the matrix form as

$$u^e = P_Q A \quad (1.95)$$

where

$$\mathbf{u}^e = [u_1, u_2, \dots, u_n]^T, \quad \text{and } \mathbf{P}_Q^T = [P(x_1), P(x_2), \dots, P(x_n)] \quad (1.96)$$

The vector of unknown coefficients from Eq. (1.95) is given as

$$\mathbf{A} = \mathbf{P}_Q^{-1} \mathbf{u}^e \quad (1.97)$$

Substituting Eq. (1.97) into Eq. (1.92) results in

$$u^h(x) = \sum_{i=1}^n \mathbf{P}^T(x) \mathbf{P}_Q^{-1} \mathbf{u}^e \quad (1.98)$$

The shape functions by the PIM are then given as

$$N(x) = \mathbf{P}^T(x) \mathbf{P}_Q^{-1} \quad (1.99)$$

It is noted that $N_i(x_i) = 1$, $\sum_{i=1}^n N_i(x) = 1$, and $m = n$ to get the square matrix $\mathbf{P}_Q$ in Eq.

(1.95), namely the number of total monomial terms in $\mathbf{P}^T(x)$ should be equal to the number of total interpolation nodes in the local domain of the field node located at x . As such, based on the number of total interpolation nodes, the appropriate monomial terms are selected from the Pascal's triangle. The essential boundary conditions are easily imposed, as the PIM shape functions satisfy the Delta function property. Several test problems of the mechanics are solved by the PIM [15, 75], and good convergence rates are reported.

When the PIM is used, the trial and test functions from Eq. (1.82) are replaced by the shape functions obtained by the PIM interpolation [15, 75], and the integrals are solved by the Gauss quadrature with a separate cell structure, which is independent of the actual field nodes. In the EFGM, the interpolation is based on the arbitrarily distributed field nodes but $m \neq n$, this means that the total monomial terms in $\mathbf{P}^T(x)$ are not equal to the total nodes in the interpolation domain, while in the PIM $m = n$.

There are few characteristics in the PIM that should be considered before the implementation. The inverse of $\mathbf{P}_Q$ in Eq. (1.97) may not exist sometimes, if the required number of different values of x and y coordinates are not present [15]. In this situation, either x or y nodal coordinates of the field nodes that are in the local interpolation

domain of the concerned node may be perturbed slightly to get the nonsingular matrix P_Q . Let us consider the 1-D interpolation computed at the interested node x by the total 10 field nodes with the 2nd-order of the field variable distribution. As such, the vector $P^T(x)$ consists of the monomials up to the 9th-order, as per the PIM method, namely $P^T(x)=[1, x, x^2, \dots, x^9]$, and $m=10$. As a result, the 2nd-order distribution of the field variable is approximated by the 9th-order polynomial. According to Runge's phenomenon [76], this may cause oscillations along the boundaries of domain, if the field nodes are distributed uniformly. Therefore, the field nodes should be arranged in a sufficiently random manner to ensure the PIM to give the good approximation of field variables.

1.6.11 Local Kriging method

This method was proposed by Li *et al.* [16], in which the Kriging interpolation is used to construct an approximation of the function, and the local weak form of governing differential equation is derived by the weighted residual method. The spline function is used as the weighting function in the local weak form. The shape functions derived by this method possess the delta function property, such that the essential boundary conditions are imposed exactly.

Kriging interpolation is a generalized linear regression method that is used to formulate an optimal estimator in the sense of the minimum mean square error. It was introduced by Krige [77] in place of the moving averages to avoid the systematic errors in an interpolation. The local Kriging method was successfully applied to the free and forced vibrations of 2-D structure, and the dynamic characteristics of microelectromechanical systems (MEMS) [16].

The approximate function by Kriging method is given as

$$u^h(x) = \sum_{i=1}^n \lambda_i u(x_i) \quad (1.100)$$

where $u(x_i)$ is the nodal value of field variable at x_i , n is the total nodes in the local domain of x , and λ_i is a weight assigned to every node in the local domain. The weights

λ_i are computed in the way to minimize the squared variance of estimated error $E = [u(x) - u^h(x)]^2$.

1.6.12 Hermite cloud method

This method was proposed by Li *et al.* [19], in which the Hermite interpolation function is combined with the point collocation method to discretize the governing differential equations. Hermite interpolation theorem is based on the fixed RKPM, and is constructed such that the total unknowns include the field variables and their corresponding derivatives of the 1st-order. This method is good typically for the local high gradient problems, as the values of the 1st-order derivative are computed by solving the system of equations instead of an approximation. The approximate function by Hermite cloud method is given as

$$f^h(x, y) = \sum_{n=1}^{N_T} N_n(x, y) f_n + \sum_{m=1}^{N_s} \left(x - \sum_{n=1}^{N_T} N_n(x, y) x_n \right) M_m(x, y) g_{xm} + \sum_{m=1}^{N_s} \left(y - \sum_{n=1}^{N_T} N_n(x, y) y_n \right) M_m(x, y) g_{ym} \quad (1.101)$$

where N_T and N_s are the numbers of total scattered points in the interior domain and along the edges for approximation of the function and its derivatives, respectively, such that $N_s \leq N_T$, and g_{xm} and g_{ym} are unknown values of the derivative parameters at the m^{th} point with respect to the x and y variables, respectively. M_m are the shape functions corresponding to the unknown 1st-order differential functions g_x and g_y with the basis $\eta \in R^{\beta-1}$, where β is the order of the basis of function approximation. $N_n(x, y)$ is a shape function at the n^{th} point by the monomial basis up to the β^{th} -order in the x and y variables. It is seen from the above equation that the total unknowns are f_n , g_{xm} and g_{ym} .

Li *et al.* [25, 78] successfully applied the Hermite-cloud method for the dynamics simulation of the pipeline near submarine bed by considering the nonlinear fluid-structure

interaction, and for steady-state analysis of the pH-sensitive hydrogel. It is shown that this method has achieved very good convergence rate as compared with other meshless methods.

1.6.13 Gradient smoothing method

In general, the meshless methods based on the strong form need a special procedure to numerically compute the derivatives of functions at the field nodes. A novel GSM method was proposed by Liu *et al.* [20], in which the techniques of gradient smoothing and directional derivatives are adopted to compute the 1st- and 2nd-order approximate derivatives at the node of interest by assigning the weights to the nodes surrounding the interested node. The governing equations are discretized in the strong form by the standard collocation method. The GSM is developed such that it can be easily applied to complex geometries discretized by the random nodes.

The problem domain Ω is initially discretized by the triangular cells, as shown in Fig. 1.5, with the total M field nodes. For the i^{th} field node, the smoothing domain Ω_i is constructed by joining the centroids and mid-edge points of the triangular cells around the i^{th} field node, as shown in Fig. 1.5. It is noted that there is no overlapping between any two smoothing cells. A smooth operation to the gradient of function is done to compute the approximate derivatives as

$$\nabla^h u(x_i) = \int_{\Omega_i} \nabla^h u(x) \Phi(x - x_i) d\Omega_i \quad (1.102)$$

where Φ is a smoothing function. Performing an integration by parts in Eq. (1.102) results in

$$\nabla^h u(x_i) = \int_{\Gamma_i} u^h(x) n(x) \Phi(x - x_i) d\Gamma - \int_{\Omega_i} u(x) \nabla \Phi(x - x_i) d\Omega_i \quad (1.103)$$

In fact, any smoothing function can be used in the GSM. Here a weighted Shepard function is used as

Fig. 1.5. Schematic of GSM working principle.

$$\Phi(x - x_i) = \frac{\phi(x - x_i)}{\sum_{j=1}^M \phi(x - x_j) A_j} \quad (1.104)$$

where $A_i = \int_{\Omega_i} d\Omega$ is the area of total smoothing domain Ω_i . The function ϕ can be used

as a piecewise constant function as

$$\phi(x - x_i) = \begin{cases} 1 & x \in \Omega_i \\ 0 & x \notin \Omega_i \end{cases} \quad (1.105)$$

Therefore, the smoothing function Φ becomes

$$\Phi(x - x_i) = \begin{cases} 1/A_i & x \in \Omega_i \\ 0 & x \notin \Omega_i \end{cases} \quad (1.106)$$

Substituting Eq. (1.106) into (1.103), the smoothed gradient of the field variable is computed as

$$\begin{aligned}\nabla^h u(x_i) &= \int_{\Gamma_i} u^h(x) n(x) \Phi(x-x_i) d\Gamma \\ \therefore \nabla^h u(x_i) &= \frac{1}{A_i} \int_{\Gamma_i} u^h(x) n(x) d\Gamma\end{aligned}\tag{1.107}$$

The second term in Eq. (1.103) is equal to zero as the smoothing function Φ is constant. As a result, the area integral becomes the line integral in Eq. (1.107) along the portions of boundary Γ_i as

$$\nabla^h u_i = \frac{1}{A_i} \sum_{j=1}^{m_i} \left(L_{ij}^{(L)} n_{ij}^{(L)} u_{ij}^{(L)} + L_{ij}^{(R)} n_{ij}^{(R)} u_{ij}^{(R)} \right) \tag{1.108}$$

where m_i is the number of total cells covering the i^{th} field node, $L_{ij}^{(L)}$ and $L_{ij}^{(R)}$ are the lengths of the smoothing boundary along the left and right sides of the edge $i-j$, respectively, $N_{ij}^{(L)}$ and $N_{ij}^{(R)}$ are the outward unit normals over the left and right sides of the smoothing boundary with respect to the edge $i-j$, respectively, similarly $u_{ij}^{(L)}$ and $u_{ij}^{(R)}$ are the values of the field variable. The 2nd-order approximate gradient of the field variable is evaluated by following the procedure, as given from Eqns. (1.102) to (1.108)

$$\nabla^2 u(x_i) = \frac{1}{A_i} \int_{\Gamma_i} \nabla^h u(x) n(x) d\Gamma \tag{1.109}$$

Several schemes are developed to compute the values of $u_{ij}^{(L)}$ and $u_{ij}^{(R)}$, and the convergence analysis of the GSM is performed in details through the problems with the uniform as well as irregular computational domains [20]. In brief, the GSM is developed for the approximation of derivatives, which is further coupled with the simple collocation method to get the system of algebraic equations with the values of field variables as the unknowns.

1.6.14 Radial point interpolation based finite difference method

The conventional FDM requires the uniform grid of points to discretize the derivative terms in the governing differential equation. In order to extend the applicability of FDM, Liu *et al.* [26] developed the radial point interpolation based finite difference method

(RFDM) and applied it to the mechanics problems. In the RFDM, the radial point interpolation (RPI) is used to approximate the values of function at the field nodes, and the FDM is used to discretize the terms of derivative from the governing differential equation. Two types of nodal patterns are employed here to impose the FDM on the field nodes distributed randomly. The domain is first sprinkled with the field nodes distributed randomly, and then a background uniform grid is created for the FDM. The nodes on the grid of FDM are interpolated by the field nodes distributed randomly with the RPI method. The governing equation is discretized by the FDM at the background uniform grid, and the terms of function values at the FDM grid nodes are replaced by their corresponding interpolations in terms of the function values at the field nodes distributed randomly. As a result, the system of algebraic equations is obtained finally with the rows corresponding to the number of total background FDM nodes and the columns corresponding to the number of total field nodes distributed randomly.

The RBF is used for the data fitting, based on the arbitrarily distributed nodes. The RBF shape function is created at any interested node by the nodes in the local support domain of the interested node. Let the function $u(x)$ be defined over the domain Ω , and the interested node be located at the coordinate x_Q , where there are n nodes in the local support domain of the node located at x_Q . As a result, the value of $u(x)$ is approximated by the n field nodes using the RBF as

$$u^h(x, x_Q) = \sum_{i=1}^n R_i(x) a_i(x_Q) + \sum_{j=1}^m p_j(x) b_j(x_Q) = \mathbf{R}^T(x) \mathbf{a}(x_Q) + \mathbf{p}^T(x) \mathbf{b}(x_Q) \quad (1.110)$$

where n and m are the number of total nodes in the support domain and the number of monomial terms in the polynomial basis, respectively. $\mathbf{a}(x_Q)$ and $\mathbf{b}(x_Q)$ are the unknown coefficients vectors of $\mathbf{R}^T(x)$ and $\mathbf{p}^T(x)$, respectively, and given as

$$\mathbf{a}^T(x_Q) = \{a_1 \quad a_2 \quad \dots \quad a_n\}, \quad \text{and} \quad \mathbf{b}^T(x_Q) = \{b_1 \quad b_2 \quad \dots \quad b_n\} \quad (1.111)$$

The polynomial terms have to satisfy the extra condition to ensure unique approximation

$$\sum_{i=1}^n p_j(x_i) a_i = \mathbf{P}_m^T \mathbf{a} = 0, \quad j = 1, 2, \dots, m \quad (1.112)$$

Several types of RBF are available. One of them is the Multi-Quadrics (MQ) RBF that is adopted in the RFDM, and given as

$$R_i(x, y) = [r_i^2 + (\alpha_c d_c)^2]^q \quad (1.113)$$

where r_i is the Euclidean distance between the interpolation point located at x and its neighbourhood node at x_i . Enforcing Eq. (1.110) to pass through all the nodes in the support domain leads to

$$U_s = R_Q a(x_Q) + P_m b(x_Q) \text{ with } P_m^T a(x_Q) = 0 \quad (1.114)$$

where

$$R_Q = \begin{bmatrix} R_1(r_1) & R_2(r_1) & \dots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \dots & R_n(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(r_n) & R_2(r_n) & \dots & R_n(r_n) \end{bmatrix}_{n \times n}, \quad P_m = \begin{bmatrix} 1 & x_1 & y_1 & \dots & p_m(x_1) \\ 1 & x_2 & y_2 & \dots & p_m(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & y_n & \dots & p_m(x_n) \end{bmatrix}_{n \times m} \quad (1.115)$$

The term r_k in $R_i(r_k)$ from Eq. (1.115) is defined as

$$r_k = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (1.116)$$

Eq. (1.114) in the matrix form is given as

$$\tilde{U}_s = \begin{bmatrix} U_s \\ 0 \end{bmatrix} = \begin{bmatrix} R_Q & P_m \\ P_m^T & 0 \end{bmatrix}_{(n+m) \times (n+m)} \begin{Bmatrix} a(x_Q) \\ b(x_Q) \end{Bmatrix}_{(n+m) \times 1} = G a_0 \quad (1.117)$$

where

$$a_0^T = \{a_1 \quad a_2 \quad \dots \quad a_n \quad b_1 \quad b_2 \quad \dots \quad b_m\}, \quad \tilde{U}_s = \{u_1 \quad u_2 \quad \dots \quad u_n \quad 0 \quad 0 \quad \dots \quad 0\} \quad (1.118)$$

$$G = \begin{bmatrix} R_Q & P_m \\ P_m^T & 0 \end{bmatrix} \quad (1.119)$$

It is noted that the matrix G is symmetric. The solution of Eq. (1.117) leads to

$$a_0 = G^{-1} \tilde{U}_s \quad (1.120)$$

Substituting Eq. (1.120) into Eq. (1.110) results in

$$u^h(x, x_Q) = \{R^T(x) \quad p^T(x)\} G^{-1} \tilde{U}_S \quad (1.121)$$

$$\therefore u^h(x, x_Q) = \tilde{\Phi}^T(x) \tilde{U}_S, \text{ where } \tilde{\Phi}^T(x) = \{R^T(x) \quad p^T(x)\} G^{-1} \quad (1.122)$$

Eq. (1.122) can be rewritten as

$$u^h(x, x_Q) = \tilde{\Phi}^T(x) \tilde{U}_S = \sum_{i=1}^n \phi_i u_i \quad (1.123)$$

Eq. (1.123) is the final expression of the RBF shape functions. The approximate derivatives of the field variables are computed by simply differentiating Eq. (1.123) as

$$u_{,d}^l(x) = \tilde{\Phi}_{,d}^T(x) \tilde{U}_S \quad (1.124)$$

where $_{,d}$ denotes the derivative in either the x or y coordinate direction.

The working principle of RFDM is schematically shown in Fig. 1.6. If there are m FD grid points and n random field nodes, and the governing differential equation contains the derivative term $\partial u / \partial x$, this derivative term is discretized at the FD grid point (x_i, y_j) by the classical FDM with the central difference scheme as

$$\left(\frac{\partial u}{\partial x} \right)_{i,j} = \frac{1}{2h} (u_{i+1,j} - u_{i-1,j}) \quad (1.125)$$

The term $u(x_i, y_j)$ from Eq. (1.125) is replaced by Eq. (1.123) as

$$u(x_i, y_j) = \sum_{k=1}^n \phi_k u_k \quad (1.126)$$

Substituting Eq. (1.126) into Eq. (1.125) leads to a single algebraic equation as

$$\left(\frac{\partial u}{\partial x} \right)_{i,j} = \{k_i\}_{1 \times n} \{u_k\}_{n \times 1} \quad (1.127)$$

Similarly, Eq. (1.125) is applied at all the m FD grid points resulting in a matrix as

$$[K]_{m \times n} \{U\}_{n \times 1} = \{F\}_{m \times 1} \quad (1.128)$$

Fig. 1.6. Schematic of the RFDM working principle.

It is seen that the stiffness matrix K in Eq. (1.128) is rectangular with $m \geq n$. In order to solve Eq. (1.128) uniquely, rank of the matrix K should be at least equal to n . The rectangular system of equations is solved by the LS approach, such that $[K]_{n \times m}^T$ is multiplied on both sides of Eq. (1.128), resulting in the square stiffness matrix as

$$K_{n \times m}^T K_{m \times n} U_{n \times 1} = K_{n \times m}^T F_{m \times 1} \quad (1.129)$$

$$\bar{K}_{n \times n} U_{n \times 1} = \bar{F}_{n \times 1} \quad (1.130)$$

Eq. (1.130) is solved for the values of unknown field variables at the random field nodes.

Several mechanics problems with the regular and irregular domains were solved and the convergence analysis was performed by the RFDM [26]. It is shown that the convergence rates obtained by the RFDM are reasonably good even for the irregular domains.

In summary, the working principles of several meshless methods based on the strong and weak forms are discussed in this subsection, and it is seen that the Galerkin-based

weak-form approach is extensively employed. Although most of the meshless methods do not require any mesh and the interconnectivity information among the field nodes, they still need an auxiliary set of nodes, which is used either for the numerical integration or for the derivative discretization.

1.7 Outline of the thesis

This thesis is composed of seven chapters, and each of the chapters consists of several sections for good organization. They are described briefly as follows.

Chapter 2 conducts the development of the RDQ method in details, with full mathematical formulations associated with the imposition of the Neumann and Dirichlet boundary conditions. Before that, formulations of the fixed RKPM and DQ methods are presented, as a preliminary work. Several approaches are also discussed for evaluation of the approximate derivatives of the field variables. Implementation details of the RDQ method are given in the form of a flowchart.

Chapter 3 exhaustively deals with the convergence and consistency analyses of the RDQ method, by solving several 1-D, 2-D and elasticity problems. The superconvergence condition is developed to improve the convergence rates of the RDQ method. Two typical configurations of the microswitches, based on the fixed-fixed and cantilever beams, are analyzed by the RDQ method for the pull-in instability.

Chapter 4 presents the in-depth stability analysis of the RDQ method via the 1st-order wave, transient heat conduction, and transverse beam deflection equations. The stability criterion for these equations is achieved by combining the results obtained from the stability and consistency analyses to ensure the stable and consistent discretization of the governing differential equations.

Chapter 5 focuses on the adaptive analysis of the RDQ method by combining the adaptive algorithm of the h -refinement with the RDQ method. In order to develop a robust adaptive RDQ method, an approach with the cross product of vectors is developed to ensure that all the newly created field nodes are always within the computational domain. An error recovery technique, based on the LS approximation, is proposed to improve the accuracy of the numerical results.

Chapter 6 employs the RDQ method for the 2-D simulation of pH-sensitive hydrogel. The effects of the solution pH and initial fixed-charge concentration are studied on the swelling of the hydrogel, and they are consistent with the nature of the problem. Effects of the Young's moduli and different geometrical shapes, such as square, rectangle and curved boundaries, are also studied for the hydrogel. The different profiles of the field variable distributions are analyzed as well.

Chapter 7 summarizes and concludes the research work, and then suggests the possible future work to be performed.

1.8 Summary

In this chapter, the fundamentals and common steps of the meshless methods are discussed, and it is noted that they are not much different from the classical FEM, except the computation of shape function and the evaluation of the integral terms. The motivation and objectives behind the present research work are also discussed.

The DQ method is discussed, and it is seen that several approaches are proposed to compute the weighting coefficients [33-37]. One of them is Shu's general approach [36-37], which is adopted in the RDQ method.

In order to get the deeper understanding of the working principle of meshless methods, several well-established methods based on the strong- or weak-form of the governing differential equations are discussed. It may be noted from the literature review that, although an extensive research work has been done in the meshless methods based on the weak-form approach, comparatively the meshless methods based on the strong-form approach are still need to be widely explored further.

Chapter 2

Development of novel strong-form meshless random differential quadrature (RDQ) method

2.1 Introduction

The main objective of the development of RDQ method is to extend the applicability of the DQ method over the irregular and non-uniform domains discretized by either the uniform or random field nodes. Two types of nodal distributions are used in the RDQ method. Firstly, the background nodes called the virtual nodes are created in a fixed pattern (uniform or Chebyshev-Gauss-Lobatto nodes), and secondly the field nodes are created either in the uniform or random manner, as shown in Fig. 2.1. The virtual nodes are interpolated by surrounding field nodes with the fixed RKPM interpolation function, and a linear transformation matrix is developed to relate the values of the field variable at the virtual nodes with the values of the nodal parameter at the field nodes. The governing differential equations and boundary conditions are discretized at the virtual nodes located within the domain and along the boundaries, respectively, by the locally applied DQ method. The local domains are created around each virtual node in the x and y directions, in which the virtual nodes falling are considered for the discretization of derivative terms from the governing partial differential equations (PDE) at the concerned virtual node in the respective directions, as shown in Fig. 2.1. In the discretized equations, the terms related to the values of the function at the virtual nodes are replaced by the corresponding interpolation equations using the linear transformation matrix developed earlier. A system of algebraic equations is obtained after simplifying it, which is solved for the unknown values of the nodal parameter u_i at the field nodes. Since the fixed RKPM function does not possess the Dirac delta function property, the values of nodal parameter u_i at the field nodes are not equal to the values of function approximation f_i^h at the field nodes, namely $u_i \neq f_i^h$. As a result, the values of approximate function f_i^h at the field nodes are computed by interpolating each field node by the nearby field nodes with the

fixed RKPM interpolation function as $f_i^h = \sum_{j=1}^{NP} N_j u_j$, where N_j and u_j are the values of shape function and nodal parameter at the j^{th} field node, respectively. Total NP field nodes surrounding the i^{th} field node are included in the interpolation domain of the i^{th} field node, such that $NP \subseteq N$, where N is the total field nodes in the domain. This is broadly the working principle of RDQ method.

Fig. 2.1. Working principle of the RDQ method for the irregular computational domain discretized by the random field nodes coupled with virtual nodes distributed in a fixed pattern.

One of the important requirements of the DQ method is that all the discretized field nodes should be always arranged in a collinear manner, and the domain should be a regular one. Suppose, while solving a dynamic problem by the DQ method, all the field nodes are scattered in a specific pattern at the beginning of computation. After the first time increment, the field nodes get randomly distributed with a modified boundary, this situation in turn restricts the application of the DQ technique for the further time increments. As a result, the DQ method is restricted only for the problems with regular

domains discretized by the field nodes distributed in a fixed pattern. This limitation is overcome by the present RDQ method via the fixed RKPM interpolation function to construct a novel strong-form meshless method.

Compared with other existing methods, the merit of the RDQ method is the combination of both the Eulerian and Lagrangian grids. The virtual nodes represent Eulerian grid that is fixed in a computational domain. The field nodes represent Lagrangian grid, which freely moves in a domain as per the computation. No interconnectivity information among the field nodes is required while computing the values of approximate function, such that they are free to move anywhere in the domain. The governing equation is easily discretized by the virtual nodes at every time increment, as they are fixed in a space. As a result, the proposed RDQ method is applicable to solve the large deformation and moving boundary problems, namely the crack propagation and hydrogel simulation, where the classical FEM faces many challenging issues due to the distortion of mesh and singularity in elements. As compared with the other meshless methods, especially the weak form methods, the RDQ method is also capable of capturing the local high gradients.

The fixed RKPM interpolation function and the formulations of DQ method are presented in the following subsections, and consequently the working principle of the RDQ method is elaborated.

2.2 Formulation of the fixed reproducing kernel particle method

The approximate equation by the fixed RKPM interpolation function is derived in this section.

The fixed RKPM interpolation function is a special case of RKPM function, in which the kernel is fixed at the central node, as explained [47]. The fixed RKPM interpolation function has several advantages over the other forms, such as the classical, moving and multiple-fixed RKPM interpolation functions. For example, it has the constant moment matrix due to the fixed kernel, and it follows the partition of unity principle. The approximate value of function as per the fixed RKPM interpolation is given as

$$f^h(x, y) = \int_{\Omega} C(x, y, u, v) K(x_k - u, y_k - v) f(u, v) du dv \quad (2.1)$$

where $C(x, y, u, v)$ is unknown correction function, $f^h(x, y)$ is an approximation of the function $f(x, y)$ at a node (x, y) , and $K(x_k - u, y_k - v)$ is the kernel function fixed at the node (x_k, y_k) . Order of the approximate function $f^h(x, y)$ is determined by the order of the monomial basis functions used in the correction function $C(x, y, u, v) = P^T(u, v) c(x, y)$, where $P^T(u, v) = \{b_1(u, v), b_2(u, v), \dots, b_m(u, v)\}$ is the m^{th} -order monomial column vector and $c(x, y)$ is the m^{th} -order unknown row vector. The unknown correction functions are computed by the consistency condition, such that the approximate function $f^h(x, y)$ passes through all the nodes located in the domain Ω as

$$C(x, y, u, v) = P^T(u, v) c(x, y) \quad (2.2)$$

where $c(x, y)_{(1 \times m)} = \{c_1(x, y), c_2(x, y), \dots, c_m(x, y)\}$ is the m^{th} -order row vector of the unknown correction function coefficients. The 1-D basis of the linear monomials in the polynomial basis function are given as

$$P^T(u, v) = \{1, u, v\}, \quad (m = 3) \quad (2.3)$$

The 2-D basis of the quadratic monomials are given as

$$P^T(u, v) = \{1, u, v, u^2, uv, v^2\}, \quad (m = 6) \quad (2.4)$$

The coefficients of unknown correction function $c(x, y)$ are determined by the consistency or reproducing condition, as the approximate function is required to satisfy every monomial of the polynomial basis function as

$$b_i(x, y) = \int_{\Omega} P^T(u, v) c(x, y) K(x_k - u, y_k - v) b_i(u, v) du dv, \quad i = 1, 2, \dots, m \quad (2.5)$$

This equation is rewritten in a discretized or particle form as

$$\{b_i(x, y)\}_{(m \times 1)} = \sum_{I=1}^{NP} P^T(x_I, y_I) c(x, y) K(x_k - x_I, y_k - y_I) b_i(x_I, y_I) dV_I \quad (2.6)$$

where NP is the total field nodes in a local interpolation domain of the virtual node (x_k, y_k) , and dV_I is a nodal volume at the I^{th} field node. Eq. (2.6) is written in a matrix form as

$$\{b_i(x, y)\} = M c(x, y) \quad (2.7)$$

where M is an $m \times m$ moment matrix, which is given in the discretized form as

$$M_{ij} = \sum_{l=1}^{NP} \{b_i(x_l, y_l)\} K(x_k - x_l, y_k - y_l) b_j(x_l, y_l) dV_l, \quad i \text{ and } j = 1, 2, \dots, m \quad (2.8)$$

It is seen from Eq. (2.8) that the moment matrix is a constant matrix. This is an advantage of the fixed RKPM interpolation function instead of its other variants, such as the moving or multiple fixed RKPM functions, as their moment matrix is not constant but needs to be computed for each local domain. Eq. (2.8) is written in generalized form as

$$M = F W F^T \quad (2.9)$$

where F is a $m \times NP$ matrix given as

$$F = \begin{bmatrix} b_1(x_1, y_1) & b_1(x_2, y_2) & \cdots & b_1(x_{NP}, y_{NP}) \\ b_2(x_1, y_1) & b_2(x_2, y_2) & \cdots & b_2(x_{NP}, y_{NP}) \\ \vdots & \vdots & \ddots & \vdots \\ b_m(x_1, y_1) & b_m(x_2, y_2) & \cdots & b_m(x_{NP}, y_{NP}) \end{bmatrix} \quad (2.10)$$

W is a $NP \times NP$ diagonal matrix given as

$$W = \begin{bmatrix} K(x_k - x_1, y_k - y_1) dv_1 & 0 & \cdots & 0 \\ 0 & K(x_k - x_2, y_k - y_2) dv_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K(x_k - x_{NP}, y_k - y_{NP}) dv_{NP} \end{bmatrix} \quad (2.11)$$

The vector $c(x, y)$ is obtained from Eq. (2.7) as

$$c(x, y)_{(m \times 1)} = M^{-1} \{b_i(x, y)\}_{(m \times 1)} \quad (2.12)$$

Substituting Eqns. (2.12) and (2.2) into Eq. (2.1) results in

$$f^h(x, y) = \int_{\Omega} P^T(x, y) M^{-1} \{b_i(u, v)\} K(x_k - u, y_k - v) f(u, v) du dv \quad (2.13)$$

This equation is rewritten in the discretized form as

$$f^h(x, y) = \sum_{l=1}^{NP} P^T(x, y) M^{-1} B(x_l, y_l) K(x_k - x_l, y_k - y_l) f(u, v) dV_l \quad (2.14)$$

$$f^h(x, y) = \sum_{I=1}^{NP} N_I(x, y) u_I \quad (2.15)$$

where $N_I(x, y)$ and u_I are the values of the shape function and unknown nodal parameter at the I^{th} field node, respectively. The $N_I(x, y)$ is given as

$$N_I(x, y) = P(x, y)_{l \times m} M_{m \times m}^{-1} \begin{bmatrix} b_1(x_I, y_I) K(x_k - x_I, y_k - y_I) \\ b_2(x_I, y_I) K(x_k - x_I, y_k - y_I) \\ \vdots \\ b_m(x_I, y_I) K(x_k - x_I, y_k - y_I) \end{bmatrix}_{m \times 1} \Rightarrow P(x, y) C_{II}^{-1} \quad (2.16)$$

where C_{II}^{-1} denotes the I^{th} column of C^{-1} , such that $I \subseteq NP$. Eq. (2.15) is a final expression for the approximation of function by the fixed RKPM interpolation.

The kernel function is a window function that approximates the distribution of field variables over a local domain, therefore it has a nonzero value over a certain sub-domain and zero outside of it.

$$K(x_k - u, y_k - v) = \frac{1}{d_x} w\left(\frac{x_k - u}{d_x}\right) \frac{1}{d_y} w\left(\frac{y_k - v}{d_y}\right) \quad (2.17)$$

$$w(z_I) = \begin{cases} 0, & z_I < -2 \\ \frac{1}{6}(z_I + 2)^3, & -2 \leq z_I \leq -1 \\ \frac{2}{3} - z_I^2(1 + \frac{z_I}{2}), & -1 \leq z_I \leq 0 \\ \frac{2}{3} - z_I^2(1 - \frac{z_I}{2}), & 0 \leq z_I \leq 1 \\ \frac{-1}{6}(z_I - 2)^3, & 1 \leq z_I \leq 2 \\ 0, & z_I > 2 \end{cases} \quad (2.18)$$

where $z_I = [(x_k - u)/d_x]$, and d_x and d_y are the cloud sizes in the x and y directions, respectively. The cubic spline function $w(z_I)$ is used in the RDQ method, as given in Eq. (2.18). It is observed from Eq. (2.1) that the fixed RKPM interpolation function does not possess the delta function property, but with the cubic spline function $w(z_I)$ in Eq.

(2.18), it has the partition of unity property, such that the summation of the shape function values at all the NP field nodes is equal to one as given by

$$\sum_{I=1}^{NP} N_I(x, y) = 1 \quad (2.19)$$

Jin *et al.* [79] and Atluri and Shen [51] discussed the equivalence between the RKPM and MLS shape functions. Atluri and Shen [51] showed that the shape functions obtained by the RKPM and MLS are identical if the same kernel and the window functions are chosen in both with the same consistency order k .

2.3 Formulation of the differential quadrature method

The key task in the DQ method is to compute the DQ weighting coefficients. Several approaches have been proposed to compute the weighting coefficients [33-36]. One of them is Shu's general approach [36] based on the Lagrange interpolation polynomials, which is adopted in the RDQ method. Locally applied DQ method is used in the RDQ method, in which the derivative terms from the governing differential equation are discretized at the concerned virtual node by the nearby surrounding virtual nodes only, as shown in Fig. 2.1. The 1st-order derivative in the x direction by locally applying the DQ method is given as

$$f_{,x}^{(1)}(x_i) = \sum_{j=1}^{N_x} a_{ij} f(x_j) \quad (2.20)$$

where $f_{,x}^{(1)}(x_i)$ is the approximate 1st-order derivative at the virtual node (x_i) in the x direction, $f(x_j)$ is the approximate function value at the j^{th} virtual node, where $j \in [1, N_x]$. a_{ij} are the DQ weighting coefficients of the 1st-order derivative, and evaluated by Shu's general approach [36] as

$$a_{ij}^x = \frac{1}{x_i - x_j} \prod_{m=1, m \neq i, j}^{N_x} \left(\frac{x_i - x_m}{x_j - x_m} \right), \quad \text{and} \quad a_{ii}^x = - \sum_{k=1, k \neq i}^{N_x} a_{ik}^x \quad (2.21)$$

The approximate derivatives of the 1st-order for 2-D problem are similarly given by

$$f_x^{(1)}(x_i, y_j) = \sum_{k=1}^{N_x} a_{ik}^x f(x_k, y_j) \quad (2.22)$$

$$f_y^{(1)}(x_i, y_j) = \sum_{k=1}^{N_y} a_{jk}^y f(x_i, y_k) \quad (2.23)$$

where a_{ik}^x and a_{jk}^y are the DQ weighting coefficients in the x and y direction, respectively, which are computed by Eq. (2.21), and N_x and N_y are the total virtual nodes in the local DQ domain of the concerned virtual node (x_i, y_j) in the x and y directions, respectively. Similarly, the approximate derivatives of the 2nd-order are as

$$\frac{d^2 f(x_i, y_j)}{dx^2} = \sum_{k=1}^{N_x} b_{ik} f^h(x_k, y_j) \quad (2.24)$$

$$\frac{d^2 f(x_i, y_j)}{dy^2} = \sum_{k=1}^{N_y} b_{jk} f^h(x_i, y_k) \quad (2.25)$$

$$\frac{d^2 f(x_i, y_j)}{dx dy} = \sum_{l=1}^{N_x} a_{il} \sum_{k=1}^{N_y} a_{jk} f^h(x_l, y_k) \quad (2.26)$$

where b_{ik} and b_{jk} are the DQ weighting coefficients, which are computed as

$$b_{ij} = 2 a_{ij} \left[a_{ii} - \frac{1}{x_i - x_j} \right], \text{ for } j=1, 2, \dots, N_x, \forall i \neq j: b_{ij}, b_{ii} = - \sum_{j=1, j \neq i}^{N_x} b_{ij} \quad (2.27)$$

The recurring formula for the computation of higher order derivatives have been developed [37].

It is difficult to ensure a good numerical solution by accurately computing the weighting coefficients. Quan and Chang observed that the accuracy of the numerical solution depends on the values of the weighting coefficient as well as the distribution of grid points in a domain [35]. They showed that the general collocation method is actually same as the DQ method, but the grid points placed at the roots of the 1st-kind of Chebyshev polynomials consistently gives better solution by the DQ method than that obtained by the orthogonal collocation method. They implemented the general procedure

to solve a PDE [34-35], and presented the numerical results with the DQ weighting coefficients computed by Eqns. (2.28) to (2.30) as

$$a_{ij} = \frac{1}{x_j - x_i} \prod_{\substack{m=1 \\ m \neq i, j}}^n \frac{x_i - x_m}{x_j - x_m}, \quad \text{for } i \neq j \text{ and } a_{ii} = \sum_{\substack{k=1 \\ k \neq i}}^n \frac{1}{x_i - x_k}, \quad \text{for } i = j \quad (2.28)$$

$$b_{ij} = \frac{2}{x_j - x_i} \left(\prod_{\substack{m=1 \\ m \neq i, j}}^n \frac{x_i - x_m}{x_j - x_m} \right) \left(\sum_{\substack{k=1 \\ k \neq i, j}}^n \frac{1}{x_i - x_k} \right), \quad \text{for } i \neq j \quad (2.29)$$

$$b_{ii} = 2 \sum_{\substack{k=1 \\ k \neq i}}^{n-1} \left[\frac{1}{x_i - x_k} \left(\sum_{\substack{l=k+1 \\ l \neq i}}^n \frac{1}{x_i - x_l} \right) \right], \quad \text{for } i = j \quad (2.30)$$

where a_{ij} and a_{ii} , and b_{ij} and b_{ii} are the weighting coefficients of the 1st- and 2nd-order derivatives, respectively, that are computed at the i^{th} node using the total n domain nodes. Even though both, Shu [36-37] and Quan and Chang [34-35], assumed Lagrange interpolation polynomial as a test function, the final weighting coefficient equations are little different. Using the linear vector space analysis, Shu [36] proved that different polynomials used in Bellman [33] and Quan and Chang [34-35] approaches are nothing but different sets of base polynomial vectors of a function approximation, such that all the base vectors satisfy the equation of function approximation if one of them do so.

In summary, the fixed RKPM function and DQ method are discussed, as a preliminary work. Out of several approaches to compute the DQ weighting coefficients, Shu's general approach [36-37] is adopted in the RDQ method. Development of the RDQ method is explained in the next subsection.

2.4 Development of the RDQ method

As explained earlier, two types of distributive nodes are used in the RDQ method. The first is called the virtual nodes that are used for the discretization of governing equation, and the second is called the field nodes that are used to determine the field variable distributions within the computational domain, as shown in Fig. 2.1.

The detailed working principle of the RDQ method along with the implementation of boundary conditions is discussed in the subsequent subsections.

2.4.1 Bridge to link the discretization of governing differential equation with the approximation of function value at the field nodes

The objective of the development of RDQ method is to extend the use of DQ method for a regular computational domain discretized by the random field nodes, and for an irregular computational domain discretized by the uniform or random field nodes. This is achieved by creating the virtual and field nodes in a computational domain, as shown in Fig. 2.1. A local domain is created around every virtual node in the x and y directions, and the virtual nodes falling in it are considered for the discretization of derivative term at the concerned virtual node; this domain is called as the DQ local domain. The derivative terms from the governing differential equation are approximated at the virtual nodes located in the internal computational domain, by applying the DQ method with the DQ local domain of the concerned virtual node. Another local domain is created around each virtual node, in which the field nodes (uniform or random) falling are considered for the approximation of field variable values at the concerned virtual node by the fixed RKPM interpolation function, as shown in Fig. 2.1. When all the equations of the approximation of field variables at the virtual nodes are combined, a transformation matrix comprising of the values of shape function is obtained that associate the values of field variable at the virtual nodes with the values of nodal parameter at all the field nodes. In the discretized governing equation, the terms of the function values at the virtual nodes are replaced by their corresponding transformation equations in the form of the shape functions that are developed earlier. The discretized governing equation in this way is represented in the form of the unknown values of nodal parameters at all the field nodes.

Any shape can be used to create the local interpolation domain. A circular shape is adopted in the RDQ method for a convenience purpose. Let us consider $d_x = \alpha_x \Delta_x$ as the domain size in the x direction [47], where α_x is the size of local domain and Δ_x is an average nodal spacing in the x direction. Two approaches are studied to define the value of α_x . First one is by the numerical analysis, and the second by developing a positivity condition. The value of α_x defined by the positivity condition ensure that the kernel is non-zero over a local domain of interpolation and zero outside of it. The positivity condition is derived based on the fact that the fixed RKPM interpolation function is

nonzero only over a certain local domain. As per Eq. (2.18), the kernel function is zero if $z_I > 2$ or $z_I < -2$, which means that to have the non-zero value of kernel function, the α_x and α_y values should satisfy $z_I < 2$ or $z_I > -2$ conditions, as given by

$$\frac{(x_k - x_i)}{(\alpha_x \Delta_x)} \geq -2, \text{ and } \frac{(x_k - x_i)}{(\alpha_x \Delta_x)} \leq 2 \quad (2.31)$$

$$\therefore \alpha_x \in \left[\left(\frac{x_k - x_i}{2 \Delta_x} \right), - \left(\frac{x_k - x_i}{2 \Delta_x} \right) \right], \text{ for the domain size } [2, -2] \quad (2.32)$$

where the index k refers the fixed virtual node, and the index i refers the field node that is in the local interpolation domain of the k^{th} virtual node. The highest absolute value of α_x is selected out of all the values obtained from Eq. (2.32). From the numerical analysis approach, α_x equal to 1.17 and 2.23 are found to give good results. Comparing the results obtained from both the approaches, α_x equal to 1.17 [47] is used in the RDQ method.

Let Eq. (2.24) be discretized at the virtual node (x_i, y_j) by the RDQ method as

$$\frac{d^2 f(x_i, y_j)}{dx^2} = \{b_{i1} \quad b_{i2} \quad \dots \quad b_{iN_x}\}_{1 \times N_x} \begin{bmatrix} f^h(x_1, y_j) \\ f^h(x_2, y_j) \\ \vdots \\ f^h(x_{N_x}, y_j) \end{bmatrix}_{N_x \times 1} \quad (2.33)$$

The terms $f^h(x_{N_x}, y_j)$ in Eq. (2.33) are replaced by the fixed RKPM interpolation functions from the linear transformation matrix as

$$\begin{bmatrix} f^h(x_1, y_j) \\ f^h(x_2, y_j) \\ \vdots \\ f^h(x_{N_x}, y_j) \end{bmatrix}_{(N_x \times 1)} = \begin{bmatrix} N_1(x_1, y_j) & N_2(x_1, y_j) & \dots & N_{NP}(x_1, y_j) \\ N_1(x_2, y_j) & N_2(x_2, y_j) & \dots & N_{NP}(x_2, y_j) \\ \vdots & \vdots & \ddots & \vdots \\ N_1(x_{N_x}, y_j) & N_2(x_{N_x}, y_j) & \dots & N_{NP}(x_{N_x}, y_j) \end{bmatrix}_{(N_x \times NP)} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{NP} \end{bmatrix}_{(NP \times 1)} \quad (2.34)$$

Substituting Eq. (2.34) in Eq. (2.33) results in

$$\frac{d^2 f(x_i, y_j)}{dx^2} = \{b_{i1} \quad \dots \quad b_{iN_x}\}_{1 \times N_x} \begin{bmatrix} N_1(x_1, y_j) & \dots & N_{NP}(x_1, y_j) \\ \vdots & \ddots & \vdots \\ N_1(x_{N_x}, y_j) & \dots & N_{NP}(x_{N_x}, y_j) \end{bmatrix}_{(N_x \times NP)} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{NP} \end{bmatrix}_{(NP \times 1)} \quad (2.35)$$

Eq. (2.35) is further simplified to

$$\frac{d^2 f(x_i, y_j)}{dx^2} = \{k_1 \quad k_2 \quad \dots \quad k_{NP}\}_{1 \times NP} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{NP} \end{bmatrix}_{NP \times 1} \quad (2.36)$$

where

$$k_m = \{b_{i \ N_x}\}_{1 \times N_x} \{N_m\}_{N_x \times m}, \text{ where } m \in [1, NP] \quad (2.37)$$

As a result, Eq. (2.36) is the final linear algebraic expression for Eq. (2.24) in terms of the unknown values of the nodal parameters at the field nodes. It is noted from Eq. (2.34) that some values of the shape function may be zero, therefore the final stiffness matrix is banded and sparse. Similarly, Eq. (2.24) is discretized at all the virtual nodes that are located in the internal computational domain of the problem, and corresponding Eq. (2.36) is obtained at all the virtual nodes. All the equations via the discretization and boundary conditions are combined to form the final stiffness matrix. Therefore, the numbers of the total rows and columns of the final stiffness matrix are given by the total virtual and field nodes, respectively. As such, the total virtual nodes should be at least equal to or greater than the total field nodes, resulting in the non-symmetric and rectangular stiffness matrix. The detailed procedure to solve the final stiffness matrix is elaborated in Section 2.4.4.

2.4.2 Imposing the Dirichlet boundary conditions

The fixed RKPM interpolation function does not possess the Kronecker delta function property, thus $u_I \neq f^h(x_I, y_I)$, namely the value of nodal parameter u_I at the field node is not equal to the value of approximate function $f^h(x_I, y_I)$. But, it is still possible to exactly impose the Dirichlet boundary condition. As the governing differential equations and boundary conditions are imposed at the virtual nodes, extra virtual nodes are created at the location of field nodes that are scattered along the Dirichlet boundary to ensure that the boundary values at all the field nodes located on the Dirichlet boundary are imposed. It is assumed for all the virtual nodes (original and extra created) located along the

Dirichlet boundary that $u_I = f^h(x_I, y_I)$. For example, in a given 1-D domain discretized by the number of total 4 virtual nodes, the 1st and 4th virtual nodes are on the Dirichlet boundary, and the corresponding function values imposed are b_1 and b_4 . The modified global matrix after imposing $u_I = f^h(x_I, y_I)$ for the 1st and 4th virtual nodes is given as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ F_2 \\ F_3 \\ b_4 \end{Bmatrix} \quad (2.38)$$

The values of u_1 and u_4 are obtained as b_1 and b_4 by solving Eq. (2.38), respectively.

2.4.3 Imposing the Neumann boundary conditions

The Neumann boundary condition is imposed by two ways. By the first approach, all the equations of Neumann boundary condition are discretized by the RDQ method at the virtual nodes that are located on the Neumann boundary. By the second approach, the Neumann boundary condition is converted to the Dirichlet boundary condition, and then imposed as the Dirichlet boundary condition. Let us take Eq. (2.22) as example, which needs to be satisfied at the virtual node (x_i, y_i) located on the Neumann boundary

$$\frac{df(x_i, y_j)}{dx} = 0 \quad (2.39)$$

Eq. (2.39) is modified as

$$\left. \frac{df}{dx} \right|_i = \frac{f_{next} - f_i}{h} = 0 \quad \Rightarrow f_{next} = f_i \quad (2.40)$$

where f_i and f_{next} are the values of function at the virtual node (x_i, y_i) and immediately next virtual node in x direction, respectively, and h is the nodal spacing as $h = x_{next} - x_i$. As a result, the Neumann boundary condition in Eq. (2.39) at the virtual node (x_i, y_i) is converted to the Dirichlet boundary condition. Finally, these boundary equations are assembled into the global stiffness matrix.

It is observed by solving several test problems via both the approaches that the first approach by the RDQ method gives good results for the problems with complex equations of Neumann boundary conditions, such as the elasticity problem which has different stresses as the Neumann boundary conditions. The second approach gives good results for the problems with simple equations for the Neumann boundary condition.

2.4.4 Solving the final system of equations and computing the approximate function values

It is necessary to have the virtual nodes at least equal to the field nodes to get the unique solution, and the solution is improved as the virtual nodes are further increased. Suppose the final system of equations is $K_{m \times n} U_{n \times 1} = F_{m \times 1}$, where m and n are the number of total virtual and field nodes, respectively, in the computational domain such that $m \geq n$. In order to get a square stiffness matrix K , if $m > n$, it is solved in a least square sense by multiplying K^T on both sides to get $K'_{n \times n} U_{n \times 1} = F'_{n \times 1}$. As a result, the solution vector is computed by minimizing the residual error ($E = K' U - F'$). This approach is similar to the LS approximation.

The final system of equations is solved by the LS approach as discussed earlier, and the values of the nodal parameters at the field nodes are then computed. The fixed RKPM interpolation function does not possess the delta function property. As such, $u_I \neq f^h(x_I, y_I)$, namely the value of the nodal parameter u_I at the I^{th} field node is not equal to the value of the approximate function $f^h(x_I, y_I)$. Therefore, a procedure is discussed in Section 2.4.2 to correctly impose the Dirichlet boundary conditions in the RDQ method. However, for the field nodes that are not located on the Dirichlet boundary, $u_I \neq f^h(x_I, y_I)$ is still true. Therefore, a second level of interpolation is performed, and the value of approximate function at each field node is computed by interpolating each field node by the surrounding field nodes with the fixed RKPM interpolation function as

$$f^h(x_i, y_j) = \sum_{I=1}^{NP} N_I(x_i, y_j) u_I \quad (2.41)$$

where $f^h(x_i, y_j)$ is an approximate value of the function at the field node (x_i, y_j) , NP is the total field nodes surrounding the field node (x_i, y_j) , such that $NP \subseteq N$, where N is the total field nodes in the domain, and N_I and u_I are the values of shape function and nodal parameter at the I^{th} field node, respectively.

2.4.5 Computing the approximate derivatives of field variables

Two approaches are employed in the RDQ method to compute the approximate derivatives of the field variables. The first one is the diffused derivative of a standard MLS approximation [4], and the second is the weighted derivatives approach [80].

2.4.5.1 Diffused derivative approach

Approximate derivatives at the field nodes are computed after the evaluation of approximate function values by Eq. (2.41). Two approaches are studied to compute the values of approximate derivatives at the field nodes. The first approach is by taking the derivative of a shape function in Eq. (2.41) given as

$$\frac{df^h(x_i, y_j)}{dx} = \sum_{I=1}^{NP} \frac{dN_I(x_i, y_j)}{dx} u_I \quad (2.42)$$

$$\frac{d^2 f^h(x_i, y_j)}{dx^2} = \sum_{I=1}^{NP} \frac{d^2 N_I(x_i, y_j)}{dx^2} u_I \quad (2.43)$$

This approach is similar to a diffused derivative of standard MLS approximation [4]. It is seen from Eq. (2.16) that the only variable in the equation of shape function is an array $b_I(x, y)$. As a result, an appropriate derivative of the array $b_I(x, y)$ is taken while computing the derivative of shape function. If the maximum complete 2nd-order of monomial is considered in the shape function, then

$$N_{I,x}(x, y) = [0 \quad 1 \quad 0 \quad 2x \quad y \quad 0] C_{II}^{-1} \quad (2.44)$$

$$N_{I,y}(x, y) = [0 \quad 0 \quad 1 \quad 0 \quad x \quad 2y] C_{II}^{-1} \quad (2.45)$$

$$N_{I,xx}(x, y) = [0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0] C_{II}^{-1} \quad (2.46)$$

$$N_{I,yy}(x, y) = [0 \ 0 \ 0 \ 0 \ 0 \ 2] C_{II}^{-1} \quad (2.47)$$

$$N_{I,xy}(x, y) = [0 \ 0 \ 0 \ 0 \ 1 \ 0] C_{II}^{-1} \quad (2.48)$$

As a result, the appropriate equations from Eqns. (2.44) to (2.48) are substituted in Eqns. (2.42) and (2.43) to compute the values of approximate function derivatives at the field nodes.

It is observed from the convergence studies that this approach does not give correct results if the order of monomial included in the shape function basis is less than the actual order of the field variable distribution. As a result, second approach called the weighted derivatives approach is developed.

2.4.5.2 Weighted derivative approach

This approach is formulated by computing the weighted derivatives at a concerned field node x_k by the weighted sum of derivative values with the field nodes located in the local domain of the concerned node x_k . For example, the 1st-order derivative at the node x_k in the x direction is given by $f(x_k)_{,x} = \sum_{i=1}^{NP} N_i(x_k) f(x_i)_{,x}$, where NP is the number of total field nodes in the local domain of node x_k , and $N_i(x_k)$ and $f(x_i)_{,x} = \{[f(x_i) - f(x_k)] / (x_i - x_k)\}$ are the values of shape function and approximate derivative at the i^{th} field node, respectively. The values of shape function are computed by the fixed RKPM interpolation function. Similarly, the 2nd-order derivative at the field node x_k in the x direction is computed by $f(x_k)_{,xx} = \sum_{i=1}^{NP} N_i(x_k) f(x_i)_{,xx}$, where $N_i(x_k)$ and $f(x_i)_{,xx} = \{[f(x_i)_{,x} - f(x_k)_{,x}] / (x_i - x_k)\}$ are the values of shape function and derivative at the i^{th} node, respectively. The derivatives in the y direction are similarly computed.

It is observed after solving several test problems that the weighted derivatives approach gives considerably good results as compared with the diffused derivative approach. As a result, it is adopted while computing the convergence rates of the RDQ method. It is also observed that the values of the approximate derivatives computed by

the weighted derivative approach are more accurate for the field nodes distributed uniformly as compared with the field nodes distributed randomly. This is because the randomness of the field nodes is not handled in the weighted derivative approach. Therefore, the improved weighted derivatives approach is developed to increase the accuracy of the weighted derivative approach for the field nodes distributed randomly.

2.4.5.3 Improved weighted derivative approach

A new method called the improved weighted derivative approach is developed in this subsection to improve the accuracy of weighted derivative approach for the field nodes distributed randomly in the computational domain.

Let us consider the i^{th} concerned field node with the NP field nodes distributed randomly in the interpolation domain, and the field node j be one of the NP nodes such that $j \in [1, NP]$. The relative position of the i to j nodes is shown in Fig. 2.2, the Euclidean length between them is computed by $S = \sqrt{(\Delta x)^2 + (\Delta y)^2}$, and the gradient between them in the direction S is given as

$$\left(\frac{\partial f}{\partial S} \right)_j = \frac{f_j - f_i}{S} \quad (2.49)$$

Fig. 2.2. Relative positioning of field nodes i and j .

Therefore, the gradient in the y direction is given as

$$\left(\frac{\partial f}{\partial y}\right)_j = \left(\frac{\partial f}{\partial S}\right)_j \cos(\theta) \Rightarrow \left\{ -\frac{\cos(\theta)}{S} \quad \frac{\cos(\theta)}{S} \right\} \begin{Bmatrix} f_i \\ f^j \end{Bmatrix} \quad (2.50)$$

All the individual values of $(\partial f / \partial y)_j$ are computed, depending on the relative location of the field nodes i and j . The $(\partial f / \partial y)_i$ is thus given as

$$\left(\frac{\partial f}{\partial y}\right)_i = \sum_{j=1}^{NP} N_j \left(\frac{\partial f}{\partial y}\right)_j \quad (2.51)$$

Similarly, any equation of the approximate derivative can be developed in the x or y direction. It is observed from Eq. (2.51) that the shape function N_j smoothens the solution, and the angle θ captures the variation of the gradient due to the random location of the field nodes.

It is noted later that the accuracy of the improved weighted derivative approach is better than the weighted derivative approach.

2.5 Summary

At first, the formulations of the fixed RKPM and DQ methods are discussed in this chapter, and the principle of RDQ method is explained. The stiffness matrix in the RDQ method is developed by discretizing the governing differential equation at the virtual nodes and imposing the appropriate boundary conditions. The final system of equations is computed by the LS approach, as the stiffness matrix may be rectangular. Two approaches are discussed to evaluate the approximate derivatives of the function. Out of them, the weighted derivatives approach is found to give more accurate solution. As a result, it is adopted in the RDQ method. In order to correctly compute the approximate derivatives of the function for the field nodes distributed randomly, a novel approach called the improved weighted derivative is proposed. It is observed from the solution of the test problems that the approach of improved weighted derivative correctly captures the gradients for the field nodes distributed randomly.

Further, the final system of equations is solved by the direct or iterative methods. Therefore, the Gauss elimination with a partial pivoting is implemented as a direct method, and portable extensible toolkit for the scientific computing (PETSC) is

implemented as an iterative method. The PETSC is a free library with different iterative solvers with the major advantages, as given below.

- It works with only the nonzero elements of the stiffness matrix, which drastically reduces the time of computation,
- it can be run in a parallel mode,
- it uses the LU pre-conditioner by default and simplifies the sparse stiffness matrix.

This is a good practice to get the converged solution even for big matrices with less computational time. In general, the convergence of iterative method degrades with an increase in the condition number of matrix. Therefore, the pre-conditioner reduces the condition number of the matrix there by increasing the accuracy of results.

It is observed that the computational time taken by PETSC is considerably lower than the Gauss elimination, also the solution obtained by PETSC matches exactly with that by the Gauss elimination for most of the problems.

The complete implementation of the RDQ method is broadly summarized in the following flowchart.

Chapter 3

Convergence and consistency analyses of RDQ method and its application for solving the microelectromechanical systems

3.1 Introduction to the convergence analysis

The DQ method is one of the efficient techniques for derivative approximation, but it always requires a regular domain with all the points distributed along the straight lines only. This severely restricts the DQ, while solving the problems with irregular domain discretized by the random field nodes. This limitation of the DQ method is overcome in the presently proposed novel strong-form RDQ method. The RDQ method extends the applicability of the DQ technique over the irregular or regular domains discretized by either the random or uniform field nodes. This is achieved by approximating a value of function with the fixed RKPM method, and discretizing a governing differential equation by the locally applied DQ method. A superconvergence condition for the RDQ method is developed in this chapter, which gives the convergence rate of function more than $O(h^{p+1})$ for the uniform and random field nodes scattered in the domain, where p is the highest order of the monomials used in the approximation of function. Finally, the convergence analysis of RDQ method is performed, and the superconvergence condition is verified by solving several 1-D, 2-D, and elasticity problems. The applicability of the RDQ method in solving the nonlinear governing differential equations is successfully demonstrated by the fixed-fixed and cantilever microswitches under the nonlinear electrostatic load. It is concluded that the RDQ method effectively handles the irregular and regular domains discretized by either the uniform or random field nodes with good rates of the convergence.

The main objectives of this chapter are to perform the convergence and consistency analyses of the RDQ method, with its application for the analysis of the MEMS switches. The superconvergence condition is developed during the study of convergence analysis, which always give the convergence of function greater than $O(h^{p+1})$, where p is the

highest order of the monomials used in the approximation of function. For the test problems solved in this chapter by the superconvergence condition, the convergence rates of $O(h^{p+\alpha})$, where $\alpha \geq 1$ for the function approximation and $\alpha \approx 0.7$ to 1 for the derivative approximation, are obtained for the uniform as well as randomly distributed field nodes.

The merit of the RDQ method as compared with the existing strong-form meshless methods is that the fixed RKPM function is used only to approximate the values of function at virtual nodes, and not to approximate the derivative terms from the governing PDE. While, the derivative terms from the governing PDE are approximated by the locally applied DQ method. As a result, the accuracy of the derivative approximation in the RDQ method is independent of the order of monomial used in the fixed RKPM interpolation. This differentiates the RDQ method from the collocation-based strong-form methods, in which the derivative terms from the governing PDE are approximated by the derivatives of the shape function. As compared with other weak-form methods, the RDQ method is capable of well capturing the local high gradients, which will be shown during the convergence analysis.

Compared with the classical FEM, the merit of RDQ method is that it can effectively handle the problems with moving boundary, which is difficult for the FEM due to the problems of mesh distortion and singularities in elements. The field nodes act as Lagrangian grid and the virtual nodes act as Eulerian grid in the RDQ method. As a result, the field nodes are free to move anywhere in the domain, and no interconnectivity information among them is required, while the virtual nodes are fixed in a space so that it is always possible to discretize the governing PDE over the virtual nodes by locally applying the DQ method.

In the subsequent portions of this chapter, the superconvergence condition is derived and several test problems are solved by the RDQ method. The microswitches of the type fixed-fixed and cantilever are then solved by the RDQ method.

3.2 Development of the superconvergence condition

Let us consider N_v and N_r as the numbers of total virtual and field nodes, respectively, distributed in a computational domain. In order to identify the relationship

between them for a given problem, it is supposed that the two graphs are plotted, viz. $\ln(E)$ versus $\ln(h_r)$ and $\ln(E)$ versus $\ln(h_r/h_v)$, where E is a global error, and h_r and h_v are the spacings of field and virtual nodes, respectively, and let m_1 and m_2 be the slopes of these graphs, respectively. The equations of m_1 and m_2 are written as

$$\ln(E) = m_1 \ln(h_r) \text{ and } \ln(E) = m_2 \ln(h_r/h_v) \quad (3.1)$$

$$\therefore \lambda = \frac{m_1}{m_2} = \frac{\ln(h_r/h_v)}{\ln(h_r)} \quad (3.2)$$

where $\lambda = (m_1/m_2)$, $N_v \geq N_r$ for a unique solution, and $h_r = L/(N_r - 1)$ and $h_v = L/(N_v - 1)$, where L is a domain length. It is stated by observation that $\ln(h_r)$ has a zero or negative value as $h_r \leq 1$, $\ln(h_r/h_v)$ always has a zero or positive value as $h_r \geq h_v$, and $\ln(E)$ has either a positive or negative value as $E > 0$ or $E < 0$. Based on this input, four conditions are possible for the signs of m_1 , m_2 and λ . Therefore, a careful consideration of these four conditions results in only one valid possibility, as given in Table 3.1

Table 3.1. Different possibilities of m_1 , m_2 and λ signs

m_1	m_2	λ	Is valid?
+ve	+ve	+ve	No
+ve	-ve	-ve	Yes
-ve	-ve	+ve	No
-ve	+ve	-ve	No

As a result, m_1 , m_2 and λ should have +ve, -ve and -ve signs, respectively. For the different number of field and fixed number of virtual nodes during the convergence studies, m_2 is not constant but has a different value at each (h_r/h_v) location. Therefore, the 3rd graph (λ) versus (h_r) be plotted to maintain a constant value of λ with respect to the different values of N_r , where the origin of this graph is at $(h_v, 0)$, as the lowest

values of $h_r = h_v$ and $\lambda = 0$. The slope of the third graph by the origin along with any two points, (λ_1, h_{r1}) and (λ_2, h_{r2}) , on the graph is given as

$$\lambda_1 - 0 = h_{r1} - h_v \text{ and } \lambda_2 - 0 = h_{r2} - h_v \quad (3.3)$$

$$\therefore \frac{\lambda_1}{\lambda_2} = \frac{(h_{r1} - h_v)}{(h_{r2} - h_v)} \quad (3.4)$$

Eq. (3.4) is simplified for a fixed value of m_1 as

$$\frac{(m_2)_2}{(m_2)_1} = \frac{(h_{r1} - h_v)}{(h_{r2} - h_v)} \quad (3.5)$$

If the values of m_1 and λ are fixed, the successive values of m_2 can be computed, and by further fixing the values of h_v and h_{r1} , the successive values of h_{r2} can be computed by Eq. (3.5). If a test problem is solved by these successive values of h_r (corresponding to fixed value of m_1) and a fixed value of N_v , therefore m_1 convergence rate should be achieved, as per Eq. (3.5). It is concluded therefore that, if a test problem is solved with the field nodes obtained by Eq. (3.5) and a value of function is approximated with the p^{th} -order monomials, a rate of convergence higher than $O(h^{p+1})$ is possibly obtained. Therefore, Eq. (3.5) is called the superconvergence condition. The application of Eq. (3.5) is demonstrated in the next section by solving several 1-D and 2-D test problems.

3.3 The convergence analysis

The convergence analysis of RDQ method is performed in this section by solving several test problems of the type 1-D, 2-D and elasticity. The convergence rates for all the presented problems are evaluated by computing the global error as [47, 81]

$$\mathcal{E} = \frac{1}{|f^e|_{\max}} \sqrt{\frac{1}{NP} \sum_{I=1}^{NP} [f_I^{(e)} - f_I^{(n)}]^2} \quad (3.6)$$

where $\mathcal{E}$ is a global error in the solution, $f_I^{(e)}$ and $f_I^{(n)}$ are the exact and numerical values of function at the I^{th} field node, respectively, and NP is the number of total field

nodes. Eq. (3.6) is same as the l^2 error norm but averaged over the total field nodes, and normalized by an absolute maximum value of exact solution.

The rate of the convergence is computed as described [82-83]. Taylor series is a mathematical representation of a continuous function as the sum of infinite terms computed by the values of derivatives at a single point as

$$f(x+h) = f(x) + h \frac{\partial f(x)}{\partial x} + \frac{h^2}{2} \frac{\partial^2 f(x)}{\partial x^2} + \dots + \frac{h^n}{n!} \frac{\partial^n f(x)}{\partial x^n} + O(h^{n+1}) \quad (3.7)$$

While computing the shape functions, if the monomials up to the p^{th} -order from the Pascal's triangle are included, the first $(p+1)$ terms from the Taylor series will be exactly reproduced. The difference between the exact f_e and numerical f_n values of function is an error that is of the order $O(h^{p+1})$, therefore the order of the rate of convergence is $O(h^{p+1})$ as

$$(f_e - f_n) = \text{error} = \frac{h^{p+1}}{(p+1)!} \left(\frac{\partial^{p+1} f}{\partial x^{p+1}} \right) \quad (3.8)$$

$$\therefore \text{the order of the rate of convergence} = O(h^{p+1}) \quad (3.9)$$

By taking log on both the sides results in

$$\log_{10}(\text{error}) = (p+1) \log_{10}(h) \Rightarrow (p+1) = \frac{\log_{10}(\text{error})}{\log_{10}(h)} \quad (3.10)$$

where $(p+1)$ is the rate of the convergence, or the rate at which the numerical solution converges to the corresponding analytical solution. This rate of the convergence is the slope of the graph $\log_{10}(\text{error})$ versus $\log_{10}(h)$. The error in Eq. (3.10) is computed by Eq. (3.6).

The field nodes are created either in the uniform or random manner during the convergence analysis, and the virtual nodes are created by the cosine distribution as

$$x_i = x_0 + \frac{L}{2} \left[1 - \cos \left(\frac{i-1}{N-1} \pi \right) \right], \text{ for } i = 1, 2, \dots, N \text{ virtual nodes} \quad (3.11)$$

where x_0 and L are the starting coordinate and domain length, respectively. According to Runge's phenomenon [76, 80], with increase in the order of the field variable interpolation by the field nodes distributed uniformly, the numerical solution gets unstable near the boundaries of domain. This can be avoided, if the nodes are distributed densely near the boundaries and progressively uniform within the domain. This requirement is fulfilled by Chebyshev nodes. The Chebyshev polynomials are one of the types of classical orthogonal polynomials. As such, the Chebyshev polynomials of the 1st-kind $[T_n(x)]$ are referred here as the orthogonal polynomials of the 1st-kind [84-85], where n is the order of polynomial. $[T_n(x)]$ are the roots of the 2nd-order Chebyshev differential equation $(1-x^2)y'' - xy' + n^2y = 0$, and they are represented by trigonometric identity as $T_n(x) = \cos[n \cos^{-1}(x)]$, where $x \in [-1, 1]$. The roots or zeros of $[T_n(x)]$ are referred as Chebyshev-Gauss (CG) points, which are evaluated as $x_k = \cos\{[\pi(2k+1)]/[2(n+1)]\}$, where $k = 0, 1, 2, \dots, n$. The CG points are generally not used to solve any PDE due to the difficulty while imposing the boundary conditions, as they do not include the values of domain boundaries $[-1, 1]$. To overcome this restriction, Chebyshev-Gauss-Lobatto (CGL) points are constructed that are also considered as the roots of $[T_n(x)]$. The CGL points are evaluated as $x_k = -\cos\{[(k-1)\pi]/(n-1)\}$, where $k = 1, 2, \dots, n$, and it is easy to impose the boundary conditions, as they cover the domain boundaries ± 1 . Eq. (3.11) is actually the equation of the computation of CGL points, but applied over the domain $[x_0, (x_0 + L)]$.

The approximate derivatives of function at the field nodes are computed by the improved weighted derivative approach [80], as explained in Chapter 2. It is observed that the percentage relative numerical error in the computed derivatives is more than that in the values of function. As such, in order to obtain a good convergence, it is essential to ensure that the error norm of derivative reduces as the field nodes are increased. Therefore, the complete or Sobolev error norm is computed as

$$(E)_0 = \sqrt{\sum_{I=1}^{N_f} (f_I^e - f_I^n)^2}, \text{ Sobolev norm of the order 0} \quad (3.12)$$

$$(E)_1 = (E)_0 + \sqrt{\sum_{I=1, j=1}^{N_r, 3} \left[\left(\frac{\partial f}{\partial x_j} \right)_I^e - \left(\frac{\partial f}{\partial x_j} \right)_I^n \right]^2}, \quad \text{Sobolev norm of the order 1} \quad (3.13)$$

where $(E)_0$ and $(E)_1$ are the Sobolev error norms of the 0th- and 1st-order, respectively, and e and n are the exact and numerical values, respectively. It is seen from Eq. (3.13) that as the square of error in the values of function and derivative is added at each field node, the contribution of the term $(E)_0$ to the term $(E)_1$ becomes less as compared with the term of derivative error. Therefore, if $(E)_1$ reduces by successively increasing the field nodes, actually the error term of derivative reduces. As a result, the Sobolev error norm computed by Eq. (3.13) is fairly taken as an indicator of the reduction in the values of derivative error at the field nodes.

3.3.1 Computation of the convergence rate for the distribution of random field nodes

It is relatively easy to compute the global error and convergence rates for the uniform distribution of field nodes, as the nodal spacing h is constant, but it is not easy for the random distribution of field nodes. In order to compute the convergence rates for the randomly distributed field nodes in the present work, the values of function at the random field nodes are first computed by the RDQ method. The solution is then approximated over the equal number of uniformly distributed points [48] by the values of function at the random field nodes. The global error for the uniformly distributed points is computed, and the convergence curves are plotted with it. For example, to compute the global error and the convergence rates for the 81 randomly distributed field nodes, the 9×9 uniform points are created, and the solution is approximated over them by the values of function at the 81 field nodes distributed randomly. The convergence curves are then plotted by the function values at the 9×9 uniform points.

3.3.2 Remarks on the effect of random nodes on the convergence rate

The random nodes are generated by a standard function provided by C++ language. This function of random number generator is based on the pseudo-random number

generator algorithm, called the Linear Congruential Generator (LCG) [86]. There is an equal or uniform probability of generating a number between some lower and maximum values, as per the LCG. Therefore, in general the LCG can be called to be based on the uniform density. It is quite common to use the LCG algorithm to generate random numbers in the computer simulation programs. A typical way to generate the pseudo-random number with the uniform probability using the LCG in a known range is to use the modulo (%) of returned value by the range span and add the initial value of the range. For example, $number = 1 + [(value\ by\ LCG) \% 50]$ generates a number between 1 and 50.

If a current time is given as a reference value to generate the random numbers in the function of random number generator, each time new set of random numbers is generated. But, if no reference value is given, the same set of random numbers is always generated. In order to maintain the reproducibility, all the results presented here are based on the random numbers generated without giving any starting reference value to the function of random number generator, therefore same results are always obtained. During the convergence analysis of RDQ method, some problems are solved by the different sets of random numbers, and it is observed that the convergence rate is affected but only slightly. It is also noted that the convergence rate obtained, by the random nodes generated without giving any starting reference value to the random number generator program, is a precise measure of the performance of RDQ method for the random field nodes. This remark is made after comparing the values of convergence rates obtained via the random nodes generated with and without giving any starting reference value to the random number generator program. This observation will be verified in the next subsection by additionally solving some test problems by the different sets of random field nodes.

It is observed by conducting an extensive numerical analysis that the accuracy of the solution is affected due to the field nodes distributed randomly, but only for the low number of random field nodes. As the number of total random field nodes is increased, there is a sharp decline in the global error, as compared with that of the equal number of uniform field nodes. This behaviour is verified from the convergence curves given in the next sections. Nevertheless, in order to make the comparison between the convergence rates obtained by the random distribution of field nodes with that of the equal number of

total uniform field nodes, all the convergence results for the random distribution of field nodes are computed with equal number of uniform points, as explained in earlier subsection. As a result, the convergence rates for the equal number of uniform and random field nodes are directly comparable.

3.3.3 1-D test problems

The 1-D test problems are solved in this section, and their convergence curves are plotted to get the convergence rates.

The first 1-D problem is a Poisson equation with a constant force term. The governing equation and boundary conditions are given as

$$\frac{d^2 f}{dx^2} = 2, \quad (0 < x < 8), \text{ and } f(x=0) = 0, f(x=8) = 64 \quad (3.14)$$

The exact solution is given as $f(x) = x^2$. This problem is solved by including up to the 2nd-order monomials in the polynomial basis of function approximation, and with 6, 21, 161, 321 field and 460 cosine virtual nodes. The convergence curves are plotted in Fig. 3.1, and it is seen that the convergence rates obtained by the random field nodes are equally good as uniform field nodes. Also, reasonably good rates of the derivative convergence are obtained by both the uniform and random field nodes. The comparison between the analytical and numerical values of function is given in Fig. 3.2.

The second 1-D problem is a mixed boundary value problem with the analytical solution containing the 4th-order monomial, therefore it is interesting to see how the RDQ method converges by including up to the 2nd-order monomials. The governing equation and boundary conditions are as

$$\frac{d^2 f}{dx^2} = \frac{105}{2}x^2 - \frac{15}{2}, \quad (-1 < x < 1), \text{ and } f(x=-1) = 1, \frac{df}{dx}(x=1) = 10 \quad (3.15)$$

The analytical solution is given as $f(x) = (35/8)x^4 - (15/4)x^2 + (3/8)$. This problem is solved by the superconvergence condition with $m_1 = 3$, $N_v = 641$, $h_{r1} = 0.05$, and $\lambda_1 = -1.2$. The successive values of λ are obtained by dividing the value of λ_1 by 3, and

by Eq. (3.5) to compute the corresponding values of h_{r2} . As such, $N_r = 58, 145, 299, 464, 569$, and 632 are obtained.

Fig. 3.1. Convergence plots by the uniform and random distributions of the field nodes combined with the cosine virtual nodes for the 1st 1-D problem of Poisson equation.

Fig. 3.2. Comparison between the numerical and analytical values of function by uniform (a) and random (b) field nodes for the 1st 1-D problem of Poisson equation.

In order to compare the results, this problem is also solved by the 2nd set of 21, 41, 81, 161, 321, 641 field and 641 cosine virtual nodes. The convergence curves obtained by both sets of the field nodes are plotted in Fig. 3.3, and the corresponding convergence rates are given in Table 3.2. It is seen from Table 3.2 that all the convergence rates obtained by the field nodes computed with Eq. (3.5) are improved, the rates of derivative convergence are superconvergent, and the convergence of function value by the random field nodes is also superconvergent. Fig. 3.4a shows the reduction in the complete

Sobolev norm with the number of field nodes obtained by the superconvergence condition.

This problem is also solved by the 4th-order monomials, and the convergence curves are given in Fig. 3.4b. When Table 3.2 and Fig. 3.4b are compared, it is observed that the rates of derivative convergence are considerably improved, and the convergence rates for the random field nodes are almost equal to or better than that of the uniform field nodes. This indicates that the RDQ method is capable of equally handling the distributions of uniform and random nodes, which is one of the objectives behind the development of RDQ method. The comparison between the analytical and numerical values of function is given in Fig. 3.5. It is seen from all the results that reasonably good convergence rates of the function and the 1st- and 2nd-order derivatives are achieved by the RDQ method with the domain discretized by either the uniform or random field nodes. It is seen from Fig. 3.4b that the RDQ method converges at the faster rate with increase in the monomial order of the function approximation.

Table 3.2. Convergence rates for the 2nd 1-D problem by the 2nd-order monomials

function	for 2 nd set (uni. nodes)	superconvergence (uniform nodes)	for 2 nd set (ran. nodes)	Superconvergence (random nodes)
f	1.7	3.0	1.8	3.8
$f_{,x}$	0.9	1.8	0.8	2.1
$f_{,xx}$	0.9	1.2	0.7	1.4

The 3rd 1-D problem is with a high local gradient, which is solved by the 2nd-order monomials. The governing equation and boundary conditions are given as

$$\frac{d^2 f}{dx^2} = -6x - \left[\left(\frac{2}{\alpha^2} \right) - 4 \left(\frac{x-\beta}{\alpha^2} \right)^2 \right] \exp \left[- \left(\frac{x-\beta}{\alpha} \right)^2 \right] \quad (0 < x < 1) \quad (3.16)$$

$$f(x=0) = \exp \left[- \left(\frac{\beta^2}{\alpha^2} \right) \right], \quad \frac{df(x=1)}{dx} = -3 - 2 \left(\frac{1-\beta}{\alpha^2} \right) \exp \left[- \left(\frac{1-\beta}{\alpha} \right)^2 \right] \quad (3.17)$$

The analytical solution is given as $f(x) = -x^3 + \exp\{-(x-\beta)/\alpha\}^2\}$. This problem is solved with the superconvergence condition by fixing $m_1 = 2$, $N_v = 641$, $h_{r1} = 0.05$, and $\lambda_1 = -1.2$.

Fig. 3.3. Convergence curves by uniform (a) and random (b) field nodes for the 2nd 1-D problem of Poisson equation, where the curves are plotted with the field nodes obtained by uniformly decreasing h and the superconvergence condition.

Fig. 3.4. Reduction in the complete error norm with the field nodes obtained by superconvergence condition (a), and convergence plots by the uniform and random field nodes with the 4th-order monomials (b) for the 2nd 1-D problem.

The successive values of λ are obtained by dividing the value of λ_1 by 2, and via Eq. (3.5) to compute the corresponding values of h_{r2} . As a result, 21, 40, 75, 133, 219, 327, 433, 517 and 572 number of total field nodes are obtained. The convergence plots are

given in Fig. 3.6, and it is noted that the convergence rate of function by the uniform field nodes is $O(h^{p+2.1})$ and all the convergence rates of the derivatives are also superconvergent. The complete or Sobolev error norm is plotted in Fig. 3.6c.

Fig. 3.5. Comparison between the numerical and analytical values of function by uniform (a) and random (b) field nodes for the 2nd 1-D problem.

3.3.4 2-D test problems

The 2-D test problems are solved in this section, and their convergence curves are plotted to get the convergence rates.

The 1st 2-D problem is a Laplace equation with Dirichlet boundary conditions as

$$\nabla^2 f = 0, \quad (0 < x < 1) \text{ and } (0 < y < 1) \quad (3.18)$$

$$f(x=0, y) = -y^3, \quad f(x=1, y) = -1 - y^3 + 3y^2 + 3y \quad (3.19)$$

$$f(x, y=0) = x^3, \quad f(x, y=1) = -1 - x^3 + 3x^2 + 3x \quad (3.20)$$

The analytical solution is given as $f(x, y) = -x^3 - y^3 + 3xy^2 + 3x^2y$. This problem is solved with the 2nd-order monomials, and 5×5 , 9×9 , 17×17 , 33×33 , 44×44 field and 44×44 cosine virtual nodes. The convergence curves are plotted in Fig. 3.7, and it is observed that better convergence rate of the function value is achieved by the random field nodes, but the convergence rates of the derivatives are remain unchanged. The comparison between the analytical and numerical values of the function by the uniform and random field nodes are given in Fig. 3.8.

Fig. 3.6. Convergence curves by uniform (a) and random (b) field nodes, and the reduction in the Sobolev error norm (c) for the 3rd 1-D problem of local high gradient.

The 2nd 2-D problem is also a Laplace equation with mixed boundary conditions as

$$\nabla^2 f = 0, \quad (0 < x < 1) \text{ and } (0 < y < 1) \quad (3.21)$$

$$f(x=0, y) = -y^3, \quad f(x=1, y) = -1 - y^3 + 3y^2 + 3y \quad (3.22)$$

$$\frac{df}{dy}(x, y=0) = 3x^2, \quad \frac{df}{dy}(x, y=1) = -3 + 6x + 3x^2 \quad (3.23)$$

The analytical solution is given as $f(x, y) = -x^3 - y^3 + 3xy^2 + 3x^2y$. This problem is solved with the 2nd-order monomials, and 5×5 , 9×9 , 17×17 , 33×33 , 41×41 , 44×44 field and 44×44 cosine virtual nodes. The convergence curves are plotted in Fig. 3.9, and it is observed that better convergence rate of the function is achieved by the random field nodes.

Fig. 3.7. Convergence curves by uniform (a) and random (b) field nodes by including up to the 2nd-order monomials for the 1st 2-D problem of Laplace equation.

Fig. 3.8. Comparison between the numerical and analytical values of function by 33×33 uniform (a) and random (b) field nodes for the 1st 2-D problem of Laplace equation.

The 3rd 2-D problem solved is of local high gradient value at the node (0.5, 0.5). The governing equation and boundary conditions are given as

$$\nabla^2 f = -6x - 6y - \left[\left(\frac{4}{\alpha^2} \right) - 4 \left(\frac{x - \beta}{\alpha^2} \right) - 4 \left(\frac{y - \beta}{\alpha^2} \right) \right] \times$$

$$\exp \left[- \left(\frac{x - \beta}{\alpha} \right)^2 - \left(\frac{y - \beta}{\alpha} \right)^2 \right], \quad (0 < x < 1) \text{ and } (0 < y < 1) \quad (3.24)$$

Fig. 3.9. Convergence plots by uniform (a) and random (b) field nodes for the 2nd 2-D problem of Laplace equation.

$$f(x=0, y) = -y^3 + \exp\left[-\left(\frac{\beta}{\alpha}\right)^2 - \left(\frac{(y-\beta)}{\alpha}\right)^2\right] \quad (3.25)$$

$$f(x=1, y) = -1 - y^3 + \exp\left[-\left(\frac{(1-\beta)}{\alpha}\right)^2 - \left(\frac{(y-\beta)}{\alpha}\right)^2\right] \quad (3.26)$$

$$\frac{df}{dy}(x, y=0) = \frac{2\beta}{\alpha^2} \exp\left[-\left(\frac{\beta}{\alpha}\right)^2 - \left(\frac{(x-\beta)}{\alpha}\right)^2\right] \quad (3.27)$$

$$\frac{df}{dy}(x, y=1) = -3 - 2\left(\frac{(1-\beta)}{\alpha^2}\right) \exp\left[-\left(\frac{(x-\beta)}{\alpha}\right)^2 - \left(\frac{(1-\beta)}{\alpha}\right)^2\right] \quad (3.28)$$

The analytical solution is given as

$f(x, y) = -x^3 - y^3 + \exp\{-[(x-\beta)/\alpha]^2 - [(y-\beta)/\alpha]^2\}$. This problem is solved with the 2nd-order monomials, and the superconvergence condition by fixing $m_1 = 2$, $N_v = 44 \times 44$, $h_{r1} = 0.25$, and $\lambda_1 = -1.2$. The successive values of λ are obtained by dividing the value of λ_1 by 2, and with Eq. (3.5) to compute the values of h_{r2} . As a result, $N_r = 5 \times 5$, 9×9 , 13×13 , and 19×19 are obtained. The convergence curves are plotted in Fig. 3.10a and Fig. 3.10b, and it is observed that the convergence rates of the

derivative are superconvergent in nature. Fig. 3.10c shows the comparison between the numerical and analytical values of the distributed field variable.

Fig. 3.10. Convergence plots by uniform (a) and random (b) field nodes, and the numerical and analytical distributions of the field variable by uniform field nodes (c) for the 3rd 2-D problem.

The 4th 2-D problem is of steady-state heat conduction in a rectangular plate domain with a heat source. The governing equation and boundary conditions are given as

$$\nabla^2 T = -2 s^2 \operatorname{sech}^2 [s(y - 0.5)] \tanh [s(y - 0.5)], (0 < x < 0.5), (0 < y < 1) \quad (3.29)$$

$$\frac{\partial f}{\partial n} = 0 \text{ along } x = 0 \text{ and } 0.5, T(y = 0) = -\tanh\left(\frac{s}{2}\right), T(y = 1) = \tanh\left(\frac{s}{2}\right) \quad (3.30)$$

Fig. 3.11. Convergence plots by uniform field nodes (a), reduction in Sobolev norm (b), and the numerical distribution of temperature within the domain (c) for the 4th 2-D problem of steady-state heat conduction.

The successive values of λ are obtained by dividing the value of λ_1 by 3, and with Eq. (3.5) to compute the corresponding values of $h_{r,2}$. As a result, $N_r = 6 \times 6, 13 \times 13, 21 \times 21, 28 \times 28$ are obtained. The convergence plots of the temperature and its gradient are given in Fig. 3.11a, and the corresponding convergence rates are 3.2 and 2.4. The reduction in the Sobolev norm with increasing the field nodes, and the comparison between the numerical and analytical values of temperature are given in Fig. 3.11b and Fig. 3.11c, respectively. It is seen from Fig. 3.11a that, good rates of the function and derivative convergences are achieved with the field nodes obtained by the superconvergence condition. The superconvergent rates of derivative confirm that the improved weighted derivative approach gives reasonably good rates of the convergence.

3.3.5 Elasticity problems

All the problems presented in this section are based on the plane-stress condition with the mechanical equilibrium equation expressed as

$$\sigma_{ij,j} + B_i = 0 \quad (3.31)$$

where σ_{ij} and B_i are the stress tensor and body force, respectively. The body force is assumed zero for the present results. The equations of 2-D strains in the plane-stress condition are given as

$$\varepsilon_{xx} = \left(\frac{1}{E}\right) [\sigma_{xx} - (\nu_0 \sigma_{yy})], \quad \varepsilon_{yy} = \left(\frac{1}{E}\right) [\sigma_{yy} - (\nu_0 \sigma_{xx})], \quad \text{and} \quad \varepsilon_{xy} = \frac{2(1+\nu_0)}{E} \sigma_{xy} \quad (3.32)$$

where ε_{xx} and ε_{yy} , and σ_{xx} and σ_{yy} are the normal strains and stresses, respectively, in the x and y directions, respectively, ε_{xy} and σ_{xy} are the shear strain and stress, respectively, and E and ν_0 are the Young's modulus and Poisson ratio, respectively. Eq. (3.32) is known as 2-D Hook's law. The geometrical equations for the plane-stress condition are given as

$$\varepsilon_{xx} = \frac{\partial u}{\partial x}, \quad \varepsilon_{yy} = \frac{\partial v}{\partial y}, \quad \text{and} \quad \gamma_{xy} = \left[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right] \quad (3.33)$$

where u and v are the displacements in the x and y directions, respectively. Substituting terms of σ_{ij} in Eq. (3.31) by Eqns. (3.32) and (3.33), and simplifying it results in

$$\frac{E}{(1-\nu_0^2)} \left[\frac{\partial^2 u}{\partial x^2} + \left(\frac{1-\nu_0}{2} \right) \frac{\partial^2 u}{\partial y^2} + \left(\frac{1+\nu_0}{2} \right) \frac{\partial^2 v}{\partial x \partial y} \right] + B_x = 0 \quad (3.34)$$

$$\frac{E}{(1-\nu_0^2)} \left[\frac{\partial^2 v}{\partial y^2} + \left(\frac{1-\nu_0}{2} \right) \frac{\partial^2 v}{\partial x^2} + \left(\frac{1+\nu_0}{2} \right) \frac{\partial^2 u}{\partial x \partial y} \right] + B_y = 0 \quad (3.35)$$

Eqns. (3.34) and (3.35) are the governing equations in the displacement form for 2-D plane-stress. The Neumann boundary conditions are given as

$$\left(\frac{E}{1-\nu_0^2} \right) \left[l \left(\frac{\partial u}{\partial x} + \nu_0 \frac{\partial v}{\partial y} \right)_s + m \left(\frac{1-\nu_0}{2} \right) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)_s \right] = \bar{X} \quad (3.36)$$

$$\left(\frac{E}{1-\nu_0^2} \right) \left[m \left(\frac{\partial v}{\partial y} + \nu_0 \frac{\partial u}{\partial x} \right)_s + l \left(\frac{1-\nu_0}{2} \right) \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)_s \right] = \bar{Y} \quad (3.37)$$

where l and m are the direction cosines of surface normal, and $\bar{X}$ and $\bar{Y}$ are the prescribed boundary values. The values of $l=0$ and $m=\pm 1$ are for a boundary parallel to x axis, and $l=\pm 1$ and $m=0$ are for a boundary parallel to y axis. The surface normals for the curved shape boundary are computed by the curve equation. In order to use Eqns. (3.34) to (3.37) for the plane-strain condition, replace $E = E / (1-\nu_0^2)$ and $\nu_0 = \nu_0 / (1-\nu_0)$ keeping the rest as same.

3.3.5.1 Cantilever beam under pure bending

A cantilever beam under the pure bending load is shown in Fig. 3.12a. The analytical solutions are derived as [87-88]

$$u = \frac{M}{E I} x y, \quad v = \left(\frac{-\nu_0 M}{2 E I} \right) y^2 - \left(\frac{M}{2 E I} \right) x^2 \quad \text{and} \quad \sigma_{xx} = \frac{M y}{I}, \quad \sigma_{yy} = 0, \quad \sigma_{xy} = 0 \quad (3.38)$$

The coordinate system for Fig. 3.12a is used as shown in [89]. The transverse and longitudinal directions are designated by the y and x axes, respectively, and the depth by z axis. The bending moment is applied about the z axis. This problem is solved to test the ability of the RDQ method to capture the 2nd-order continuity of field variable distribution in the x and y directions, as seen in Eq. (3.38). A negative bending moment M is applied at the free end of the cantilever beam, which will generate pure normal stress. It was assumed while deriving the analytical equations of the displacements that the stresses $\sigma_{yy} = 0$ and $\sigma_{xy} = 0$, and σ_{xx} is varying linearly with respect to x . When the analytical equation $\sigma_{xx} = (M y) / I$ is derived with these constraints, it coincides with the solution given by mechanics of materials [88]. That means, when the normal stress, caused by M and proportional to y , is applied at the free end, the exact solution given in Eq. (3.38) should obtain. But, if the normal stress is applied by any other manner, Eq. (3.38) will not represent the exact solution. However, the numerical solution will approach the exact solution away from the boundary, as per the Saint-Venant's principle. The intention here is to solve the problem of a cantilever beam under a pure bending for

the 2-D plane stress, rather than only the transverse deflection of a beam. In the transverse deflection of the beam, the longitudinal strain ε_x is non-zero, and the transverse strains ε_y and ε_z are equal to zero, and the change in the length of the beam is negligible as the beam deflects transversely [89]. Due to the 2-D plane stress, here both ε_x and ε_y are non-zero. Therefore, both the nodal displacements u and v exist in the x and y directions, respectively, [15, 47, 48].

As per the boundary conditions, the displacements u and v are set to zero at a node $(0,0)$ due to hinge support, and $u = 0$ at all the nodes along $(0, y)$. The stresses, σ_{xy} , σ_{yy} and σ_{xx} , are imposed as zero along all the boundaries except at $x=0$ and $x=L$, where σ_{xx} is imposed as a normal stress caused by the bending moment M , as per Eq. (3.38). The condition $dv/dx=0$ is imposed at the node $(0,0)$ to avoid the rotation of beam.

Fig. 3.12. Schematic of a cantilever beam under pure bending load (a), and the convergence plots of displacements u and v by the uniform and random field nodes coupled with the cosine virtual nodes (b).

This problem is solved for $L = 48$, $D = 12$, $M = -24000$, $\nu_0 = 0.3$, $E = 3.0 \times 10^{07}$, and 13×13 , 17×17 , 21×21 , 29×29 field and 41×41 cosine virtual nodes with up to the 1st-order monomials included in the approximation of function. The convergence rates of the displacements u and v , by separately discretizing the domain with the uniform and random field nodes, are given as 1.0 and 1.3, and 1.96 and 2.0, respectively, the

corresponding convergence plots are given in Fig. 3.12b. When this problem is solved by 9×9 uniform field and 41×41 cosine virtual nodes with the 2nd-order monomials in the approximation of function, the analytical solutions are almost exactly reproduced with the global error values for the displacements u and v are 3.53×10^{-13} and 2.54×10^{-12} , respectively, as shown in Fig. 3.13. It is also observed from Fig. 3.13 that, if the complete required order of the monomials is included in the polynomial basis of the function approximation, the field variables are almost exactly reproduced. It is noted from Eq. (3.38) and Fig. 3.12a that the lower half of the beam is expected to have +ve displacement u due to tension, and the upper half is expected to have -ve displacement v due to the compression.

Fig. 3.13. Analytical displacement u (a), numerical displacement u (b), analytical displacement v (c), and numerical displacement v (d), where the domain is discretized by 9×9 uniform field and 41×41 cosine virtual nodes and the functions are approximated by the 2nd-order monomials in the polynomial basis for a cantilever beam under pure bending load.

Overall, the distribution of u should be like a parabola (as seen from analytical equation), and the displacement v should have the maximum value near the free end and zero towards the fixed end. The displacement v is also a function of the coordinate y at a specific value of the coordinate x . As a result, the vertical distribution of v is expected. These expected behaviours in the displacements u and v are verified in Fig. 3.13.

3.3.5.2 Cantilever beam under pure shear

A cantilever beam is loaded under a pure shear, as shown in Fig. 3.14a. The displacements u and v are set to zero at the node $(0,0)$ due to the hinge support. The stresses σ_{xx} and σ_{xy} are imposed along the boundaries $x=0$ and $x=L$ according to

$$\sigma_{xx} = \frac{P(L-x)y}{I}, \quad \sigma_{xy} = \frac{-P}{2I} \left(\frac{D^2}{4} - y^2 \right) \quad \text{and} \quad \sigma_{yy} = 0 \quad (3.39)$$

The stresses $\sigma_{xy} = 0$ and $\sigma_{yy} = 0$ are imposed along the boundary $y = \pm(D/2)$. The condition $dv/dx = 0$ is imposed at the node $(0,0)$ to avoid the beam rotation. The analytical equations are derived below for the coordinate system shown in Fig. 3.14a.

Fig. 3.14. Schematic of a cantilever beam under a pure shear load (a), and the convergence plots of the displacements u and v by the uniform and random field nodes coupled with the cosine virtual nodes (b).

The distribution of stresses given in Eq. (3.39) is initially assumed, and substituting it in Eq. (3.33) results in

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} = \frac{\sigma_{xx}}{E} = \frac{P y (L-x)}{E I}, \quad \varepsilon_{yy} = \frac{-\nu_0 \varepsilon_{xx}}{E} = \frac{-\nu_0 P y (L-x)}{E I}, \text{ and } (3.40)$$

$$\gamma_{xy} = \left[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right] = \frac{-P}{2 G I} \left(\frac{D^2}{4} - y^2 \right) \quad (3.41)$$

Integrating Eq. (3.40) results in

$$u = \frac{P y x}{E I} \left(L - \frac{x}{2} \right) + f(y) \quad \text{and} \quad v = \frac{-\nu_0 P y^2 (L-x)}{2 E I} + f_1(x) \quad (3.42)$$

where $f(y)$ and $f_1(x)$ are the integral constants. Differentiate the displacements u and v in Eq. (3.42) with respect to y and x , respectively, and substitute in Eq. (3.41) to get

$$f_1(x) = d x - \frac{P}{E I} \left(\frac{L x^2}{2} - \frac{x^3}{6} \right) + h \quad \text{and} \quad f(y) = e y - \frac{\nu_0 P y^3}{6 E I} + \frac{P y^3}{6 G I} + g \quad (3.43)$$

where d and e are the functions of x and y , respectively. Substituting Eq. (3.43) in Eq. (3.42) gives

$$u = \frac{P y x}{E I} \left(L - \frac{x}{2} \right) + e y - \frac{\nu_0 P y^3}{6 E I} + \frac{P y^3}{6 G I} + g \quad (3.44)$$

$$v = \frac{-\nu_0 P y^2 (L-x)}{2 E I} + d x - \frac{P}{E I} \left(\frac{L x^2}{2} - \frac{x^3}{6} \right) + h \quad (3.45)$$

Imposing the boundary conditions in Eqns. (3.44) and (3.45) as $u(0,0) = 0$, $v(0,0) = 0$ and $(\partial v / \partial x)_{(0,0)} = 0$ results in $g = h = d = 0$ and $e = (-P D^2) / (8 G I)$. Therefore, Eqns. (3.44) and (3.45) are simplified as

$$u = \frac{P y x}{E I} \left(L - \frac{x}{2} \right) - \frac{\nu_0 P y^3}{6 E I} + \frac{P y^3}{6 G I} - \frac{P D^2 y}{8 G I} \quad (3.46)$$

$$v = \frac{-\nu_0 P y^2 (L-x)}{2 E I} - \frac{P x^2}{E I} \left(\frac{L}{2} - \frac{x}{6} \right) \quad (3.47)$$

where $G = E / [2(1+\nu_0)]$ and $I = (D^3 / 12)$ (beam has unit thickness).

Fig. 3.15. Comparison of the displacements u (a) and v (b) by the 3rd-order monomials in the polynomial basis of function approximation for a cantilever beam under pure shear load.

Fig. 3.16. Analytical σ_{xx} (a), numerical σ_{xx} (b), analytical σ_{xy} (c) and numerical σ_{xy} (d) stresses, where the domain is discretized by 9×9 uniform field and 41×41 cosine virtual nodes for a cantilever beam under a pure shear load.

As a result, the analytical expressions given in Eqns. (3.46) and (3.47) correspond to the distributions of the stresses given in Eq. (3.39), and it is noted that the displacements u and v are 3rd-order continuous in the space domain.

This problem is solved for $L = 48$, $D = 12$, $P = -1000$, $\nu_0 = 0.3$, $E = 3.0 \times 10^{07}$, and 13×13 , 17×17 , 21×21 , 29×29 field and 41×41 cosine virtual nodes with the 2nd-order monomials included in the monomial basis of shape function while approximating the function. The convergence rates of the displacements u and v by the uniform and random field nodes are 1.94 and 1.9, and 1.5 and 2.0, respectively, and the corresponding convergence curves are plotted in Fig. 3.14b. When this problem is solved by 9×9 field and 41×41 virtual nodes with the 1st-, 2nd- and 3rd-order monomials, the numerical results are steadily improved, as shown in Table 3.3. The plots in Fig. 3.15 and Fig. 3.16 show the comparison between the numerical and analytical values of the displacement and stresses, respectively, when the numerical results are obtained by including up to the 3rd-order monomials in the polynomial basis of shape function.

Table 3.3. Decrease in the values of global error with increase in the monomial order

Monomial order	Global error in u	Global error in v
1	1.38×10^{-04}	1.2
2	3.78×10^{-05}	5.03×10^{-02}
3	3.26×10^{-09}	2.43×10^{-09}

3.3.5.3 A beam under the tensile and compressive normal stresses in the x and y directions, respectively

A beam is loaded with normal stresses in the x and y directions, as shown in Fig. 3.17. The displacements u and v are set to zero along the edge $x = 0$ and $y = 0$, respectively. The values of stresses σ_{xx} and σ_{xy} are imposed as 1.0 and 0 along the edge $x = L$, respectively, and the values of stresses σ_{xy} and σ_{yy} are imposed as 0 and -0.5 along the edge $y = L$, respectively. The analytical expressions of the displacements u and v are given as

$$u = \frac{9}{8}x \text{ and } v = \frac{-3}{4} \left[y + \frac{h}{2} \right] \quad (3.48)$$

As the displacements u and v are 1st-order continuous, the problem is treated as a test case to study the reproducibility of the 1st-order continuous field variable by the RDQ method. This problem is solved for $L = 9$ mm and $h = 3$ mm, and by 9×9 uniform field nodes coupled with 41×41 cosine distributed virtual nodes, the global errors in the displacements u and v are obtained as 3.66×10^{-11} and 4.53×10^{-11} respectively, and the displacements and stresses are shown in Fig. 3.18. When this problem is solved by 9×9 randomly distributed field nodes coupled with 41×41 cosine distributed virtual nodes, the global errors in the displacements u and v are obtained as 1.53×10^{-11} and 4.64×10^{-12} respectively, and the displacements and stresses are shown in Fig. 3.19.

Fig. 3.17. Schematic of a beam under the loading of constant normal stresses.

3.3.5.4 Semi-infinite plate with a central hole

The RDQ method is applied to solve the problem of semi-infinite plate with a central hole, as shown in Fig. 3.20a. Only one quarter of the plate is used as the computation domain due to the axis-symmetric nature, as shown in Fig. 3.20b. The analytical solutions in Cartesian coordinate system are given as

$$u = \left(\frac{1+\nu_0}{E} \right) P \left[\frac{r}{1+\nu_0} \cos(\theta) + \left(\frac{2}{1+\nu_0} \right) \frac{b^2}{r} \cos(\theta) + \frac{b^2}{2r} \cos(3\theta) - \frac{b^4}{2r^3} \cos(3\theta) \right] \quad (3.49)$$

$$v = \left(\frac{1+\nu_0}{E} \right) P \left[\left(\frac{-\nu_0}{1+\nu_0} \right) \frac{r}{r} \sin(\theta) - \left(\frac{1-\nu_0}{1+\nu_0} \right) \frac{b^2}{r} \sin(\theta) + \frac{b^2}{2r} \sin(3\theta) - \frac{b^4}{2r^3} \sin(3\theta) \right] \quad (3.50)$$

$$\sigma_{xx} = P \left[1 - \frac{a^2}{r^2} \left(\frac{3}{2} \cos(2\theta) + \cos(4\theta) \right) + \frac{3}{2} \frac{a^4}{r^4} \cos(4\theta) \right] \quad (3.51)$$

$$\sigma_{yy} = -P \left[\frac{a^2}{r^2} \left(\frac{1}{2} \cos(2\theta) - \cos(4\theta) \right) + \frac{3}{2} \frac{a^4}{r^4} \cos(4\theta) \right] \quad (3.52)$$

$$\sigma_{xy} = -P \left[\frac{a^2}{r^2} \left(\frac{1}{2} \sin(2\theta) + \sin(4\theta) \right) - \frac{3}{2} \frac{a^4}{r^4} \sin(4\theta) \right] \quad (3.53)$$

Fig. 3.18. The displacements u (a) and v (b), and stresses σ_{xx} (c) and σ_{yy} (d) when the beam under the normal stresses is solved by uniform field nodes coupled with cosine virtual nodes.

where r and θ are the local polar coordinates. The conditions, $u = 0$ and $\sigma_{xy} = 0$, are imposed along the edge 1 due to the symmetric boundary condition, and all the traction components are equal to zero along the edge 2, therefore

$$\sigma'_{xx} = n_x t_x + n_y t_y = 0 \text{ and } \sigma'_{xy} = -n_y t_x + n_x t_y = 0 \quad (3.54)$$

where $t_x = n_x \sigma_{xx} + n_y \sigma_{xy}$ and $t_y = n_x \sigma_{yx} + n_y \sigma_{yy}$, n_x and n_y are the direction cosines in the x and y directions, respectively. The conditions, $v = 0$ and $\sigma_{xy} = 0$, are imposed along the edge 3 due to the symmetric boundary condition. The stresses σ_{xx} and σ_{xy} are imposed along the edge 4 by Eq. (3.51) and Eq. (3.53), respectively. The stresses σ_{yy} and σ_{yx} are imposed along the edge 5 by Eq. (3.52) and Eq. (3.53), respectively. The problem is solved for $a = 1$, $b = 5$, $P = 1$, $\nu_0 = 0.3$, $E = 1000$, and 6×6 , 11×11 , 21×21 , 31×31 uniform field and 34×34 cosine virtual nodes with the 2nd-order monomials in the polynomial basis of shape function computation.

Fig. 3.19. The displacements u (a) and v (b), and stresses σ_{xx} (c) and σ_{yy} (d) when the beam under the normal stresses is solved by random field nodes coupled with cosine virtual nodes.

The convergence rates of the displacements u and v by the uniform field nodes are obtained as 0.3, 0.3, respectively, and the convergence curves are plotted in Fig. 3.21a. The numerical and analytical values of σ_{xx} along the boundary 1 are given in Fig. 3.21b.

Fig. 3.20. Schematic of the semi-infinite plate with a central hole (a), and actual computational domain with the boundary and loading conditions (b).

Fig. 3.21. Convergence plots of the displacements u and v by uniform field nodes (a), and comparison between the numerical and analytical values of σ_{xx} along the boundary 1 (b) for a problem of semi-infinite plate with a central hole.

It is observed from all the test problems that the RDQ method gives convincingly good rates of convergence for the function as well as derivatives. The convergence rates obtained by discretizing the domain with either the uniform or random field nodes are almost equal. In fact, sometimes the results obtained by the random field nodes are better than the uniform field nodes. If the number of total field nodes is computed by the superconvergence condition, the function and its approximate derivatives converge at the rate of $O(h^{p+\alpha})$, where $\alpha \geq 1$ for the approximated function and $\alpha \approx 0.7$ to 1 for the approximated derivative. The convergence analysis also shows that, if the complete and

consistent order of the monomials is included in the approximation of function, the function and derivatives are almost exactly reproduced.

In the next section, the RDQ method is applied to analyze the MEMS devices of fixed-fixed and cantilever microswitch for the pull-in instability, caused by the nonlinear electrostatic force between the fixed bottom plate and beam.

3.4 Application of RDQ method for solving the fixed-fixed and cantilever microswitches under the nonlinear electrostatic loading

Several types of MEMS devices were developed in the last decade for the various areas of applications. They can be broadly categorised based on the actuation systems, viz. elastic, thermal-elastic, electrostatic-elastic, magnetic-elastic, and microfluidics devices. At early stage, the MEMS devices chiefly employed the mechanical deformation or electrical potential as the actuation mechanisms. Today's MEMS devices, especially BioMEMS, employ various actuating mechanisms mostly from the transport phenomenon, such as the difference in the ionic concentration, pH, and temperature, between the environmental solution and MEMS, resulting in the ionic diffusion from the solution to MEMS. If the FEM is used to simulate the diffusion phenomenon coupled with large deformation, sometimes it may lead to the distorted elements, as the diffusion of particles may result in the large deformation. However, it is relatively easy for the particle-based meshless methods to simulate this diffusion phenomenon, as they do not involve any mesh, and no connectivity information between the particles is required, such that the free movement of particles is allowed. Currently most of the MEMS problems are solved as the multiphysics problems involving the governing differential equations from the different engineering areas, such as the chemical, mechanical, and electrical fields. If a meshless method is employed to solve these multiphysics problems, the user does not need to worry about the quality and correctness of the mesh, which may demand a considerable attention, and the user can better focus on the physics of the problem. Therefore, as the actuation mechanism gets complicated, it becomes exceedingly important to go beyond the traditional techniques of analysis, such as the FEM and FDM, and develop more sophisticated methods of simulation. Out of this motivation, the RDQ method is applied to analyze the pull-in instability in two typical configurations of the

microswitch viz. fixed-fixed and cantilever beam, as shown in Fig. 3.22. The beam in the microswitch deflects toward the fixed electrode due to the applied voltage V , whereas the deflection of beam can be controlled by the applied voltage. For a certain value of voltage V , the peak deflection w of the beam is equal to the initial gap g_0 . This behaviour is called the pull-in instability, and the corresponding value of voltage is defined as the pull-in voltage.

The RDQ method is tested in this section for the reproducibility of the 4th-order field variable by solving the microswitches of the type fixed-fixed and cantilever beam, as shown in Fig. 3.22a and Fig. 3.22b, respectively. The beam configurations are firstly solved for the values of slope and deflection under the uniformly distributed loads (UDL) by the thin beam theory [89]. The same beam configurations are then solved by applying the nonlinear electrostatic force field. The governing equation of the thin beam based on the thin beam theory [87, 89] is given as

Fig. 3.22. Schematics of the fixed-fixed (a) and cantilever (b) microswitches.

$$E I \frac{d^4 w(x)}{dx^4} = q(x) \quad (3.55)$$

where $q(x)$ is an applied load, $E I$ is Flexural rigidity, and w is the beam deflection.

The fixed-fixed beam under the UDL is solved with the boundary conditions as

$$w(x=0) = 0, \quad w(x=L) = 0, \quad \frac{dw(x=0)}{dx} = 0, \quad \text{and} \quad \frac{dw(x=L)}{dx} = 0 \quad (3.56)$$

The beam parameters are considered as $q = -3.0 \text{ N}$, $E = 3 \times 10^7 \text{ Pa}$, $\nu_0 = 0.3$, $length(L) = 20 \text{ m}$, $thickness(t) = 0.1 \text{ m}$, and $width(D) = 1.0 \text{ m}$. The analytical solutions of the deflection and slope are respectively given as

$$w(x) = \frac{q x^2 (L-x)^2}{24 EI}, \quad \text{and} \quad \frac{dw(x)}{dx} = \left(\frac{q}{EI} \right) \left[\frac{x L^2}{12} - \frac{L x^2}{4} + \frac{x^3}{6} \right] \quad (3.57)$$

The problem is solved by the 41 uniform and random field nodes separately combined with the 41 cosine virtual nodes. The numerical solutions are plotted in Fig. 3.23, and the numerical results are almost exactly matching with the corresponding analytical values. The nonlinear electrostatic force field is then applied on the fixed-fixed beam as

$$q(x) = \left(\frac{\epsilon_0 \tilde{v}^2 \tilde{w}}{2 g^2} \right) \left[1 + 0.65 \left(\frac{g}{\tilde{w}} \right) \right] \quad (3.58)$$

where ϵ_0 , $\tilde{v}$, and $\tilde{w}$ are the vacuum permittivity ($8.8541878176 \times 10^{-12} \text{ F/m}$), applied voltage, and beam width, respectively, and $g = g_0 - w(x)$, where g_0 is an initial gap between the beam and bottom fixed plate. The parameters of beam are $E = 169 \text{ GPa}$, $\nu_0 = 0.3$, $L = 80 \mu\text{m}$, $t = 0.5 \mu\text{m}$ and $\tilde{w} = 10 \mu\text{m}$. As the beam starts deflecting due to the applied electrostatic force given by Eq. (3.58), the load becomes increasingly nonlinear.

Fig. 3.23. Comparison between the numerical and analytical values of deflection and slope by the uniform (a) and random (b) field nodes for a fixed-fixed beam under the UDL.

As a result, it becomes an implicit problem that needs to be solved by inner iterations with a relaxation technique. Different values of the voltage are applied to study the pull-in behaviour, and the results are plotted in Fig. 3.24. It is seen from Fig. 3.24b that the beam touches the bottom electrode for $V = 15.05$ volt. This value is closely matching with the literature value of 15.07 volt [90-91]. The convergence of the peak deflection value during inner iterations by $V = 15.05$ volt is shown in Fig. 3.24c. It is seen from Fig. 3.24b that the values of the field node deflection are smoothly captured, which demonstrate that the RDQ method can effectively handle the problems of nonlinear deformation. The inner iterations are performed by the fixed point and Newton methods to compare the results, and both are found to give similar results.

Fig. 3.24. Peak deflection of beam for different applied voltages (a), full deflection of beam for different applied voltages (b), and convergence of peak deflection with the inner iterations at $V = 15.05$ volt (c) for a fixed-fixed microswitch under the influence of nonlinear electrostatic force.

A cantilever beam microswitch, as shown in Fig. 3.22b, under the UDL is solved with the boundary conditions as

$$w(x=0)=0, \frac{dw(x=0)}{dx}=0, \frac{d^2w(x=L)}{dx^2}=0, \text{ and } \frac{d^3w(x=L)}{dx^3}=0 \quad (3.59)$$

All parameters of beam are same as the fixed-fixed beam with the analytical solutions as

$$w(x)=\frac{q}{24 EI}\left[x^4-4x^3L+6x^2L^2\right], \text{ and } \frac{dw(x)}{dx}=\frac{q}{EI}\left[\frac{x^3}{6}-\frac{Lx^2}{2}+\frac{x^2L}{2}\right] \quad (3.60)$$

The problem is solved by the 41 uniform and random field nodes separately combined with the 41 cosine virtual nodes, and the numerical solutions are plotted in Fig. 3.25. The nonlinear electrostatic force is then applied on the cantilever beam, as given in Eq. (3.58). The beam parameters are $E = 169 \text{ GPa}$, $\nu_0 = 0.3$, $L = 80 \mu\text{m}$, $t = 0.5 \mu\text{m}$, and $\tilde{w} = 10 \mu\text{m}$. The deflections of the beam by the different values of applied voltages are plotted in Fig. 3.26a, and it is observed that the pull-in instability occurs at $V = 2.5286 \text{ volt}$, which closely matches with the value 2.33 volt found [90-91].

Fig. 3.25. Comparison between the numerical and analytical values of deflection and slope by the uniform (a) and random (b) field nodes for a cantilever beam under the UDL.

As such, there should be only one equation of discretization per node for the technique of Newton iteration, but it is seen from Eq. (3.59) that the two boundary conditions are imposed at each of the nodes located at the leftmost and rightmost

boundaries of the domain. Therefore, this difficulty is overcome by transferring one of the boundary conditions to their neighbouring nodes as

$$\begin{aligned} \left(\frac{dw}{dx} \right)_1 &= \frac{(w_2 - w_1)}{(x_2 - x_1)} = 0, \quad \text{as } w_1 = 0, \text{ so } w_2 = 0, \text{ and} \\ w_{,xxx}|_N &= \frac{w_{,xx}|_N - w_{,xx}|_{N-1}}{(x_N - x_{N-1})} = 0, \text{ as } w_{,xx}|_N = 0 \therefore w_{,xx}|_{N-1} = 0 \end{aligned} \quad (3.61)$$

where w_1 and w_2 , and x_1 and x_2 are the values of deflection and location at the 1st and 2nd domain nodes, respectively, and N is the number of total field nodes. Therefore, the modified boundary conditions are given as

$$w_1 = 0, \quad w_2 = 0, \quad \left(\frac{d^2 w}{dx^2} \right)_N = 0, \text{ and } \left(\frac{d^2 w}{dx^2} \right)_{N-1} = 0 \quad (3.62)$$

The Newton method is employed by Eq. (3.62) with Eqns. (3.55) and (3.58) imposed at the virtual nodes with the residual $R = KU - F$ as a function. The values of the virtual node deflection at the n^{th} iteration, where $n > 1$, are computed by the values of the deflections at the 0th and 1st iterations as

$$J \left[\frac{\partial R(w_{vir}^1)}{\partial w} \right] [w_{vir}^2 - w_{vir}^1] = -R(w_{vir}^1) \text{ applied at the } j^{\text{th}} \text{ virtual node} \quad (3.63)$$

$$J \left[\frac{\partial R(w_{vir}^1)}{\partial w} \right]_j = \frac{k_j U^1 - k_j U^0}{w_j^1 - w_j^0} - \frac{\partial F(w^1)}{\partial w} \Big|_j, \quad \frac{\partial F(w^1)}{\partial w} \Big|_j = \frac{F(w_j^1) - F(w_j^0)}{w_j^1 - w_j^0}, \text{ and}$$

$$R(w^1)_j = K_j U^1 - F(w_j^1) \quad (3.64)$$

where $[\partial R(w_{vir}^1)/\partial w]_j$ is the Jacobean matrix computed at the j^{th} virtual node by the residual function R . Eqns. (3.63) and (3.64) are together iteratively solved till the peak deflection value of the field node located at the free end converge. The beam deflection is computed by Newton method for the given beam parameters and $\tilde{v} = 2.64453$ volt, and plotted in Fig. 3.26b. Fig. 3.26c shows the convergence of the peak deflection value during the inner iterations. It is observed in Fig. 3.26a that for the pull-in instability voltage $V = 2.5286$ volt, the beam gets unstable and the Neumann boundary conditions

are not exactly imposed. But, this behaviour is rectified and the Neumann boundary conditions are exactly imposed by the modified boundary conditions given in Eq. (3.62) for $\tilde{v} = 2.64453$ volt, as evident from Fig. 3.26b. It is seen in Fig. 3.26c that the values of the peak beam deflection are smoothly converged, therefore it can be said that the modified boundary conditions given in Eq. (3.62) leads to the stable solution.

Fig. 3.26. Deflection of beam under different applied voltages (a), deflection of beam at $\tilde{v} = 2.64453$ volt by modified boundary conditions (b) and the convergence of peak deflection during the inner iterations by Newton method (c).

In summary, the RDQ method is applied in this section to study the phenomenon of pull-in instability in the fixed-fixed and cantilever microswitches. It is noted from the simulation results that the pull-in voltages obtained by the RDQ method are closely

matching with the literature values, which are based on the experiments. It can also be said that the RDQ method handles the nonlinear electrostatic force field well.

It is seen from Fig. 3.24b and Fig. 3.26b that the peak deflection equal to $7\mu\text{m}$ of the beam is about 9% of the total length equal to $80\mu\text{m}$ of the beam, which is a case of large deflection. Therefore, the linear structural model, as given by Eq. (3.55), may not be sufficient to correctly solve this problem. However, it is adequate to correctly capture the pull-in voltages, since no transient analysis is performed. Therefore, the re-meshing of the geometry is not required. Nonlinear structural model is necessary for the transient analysis of the same problems. It is also noted that the main objective here is to test the ability of the RDQ method in correctly capturing the pull-in voltages in spite of the linear structural model, which is fairly achieved.

3.5 Introduction to the consistency analysis of RDQ method

The objective of the presented work is to perform the detailed consistency analysis of the locally applied DQ method, and verify the results by the convergence analysis of the RDQ method. The performance of the RDQ method is systematically studied by discretizing the domain with the uniformly or randomly distributed field nodes coupled with the cosine or uniform distribution of the virtual nodes. The convergence analysis of RDQ method was performed in the previous section by considering only the cosine distribution of virtual nodes, whereas it is performed in the present section by considering the cosine as well as the uniform distributions of the virtual nodes, and their performance is compared by solving several 1-D, 2-D and elasticity problems.

The main difference between the DQ and RDQ methods is that the derivative at a specific node in the former method is approximated by all the nodes in a domain, whereas in the latter case, it is approximated by the surrounding nodes in a local domain associated with the concerned node. Therefore, the consistency equations obtained after the derivative discretization will be different in both the methods. Another important difference is that in the RDQ method, the terms of the function values at the virtual nodes in the expression of derivative approximation are replaced by their expression of interpolation correlating with the values of nodal parameter at the field nodes obtained by the function of fixed RKPM interpolation. As a result, it is important to investigate the

effect of this substitution on the overall consistency of the RDQ method. The terms of the virtual node function values coming from the discretized governing equation are replaced by Taylor series expansion in the present consistency analysis to get a reasonable judgment of an error in the analytical discretization. Therefore, with reference to the final consistency equations obtained after using Taylor series expansion, the present analysis represents the consistency analysis for the DQ method applied over a local domain or the RDQ method.

A problem of semi-infinite plate with a central hole is solved here to demonstrate that the RDQ method can handle the problems of irregular boundary. The purpose of this demonstration is to verify the applicability of RDQ method to solve the problems of irregular boundary. Only thing that will change with the different types of irregular boundaries is how the virtual nodes are created within domain. The RDQ method is directly applied after this as it is independent of the type or domain of the problem.

3.6 Consistency analysis of the locally applied DQ method

Given a PDE $Pu = f$ and a scheme of derivative discretization $P(h_x, h_y)v = f$, the scheme of derivative discretization becomes consistent with the PDE if any smooth function $\phi(x, y)$ satisfies the condition

$$P\phi - P(h_x, h_y)\phi \rightarrow 0, \text{ as } h_x, h_y \rightarrow 0 \quad (3.65)$$

where P is a differential operator, $h_x = L_x / (N_x - 1)$ and $h_y = L_y / (N_y - 1)$ are the grid spacings, L_x and L_y are the domain lengths, and N_x and N_y are the numbers of total field nodes in the x and y directions, respectively. The $h_x = h$ and $h_y = t$ are the grid spacings in the space (x) and time (T) domains, respectively, for a hyperbolic differential equation. In a nutshell, the consistency analysis determines how close the continuous form of the governing differential equation is represented by the corresponding discretized form.

The consistency studies of the locally applied DQ method are performed in this section, based on the theory [92], and by the virtual nodes with different configurations of the DQ local domains. The 1-D wave and Laplace equations are discretized to perform

the consistency analysis. There are two important differences between the DQ and RDQ methods. The first is that the RDQ method employs the fixed RKPM interpolation function to approximate the value of function at the concerned virtual node in the form of the values of nodal parameter at the field nodes, which are surrounding the concerned virtual node. The second is that the DQ method approximates the derivative at a specific node by including all the nodes in a domain, but it is performed in the RDQ method by the virtual nodes located in the surrounding region of the concerned virtual node. Therefore, the final consistency equations for both the cases would be different due to these differences. In the present work of consistency analysis, the governing equation is discretized at the concerned virtual node by the virtual nodes located in the local domains of the concerned virtual node. The terms of function values at the virtual nodes resulting from the discretized form of the governing equation are replaced by Taylor series expansion. The important task here is to numerically verify the conclusions drawn from the consistency analysis by the RDQ method. This is done while performing the convergence analysis of the RDQ method by the cosine and uniform distributions of the virtual nodes.

The weighting coefficients in this section are computed by Quan and Chang [34-35] and Shu's general approach [36-37]. The governing equation is discretized at a virtual node (x_m, t_m) by the uniform and cosine distributions of the virtual nodes with three different possibilities of the location of the concerned virtual node, viz. in-between, at the left end, and at the right end of the computational domain. It is shown for certain configurations of the DQ local domains that the Shu's approach gives consistent discretization of the governing equation. It is also shown that the governing equation that is discretized by the cosine distribution of virtual nodes converges to that of the uniform distribution of virtual nodes. All the configurations of the local domains of the DQ method are presented here for an illustration of the consistency of the DQ method only, the stability aspects of the corresponding schemes are not considered.

3.6.1 Consistency analysis of the 1-D wave equation by the uniform distribution of virtual nodes

3.6.1.1 Shu's general approach

The DQ weighting coefficients in this section are computed by the Shu's approach [36-37] with the uniform distribution of virtual nodes. A virtual node (x_m, t_m) is initially assumed as an internal domain node around which the DQ local domain is shown in Fig. 3.27a. The 1-D wave equation is given as

$$\phi_{,t} + a\phi_{,x} = 0, \quad P\phi = 0 \quad (3.66)$$

where $P = (\partial/\partial t) + a(\partial/\partial x)$ is a derivative operator, and m and n are the nodal indices for the space and time domains, respectively. Eq. (3.66) is discretized at the virtual node (x_m, t_n) to get

$$P(h, t) \phi = \frac{\partial \phi(x_m, t_n)}{\partial t} + a \frac{\partial \phi(x_m, t_n)}{\partial x} = 0 \quad (3.67)$$

where h and t are the nodal spacings along the space (x) and time (T) axes, respectively. The term of space derivative in Eq. (3.67) is discretized by the DQ method to give

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \{a_{m,m-1} \quad a_{m,m} \quad a_{m,m+1}\} \begin{Bmatrix} \phi(x_{m-1}, t_n) \\ \phi(x_m, t_n) \\ \phi(x_{m+1}, t_n) \end{Bmatrix} \quad (3.68)$$

The weighting coefficients by Shu's approach are computed as

$$a_{i,j} = \frac{1}{x_i - x_j} \prod_{k=1, k \neq i,j}^{N_x} \frac{(x_i - x_k)}{(x_j - x_k)}, \quad a_{ii} = - \sum_{j=1, j \neq i}^{N_x} a_{ij} \quad (3.69)$$

The weighting coefficients from Eq. (3.68) are computed by Eq. (3.69) and given as

$$a_{m,m-1} = \frac{-1}{2h}, \quad a_{m,m+1} = \frac{1}{2h}, \quad a_{m,m} = 0 \quad (3.70)$$

As per Taylor series expansion

$$\phi(x_{m-1}, t_n) = \phi(x_m - h) = \phi - h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} - O(h)^3 \quad (3.71)$$

$$\phi(x_{m+1}, t_n) = \phi(x_m + h) = \phi + h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} + O(h)^3 \quad (3.72)$$

where $\phi = \phi(x_m, t_n)$. Substituting Eqns. (3.70) to (3.72) into Eq. (3.68) results in

$$\frac{\partial\phi(x_m, t_n)}{\partial x} = \frac{-1}{2h} \left[\phi - h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} - O(h)^3 \right] + 0[\phi] + \frac{1}{2h} \left[\phi + h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} + O(h)^3 \right] \quad (3.73)$$

Simplify Eq. (3.73) results in

$$\frac{\partial\phi(x_m, t_n)}{\partial x} = \phi_{,x} + O(h)^2 \quad (3.74)$$

Eq. (3.74) is a final expression for the space derivative term in Eq. (3.67). The time derivative term is also similarly determined as

$$\frac{\partial\phi(x_m, t_n)}{\partial t} = \left\{ a_{n,n-1} \quad a_{n,n} \quad a_{n,n+1} \right\} \begin{Bmatrix} \phi(x_m, t_{n+1}) \\ \phi(x_m, t_n) \\ \phi(x_m, t_{n-1}) \end{Bmatrix} \quad (3.75)$$

$$a_{n,n-1} = \frac{-1}{2t}, \quad a_{n,n+1} = \frac{1}{2t}, \quad a_{n,n} = 0 \quad (3.76)$$

As per Taylor series expansion

$$\phi(x_m, t_{n-1}) = \phi(t_m - t) = \phi - t\phi_{,t} + \frac{t^2}{2}\phi_{,tt} - O(t)^3 \quad (3.77)$$

$$\phi(x_m, t_{n+1}) = \phi(t_n + t) = \phi + t\phi_{,t} + \frac{t^2}{2}\phi_{,tt} + O(t)^3 \quad (3.78)$$

where $\phi = \phi(x_m, t_n)$. Substitute Eqns. (3.76) to (3.78) into Eq. (3.75) and simplify it to

$$\frac{\partial\phi(x_m, t_n)}{\partial t} = \phi_{,t} + O(t)^2 \quad (3.79)$$

Substituting Eq. (3.74) and Eq. (3.79) into Eq. (3.67) results in

$$P(h, t)\phi = \phi_{,t} + O(t)^2 + a \left[\phi_{,x} + O(h)^2 \right] \quad (3.80)$$

Subtracting Eq. (3.80) from Eq. (3.66) gives

$$P\phi - P(h,t)\phi = \phi_{,t} + a\phi_{,x} - \phi_{,t} - O(t)^2 - a\phi_{,x} - a O(h)^2 \quad (3.81)$$

Eq. (3.81) tends to zero as h and $t \rightarrow 0$, which means that the presented discretization scheme is consistent.

The virtual node (x_m, t_n) is assumed at the left end of a domain with the DQ local domain around it is shown in Fig. 3.27b. The term of space derivative in Eq. (3.67) is discretized as per the DQ method as

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \{a_{m,m} \quad a_{m,m+1} \quad a_{m,m+2}\} \begin{Bmatrix} \phi(x_m, t_n) \\ \phi(x_{m+1}, t_n) \\ \phi(x_{m+2}, t_n) \end{Bmatrix} \quad (3.82)$$

The weighting coefficients are computed by Eq. (3.69) as

$$a_{m,m+1} = \frac{2}{h}, \quad a_{m,m+2} = \frac{-1}{2h}, \quad a_{m,m} = -\frac{3}{2h} \quad (3.83)$$

As per Taylor series

$$\phi(x_{m+1}, t_n) = \phi(x_m + h) = \phi + h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} - O(h)^3 \quad (3.84)$$

$$\phi(x_{m+2}, t_n) = \phi(x_m + 2h) = \phi + (2h)\phi_{,x} + \frac{(2h)^2}{2}\phi_{,xx} - O(2h)^3 \quad (3.85)$$

where $\phi = \phi(x_m, t_n)$. Substitute Eqns. (3.83) to (3.85) into Eq. (3.82) to give

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \frac{-3}{2h}[\phi] + \frac{2}{h} \left[\phi + h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} + O(h)^3 \right] - \frac{1}{2h} \left[\phi + (2h)\phi_{,x} + \frac{(2h)^2}{2}\phi_{,xx} + O(2h)^3 \right] \quad (3.86)$$

Simplify Eq. (3.86) to obtain

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \phi_{,x} + 2 O(h)^2 - O(2h)^2 \quad (3.87)$$

Eq. (3.87) is a final expression for the term of space derivative in Eq. (3.67). The term of time derivative is discretized in a similar manner as

$$\frac{\partial \phi(x_m, t_n)}{\partial t} = \begin{Bmatrix} a_{n,n} & a_{n,n+1} & a_{n,n+2} \end{Bmatrix} \begin{Bmatrix} \phi(x_m, t_n) \\ \phi(x_m, t_{n+1}) \\ \phi(x_m, t_{n+2}) \end{Bmatrix} \quad (3.88)$$

The discretization of the time derivative term similar to space derivative is obtained as

$$\frac{\partial \phi(x_m, t_n)}{\partial t} = \phi_{,t} + 2 O(t)^2 - O(2t)^2 \quad (3.89)$$

Substitute Eqns. (3.87) and (3.89) into Eq. (3.67), and subtract it from Eq. (3.66) to get

$$P\phi - P(h, t)\phi = \phi_{,t} + a\phi_{,x} - \phi_{,t} - 2 O(t)^2 + O(2t)^2 - a\phi_{,x} - 2a O(h)^2 + a O(2h)^2 \quad (3.90)$$

Eq. (3.90) tends to zero as h and $t \rightarrow 0$, which indicates that the presented discretization scheme is consistent.

The virtual node (x_m, t_n) is assumed at the right end of the domain with the DQ local domain, as shown in Fig. 3.27c. The space derivative term in Eq. (3.67) is discretized as

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \begin{Bmatrix} a_{m,m-2} & a_{m,m-1} & a_{m,m} \end{Bmatrix} \begin{Bmatrix} \phi(x_{m-2}, t_n) \\ \phi(x_{m-1}, t_n) \\ \phi(x_m, t_n) \end{Bmatrix} \quad (3.91)$$

The weighting coefficients are determined by Eq. (3.69) as

$$a_{m,m-2} = \frac{1}{2h}, \quad a_{m,m-1} = \frac{-2}{h}, \quad a_{m,m} = \frac{3}{2h} \quad (3.92)$$

As per Taylor series

$$\phi(x_{m-1}, t_n) = \phi(x_m - h) = \phi - h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} - O(h)^3 \quad (3.93)$$

$$\phi(x_{m-2}, t_n) = \phi(x_m - 2h) = \phi - (2h)\phi_{,x} + \frac{(2h)^2}{2}\phi_{,xx} - O(2h)^3 \quad (3.94)$$

where $\phi = \phi(x_m, t_n)$. Substitute Eqns. (3.92) to (3.94) into Eq. (3.91), simplify it to obtain

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \phi_{,x} - O(2h)^2 + 2 O(h)^2 \quad (3.95)$$

Fig. 3.27. The DQ local domains at the concerned virtual node (x_m, t_n) when the virtual node is assumed in internal domain (a), at left end (b) and right end (c) of the domain for the 1-D wave equation.

Eq. (3.95) is a final expression for the space derivative term in Eq. (3.67). The discretization of time derivative term is obtained by following the similar procedure as

$$\frac{\partial \phi(x_m, t_n)}{\partial t} = \begin{Bmatrix} a_{n,n} & a_{n,n+1} & a_{n,n+2} \end{Bmatrix} \begin{Bmatrix} \phi(x_m, t_n) \\ \phi(x_m, t_{n+1}) \\ \phi(x_m, t_{n+2}) \end{Bmatrix} \quad (3.96)$$

Eq. (3.96) is given by weighting coefficients and Taylor series expansions as

$$\frac{\partial \phi(x_m, t_n)}{\partial t} = \phi_{,t} + 2 O(t)^2 - \left(\frac{1}{2}\right) O(2t)^2 \quad (3.97)$$

Substitute Eq. (3.95) and Eq. (3.97) into Eq. (3.67) and subtract it from Eq. (3.66) to get

$$P\phi - P(h,t)\phi = \phi_{,t} + a\phi_{,x} - \phi_{,t} - 2 O(t)^2 + (0.5) O(2t)^2 - a \phi_{,x} + a O(2h)^2 - (2a) O(h)^2 \quad (3.98)$$

Eq. (3.98) tends to zero as h and $t \rightarrow 0$, which shows that the presented discretization scheme is consistent. It is seen from Eqns. (3.81), (3.90) and (3.98) that the weighting coefficients computed by Shu's general approach [36-37] give consistent discretization of the governing equation for the locally applied DQ method with the uniform virtual nodes.

3.6.1.2 Quan and Chang approach

The weighting coefficients are computed in this section by Quan and Chang approach [34-35] with the uniform distribution of virtual nodes.

The virtual node (x_m, t_n) is assumed an internal domain node with the DQ local domain, as shown in Fig. 3.27a. The space term derivative in Eq. (3.67) is discretized as

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \begin{Bmatrix} a_{m,m-1} & a_{m,m} & a_{m,m+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_{m-1}, t_n) \\ \phi(x_m, t_n) \\ \phi(x_{m+1}, t_n) \end{Bmatrix} \quad (3.99)$$

As per Quan and Chang approach [34-35], the weighting coefficients are given as

$$a_{ij} = \frac{1}{x_j - x_i} \prod_{k=1, k \neq i, j}^{N_x} \frac{(x_i - x_k)}{(x_j - x_k)}, \quad a_{ii} = \sum_{j=1, j \neq i}^{N_x} \frac{1}{(x_i - x_j)} \quad (3.100)$$

The weighting coefficients in Eq. (3.99) are computed by Eq. (3.100) and given as

$$a_{m,m-1} = \frac{1}{2h}, \quad a_{m,m+1} = \frac{-1}{2h}, \quad a_{m,m} = 0 \quad (3.101)$$

Substitute Eqns. (3.71) to (3.72) and (3.101) into Eq. (3.99), simplify it to

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = -\phi_{,x} - O(h)^2 \quad (3.102)$$

Eq. (3.102) is a final equation for the term of space derivative in Eq. (3.67). The term of time derivative in Eq. (3.67) is discretized as given in Eq. (3.75), and the corresponding weighting coefficients are computed by Eq. (3.100) as

$$a_{n,n-1} = \frac{1}{2t}, \quad a_{n,n+1} = \frac{-1}{2t}, \quad a_{n,n} = 0 \quad (3.103)$$

Substituting Eqns. (3.77) to (3.78) and (3.103) into Eq. (3.75), and simplifying it results in

$$\frac{\partial \phi(x_m, t_n)}{\partial t} = -\phi_t - O(t)^2 \quad (3.104)$$

Substitute Eq. (3.102) and Eq. (3.104) into Eq. (3.67), subtract it from Eq. (3.66) to get

$$P\phi - P(h, t)\phi = \phi_t + a\phi_{,x} + \phi_t + O(t)^2 + a\phi_{,x} + aO(h)^2 \quad (3.105)$$

Eq. (3.105) tends to $2P\phi$ as h and $t \rightarrow 0$. As a result, the presented discretization scheme is not consistent. As the governing equation discretized at the virtual node located in the internal domain, with the weighting coefficients computed by Quan and Chang approach, is not consistent, it is unnecessary to consider the cases of the virtual node located at the left and right ends of the domain. It is also observed that the consistent discretization of governing equation by Quan and Chang approach [34-35] is highly dependent on the pattern of distribution of virtual nodes, which was also observed in [34].

3.6.2 Consistency analysis of 1-D wave equation by the cosine distribution of virtual nodes

The virtual nodes are distributed by cosine law in this section resulting in the nodal coordinates similar to the roots of the N^{th} -order orthogonal polynomial of the 1st-kind as

$$x_i = x_0 + \frac{L}{2} \left[1 - \cos\left(\frac{i-1}{N-1}\pi\right) \right], \quad \text{for } i=1, 2, \dots, N \quad (3.106)$$

where x_0 is a starting coordinate of the domain, L is the domain length, and N is the number of total virtual nodes in a domain. The index i is replaced by $i-1$ for node x_{i-1}

$$x_{i-1} = x_0 + \frac{L}{2} \left[1 - \cos\left(\frac{i-2}{N-1}\pi\right) \right] \quad (3.107)$$

The following notation is used to simplify the equation when the difference between the two cosine distributed virtual nodes is computed as

$$x_i - x_{i-1} = \frac{L}{2} \cos\left(\frac{i-2}{N-1} \pi\right) - \frac{L}{2} \cos\left(\frac{i-1}{N-1} \pi\right) = \Delta x(i-2, i-1) \frac{L}{2} \quad (3.108)$$

The virtual node (x_m, t_n) is assumed an internal domain node with the DQ local domain shown in Fig. 3.27a. The term of space derivative in Eq. (3.67) is discretized by the DQ method, as given in Eq. (3.68), and the weighting coefficients are computed by Eq. (3.69)

$$a_{m,m-1} = \frac{\Delta x(m, m-1)}{\Delta x(m-1, m-2) \Delta x(m, m-2)} \left(\frac{2}{L}\right), \quad a_{m,m+1} = \frac{\Delta x(m-2, m-1)}{\Delta x(m-2, m) \Delta x(m-1, m)} \left(\frac{2}{L}\right)$$

$$a_{m,m} = -(a_{m,m-1} + a_{m,m+1}) \quad (3.109)$$

A per Taylor series

$$\phi(x_{m-1}, t_n) = \phi(x_m - h_1) = \phi - h_1 \phi_{,x} + \frac{h_1^2}{2} \phi_{,xx} - O(h_1)^3 \quad (3.110)$$

$$\text{where } h_1 = x_m - x_{m-1} = \Delta x(m-2, m-1) \frac{L}{2} \quad (3.111)$$

$$\phi(x_{m+1}, t_n) = \phi(x_m + h_2) = \phi + h_2 \phi_{,x} + \frac{h_2^2}{2} \phi_{,xx} + O(h_2)^3 \quad (3.112)$$

$$\text{where } h_2 = x_{m+1} - x_m = \Delta x(m-1, m) \frac{L}{2} \quad (3.113)$$

where $\phi = \phi(x_m, t_n)$. Substituting Eqns. (3.110) and (3.112) in Eq. (3.67) results in

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \phi_{,x} [a_{m,m-1} (-h_1) + a_{m,m+1} h_2] + \phi_{,xx} \left[a_{m,m-1} \frac{h_1^2}{2} + a_{m,m+1} \frac{h_2^2}{2} \right] -$$

$$a_{m,m-1} O(h_1)^3 + a_{m,m+1} O(h_2)^3 \quad (3.114)$$

Simplify Eq. (3.114) by Eqns. (3.109), (3.111) and (3.113) to obtain

$$\frac{\partial \phi(x_m, t_n)}{\partial x} = \phi_{,x} - a_{m,m-1} O(h_1)^3 + a_{m,m+1} O(h_2)^3 \quad (3.115)$$

Eq. (3.115) is a final expression for the discretization of space derivative term in Eq. (3.67). Similarly, the term of time derivative in Eq. (3.67) is discretized and simplified as

$$\frac{\partial \phi(x_m, t_n)}{\partial t} = \phi_{,t} - a_{n,n-1} O(t_1)^3 + a_{n,n+1} O(t_2)^3 \quad (3.116)$$

Substituting Eqns. (3.115) and (3.116) in Eq. (3.67), and then subtracting it from Eq. (3.66) results in

$$P\phi - P(h, t)\phi = \phi_{,t} + a\phi_{,x} - \phi_{,t} + a_{n,n-1} O(t_1)^3 - a_{n,n+1} O(t_2)^3 - a\phi_{,x} + a a_{m,m-1} O(h_1)^3 - a a_{m,m+1} O(h_2)^3 \quad (3.117)$$

Eq. (3.117) is a consistency equation obtained by the cosine distribution of virtual nodes. As h and $t \rightarrow 0$, $(N_x - 1) \rightarrow \infty$ and $(N_t - 1) \rightarrow \infty$, and $a_{m,m-1}, a_{m,m+1}, a_{n,n-1}, a_{n,n+1} \rightarrow 0$, therefore Eq. (3.117) tends to zero. Therefore, the presented discretization scheme is consistent by the cosine distribution of virtual nodes. If the cosine distribution of virtual nodes is changed to the uniform one, the new weighting coefficients are given as

$$a_{m,m-1} = \frac{\Delta x(m, m-1)}{\Delta x(m-1, m-2) \Delta x(m, m-2)} \frac{2}{L} = \frac{-1}{2h}, \quad a_{m,m+1} = \frac{\Delta x(m-2, m-1)}{\Delta x(m-2, m) \Delta x(m-1, m)} \frac{2}{L} = \frac{1}{2h}$$

$$a_{m,m} = -(a_{m,m-1} + a_{m,m+1}) = 0 \quad (3.118)$$

$$h_1 = \Delta x(m-2, m-1) = x_m - x_{m-1} = h \quad (3.119)$$

$$h_2 = \Delta x(m-1, m) = x_{m+1} - x_m = h \quad (3.120)$$

$$t_1 = \Delta x(n-2, n-1) = t_n - t_{n-1} = t \quad (3.121)$$

$$t_2 = \Delta x(n-1, n) = t_{n+1} - t_n = t \quad (3.122)$$

Substituting Eqns. (3.118) to (3.122) in Eq. (3.117) results in

$$P\phi - P(h, t)\phi = \phi_{,t} + a\phi_{,x} - \phi_{,t} + \left(\frac{-1}{2t}\right)O(t)^3 - \left(\frac{1}{2t}\right)O(t)^3 - a\phi_{,x} + a\left(\frac{-1}{2t}\right)O(h)^3 - a\left(\frac{1}{2t}\right)O(h)^3 \quad (3.123)$$

Simplify Eq. (3.123) to get

$$P\phi - P(h, t)\phi = \phi_{,t} + a\phi_{,x} - \phi_{,t} - O(t)^2 - a\phi_{,x} - a O(h)^2 \quad (3.124)$$

Eq. (3.124) is same as Eq. (3.81). Therefore, it is shown that the consistency equation obtained by the cosine distribution of virtual nodes converges to that obtained by the

uniform distribution of virtual nodes. Similar results are obtained for the other configurations of the DQ local domains as well.

3.6.3 Consistency analysis of the 1-D Laplace equation by the uniform distribution of virtual nodes

The consistency analysis is performed in this section for the 1-D Laplace equation by the uniform virtual nodes. The weighting coefficients are computed by Eq. (3.69).

The virtual node (x_m) is assumed an internal domain node with the DQ local domain shown in Fig. 3.28a. The 1-D Laplace equation is given as

$$\phi_{,xx} = 0, \quad P\phi = 0 \quad (3.125)$$

where $P = (\partial^2 / \partial x^2)$. Eq. (3.125) is discretized at the virtual node (x_m) , therefore

$$P(h)\phi = \frac{\partial^2 \phi(x_m)}{\partial x^2} \quad (3.126)$$

where h is a virtual node spacing along the x axis. The term of space derivative in Eq. (3.126) is approximated by the locally applied DQ method as

$$P(h)\phi = \{b_{m,m-2} \quad b_{m,m-1} \quad b_{m,m} \quad b_{m,m+1} \quad b_{m,m+2}\} \phi \begin{bmatrix} \phi(x_{m-2}) \\ \phi(x_{m-1}) \\ \phi(x_m) \\ \phi(x_{m+1}) \\ \phi(x_{m+2}) \end{bmatrix} \quad (3.127)$$

where $b_{m,k}$, for $k = m-2, \dots, m+2$, are the DQ weighting coefficients for the 2nd-order derivative as given

$$b_{ij} = 2 a_{ij} \left[a_{ii} - \frac{1}{x_i - x_j} \right] \quad (3.128)$$

where a_{ij} and a_{ii} are the DQ weighting coefficients for the 1st-order derivative. The weighting coefficients in Eq. (3.127) are computed by Eqns. (3.69) and (3.128), and given in Eq. (3.130) as

$$a_{m,m-2} = \frac{1}{12h}, \quad a_{m,m-1} = \frac{-2}{3h}, \quad a_{m,m+2} = \frac{-1}{12h}, \quad a_{m,m+1} = \frac{2}{3h}, \quad a_{m,m} = 0 \quad (3.129)$$

$$b_{m,m-2} = \frac{-1}{12h^2}, \quad b_{m,m-1} = \frac{4}{3h^2}, \quad b_{m,m+1} = \frac{4}{3h^2}, \quad b_{m,m+2} = \frac{-1}{12h^2}, \quad b_{m,m} = \frac{-5}{2h^2} \quad (3.130)$$

As per Taylor series

$$\phi(x_{m-2}) = \phi(x_m - 2h) = \phi - (2h)\phi_{,x} + \frac{(2h)^2}{2}\phi_{,xx} - O(2h)^3 \quad (3.131)$$

$$\phi(x_{m-1}) = \phi(x_m - h) = \phi - h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} - O(h)^3 \quad (3.132)$$

$$\phi(x_{m+2}) = \phi(x_m + 2h) = \phi + (2h)\phi_{,x} + \frac{(2h)^2}{2}\phi_{,xx} + O(2h)^3 \quad (3.133)$$

$$\phi(x_{m+1}) = \phi(x_m + h) = \phi + h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} + O(h)^3 \quad (3.134)$$

Substitute Eqns. (3.130) to Eq. (3.134) into Eq. (3.127) with simplification results in

$$P(h)\phi = \phi_{,xx} \quad (3.135)$$

Subtract Eq. (3.135) from Eq. (3.125) to get

$$P\phi - P(h)\phi = \phi_{,xx} - \phi_{,xx} = 0 \quad (3.136)$$

The presented discretization scheme is consistent, as Eq. (3.136) is equal to zero.

The virtual node (x_m) is assumed at the right end of the domain with the DQ local domain shown in Fig. 3.28b. The term of space derivative in Eq. (3.126) is discretized by the locally applied DQ method as

$$P(h)\phi = \begin{Bmatrix} b_{m,m-2} & b_{m,m-1} & b_{m,m} \end{Bmatrix} \begin{bmatrix} \phi(x_{m,m-2}) \\ \phi(x_{m,m-1}) \\ \phi(x_{m,m}) \end{bmatrix} \quad (3.137)$$

The weighting coefficients are computed by Eqns. (3.69) and (3.128)

$$b_{m,m-2} = \frac{1}{h^2}, \quad b_{m,m-1} = \frac{-2}{h^2}, \quad b_{m,m} = \frac{1}{h^2} \quad (3.138)$$

As per Taylor series

$$\phi(x_{m-2}) = \phi(x_m - 2h) = \phi - (2h)\phi_{,x} + \frac{(2h)^2}{2}\phi_{,xx} - O(2h)^3 \quad (3.139)$$

$$\phi(x_{m-1}) = \phi(x_m - h) = \phi - h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} - O(h)^3 \quad (3.140)$$

Substituting Eqns. (3.138) to (3.140) into Eq. (3.137), we simplify the result and subtract it from Eq. (3.125) to obtain

$$P\phi - P(h)\phi = \phi_{,xx} - \phi_{,xx} + O(8h) + 2O(h) \quad (3.141)$$

Fig. 3.28. The DQ local domains at the virtual node (x_m) when the virtual node is assumed inside the internal domain (a) and right end of the domain (b) for the 1-D Laplace equation.

Eq. (3.141) tends to zero as $h \rightarrow 0$, therefore the current scheme of discretization is consistent. It is shown from Eqns. (3.136) and (3.141) that the discretized form of the 1-D Laplace equation by locally applied DQ method is consistent with the continuous form given in Eq. (3.125). It is observed from the consistency analysis that the discretization error by uniform distribution of virtual nodes is much smaller than by cosine distribution of virtual nodes.

3.7 Effect of the uniform and cosine distributions of virtual nodes on the convergence of RDQ method

Several 1-D, 2-D and elasticity problems are solved in this section by the RDQ method. Their convergence analysis is performed by considering the observations made from the consistency analysis. All the given problems are solved by the uniform and cosine distributions of virtual nodes, and their rates of function convergence are

compared. The distribution pattern of the field and virtual nodes is identified, which gives better rates of convergence. For all the problems solved here, the global error is computed by Eq. (3.6), which is same as the l_2 error norm but averaged over the total field nodes, and normalized by an absolute maximum value of the analytical function.

3.7.1 1-D test problems

The first 1-D problem is a Poisson equation with a variable force term of the function x and mixed boundary conditions. The governing equation and boundary conditions are given as

$$\frac{d^2 f}{dx^2} = -x, \quad (0 < x < 1) \quad (3.142)$$

$$f(x=0) = 0, \quad \frac{df}{dx}(x=1) = 0 \quad (3.143)$$

The analytical solution with the 3rd-order continuity is given as $f(x) = -(x^3)/6 + (x/2)$. This problem is solved by including up to the 2nd-order monomial terms in the basis of shape function, and with 21, 41, 81, 161, 321, 641 field and 641 virtual nodes, respectively. The uniform and randomly distributed field nodes are separately combined with the cosine and uniformly distributed virtual nodes. All the errors are computed by Eq. (3.6). The convergence rates for both types of virtual nodes are given in Table 3.4 and Table 3.5, and the corresponding convergence curves are plotted in Fig. 3.29. It is seen from Table 3.4 and Table 3.5 that the convergence rate of the function f by the uniform distribution of virtual nodes is improved as compared with that by the cosine distribution of virtual nodes. It is observed that, good convergence rate of the function f is obtained by combining the uniformly distributed virtual nodes with the randomly distributed field nodes, whereas good convergence rate of the derivative of function is obtained by combining the uniformly distributed virtual nodes with the uniformly distributed field nodes.

Fig. 3.29. Convergence curves when the uniform field nodes are combined with the cosine (a) and uniform (b) virtual nodes, and the random field nodes are combined with the cosine (c) and uniform (d) virtual nodes for the 1st 1-D problem.

This problem is additionally solved by five different sets of the random nodes to study the effect of random field nodes on the rate of convergence. The corresponding convergence plots of the function value are shown in Fig. 3.30. When the convergence rates shown in Fig. 3.30 are compared with the value 1.73 given in Table 3.4, which is obtained by the random nodes generated without giving any starting reference value, as explained earlier, it is observed that 1.73 is a good measure of the performance of RDQ method. It is also observed from Fig. 3.30 that the global error varies significantly for the low number of random field nodes, but it converges to a fix value as the number of random field nodes is increased. The results obtained by solving this problem with five different sets of the random field nodes indicate that, even if different sets of the random field nodes are used, the values of the global error may vary a little, but the convergence

rate of the function will be almost the same. This is an important result, and it signifies that the global error is almost constant for any set of sufficiently high number of random field nodes. As a result, the computed values of function at the random field nodes have an equal numerical accuracy.

The second 1-D problem is also a Poisson equation with the 2nd-order continuous force term and mixed boundary conditions. The 4th-order continuous analytical solution is evaluated by including up to the 2nd-order monomial terms in the polynomial basis of a shape function. The governing equation and boundary conditions are given as

Fig. 3.30. Convergence curves of the function by five different sets of the random field nodes for the 1st 1-D problem, where the global error varies for the low number of field nodes but converges to almost the constant value as the total random field nodes are increased.

$$\frac{d^2 f}{dx^2} = \frac{105}{2}x^2 - \frac{15}{2}, \quad (-1 < x < 1) \quad (3.144)$$

$$f(x = -1) = 1, \quad \frac{df}{dx}(x = 1) = 10 \quad (3.145)$$

The analytical solution is given as $f(x) = (35/8)x^4 - (15/4)x^2 + (3/8)$. This problem is solved by 21, 41, 81, 161, 641 field and 641 virtual nodes, respectively. The uniform and randomly distributed field nodes are separately combined with the cosine and uniformly distributed virtual nodes. The convergence rates for both types of virtual nodes are given in Table 3.6 and Table 3.7, corresponding convergence curves are plotted in Fig. 3.31. It is noted from Table 3.6 and Table 3.7 that the convergence rate of the function by the

uniform distribution of virtual nodes is improved, as compared with the cosine distribution of virtual nodes.

Table 3.4. Convergence rates for the 1st 1-D problem of Poisson equation when the cosine distributed virtual nodes are combined with the uniform and randomly distributed field nodes, respectively

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	1.6	1.73
$f_{,x}$	1.97	1.0
$f_{,xx}$	2.1	0.8

This problem is also solved in the earlier section of convergence analysis, but only for the cosine distribution of virtual nodes combined with the uniform and random distributions of field nodes. It is additionally solved here for the uniform virtual nodes combined with the uniform and random field nodes, and their convergence rates are compared.

The 3rd 1-D example is a local high gradient problem with the force term containing an exponential expression. Exponential function can be approximated by an infinite series, such as the power series by Taylor expansion. It is solved here by including up to the 2nd-order monomial terms in the polynomial basis of a shape function computation. The governing equation and boundary conditions are given as

$$\frac{d^2 f}{dx^2} = -6x - \left[\left(\frac{2}{\alpha^2} \right) - 4 \left(\frac{x-\beta}{\alpha^2} \right)^2 \right] \exp \left[- \left(\frac{x-\beta}{\alpha} \right)^2 \right], \quad (0 < x < 1) \quad (3.146)$$

$$f(x=0) = \exp \left[- \left(\frac{\beta^2}{\alpha^2} \right) \right], \quad \frac{df(x=1)}{dx} = -3 - 2 \left(\frac{1-\beta}{\alpha^2} \right) \exp \left[- \left(\frac{1-\beta}{\alpha} \right)^2 \right] \quad (3.147)$$

The analytical solution is given as $f(x) = -x^3 + \exp \{ -[(x-\beta)/\alpha]^2 \}$. This problem is solved by 21, 41, 81, 161, 321, 641 field and 641 virtual nodes, respectively.

Table 3.5. Convergence rates for the 1st 1-D problem of Poisson equation when the uniformly distributed virtual nodes are combined with the uniform and randomly distributed field nodes, respectively

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	2.8	3.2
$f_{,x}$	3.1	1.4
$f_{,xx}$	2.9	1.3

Table 3.6. Convergence rates for the 2nd 1-D problem of Poisson equation when the cosine distributed virtual nodes are combined with the uniform and randomly distributed field nodes, respectively

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	1.7	1.8
$f_{,x}$	0.9	0.8
$f_{,xx}$	0.9	0.7

Table 3.7. Convergence rates for the 2nd 1-D problem of Poisson equation by combining the uniform distribution of virtual nodes with the uniform and random field nodes, respectively

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	2.1	2.7
$f_{,x}$	1.3	0.8
$f_{,xx}$	1.4	0.7

The uniform and randomly distributed field nodes are separately combined with the cosine and uniformly distributed virtual nodes. The convergence rates for both types of

virtual nodes are given in Table 3.8 and Table 3.9, corresponding convergence curves are plotted in Fig. 3.32.

Fig. 3.31. Convergence curves when uniform field nodes are combined with the cosine virtual nodes (a), random field nodes are combined with the cosine virtual nodes (b), uniform field nodes are combined with the uniform virtual nodes (c) and the random field nodes are combined with the uniform virtual nodes (d) for the 2nd 1-D problem.

It is again observed from Table 3.8 and Table 3.9 that very good convergence rates of the function and its derivative are achieved by combining the uniform distribution of virtual nodes with the random and uniform distributions of field nodes, respectively. Fig. 3.33 shows the comparison between the numerical and analytical values of the function, and its 1st- and 2nd-order derivatives, respectively, by 641 uniform field and cosine distributed virtual nodes. It is seen that the locally high gradient function value at $x = 0.5$ is captured very well, and the 1st- and 2nd-order derivatives are also accurately captured.

Fig. 3.32. Convergence curves when the uniform field nodes are combined with the cosine virtual nodes (a), random field nodes are combined with the cosine virtual nodes (b), uniform field nodes are combined with the uniform virtual nodes (c) and the random field nodes are combined with the uniform virtual nodes (d) for the 3rd 1-D problem.

This problem is additionally solved to study the effect of different sets of random nodes on the rate of convergence. The corresponding convergence plots of the function values are shown in Fig. 3.33d by the five different sets of random field nodes. If the convergence rates of the function shown in Fig. 3.33d are compared with the rate 2.5 given in Table 3.8, which is obtained by the random nodes generated without giving any starting reference value, as explained earlier, it is noted that 2.5 is a good measure of the performance of RDQ method in a sense that it averagely represents the rates of convergence shown in Fig. 3.33d. It is also observed here that the global error varies for the low number of field nodes, but their difference reduces with increase in the random field nodes.

It is seen from all the results that the weighted derivative approach to compute the approximate derivatives gives good rates of derivative convergence. It is observed from the consistency analysis that the uniform distribution of virtual nodes gives smaller error in the discretization of governing equation as compared with that by the cosine distribution. As a result, this leads to the better rates of convergence by the uniform distribution of virtual nodes. This observation is verified in this section by comparing the convergence results obtained by the uniform and cosine distributions of virtual nodes. For all the problems, much better convergence values of the function are achieved by the uniform distribution of virtual nodes than the cosine one.

Table 3.8. Convergence rates for the 3rd 1-D problem by the uniform and random distributions of the field nodes combined respectively with the cosine distribution of virtual nodes

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	3.3	2.5
$f_{,x}$	2.2	1.6
$f_{,xx}$	1.8	1.3

Table 3.9. Convergence rates for the 3rd 1-D problem by the uniform and random distributions of the field nodes combined respectively with the uniform distribution of virtual nodes

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	3.0	3.3
$f_{,x}$	1.9	1.2
$f_{,xx}$	1.6	1.2

It is also observed from all the results that the higher convergence rates of the function values are achieved by the random distribution of field nodes. This is a very important result since it highlights one of the objectives of the development of RDQ method. All

the convergence curves show that the global error by the random distributed field nodes reduces very fast after sufficiently high number of random field nodes.

3.7.2 2-D test problems

The 1st 2-D problem is a Laplace equation with mixed boundary conditions. The governing equation and boundary conditions are given as

$$\frac{d^2 f(x, y)}{dx^2} + \frac{d^2 f(x, y)}{dy^2} = 0, \quad (0 < x < 1) \text{ and } (0 < y < 1) \quad (3.148)$$

$$f(x=0, y) = -y^3, \quad f(x=1, y) = -1 - y^3 + 3y^2 + 3y \quad (3.149)$$

$$\frac{df}{dy}(x, y=0) = 3x^2, \quad \frac{df}{dy}(x, y=1) = -3 + 6x + 3x^2 \quad (3.150)$$

The analytical solution is given as $f(x, y) = -x^3 - y^3 + 3xy^2 + 3x^2y$. This problem is solved by including up to the 2nd-order monomials in the shape function basis, and with 5×5 , 17×17 , 33×33 , 41×41 , 44×44 field and 44×44 virtual nodes respectively. The uniform and randomly distributed field nodes are separately combined with the cosine and uniformly distributed virtual nodes. The convergence rates for both types of virtual nodes are given in Table 3.10 and Table 3.11, and the corresponding convergence curves are plotted in Fig. 3.34. It is noted by comparing Table 3.10 and Table 3.11 that the convergence rate of the function by the uniform distribution of virtual nodes is better as compared with that by the cosine distribution of virtual nodes, notably with the random field nodes. The convergence rates of the derivatives are not much affected by the different distributions of the field and virtual nodes.

This problem is also solved during the convergence analysis, but only by the cosine distribution of virtual nodes combined with the uniform and random distributions of field nodes. It is additionally solved here by the uniform distribution of virtual nodes separately combined with the uniform and random field nodes. Nevertheless, all the plots of the convergence are given here for an easy comparison among themselves.

The 2nd 2-D problem is of steady-state heat conduction without an internal heat generation in a slab with no temperature gradient in the z direction, as shown in Fig. 3.35a. The governing equation and boundary conditions are given as

Fig. 3.33. Comparison between the analytical and numerical values of function f (a), the 1st derivative $f_{,x}$ (b) and the 2nd derivative $f_{,xx}$ (c) for the 3rd 1-D problem by 641 uniform field nodes combined with the 641 cosine virtual nodes, and the plots of function value convergence by five different sets of random field nodes combined with the cosine distribution of virtual nodes (d), where the global error varies significantly for the lower number of field nodes but their difference reduces with increase in the number of random field nodes.

$$\frac{d^2T}{dx^2} + \frac{d^2T}{dy^2} = 0, \quad (0 < x < a) \text{ and } (0 < y < b) \quad (3.151)$$

$$T(x=0)=0, \quad T(x=a)=0, \quad T(y=0)=0, \quad T(y=b)=100 \quad (3.152)$$

The analytical solution derived using the Fourier series is given as $T(x, y) = T_s \sum_{n=1}^{\infty} [n\pi \sinh(n\pi b/a)]^{-1} 2[1 - (-1)^n] \sin(n\pi x/a) \sinh(n\pi y/a)$. This problem is solved, by including up to the 2nd-order monomials in the shape function basis, and with 5×5 , 9×9 , 17×17 , 21×21 , 26×26 , 33×33 field and 44×44 virtual nodes, respectively. The uniform and randomly distributed field nodes are combined with the

cosine and uniformly distributed virtual nodes. The convergence rates of both types of virtual nodes are given in Table 3.12 and Table 3.13, corresponding convergence curves are plotted in Fig. 3.35. It is seen from Table 3.12 and Table 3.13 that the convergence rate of function is improved by the uniform distribution of virtual nodes as compared with the cosine one.

Fig. 3.34. Convergence curves when the uniform field nodes are combined with the cosine virtual nodes (a), random field nodes are combined with the cosine virtual nodes (b), uniform field nodes are combined with the uniform virtual nodes (c) and the random field nodes are combined with the uniform virtual nodes (d) for the 1st 2-D problem.

The comparison between the numerical and analytical values of temperature along the central x-axis plane is given in Fig. 3.35d by the uniform distribution of 33×33 field and the cosine distribution of 44×44 virtual nodes, and it is noted that the boundary conditions are exactly imposed.

3.7.3 Elasticity problems

All the problems presented in this section are based on the plane-stress condition with the conventional equilibrium equation, expressed in the form of the displacements, as explained earlier.

Table 3.10. Convergence rates for the 1st 2-D problem by the cosine distribution of virtual nodes combined with the uniform and random distributions of the field nodes, respectively

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	2.25	3.75
$f_{,x}$	1.2	1.2
$f_{,y}$	1.2	1.2
$f_{,xx}$	0.8	0.8
$f_{,yy}$	0.8	0.8

Table 3.11. Convergence rates for the 1st 2-D problem by combining the uniformly distributed virtual nodes with the uniformly and randomly distributed field nodes, respectively

Function	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
f	1.95	4.73
$f_{,x}$	1.2	1.2
$f_{,y}$	1.2	1.2
$f_{,xx}$	0.8	0.8
$f_{,yy}$	0.8	0.8

a

b

c

d

Fig. 3.35. Computational domain and boundary conditions for a steady-state heat conduction problem with a heat source at $y = b$ (a), convergence curves for the uniform distribution of field nodes combined with the uniform and cosine distributions of virtual nodes (b), convergence curves for the random distribution of field nodes combined with the uniform and cosine distributions of virtual nodes (c), and the comparison between the numerical and analytical values of temperature along the central x-axis plane (d).

3.7.3.1 Cantilever beam under a pure bending load

A cantilever beam is loaded with a bending load, as shown in Fig. 3.36a. This problem is solved earlier by combining the cosine virtual nodes with the uniform and random field nodes. Here it is solved again by combining the uniformly distributed virtual nodes with the uniform and randomly distributed field nodes for comparison.

This problem is solved for $L = 48$, $D = 12$, $M = -24000$, $\nu_0 = 0.3$, $E = 3.0 \times 10^7$, and 13×13 , 17×17 , 21×21 , 29×29 field and 41×41 virtual nodes respectively, by including up to the 1st-order monomial terms in the shape function basis. The convergence rates in the displacements u and v by the cosine distribution of virtual

nodes combined with the uniform and random distributions of field nodes, respectively, are found to be 1.0, 1.3 and 1.96, 2.0, respectively, and the corresponding convergence curves are given in Fig. 3.36b.

Table 3.12. Convergence rates for the 2nd 2-D problem by the cosine distribution of virtual nodes combined with the uniform and random distributions of the field nodes

Function	Convergence rates	Convergence rates
	(uniform field nodes)	(random field nodes)
f	0.6	0.65
$f_{,x}$	0.8	0.9

Table 3.13. Convergence rates for the 2nd 2-D problem by combining the uniform distribution of virtual nodes with the uniform and random distributions of the field nodes

function	Convergence rates	Convergence rates
	(uniform field nodes)	(random field nodes)
f	0.7	0.9
$f_{,x}$	0.8	0.9

Similarly, the convergence rates in the displacements u and v by combining the uniform distribution of virtual nodes with the uniform and random distributions of the field nodes, respectively, are found to be 0.2, 0.2 and 0.4, 0.4, respectively, and the corresponding curves are plotted in Fig. 3.36. It is seen that the convergence rates by the cosine distribution of virtual nodes are better than that of the uniform distribution.

It is observed from Fig. 3.36c that the value of global error in the displacement v decreases slowly after 21×21 field nodes, as compared with that in the displacement u . With reference to the convergence curve of displacement u , the convergence curve of displacement v is shifted in the y direction because the absolute values of the global error

in the displacement u are lower as compared with those in the displacement v for the equal number of field nodes. It is also observed by comparing Fig. 3.36b with Fig. 3.36c and Fig. 3.36d that the global error by the cosine virtual nodes is higher at lower field nodes, as seen in Fig. 3.36b, as compared with that by the uniform virtual nodes, as seen in Fig. 3.36c and Fig. 3.36d, therefore naturally the rate of decrease of error using the cosine virtual nodes is better as compared with the uniform virtual nodes.

Fig. 3.36. Schematic of a cantilever beam under a pure bending load (a), convergence plots obtained by combining the cosine virtual nodes with the uniform and random field nodes (b), and convergence plots obtained by combining the uniform virtual nodes with the uniform (c) and random (d) field nodes.

3.7.3.2 Semi-infinite plate with a central hole

A semi-infinite plate with a central hole along with the boundary conditions is shown in Fig. 3.37a. This problem is solved earlier by combining the cosine virtual nodes with the uniform and random field nodes, respectively. Here it is solved again by combining the

uniform distribution of virtual nodes with the uniform and random distributions of field nodes for comparison.

This problem is solved for $a = 1$, $b = 5$, $p = 1$, $\nu_0 = 0.3$, and $E = 1000$ with 6×6 , 11×11 , 21×21 , 31×31 uniform field nodes and 34×34 cosine and uniformly distributed virtual nodes, respectively, by including up to the 2nd-order monomial terms in the polynomial basis of a function approximation. The rates of convergence in the displacements u and v by the uniform field nodes combined with the cosine and uniform virtual nodes, respectively, are 0.3, 0.3, and 0.3, 0.3 respectively, and the corresponding convergence plots are given in Fig. 3.37b.

Fig. 3.37. One quarter computational domain of a semi-infinite plate (a) and the convergence plots of the displacements u and v by the uniform field nodes combined with the cosine and uniformly distributed virtual nodes (b) for a problem of semi-infinite plate with a central hole.

3.8 Summary

At first, the convergence analysis of RDQ method is performed by solving several 1-D and 2-D test problems. The superconvergence condition is derived at the beginning for the RDQ method. If the field nodes are computed by the superconvergence condition given in Eq. (3.5), the function and its approximate derivatives converge at the rate of $O(h^{p+\alpha})$, where $\alpha \geq 1$ for the approximated function and $\alpha \approx 0.7$ to 1 for the approximated derivative. It is noted from the test problems that the RDQ method can work well for the domain discretized by either the uniform or random field nodes.

Next, the RDQ method is applied to solve several elasticity problems with the 1st-, 2nd- and 3rd-order distributions of the field variables in the x and y directions. It is seen from the results that, when the complete order of monomials is added in the approximation of function, the numerical results closely match with the corresponding analytical solution.

Further, the RDQ method is applied to study the pull-in instability in the microswitches of the type fixed-fixed and cantilever under the influence of nonlinear electrostatic force field. The simulation results show that the pull-in voltages obtained by the RDQ method are closely matching with the literature values. A novel approach of transferring the multiple boundary conditions from the boundary nodes to their neighbouring nodes is used to get the modified boundary conditions. It is shown by the cantilever microswitch that the simulation results obtained by implementing the modified boundary conditions are more stable than the one obtained by original boundary conditions.

Finally, it is shown that the RDQ method successfully handles the uniformly as well as randomly distributed field nodes using the fixed RKPM interpolation function. It is seen from the convergence rates of the derivatives that the new approach called the weighted derivative approach gives very good convergence of the approximate derivatives. It is observed from Fig. 3.33 (a to c) that the 1st- and 2nd-order derivatives are well captured even though the function has a high local gradient. In order to perform the consistency analysis, the terms of function values at the virtual nodes from the discretized governing equation are replaced by Taylor series expansion to get a reasonable judgment of an analytical error in the discretization. As a result, with reference to the final consistency equations obtained by Taylor series, the same analysis can be applied to both the locally applied DQ and RDQ methods. It is concluded from the consistency analysis that the DQ method is consistent for the certain configurations of DQ local domains if the weighting coefficients are computed by Shu's general approach instead of Quan and Chang approach. It is also explained that the consistency of the discretized governing equation by the locally applied DQ method is independent of the distribution pattern used for the virtual nodes, such as even if the governing equation is discretized by the cosine distribution of virtual nodes, it converges to the consistency equation discretized by the

uniform distribution of virtual nodes when the nodal spacing is changed from the cosine to uniform one. This observation indicates that the error in the discretization of governing equation by the uniform distribution of virtual nodes should be less in comparison with the cosine one. Extensive numerical analysis is performed by the RDQ method, taking into consideration the observations made during the consistency analysis, by solving several 1-D, 2-D and elasticity problems with different orders of the field variable distributions.

When the convergence rates obtained by the cosine and uniform distributions of virtual nodes are compared, it is seen that good convergence rates of the function are achieved by combining the uniform distribution of virtual nodes with the random distribution of field nodes, whereas good convergence rates of the function derivatives are achieved by combining the uniform distribution of virtual nodes with the uniform distribution of field nodes. This is a very important result as it highlights the applicability of the RDQ method.

Overall, it is concluded that the RDQ method is capable of effectively handling the uniform as well as randomly distributed field nodes coupled with the uniform or cosine distributed virtual nodes over the regular or irregular domains. The weighted derivative approach also ensures the convergence of derivatives.

Chapter 4

Stability analysis of RDQ method using the location of zeroes of polynomials

4.1 Introduction

The stability characteristics of RDQ method are studied in this chapter by the location of zeros or roots of its characteristic polynomials with respect to unit circle in the complex plane, by discretizing the domain with either the uniform or random field nodes. This is achieved by performing the stability analysis of RDQ method by the 1st-order wave, transient heat conduction and transverse beam deflection equations via both the analytical and numerical approaches. The stability analysis of the locally applied DQ and RDQ methods is performed by different single and multi-step schemes using Von Neumann (VN) and Schur polynomials [92]. The stable schemes are identified, and their consistency analysis is performed to obtain the additional constraints on the temporal spacing. The analytical results obtained by the stability and consistency analyses of the stable schemes are effectively verified by numerically implementing the RDQ method to solve the 1st-order wave, transient heat conduction and transverse beam deflection equations with the domain discretized by either the uniform or random field nodes. As a result, it is shown that the time dependent problems can be successfully solved by the RDQ method via the systematic stability analysis through the Neumann and Schur polynomials [92], compared with the FEM and other meshless methods.

The stability and consistency analyses of the numerical schemes have been one of the major topics of interest among the research community for a long time. A lot of studies have been performed in this area for the schemes, such as the FDM and FEM and well developed theories are available. The DQ and SPH methods are relatively new techniques as compared with these schemes. In general, the “stability analysis” is composed of the two focuses. The first is to study the ill-conditioning of the stiffness matrix, and the second is to study the propagation or magnification of numerical errors with time when different schemes of the discretizations in the time and space are implemented. This

chapter is involved in the second focus of the stability analysis. In order to ensure the applicability of a numerical method in solving the time dependent problems, it becomes vital to know its behaviour for different schemes of the time and space discretizations. A computing error may be introduced in the numerical solution during an initial time increment for a specific scheme of discretization. Sometimes it may not be possible to completely eliminate the error, but it may be quite possible to restrict its growth or magnification during the time marching. Therefore, the study of the numerical error propagation or magnification is a very important aspect of the strong-form method. Since the RDQ method is one of the strong-form meshless approaches, the stability analysis is pursued in this chapter. As the 1st-order wave, transverse beam deflection, and heat conduction equations are typical examples of the hyperbolic and parabolic PDE, they are chosen for the present stability analysis.

The SPH method was proposed by Lucy [2] and Gingold and Monaghan [3], in which the function is approximated by its integral form via the window function. Balsara [93] performed the VN stability analysis of the SPH method, and suggested the optimal ranges of the parameters used in the SPH. He performed the stability analysis by applying a constant velocity to an unperturbed state, verified the results using different kernels, and drawn conclusions based on the extensive numerical analyses. Børve *et al.* [94] applied the SPH method to solve the equations of magnetohydrodynamics, and performed their stability analysis by the perturbation with small amplitudes. Swegle *et al.* [95] obtained the stability criterion of SPH in terms of the stress state because it was observed that, when the SPH method is used with the cubic B-spline kernel, it gets unstable in the tensile region but remains stable in the compressive region. They explained that the instability is caused by a negative effective modulus resulting from the kernel function interaction with the constitutive equations. In other words, the instability is caused due to the effective stress, which amplifies rather than reduces the applied strain. Sigalotti and López [96] proposed an alternate method to reduce the tensile instability by an adaptive density kernel estimation algorithm. As a result, the amount of smoothing applied to the data is controlled and the final smoothing length of the kernel is changed from point to point. Di Lisio *et al.* [97] attempted the convergence of SPH method for a generic polytropic fluid. A unified stability analysis was given by Belytschko *et al.* [98] while

addressing three types of instabilities, namely the instability due to rank deficiency, tensile, and material instability. They applied the perturbation method to momentum equation and studied Eigenvalues of the resulting matrix. Fung [99] used the DQ method to solve the 2nd-order ordinary differential equation in time. He studied the stability and accuracy properties of the DQ method by different spacings of the grid points. Tomasiello [100] performed the stability and accuracy analyses of the iterative based DQ method called the iterative differential quadrature method. Aceto and Trigliante [101] reviewed the stability properties of linear multi-step methods. Ata and Soulaïmani [102] used the SPH method to solve the shallow water equations, and improved the dynamic stability of the method by introducing the Lax-Friedrich scheme. Different approaches are discussed in the past to impose the Dirichlet boundary conditions as the SPH method does not possess the property of delta function.

When the PDE contains the terms of time and space derivatives, the stable and consistent (without oscillations or dispersion in the solution) discretization of the PDE depends on how accurately the temporal marching is performed. The criterion of temporal marching can be determined by performing the stability analysis of numerical method by the specific PDE. Therefore, it is crucial for the numerical method to appropriately carry out the stability analysis of a time dependent problem. It is shown in the presented work that the RDQ method can be used very well for the stability analysis, and it provides the stable results with the domain discretized by either the uniform or random field nodes. This capability of the RDQ method makes it suitable to solve the complex problems involving the phenomenon, such as the moving boundaries, hydrogel swelling or de-swelling, in which the solution is highly sensitive to the chosen increment in time.

The motivation behind the presented work is to comprehensively study the stability characteristics of the RDQ method by the analytical and numerical approaches with the domains discretized by either the uniform or random field nodes. Most of the research on the stability analysis performed in the past is based on either the Eigen value analysis or the perturbation theory, or the numerical results based on the uniform spacing of the field nodes. In the present novel approach, the stability analysis of the locally applied DQ and RDQ methods is demonstrated for several single and multi-step schemes of the temporal

and spatial discretization based on the VN and Schur polynomials [92, 103]. The stable schemes are identified for the 1st-order wave equation, and their consistency analysis is performed to identify the additional constraints on the temporal spacing. The analytical results of both the analyses are numerically verified by implementing the stable schemes via the RDQ method with the domains discretized by either the uniform or random field nodes. The stability analysis is further extended for the equations of transient heat conduction and transverse beam deflection. An analytical equation is developed for the problem of transverse beam deflection to compute the reduction in the successive amplitudes of beam deflection for both the implicit and explicit approaches, and it is shown by the RDQ method that the numerical reduction in the values of beam amplitude closely matches with the analytically determined values. For the transverse beam deflection, it is also demonstrated by the RDQ method that an explicit approach involves the dispersion effect, thus $(v_2/v_1) > 1.0$ and $(v_2/v_1) \rightarrow 1.0$ as $k \rightarrow 0$, and an implicit approach involves the dissipation effect, therefore $(v_2/v_1) < 1.0$ and $(v_2/v_1) \rightarrow 1.0$ as $k \rightarrow 0$, where k is the temporal spacing, v_1 and v_2 are the reference amplitudes of beam deflection at the field nodes with respect to the time t_1 and $[t_1 + (2\pi/\omega_d)]$, respectively, and ω_d is the damped natural frequency of the system. Therefore, broadly the motivation of this chapter is to study, analytically and numerically, the propagation or magnification of numerical error with increment in time, when different schemes of the space and time discretizations are implemented by the RDQ method.

The objective of the present work is to study the analytical and numerical stability characteristics of the locally applied DQ and RDQ methods in accordance with the VN and Schur polynomials. Moreover, the governing equations of the 1st-order wave, transient heat conduction, and the transverse beam deflection are solved with their stability criteria, as an application of the stability analysis by the RDQ method coupled with the domains discretized by either the uniform or random field nodes. It may not be always possible to explicitly evaluate the stability criterion for the complex PDE, therefore this limitation is overcome here by plotting the roots ϕ of the characteristic polynomial as $|\phi|^2$ versus (θ/π) , and identifying the stability criterion from $|\phi|^2$, namely

the scheme is stable if $|\phi|^2 \leq 1$ and unstable if $|\phi|^2 > 1$. Therefore, a broad framework is developed by the present work that facilitates the solution of transient PDE by the RDQ method via their detailed stability analysis.

It is shown in the present work that the time dependent problems can be very well solved by the RDQ method via their stability analysis, as compared with the other existing meshless methods and the FEM, as no mesh is involved. As a result, the RDQ method can be effectively used in the time dependent problems (linear and nonlinear), such as hydrogel swelling and crack propagation.

The outline of the subsequent subsections of this chapter is as follows. The VN and Schur polynomials are explained in the subsection 2, where detailed stability analysis of the 1st-order wave equation is performed by the locally applied DQ and RDQ methods. The stable schemes are identified, and their elaborate consistency analysis is performed to obtain the additional constraints on the temporal spacing. In the subsection 3, the stability analysis is further extended to solve the transient heat conduction equation by the RDQ method with the spatial domain discretized by either the uniform or random field nodes. The transverse beam deflection equation is solved by the RDQ method in the subsection 4 with the implicit and explicit approaches, and the numerical reduction of amplitude in the beam deflection is compared with the corresponding analytical values. Finally, the summary is given in the subsection 5.

4.2 Stability analysis of the 1st-order wave equation by RDQ method

In this section, the stability analysis of the 1st-order wave equation $\phi_t = a\phi_{,x}$ is performed by the RDQ method with several single and multi-step schemes. The single-step schemes are studied by the VN stability analysis [92], and the multi-step schemes by Schur and Neumann polynomials [92, 103].

The VN and Schur polynomials appear frequently in the studies of PDE. Let us consider a general polynomial represented as $\phi(z) = \sum_{l=0}^d a_l z^l$, where d is the order of the polynomial and a_l are the coefficients. The polynomial $\phi(z)$ is called the Schur polynomial, if all its roots are located within the circle of unit radius, namely $|\phi_i| < 1.0$,

where $i = 1, 2, \dots, d$. The polynomial $\phi(z)$ is called the VN polynomial, if some of the roots are located within the circle of unit radius and the remaining are on the circle but none is outside of it, namely $|\phi_i| \leq 1.0$, where $i = 1, 2, \dots, d$.

In the Neumann analysis, the amplification factor $g(h\xi)$ of the discretized wave equation is derived by taking the inverse Fourier transform and simplifying it to get $\hat{v}_n(\xi) = g(h\xi) \hat{v}_0(\xi)$, where h is the spatial spacing, $\xi \in [-\pi, \pi]$ is the Fourier domain, and $\hat{v}_n$ and $\hat{v}_0$ are the values of function at the n^{th} and 0^{th} level of time, respectively [92]. The amplification factor $g(h\xi)$ is further analyzed for the stability criterion. In Schur and Neumann polynomial analyses, the amplification or characteristic polynomial is derived by discretizing the wave equation, and the roots of this polynomial are examined to classify the polynomial as either Schur $P(n, 0, 0)$ or VN $P(k, n-k, 0)$, [103] where n are the total roots of polynomial out of which k roots lie within the circle of unit radius on the complex plane.

4.2.1 Stability analysis of the 1st-order wave equation by different schemes for discretization of domains

Different schemes of the domain discretization are studied in this section for the stability of the 1st-order wave equation and the stable schemes are identified. The constraints or conditions obtained from the roots of the polynomial of stable schemes are the stability criteria of the wave equation.

4.2.1.1 Central time and central space multi-step scheme

In this scheme, three virtual nodes are considered in the stencil of time and space domains, respectively, as shown in Fig. 4.1a. The 1st-order wave equation $\phi_t = a\phi_{,x}$ in the discretized form is given as

$$\begin{Bmatrix} a_{j,j-1} & a_{j,j} & a_{j,j+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_{j-1}) \\ \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i-1} & a_{i,i} & a_{i,i+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_{i-1}, t_j) \\ \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \end{Bmatrix} \quad (4.1)$$

The weighting coefficients in Eq. (4.1) are computed by the DQ method, as explained earlier. The inverse Fourier transform is employed in the resulting equation to get the characteristic polynomial as (please refer Appendix A)

$$\phi(z) = z^2 - [2 r i \sin(\theta)] z - 1 \quad (4.2)$$

where $r = (a t / h)$ is Courant number. Comparing Eq. (4.2) with the generalized polynomial given in Eq. (4.3), the coefficients of Eq. (4.2) are given in Eq. (4.4) as

$$\phi(z) = \sum_{l=0}^d a_l z^l \quad (4.3)$$

$$a_0 = -1, \quad a_1 = 2 r i \sin(\theta), \quad \text{and} \quad a_2 = 1 \quad (4.4)$$

where d is the order of polynomial. The complex conjugate polynomial of Eq. (4.2) is given in Eq. (4.6) by Eq. (4.5) as

$$\phi^*(z) = \sum_{l=0}^d \overline{a_{d-l}} z^l \quad (4.5)$$

$$\phi^*(z) = 1 + (2 r i \sin \theta) z - z^2 \quad (4.6)$$

It is observed from Eqns. (4.2) and (4.6) that

$$|\phi^*(0)| = 1 \text{ and } |\phi(0)| = 1, \text{ so } \phi_1(z) = \frac{\phi_0^*(0) \phi(z) - \phi(0) \phi_0^*(z)}{z} = 0 \quad (4.7)$$

where $\phi_1(z)$ is a reduced polynomial (Please refer Appendix B). As $\phi_1(z) = 0$, $\phi(z)$ is a self-inversive polynomial [103]. As per the theorem given by Miller [103], a self-inversive polynomial $\phi(z)$ is the VN polynomial if $\phi^{(1)}(z)$ is the VN polynomial, where $\phi^{(1)}(z)$ is the 1st-order derivative of $\phi(z)$. Therefore, the 1st-order derivative of Eq. (4.2) is written as

$$g(z) = \phi^{(1)}(z) = 2 z - 2 r i \sin(\theta) \quad (4.8)$$

The complex conjugate polynomial of Eq. (4.8) is given as

$$g^*(z) = 2 + 2 r i z \sin(\theta) \quad (4.9)$$

It is seen from Eqns. (4.8) and (4.9) that $|g^*(0)| = 2$ and $|g(0)| = 2 r i \sin(\theta)$. The condition $|g^*(0)|^2 - |g(0)|^2 \geq 0$ must hold true for Eq. (4.8) to be the VN polynomial, which results in

$$|r| \leq 1 \quad (4.10)$$

This shows that Eq. (4.10) must be satisfied for Eq. (4.8) to be the VN polynomial. Therefore, if Eq. (4.10) is satisfied, $g(z)$ becomes the VN polynomial and consequently Eq. (4.2) becomes the VN polynomial. As a result, it is concluded that Eq. (4.10) is the stability condition for this scheme. As this scheme is multi-step in the time domain, it needs some other scheme to initiate the computation, such that the forward time (two nodes in the time domain) and central space (three nodes in the space domain) scheme can be used as an initialization scheme [92]. The initialization scheme does not affect the stability of multi-step schemes as the amplification factor is controlled by the stability criterion, such that even if there is any instability in the solution due to initialization scheme, it does not propagate with time.

4.2.1.2 Central time and forward space multi-step scheme

In this scheme, three virtual nodes are considered in the time and space domains, respectively, as shown in Fig. 4.1b.

Fig. 4.1. Local DQ domain around a central virtual node (x_i, y_j) that is inside computational domain (a), and central time and forward space scheme of discretization of the time and space domains (b) for the 1st-order wave equation.

The 1st-order wave equation in the discretized form is given as

$$\phi_{,t} = a \phi_{,x} \Rightarrow \left\{ a_{j,j-1} \quad a_{j,j} \quad a_{j,j+1} \right\} \begin{Bmatrix} \phi(x_i, t_{j-1}) \\ \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \left\{ a_{i,i} \quad a_{i,i+1} \quad a_{i,i+2} \right\} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \\ \phi(x_{i+2}, t_j) \end{Bmatrix} \quad (4.11)$$

The weighting coefficients are computed by the DQ method, and the characteristic polynomial is derived by applying the inverse Fourier transform to obtain [103] (Please refer Appendix A)

$$\phi(z) = z^2 + (\alpha + i\beta) z - 1 \quad (4.12)$$

where $\gamma = \cos(\theta) - 2$, $\alpha = 2r [1 + \cos(\theta) \gamma]$, $\beta = 2r \sin(\theta) \gamma$. Comparing Eq. (4.12) with Eq. (4.3), the coefficients of Eq. (4.12) are given as

$$a_0 = -1, \quad a_1 = \alpha + i\beta, \quad \text{and} \quad a_2 = 1 \quad (4.13)$$

The complex conjugate polynomial of Eq. (4.12) is given as

$$\phi^*(z) = 1 + (\alpha - i\beta) z - z^2 \quad (4.14)$$

It is observed from Eqns. (4.12) and (4.14)

$$|\phi^*(0)| = 1 \text{ and } |\phi(0)| = 1, \therefore \phi_1(z) = 0 \quad (4.15)$$

where $\phi_1(z)$ is a reduced polynomial (Please refer Appendix B). $\phi(z)$ is a self-inversive polynomial, as $\phi_1(z) = 0$. Therefore, Eq. (4.12) is the VN polynomial only if $\phi^{(1)}(z)$ is the VN polynomial [103], where $\phi^{(1)}(z)$ is the 1st-order derivative of $\phi(z)$ given as

$$g(z) = \phi^{(1)}(z) = 2z + (\alpha + i\beta) \quad (4.16)$$

$$g^*(z) = 2 + (\alpha - i\beta) z \quad (4.17)$$

It is noted from Eqns. (4.16) and (4.17) that $|g^*(0)| = 2$ and $|g(0)| = \alpha + i\beta$. The condition $|g^*(0)|^2 - |g(0)|^2 \geq 0$ must hold true for Eq. (4.17) to be the VN polynomial, therefore this results in

$$r^2 [1 + \gamma^2 + 2\gamma \cos(\theta)] \leq 1 \quad (4.18)$$

where $|1 + \gamma^2 + 2\gamma \cos(\theta)| \leq 1$ for $|r| \leq 1$. The maximum value of $[1 + \gamma^2 + 2\gamma \cos(\theta)]$ is at $\cos(\theta) = (4/3)$, which gives

$$|r| \leq 3i \quad (4.19)$$

Therefore, Eq. (4.19) is the stability condition for this scheme. It is seen that Eq. (4.19) is a complex number, however it is impossible to have a complex number as the stability criterion. As a result, $g(z)$ is not the VN polynomial, and consequently $\phi(z)$ is also not the VN polynomial. Therefore, it is concluded that this scheme is unstable. Let us consider ϕ_+ and ϕ_- as the roots of Eq. (4.12). A graph of $|\phi_+|^2$ versus (θ/π) is plotted and ϕ_+ is found to be unconditionally stable, namely $|\phi|^2 \leq 1$ even for $r > 1$, and a graph of $|\phi_-|^2$ versus (θ/π) is plotted and ϕ_- is found to be unstable, namely $|\phi|^2 > 1$ even for $r < 1$, as shown in Fig. 4.2a and Fig. 4.2b respectively, by considering $0 \leq r \leq 2$ and $0 \leq r \leq 1$, respectively.

Fig. 4.2. $|\phi_+|^2$ versus (θ/π) plot with r from 0 to 2 (a) and $|\phi_-|^2$ versus (θ/π) plot with r from 0 to 1 (b) by numerically implementing the 2nd stability scheme.

4.2.1.3 Forward time and forward space multi-step scheme

In this scheme, three virtual nodes are considered in the stencils of time and space domains, respectively, as shown in Fig. 4.3a.

The 1st-order wave equation $\phi_{,t} = a\phi_{,x}$ in the discretized form is given as

$$\begin{Bmatrix} a_{j,j} & a_{j,j+1} & a_{j,j+2} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \\ \phi(x_i, t_{j+2}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i} & a_{i,i+1} & a_{i,i+2} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \\ \phi(x_{i+2}, t_j) \end{Bmatrix} \quad (4.20)$$

The weighting coefficients from Eq. (4.20) are computed by the DQ method, and the characteristic polynomial is given as (Please refer Appendix A)

$$\phi(z) = z^2 - 4z + 3 - (\alpha + i\beta) \quad (4.21)$$

where $\gamma = \cos(\theta) - 2$, $\alpha = 2r [1 + \cos(\theta) \gamma]$, $\beta = 2r \sin(\theta) \gamma$. The complex conjugate polynomial of Eq. (4.21) is given as

Fig. 4.3. Multi-step discretization scheme of forward time and space domains (a), plot of $|\phi_+|^2$ versus (θ/π) (b) and $|\phi_-|^2$ versus (θ/π) (c) by numerically implementing this scheme with r from 0 to 1 for the 1st-order wave equation.

$$\phi^*(z) = 1 - 4z + (3 - \alpha + i\beta)z^2 \quad (4.22)$$

The conditions $|\phi^*(0)| = 1$ and $|\phi(0)| = 3 - \alpha - i\beta$ are obtained from Eqns. (4.21) and (4.22). For Eq. (4.21) to be the VN polynomial, $\left[|\phi^*(0)|^2 - |\phi(0)|^2 \right] \geq 0$ results in [103]

$$1 - (3 - \alpha)^2 - \beta^2 \geq 0 \quad (4.23)$$

It is difficult to analytically solve Eq. (4.23), therefore the roots of $\phi(z)$ are studied by plotting the graph of $|\phi|^2$ versus (θ/π) by incrementing r from 0 to 1 and angle θ from 0 to 2π . The graph of $|\phi_+|^2$ versus (θ/π) shows that, $|\phi_+|^2$ has the lowest magnitude of 9, as shown in Fig. 4.3b. The graph of $|\phi_-|^2$ versus (θ/π) shows that $|\phi_-|^2$ is stable for $|r| \leq 1$, as shown in Fig. 4.3c. As a result, this scheme is unstable due to an instability in $|\phi_+|$, as $|\phi_+|^2 \geq 9$.

4.2.1.4 Forward time and forward space single step scheme

In this scheme, two and three virtual nodes are used in the time and space domains, respectively, as shown in Fig. 4.4a. The 1st-order wave equation in the discretized form is given as

$$\phi_{,t} = a\phi_{,x} \Rightarrow \begin{Bmatrix} a_{j,j} & a_{j,j+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i} & a_{i,i+1} & a_{i,i+2} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \\ \phi(x_{i+2}, t_j) \end{Bmatrix} \quad (4.24)$$

The characteristic polynomial of Eq. (4.24), by computing the DQ weighting coefficients, is given as (Please refer Appendix A)

$$\phi(z) = z - (1 + \alpha + i\beta) \quad (4.25)$$

where $\gamma = 2 - \cos(\theta)$, $\alpha = r [\cos(\theta) \gamma - 1]$ and $\beta = r \sin(\theta) \gamma$. The complex conjugate polynomial of Eq. (4.25) is given as

$$\phi^*(z) = - (1 + \alpha - i\beta) z + 1 \quad (4.26)$$

It is noted from Eqns. (4.25) and (4.26) that $|\phi^*(0)|=1$ and $|\phi(0)|=-(1+\alpha+i\beta)$. The condition $\left[|\phi^*(0)|^2-|\phi(0)|^2\right]\geq 0$ must hold true for Eq. (4.25) to be the VN polynomial, which results in

$$1-\left[(1+\alpha)^2+\beta^2\right]\geq 0 \quad (4.27)$$

Simplifying Eq. (4.27) results in

$$|r|\leq \frac{2}{7} \text{ and } |r|>0 \quad (4.28)$$

Fig. 4.4. Single step scheme of forward time and forward space domains (a), and the plot of $|\phi|^2$ versus (θ/π) (b) with r from 0 to 1, in which the root is inside the unit circle when $|r|\leq 0.2$ for the discretization of the 1st-order wave equation.

Therefore, Eq. (4.28) is the stability condition for this scheme. The plot of $|\phi|^2$ versus (θ/π) by incrementing r from 0 to 1 and angle θ from 0 to 2π also shows that the scheme is stable for $|r|\leq 0.2$ which is very close to Eq. (4.28), as shown in Fig. 4.4b. This scheme is nothing but an upwind technique, as the wave $\phi_{,t} = a \phi_{,x}$ is traveling towards the left side of the computational domain.

4.2.1.5 Forward time and central space single step scheme

In this scheme, two and three virtual nodes are used in the time and space domains, respectively, as shown in Fig. 4.5a. The 1st-order wave equation in the discretized form is given as

$$\phi_{,t} = a\phi_{,x} \Rightarrow \begin{Bmatrix} a_{j,j} & a_{j,j+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i-1} & a_{i,i} & a_{i,i+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_{i-1}, t_j) \\ \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \end{Bmatrix} \quad (4.29)$$

The amplification factor of Eq. (4.29) after computing the DQ weighting coefficients is given as

$$g^{n+1} = \left\{ \frac{r}{2} [2i \sin(\theta)] + 1 \right\} g^n \quad (4.30)$$

$$\therefore g(\theta) = 1 + i r \sin(\theta) \quad (4.31)$$

Eq. (4.31) is found unstable when it is plotted in the complex plane, as shown in Fig. 4.5b. If $\phi^m = 0.5(\phi_{m-1}^n + \phi_{m+1}^n)$ is substituted in Eq. (4.29), similar to Lax-Friedrich scheme in the FDM, it is rewritten as

$$\phi_m^{n+1} = \frac{r}{2} [\phi_{m+1}^n - \phi_{m-1}^n] + \left[\frac{\phi_{m-1}^n + \phi_{m+1}^n}{2} \right] \quad (4.32)$$

Fig. 4.5. Single step scheme of forward time and central space domains (a) and the plot of amplification factor in the complex plane with a circle having a unit radius (b), where the scheme is found unstable for the discretization of the 1st-order wave equation.

The new amplification factor from Eq. (4.32) is given as

$$g(\theta) = \cos \theta + i r \sin \theta, \quad \therefore |g(\theta)| \Rightarrow |r| \leq 1 \quad (4.33)$$

It is concluded from Eq. (4.33) that the scheme is stable for $|r| \leq 1$.

4.2.2 Consistency analysis of the stable schemes and their verification by numerically implementing the 1st-order wave equation by the locally applied DQ method

In order to obtain the stable discretization of the 1st-order wave equation, it is essential to ensure that the constraints on Courant number r , obtained from the stability analysis, does not contradict with those obtained from the consistency analysis. To fulfill this requirement therefore, it is essential to perform the consistency analysis of the stable schemes, which are obtained in the previous section. As a result, the final condition of constraint is obtained by combining the results obtained from the stability and consistency analyses. The stable schemes are then numerically implemented to solve the 1st-order wave equation $\phi_t = a \phi_{,x}$ with $\phi(x, 0) = \sin(2\pi x)$ and $\phi(0, t) = \phi(1, t) = 0$ when $x \in [0, 1]$ as the initial and periodic boundary conditions, respectively.

Let us consider the PDE in the continuous and discretized forms as $Pu = f$ and $P(h_x, h_y)u = f$, respectively, where $h_x = L_x / (NP_x - 1)$ and $h_y = L_y / (NP_y - 1)$ are the nodal spacings, L_x and L_y are the domain lengths, and NP_x and NP_y are the numbers of total field nodes in the x and y directions, respectively. The numerical discretization scheme is considered consistent, if the discretized form of PDE closely approximates the continuous form as h_x and $h_y \rightarrow 0$, namely $Pu - P(h_x, h_y)u \rightarrow 0$ as $h_x, h_y \rightarrow 0$, such that there is no discretization error involved in the numerical simulation.

The discretized form of the 1st-order wave equation by the central time and space scheme (Scheme 1) is given in Eq. (4.1). As such, the terms of function values in Eq. (4.1) are substituted by Taylor series expansion [80] and written as (Please refer Appendix C)

$$\phi_{,t} - a \phi_{,x} = \phi_{,xxx} \left[\frac{a h^2}{6} - \frac{t^2 a^3}{6} \right] \quad (4.34)$$

This scheme is consistent as $\phi_{,t} - a \phi_{,x} \rightarrow 0$ when h and $t \rightarrow 0$. Eq. (4.34) is the exact or actual wave equation that is numerically solved. The term $\phi_{,xxx}$ in Eq. (4.34) represents the numerical dispersion with respect to the term of space domain $\phi_{,x}$ from the wave equation, therefore it is desirable to have a negative or zero value for $\phi_{,xxx}$ to reduce the oscillations in the solution as

Fig. 4.6. The wave is preserved for $r=1$ (a) and $r=0.5$ (b), but the oscillations are observed for $r=2$ (c), when the 1st-order wave equation is solved by locally applied DQ method with the 1st discretization scheme.

$$\left[\frac{a h^2}{6} - \frac{t^2 a^3}{6} \right] \leq 0 \quad (4.35)$$

Simplifying Eq. (4.35) results in

$$|r| \geq 1 \quad (4.36)$$

where $r = (at/h)$. It is concluded from Eq. (4.36) that the consistency condition matches with the stability condition, as given in Eq. (4.10), for $|r| = 1$. This scheme is numerically implemented to solve the 1st-order wave equation. The wave is preserved during the total time of simulation for $r=1$ and no instability is observed, as shown in Fig. 4.6a. The initial wave is also preserved throughout the time of simulation for $r=0.5$ and no dissipation or the damping is observed, as shown in Fig. 4.6b. This is expected as no dissipation term is present in Eq. (4.34). However, the instability is observed for $r=2$ due to the violation of stability condition from Eq. (4.10), as shown in Fig. 4.6c.

The discretized form of the wave equation by the forward time and forward space scheme (scheme 4) is given in Eq. (4.24). As such, the terms of the function values are substituted by Taylor series expansion [80] and written as (Please refer Appendix C)

$$\phi_{,t} - a \phi_{,x} = -\phi_{,xx} \left[\frac{a h r}{2} \right] \quad (4.37)$$

This scheme is consistent as $\phi_{,t} - a \phi_{,x} \rightarrow 0$ when h and $t \rightarrow 0$. It is noted from Eq. (4.37) that there is negative dissipation effect (removal of dissipation from the system) for the positive wave speed a , as the term $\phi_{,xx}$ represents the dissipation or the damping corresponding to the term $\phi_{,x}$ from the wave equation. As such, this may not be completely avoided irrespective of the values of h and r . In other words, even if this scheme is stable from the aspects of stability, there is going to be some negative dissipation effect due to the +ve wave speed a from the point of view of consistency that will result in the oscillations. There is going to be a positive dissipation effect due to the –ve wave speed a that will result in the damping of wave. This observation is verified by numerically implementing this scheme to solve the 1st-order wave equation. The results are stable for $r = 0.2$ as the stability condition given in Eq. (4.28) is satisfied, but slight instability is added due to the negative dissipation effect, as given in Eq. (4.37) and shown in Fig. 4.7a. The wave becomes unstable for $r = 0.5$, as shown in Fig. 4.7b, due

to the violation of stability condition. The instability in the wave is identified when the maximum value of function along the y axis exceeds 1.

The wave equation is discretized by the forward time and central space scheme (scheme 5) given in Eq. (4.29), and the terms of the function values are substituted by Taylor series expansion [80] and written as (Please refer Appendix C)

$$\phi_{,t} - a \phi_{,x} = \phi_{,xx} \left[\frac{h^2}{t} - \frac{a h r}{2} \right] \quad (4.38)$$

Fig. 4.7. Small effect of dispersion is observed for $r = 0.2$ (a) and the oscillations are observed for $r = 0.5$ (b), when the 1st-order wave equation is solved by the 4th-scheme.

This scheme is consistent as $\phi_{,t} - a \phi_{,x} \rightarrow 0$ when h and $t \rightarrow 0$. As the term $\phi_{,xx}$ represents the dissipation or damping with respect to the term of space domain $\phi_{,x}$ from the governing wave equation, it is desirable to have a positive or zero value for the coefficient of $\phi_{,xx}$ as

$$\left[\frac{h^2}{t} - \frac{a h r}{2} \right] \geq 0 \quad (4.39)$$

Simplifying Eq. (4.39) results in

$$|r| \leq \sqrt{2} \quad (4.40)$$

Eq. (4.40) gives the constraint on r by the approach of consistency. It is clear by comparing Eq. (4.33) with Eq. (4.40) that, if the stability condition $|r| \leq 1$ is satisfied, automatically the consistency condition $|r| \leq \sqrt{2}$ is also satisfied, but the vice a versa is not true. As a result, the stability condition is taken as the final criteria.

This scheme is implemented to numerically solve the 1st-order wave equation. The results are stable for $r=1$ and the initial wave is preserved for all the time steps, as shown in Fig. 4.8a. The dissipation effect is observed for $r = 0.5$, which is expected from the consistency results as given in Eq. (4.38), although the results are stable, as shown in Fig. 4.8b. The instability is observed for $r = 2$, as shown in Fig. 4.8c, due to the violation of both the stability and consistency conditions.

In summary, all the schemes that are found stable in the previous section are analyzed for the consistency. The analytical results obtained from the stability and consistency analyses are successfully verified by numerically solving the 1st-order wave equation by the locally applied DQ method. It is observed that the numerical behaviour of the method is similar to what was analytically predicted. The 1st-order wave equation is further solved by the RDQ method in the next section with Scheme 5 to test whether the RDQ method satisfies the analytical conditions obtained so far from the stability and consistency analyses.

4.2.3 Implementation of the RDQ method for the 1st-order wave equation by the forward time and central space scheme

The RDQ method is implemented in this section to solve the 1st-order wave equation by the forward time and central space scheme (scheme 5).

A local domain of interpolation is created around each virtual node, as shown in Fig. 4.9, and the nearby field nodes falling in it are considered for the approximation of function value at the concerned virtual node. As the fixed RKPM interpolation function does not possess the property of delta function, a second level interpolation is performed. A local domain is created around each field node, in which the nearby surrounding field nodes are considered for the approximation of function at the concerned field node. The

unknown values of nodal parameter at the field nodes are computed for the time $t = 0$ by the initial condition as

$$\{\phi\}_{M \times 1}^0 = [N]_{M \times M} \{U\}_{M \times 1}^0 \quad (4.41)$$

where 0 indicates $t = 0$, M is the number of total field nodes in whole domain, $\{\phi\}_{M \times 1}^0$ is a known vector of function values at all the field nodes at the time $t = 0$, $[N]_{M \times M}$ is a matrix of shape functions representing the interpolation of function values at the field nodes by the surrounding field nodes, and $\{U\}_{M \times 1}^0$ is the unknown vector of nodal parameter values at all the field nodes at the time $t = 0$. The values of function at all the virtual nodes are then computed at the time $t = 0$ as

Fig. 4.8. The wave is preserved for $r = 1$ (a), numerical damping is observed for $r = 0.5$ (b) and the oscillations are observed for $r = 2$ (c) when the 1st-order wave equation is solved by the locally applied DQ method with the 5th scheme.

$$\{\phi\}_{N \times 1}^0 = [N]_{N \times M} \{U\}_{M \times 1}^0 \quad (4.42)$$

where N is the number of total virtual nodes in a whole domain, $\{\phi\}_{N \times 1}^0$ is the unknown vector of function values for all the virtual nodes at $t = 0$, $[N]_{N \times M}$ is a matrix of shape functions interpolating the function values at all the virtual nodes by the function values at the surrounding field nodes, and $\{U\}_{M \times 1}^0$ is the known vector of nodal parameter values at all the field nodes that is already computed by Eq. (4.42). The values of function at all the virtual nodes for all the time increments are computed as

$$\{\phi\}_{N \times 1}^t = [A]_{N \times N} \{\phi\}_{N \times 1}^{t-1} \quad (4.43)$$

where $[A]_{N \times N}$ is a coefficient matrix of the discretization scheme from Eq. (4.32), $\{\phi\}_{N \times 1}^t$ and $\{\phi\}_{N \times 1}^{t-1}$ are the vectors of function values for all the virtual nodes at the time t and $t - 1$, respectively. Eqns. (4.42) and (4.41) are used successively after computing the values of function at all the virtual nodes to compute the unknown values of nodal parameter and function at all the field nodes, respectively.

Fig. 4.9. Local interpolation of the function and the creation of DQ local domains for the discretization of governing equation at all virtual nodes by the RDQ method.

4.2.4 Remarks on the solution of the 1st-order wave equation by the RDQ method

The 1st-order wave equation is solved by the RDQ method, as discussed in the earlier subsection, with the cosine distribution of virtual nodes and the uniform distribution of field nodes in the computational domain. The numerical results are analyzed by the consistency conditions of the wave equation. The wave equation in the discretized form is given as

$$\begin{Bmatrix} a_{j,j} & a_{j,j+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i-1} & a_{i,i} & a_{i,i+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_{i-1}, t_j) \\ \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \end{Bmatrix} \quad (4.44)$$

If the time and space domains contain the uniform and cosine distributed virtual nodes, respectively, Eq. (4.44) is simplified by computing the weighting coefficients for the time domain using Shu's general approach [37] as discussed earlier and given as

$$\frac{1}{t} \begin{Bmatrix} -1 & 1 \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i-1} & a_{i,i} & a_{i,i+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_{i-1}, t_j) \\ \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \end{Bmatrix} \quad (4.45)$$

where t is the time increment. Eq. (4.45) is simplified by rearranging the terms, and the final consistency equation $P u - P(h_x, h_y) u$, by subtracting Eq. (4.45) from the 1st-order wave equation for the scheme 5, is given as

$$\phi_{,t} - a\phi_{,x} - \phi_{,t} + a\phi_{,x} \left[\frac{Bh_2 - Ah_1}{a t} \right] = \phi \left[\frac{1 - A - B}{t} \right] + \phi_{,xx} \left[\frac{Ah_1^2 + Bh_2^2}{2t} + \frac{a^2 t}{2} \right] \quad (4.46)$$

where h_1 and h_2 are the spacings between the virtual nodes ϕ_m^n and ϕ_{m-1}^n , and ϕ_m^n and ϕ_{m+1}^n , respectively, and A and B are the constants given as $A = 0.5 + (0.5 a t a_{i,i}) + (a t a_{i,i-1})$ and $B = 0.5 + (0.5 a t a_{i,i}) + (a t a_{i,i+1})$, respectively. Following conditions are derived for Eq. (4.46) to be consistent, namely for $P u - P(h_x, h_y) u \rightarrow 0$ to be true

$$\frac{Bh_2 - Ah_1}{at} = 1, \frac{1 - A - B}{t} = 0, \text{ and } \left[\frac{Ah_1^2 + Bh_2^2}{2t} + \frac{a^2 t}{2} \right] \geq 0 \text{ for +ve or zero dissipation (4.47)}$$

Eq. (4.47) provide the constraints, which when satisfied results in the consistent discretization of the wave equation by the uniform and cosine distributions of the virtual nodes in the time and space domains, respectively.

In order to verify the satisfaction of Eq. (4.47) by the uniform distribution of virtual nodes in the spatial domain, the $\phi_t = a\phi_x$ is numerically solved by the RDQ method with the spatial domain discretized separately by the uniform and random field nodes, and the time domain discretized by uniform field nodes coupled with the uniform virtual nodes in both the time and space domains, respectively. It is seen from Fig. 4.10a and Fig. 4.10b that the constraints given in Eq. (4.47) are exactly satisfied, which results in the consistent discretization of the 1st-order wave equation.

When the 1st-order wave equation is solved by RDQ method with the spatial domain discretized by the uniform and random field nodes coupled with the cosine virtual nodes, and the temporal domain discretized by the uniform field nodes coupled with the uniform virtual nodes, the constraints given in Eq. (4.47) are not fully satisfied. This clearly shows that some amount of dissipation or the damping is involved, as indicated by Eq. (4.47), when the spatial domain has the cosine distributed virtual nodes. This is verified by tracking the numerical and analytical values of function for the field node located at $x = 1$ with the time, as shown in Fig. 4.10c.

In summary, several single and multi-step schemes of the discretization of time and space domains are studied in this section by the VN and Schur polynomials to achieve a stable discretization of the 1st-order wave equation, and the stable schemes are identified. The identified stable schemes are further studied for their consistency by discretizing the wave equation. The analytically predicted characteristics of the stable schemes are numerically verified by solving the 1st-order wave equation with the locally applied DQ and RDQ methods. It is seen with the 5th scheme of the discretization that the numerical dissipation is involved when the wave equation is solved by the RDQ method with the cosine distribution of virtual nodes in the space domain. But, the wave is preserved during all the time steps by the uniform distribution of virtual nodes in the space domain.

This is a very important result which highlights the applicability of RDQ method when the computational domain is discretized by random field nodes.

Fig. 4.10. The comparison between numerical and analytical function values with the spatial domain discretized by uniform (a) and random (b) field nodes coupled with the uniform virtual nodes in both the time and space domains, and the time history plot of comparison between the numerical and analytical function values at the field node $(1, t)$ when the spatial domain is discretized by the uniform and random field nodes coupled with the cosine virtual nodes (c).

As a result, in order to accomplish the stable and consistent discretization by the RDQ method, the spatial domain is discretized by either the uniform or random field nodes coupled with the uniform virtual nodes, and the temporal domain is discretized by the uniform field nodes coupled with the uniform virtual nodes.

4.3 Stability analysis of transient heat conduction equation

The transient heat conduction equation $u_{,t} = \alpha u_{,xx}$, where α is the thermal conductivity of material, is analyzed in this section for the stability and consistency, and numerically solved by the RDQ method with the spatial domain discretized by either the uniform or random field nodes.

Several discretization schemes of the time and space domains for the transient heat conduction equation are evaluated at first for the stability by the RDQ method, and the stable schemes are identified. The consistency analysis of the stable schemes is then performed to ensure that the constraints obtained by the consistency analysis are not conflicting with those obtained by the stability analysis. The transient heat conduction equation is finally solved by the RDQ method with the stable and consistent scheme.

4.3.1 Forward time and forward space scheme

In this scheme, three and two virtual nodes are used in the stencils of the space and time domains, respectively, as shown in Fig. 4.4a. The transient heat conduction equation is discretized at the uniformly distributed virtual nodes in the time and space domains with the weighting coefficients computed by Shu's general approach as

$$\frac{\partial u(x_i, t_j)}{\partial t} = \alpha \frac{\partial^2 u(x_i, t_j)}{\partial x^2} \Rightarrow \left[u(x_i, t_{j+1}) - u(x_i, t_j) \right] = \left(\frac{\alpha k}{h^2} \right) \left[u(x_i, t_j) - 2 u(x_{i+1}, t_j) + u(x_{i+2}, t_j) \right] \quad (4.48)$$

where k and h are the spacings between the virtual nodes in the temporal and spatial domains, respectively. The terms u in Eq. (4.48) are substituted by Taylor series [80] and simplified to give the discretized form as (please refer Appendix C)

$$u_{,t} - \alpha u_{,xx} = \left(\frac{-k}{2} \right) u_{,tt} - (2\alpha h) u_{,xxx} \quad (4.49)$$

The terms $u_{,tt}$ and $u_{,xxx}$ represent the dissipation with respect to the terms of time and space domains $u_{,t}$ and $u_{,xx}$, respectively. It is observed that both the dissipation terms have negative coefficients, which means that there is going to be negative dissipation

(dissipation removal from the system), therefore this scheme is not stable. As the equation of heat conduction is inconsistently discretized by this scheme, there is no need to further carry out the stability analysis.

The equation of heat conduction is diffusive in behaviour, therefore the values of temperature at the field nodes are governed by the boundary conditions. This can be captured by including the virtual nodes from both, the left and right, sides of the central concerned virtual node in the stencil of the space discretization. Therefore, the spatial stencil is changed in the next subsection by considering the virtual nodes on both sides of the concerned virtual node.

4.3.2 Forward time and central space scheme

The forward time and central space scheme is used in this section to discretize the transient heat conduction equation by the uniform and cosine distributions of the virtual nodes with the spatial domain discretized by the field nodes distributed uniformly and randomly. The stability analysis is firstly performed by the uniform distribution of virtual nodes in the space and time domains, then the consistency analysis is performed with the feasible working range of the temporal spacing k is obtained by combining both the results. The consistency analysis is further extended to the cosine distributed virtual nodes in the space domain, and the constraint on k is derived, which gives the stable discretization. The transient heat conduction equation is finally numerically solved by the RDQ method with the space domain discretized by either the uniform or random field nodes combined with either the uniform or cosine virtual nodes, and the time domain discretized by the uniform field nodes combined with the uniform virtual nodes. It is observed that the stable and accurate numerical solution of the transient heat conduction equation by the RDQ method follows the constraints on the temporal spacing k obtained from the consistency and stability analyses. The numerical results of the temperature distribution are compared with the corresponding analytical and FEM solutions, and it is noted that they are almost exactly matching with the analytical solutions.

4.3.2.1 Stability analysis of the forward time and central space scheme

In this scheme, three and two virtual nodes are considered for the discretization stencils of the space and time domains, respectively, as shown in Fig. 4.5a. The discretized form of the heat conduction equation by the RDQ method is given as

$$\left[u(x_i, t_{j+1}) - u(x_i, t_j) \right] = \left(\frac{\alpha k}{h^2} \right) \left[u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j) \right] \quad (4.50)$$

The inverse Fourier transform is performed on Eq. (4.50), and the characteristic or amplification polynomial is given as (Please refer Appendix A)

$$\phi(\theta) = 1 - 4m \sin^2\left(\frac{\theta}{2}\right), \text{ where } m = \left(\frac{\alpha k}{h^2} \right) \quad (4.51)$$

It is observed in Eq. (4.51) that $|\phi| \leq 1$ for $m \leq 0.5$ (obtained from MATLAB), therefore

$$m = \frac{1}{2} \leq \frac{\alpha k}{h^2} \Rightarrow k \leq \frac{h^2}{2\alpha} \quad (4.52)$$

Eq. (4.52) is the stability condition for the stable discretization of transient heat conduction equation by this scheme. The graph of $|\phi|^2$ versus (θ/π) is plotted in Fig. 4.11a to study the value of Eq. (4.51) by changing m as $0 \leq m \leq 1/2$, and it is observed that $\phi(\theta) \leq 1.0$. As a result, this scheme is stable with Eq. (4.52) as the stability condition.

4.3.2.2 Consistency analysis of forward time and central space scheme

Let us begin the consistency analysis with Eq. (4.50), and the terms of function value u in Eq. (4.50) are substituted by Taylor series [80] and simplifying it results in (please refer Appendix C)

$$u_{,t} - \alpha u_{,xx} = \frac{\alpha h^2}{12} u_{,xxxx} - \frac{k}{2} u_{,tt} - \frac{k^2}{2} u_{,ttt} \quad (4.53)$$

where h and k are the spacings between the virtual nodes situated in the space and time domains, respectively. The terms $u_{,xxxx}$, $u_{,tt}$ and $u_{,ttt}$ in Eq. (4.53) represent the dispersion

corresponding to the space domain term $u_{,xx}$, and the dissipation and dispersion corresponding to the term $u_{,t}$ in the time domain, respectively. The dissipation or damping in the time domain term is replaced as $u_{,tt} = \alpha^2 u_{,xxxx}$ by $u_{,t} = \alpha u_{,xx}$. As a result, Eq. (4.53) is modified as

$$u_{,t} - \alpha u_{,xx} = \left(\frac{\alpha h^2}{12} \right) u_{,xxxx} - \left(\frac{k \alpha^2}{2} \right) u_{,xxxx} - \left(\frac{k^2}{2} \right) u_{,ttt} \quad (4.54)$$

In order to make the dispersion or oscillation term $u_{,xxxx} \leq 0$

$$\left[\left(\frac{\alpha h^2}{12} \right) - \left(\frac{k \alpha^2}{2} \right) \right] \leq 0 \Rightarrow k \geq \frac{h^2}{6\alpha} \quad (4.55)$$

Eq. (4.55) is the stability constraint on k from the consistency analysis. The final constraint on the temporal spacing k is obtained by combining Eqns. (4.52) and (4.55) as $(h^2 / 6\alpha) \leq k \leq (h^2 / 2\alpha)$.

The space domain in the RDQ method has either the uniform or cosine distribution of virtual nodes. Out of them, the constraints for the uniform distribution of virtual nodes are given by Eqns. (4.52) and (4.55). As such, it is essential to study the consistency analysis of the transient heat conduction equation by the cosine distribution of virtual nodes as well, which is done in the next subsection.

4.3.2.3 Consistency analysis of forward time and central space scheme by the cosine distribution of virtual nodes in the space

The RDQ method uses the uniform or cosine distribution of virtual nodes to discretize the governing equation, therefore it is crucial to understand the stability behaviour of the RDQ method by both the distributions. The spatial nodal spacing h is not constant for the cosine virtual nodes but changes at each node as

$$x_i = x_0 + \frac{L}{2} \left[1 - \cos \left(\frac{i-1}{NP-1} \pi \right) \right] \quad \text{for } i = 1, 2, \dots, NP \quad (4.56)$$

where x_0 and x_i are the coordinates of starting and the i^{th} virtual node, and L and NP are the length of computational domain and the total virtual nodes in the domain, respectively. The heat conduction equation in the discretized form by the cosine virtual nodes is given as

$$u_{,t} - \alpha u_{,xx} = \frac{\alpha}{3} (h_2 - h_1) u_{,xxx} + \alpha \left[\left(\frac{h_1^2 - h_1 h_2 + h_2^2}{12} \right) - \left(\frac{k \alpha}{2} \right) \right] u_{,xxxx} \quad (4.57)$$

where the spatial node spacings h_1 and h_2 are given as $h_1 = x_i - x_{i-1}$ and $h_2 = x_{i+1} - x_i$, respectively. In order to keep the dispersion or oscillation term $u_{,xxxx} \leq 0$

$$k \geq \left[\frac{h_1^2 - h_1 h_2 + h_2^2}{6 \alpha} \right] \quad (4.58)$$

The nodal spacings h_1 and h_2 are different at each virtual node due to the cosine distribution. As a result, the temporal spacing k is computed at each virtual node by Eq. (4.58), and the lowest value among them is chosen to get the stable solution. Therefore, Eq. (4.58) is the stability criterion when the transient heat conduction equation is solved by discretizing the spatial domain with the cosine distribution of virtual nodes.

The transient heat conduction equation is numerically solved in the next subsection via the RDQ method with the spatial domain discretized by either the uniform or random field nodes coupled with either the uniform or cosine virtual nodes, and the time domain discretized by the uniform field nodes coupled with the uniform virtual nodes.

4.3.2.4 Numerical solution of transient heat conduction equation by the RDQ method

In this subsection, the RDQ method is implemented to solve the transient heat conduction equation. The numerical results are compared with the analytical and FEM solutions, and they are shown to be closely matching.

The transient heat conduction equation is discretized at the virtual nodes located in the internal computational domain, with the space domain discretized by the uniform virtual nodes combined with the uniform or random field nodes as

$$u_i^{n+1} = m(u_{i-1}^n + u_{i+1}^n) + (1 - 2m)u_i^n, \text{ where } i = 1, 2, \dots, NP \quad (4.59)$$

$$\frac{\partial u}{\partial t}(1, t) = 0 \Rightarrow u_{NP}^n = \left(\frac{2}{3}\right) \left(2u_{NP-1}^n - \frac{u_{NP-2}^n}{2}\right) \quad (4.60)$$

where NP is the number of total virtual nodes in the domain, and n indicates the n^{th} time step. Eq. (4.60) is obtained by the locally applied DQ method, and is used to impose the Neumann boundary condition for the virtual node located at $x=1$. Note that both sides of Eq. (4.60) are at the n^{th} time level.

The governing differential equation, and the initial and boundary conditions are given as

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \text{ for } 0 \leq x \leq 1, \text{ where } \alpha = 1 \quad (4.61)$$

$$u(x, 0) = 1.0, \text{ and } u(0, t) = 0.0 \text{ and } \frac{\partial u}{\partial t}(1, t) = 0.0 \quad (4.62)$$

At the time $t=0$, the values of temperature at all the field and virtual nodes are computed by Eq. (4.62), and the values of boundary condition on the left and right sides of the domain are computed by Eqns. (4.62) and (4.60), respectively. Let $total_it = T/k$ be the total increments of time, where T and k be the total simulation time and the single time increment, respectively. Eq. (4.59) is applied for the increment number $n=1$ to $total_it$ to compute the values of temperature at all the virtual nodes, the unknown values of the nodal parameters at all the field nodes are computed by $[S]_{NP \times N} \{U\}_{N \times 1}^{n+1} = \{u\}_{NP \times 1}^{n+1} \Rightarrow \{U\}_{N \times 1}^{n+1}$, where $[S]_{NP \times N}$ is a shape function matrix approximating the values of function $\{u\}_{NP \times 1}^{n+1}$ at all the virtual nodes by their surrounding field nodes as per the RDQ method, $\{U\}_{N \times 1}^{n+1}$ is the unknown vector of nodal parameter values at all the field nodes at the $(n+1)^{\text{th}}$ time level, and NP and N are the numbers of total virtual and field nodes in the computational domain, respectively. The values of temperature at all the field nodes are then computed by $\{u_f\}_{N \times 1}^{n+1} = [S']_{N \times N} \{U\}_{N \times 1}^{n+1}$, where $[S']_{N \times N}$ is the matrix of shape function interpolating the values of function $\{u_f\}_{N \times 1}^{n+1}$ at all the field nodes by their surrounding field nodes, as per the RDQ method. Eq. (4.55) and

Eq. (4.58) are used to compute the temporal spacing k for the uniform and cosine distributions of the virtual nodes by the RDQ method, respectively.

As the available FEM solution is computed with four 1-D quadratic elements (total 9 field nodes) [104], for a comparison purpose, the results obtained by the RDQ method are also computed by the 9 field nodes with either the uniform or random distribution. The transient distribution of temperature for the field node located at $x=1$ is computed by separately discretizing the spatial domain with the uniform and random field nodes coupled with the uniform and cosine virtual nodes, and compared with the analytical and FEM solutions, as shown in Fig. 4.11b. It is observed from Fig. 4.11b that the solutions obtained by the RDQ method are closely matching with the analytical and FEM solutions, also accurate results are obtained by the randomly scattered field nodes in the domain. This problem is solved by separately considering the 31 uniform and 21 random field nodes scattered in the domain, the diffusion from the time $t = 0$ to 1 sec. is shown in Fig. 4.12 and Fig. 4.13. It is seen from Fig. 4.12b and Fig. 4.13b that the diffusion is well captured by the random field nodes.

Fig. 4.11. The $|\phi|^2$ versus (θ/π) plot by $0 \leq m \leq 0.5$ with the forward time and central space scheme (a), and the time history plot of temperature at the field node $(1, t)$ and its comparison with analytical and FEM solutions (b) for the transient heat conduction.

In summary, two different schemes of the domain discretization are studied in this section for the stability of transient heat conduction equation. Out of them, the forward time and central space scheme is found to be stable. This stable scheme is further analyzed for the consistent discretization of governing equation with the spatial domain

having either the uniform or cosine virtual nodes, and the stability conditions are derived in Eqns. (4.55) and (4.58), respectively.

Fig. 4.12. The diffusion of temperature from the time $t = 0$ to 1 sec. by uniform field and uniform virtual nodes (a), and random field and uniform virtual nodes (b).

The problem is then solved with the uniform and random field nodes scattered in the domain, and it is demonstrated that the results obtained by the RDQ method are closely matching with the analytical and FEM solutions, as seen in Fig. 4.11b. The equation of transverse beam deflection is solved in the next section by the RDQ method with the explicit and implicit approaches.

Fig. 4.13. The diffusion of temperature from the time $t = 0$ to 1 sec. by uniform field and cosine virtual nodes (a), and random field and cosine virtual nodes (b).

4.4 Stability analysis of the transverse beam deflection equation

The transverse motion of the beam is studied in this section for the fixed-fixed configuration by Euler-Bernoulli beam equation $\rho A(\partial^2 v / \partial t^2) + EI(\partial^4 v / \partial x^4) = 0$, where ρ , A , E and I are the density of material, cross sectional area, modulus of elasticity, and moment of inertial of the beam, respectively. This is accomplished by performing the stability and consistency analyses of the transverse beam deflection equation. The governing equation is then numerically solved for the free vibration case by the RDQ method with the temporal spacing k obtained from the stability and consistency analyses, and it is shown that the numerical results are closely matching with the analytically predicted values. The discretization of domains is performed by both the explicit and implicit approaches, and it is noted that the explicit approach adds the numerical dispersion or oscillations. As a result, the ratio of the values of successive beam deflection is more than 1, which leads to the instability. An implicit approach adds numerical dissipation or damping. As a result, the ratio of the values of successive beam deflection is less than 1, which leads to the stable solution.

4.4.1 Explicit approach to solve the transverse beam deflection equation by the RDQ method

The terms of derivatives from the governing differential equation are discretized in this section by the forward time and central space approach, as shown in Fig. 4.14. The DQ weighting coefficients for the m^{th} -order derivative of function are given as

$$(w_{i,j})^m = m \left[a_{i,j} (w_{i,i})^{m-1} - \frac{(w_{i,j})^{m-1}}{x_i - x_j} \right], \text{ and } (w_{i,i})^m = - \sum_{k=1, k \neq i}^{N_p} (w_{i,k})^m \quad (4.63)$$

where m is the order of the derivative, x_i and x_j are the nodal coordinates at the i^{th} and j^{th} virtual node, respectively, N_p is the total virtual nodes in the local discretization domain of the virtual node (x_i, t_j) , and $a_{i,j}$ are the DQ weighting coefficients for the 1st-order derivative. The discretized beam equation at the virtual node (x_i, t_j) by the RDQ method is given as

$$\left\{ \begin{matrix} b_{j,j} & b_{j,j+1} & b_{j,j+2} \end{matrix} \right\} \left\{ \begin{matrix} v_i^j \\ v_i^{j+1} \\ v_i^{j+2} \end{matrix} \right\} + \frac{EI}{\rho A} \left\{ (w_{i,i-2})^4 \quad (w_{i,i-1})^4 \quad (w_{i,i})^4 \quad (w_{i,i+1})^4 \quad (w_{i,i+2})^4 \right\} \times$$

$$\left\{ \begin{matrix} v_{i-2}^j \\ v_{i-1}^j \\ v_i^j \\ v_{i+1}^j \\ v_{i+2}^j \end{matrix} \right\} = 0 \quad (4.64)$$

The values of the 4th-order DQ weighting coefficients in Eq. (4.64) are computed by recursively applying Eq. (4.63) with $m = 4$. The final equation of transverse beam deflection in the discretized form is given as

$$\left\{ \begin{matrix} 1 & -2 & 1 \end{matrix} \right\} \left\{ \begin{matrix} v_i^j \\ v_i^{j+1} \\ v_i^{j+2} \end{matrix} \right\} + \left(\frac{EI k^2}{\rho A h^4} \right) \left\{ \begin{matrix} 1 & -4 & 6 & -4 & 1 \end{matrix} \right\} \left\{ \begin{matrix} v_{i-2}^j \\ v_{i-1}^j \\ v_i^j \\ v_{i+1}^j \\ v_{i+2}^j \end{matrix} \right\} = 0 \quad (4.65)$$

where k and h are the temporal and spatial spacings, respectively. Eq. (4.65) is used for the stability and consistency analyses, and it is found to be similar as the FDM equation for the beam deflection equation. But, the important difference between the FDM and RDQ techniques is that the domain is discretized by either the uniform or random nodes in the RDQ method, while it is discretized by uniform nodes in the FDM.

Fig. 4.14. Discretization stencils of the time and space domains for the transverse beam deflection problem by the explicit approach.

4.4.1.1 Stability analysis of the discretization of transverse beam deflection equation by the explicit scheme

The stability analysis of explicit approach is performed in this section by Eq. (4.65) as

$$v_i^{j+2} = 2 v_i^{j+1} - (1 + 6 M) v_i^j + M (4v_{i-1}^j - v_{i-2}^j + 4v_{i+1}^j - v_{i+2}^j) \quad (4.66)$$

where $M = (E I k^2) / (\rho A h^4)$. The characteristic polynomial, obtained by the inverse Fourier transform of Eq. (4.66), is given as (Please refer Appendix A)

$$\phi(z) = z^2 - 2 z - \{4 M \cos(\theta) [2 - \cos(\theta)] + 1 + 4 M\} \quad (4.67)$$

The roots, $\phi_+ = 1 + 2 \sqrt{M \cos(\theta) \alpha - M}$ and $\phi_- = 1 - 2 \sqrt{M \cos(\theta) \alpha - M}$ where $\alpha = [2 - \cos(\theta)]$, of Eq. (4.67) are plotted by $0 \leq M \leq 1 \times 10^{-5}$ as shown in Fig. 4.15, and it is seen that the maximum values of the roots are close to 1 for $M = 1 \times 10^{-5}$. As a result, the temporal spacing is given as

$$k \leq \sqrt{\frac{(1 \times 10^{-5}) \rho A h^4}{(E I)}} \quad (4.68)$$

Eq. (4.68) is the stability condition for the deflection of beam with the explicit approach.

Fig. 4.15. Plots of roots $|\phi_+|^2$ versus (θ/π) (a) and $|\phi_-|^2$ versus (θ/π) (b) for the problem of transverse beam deflection by the explicit approach.

4.4.1.2 Consistency analysis for the explicit approach of the discretization of transverse beam deflection equation

The terms v from Eq. (4.65) are replaced by the corresponding Taylor series expansion [80], and the discretized equation is given as (please refer Appendix C)

$$m v_{,tt} + EI v_{,xxxx} = -(m k) v_{,ttt} - (EI h^2) v_{,xxxxx} \quad (4.69)$$

where $m = \rho A$. It is noted from Eq. (4.69) that the discretization of the time and space domains is 1st- and 2nd-order accurate, respectively. The terms $v_{,ttt}$ and $v_{,xxxxx}$ represent the dissipation and dispersion with respect to the time and space terms $v_{,tt}$ and $v_{,xxxx}$, respectively, and they have negative coefficients, which indicate that there is a dissipation removal from the system with respect to time and the dispersion removal with respect to space. The term h^2 is small and can be neglected. As a result, the final discretized equation is given as

$$m v_{,tt} + EI v_{,xxxx} = -(m k) v_{,ttt} \quad (4.70)$$

It is seen from Eq. (4.70) that the original governing equation is consistently discretized as $k \rightarrow 0$, $m v_{,tt} + EI v_{,xxxx} \rightarrow 0$, but there is going to be numerical dispersion due to -ve term $v_{,ttt}$. It is essential to know the analytical value of dispersion as it can be compared with the numerical results obtained by the RDQ method, therefore the expressions of the natural frequency (ω_n) and ratio of successive amplitude reduction are derived in the next section.

4.4.1.3 Computation of natural frequency ω_n and the ratio of successive amplitude reduction

In this section, the natural frequency of the fixed-fixed beam configuration is derived by the method of separation of variables, and subsequently the ratio of reduction of amplitude is computed [105].

The terms $v(x, t)$ in Eq. (4.70) can be substituted as $v(x, t) = \psi(t) \xi(x)$ to obtain two ordinary differential equations (ODE) as

$$\frac{1}{\psi} \left[k \frac{\partial^3 \psi}{\partial t^3} + \frac{\partial^2 \psi}{\partial t^2} \right] = -\omega_n^2 \quad (4.71)$$

$$\frac{\partial^4 \phi}{\partial x^4} - \frac{m \phi}{EI} \omega_n^2 = 0 \quad (4.72)$$

Eqns. (4.71) and (4.72) give the variations in the amplitude of the deflection with respect to the time and space domains, respectively. Eq. (4.72) is solved for ω_n by assuming the general solution as a linear combination of the trigonometric equations and $k_n^4 = (m \omega_n^2) / (EI)$

$$\begin{aligned} \phi(x) = & C_1 [\cos(k_n x) + \cosh(k_n x)] + C_2 [\cos(k_n x) - \cosh(k_n x)] + \\ & C_3 [\sin(k_n x) + \sinh(k_n x)] + C_4 [\sin(k_n x) - \sinh(k_n x)] \end{aligned} \quad (4.73)$$

The constants from Eq. (4.73) are solved by the boundary conditions $v(0) = 0.0$, $v(1.0) = 0.0$, $\partial v / \partial x|_{(0)} = 0.0$, and $\partial v / \partial x|_{(1)} = 0.0$ for the fixed-fixed beam configuration to obtain

$$C_1 = 0, C_3 = 0, C_4 = -C_2 \left[\frac{\cos(M) - \cos(hM)}{\sin(M) - \sin(hM)} \right] \text{ and } \cos(M) \cos(hM) = 1.0 \quad (4.74)$$

where $M = k_n L$. Eq. (4.74) is solved either numerically or by the trial and error to get the values of M as 4.73004074486 and 7.85320462 (here it is solved in MATLAB). The first mode is selected as

$$M = k_n L = 4.73004074486 \Rightarrow \omega_n = 22.3733 \text{ rad/sec. for } L = 1 \quad (4.75)$$

Eq. (4.75) gives the natural frequency of the fixed-fixed beam. The value of frequency ω_n from Eq. (4.75) can be easily verified by considering few uniformly distributed virtual nodes and computing the eigenvalues of their spatial discretization matrix by Eq. (4.65). Eq. (4.71) is used to compute the ratios of the reduction in the successive amplitudes of the beam deflection with respect to time. The roots of Eq. (4.71) are computed and the general solution is given as (Please refer Appendix D)

$$\psi(t) = e^{-v_1 t} [c_2 \sin(\omega_d t)] \quad (4.76)$$

where $\omega_d = \left[\sqrt{3} / (24k) \right] \left[2\beta - (8/\beta) \right]$ is the damped natural frequency, $v_1 = (1/12k) (\beta + (4/\beta) - 4)$, where $\beta = \sqrt[3]{-\alpha - 8 + 12\sqrt{3} \left(\sqrt{27\omega_n^2 k^2 + 4} \right) \omega_n k}$ and $\alpha = 108 \omega_n^2 k^2$. The ratio $\psi[t_1 + (2\pi/\omega_d)]/\psi(t_1)$ for the $t = t_1$ and $[t_1 + (2\pi/\omega_d)]$, namely for the two successive peaks in time, is taken by Eq. (4.76) and simplified by the periodicity property of sin function and given as

$$ratio = \exp\left(\frac{-2\pi v_1}{\omega_d}\right) \quad (4.77)$$

Eq. (4.77) is used to compute the ratios of successive amplitude reduction in the beam deflection by the explicit approach.

The transverse beam deflection problem is solved by the explicit approach with the governing equation, boundary and initial conditions as [104]

$$\frac{\partial^2 v}{\partial t^2} + \frac{\partial^4 v}{\partial x^4} = 0 \quad \text{for } (0 < x < 1) \quad (4.78)$$

$$v(0, t) = 0.0, \quad v(1, t) = 0.0, \quad \frac{\partial v}{\partial x}(0, t) = 0.0, \quad \text{and} \quad \frac{\partial v}{\partial x}(1, t) = 0.0 \quad (4.79)$$

$$v(x, 0) = \sin(\pi x) - \pi x (1 - x), \quad \text{and} \quad \frac{\partial v}{\partial x}(x, 0) = 0 \quad (4.80)$$

It is seen from Eq. (4.79) that there are two boundary conditions at the virtual nodes located on the left and right sides of the domain, therefore one of the boundary conditions are transferred to their neighbouring nodes in order to get a single algebraic equation at each virtual node located on the left and right sides of the domain boundaries, as given [106]

$$\frac{\partial v}{\partial x}(0, t) = 0 = \frac{v_2 - v_1}{x_2 - x_1} \Rightarrow v_2(x_2, t) = 0.0, \quad \frac{\partial v}{\partial x}(1, t) = 0 = \frac{v_{NP-1} - v_{NP}}{x_{NP-1} - x_{NP}} \Rightarrow v_{NP-1}(x_{NP-1}, t) = 0.0 \quad (4.81)$$

where NP is the number of total virtual nodes, and the suffices 1, 2, $(NP-1)$, and NP indicate the ids of the virtual nodes. The modified boundary conditions are given as

$$v(0, t) = 0.0, v(1, t) = 0.0, v(x_2, t) = 0.0 \text{ and } v(x_{NP-1}, t) = 0.0 \quad (4.82)$$

This problem is solved by the FEM [104] with 2 Euler-Bernoulli beam elements, therefore here also it is solved by 5 ($h = 0.25$) uniformly distributed field and virtual nodes for a comparison purpose, and the temporal spacing $k = 2 \times 10^{-4}$ sec. is computed by Eq. (4.68). The problem is solved initially by the locally applied DQ and RDQ methods with the spatial domain discretized by the uniform and random field nodes. The deflection at the central field node $(0.5, t)$ is shown in Fig. 4.16a. It is seen that the values of the deflection are identical, which indicates that the RDQ method can effectively handle the distribution of the random field nodes. The problem is further solved by the RDQ method with the spatial domain discretized by either the uniform or random field nodes coupled with the uniform virtual nodes, and their results are compared with the analytical and FEM solutions, as shown in Fig. 4.16b. It is noted from Fig. 4.16b that there is a slight effect of dispersion over time, as predicted from Eq. (4.70). The successive peak values of the amplitude of beam deflection at the time $t_1 = 0.1624$ sec. and $t_2 = 0.3248$ sec. are obtained from Fig. 4.16b, and their ratio is found to be $v_2 / v_1 = (0.2253) / (0.2199) \approx 1.0245$. The analytical ratio of amplitude reduction computed by Eq. (4.77) is found to be $ratio = \exp\{[-2\pi(-0.050058)] / 22.3723\} \approx 1.014$.

Fig. 4.16. Time history plot at the field node $(0.5, t)$ by the locally applied DQ and RDQ methods (a), and the comparison of RDQ method with the FEM and analytical solutions (b) for the problem of transverse beam deflection.

It is observed that the numerical ratio of amplitude reduction is closely matching with the analytical value. It is also observed from the ratio that $\nu_2 / \nu_1 > 1$, which indicates the dispersion in the solution. When this problem is solved with the lower temporal increment $k = 6.25 \times 10^{-5}$ sec., the numerical and analytical ratios of the amplitude reduction are found to be $\nu_2 / \nu_1 \approx 1.007$ and 1.005, respectively, which are also closely matching. As a result, it is concluded that $(\nu_2 / \nu_1) > 1$ and $(\nu_2 / \nu_1) \rightarrow 1.0$ as $k \rightarrow 0$ for the explicit approach.

4.4.2 Implicit approach to solve the transverse beam deflection equation by the RDQ method

As seen from Eq. (4.68) that the explicit approach has stringent requirement on the temporal spacing, therefore the computational cost is very high for the large number of field nodes. Also, Eq. (4.70) shows that there is a reduction in the dissipation with the time, therefore the numerical discretization of the beam deflection equation is consistent only when $k \rightarrow 0$. An implicit approach of time discretization is studied in this section to overcome these limitations.

The derivative terms from the governing differential equation are discretized at the same time level in the implicit approach, as shown in Fig. 4.17a. The discretization of the beam deflection equation by the RDQ method with the scheme shown in Fig. 4.17a is given as

$$\left\{ b_{j,j-2} \quad b_{j,j-1} \quad b_{j,j} \right\} \begin{Bmatrix} v_i^{j-2} \\ v_i^{j-1} \\ v_i^j \end{Bmatrix} + \frac{EI}{\rho A} \left\{ (w_{i,i-2})^4 \quad (w_{i,i-1})^4 \quad (w_{i,i})^4 \quad (w_{i,i+1})^4 \quad (w_{i,i+2})^4 \right\} \begin{Bmatrix} v_{i-2}^j \\ v_{i-1}^j \\ v_i^j \\ v_{i+1}^j \\ v_{i+2}^j \end{Bmatrix} = 0 \quad (4.83)$$

The weighting coefficients from Eq. (4.83) are computed by Shu's general approach [37] and substituted to get

$$\begin{Bmatrix} 1 & -2 & 1 \end{Bmatrix} \begin{Bmatrix} v_i^{j-2} \\ v_i^{j-1} \\ v_i^j \end{Bmatrix} + \left(\frac{EI k^2}{\rho A h^4} \right) \begin{Bmatrix} 1 & -4 & 6 & -4 & 1 \end{Bmatrix} \begin{Bmatrix} v_{i-2}^j \\ v_{i-1}^j \\ v_i^j \\ v_{i+1}^j \\ v_{i+2}^j \end{Bmatrix} = 0 \quad (4.84)$$

Rearranging the terms from Eq. (4.84) leads to

$$v_i^j (1 + 6M) + M (v_{i-2}^j - 4v_{i-1}^j - 4v_{i+1}^j + v_{i+2}^j) = 2v_i^{j-1} - v_i^{j-2} \quad (4.85)$$

where $M = (EI k^2) / (\rho A h^4)$. The application of Eq. (4.85) at all the virtual nodes located in the internal computational domain results in the matrix, which can be solved by any of the direct or iterative solvers.

4.4.2.1 Consistency analysis of the transverse beam deflection equation discretized by the implicit approach

It is essential to know the numerical form of the transverse beam deflection equation that is exactly solved after discretizing it with the implicit approach. Therefore, the consistent equation after expanding the terms v from Eq. (4.85) by Taylor series [80] is given as (Please refer Appendix C)

$$m v_{,tt} + EI v_{,xxxx} = m k v_{,ttt} \quad (4.86)$$

where $m = \rho A$. It is seen from Eq. (4.86) that the numerical damping with respect to the time domain is added, but still the discretization is consistent because $m v_{,tt} + EI v_{,xxxx} \rightarrow 0$ as $k \rightarrow 0$. As a result, the discretized governing PDE in Eq. (4.86) approaches the continuous form given in Eq. (4.78) with the reduction in the temporal spacing. In order to analytically compute the numerical dissipation in Eq. (4.86), the equation for the ratio of successive amplitude reduction, as also mentioned in [105], is developed in the next subsection.

4.4.2.2 Computation of the amplitude reduction ratio for the implicit approach

The terms $v(x, t)$ from Eq. (4.86) are substituted by $v(x, t) = \psi(t) \xi(x)$, as explained earlier during the method of separation of variables, and it is converted to two ODE as

$$k \frac{\partial^3 \psi}{\partial t^3} - \frac{\partial^2 \psi}{\partial t^2} - \omega_n^2 \psi = 0 \quad (4.87)$$

$$\frac{\partial^4 \phi}{\partial x^4} - \frac{m\phi}{EI} \omega_n^2 = 0 \quad (4.88)$$

Eq. (4.87) is solved to compute the ratios of the successive amplitude reduction with respect to time. The root of Eq. (4.87) are evaluated and the general solution is given as

$$\psi(t) = e^{v_1 t} [c_2 \sin(\omega_d t)] \quad (4.89)$$

where $\omega_d = \left[\sqrt{3} / (24k) \right] [2\beta - (8/\beta)]$ is the damped natural frequency, and $v_1 = (1/12k) [-\beta - (4/\beta) + 4]$, where $\beta = \sqrt[3]{\alpha + 8 + 12\sqrt{3} \left(\sqrt{27\omega_n^2 k^2 + 4} \right) \omega_n k}$ and $\alpha = 108 \omega_n^2 k^2$. The ratio $\psi[t_1 + (2\pi/\omega_d)] / \psi(t_1)$ for the $t = t_1$ and $[t_1 + (2\pi/\omega_d)]$, namely for two successive peaks in time, is taken by Eq. (4.89) and simplified using the periodicity property of sin function as

$$ratio = \exp\left(\frac{2\pi v_1}{\omega_d}\right) \quad (4.90)$$

Eq. (4.90) is used to compute the ratios of the successive amplitude reduction by the implicit approach.

The problem of transverse beam deflection is solved by the implicit approach with the governing differential equation, initial and boundary conditions, as given in Eqns. (4.78), (4.80) and (4.82), respectively. The problem is solved by discretizing the domain with the number of total 5 uniform or random field nodes coupled with the number of total 5 uniform virtual nodes and $k = 2 \times 10^{-4}$ sec., to compare the results with those obtained by the explicit approach. Fig. 4.17b shows the comparison of results obtained by the RDQ method via the implicit approach with the FEM and analytical solutions, and it is seen that the deflection values at the field node $(0.5, t)$ are damping with time, as expected

from Eq. (4.86), and identical results are obtained by the uniform and random distributions of the field nodes. The successive peak values are obtained from Fig. 4.17b and their ratio is evaluated as $v_2 / v_1 = (0.2043) / (0.2094) \approx 0.9756$, and the analytical ratio of amplitude reduction computed by Eq. (4.90) is evaluated as $ratio = \exp[2\pi(v_1) / 22.3723] \approx 0.9860$. Therefore, the numerical and analytical values of the amplitude reduction ratios are closely matching. It is also observed that $v_2 / v_1 < 1$, which clearly indicates the dissipation in the solution. This problem is again solved with the lower temporal increment $k = 6.25 \times 10^{-5}$ sec., similar to explicit approach. The new numerical and analytical values of the successive amplitude reduction ratio are $(v_2 / v_1) \approx 0.992$ and 0.9956 , respectively, which are also closely matching. Therefore, it is concluded that $(v_2 / v_1) < 1$ and $(v_2 / v_1) \rightarrow 1.0$ as $k \rightarrow 0$ for the implicit approach.

In summary, the stability analysis of the transverse beam deflection equation is performed in this section by the explicit and implicit approaches. It is observed that the temporal spacing has to be very small with increase in the field nodes to get the stable solution by the explicit approach. This results in high computational cost, and the numerical discretization of the governing PDE involves the negative dissipation, as shown by Eq. (4.70). The implicit approach is used for the temporal discretization to overcome these problems, and the numerical discretization of governing PDE is found to be consistent with the continuous form with some numerical damping, as shown by Eq. (4.86), also there is no direct constraint on the temporal spacing from the point of view of stability. It is also shown that the numerical ratios of successive amplitude reduction are matching with their corresponding analytical values for both the explicit and implicit approaches.

4.5 Summary

As first, the stability analysis of the 1st-order wave equation is performed by the VN and Schur polynomials with the five different single and multi-step schemes. The stable schemes are identified, and further analyzed for additional constraints on r based on the consistency analysis. The observations made from the stability and consistency analyses of the stable schemes are verified by numerically implementing them to solve the 1st-

order wave equation by the methods of locally applied DQ (for Schemes 1, 4 and 5) and RDQ (for Scheme 5). The results are analyzed by the virtual nodes distributed with the uniform and cosine manners in the space domain. It is observed that some numerical dissipation is involved when the cosine distribution of virtual nodes is used in the space domain. This observation is analytically verified by deriving the constraint conditions for the scheme 5 with the consistency analysis, as given in Eq. (4.47).

Fig. 4.17. Discretization stencils of the time and space domains (a) and the comparison of the numerical solution of the beam deflection at the field node $(0.5, t)$ with the FEM and analytical solutions (b) when the problem of transverse beam deflection is solved by the implicit approach.

Next, the equation of transient heat conduction is solved by the discretization scheme of forward time and central space with the stable range of the temporal spacing k is identified from the stability and consistency analyses. The space domain is discretized with either the uniform or random field nodes coupled with the uniform or cosine virtual nodes, and their results at the field node $(0.5, t)$ are compared with the FEM and analytical solutions, as shown in Fig. 4.11b. The profiles of temperature distribution from the time $t = 0$ to 1 sec. are plotted in Fig. 4.12 and Fig. 4.13, and it is seen that the results are almost identical, which indicate that the RDQ method is capable of equally handling the field nodes distributed either uniformly or randomly in the space.

Finally, the equation of the transverse beam deflection is solved by the RDQ method with the explicit and implicit approaches of the discretization, and it is shown that even though the explicit approach gives the consistent discretization of the governing PDE, the numerical dispersion is involved, which makes it unstable. This is verified by comparing

the numerically computed ratios of successive amplitude reduction with the corresponding analytical values, and noting that $(v_2 / v_1) > 1$ and $(v_2 / v_1) \rightarrow 1.0$ as $k \rightarrow 0$. The implicit approach results in the consistent discretization of the governing PDE with a numerical damping term, but the solution is numerically stable. This is verified by comparing the numerically computed ratios of the successive amplitude reduction with the analytical values, and observing that $(v_2 / v_1) < 1$ and $(v_2 / v_1) \rightarrow 1.0$ as $k \rightarrow 0$.

In summary, a broad conceptual framework, based on the physical interpretation of mathematical terms, is developed in the present work to solve the transient PDE by combining the stability and consistency analyses. The approach can be successfully applied in solving any transient PDE, as demonstrated by several problems solved in this chapter.

Chapter 5

Adaptive analysis of the RDQ method and the development of adaptive random differential quadrature (ARDQ) method with application in the locally high gradient problems

5.1 Introduction

An adaptive numerical method called the adaptive random differential quadrature (ARDQ) method is developed in this chapter. In the ARDQ method, the RDQ method is coupled with *a posteriori* error estimator based on the relative error norm in the displacement field. An error recovery technique based on the least square averaging over the local domain of interpolation is proposed, which improves the accuracy of solution as the spacing $h \rightarrow 0$. A novel convex hull approach with the cross product of vectors is proposed in the adaptive refinement to ensure that the newly created nodes are always within the domain of computation. The numerical accuracy of the ARDQ method is successfully evaluated by solving several 1-D, 2-D problems of irregular domain with the local high gradients in the distribution of field variables. It is concluded from the convergence rates that the ARDQ method coupled with the error recovery technique can be effectively used to solve the locally high gradient problems of the initial and boundary values.

Adaptive generation of mesh in the FEM has been one of the hot research areas for a long time, and a lot of theoretical studies are available. In the classical FEM with the uniform refinement of mesh, the numerical error is monotonically decreased with the element size h , as the elements are said to be quasi-uniform. But, the elements may not be quasi-uniform in the adaptive FEM. As a result, the monotonic decrease of the error may not be guaranteed. Therefore, it becomes essential to have a good error indicator during the adaptive refinement of mesh that ensures the continuous decrease in the error. The same requirement is applied to the meshless methods, wherein the field nodes may be randomly scattered in the domain. As a result, the nodal spacing h is not constant. *A priori* error estimator gives the rough estimate of error reduction by assuming the

asymptotic behaviour of method, and *a posteriori* error estimator evaluates the error based on the residual of discretized governing equation at the field nodes or by comparing the numerical value of any quantity of interest with the corresponding analytical value.

In general, there are two types of *a posteriori* error estimators viz. the explicit and implicit [107]. The explicit error estimators evaluate the error based on the currently available data, therefore the direct solution is available, namely the least square error estimator [107]. The implicit *a posteriori* error estimators involve the solution of linear system of algebraic equations. Some examples of the implicit error estimator are the subdomain residual method and element residual method [107]. The recovery-based error estimators are also developed, which are based on the error evaluation between the numerically computed gradient and the approximation to the gradient, therefore the required order of the approximation of gradient is achieved. Several adaptive FEM methods based on the error-controlled principle are applied to different engineering problems, such as elasto-plasticity, contact problems, and the shell under the transient loading [108]. Babuška [109] explained the feedback controlled FEM with the h , p and $h-p$ versions, and demonstrated by the 1-D example that the $h-p$ version has the highest rate of convergence. Zienkiewicz and Craig [109] pointed out that the use of p refinement is better than h refinement in the case of singularities. They also explained the hierarchical FEM, in which there is no need to discard the earlier computed shape functions after each refinement; this leads to the mixed-order interpolation.

Several *a posteriori* error estimators based on the different concepts are developed for FEM. Bank and Weiser [110] presented three *a posteriori* error estimators for FEM in the energy norm, based on solving the local Neumann problem in each element. Babuška and Rheinboldt [111] derived the error estimators based on the approach, which is similar to the residual method, but uses negative norm of Sobolev spaces. Diening and Kreuzer [112] studied the adaptive FEM (AFEM) for p -Laplacian like PDE using the linear functions, and presented the rates of convergence by applying the AFEM to nonlinear Laplace equations. Zienkiewicz and Zhu [113] proposed an error estimator based on the stress and coupled it with the h -refinement, in which the stress field is interpolated by the same shape functions that are used for the interpolation of displacement. The residual

error between the computed and assumed stress fields is reduced over the domain. The error is evaluated by the newly interpolated and the earlier computed values of stress. Babuška and Miller [114-116] explained the post-processing of FEM solutions in details and an adaptive mesh refinement based on *a posteriori* error estimation. Zienkiewicz and Zhu [117-118] proposed a stress recovery technique with the values of stress at the Gauss points, as generally they are superconvergent. They considered a patch of elements around a concerned vertex node, and the value of stress at the concerned vertex node is interpolated by the values of stress with the LS procedure at the surrounding Gauss points located within the patch of elements. The newly computed value of stress is used to evaluate an error at the concerned node, which in turn is used in the adaptive analysis.

Several *a posteriori* error estimators are developed for the meshless methods [119-126]. Duarte and Oden [119] used a partition of unity based $h-p$ cloud method for adaptive analysis, and derived *a posteriori* error estimator following the procedure used in the FEM. Gavete *et al.* [120] proposed an error indicator for EFGM, in which the FEM shape functions are replaced by the functions of local LS approximation. Han and Meng [121] did the theoretical analysis of error estimation in the RKPM. Liszka *et al.* [122] tested the h and p adaptivity of the hp - meshless method, and discussed in details about the refinement of interpolation domain. Rüter and Stenberg [123] developed *a posteriori* error estimates for the mixed finite elements. As per the formulation, the approximate field of displacement is one order lower than the field of stress, therefore they used an averaging technique to improve the displacement field. Lee and Shuai [124-126] developed the procedure of adaptive refinement by the RKPM functions, and formulated an error estimator based on the superconvergence property of some extraction functions. Several meshless methods are extended to perform an adaptive analysis [127-129] and to solve numerous engineering problems [130-135].

It is observed in the meshless methods that the accuracy of the approximated function at the concerned node is affected by the number of surrounding nodes included in the domain of interpolation. Numerous studies have been performed to develop a method to obtain the optimal number of support nodes [119, 122, 136]. It is observed in the RDQ method that the number of nodes included in the local interpolation affects the accuracy near the locally high gradient region and not much in the remaining region. This

phenomenon is called the numerical pollution. Therefore, it becomes difficult to define the number of interpolation nodes to be used near the locally high gradient regions. Out of this motivation, a new error recovery technique, based on the least square concept, is proposed to improve the accuracy of solution by recovering the numerical error. The proposed error recovery technique not only smoothens the values of approximated function but also improves them. Also, it can be easily incorporated into the existing code of the RDQ method due to the simplicity.

The objective of the present work is to develop an adaptive RDQ method by coupling with *a posteriori* error estimator based on the l_2 error norm. The local refinement is performed at the concerned field node, if the computed relative error in the value of function at the concerned field node is greater than the minimum permitted value. An error recovery technique based on the LS approximation is developed to improve the accuracy of solution. During an adaptive refinement procedure, it is possible to generate a node, which falls outside the computational domain. Therefore, in order to develop a robust adaptive algorithm for the ARDQ method, a convex hull approach with the cross product of vectors is used in a novel manner, which ensures all the newly generated field nodes to be within the computational domain. The proposed approach is demonstrated here by the convex geometry, but it can also be used well for non-convex or irregular geometry, as explained later. The convergence study of the ARDQ method is performed by solving several 1-D and 2-D locally high gradient problems. It is observed from this study that the convergence rates obtained after using the error recovery technique are better than those obtained by directly using the numerically computed values of function, also the values of local high gradient are well captured. A semi-infinite plate with a central hole with an irregular or non-convex geometry along the boundary is solved to demonstrate the applicability of the ARDQ method in solving the problems of irregular geometry. With the different geometries of irregular boundary, only thing to be changed in the ARDQ method is the way to create the field and virtual nodes along the irregular boundary. Then the local parametric equations are developed for the refinement of field nodes during the adaptive iterations. Once this is achieved, the ARDQ method is directly applied as it is independent of the problems or the computational domains.

The motivation behind the development of ARDQ method with an error recovery technique is to solve the locally high gradient problems by the fine and relatively coarse nodal distributions near the peak and remaining domain, respectively. The potential applications of the ARDQ method include the problems, such as the hydrogel simulation, crack propagation, and other problems involving the moving boundaries etc.

There are few other meshless methods proposed to solve the locally high gradient problems. Zhang *et al.* [129] proposed a method called the GSM, in which the derivatives of the 1st- and 2nd-order are approximated by the gradient smoothing technique. The mesh refinement is performed by the GSM via Delaunay diagram, which is numerically intensive when compared with the proposed ARDQ method, as explained later. The GSM uses conventional mesh, and it requires more field nodes as compared with the ARDQ method for an equal reduction in the true error norm. Better convergence rates are observed in the ARDQ method, which can even work well with the initial distribution of random field nodes, while the GSM requires uniform triangular grid [129]. Duarte and Oden [119] and Liszka *et al.* [122] developed the *h-p* adaptive and *hp* meshless methods, respectively, by a partition of unity concept applied over the cloud of field nodes. They [119] also created four new field nodes in the “refine” stage, as done in the ARDQ method. However, these methods [119, 122] are based on the Galerkin approach. As a result, the integration of domain is performed with the sufficiently dense background structure of cells to ensure the precise integration of function values. However, no integration is required in the ARDQ method, as it is based on the strong-form approach. Therefore, the ARDQ method is relatively easy to implement and able to accurately capture the high local gradients. Kee *et al.* [127] developed a regularized least-squares radial point collocation method (RLS-RPCM), which involves the regularization technique to stabilize the RPCM method. The new nodes are added by Delaunay cells in RLS-RPCM. As a result, the RLS-RPCM is computationally intensive as compared with the ARDQ method.

The ARDQ method offers several advantages as compared with the existing meshless methods capable of capturing the locally high gradients. It has simple yet effective algorithm of the adaptive refinement, which ensures the newly created field nodes to be always within the computational domain. It can work with initial distributions

of either the uniform or random field nodes. The ARDQ method is based on the strong-form approach, such that no numerical integration is required. It is easy to implement, and offers flexibility while generating the new field nodes, as either the uniformly or randomly distributed. The ARDQ method coupled with the error recovery technique ensures that the effect of numerical pollution is minimized, and the function is reproduced up to the required order of approximation.

The error recovery technique based on the LS approximation is developed in the next subsection.

5.2 Error recovery technique in the ARDQ method

It is observed in the RDQ method that the numerically computed solution near the locally high gradient region is affected by the number of total field nodes included in the interpolation of field variable. Therefore, it becomes difficult to decide the number of nodes to be included in the fixed RKPM interpolation. Due to this motivation, an error recovery technique based on the LS averaging is formulated and proposed in this section. It is shown after the implementation that the newly proposed technique not only smoothens the solutions but also improves them, such that this technique is coupled with the ARDQ method.

In the ARDQ method, the nodal parameters u_i at the field nodes are evaluated based on the certain order of the monomials in the fixed RKPM interpolation function. Therefore, if the m nodes are used in the interpolation domain of the k^{th} field node

$$f_k^* = \sum_{i=1}^m N_i u_i^* \quad (5.1)$$

where f_k^* , N_i , and u_i^* are the recovered values of the approximate function at the k^{th} field node, and the shape function and nodal parameter at the i^{th} field node, respectively. The model expression of the nodal parameter value at the i^{th} field node is written as

$$u_i^* = \sum_{j=1}^n X_{ij} \beta_j, \text{ where } X_{ij} = \{1 \quad x_i \quad y_i \quad \dots \quad y_n\} \quad (5.2)$$

where $j=1$ to n are the monomial terms in the interpolation of the i^{th} field node, $i=1$ to m are the number of total field nodes used in the local interpolation domain of the k^{th} field node, and X_{ij} and β_j are the vectors of the n^{th} -order monomials and unknown coefficients, respectively. Let us take a 2-D problem as an example,

$$X_{ij} = \{1 \quad x_i \quad y_i \quad x_i^2 \quad x_i y_i \quad y_i^2\}, \quad n = 6 \quad (5.3)$$

The residual error at the i^{th} interpolation node is given as

$$r_i = \sum_{j=1}^n X_{ij} \beta_j - u_i \quad (5.4)$$

where u_i is the numerically computed value of the nodal parameter at the i^{th} field node by the RDQ method. The total residual error over the domain of m local interpolation nodes is given as

$$E = \sum_{i=1}^m (r_i)^2 \quad (5.5)$$

The minimized error with respect to β_j is given as

$$\frac{\partial E}{\partial \beta_j} = 2 \sum_{i=1}^m r_i \frac{\partial r_i}{\partial \beta_j} = 0 \Rightarrow \frac{\partial E}{\partial \beta_j} = 2 \sum_{i=1}^m X_{ij} \left[\sum_{j=1}^n X_{ij} \beta_j - u_i \right] = 0 \quad (5.6)$$

Eq. (5.6) in the matrix form is written as

$$\begin{bmatrix} \sum_{i=1}^m X_{i1} X_{i1} & \sum_{i=1}^m X_{i1} X_{i2} & \dots & \sum_{i=1}^m X_{i1} X_{in} \\ \sum_{i=1}^m X_{i2} X_{i1} & \sum_{i=1}^m X_{i2} X_{i2} & \dots & \sum_{i=1}^m X_{i2} X_{in} \\ \vdots & \dots & \ddots & \vdots \\ \sum_{i=1}^m X_{in} X_{i1} & \sum_{i=1}^m X_{in} X_{i2} & \dots & \sum_{i=1}^m X_{in} X_{in} \end{bmatrix} \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^m X_{i1} u_i \\ \sum_{i=1}^m X_{i2} u_i \\ \vdots \\ \sum_{i=1}^m X_{in} u_i \end{Bmatrix} \quad (5.7)$$

The values β_j computed by Eq. (5.7) are substituted in Eq. (5.2) to determine the recovered values of nodal parameter, which is in turn substituted in Eq. (5.1) to evaluate the recovered values of the function approximation. It is noted from Eq. (5.2) that the

recovered values of nodal parameter u_i^* are only valid for the expression of error recovery, as given by Eq. (5.1), for the k^{th} field node only. It means that if any i^{th} field node is included in the multiple interpolation domains, the recovered values of the nodal parameter u_i^* corresponding to the i^{th} field node should be evaluated separately for the respective interpolation domains.

The technique of error recovery developed in this section is not much affected by the number of field nodes considered in the local interpolation domain of the function at the field nodes, as the computed value u_i^* is multiplied by the shape function N_i in Eq. (5.1), smoothening u_i^* as per the location of the i^{th} node with respect to the concerned node k . The recovered values of the approximate function are considered for the evaluation of error during the refinement stage of the ARDQ method.

5.3 Adaptive RDQ method

The algorithm of the ARDQ method is broadly divided into two steps, the first for the error evaluation, and the second the refinement procedure.

5.3.1 Computation of error in the ARDQ method

In order to compute the convergence rates, the global error is evaluated by l_2 error norm averaged over the number of total field nodes, and normalized by the maximum analytical value of the function as given [81]

$$\varepsilon = \frac{1}{|f^e|_{\max}} \sqrt{\frac{1}{NP} \sum_{I=1}^{NP} [f_I^{(e)} - f_I^{(n)}]^2} \quad (5.8)$$

where ε is a global error in the solution, NP is the number of total field nodes scattered in the domain, and $f_I^{(e)}$ and $f_I^{(n)}$ are the analytical and numerical values of the function at the I^{th} field node, respectively. The true error norm based on l_2 error norm is given

$$\xi = \sqrt{\frac{\sum_{I=1}^{NP} [f_I^{(e)} - f_I^{(n)}]^2}{\sum_{I=1}^{NP} [f_I^{(e)}]^2}} \quad (5.9)$$

In the ARDQ method, as the domain is locally refined where the numerical error is high, the field node spacing h can not be constant throughout the domain. As a result, an average nodal spacing is evaluated to compute the convergence rates as given [47]

$$h_x = \frac{L_x}{\sqrt{NP}-1}, \quad h_y = \frac{L_y}{\sqrt{NP}-1} \quad (5.10)$$

where h_x and h_y , and L_x and L_y are the average nodal spacings and the domain lengths in the x and y directions, respectively, and NP is the number of total field nodes in the domain. The spacing $h = h_x = h_y$ is used for the purpose of convergence plots.

The relative error in the approximated function at each field node and the global error computed by Eq. (5.8) are used as the local refinement and adaptive refinement termination criteria in the ARDQ method, respectively. When the analytical solution $f_I^{(e)}$ is not available, a resultant function gradient at each field node, $(F_1)_i = \sqrt{f_{,x}^2 + f_{,y}^2 + f_{,xy}^2}$, where $i \in [1, NP]$, and $f_{,x}$, $f_{,y}$ and $f_{,xy}$ are the derivatives of function with respect to x and y variables, is used as a criterion of local refinement. The difference between the values of function at the field nodes obtained during two successive adaptive iterations is computed and used as the refinement termination criterion. Let the previous and current iteration counts of adaptive refinement be n and $(n+1)$, respectively, having NP and $(4 NP)$ numbers of total field nodes, respectively, such that $3 NP$ field nodes are newly created in the $(n+1)^{\text{th}}$ iteration. The values of function, at the newly created $3 NP$ field nodes, corresponding to the n^{th} iteration are computed by an interpolation using the existing total NP field nodes at the n^{th} level of iteration. As the solution gets converge, the interpolated values of function at $3 NP$ field nodes corresponding to the n^{th} adaptive iteration step will closely match with their numerically computed values during the $(n+1)^{\text{th}}$ adaptive iteration. The second norm of the difference between the values of

function at all 4 NP field nodes is computed by their values at the n^{th} and $(n+1)^{\text{th}}$ adaptive iterations. This second norm is used as the termination criterion in the adaptive refinement.

5.3.2 Procedure of adaptive refinement in the ARDQ method

The h -adaptive refinement is implemented in the ARDQ method. It is sometimes possible to generate a node during the adaptive refinement that falls outside the computational domain. In order to overcome this problem, a novel approach of convex hull with the cross product of vectors is developed for the ARDQ method. This approach ensures that the newly created field node is always inside the domain of computation. In order to maintain the adaptive algorithm of ARDQ method as simple as possible without losing its effectiveness while capturing the values of locally high gradients, two arrangements are adopted to create new field nodes viz. uniform and random. As explained below, both the arrangements will ensure that the newly created nodes will increase the probability of correctly capturing the locally high gradient values irrespective of their directions. The following algorithm explains the procedure of 2-D adaptive refinement for the ARDQ method.

- (A) The governing PDE is solved initially by the RDQ method coupled with the error recovery technique, and the relative error in the approximated function is computed at each field node by $E_i = [(f_i - f_i^h) / f_i] \times 100$, where f_i and f_i^h are the exact and numerical values of function at the i^{th} field node, respectively. The global and true error norms are computed by Eqns. (5.8) and (5.9), respectively.
- (B) The local refinement around the concerned i^{th} field node is performed if $E_i \geq E_{ip}$, where E_{ip} is the permitted value of the local error. The smallest distance d_m between the i^{th} field node and its surrounding node is computed. If $d_m \geq \text{Min_Per_Dist}$, where Min_Per_Dist is the minimum permitted distance between any two nodes in the domain, four new nodes are created around the concerned i^{th} field node, as shown in Fig. 5.1, based on the location of the i^{th} field node. If the i^{th} node is within the computational domain, the new nodes are created at

$$\left(x + \frac{d_m}{2}, y + \frac{d_m}{2}\right), \left(x + \frac{d_m}{2}, y - \frac{d_m}{2}\right), \left(x - \frac{d_m}{2}, y + \frac{d_m}{2}\right) \text{ and} \\ \left(x - \frac{d_m}{2}, y - \frac{d_m}{2}\right) \quad (5.11)$$

If the i^{th} field node is along the boundary $x = 0$, as shown in Fig. 5.1, the new nodes are created at

$$\left(x, y + \frac{d_m}{2}\right), \left(x + \frac{d_m}{2}, y + \frac{d_m}{2}\right), \left(x + \frac{d_m}{2}, y - \frac{d_m}{2}\right) \text{ and} \\ \left(x, y - \frac{d_m}{2}\right) \quad (5.12)$$

Fig. 5.1. Mesh refinement for the concerned node when it is located inside the domain or along the boundary (a), or at the corner of the computational domain (b) in the ARDQ method.

If there is an existing node at the location of newly created node during the refinement procedure, the next higher d_m is considered. The four nodes are added around the concerned i^{th} field node to ensure that at least one node is created in each of the four quadrants with reference to the concerned i^{th} field node. The motivation behind this approach is to capture the high gradients, $f_{,x}$, $f_{,y}$, and $f_{,xy}$, whether they are in the x or y directions. If the number of total field nodes (newly created plus with already existing) are exceeding the number of total virtual nodes during the adaptive refinement, depending on the difference between the number of total virtual and field

nodes, only one or two new field nodes instead of four are created around only those field nodes having the values of absolute local gradient $(F_1)_i$ more than a certain cut-off value. This precaution is necessary to ensure that the total field nodes are always lower than or equal to the total virtual nodes to avoid the rank deficiency in the final stiffness matrix.

Instead of uniformly distributing the newly generated field nodes around the concerned i^{th} field node, as shown in Fig. 5.1, they can also be distributed randomly with the motivation of capturing the high gradients in whichever directions. This is achieved by initially arranging the entire field nodes in the ascending order by their Euclidean distance from the concerned i^{th} field node. The location of each node with respect to the concerned i^{th} node (x_i, y_i) is checked by this sequence. Such as, the p^{th} node at (x_p, y_p) near the i^{th} field node is falling in its 1st quadrant such that $x_p \geq x_i$ and $y_p \geq y_i$, then compute $\Delta x = |x_p - x_i|$ and $\Delta y = |y_p - y_i|$. Using the i^{th} node coordinates (x_i, y_i) , and Δx and Δy , one can generate a random node with the x and y coordinates as $x = x_i + [\text{rand}() / \text{RND_MAX}] \Delta x$ and $y = y_i + [\text{rand}() / \text{RND_MAX}] \Delta y$, where $\text{rand}()$ and RND_MAX are the function of random value generation and the maximum possible generated random value, respectively. If the newly generated node is not overlapping with any of the existing nodes, it will be included in the refinement, and similarly new nodes are generated in the remaining three quadrants.

(C) The process of adaptive iteration is terminated, if $\varepsilon < E_{gp}$ or $NP > NV$, where E_{gp} is the permitted global error, and NP and NV are the total numbers of field and virtual nodes distributed in the domain, respectively.

The objective of the variable Min_Per_Dist is to avoid any singularities in the moment matrix, while computing the shape functions by the fixed RKPM interpolation function. Therefore, its numerical value depends on the total length of the computational domain, and the number of total field nodes generated by the adaptive algorithm. For all the test problems presented in this chapter, it is assigned a numerical value equal to 1×10^{-6} . Therefore, this algorithm ensures that the moment matrix computed in the fixed

RKPM interpolation function is not singular by setting the parameter Min_Per_Dist , also the adaptive procedure can be easily extended to the problems with 3-D domains.

One of the important tasks in the refinement algorithm of the proposed ARDQ method is that the newly created node has to be checked whether it falls within the computational domain or outside of it. This is done in a novel way by the approach of convex hull with the cross product of vectors, where the domain of computation is treated as a convex hull. The convex hull is an outermost envelope covering all the given points. As a result, if any two points within the convex hull are joined together to form a straight line, all points on the newly created line also fall within the convex hull. For a given computational domain in the ARDQ method, a convex hull is constructed starting with a node having the smallest x and y coordinates and continuing with the outermost nodes of domain. For example in the 2-D rectangular domain, the four corner nodes will form a convex hull covering the whole computational domain. Once the convex hull is formed, the newly created node is tested whether it falls within the generated convex hull or outside of it by taking the cross product of vectors formed by each side of the convex hull with the newly created node. For example, let us consider the nodes $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$ form a convex hull with the edges given as P_0 to P_1 , P_1 to P_2 , P_2 to P_3 , and P_3 to P_0 . If a new node $P_n(x_n, y_n)$ is generated, the cross product of vectors, $\vec{P}_{01} = P_1 - P_0$ and $\vec{P}_{0n} = P_n - P_0$, is computed. If $\vec{P}_{01} \times \vec{P}_{0n} \geq 0$, and similarly if $\vec{P}_{12} \times \vec{P}_{1n} \geq 0$, $\vec{P}_{23} \times \vec{P}_{2n} \geq 0$, and $\vec{P}_{30} \times \vec{P}_{3n} \geq 0$, it is concluded that the newly generated node P_n falls within the convex hull formed by the field nodes P_0 , P_1 , P_2 and P_3 . If the P_n does not fall in the convex hull, its coordinates has to be adjusted with reference to the specific edge of the convex hull for which the cross product was less than zero. It is ensured this way in the ARDQ method that all the newly created field nodes are within the computational domain. The cross product approach is used in Graham scan algorithm to generate the convex hull, but it is used very differently and innovatively in the ARDQ method. The convex hull approach explained above is shown in Fig. 5.2.

Fig. 5.2. Working principle of the convex hull approach, when the newly added field node P_n is inside (a) and outside (b) of the computational domain.

Although the algorithm is explained here by the computational domain with a convex or uniform geometry of boundary, it can be very well applied for an irregular or non-convex geometry, as demonstrated in the test problem of semi-infinite plate with a central hole. As long as the computational domain has a well defined boundary, which is normally required to have a well-posed problem, the proposed algorithm will work. In case of an irregular or non-convex geometric boundary, a local parametric equation $f[x(t), y(t), z(t)]$ where t is a parameter, is developed to model the geometry by the cubic spline or non-uniform rational B-spline (NURBS) curves with the nodes along the irregular boundary as the control points. While checking a newly created node $P_n(x_n, y_n)$ as per the above explained algorithm, the irregular boundary is traced by increasing the parameter t and obtaining the successive nodes on the curve, which can be used to compute the cross product of vectors. A similar approach is adopted while solving the problem of semi-infinite plate with a central hole.

In summary, the adaptive refinement procedure for the ARDQ method with the error recovery technique is discussed in this section. It is shown that the refinement algorithm is fairly flexible and can be modified at any level based on the requirement, and readily extended to the problems with the 3-D domains. Several 1-D and 2-D test

problems are solved in the next subsection by the ARDQ method, and it is shown that the ARDQ method converges with a reasonable rate of convergence.

5.4 Convergence analysis of ARDQ method

The ARDQ method is applied to solve several locally high gradient problems in this section and their convergence analysis is performed. The values of $E_{ip} = 0.01$ and $E_{gp} = 1 \times 10^{-6}$ are set for all the test problems. The objective of this section is to test how accurately the ARDQ method converge the numerical solution to the corresponding analytical values with each iteration of adaptive refinement.

The analytical solutions for all the present test problems contain at least the 3rd- or higher order monomial terms. Therefore, it is essential to include maximum up to the 2nd-order monomial terms in the approximation of function, while performing the convergence studies. When the convergence rates obtained by the ARDQ method are compared with those by the RDQ method, it is essential for both the methods to maintain the same order of the continuity of function approximation, and the same distribution pattern of nodes either uniform or random, at the beginning of computation. Therefore, all the test problems are solved here by including up to the complete 2nd-order monomial terms in the polynomial basis of the function approximation. It means the approximated function is expected to have the 2nd-order continuity. The model expression of the nodal parameter value, as given in Eq. (5.2), enforces the required order of continuity on the nodal parameter value, and the residual error is computed correspondingly in Eq. (5.4). In order to achieve a consistency between the ARDQ method and error recovery technique while approximating a function, complete monomial terms up to the 2nd-order are included in Eq. (5.3), namely the 2nd-order continuity is enforced on the refined values of nodal parameter.

5.4.1 1-D test problems

The first 1-D problem is a mixed boundary condition problem with the 4th-order monomial in the analytical solution. The problem is approximated by including up to the

2nd-order monomial terms in the polynomial basis of function approximation. The governing equation and boundary conditions are given as

$$\frac{d^2 f}{dx^2} = \frac{105}{2}x^2 - \frac{15}{2}, \quad (-1 < x < 1) \quad (5.13)$$

$$f(x = -1) = 1, \quad \frac{df}{dx}(x = 1) = 10 \quad (5.14)$$

The analytical solution is given as $f(x) = (35/8)x^4 - (15/4)x^2 + (3/8)$. This problem is solved by the ARDQ method with the domain discretized by 21 uniform and randomly distributed field nodes separately, at the beginning of computation, and combined with 1001 and 641 uniform and cosine distributed virtual nodes. Fig. 5.3 shows the successive distributions of the field nodes according to the adaptive algorithm when 21 number of total random field nodes are used at the beginning of the adaptive refinement. Fig. 5.4a and Fig. 5.4b show the convergence plots by the uniform and random distributions of the field nodes at the beginning of computation, respectively, and the corresponding convergence rates are given in Table 5.1 and Table 5.2. Fig. 5.5 shows the reduction in the true error norm, and the comparison between the numerical and analytical values of the function.

Fig. 5.3. Distributions of the field nodes during the successive adaptive iterations when 21 randomly distributed field nodes at the beginning of computation are combined with 1001 uniform virtual nodes for the first 1-D problem.

It is seen from Fig. 5.4 and Fig. 5.5 that the successive values of the global and true error norms are steadily reduced with the best reduction is obtained by the random field nodes

at the beginning of computation coupled with the uniform virtual nodes, as the true error is reduced from 0.21 to 1.3386×10^{-4} .

Table 5.1. Convergence rates obtained by the ARDQ method for the first 1-D problem of Poisson equation by combining the uniform field nodes at the beginning of computation with the cosine and uniform virtual nodes

Function	Convergence rates (cosine virtual nodes)	Convergence rates (uniform virtual nodes)
f	1.4	1.6
$f_{,x}$	1.43	1.6
$f_{,xx}$	0.9	0.9

The second 1-D problem is a locally high gradient problem, which is solved by the ARDQ method by including up to the 2nd-order monomial terms in the polynomial basis of function approximation. The governing equation and boundary conditions are given as

$$\frac{d^2 f}{dx^2} = -6x - \left[\left(\frac{2}{\alpha^2} \right) - 4 \left(\frac{x - \beta}{\alpha^2} \right)^2 \right] \exp \left[- \left(\frac{x - \beta}{\alpha} \right)^2 \right] \quad (0 < x < 1) \quad (5.15)$$

Fig. 5.4. Convergence plots when the uniformly (a) and randomly (b) distributed field nodes at the beginning of computation are combined with the uniform and cosine distributed virtual nodes for the first 1-D problem of Poisson equation.

Fig. 5.5. Reduction in the true error norm (a), where the lowest value is obtained by combining the random field nodes distributed at the beginning with the uniform virtual nodes, and the comparison between the numerical and analytical values of field variable distribution (b) when the first 1-D problem is solved by the ARDQ method.

$$f(x=0) = \exp\left[-\left(\frac{\beta^2}{\alpha^2}\right)\right], \quad \frac{df(x=1)}{dx} = -3 - 2\left(\frac{1-\beta}{\alpha^2}\right)\exp\left[-\left(\frac{1-\beta}{\alpha}\right)^2\right] \quad (5.16)$$

The analytical solution is given as $f(x) = -x^3 + \exp\left[-[(x-\beta)/\alpha]^2\right]$. This problem is solved with the domain discretized by 21 uniform and randomly distributed field nodes, at the beginning of computation, separately coupled with 641 uniform and cosine distributed virtual nodes. The successive distributions of the field nodes by the adaptive algorithm are shown in Fig. 5.6. The convergence plots for the uniform and random field nodes, at the beginning of computation, are shown in Fig. 5.7a and Fig. 5.7b, respectively, and the corresponding convergence rates are given in Table 5.3 and Table 5.4. The reduction in the true error norm is plotted in Fig. 5.8, and the corresponding values are given in Table 5.5 and Table 5.6. It is seen from Fig. 5.8 that the lowest value of error norm is obtained by combining the random field nodes, at the beginning of computation, with the uniform virtual nodes. Fig. 5.9 shows the comparison between the numerical and analytical values of function and the 1st- and 2nd-order derivatives.

It is seen from Fig. 5.9 that the local peak value of the function is correctly captured by the ARDQ method. Comparing the values of global error norms, obtained by the ARDQ method coupled with the error recovery technique, with that of the RDQ method by the uniform increment in the field nodes, it is seen that the ARDQ method has much

lower value of the global error norm, as compared with the RDQ method, for the same number of field nodes in the domain, as also observed from Table 5.7.

Table 5.2. Convergence rates for the first 1-D problem of Poisson equation by the ARDQ method when the random field nodes at the beginning of computation are combined with the cosine and uniform virtual nodes

Function	Convergence rates (cosine virtual nodes)	Convergence rates (uniform virtual nodes)
f	2.31	3.8
$f_{,x}$	1.32	1.1
$f_{,xx}$	0.4	0.5

In summary, it is observed from all the solutions of 1-D test problems that the ARDQ method gives good rates of convergence for the locally high gradient problems. The local peak values are well captured, which may be attributed to the fact that the ARDQ method is a strong-form method, and it discretizes the governing differential equations and boundary conditions at every virtual node located within the domain and along the boundaries, respectively.

Fig. 5.6. Distributions of the field nodes during the adaptive iterations of ARDQ method when 21 randomly distributed field nodes at the beginning of computation are combined with the uniform virtual nodes for the second 1-D problem.

Fig. 5.7. Convergence curves when the uniformly (a) and randomly (b) distributed field nodes at the beginning of computation are combined with the uniform and cosine virtual nodes, during the solution of second 1-D problem by the ARDQ method.

The comparison between the RDQ method with the uniform increment of the field nodes and the ARDQ method coupled with the error recovery technique, as shown in Fig. 5.10, demonstrates that the global error has been reduced by implementing the error recovery technique in the ARDQ method, which leads to the reduction in the numerical pollution.

Fig. 5.8. Reduction in the true error norm with increase in the field nodes when the second 1-D problem is solved by the ARDQ method.

In order to compare the convergence rates obtained by the ARDQ and RDQ methods, the virtual nodes remain fixed, and the variation of the field nodes is allowed only. The adaptive algorithm stops the refinement, when the number of total field nodes becomes greater than the number of total virtual nodes. Therefore, if the field nodes are randomly distributed at the beginning, sometimes the adaptive nature of the refinement may not be

apparent, since the adaptive refinement stops when the total field nodes become more than the total virtual nodes.

Table 5.3. Convergence rates for the second 1-D problem of Poisson equation by combining the uniform field nodes at the beginning of computation with the cosine and uniform virtual nodes when solved by the ARDQ method

Function	Convergence rates (cosine virtual nodes)	Convergence rates (uniform virtual nodes)
f	4.22	3.2
$f_{,x}$	2.1	0.5
$f_{,xx}$	0.8	0.5

Table 5.4. Convergence rates for the second 1-D problem of Poisson equation by combining the random field nodes at the beginning of computation with the cosine and uniform virtual nodes when solved by the ARDQ method

Function	Convergence rates (cosine virtual nodes)	Convergence rates (uniform virtual nodes)
f	4.0	3.5
$f_{,x}$	1.3	0.45
$f_{,xx}$	0.5	0.5

Comparing the convergence rates obtained by the ARDQ method in this chapter with those by the RDQ method in the third chapter, it is noted that the ARDQ method achieves the values of equal or better convergence rates at relatively lower number of field nodes. This is the key result as it highlights one of the objectives of the development of the ARDQ method to achieve better convergence rates at relatively lower number of field nodes.

5.4.2 2-D test problems

Several 2-D test problems are solved numerically in this section, and the corresponding convergence analysis is performed.

The first 2-D problem is of steady-state heat conduction without an internal heat generation in the slab. There is no temperature gradient in the z -direction, but there is a

temperature gradient along the y direction. The governing equation and boundary conditions are given as

Fig. 5.9. Numerical and analytical distributions of the values of f (a), (df/dx) (b), and (d^2f/dx^2) (c) when the second 1-D problem is solved by the ARDQ method.

Table 5.5. Reduction in the true error norm for the second 1-D problem when the uniform and random field nodes at the beginning of computation are respectively combined with the cosine virtual nodes

Adaptive iteration number	1	2	3	4	5
Uniform field nodes	21	63	175	331	441
True error norm	2292.1	0.18865	0.02795	0.00743	0.00263
Random field nodes	21	63	189	253	
True error norm	20.5	0.252	0.00203	0.001444	

$$\frac{d^2T}{dx^2} + \frac{d^2T}{dy^2} = 0, \quad (0 < x < a) \text{ and } (0 < y < b) \quad (5.17)$$

Table 5.6. Reduction in the true error norm for the second 1-D problem when the uniform and random field nodes at the beginning of computation are separately combined with the uniform virtual nodes

Adaptive iteration	1	2	3	4	5	6	7
Uniform nodes	21	63	159	277	441		
True error norm	5.5	0.1	6.23×10^{-3}	1.1×10^{-3}	1.2×10^{-3}		
Random nodes	21	63	155	173	211	319	487
True error norm	10.6	0.1	1.63×10^{-3}	1.7×10^{-3}	1.6×10^{-3}	9.1×10^{-4}	5.4×10^{-4}

Fig. 5.10. Comparison between the values of global error norms obtained by solving the second 1-D problem using the ARDQ method coupled with the error recovery technique and the RDQ method with the uniform refinement of field nodes.

Table 5.7. Reduction in the global error norm when the second 1-D problem is solved by the ARDQ method coupled with the error recovery technique, and the RDQ method with the uniform increment in the field nodes

Field nodes	By RDQ method (%)	By ARDQ method (%)
21	1.843×10^2	1.843×10^2
63	13.4	8.526×10^{-2}
175	0.2332	1.21×10^{-2}
331	0.02225	2.363×10^{-3}
441	5.96×10^{-3}	7.238×10^{-4}

$$T(x=0)=0, \quad T(x=a)=0, \quad T(y=0)=0 \text{ and } T(y=b)=100 \quad (5.18)$$

The analytical solution based on the Fourier series is given as

$$T(x, y) = T_s \sum_{n=1}^{\infty} [n\pi \sinh(n\pi b/a)]^{-1} 2[1 - (-1)^n] \sin(n\pi x/a) \sinh(n\pi y/a) . \quad \text{This}$$

problem is solved by separately combining the uniform and random distributions of the field nodes, at the beginning of computation, with the cosine and uniform distributions of the virtual nodes. The initial and final distributions of the field nodes for the total 25 random field nodes at the beginning of adaptive computation are shown in Fig. 5.11. The convergence plots of the temperature and curves of reduction in the true error norm are plotted in Fig. 5.12a and Fig. 5.12b, respectively, and the corresponding values are given in Table 5.8, Table 5.9 and Table 5.10. Fig. 5.13a and Fig. 5.13b show the distributions of temperature and its contour plots, respectively, corresponding to the final stage shown in Fig. 5.11b. It is seen from Fig. 5.13 that the gradient of temperature in the y direction is captured well and the boundary conditions are exactly imposed.

Fig. 5.14a shows the comparison of temperature at the field nodes along the $(0.5, y)$ when computed by the RDQ method with uniform increment in the field nodes, and the ARDQ method coupled with the error recovery technique. Fig. 5.14b shows the global error norms computed by both the RDQ and ARDQ methods, and it is seen that the temperature values computed by the ARDQ method are more accurate than by the RDQ method.

The second 2-D problem is also with a high local gradient at the location $(0.5, 0.5)$. The governing equation and boundary conditions are given as

$$\begin{aligned} \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} = -(6x) - (6y) - \left[\left(\frac{4}{\alpha^2} \right) - 4 \left(\frac{x-\beta}{\alpha^2} \right) - 4 \left(\frac{y-\beta}{\alpha^2} \right) \right] \times \\ \exp \left[- \left(\frac{x-\beta}{\alpha} \right)^2 - \left(\frac{y-\beta}{\alpha} \right)^2 \right], \quad (0 < x < 1) \text{ and } (0 < y < 1) \end{aligned} \quad (5.19)$$

Fig. 5.11. Distributions of the random field nodes at the beginning (a) and end (b) of the solution of first 2-D problem of steady-state heat conduction by the ARDQ method.

Fig. 5.12. Convergence curves when the uniform and random field nodes at the beginning of computation are combined with the cosine and uniform virtual nodes (a) and the reduction in true error norm (b), where the lowest true error is obtained by the uniform field nodes at the start combined with the uniform virtual nodes for the first 2-D problem.

$$f(x=1, y) = -1 - y^3 + \exp \left[- \left(\frac{(1-\beta)}{\alpha} \right)^2 - \left(\frac{(y-\beta)}{\alpha} \right)^2 \right] \quad (5.20)$$

Table 5.8. Convergence rates for the first 2-D problem when the uniform and random field nodes at the beginning of computation are combined with the cosine and uniform virtual nodes

	Convergence rates (uniform field nodes)	Convergence rates (random field nodes)
Cosine virtual nodes	0.7	1.0
Uniform virtual nodes	0.7	0.7

Fig. 5.13. Numerical distribution of the temperature (a) and the temperature contours over the uniform grid (b) by 625 field nodes when the first 2-D problem is solved by the ARDQ method.

Fig. 5.14. Distribution of temperature at the field nodes along (0.5, y) (a) and the reduction in global error norm (b), where the global error obtained by the ARDQ method is lower than the one obtained by RDQ method for the equal number of field nodes, when the first 2-D problem is solved by the ARDQ method.

$$\frac{df}{dy}(x, y=0) = \frac{2\beta}{\alpha^2} \exp \left[-\left(\frac{\beta}{\alpha} \right)^2 - \left(\frac{(x-\beta)}{\alpha} \right)^2 \right] \quad (5.21)$$

$$\frac{df}{dy}(x, y=1) = -3 - 2 \left(\frac{(1-\beta)}{\alpha^2} \right) \exp \left[-\left(\frac{(x-\beta)}{\alpha} \right)^2 - \left(\frac{(1-\beta)}{\alpha} \right)^2 \right] \quad (5.22)$$

The analytical solution is given as $f(x, y) = -x^3 - y^3 + \exp \{ -[(x-\beta)/\alpha]^2 - [(y-\beta)/\alpha]^2 \}$.

Table 5.9. Reduction in the true error norm for the first 2-D problem when the uniform and random field nodes at the beginning of computation are combined with cosine virtual nodes

	1	2	3
Uniform field nodes	25	125	625
True error norm	0.506	0.2477	0.1507
Random field nodes	25	125	621
True error norm	0.5136	0.2812	0.191

Table 5.10. Reduction in the true error norm for the first 2-D problem when the uniform and random field nodes at the beginning of computation are combined with uniform virtual nodes

Adaptive iteration	1	2	3
Uniform field nodes	25	125	625
True error norm	0.516	0.2606	0.1464
Random field nodes	25	125	609
True error norm	0.5172	0.2733	0.1836

Fig. 5.15. Initial total 25 random field nodes (a), and final total 625 field nodes (b) during the solution of the second 2-D problem by the ARDQ method.

This problem is solved by including up to the 2nd-order monomials in the polynomial basis of function approximation, and by separately combining the total 25 uniform and random distributions of the field nodes, at the start of computation, with the total 33×33 cosine virtual nodes. The initial and final nodal distributions are shown in Fig. 5.15.

Fig. 5.16. Convergence curves when the uniform and random field nodes at the beginning are separately combined with the cosine virtual nodes (a) and the reduction in true error norm (b) for the solution of the second 2-D problem by the ARDQ method.

The convergence plots are given in Fig. 5.16a, and the values are obtained as 2.3 and 2.4 for the uniform and random field nodes, respectively. The reduction in the true error norm is shown in Fig. 5.16b, and the comparison between the numerical and analytical values of function and its corresponding derivatives are given from Fig. 5.17 to Fig. 5.21.

Fig. 5.17. Comparison between the analytical (a) and numerical (b) values of function for the second 2-D problem by the total 876 field nodes obtained at the end of the adaptive refinement by the ARDQ method.

The third 2-D problem is of steady-state heat conduction with a heat source. The governing equation and boundary conditions are given as

$$\nabla^2 T = -2 s^2 \sec h^2 [s(y-0.5)] \tanh[s(y-0.5)], \quad (0 < x < 0.5) \text{ and } (0 < y < 1) \quad (5.23)$$

Fig. 5.18. Plots of the 1st-order derivative of function with respect to x $f_{,x}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 51×51 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.

$$\frac{\partial T(x=0)}{\partial x} = 0, \text{ and } \frac{\partial T(x=0.5)}{\partial x} = 0 \quad (5.24)$$

$$T(y=0) = -\tanh\left(\frac{s}{2}\right), \text{ and } T(y=1) = \tanh\left(\frac{s}{2}\right) \quad (5.25)$$

The analytical solution is given as $T = \tanh[s(y-0.5)]$. This problem is solved by the number of total 36 uniform and random field nodes, at the beginning, separately combined with the cosine and uniform virtual nodes. The initial (36 nodes) and final (896 nodes) distributions of the field nodes are shown in Fig. 5.22. The comparison between the convergence rates of the temperature, computed with and without the error recovery technique, is plotted in Fig. 5.23. The reduction in the true error norm is shown in Fig.

5.24, and the comparison between the analytical and numerical values of the temperature is shown in Fig. 5.25.

Fig. 5.19. Plots of the 1st-order derivative of function with respect to y $f_{,y}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 41×41 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.

The fourth 2-D problem is also having a local high gradient with a peak at $(0.5, 0.5)$. It is solved by including up to the 2nd-order monomials in the function approximation. The governing equation and boundary conditions are given as

$$\nabla^2 f = \left[-400 + (200x - 100)^2 + (200y - 100)^2 \right] e^{\left[-100 \left(x - \frac{1}{2} \right)^2 - 100 \left(y - \frac{1}{2} \right)^2 \right]}, \quad x, y \in [0, 1] \quad (5.26)$$

Fig. 5.20. Plots of the 2nd-order derivative of function with respect to x $f_{,xx}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 41×41 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.

$$\frac{\partial f}{\partial n} = 0 \quad \text{along } x = 0 \text{ and } y = 0, \text{ and } f = 0 \quad \text{along } x = 1 \text{ and } y = 1 \quad (5.27)$$

The analytical solution is given as $f(x, y) = e^{[-100 (x-1/2)^2 - 100 (y-1/2)^2]}$. This problem is solved by the number of total 36 field nodes uniformly distributed at the beginning of computation and the number of total 33×33 cosine virtual nodes. The initial and final distributions of the field nodes are shown in Fig. 5.26.

Fig. 5.21. Plots of the 2nd-order derivative of function with respect to y $f_{,yy}$, as analytical (a) and numerical (b) values, and contour plot by the uniform grid of 41×41 nodes (c) for the second 2-D problem of local high gradient when the number of total 876 field nodes is obtained at the end of adaptive refinement by the ARDQ method.

Fig. 5.22. Initial number of total 36 field nodes (a) and final number of total 896 field nodes (b) during the solution of the third 2-D problem by the ARDQ method.

Fig. 5.23. Convergence plots when the ARDQ method is coupled with the error recovery technique (a), and the convergence rates obtained with and without using the error recovery technique (b) when the third 2-D problem is solved by the ARDQ method.

Fig. 5.24. Reduction in the true error norm with increase in the field nodes when the third 2-D problem is solved by the ARDQ method.

Fig. 5.25. Comparison between the analytical (a) and numerical (b) distributions of the temperature for the third 2-D problem.

Fig. 5.26. Distributions of the initial number of total 36 uniform field nodes (a) and final number of total 900 field nodes (b) obtained during the solution of the fourth 2-D problem by the ARDQ method.

The convergence plot of the function is given in Fig. 5.27a, which shows that the convergence rate is indeed improved by the error recovery technique. The reduction in the true error norm with an adaptive refinement by the ARDQ method is shown in Fig. 5.27b. The comparison between the analytical and numerical values of the function is shown in Fig. 5.28, and the contour plots of the numerical values of function are given in Fig. 5.29.

Fig. 5.27. Convergence curves with and without using the error recovery technique (a) and the reduction in true error norm (b) when the fourth 2-D problem is solved by the ARDQ method.

Fig. 5.28. Comparison between the analytical (a) and numerical (b) values of function for the fourth 2-D problem of local high gradient solved by the ARDQ method.

5.4.3 *Semi-infinite plate with a central hole*

In this section, a semi-infinite plate with a central hole under the action of normal load is solved by the plane-stress criterion. Only one quarter of the domain is considered for the actual computation due to symmetric geometry and loading, as shown in Fig. 5.30. The equations of mechanical equilibrium in the displacement form according to the plane-stress condition are given as

$$\frac{E}{(1-\nu_0^2)} \left[\frac{\partial^2 u}{\partial x^2} + \left(\frac{1-\nu_0}{2} \right) \frac{\partial^2 u}{\partial y^2} + \left(\frac{1+\nu_0}{2} \right) \frac{\partial^2 v}{\partial x \partial y} \right] + B_x = 0 \quad (5.28)$$

$$\frac{E}{(1-\nu_0^2)} \left[\frac{\partial^2 v}{\partial y^2} + \left(\frac{1-\nu_0}{2} \right) \frac{\partial^2 v}{\partial x^2} + \left(\frac{1+\nu_0}{2} \right) \frac{\partial^2 u}{\partial x \partial y} \right] + B_y = 0 \quad (5.29)$$

where B_x and B_y , and u and v are the body forces and displacements, respectively, in the x and y directions, respectively, and E and ν_0 are the modulus of elasticity and Poisson ratio, respectively. The body forces B_x and B_y are neglected in the present results. The analytical solutions of this problem are given in Chapter 3.

This problem is solved by an adaptive refinement with initial the number of total 35 uniformly distributed nodes and the number of total 961 cosine virtual nodes. The distributions of the initial and final field nodes and the virtual nodes are shown in Fig. 5.31. This problem is also solved by the uniform refinement with the total 6×6 , 11×11 ,

21×21 and 31×31 field nodes combined with the total 34×34 cosine virtual nodes. The convergence curves corresponding to the uniform and adaptive refinements are plotted in Fig. 5.32a, and better convergence rates are obtained by the adaptive refinement.

Fig. 5.29. The analytical (a) and numerical (b) plots of field variable distribution, and contour plot by 41×41 uniform grid (c) for the fourth 2-D problem.

Fig. 5.30. The computational domain for the problem of semi-infinite plate with a central hole.

The resultant error in the displacements u and v is computed by $\sqrt{\varepsilon_u^2 + \varepsilon_v^2}$, where ε_u and ε_v are the global errors in the displacements u and v , respectively, and plotted in Fig. 5.32b. The normal stress σ_{xx} along the boundary $x = 0$ is plotted in Fig. 5.33a, and the contour plots of the analytical and numerical values, at the field nodes with the relative error within $\pm 2\%$, are given in Fig. 5.33b and Fig. 5.33c, respectively. It is observed from Fig. 5.33a and Fig. 5.33c that the local peak value of the normal stress σ_{xx} is well captured. The numerical and analytical values of the stress σ_{yy} along the boundary $y = 0$ are plotted in Fig. 5.34.

In summary, it is observed from the numerical results of 2-D test problems that the ARDQ method well captures the local peak values, which was one of the objectives behind the development. It is evident from Fig. 5.23b and Fig. 5.27a that the convergence rates achieved by the ARDQ method are improved by the error recovery technique, which indicates that the numerical pollution effect has been minimized by the error recovery technique. It is seen in Fig. 5.10, Fig. 5.14 and Fig. 5.32 that the global error is reduced with an improvement in the values of the field variable distribution when the solution, obtained by the ARDQ method with an adaptive refinement coupled with the error recovery technique, is compared with that of the RDQ method with the uniform increment of the field nodes. The various contour plots shown in Fig. 5.13, Fig. 5.18 to Fig. 5.21, and Fig. 5.29 indicate that the ARDQ method well captures the values of function and the 1st- and 2nd-order derivatives.

5.5 Summary

At first, a novel strong-form adaptive meshless method called the ARDQ method is presented in this chapter. In the ARDQ method, the governing PDE is solved by the RDQ method after each step of the adaptive refinement. An error recovery technique, based on the LS averaging, is proposed to overcome a difficulty in deciding the number of interpolation nodes to be included in the domain of fixed RKPM interpolation for the approximation of function near the local peak region.

Fig. 5.31. Distributions of initial total 35 uniform field nodes (a), final total 875 nodes (b), and total 961 cosine virtual nodes (c) for the solution of the problem of semi-infinite plate with a central hole.

Fig. 5.32. Convergence plots of the displacements (a) and the decrease in resultant true error norm (b) by the ARDQ method, where the convergence rates obtained by ARDQ method, based on the adaptive refinement, are higher than the RDQ method with the uniform refinement of nodes.

Fig. 5.33. Comparison between the numerical and analytical values of stress σ_{xx} along the boundary $x = 0$ (a), and analytical (b) and numerical (c) values of the stress σ_{xx} , where the relative error is within $\pm 2\%$ during the final distribution of field nodes.

Fig. 5.34. Comparison between the numerical and analytical values of stress σ_{yy} along the boundary $y = 0$ for the problem of semi-infinite plate with a central hole by the ARDQ method.

The refinement procedure in the ARDQ method is easy to implement, and can be directly extended to the 3-D problems with the difference of the shape of an interpolation domain from circular (2-D) to sphere (3-D).

Further, a novel approach with the cross product of vectors along with the convex hull property of the computational domain is used in the ARDQ method to ensure that all the newly created field nodes are always within the computational domain. The proposed approach only requires a closed boundary (regular or irregular). In the case of an irregular or non-convex boundary, a local parametric equation of the curve is developed, which can be used in conjunction with the proposed approach of the cross product. This has been demonstrated here by solving the problem of semi-infinite plate with a central hole with an irregular boundary.

Finally, the detailed convergence analysis of the ARDQ method is performed by solving several 1-D and 2-D test problems. It is seen that the convergence rates are improved by the proposed error recovery technique. The various contour plots demonstrate that the ARDQ method well captures the values of local peak and the 1st- and 2nd-order derivatives.

While performing the numerical computation, the absolute values of the global error norms are equally important, compared with the convergence rates. For some problems solved in this chapter, the convergence rates obtained by the ARDQ method are not significantly high, compared those with the RDQ method. A closer look at the global error norm reveals that, with the same number of field nodes, the global error by the ARDQ method is lower than that by the RDQ method, as given in Table 5.7. It is also observed that the adaptive refinement gives accurate result with the less number of field nodes as compared with the RDQ method. For example, for the 1st 1-D problem, 641 uniform field nodes give the global error equal to 1.0×10^{-3} by the RDQ method, while the 585 adaptive field nodes result in the global error equal to 4.6×10^{-5} . Therefore, it is concluded that the adaptive refinement technique with the ARDQ method can work well.

Over all, it is concluded that the ARDQ method, coupled with the error recovery technique and the convex hull approach of the computational domain, is a robust and good convergent adaptive meshless method.

Chapter 6

2-D simulation of pH-sensitive hydrogels by the RDQ method

6.1 Introduction

The objective of the present work is to simulate the response of the 2-D hydrogel when subjected to the varying pH of buffer solution. This is almost the first attempt of 2-D simulation of the pH-responsive hydrogels by the novel strong-form meshless RDQ method. The ionic diffusion between the hydrogel and solution is simulated by the system of Poisson-Nernst-Planck (PNP) equations, and the hydrogel swelling is captured by the mechanical equilibrium equations. So far this problem has been solved by simplifying it to the 1-D hydrogel deformation only. This simplification holds well if the hydrogel domain is regular and uniform. However, it is incorrect to assume the 1-D deformation for irregular domains, and it becomes truly the 2-D deformation problem.

At first, the PNP equations are studied for the type, as the elliptic, parabolic or hyperbolic PDE. A novel approach is proposed to correctly impose the Neumann boundary condition for the non-uniform boundary. Effects of the solution pH and initial fixed-charge concentration are also investigated on the swelling of the hydrogel. It is shown that the simulation results are in good qualitative agreement with the physics of the problem and the experimental results.

There have been an extensive research works performed by different research groups, in both the theory and experiment, in order to understand the behaviour of polyelectrolyte gels or hydrogels responsive to different environmental stimuli, such as the solution temperature, pH, and ionic strength etc. The key motivation behind this is that the hydrogels are excellent candidates, since they can convert different forms of the potentials like chemical or electrical ones into the mechanical work. Because of the responsive behaviour, the hydrogels can be effectively used in the controlled drug release, micro-fluidic flow control, and sensors etc.

The hydrogels generally are the triphasic mixtures containing polymer crosslinked network, interstitial fluid and fixed-charge ions, as shown in Fig. 6.1.

Fig. 6.1. Schematic of the charged hydrogel and solution (a) and schematic of the working principle of the RDQ method by irregular computational domain (b).

Depending on the nature of fixed-charge (anions or cations), the hydrogels are called as the acidic or basic hydrogels. In the present work, the acidic hydrogel responsive to the solution pH is studied. In the pH-responsive acidic hydrogels, anionic fixed-charge is bound to the polymer crosslinks, and the surrounding solution contains various salt ions (Na^+ and Cl^-) along with the H^+ ions. Due to the different values of ionic concentrations inside and outside of the hydrogel at the beginning, the electrical potential develops and initiates the diffusion of mobile ions from the solution into the hydrogel through the porous surface. Some diffused H^+ ions combine with the fixed-charge anions. The ionic diffusion between the hydrogel and solution causes the difference in the ionic concentration, which leads to the generation of osmotic pressure. This osmotic pressure serves as a surface traction, and then causes the deformation of hydrogel. This deformation is stored as the strain energy in the crosslinked polymers, which then try to restrict the hydrogel deformation by their stiffness. The hydrogel is said to be in equilibrium with the surrounding solution when it stops expanding, as the external traction is balanced by the hydrogel internal stress.

In 1943, Flory and Rehner [137-138] developed a statistical model to represent the polymer crosslinked network and its interaction with the solvent, without diffusion considered. Hon *et al.* [139] developed a set of equations for the mixture theory, based on the triphasic system from the generalized first law of thermodynamics, in which the ionic

concentration is expressed by the chemical energy. De and Aluru [140] used the mechanical equilibrium equation to compute the hydrogel deformation. After computing the concentration of H^+ ions, the other ionic concentrations are computed by Donnan membrane theory, and subsequently the osmotic pressure is computed. Li *et al.* [141-142] developed a chemo-electro-mechanical model with the PNP equations by incorporating the relationship between the concentrations of ionized fixed-charge groups and the diffusive hydrogen ion via the Langmuir isotherm. They also validated the model by 1-D simulation via the meshless Hermite-cloud method. Hong *et al.* [143] formulated a theory of mass transport and large deformation with the free energy of hydrogel and solution system. The free energy of the gel comes from two molecular processes: stretching the polymer network and mixing of the polymer and small molecules. The small molecules and long polymers are considered incompressible by enforcing a constraint via the Lagrange multipliers. Zhang *et al.* [144] developed the FEM model for transient analysis of the large deformation and mass transport in the hydrogels. They used the free energy of system by following the model of Hong *et al.* [143], and developed the governing equations in the weak form with the equations of mass conservation and mechanical equilibrium. Using the developed FEM model, they solved several numerical examples, such as the free swelling of 3-D gel cube, free swelling of a thin sheet, and swelling of a partially constrained gel. Although several models for ionic transportation and gel deformation have been developed, none of them fully captures the whole phenomenon, which is highly complex in nature.

Several researchers performed the numerical simulations of the hydrogel by considering different working conditions. Wallmersperger *et al.* [145-146] performed the FEM simulation of 1-D and 2-D hydrogel deformations by the system of PNP and mechanical equilibrium equations. They also studied the effects of chemical and mechanical stimulations on the gel deformation. In the work of Wallmersperger *et al.* [147-148], the chemo-electro-mechanical model was used for the analysis of time dependent effect due to an electrical stimulation, where the larger difference in the electric potential across the hydrogel results in the larger change in the values of ionic concentrations with time. Samson *et al.* [149] numerically solved the transient PNP equations by two different iterative schemes, viz. the Picard iteration and Newton-

Raphson (NR) iteration, and concluded that the Picard iteration technique cannot be used to solve the PNP equations for all the cases, while all the PNP equations can be easily coupled and solved for any number of ionic species with different valence numbers by the NR method. This is a very important result because the PNP equations are nonlinear PDE, which need to be solved iteratively by the stable iterative technique. Li *et al.* [25, 150-158] simulated the chemo-electro-mechanical model of the 1-D hydrogel subject to the different solution stimuli, such as the temperature, pH, ionic strength and electrical potential etc. They extensively investigated the behaviour of hydrogel by changing the various simulation parameters, such as the solution pH, initial fixed-charge concentration inside the gel, hydrogel Young's modulus, ionic strength of solution, and externally applied electrical potential. They also conducted 1-D transient analysis of the hydrogel subjected to external electric field. Luo *et al.* [159-161] studied the responsive behaviour of 1-D hydrogel due to the effect of the pH, external electric stimulus, and ionic strength of solution. They also developed the PNP and mechanical equilibrium equations in Lagrangian coordinates, and compared the results with the same set of equations in Eulerian coordinates. It is shown that the deformations of the hydrogel computed by the governing equations developed in the Eulerian and Lagrangian coordinate systems are almost equal for the lower values of pH, but the differences appear for the higher values of pH. The additional results about the 1-D hydrogel can be found [162-190].

The simulation of pH-sensitive hydrogel to this point in the literature is performed by simplifying it as the 1-D problem. This simplification is based on the fact that the hydrogel is constrained at the top and bottom surfaces in the micro fluidic flow channel, and the cross-section of the hydrogel is essentially circular. But it may deform unequally in the different directions when the cross-sectional geometry of the hydrogel is non-uniform. As a result, it becomes truly the 2-D problem. The objectives of the present work are to perform the multiphysics 2-D simulation of pH-sensitive hydrogel with the non-uniform boundary of domain, and to study the effects of change in the values of initial fixed-charge concentrations and Young's modulus of hydrogel, as well as the solution pH on the deformation of hydrogel. It is also interesting to see how the geometrical shape of the hydrogel at the dry or initial state affects the final deformation under the various values of the solution pH. Therefore, three different geometries of the

hydrogel disc are studied for the various values of solution pH and Young's moduli. In the present work, the RDQ method is implemented to solve the PNP equations over the hydrogel and solution domains for the simulation of ionic diffusion, and to solve the mechanical equilibrium equations over the hydrogel domain for the simulation of the hydrogel deformation. Since the PNP equations are nonlinear PDE, they are solved iteratively by the NR method till the convergence is reached.

The motivation behind the present work is to demonstrate the capability of the RDQ method for the simulation of complex multiphysics problems, such as the 2-D simulation of pH-sensitive hydrogels with the non-uniform moving boundary. There are several challenges in this problem, such as the nonlinear and complex nature of the 2-D PNP equations, the non-uniform and moving boundary of the hydrogel, and the sudden jump in the values of the ionic concentrations and electrical potential across the interface between the hydrogel and solution. As explained earlier, the challenges arising from the problem of the 2-D simulation of the pH-sensitive hydrogel quite demand to be overcome by the strong-form based meshless method. As such, the meshless method has to be consistent, convergent, stable and capable of effectively handling the local high gradients over the irregular boundaries of domain to overcome these challenges, as demonstrated by the RDQ method [80, 106, 191-193]. According to the literature survey done, so far the simulation of the smart hydrogel has been simplified to the 1-D problem only, and solved by several meshless methods. Probably this is the first attempt to solve it as the 2-D problem by a strong-form meshless method.

The subsequent sections of this chapter are organized as follows. The detailed 2-D simulation of the pH-sensitive hydrogel is presented in Section 2. In Section 3, the simulation results of pH-sensitive hydrogel are discussed. In Section 4, the effects of solution pH and initial fixed-charge concentration are investigated on the hydrogel deformation. Effects of the Young's moduli and different geometrical shapes of the hydrogel at the dry or initial state on the swelling of hydrogel are studied in Section 5. Finally, the summary is given in Section 6.

6.2 Model development of the 2-D pH-sensitive hydrogel

The RDQ method is implemented in this section to simulate the problem of 2-D pH-sensitive acidic hydrogel, where the ionic diffusion is simulated by solving 2-D PNP equations over the hydrogel and solution domains. The diffusion of each ionic species is captured by the Nernst-Planck (NP) equation corresponding to individual ionic species. The electric field, due to the uneven ionic distribution over the hydrogel and solution domains, is simulated by developing Poisson equation for the electrical potential. After the concentrations of all the mobile ions are obtained, the osmotic pressure is computed and imposed as the surface traction on the boundary of the hydrogel. Swelling of the hydrogel due to the osmotic pressure is computed by solving the mechanical equilibrium equations in the plane-stress state. The 2-D computational domain of the hydrogel is obtained by cutting a disc from the full hydrogel, such that the thickness of the disc is much smaller than the length and height. As a result, it becomes a 2-D plane-stress problem. The general model to solve the 2-D hydrogel can be found [141-155].

Fig. 6.2a shows the schematic of the hydrogel and solution domains, where L , W , and R are the hydrogel length, width and boundary radius, respectively. It is seen that the hydrogel domain is non-uniform due to the curved boundary. Since the hydrogel and solution domains are symmetric about the central x and y direction axes, as shown in Fig. 6.2a, only a quarter portion of the whole domain is considered for the actual computation. Fig. 6.2b shows the actual computational domains of the hydrogel and solution along with the symmetric boundary conditions for the solution domain. These multi-domains are utilized to solve the 2-D PNP equations for the concentrations of all the ionic species and the electrical potential. Fig. 6.2c shows the hydrogel domain along with the symmetric boundary conditions, which is used to solve the mechanical equilibrium equations for the hydrogel deformation in the xy plane.

6.2.1 Model of 2-D pH-sensitive hydrogel

In this chapter, the acidic hydrogel is modeled with the anions as the ionized fixed-charge groups bound to the polymeric chains of the hydrogel. The bath solution contains Na^+ , H^+ , and Cl^- mobile ions as $C_1 = C_{\text{Na}^+}$, $C_2 = C_{\text{H}^+}$, and $C_3 = C_{\text{Cl}^-}$, and ψ is the

electrical potential, as shown in Fig. 6.2b. In order to use the NR method, the governing equations and boundary conditions are written as the residual equations in the non-dimensional form to avoid any singularity in the Jacobian matrix during the NR iterations. The equation of mass conservation for the ionic species C_1 by the ionic diffusion is developed via the chemical potential and electrical flux due to the gradient in the electrical potential, and then by Reynold's transport theorem [155], we have

$$R_1 = 0 \Rightarrow \frac{\partial^2 \bar{C}_1}{\partial \bar{x}^2} + \frac{\partial^2 \bar{C}_1}{\partial \bar{y}^2} + (\eta Z_1) \left[\left(\frac{\partial \bar{C}_1}{\partial \bar{x}} \frac{\partial^2 \bar{\psi}}{\partial \bar{x}} \right) + \left(\frac{\partial \bar{C}_1}{\partial \bar{y}} \frac{\partial^2 \bar{\psi}}{\partial \bar{y}} \right) \right] + (\eta Z_1 \bar{C}_1) \left(\frac{\partial^2 \bar{\psi}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{\psi}}{\partial \bar{y}^2} \right) = 0 \quad (6.1)$$

Fig. 6.2. Schematic of the hydrogel and solution domains (a), schematic of the computational domains of the hydrogel and solution along with the solution boundary conditions (b) and schematic of the hydrogel computational domain along with the boundary conditions (c).

where $\bar{C}_1 = C_1 / C_{ref}$, $\bar{x} = x / L_{ref}$, $\bar{y} = y / L_{ref}$, and $\bar{\psi} = (\psi F) / (\eta RT)$ are the non-dimensionalized values of the concentration of ion C_1 , x and y coordinates, and electrical

potential, respectively, and C_{ref} , L_{ref} , F , η , R , and T are the reference concentration, reference length, Faraday constant, weighted coefficient, gas constant, and temperature, respectively. Eq. (6.1) is the 2-D NP equation for the ionic species C_1 . Similarly, the residual equations R_2 and R_3 can be written for the ionic species C_2 and C_3 , respectively. All the governing equations are solved with the non-dimensionalized field variables to avoid any singularity in the Jacobian matrix of the NR iteration due to different scaling of the parameters, and at the end, the non-dimensional solutions are converted back to their original form with the conversion equations given earlier. The flux due to convection is neglected while developing Eq. (6.1), as the surrounding solution is assumed to be stationary. There are several possible ways to describe the spatial charge. One of them, the electrical potential governed by 2-D Poisson equation [155], is adopted in the present work as

$$R_4 = 0 \Rightarrow \frac{\partial^2 \bar{\psi}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{\psi}}{\partial \bar{y}^2} = \frac{-F^2 L_{ref}^2 C_{ref}^2}{\epsilon_r \epsilon_0} (Z_1 \bar{C}_1 + Z_2 \bar{C}_2 + Z_3 \bar{C}_3 + Z_f \bar{C}_f) \quad (6.2)$$

where ϵ_r and ϵ_0 are the relative and vacuum permittivities of material, respectively, and Z_1 , Z_2 , Z_3 , and Z_f are the valance of ions C_1 , C_2 , C_3 , and C_f respectively, and $\bar{C}_f$ is the non-dimensionalized ionized concentration of the hydrogel fixed-charge. The residual equations R_1 to R_4 are applied over the internal computational domain $(\Pi + \Omega)$, as shown in Fig. 6.2b. Eq. (6.1) is applied at the virtual node (x_i, y_j) that is located in the internal domain of the solution as

$$R_1(x_i, y_j) = \{b_{ik}\} \{C_1^h(x_k, y_j)\} + \{b_{jl}\} \{C_1^h(x_i, y_l)\} + (Z_1 \eta) [\{a_{ik}\} \{C_1^h(x_k, y_j)\} \{a_{im}\} \{\psi^h(x_m, y_j)\} + \{a_{jl}\} \{C_1^h(x_i, y_l)\} \{a_{jn}\} \{\psi^h(x_i, y_n)\}] + (Z_1 \eta C_1^h(x_i, y_j)) [\{b_{im}\} \{\psi^h(x_m, y_j)\} + \{b_{jn}\} \{\psi^h(x_i, y_n)\}] \quad (6.3)$$

where $m = 1, 2, \dots, N_x$ and $n = 1, 2, \dots, N_y$ indices are used for the discretization of ψ in the x and y directions, respectively, and C_1^h and ψ^h are the values of approximate function. The discretized equations $R_2(x_i, y_j)$ and $R_3(x_i, y_j)$ are also written similarly to Eq. (6.3). The discretized equation $R_4(x_i, y_j)$ from Eq. (6.2) is given as

$$R_4(x_i, y_j) = [\{b_{im}\} \{\psi^h(x_m, y_j)\} + \{b_{jn}\} \{\psi^h(x_i, y_n)\}] = \lambda[Z_1 C_1^h(x_i, y_j) + Z_2 C_2^h(x_i, y_j) + Z_3 C_3^h(x_i, y_j) + Z_f C_f(x_i, y_j)] \quad (6.4)$$

The residual equations R_5 to R_{20} for the domain boundaries 5 to 8, as shown in Fig. 6.2b, are discretized at the virtual node (x_i, y_j) as

$$\begin{aligned} R_5 = 0 &\Rightarrow \left(\frac{\partial C_1}{\partial x} \right)_{(x_i, y_j)} = \sum_{k=1}^{N_x} a_{ik} C_1^h(x_k, y_j) = 0 \\ R_6 = 0 &\Rightarrow \left(\frac{\partial C_2}{\partial x} \right)_{(x_i, y_j)} = \sum_{k=1}^{N_x} a_{ik} C_2^h(x_k, y_j) = 0 \end{aligned} \quad (6.5)$$

$$\begin{aligned} R_7 = 0 &\Rightarrow \left(\frac{\partial C_3}{\partial x} \right)_{(x_i, y_j)} = \sum_{k=1}^{N_x} a_{ik} C_3^h(x_k, y_j) = 0 \\ R_8 = 0 &\Rightarrow \left(\frac{\partial \psi}{\partial x} \right)_{(x_i, y_j)} = \sum_{k=1}^{N_x} a_{ik} \psi^h(x_k, y_j) = 0 \end{aligned} \quad (6.6)$$

$$\begin{aligned} R_9 = 0 &\Rightarrow \left(\frac{\partial C_1}{\partial y} \right)_{(x_i, y_j)} = \sum_{l=1}^{N_y} a_{jl} C_1^h(x_i, y_l) = 0 \\ R_{10} = 0 &\Rightarrow \left(\frac{\partial C_2}{\partial y} \right)_{(x_i, y_j)} = \sum_{l=1}^{N_y} a_{jl} C_2^h(x_i, y_l) = 0 \end{aligned} \quad (6.7)$$

$$\begin{aligned} R_{11} = 0 &\Rightarrow \left(\frac{\partial C_3}{\partial y} \right)_{(x_i, y_j)} = \sum_{l=1}^{N_y} a_{jl} C_3^h(x_i, y_l) = 0 \\ R_{12} = 0 &\Rightarrow \left(\frac{\partial \psi}{\partial y} \right)_{(x_i, y_j)} = \sum_{l=1}^{N_y} a_{jl} \psi^h(x_i, y_l) = 0 \end{aligned} \quad (6.8)$$

$$\begin{aligned} R_{13} \text{ and } R_{17} = 0 &\Rightarrow C_1^h(x_i, y_j) - \tilde{C}_1(x_i, y_j) = 0 \\ R_{14} \text{ and } R_{18} = 0 &\Rightarrow C_2^h(x_i, y_j) - \tilde{C}_2(x_i, y_j) = 0 \end{aligned} \quad (6.9)$$

$$\begin{aligned} R_{15} \text{ and } R_{19} = 0 &\Rightarrow C_3^h(x_i, y_j) - \tilde{C}_3(x_i, y_j) = 0 \\ R_{16} \text{ and } R_{20} = 0 &\Rightarrow \psi^h(x_i, y_j) = 0 \end{aligned} \quad (6.10)$$

where $k = 1, 2, \dots, N_x$ and $l = 1, 2, \dots, N_y$. Eqns. (6.5) and (6.6), and (6.7) and (6.8) are applied over the boundaries 5 and 6, respectively, and Eqns. (6.9) and (6.10) are applied over the boundaries 7 and 8.

The 2-D plane-stress mechanical equilibrium equation for the hydrogel domain is as

$$\nabla \cdot [(-P_{os}) I + \sigma_{ij}] = 0 \quad (6.11)$$

where I and σ_{ij} are the identity matrix and Cauchy stress tensor, respectively, and P_{os} is the osmotic pressure computed by the concentration values of the mobile ionic species across the interface between the hydrogel and solution as

$$\bar{P}_{os} = R T C_{ref} \sum_{k=1}^3 (\bar{C}_k - \bar{C}_{k0}) \quad (6.12)$$

where $\bar{P}_{os}$ is non-dimensionalized osmotic pressure, and $\bar{C}_k$ and $\bar{C}_{k0}$ are the non-dimensionalized concentrations of the k^{th} mobile ion in the hydrogel and solution domains, respectively. Eq. (6.11) can be expanded as

$$\left(\frac{E}{1-\nu_0^2} \right) \left[\frac{\partial^2 u}{\partial x^2} + \left(\frac{1-\nu_0}{2} \right) \frac{\partial^2 u}{\partial y^2} + \left(\frac{1+\nu_0}{2} \right) \frac{\partial^2 v}{\partial x \partial y} \right] = \frac{\partial P_{os}}{\partial x} \quad (6.13)$$

$$\left(\frac{E}{1-\nu_0^2} \right) \left[\frac{\partial^2 v}{\partial y^2} + \left(\frac{1-\nu_0}{2} \right) \frac{\partial^2 v}{\partial x^2} + \left(\frac{1+\nu_0}{2} \right) \frac{\partial^2 u}{\partial x \partial y} \right] = \frac{\partial P_{os}}{\partial y} \quad (6.14)$$

where u and v are the displacements of hydrogel in the x and y directions, respectively. Eqns. (6.13) and (6.14) are solved over the hydrogel internal domain Ω along with the symmetric boundary conditions, as shown in Fig. 6.2c, and given as

$$u = 0 \text{ and } \sigma_{xy} = 0 \text{ along Boundary 1; } v = 0 \text{ and } \sigma_{xy} = 0 \text{ along Boundary 2} \quad (6.15)$$

$$\sigma_n = P_{os} \text{ along Boundary 3} \quad (6.16)$$

$$\therefore T_x = P_{os} n_x = \sigma_n n_x = \sigma_{xx} n_x + \sigma_{yx} n_y, T_y = P_{os} n_y = \sigma_n n_y = \sigma_{xy} n_x + \sigma_{yy} n_y \quad (6.17)$$

where T_x and T_y , and n_x and n_y are the traction components and surface normals in the x and y directions, respectively, and σ_{xx} , σ_{yy} , and σ_{xy} are the normal and shear stress

components, respectively. Along Boundary 3, the surface normals for the curved boundary are given as $n_x = \cos(\theta)$ and $n_y = \sin(\theta)$.

$$\sigma_{xy} = 0 \text{ and } \sigma_{yy} = P_{os} \text{ along boundary 4} \quad (6.18)$$

The plane-stress equations for the stresses are given as [87]

$$\sigma_{xx} = \left(\frac{E}{1-\nu_0^2} \right) \left[\frac{\partial u}{\partial x} + \nu_0 \frac{\partial v}{\partial y} \right], \sigma_{yy} = \left(\frac{E}{1-\nu_0^2} \right) \left[\frac{\partial v}{\partial y} + \nu_0 \frac{\partial u}{\partial x} \right], \sigma_{xy} = \left(\frac{E}{2(1+\nu_0)} \right) \left[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] \quad (6.19)$$

As such, Eqns. (6.15) to (6.19) are applied over the hydrogel boundaries, and the final system of equations is solved for the displacements u and v at the field nodes of the hydrogel.

It is essential to know the classification of the PNP equations, namely the elliptic, parabolic or hyperbolic, to decide the discretization stencils for the derivative terms from the governing equations. Therefore, the PNP system is initially studied for the classification, and then the spatial discretization stencils are decided.

6.2.2 Classification of the PNP equations

In this section, the PNP equations are studied for their nature, namely the elliptic, parabolic or hyperbolic PDE. Let us consider the residual equations R_1 and R_4 , as given in Eqns. (6.1) and (6.2), respectively. In order to ensure the continuity of $C_{1,x}$, $C_{1,y}$, $\psi_{,x}$, and $\psi_{,y}$ along the characteristic line, where $_{,x}$ and $_{,y}$ indicate the derivatives with respect to the x and y directions respectively, the total derivatives are given as

$$dC_{1,x} = \frac{\partial C_{1,x}}{\partial x} dx + \frac{\partial C_{1,x}}{\partial y} dy \Rightarrow C_{1,xx} dx + C_{1,xy} dy \quad (6.20)$$

$$dC_{1,y} = \frac{\partial C_{1,y}}{\partial x} dx + \frac{\partial C_{1,y}}{\partial y} dy \Rightarrow C_{1,yx} dx + C_{1,yy} dy \quad (6.21)$$

$$d\psi_{,x} = \frac{\partial \psi_{,x}}{\partial x} dx + \frac{\partial \psi_{,x}}{\partial y} dy \Rightarrow \psi_{,xx} dx + \psi_{,xy} dy \quad (6.22)$$

$$d\psi_{,y} = \frac{\partial \psi_{,y}}{\partial x} dx + \frac{\partial \psi_{,y}}{\partial y} dy \Rightarrow \psi_{,yx} dx + \psi_{,yy} dy \quad (6.23)$$

It is written from Eqns. (6.1) to (6.2), and (6.20) to (6.23) that

$$\begin{bmatrix} 1 & 0 & 1 & \lambda_1 C_1 & 0 & \lambda_1 C_1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ dx & dy & 0 & 0 & 0 & 0 \\ 0 & dx & dy & 0 & 0 & 0 \\ 0 & 0 & 0 & dx & dy & 0 \\ 0 & 0 & 0 & 0 & dx & dy \end{bmatrix} \begin{bmatrix} C_{1,xx} \\ C_{1,xy} \\ C_{1,yy} \\ \psi_{,xx} \\ \psi_{,xy} \\ \psi_{,yy} \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \end{bmatrix} \Rightarrow A f = R \quad (6.24)$$

where $\lambda_1 = Z_1(F / RT) Z_1$. As the 2nd-order derivative may be discontinuous, thus $|A| = 0$

$$\therefore |A| = 0 \Rightarrow dx^4 + 2dx^2dy^2 + dy^4 = 0 \quad (6.25)$$

$$\therefore \left(\frac{dy}{dx}\right)^4 + 2\left(\frac{dy}{dx}\right)^2 + 1 = 0 \Rightarrow M^2 + 2M + 1 = 0 \quad (6.26)$$

where $(dy / dx)^2 = M$. The solution of Eq. (6.26) is given as

$$M = -1 \Rightarrow \frac{dy}{dx} = i \quad (6.27)$$

As the roots are imaginary, the residual equations R_1 and R_4 have the elliptic nature. Therefore, as the PNP equations are coupled together, they give the elliptic PDE.

It is concluded from this section that, the PNP equations are elliptic in nature, and there is no need to choose a specific technique of spatial stencil (namely the upwind or downwind). As a result, 2 virtual nodes from either sides of the concerned virtual node and the concerned virtual node itself are chosen in the present work for discretizing the governing equations and boundary conditions at the concerned virtual node.

6.2.3 Derivation of the analytical displacements along the moving boundaries 3 and 4 of the hydrogel

In this section, the analytical expressions of the displacements u and v are derived by the semi-inverse method [87]. Let ϕ be the 2nd-order Airy stress function as

$$\phi = a x^2 + b y^2 \quad (6.28)$$

The stresses along the hydrogel boundaries are given as

$$\sigma_{xx} = \frac{\partial^2 \phi}{\partial y^2} = 2b = P_x \quad \therefore \Rightarrow b = \frac{P_x}{2}, \quad \sigma_{yy} = \frac{\partial^2 \phi}{\partial x^2} = 2a = P_y \quad \therefore \Rightarrow a = \frac{P_y}{2} \quad (6.29)$$

where P_x and P_y are the components of osmotic pressure in the x and y directions, respectively. It is noted that the substitution, $\sigma_x = P_x$ and $\sigma_y = P_y$, is valid only for the boundaries 3 and 4, respectively, when they are uniform.

$$\therefore \phi = \frac{P_y}{2} x^2 + \frac{P_x}{2} y^2 \quad (6.30)$$

It is verified by Eq. (6.30) that the compatibility equation $\nabla^4 \phi = 0$ is satisfied. The strains are then computed as

$$\epsilon_{xx} = \frac{\partial u}{\partial x} = \frac{1}{E} [\sigma_x - \nu_0 \sigma_y] \Rightarrow \epsilon_{xx} = \frac{1}{E} [P_x - \nu_0 P_y] \quad (6.31)$$

Integrating Eq. (6.31) results in

$$u = \frac{x}{E} [P_x - \nu_0 P_y] + c_1 \quad (6.32)$$

$$u = 0 \text{ at } x = 0 \Rightarrow c_1 = 0, \Rightarrow u = \frac{x}{E} [P_x - \nu_0 P_y] \quad (6.33)$$

The displacement in the y direction is similarly computed as

$$\epsilon_{yy} = \frac{\partial v}{\partial y} = \frac{1}{E} [\sigma_y - \nu_0 \sigma_x] \Rightarrow \epsilon_{yy} = \frac{1}{E} [P_y - \nu_0 P_x] \quad (6.34)$$

Integrating Eq. (6.34) results in

$$v = \frac{y}{E} [P_y - \nu_0 P_x] + c_2 \quad (6.35)$$

$$v = 0 \text{ at } y = 0 \Rightarrow c_2 = 0, \Rightarrow v = \frac{y}{E} [P_y - \nu_0 P_x] \quad (6.36)$$

Therefore, Eqns. (6.33) and (6.36) give the displacements u and v , respectively, over the field nodes along the hydrogel boundaries 3 and 4, respectively, and it is seen that they are 1st-order continuous when P_x and P_y are constant, namely not the functions of x or y nodal coordinates. This observation will be verified while discussing the simulation results for the square geometry of the hydrogel.

In summary, the analytical solutions of the displacements u and v over the nodes along the hydrogel boundaries 3 and 4 are derived for the uniform boundary, and noted that the displacements are 1st-order continuous.

6.2.4 Discretization of PNP equations and the formulation of Jacobian

In this section, the discretization of 2-D PNP equations and the formulation of Jacobian matrix in the NR iteration are discussed. In order to obtain the valid solution at each iteration step, it is ensured that the Jacobian matrix is not singular, such that its inverse exists. Therefore, the direct inverse of Jacobian matrix is avoided in the present work, and instead it is solved by the Gauss elimination method.

The residual equations R_1 to R_{20} given by Eqns. (6.3) to (6.10) are used to formulate the Jacobian matrix in the NR method. Let N be the number of total virtual nodes in the computational domain of the PNP equations as shown in Fig. 6.2b, NV be the number of total virtual nodes in the internal computational domain (i.e. excluding boundaries) of the PNP equations, and NV_5 , NV_6 , NV_7 and NV_8 be the numbers of total virtual nodes along the boundaries of solution 5 to 8, respectively, as shown in Fig. 6.2b. The Jacobian matrix is formulated by taking the derivative of each residual equation discretized at N virtual nodes. Let the residual equation R_1 be discretized at all NV virtual nodes and represented as R_1^j , where $j \in [1, NV]$. The portion of the Jacobian matrix corresponding to the discretized equations R_1^j is given as

$$\begin{bmatrix} \frac{\partial R_1^1}{\partial C_1^1} & \cdots & \frac{\partial R_1^1}{\partial C_1^N} & \frac{\partial R_1^1}{\partial C_2^1} & \cdots & \frac{\partial R_1^1}{\partial C_2^N} & \frac{\partial R_1^1}{\partial C_3^1} & \cdots & \frac{\partial R_1^1}{\partial C_3^N} & \frac{\partial R_1^1}{\partial \psi^1} & \cdots & \frac{\partial R_1^1}{\partial \psi^N} \\ \frac{\partial R_1^2}{\partial C_1^1} & \cdots & \frac{\partial R_1^2}{\partial C_1^N} & \frac{\partial R_1^2}{\partial C_2^1} & \cdots & \frac{\partial R_1^2}{\partial C_2^N} & \frac{\partial R_1^2}{\partial C_3^1} & \cdots & \frac{\partial R_1^2}{\partial C_3^N} & \frac{\partial R_1^2}{\partial \psi^1} & \cdots & \frac{\partial R_1^2}{\partial \psi^N} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial R_1^{NV}}{\partial C_1^1} & \cdots & \frac{\partial R_1^{NV}}{\partial C_1^N} & \frac{\partial R_1^{NV}}{\partial C_2^1} & \cdots & \frac{\partial R_1^{NV}}{\partial C_2^N} & \frac{\partial R_1^{NV}}{\partial C_3^1} & \cdots & \frac{\partial R_1^{NV}}{\partial C_3^N} & \frac{\partial R_1^{NV}}{\partial \psi^1} & \cdots & \frac{\partial R_1^{NV}}{\partial \psi^N} \end{bmatrix}_{NV \times 4N} \quad (6.37)$$

An individual term from Eq. (6.37) is given as

$$\frac{\partial R_i^j}{\partial f^k} \quad (6.38)$$

where i , j and k represent the residual equation number, such that $i \in [1, 20]$, index of the virtual node from the internal computational domain of the PNP equations, such that $j \in [1, NV]$, and the overall index of virtual nodes, such that $k \in [1, N]$, respectively. Therefore, Eq. (6.38) is the $(R_i)^{\text{th}}$ residual equation discretized at the j^{th} virtual node from the internal computational domain, and its derivative is taken with respect to the function value at the k^{th} virtual node from the full computational domain. Jacobian matrix for the remaining residual equations is constructed as Eq. (6.37), to get the complete Jacobian matrix $[J]_{(4N \times 4N)}$. Let the residual equation R_i , as given in Eq. (6.3), be discretized at the i^{th} virtual node, then Eq. (6.3) is modified as

$$\begin{aligned} R_1(x_i, y_i) = R_1^i = & \{b_{ik}\} \{C_1^h(x_k, y_k)\} + \{b_{il}\} \{C_1^h(x_l, y_l)\} + (Z_1 \eta) [\{a_{ik}\} \{C_1^h(x_k, y_k)\} \times \\ & \{a_{im}\} \{\psi^h(x_m, y_m)\} + \{a_{il}\} \{C_1^h(x_l, y_l)\} \{a_{in}\} \{\psi^h(x_n, y_n)\}] + (Z_1 \eta C_1^h(x_i, y_i)) \times \\ & [\{b_{im}\} \{\psi^h(x_m, y_m)\} + \{b_{in}\} \{\psi^h(x_n, y_n)\}] \end{aligned} \quad (6.39)$$

Three cases are possible to compute $(\partial R_1^i / \partial C_1^j)$. The first case is that, if $j \in k$ and l , where $k = 1, 2, \dots, N_x$ and $l = 1, 2, \dots, N_y$

$$\frac{\partial R_1^i}{\partial C_1^j} = b_{ij} + b_{ij} + (Z_1 \eta) [a_{ij} \{a_{im}\} \{\psi^h(x_m, y_m)\} + a_{ij} \{a_{in}\} \{\psi^h(x_n, y_n)\}] +$$

$$(Z_1 \eta \delta_{ij}) [\{b_{im}\} \{\psi^h(x_m, y_m)\} + \{b_{in}\} \{\psi^h(x_n, y_n)\}] \quad (6.40)$$

The second case is that, if $j \in k$

$$\frac{\partial R_1^i}{\partial C_1^j} = b_{ij} + (Z_1 \eta) [a_{ij} \{a_{im}\} \{\psi^h(x_m, y_m)\}] + (Z_1 \eta \delta_{ij}) [\{b_{im}\} \{\psi^h(x_m, y_m)\} + \{b_{in}\} \{\psi^h(x_n, y_n)\}] \quad (6.41)$$

The third case is that, if $j \in l$

$$\frac{\partial R_1^i}{\partial C_1^j} = b_{ij} + (Z_1 \eta) [a_{ij} \{a_{in}\} \{\psi^h(x_n, y_n)\}] + (Z_1 \eta \delta_{ij}) [\{b_{im}\} \{\psi^h(x_m, y_m)\} + \{b_{in}\} \{\psi^h(x_n, y_n)\}] \quad (6.42)$$

Three cases are possible to compute $(\partial R_1^i / \partial \psi^j)$ as well. The first case is that, if

$j \in m$ and n , where $m = 1, 2, \dots, N_x$ and $n = 1, 2, \dots, N_y$

$$\frac{\partial R_1^i}{\partial \psi^j} = (Z_1 \eta) [a_{ij} \{a_{ik}\} \{C_1^h(x_k, y_k)\} + a_{ij} \{a_{il}\} \{C_1^h(x_l, y_l)\}] + (Z_1 \eta C_1^h(x_i, y_i)) [b_{ij} + b_{ij}] \quad (6.43)$$

The second case is that, if $j \in m$

$$\frac{\partial R_1^i}{\partial \psi^j} = (Z_1 \eta) [a_{ij} \{a_{ik}\} \{C_1^h(x_k, y_k)\}] + [Z_1 \eta C_1^h(x_i, y_i) b_{ij}] \quad (6.44)$$

The third case is that, if $j \in n$

$$\frac{\partial R_1^i}{\partial \psi^j} = (Z_1 \eta) [a_{ij} \{a_{il}\} \{C_1^h(x_l, y_l)\}] + [Z_1 \eta C_1^h(x_i, y_i) b_{ij}] \quad (6.45)$$

Similar to the derivatives of equations R_1^i , as given from Eqns. (6.40) to (6.45), the derivatives of the residual equations R_2^i and R_3^i are computed. Let the residual equation R_4 , as given in Eq. (6.4), be discretized at the i^{th} virtual node. Three cases are possible to compute $(\partial R_4^i / \partial \psi^j)$. The first case is that, if $j \in m$ and n

$$\frac{\partial R_4^i}{\partial \psi^j} = b_{ij} + b_{ij} \quad (6.46)$$

The second and third cases are that, if $j \in m$ or $j \in n$, respectively, as

$$\frac{\partial R_4^i}{\partial \psi^j} = b_{ij} \quad (6.47)$$

Other derivatives of R_4^i are computed as

$$\frac{\partial R_4^i}{\partial C_1^j} = -k Z_1 \delta_{ij}, \quad \frac{\partial R_4^i}{\partial C_2^j} = -k Z_2 \delta_{ij}, \quad \text{and} \quad \frac{\partial R_4^i}{\partial C_3^j} = -k Z_3 \delta_{ij} \quad (6.48)$$

Let the residual equation R_5 be discretized at the i^{th} virtual node such that, if $j \in k$

$$\frac{\partial R_5^i}{\partial C_1^j} = a_{ij} \quad (6.49)$$

The derivatives of the residual equations R_6^i to R_{12}^i are similarly computed. The derivatives of the residual equations R_{13} to R_{16} applied at the i^{th} virtual node are

$$\frac{\partial R_{13}^i}{\partial C_1^j} = \delta_{ij}, \quad \frac{\partial R_{14}^i}{\partial C_2^j} = \delta_{ij}, \quad \frac{\partial R_{15}^i}{\partial C_3^j} = \delta_{ij}, \quad \text{and} \quad \frac{\partial R_{16}^i}{\partial \psi^j} = \delta_{ij} \quad (6.50)$$

The derivatives of the residual equations R_{17} to R_{20} are similarly computed, and the complete Jacobian matrix is assembled. New values of the field variables at the virtual nodes are obtained after solving the Jacobean matrix as

$$\begin{aligned} (C_1^j)^{n+1} &= (1 - \text{SOR}) (C_1^j)^n + \text{SOR} \Delta C_1^j \\ (C_2^j)^{n+1} &= (1 - \text{SOR}) (C_2^j)^n + \text{SOR} \Delta C_2^j \\ (C_3^j)^{n+1} &= (1 - \text{SOR}) (C_3^j)^n + \text{SOR} \Delta C_3^j \\ (\psi^j)^{n+1} &= (1 - \text{SOR}) (\psi^j)^n + \text{SOR} \Delta \psi^j \end{aligned} \quad (6.51)$$

where n and SOR are the previous iteration count and the successive relaxation factor, respectively, and ΔC_1^j , ΔC_2^j , ΔC_3^j , and $\Delta \psi^j$ are the incremental values of the field variables obtained by solving the Jacobean matrix with the Gauss elimination method. Since the PNP equations are nonlinear in nature with the locally high gradient distributions of the field variables, they are prone to get unstable during the NR iterations, therefore the relaxation factor is used for the stable computation.

The formulation of the Jacobian matrix for the NR iteration is discussed in this section. The results obtained by this approach are presented in the subsequent sections.

6.2.5 Exact imposition of the Neumann boundary conditions

In order to correctly solve the PNP equations, the Neumann boundary conditions along the boundaries 5 and 6 are imposed exactly as

$$\frac{\partial C_1^i}{\partial x} = \frac{C_1^{next} - C_1^i}{\Delta x} = 0 \Rightarrow C_1^i = C_1^{next}, \text{ at boundary } x = 0 \quad (6.52)$$

$$\frac{\partial C_1^i}{\partial y} = \frac{C_1^{next} - C_1^i}{\Delta y} = 0 \Rightarrow C_1^i = C_1^{next}, \text{ at boundary } y = 0 \quad (6.53)$$

where i and $next$ are the indices of the virtual node on the Neumann boundary, and the immediate next virtual node in the specific x or y direction, respectively, and Δx and Δy are the spacings between the i^{th} and $next$ nodes in the x and y directions, respectively. The Neumann boundary condition is first converted to the Dirichlet boundary condition in this way, and then applied.

For the hydrogel domain shown in Fig. 6.2c, the last virtual node along the boundary $y = 0$ does not have any virtual node in the y direction, therefore following strategy is adopted while computing the gradient $\partial/\partial y$ at this virtual node. Let us consider the last virtual node along the boundary $y = 0$ with an id 6, and the last virtual node along the curved boundary with an id 11. The end portion of the hydrogel is enlarged, as shown in Fig. 6.3. The virtual node 6 is connected with the virtual node 11, and the Euclidean length between them is computed as $S = \sqrt{(\Delta x)^2 + (\Delta y)^2}$, as shown in Fig. 6.3.

Fig. 6.3. The computation of derivative in the y direction at the last virtual node along the boundary $y = 0$ of the hydrogel.

The gradient between the virtual nodes 6 and 11 along the direction S is given as

$$\left(\frac{\partial f}{\partial S}\right) = \frac{f^{11} - f^6}{S} \quad (6.54)$$

The gradient at the virtual node 6 in the y direction is given by Eq. (6.54) as

$$\left(\frac{\partial f}{\partial y}\right)_6 = \left(\frac{\partial f}{\partial S}\right)_{6,11} \cos(\theta) \Rightarrow \left\{ -\frac{\cos(\theta)}{S} \quad \frac{\cos(\theta)}{S} \right\} \begin{Bmatrix} f^6 \\ f^{11} \end{Bmatrix} \quad (6.55)$$

As a result, Eq. (6.55) is used to compute the gradient $\partial/\partial y$ at the virtual node 6.

6.2.6 Implementation of the RDQ method for the 2-D simulation of pH-sensitive hydrogels

In this section, implementation of the RDQ method is discussed in details to solve the problem of 2-D pH-sensitive hydrogel.

Since the PNP equations are solved iteratively by the NR method, the convergence is detected by the true error norm in the percentage as

$$\xi = 100 \sqrt{\sum_{i=1}^N \frac{[(\Delta C_1^i)^2 + (\Delta C_2^i)^2 + (\Delta C_3^i)^2 + (\Delta \psi^i)^2]}{[(C_1^i)^2 + (C_2^i)^2 + (C_3^i)^2 + (\psi^i)^2]^n}} \% \quad (6.56)$$

where N and n indicate the number of total virtual nodes in the solution domain, and the previous iteration count, respectively. Following procedure is adopted while iteratively solving the coupled PNP and mechanical equilibrium equations.

- The hydration H of the hydrogel is computed as

$$H = \frac{\text{current hydrogel volume}}{\text{initial dry solid hydrogel volume}} \quad (6.57)$$

The value $H = 1$ is taken at the beginning of NR iterations, since it is assumed that the volumes of the hydrogel in the current and initial dry states are identical. The concentration values of the ionized fixed-charge at the field and virtual nodes of the hydrogel are computed by Langmuir monolayer adsorption theory [141] as

$$Z_f C_f = -\frac{1}{H} \left[\frac{C_{f,s}^0 K_d}{(K_d + C_{H^+})} \right] \text{ (for acidic hydrogel)} \quad (6.58)$$

where K_d is the dissociation constant of hydrogel, and $C_{f,s}^0$ and C_{H^+} are the concentration values of the initial fixed-charge and mobile hydrogen ion within the hydrogel, respectively.

- The residual equations R_1 to R_{20} are applied over the virtual nodes in the solution domain to solve the PNP equations by the NR iterative method. Convergence of the NR method is ensured when $\xi \leq k^{PNP}$, where k^{PNP} is the predefined true error norm in the percentage.
- The values of the nodal parameters at the field nodes are computed after the convergence of PNP equations as

$$\{U^F\}_{n \times 1} = \bar{A}^{-1} \{f^V\}_{m \times 1} \quad (6.59)$$

$$\text{where } A = \begin{cases} [N]_{m \times n} & \text{if } m \equiv n \\ [N^T]_{n \times m} [N]_{m \times n} & \text{if } m \neq n \end{cases} \text{ and } \bar{A}^{-1} = \begin{cases} A^{-1} & \text{if } m \equiv n \\ A^{-1} [N^T]_{n \times m} & \text{if } m \neq n \end{cases} \quad (6.60)$$

where $[N]$ is the linear transformation matrix interpolating the values of the function at the virtual nodes with the values of nodal parameters at the surrounding field nodes, $\{U^F\}$ and $\{f^V\}$ are the vectors of the nodal parameter values at the field nodes, and the field variable values at the virtual nodes, respectively, and m and n are the numbers of total virtual and field nodes in the solution domain, respectively. As such, due to the lack of Kronecker delta property, $f^i \neq u^i$ where $i \in [1, n]$, the values of function at the field nodes are computed by interpolating each field node by the surrounding field nodes with the fixed RKPM interpolation as $f^h(x_i, y_i) = \sum_{k=1}^{NP} N_k u_k$, where the total NP

field nodes are used in the local interpolation domain of the i^{th} field node, such that $NP \subseteq N$, where N is the total field nodes in the whole domain, and N_k and u_k are the values of the shape function and nodal parameter at the k^{th} field node, respectively. All the values of functions, C_1 , C_2 , C_3 and ψ , at the field nodes are thus computed.

- The values of the osmotic pressure at the field nodes located along the hydrogel boundaries 3 and 4 are computed by Eq. (6.12). The mechanical equilibrium equations are solved for the hydrogel swelling by imposing the osmotic pressure as the Neumann boundary condition.
- The expanded hydrogel volume is evaluated by numerically computing the hydrogel area, and then multiplying it by the hydrogel thickness. The field nodes along the hydrogel boundaries 3 and 4 are utilized for tracking the moving boundaries by projecting them on the boundary 2. The deformed area of the hydrogel is subdivided, where each portion is formed by the nodes along the boundaries 3 and 4, and their projections on the boundary 2. Let Fig. 6.4 shows the portion of the deformed hydrogel domain with the field nodes 1 to 4 along the corners. The area covered by the nodes 1 to 4 is subdivided into two areas viz. A_1 and A_2 , which are computed by assuming them to be a rectangle and triangle, respectively, and given as

$$A_1 = (diff_y)(x_4 - x_3) \text{ and } A_2 = \frac{1}{2} |(y_2 - y_1)| (x_4 - x_3) \quad (6.61)$$

Fig. 6.4. Schematic of the approach to compute the expanded hydrogel area by the field nodes along the hydrogel boundaries 3 and 4, and their projection on the boundary 2.

The total area between the nodes 1 to 4 is given as $\Delta A = A_1 + A_2$. Similarly, the remaining area of the deformed hydrogel is computed by the subsequent strip portions,

and the total area is computed by adding all the individual areas ΔA . The new value of the hydration H is computed that is used during the next iteration of PNP equations.

- In this way, the PNP equations coupled with mechanical equilibrium equations are solved iteratively till they converge. The true error norm ζ for the values of the hydrogel displacements u and v is computed in the percentage as

$$\zeta = 100 \sqrt{\sum_{i=1}^N \frac{(u_i^{new} - u_i^{old})^2 + (v_i^{new} - v_i^{old})^2}{(u_i^{old})^2 + (v_i^{old})^2}} \% \quad (6.62)$$

where *new* and *old* indicate the values of deformation corresponding to the current and previous iterations, respectively, at N field nodes in the hydrogel. The convergence is detected when $\zeta \leq k^{PNPME}$, where k^{PNPME} is the predefined true error norm in the percentage.

In summary, the implementation of the RDQ method to solve the 2-D hydrogel for the pH response is discussed in this section. It is noted that the stability of the simulation depends on how the values k^{PNP} and k^{PNPME} are chosen suitably. For the results presented here, the values $k^{PNP} = 1 \times 10^{-4}$ and $k^{PNPME} = 1 \times 10^{-2}$ are used. It is also noted during the iterations between the PNP and mechanical equilibrium equations, the displacement values of the hydrogel field nodes are utilized for computing the expanded hydrogel volume and the hydration H , which consequently gives the new values of the concentration $C_{,f}$ of the fixed-charge at the field nodes of the hydrogel. However, the PNP and mechanical equilibrium equations are always solved by the virtual and field nodes in the original reference configuration X , such that no new discretization is required during the iterations. The simulation procedure of 2-D pH-sensitive hydrogel by the RDQ method is presented in Fig. 6.5 as a flowchart.

6.3 2-D simulation of pH-sensitive hydrogels by the RDQ method

The steady-state simulation results of 2-D pH-responsive hydrogels are presented in this section. In order to demonstrate the capability of RDQ method, a non-uniform hydrogel domain is defined as shown in Fig. 6.2a. The hydrogel and solution domains are discretized by the uniform or random field nodes, which are separately coupled with the

uniform virtual nodes. The dimensions of the hydrogel domain are considered as $W = 2$ mm, $L = 4$ mm and $R = 1$ mm, while the parameters of the solution domain are used as $W = 10$ mm and $L = 10$ mm, according to Fig. 6.2a.

Fig. 6.5. The complete flowchart of the 2-D simulation of the pH-responsive hydrogel by the RDQ method.

The simulation results are initially presented for the solution $\text{pH} = 3$ with the remaining parameters, as given in Table 6.1. The coupled PNP and mechanical equilibrium

equations are solved by first discretizing the solution and hydrogel domains by the uniform field nodes coupled with the uniform virtual nodes. Fig. 6.6 shows the distributions of the field variables, C_f , C_1 , C_2 , C_3 , and ψ , for the solution pH = 3 when the hydrogel and solution are in equilibrium with each other.

Table 6.1. List of the parameters used in the 2-D simulation of the pH-responsive hydrogels by the RDQ method

Parameter	Value	Unit
Na ⁺ mobile ion valance (Z_1)	+1	-
H ⁺ mobile ion valance (Z_2)	+1	-
Cl ⁻ mobile ion valance (Z_3)	-1	-
C_f fixed-charge ion valance (Z_f)	-1	-
Na ⁺ mobile ion concentration ($\tilde{C}_1$)	10.0	mM
H ⁺ mobile ion concentration ($\tilde{C}_2$)	1.0	mM
Cl ⁻ mobile ion concentration ($\tilde{C}_3$)	11.0	mM
$C_{f,s}^0$ initial fixed-charge ion concentration	1.050	mM
Temperature (k)	298	K
Gas constant (R)	8.314	J/(mol K)
Vacuum permittivity (ϵ_0)	8.8542×10^{-12}	C ² / (N m ²)
Relative permittivity (ϵ_r)	80	-
Dissociation constant (K_d)	0.02	mM
Faraday constant (F)	96485.3399	C/mol
Young's modulus (E)	29	MPa
Poisson's ratio (ν_0)	0.45	-

As the simulation is performed for the acidic hydrogel, the positive mobile ions diffuse into the hydrogel and combine with the negative fixed-charge groups leading to the higher values of the concentration of mobile ions, $C_{Na^+} = C_1$ and $C_{H^+} = C_2$, in the

hydrogel domain as compared with the solution domain, as seen in Fig. 6.6b and Fig. 6.6c, respectively. Simultaneously, the negative mobile ion $C_{\text{Cl}^-} = C_3$ has a lower value of concentration inside the hydrogel when compared with the solution, as shown in Fig. 6.6d and Fig. 6.6e, due to the repulsive force between the anionic fixed-charge and the mobile ion C_3 . Also, as the anionic fixed-charge groups are bound to the hydrogel crosslinked network of polymers, there is higher negative chemical potential inside the hydrogel than that in the surrounding solutions, which further reduces the diffusion of mobile ions C_3 into the hydrogel. Fig. 6.7 shows the distribution of cumulative concentrations of all the mobile ions ($C_1 + C_2 + C_3$) in the computational domain. It is seen that, while the solution almost satisfies the electroneutrality condition, the hydrogel has an average concentration value of mobile ions as 0.35 mM, which closely matches with the average negative fixed-charge concentration value of 0.36 mM inside the hydrogel, as shown in Fig. 6.6a. The net difference between all the ionic concentrations in the hydrogel leads to the constant electrical potential inside the hydrogel, as shown in Fig. 6.6f. However, the electrical potential is almost equal to zero in the surrounding solution, since there is an electroneutrality for the total concentrations of all the ionic species in the solution. It is seen from Fig. 6.7 that the electroneutrality condition is not exactly satisfied at few field nodes in the solution domain. As a result, negligible value of the electrical potential is present at these locations, as seen in Fig. 6.6f. Fig. 6.8 shows the swelling of hydrogel till the equilibrium is reached between the solution and hydrogel. The displacements u and v at the field nodes of the hydrogel and the comparison between the initial and deformed configurations of the hydrogel are shown in Fig. 6.9, and it is noted that the Dirichlet boundary conditions are imposed exactly. The profiles of the displacements u and v almost look like 1st-order continuous, except at the field nodes near the curved boundary.

The coupled PNP and mechanical equilibrium equations are again solved by discretizing the solution and hydrogel domains with the random field nodes coupled with the uniform virtual nodes. The random field nodes are generated with the uniform probability by the linear congruential generator algorithm [80].

Fig. 6.6. Schematic of the profiles of various field variables after solving the 2-D pH-responsive hydrogel for $\text{pH} = 3$, $E = 29$ and uniform field nodes, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), top view of C_3 (d), bottom view of C_3 (e) and ψ (f).

Fig. 6.7. Cumulative concentration values of the mobile ions ($C_1 + C_2 + C_3$) distributed in the hydrogel and solution domains, where the electroneutrality condition is satisfied in the solution and an average value of 0.35 mM is obtained in the hydrogel domain, which closely matches with the average concentration value of hydrogel fixed-charge of 0.36 mM.

Fig. 6.8. Increase in the hydrogel volume during the iterations of the coupled PNP and mechanical equilibrium equations, where the initial configuration volume was 1.75 mm².

Fig. 6.10 illustrates various profiles of the ionic concentrations in the hydrogel and solution. Fig. 6.10e shows that the average cumulative concentration value, 0.29 mM, of mobile ions in the hydrogel matches well with the concentration value, 0.29 mM, of

fixed-charge inside the hydrogel, as shown in Fig. 6.10a. The displacements u and v at the field nodes of the hydrogel, and the configurations of the hydrogel in the initial and swelled states are shown in Fig. 6.11.

In summary, the PNP and mechanical equilibrium equations are solved iteratively for the solution pH = 3 with the computational domains discretized by either the uniform or random field nodes.

Fig. 6.9. The displacements u (a) and v (b), and the comparison between the original and deformed hydrogel (c) for a curved boundary hydrogel with the uniform distribution of field nodes, where the central node is fully constrained and the Dirichlet boundary conditions are exactly imposed.

Fig. 6.10. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for pH = 3 and random field nodes, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), and the distribution of cumulative concentrations of mobile ions (e).

Fig. 6.11. The displacements u (a) and v (b), and the comparison between the original and deformed hydrogel (c) for a curved boundary hydrogel with the randomly distributed field nodes.

It is seen from the various distributions of the field variables that the RDQ method is capable of capturing the jump in the field variables across the interface between the multi-domain boundaries. It is observed by comparing Fig. 6.6 and Fig. 6.10 that the peak values of the concentrations of the mobile ions are almost identical when the domains are discretized by either the uniform or random field nodes. This demonstrates that the RDQ method evenly handle the uniform and random field nodes.

6.4 Effects of the solution pH and initial fixed-charge concentration on the swelling of 2-D hydrogel

In order to appropriately understand the swelling behaviour of 2-D pH-responsive hydrogel, it is essential to study the effects of solution pH and the initial fixed-charge concentration. Therefore, the PNP and mechanical equilibrium equations are solved together in this section by changing the values of solution pH as 3, 7 and 11, and the initial fixed-charge as 0.5, 1.05 and 1.5 mM, inside the hydrogel. Fig. 6.12 shows the distributions of all the mobile ions and electrical potential with the different values of initial fixed-charge concentration inside the hydrogel at solution pH = 3. It is easily verified from Fig. 6.12f that, while the solution satisfies the electroneutrality condition, the average cumulative concentration value of mobile ions inside the hydrogel matches closely with the average concentration of fixed-charge inside the hydrogel, as shown in Fig. 6.12a. Fig. 6.13 shows the displacements u and v , and the swelling of the hydrogel by different values of initial fixed-charge concentration at solution pH = 3.

It is seen from Fig. 6.13 that the displacements of the hydrogel are almost linear, except at the field nodes near the curved boundary, even at the higher concentrations of the initial fixed-charge inside the hydrogel. Fig. 6.14a and Fig. 6.14b, and Fig. 6.14c and Fig. 6.14d show the displacements u and v , respectively, at the hydrogel nodes corresponding to the solution pH = 7 and 11, respectively. It is observed by comparing Fig. 6.13 and Fig. 6.14 that, increasing the pH of solution and concentration of initial fixed-charge results in the larger swelling of the hydrogel. This is an expected behaviour according to the nature of the problem. The concentration of H^+ mobile ions reduces with an increase in the pH of solution. As such, more Na^+ ions have to diffuse into the hydrogel to maintain the electroneutrality by combining with the increasing values of the initial fixed-charge concentration, resulting in the higher gradient of the different concentrations across the interface between the solution and hydrogel. This results in higher osmotic pressure, and then larger swelling of the hydrogel. Fig. 6.15 demonstrates an increase in the swelling of the hydrogel with an increase in the values of the pH and initial fixed-charge concentration, as explained earlier.

Fig. 6.12. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for different concentration values of initial fixed-charge and pH = 3, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and cumulative mobile ions distribution (f).

Fig. 6.13. The displacements u (a) and v (b), and the comparison between the original and deformed hydrogel (c) for the different concentration values of the initial fixed-charge with the solution pH = 3.

A square hydrogel domain is considered with the dimensions $W = 2$ mm and $L = 2$ mm to compare the analytical and numerical values of the displacement at the field nodes along Boundaries 3 and 4. The PNP and mechanical equilibrium equations are solved over the square domain of hydrogel with pH = 3 and $C_{f,s}^0 = 1.05$ mM. Fig. 6.16 shows the x and y coordinates of the deformed hydrogel, and the analytical values corresponding to the field nodes along Boundaries 3 and 4 by Equations (6.33) and (6.36), respectively. It is observed in Fig. 6.16a that the final numerical values of x

coordinate at the field nodes along Boundary 3 are closely matching with the corresponding analytical values, similarly the final numerical values of y coordinate at the field nodes along Boundary 4 are closely matching with the corresponding analytical values in Fig. 6.16b.

Fig. 6.14. The displacements u (a) and v (b) for the solution pH = 7, and the displacements u (c) and v (d) for the solution pH = 11 at the different values of the initial fixed-charge concentration.

In summary, effects of the solution pH and the concentration of initial fixed-charge inside the hydrogel are studied in this section, and it is observed that the hydrogel swelling increases with the increase of the solution pH and the initial concentration of fixed-charge inside the hydrogel.

Fig. 6.15. Change in the volume of the hydrogel with respect to pH = 3, 7 and 11 as a function of initial fixed-charge concentration (a), and change in the volume of the hydrogel with respect to the values of initial fixed-charge concentration 0.5, 1.05 and 1.5 mM as a function of solution pH (b).

Fig. 6.16. Comparison between the numerical and analytical values of x (a) and y (b) coordinates along the boundaries 3 and 4, respectively, of the deformed square hydrogel, where both the deformations closely match each other.

6.5 Effects of the Young's moduli and geometrical shape of hydrogel at the dry state on the swelling of hydrogel

In this subsection, the 2-D simulation of pH-responsive hydrogel is performed by the Young's moduli, $E = 25, 29$ and 36 , and three different geometrical shapes of the hydrogel, as shown in Fig. 6.17, with the values of ratio $r = W / L$ as 1:1, 1:2, and 1:3.

The hydrogel with the geometrical shape, as shown in Fig. 6.17a, is already solved for the solution pH = 3 and Young's modulus $E = 29$, and the results are given from Fig. 6.6 to Fig. 6.9. It is solved further in this subsection for the strains and stresses inside the hydrogel. The expansion of the hydrogel causes the strains and stresses due to the stiffness of crosslinked polymer network. As a result, all the strains and stresses inside the hydrogel are plotted in Fig. 6.18 and Fig. 6.19, respectively.

Fig. 6.17. Schematic of the solution and various hydrogel domains, where the three geometrical shapes of the hydrogel are considered with the different values of the ratio $r = W / L$ as $r = 1 : 2$ (a), $r = 1 : 3$ (b) and $r = 1 : 1$ (c).

It is noted from Fig. 6.18 that the absolute nodal values of the strains are almost constant with the difference that ϵ_{xx} increases angularly from the y to x axis, and ϵ_{yy} increases angularly from the x to y axis. This is expected due to the Dirichlet boundary conditions

along the x and y axes. It is observed from Fig. 6.19 that the maximum values of the normal stresses are occurred at the central fully constrained node. It is also seen in Fig. 6.19c that the boundary condition of the shear stress $\sigma_{xy} = 0$ is exactly imposed.

Fig. 6.18. Profiles of different strains ϵ_{xx} (a), ϵ_{yy} (b) and ϵ_{xy} (c) when the 2-D pH-sensitive hydrogel is solved for pH = 3 and $E = 29$.

The computed state of stress inside the hydrogel is different at each field node due to the unequal values of the osmotic pressure along the curved boundary. As a result, by considering the state of stress at each field node, with one field node at a time, the corresponding orientation of the principle plane is determined by Mohr's circle [87], and the surface normals to the principle plane are plotted in Fig. 6.19d. It is verified from Fig. 6.19d that the field nodes along the boundaries $y = 0$ and $x = 0$ have the principle planes

normal to the x and y axes, respectively, which indicates $\sigma_{xy} = 0$ along these boundaries. This is expected due to the symmetric boundary conditions.

Fig. 6.19. Profiles of different stresses σ_{xx} (a), σ_{yy} (b) and σ_{xy} (c), and the surface normals to the principle planes (d) when the 2-D pH-sensitive hydrogel is solved for $\text{pH} = 3$ and $E = 29$.

This problem is solved with the hydrogel cross-section, as shown in Fig. 6.17a, by the solution $\text{pH} = 3, 7$ and 11 , and Young's moduli, $E = 25, 29$ and 36 , with the remaining parameters given in Table 6.1. All the profiles of the various field variables for $\text{pH} = 3$, and $E = 25, 29$ and 36 are shown in Fig. 6.20, and it is observed that the profiles of the mobile ions are as per the nature of the problem, and the solution satisfies the electroneutrality condition. Cumulative average value of the concentration of the mobile ions inside the hydrogel closely matches with the concentration value of the fixed-charge group, as seen from Fig. 6.20f and Fig. 6.20a.

Fig. 6.20. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for pH = 3, and $E = 25, 29$ and 36 , where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).

The displacements and expansion of the hydrogel are shown in Fig. 6.21, and it is noted that the hydrogel swelling is decreased with an increase in the value of the normalized

Young's modulus $\varepsilon = E/E_0$, where $E_0 = 25$. This is expected since the hydrogel crosslinked polymers offer more resistance to deformation for the higher values of E .

Fig. 6.21. The displacements u (a) and v (b), and the original and deformed shapes (c) for a curved boundary hydrogel with $\text{pH} = 3$, and $E = 25, 29$ and 36 , where the hydrogel expansion decreases with an increase in the ratio of the normalized Young's modulus ε .

Fig. 6.22 shows various profiles of the field variables for $\text{pH} = 7$, and $E = 25, 29$ and 36 . It is observed from Fig. 6.22 that the concentration values of the mobile ions inside the hydrogel increased slightly as compared with Fig. 6.20 due to the increase in the pH of the solution.

Fig. 6.22. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for pH = 7, and $E = 25, 29$ and 36 , where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), Ψ (e) and the distribution of cumulative mobile ions (f).

Fig. 6.22f shows that the solution satisfy the electroneutrality condition, while the cumulative average concentration value of mobile ions inside the hydrogel closely matches with the concentration value of negative fixed-charge ions, as seen in Fig. 6.22a.

Fig. 6.23. The displacements u (a) and v (b), and the original and deformed shapes (c) for a curved boundary hydrogel with $\text{pH} = 7$, and $E = 25, 29$ and 36 .

Fig. 6.23 shows the displacements at the field nodes of the hydrogel, and it is easily verified that the hydrogel swelling reduces with an increase in the value of E . Fig. 6.24 shows all the profiles of the field variables for $\text{pH} = 11$, and $E = 25, 29$ and 36 . It is seen from Fig. 6.24c that there is a jump in the concentration value of ion C_2 at the hydrogel boundary for $E = 36$, while the electroneutrality condition is satisfied in the solution. Fig.

6.25 shows the displacements at the field nodes of the hydrogel, and the expansion for the different values of the ratio ε .

Fig. 6.24. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel for pH = 11, and $E = 25, 29$ and 36 , where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).

It is noted again that the expansion of the hydrogel reduces with an increase in the value of E . Fig. 6.26a shows the influence of the solution pH on the swelling of the hydrogel for the different ratios ε .

Fig. 6.25. The displacements u (a) and v (b), and the original and deformed shapes(c) for a curved boundary hydrogel with $\text{pH} = 11$, and $E = 25, 29$ and 36 .

It is noted that, for a given constant value of ε , swelling of the hydrogel increases with an increase in the pH of solution. The swelling of hydrogel decreases with an increase in the values ε . Fig. 6.26b shows the influence of ε on the swelling of hydrogel, and again it is noted that, the hydrogel swelling reduces with an increase in the value of ε for a

constant pH. The influence of ε on the hydrogel becomes more prominent at the higher values of pH, as observed from Fig. 6.26a.

Fig. 6.26. Influence of the solution pH on the hydrogel swelling at different values of normalized Young's modulus ε (a) and the influence of ε on the hydrogel swelling at different values of the solution pH (b).

Fig. 6.27. Maximum values of the major and minor principle stresses in the hydrogel at the central fully constrained node (a) and total strain energy in the hydrogel domain (b) for the different values of solution pH.

The principle (major and minor) stresses are now computed at each field node of the hydrogel by considering the state of stress at each field node [87], and it is found that the maximum stress exists at the central fully constrained field node.

The maximum values of the principle stresses in the hydrogel with the different values of pH are plotted in Fig. 6.27a. The total strain energy, $U = 0.5 \sigma_{ij} \varepsilon_{ij}$, over the hydrogel domain is computed and plotted in Fig. 6.27b. It is concluded from Fig. 6.27 that the values of strain and stress inside the hydrogel generally increases with an increase in the values of ε . Correspondingly, the strain energy also increases.

Finally, in order to study the influence of the geometrical shape of the hydrogel on the distributions of different ions and the electrical potential, the 2-D simulation of pH-sensitive hydrogel is performed with different geometrical shapes of the hydrogel, as shown in Fig. 6.17b and Fig. 6.17c. At first, the hydrogel with $r = 1:3$, as shown in Fig. 6.17b, is solved and the various profiles of the field variable distributions are plotted in Fig. 6.28. It is seen from Fig. 6.28f that the solution satisfies the electroneutrality condition, and all jumps in the values of the field variables across the interface between the hydrogel and solution are smoothly captured by the RDQ method. Fig. 6.29 shows the displacements and swelling at the field nodes of the hydrogel. The maximum values of the x and y coordinates in the original and deformed configurations from Fig. 6.29c are obtained as 3.0 and 1.0, and 4.96 and 1.68 mm, respectively, and it is seen that their ratios are almost equal, namely $(4.96/3) \approx 1.66$ and $(1.68/1.0) = 1.68$. This indicates that the hydrogel is expanded almost equally in the x and y directions. This is expected as almost the equal magnitudes of the osmotic pressure act on Boundaries 3 and 4, leading to the uniform expansion of the hydrogel.

The hydrogel with a square cross-section, as shown in Fig. 6.17c, is solved, and the profiles of field variables are plotted in Fig. 6.30. It is observed again from Fig. 6.30f that the solution satisfies the electroneutrality condition, and the distributions of all the field variables are smoothly captured across the interface between the hydrogel and solution. Fig. 6.31 shows the displacement values at the field nodes of hydrogel and its expansion. The analytical values of the displacements u and v along the boundaries $x = 2$ and $y = 2$, respectively, are computed by Eqns. (6.33) and (6.36), respectively, and plotted in Fig. 6.31, and it is noted that the analytical and numerical values of the displacements match closely.

Fig. 6.28. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel with rectangular shape for pH = 3 and $E = 29$, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).

Fig. 6.29. The displacements u (a) and v (b), and the original and deformed shapes (c) for a rectangular hydrogel with $\text{pH} = 3$ and $E = 29$.

As the analytical values by Eqns. (6.33) and (6.36) are computed via the numerical values of the osmotic pressure, there is a slight difference between the numerical and analytical displacements in Fig. 6.31 away from the constrained boundaries. This shows that the analytical expressions given by Eqns. (6.33) and (6.36) offer a good judgment about the numerical values of the displacement at the hydrogel field nodes. The maximum values of the x and y coordinates in the original and deformed configurations from Fig. 6.31c are obtained as 2.0 and 2.0, and 3.1353 and 3.1353 mm, respectively, and it is seen that their ratios are exact, namely $(3.1353/2.0) \approx 1.57$. This indicates that the hydrogel is expanded equally in the xy plane.

Fig. 6.30. Schematic of the profiles of various field variables after solving the 2-D pH-sensitive hydrogel with square shape for $\text{pH} = 3$ and $E = 29$, where the different profiles are given as C_f (a), C_1 (b), C_2 (c), C_3 (d), ψ (e) and the distribution of cumulative mobile ions (f).

Fig. 6.31. The displacements u (a) and v (b), and the original and deformed shapes (c) for a square hydrogel domain with $\text{pH} = 3$ and $E = 29$.

The results of the hydrogel deformation by different cross-sections indicate that, when the domain is square ($r = 1:1$), equal magnitude of the osmotic pressure is imposed on all the moving boundaries. As a result, the hydrogel deforms equally in the x and y directions. Therefore, the normal stresses σ_{xx} and σ_{yy} for the square shape of the hydrogel are equal in magnitude, as seen in Fig. 6.32a, and there is a lower shear stress generated in the hydrogel, which results in the lower value of the strain energy, as seen in Fig. 6.32b. For $r = 1:3$ however, the hydrogel deforms unequally in the x and y directions, leading to unequal values of the stresses and higher strain energy, as observed in Fig. 6.32a and Fig. 6.32b.

Fig. 6.32. Comparison between the principle stresses (a) and strain energy (b) for the different geometrical shapes of the hydrogel.

When the hydrogel domain becomes non-uniform, the moving boundaries experience the unequal values of the osmotic pressure, resulting in the unequal deformations, $(3.3408/2.0) = 1.67$ and $(1.74/1.0) = 1.74$, in the x and y directions, respectively, as shown in Fig. 6.21c, and high stresses and strain energy, as shown in Fig. 6.32.

6.6 Summary

In the present work, the working model of 2-D pH-sensitive hydrogel is explained by the PNP and mechanical equilibrium equations. The PNP equations are studied for their continuity along the characteristic line, and are found to be elliptic in nature. As such, 4 virtual nodes nearest to the concerned virtual node in the x and y directions and the concerned virtual node itself are chosen to discretize the governing PDE at the concerned virtual node by the locally applied DQ method. The analytical equations of the displacements at the field nodes located along the hydrogel Neumann boundaries 3 and 4 are then derived by Airy stress function. It is observed that the displacements are 1st-order continuous, if the osmotic pressure values are constant, namely not the functions of x or y coordinates of the field nodes. The discretization of governing equations and the formulation of Jacobian matrix by the RDQ method are further explained in details. A novel approach is suggested to compute the approximate gradient $\partial/\partial y$ at the virtual

node of the hydrogel, where there is no any virtual node in the y direction, while exactly imposing the Neumann boundary condition.

Next, the simulations of the 2-D pH-sensitive hydrogel are performed by the RDQ method. The results are discussed according to the nature of the problem. The effects of the solution pH and initial fixed-charge concentration on the swelling of the hydrogel are studied with the pH values of 3, 7 and 11, and the initial fixed-charge concentration values of 0.5, 1.05 and 1.5 mM. It is seen from all the simulation results that the RDQ method is well capable of capturing the moving boundary interfaces, and the jump of the field variable distributions over the interface between the multi-domain boundaries. It is also seen that the RDQ method is capable of equally handling the uniform and random distributions of the field nodes in the computational domains. It is observed from Fig. 6.31 that the displacements u and v , computed by the analytical expressions given in Equations (6.33) and (6.36) respectively, match closely with the corresponding numerical values for the field nodes located along Boundaries 3 and 4. As a result, Eqns. (6.33) and (6.36) closely predict the displacements at the field nodes located along the Neumann boundaries 3 and 4 of the hydrogel.

Further, the 2-D hydrogel model with an irregular domain is studied by changing the values of the Young's moduli as 25, 29 and 36 and the pH of solution as 3, 7 and 11. It is seen from the simulation results that the electroneutrality is always maintained in the solution for all the case studies, and the jumps in the distributions of the field variables are smoothly captured across the interface between the hydrogel and solution. It is also noted from the results that the hydrogel swelling increases for a constant value of E with an increase in the value of solution pH. For a constant pH however, the hydrogel swelling reduces with an increase in the value of E . This result is qualitatively in good agreement with the other simulations results as well as the experiments. The different stresses across the hydrogel domain are computed. The principle stresses and planes are determined by Mohr's circle [87] by considering the state of stress corresponding to one field node at a time. It is observed that the maximum values of the principle stresses and the strain energy in the hydrogel domain generally increases with an increase in the value of E .

Finally, three different geometrical shapes of the hydrogel disc are considered to study their effect on the deformation. It is observed from the results that the hydrogel

expands equally in the x and y directions for the uniform geometrical shape, resulting in the lower values of the stress and strain energy. However, the hydrogel expands unequally for the non-uniform geometrical shape, leading to the higher values of the stress and strain energy.

The present main objective is to demonstrate the ability of the RDQ method in solving the complex multiphysics problems, such as the 2-D simulation of the pH-sensitive hydrogel. Therefore, the geometries of the hydrogel and the input data are chosen and they are challenging to be solved by the RDQ method. The simulation results are discussed qualitatively, and verified as per the nature of the problem. Therefore, it can be fairly concluded that the present analyses and results given in this chapter are new, and they could only be verified quantitatively by the further research work. It is finally concluded that the RDQ method is capable of effectively handling the complex problems, such as the 2-D simulation of pH-responsive hydrogels involving nonlinear governing equations, moving boundaries, and the multiple computational domains with the jump in the field variable distributions.

Chapter 7

Conclusions and the recommendations for future work

In this chapter, several conclusions are made from the present research work, After that, numerous recommendations are made for the possible future research work, which may be carried out in the area of developing novel meshless methods in general, specifically in the RDQ method as well.

7.1 Conclusions from the present research

A novel meshless method called the RDQ method is formulated in the present work. The convergence, consistency, stability, and adaptive analyses of the RDQ method are performed to ensure the method to be convergent, consistent, and stable while solving the complex static and dynamic problems.

7.1.1 *Convergence analysis of the RDQ method*

The convergence analysis of the RDQ method is performed in details in Chapter 3 by solving several 1-D, 2-D, and elasticity problems. When the monomials up to the complete order p from Pascal's triangle are included in the approximation of function, the order of the convergence rate by Taylor series expansion is given as $O(h^{p+1})$. Several 1-D test problems are solved by including up to the 2nd-order monomials in the polynomial basis of function approximation. The convergence rate close to 2 is achieved for the function, and the convergence rates close to 1 are achieved for the 1st- and 2nd-order derivatives with the uniform as well as random field nodes, as given in Table 3.2. For the 2-D Laplace problem with mixed boundary conditions and up to the 2nd-order of function approximation, the convergence rates equal to 2.25 and 3.75 are achieved for the function by the uniform and random field nodes, respectively. The convergence rates for the 1st- and 2nd-order derivatives are found to be equal to 1.2 and 0.8, respectively, by both the uniform as well as random field nodes as shown in Fig 3.9. A cantilever beam under a pure bending load is solved by including up to the 1st-order monomials in the

polynomial basis of function approximation. The convergence rates for the displacements u and v , by discretizing the domain with the uniform and random field nodes separately, are found to be equal to 1.0 and 1.3, and 1.96 and 2.0, respectively, as shown in Fig. 3.12b. A cantilever beam under a pure shear load is also solved by including up to the 2nd-order monomials in the polynomial basis of function approximation. The convergence rates for the displacements u and v , by discretizing the domain with the uniform and random field nodes separately, are equal to 1.94 and 1.9, and 1.5 and 2.0, respectively, as shown in Fig. 3.14b. Therefore, it is concluded that the RDQ method gives good convergence rates of the function and its derivatives. By implementing the novel approaches named the weighted derivative and improved weighted derivative, the convergence results demonstrate that the novel approaches can provide satisfactory rates of the derivative convergence for the field nodes distributed in either the uniform or random manner, since the above convergence rates of the derivatives are obtained by the weighted derivative technique and the improved weighted derivative approach. The problem of a circular plate with a central hole involving the irregular domain also demonstrates that the RDQ method can be employed effectively for the irregular boundary problems. When the test problems are solved by combining the uniform and random field nodes separately with the uniform and cosine virtual nodes, it is observed by comparison between Tables 3.4 and 3.5, Tables 3.6 and 3.7, Tables 3.8 and 3.9, and Tables 3.10 and 3.11 that the best rate of the function convergence is obtained by combining the random field nodes with the uniform virtual nodes, and the best rates of the derivative convergence by combining the uniform field nodes with the uniform virtual nodes. Therefore, the main objective of developing the RDQ method, to employ the DQ method with the random distribution of field nodes, is achieved. A superconvergence condition is developed for the RDQ method, which gives more than $O(h^{p+1})$ convergence rate of the function for the uniform as well as random field nodes, where p is the highest order of the monomials used in the approximation of function. Several 1-D and 2-D test problems are solved by the superconvergence condition, and it is observed from Table 3.2, and Figs. 3.6, 3.10 and 3.11 that indeed the convergence rates have been improved. When the 1-D locally high gradient problem is solved by including up to the

2nd-order monomials, the convergence rate of the function is given by $O(h^{p+2.1})$, and all the convergence rates of the derivative are also superconvergent as shown in Fig. 3.6.

7.1.2 Consistency analysis of the RDQ method

The detailed consistency analysis of the RDQ method is performed with the DQ weighting coefficients computed by Shu's general approach [36-37] and Quan and Chang approach [34-35], and it is concluded that the RDQ method gives the consistent discretization of governing differential equation, when the DQ weighting coefficients are computed by Shu's general approach [36-37]. It is also shown that the consistency equation obtained by discretizing the governing equation with the cosine distribution of virtual nodes converges to that obtained by the uniform distribution of virtual nodes for the sufficiently large number of virtual nodes.

7.1.3 Stability analysis of the RDQ method

The stability analysis of the RDQ method is performed by the equations of 1st-order wave, transient heat conduction, and transverse beam deflection. It is shown that the time dependent problems can be solved successfully by the RDQ method via their systematic stability analysis using Neumann and Schur polynomials [92, 103, 191]. Therefore, it can be effectively employed while solving the problems with the terms of the time derivative.

7.1.4 Adaptive analysis of the RDQ method

The ARDQ method is developed to solve the locally high gradient problems by implementing the adaptive algorithm of h -refinement with the RDQ method. A novel approach with the cross product of vectors in the ARDQ method can ensure the newly created field nodes to be always within the convex hull of the computational domain, irrespective of the regular or irregular boundaries of the domain. An error recovery technique is proposed, which ensures the error in the solution to be always minimized, irrespective of the number of field nodes used in the local interpolation domain of the field variables. It is concluded by solving several test problems that the ARDQ method

coupled with the approaches with the cross product of vectors and the error recovery technique results in a robust adaptive meshless method.

By comparing the values of the global error norm obtained by the ARDQ method with those by the RDQ method through the uniform increment in the field nodes, it is seen that the ARDQ method achieves much lower values of the global error norm for the same number of the field nodes in the domain, as given in Table 5.7. Comparing the convergence rates of the several test problems that are solved with the RDQ and ARDQ methods in Chapters 3 and 5, respectively, it is noted that the ARDQ method achieves the equal or better convergence rates at relatively lower number of the field nodes. This is an important result since it highlights one of the objectives of the development of the ARDQ method. The 2-D problem of steady-state heat conduction is also solved by the RDQ and ARDQ methods, the temperature distribution along the line $(0.5, y)$ is shown in Fig. 5.14a, and the global error norms are plotted in Fig. 5.14b. It is noted from Fig. 5.14a that the numerical values of temperature computed by the ARDQ method are more accurate than those by the RDQ method. The problem of a circular plate with a central hole is also solved by the RDQ and ARDQ methods with the convergence curves plotted in Fig. 5.32a, and it is seen that the better rates of the convergence are obtained by the ARDQ method.

7.1.5 Study of the pull-in instability in the fixed-fixed and cantilever microswitches under the nonlinear electrostatic load

Two configurations of the MEMS microswitches, based on the fixed-fixed and cantilever beams, are simulated by the RDQ method for the study of pull-in instability due to the applied voltage. The simulation results are presented for the UDL and nonlinear electrostatic load. It is shown that the pull-in voltage equal to $V = 15.05$ volt determined by the RDQ method for the fixed-fixed microswitch closely matches with the reported value equal to 15.07 volt in open literature [90-91], and the pull-in voltage equal to $V = 2.5286$ volt determined by the RDQ method for the cantilever microswitch closely matches with the reported value equal to 2.33 volt in open literature [90-91]. This demonstrates the ability of the RDQ method in handling the nonlinear force field.

7.1.6 2-D simulation of pH-sensitive hydrogel by the RDQ method

The 2-D simulation of the pH-sensitive hydrogel is performed to demonstrate the applicability of the RDQ method for solving the complex multiphysics problems involving the coupled nonlinear PDE. The simulation results show that the RDQ method smoothly captures the jump in the distribution of the field variables across the interface between the boundaries of the solution and hydrogel. The analysis and discussion of the simulation results show that they are consistent with the nature of the problem. After the multiphysics 2-D simulation of the pH-sensitive hydrogel, the deformation of the hydrogel is successfully studied in details by changing the values of the initial fixed-charge concentration, pH of solution, and Young's modulus of the hydrogel. Three different geometries of the hydrogel discs are also studied for the deformation by the various values of the solution pH and Young's moduli. The geometries of the hydrogel and the input data are chosen and they are challenging to be solved by the RDQ method. All the simulation results are discussed qualitatively and they match well with the nature of the problem.

7.2 Recommendations for the possible future work

7.2.1 Criterion to decide the number of total nodes in the local domain of interpolation

When the convergence analysis of the RDQ method is performed, it is noted that the accuracy of the numerical solution fluctuates with the change in the number of nodes used in the local domain of the fixed RKPM interpolation function. The fluctuations in the numerical solution become small if the distribution of the field variable is uniform, but they are high for the locally high gradient distribution of the field variable. As a result, it becomes difficult to decide the number of nodes used in the local domain of the interpolation. Therefore, a further research is essential to identify a criterion or condition, which may accurately predict the number of nodes to be used in the local domain of the interpolation.

7.2.2 Development of a priori or posteriori error estimators for the meshless methods

It is difficult to develop *a priori* or *posteriori* error estimators for the meshless methods, as the spacing between the nodes is not constant. Therefore, this provides an avenue for further research work in the development of good error estimators, which can be coupled with the meshless technique to give a robust adaptive method.

7.2.3 2-D Transient simulation of pH-sensitive hydrogel

The 2-D simulation of pH-sensitive hydrogel is performed here for an equilibrium case. However, it will be interesting and challenging to solve the transient case by the RDQ method, since the 2-D transient analysis of pH-sensitive hydrogel has not been done so far by the meshless method. It will also be interesting to solve the 2-D simulation of the pH-sensitive hydrogel by the standard FEM package, such as ANSYS, and comparing the FEM results with those obtained by the RDQ method.

7.2.4 Incorporation of the large deformation theory in the 2-D simulation of pH sensitive hydrogels

The 2-D simulation of pH-sensitive hydrogel is performed by the linear deformation theory in the present work, as the main objective here is to test the effectiveness of the RDQ method in solving the complex problems from engineering. The conversion of the 2-D PNP equations in the form of Lagrangian coordinates is quite difficult to be simplified and solved. Therefore, it will be interesting to convert all the 2-D PNP and mechanical equilibrium equations in the form of Lagrangian coordinates and then solve by the RDQ method.

7.2.5 Application of the RDQ method in the fluid-structure interaction problems

Several difficult problems exist in the area of fluid-structure interaction (FSI). These problems are particularly challenging to be solved due to their peculiar nature. The fluid domain is generally solved by the Eulerian coordinate system, while the structural

domain by the Lagrangian coordinates system. In the FSI therefore, it is quite difficult to accurately exchange the values of the field variables across the interface between the fluid and structure domains. As such, the numerical method used in the FSI should facilitate this exchange. It may be relatively easy to solve the FSI problems by the strong-form methods. Therefore, it will be interesting to apply the RDQ method in the FSI problems, since the RDQ method may easily exchange the data between the fluid and structure domains, due to the use of fixed RKPM interpolation for the approximation of function.

7.2.6 Development of a novel numerical strong-form meshless method

A novel strong-form meshless method is proposed here with a novel interpolation function. In general, a strong-form meshless method contains three important steps, viz. (1) a local domain creation around the field nodes for the approximation of function, (2) an interpolation function to approximate the value of function at every field node by the local domain of interpolation around the concerned field node, and (3) a technique of derivative approximation to discretize the derivate terms from the governing differential equation.

In the proposed meshless method, a local domain is created around each field node by the concept of voronoi diagrams and voronoi cells [7, 27-28], and the function value at every field node is approximated by a novel interpolation function, similar to the iso-parametric elements used in the classical FEM. The governing equation is then discretized by the DQ method applied over the local DQ domain. The major steps in the proposed method are explained below.

7.2.6.1 Creation of a local domain

The approximation of function in a meshless method directly affects the computational accuracy of the solution. Therefore, it is important to create a local domain for the function value approximation, which captures the localized variation of function well.

In the proposed meshless method, a local domain is created around each field node by the voronoi diagrams [7], which are used for the geometric surface tessellation, based on

the randomly distributed field nodes. The voronoi diagram is created first by the voronoi cells, based on the concept of natural neighbour [7, 27-28]. The natural neighbour nodes around each field node are then determined by the voronoi cells. These natural neighbour nodes are used for the approximation of function at the concerned field node.

7.2.6.2 Approximation of a function

The interpolation function, called the iso-parametric interpolation, is proposed to approximate the value of function at every field node, based on the concept of the iso-parametric elements in the classical FEM. Therefore, the value of the function and the nodal coordinates at the concerned field node are approximated by the same vector of weighting coefficients, as per the iso-parametric interpolation. The preliminary formulation of the interpolation is given below.

Let us consider two types of nodes, namely the virtual and field nodes, such that the governing equation is discretized at the virtual nodes, and the distribution of field variable is captured by the field nodes. Let the i^{th} virtual node (x_i, y_i) be approximated by n field nodes, which are surrounding the i^{th} virtual node. As such, $n \subseteq N$ where N is the number of total field nodes in the whole domain. The coordinates (x_i, y_i) are approximated as

$$\begin{Bmatrix} x_i \\ y_i \end{Bmatrix}_{2 \times 1} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix}_{2 \times n} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{Bmatrix}_{n \times 1} \Rightarrow [X_i]_{2 \times 1} = [X]_{2 \times n} \{\alpha\}_{n \times 1} \quad (7.1)$$

where $\{\alpha\}$ is the vector of unknown coefficients. Multiplying both sides of Eq. (7.1) by $[X]^T$ results in

$$[X]_{n \times 2}^T [X_i]_{2 \times 1} = [X]_{n \times 2}^T [X]_{2 \times n} \{\alpha\}_{n \times 1} \Rightarrow [X_V]_{n \times 1} = [X_G]_{n \times n} \{\alpha\}_{n \times 1} \quad (7.2)$$

The vector $\{\alpha\}$ is obtained as

$$\{\alpha\}_{n \times 1} = [X_G]_{n \times n}^{-1} [X_V]_{n \times 1} \quad (7.3)$$

As a result, the approximate function $f^h(x_i, y_i)$ at the i^{th} node is computed by

$$f^h(x_i, y_i) = \sum_{j=1}^n \alpha_j u_j \Rightarrow [\alpha]_{1 \times n} \{U\}_{n \times 1} \quad (7.4)$$

The main issue of the existence of $[X_G]^{-1}$ arises in Eq. (7.3). The matrix $[X]_{2 \times n}$ has 2 rows and $n > 2$, such that the highest rank of the matrix $[X]$ becomes 2. As a result, the matrix $[X_G]$ becomes non-invertible due to the rank deficiency. This problem can be overcome by assembling all the local approximations of the virtual node coordinates with Eq. (7.1) to get the global matrix of approximation as

$$\begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_m \\ y_m \end{Bmatrix}_{2m \times 1} = [X]_{2m \times N} \{\alpha\}_{N \times 1} \quad (7.5)$$

where m is the number of total virtual nodes in the domain such that $m \geq N$. The Dirichlet boundary conditions are then imposed in Eq. (7.5). Now it is seen from Eq. (7.5) that the matrix $[X]$ is invertible as $2m \geq N$. Eq. (7.4) is then used to compute the value of approximate function at the i^{th} virtual node.

The properties of the iso-parametric interpolation function are not investigated yet, which should be done to gain more insight into it. If the interpolation function does not possess the delta function property, the value of nodal parameter is not the same as the value of function approximation, namely $u_i \neq f^h(x_i, y_i)$. As a result, after solving the governing PDE for the unknown nodal parameters at the field nodes, the values of the approximate function at all the field nodes are computed by interpolating each field node via the nearby field nodes with the voronoi diagrams, as explained earlier.

7.2.6.3 Discretization of a governing equation

The background nodes called the virtual nodes are created in a domain, based on the specific pattern, such as uniform or cosine. A local DQ domain is created around every virtual node, in which all the virtual nodes falling are considered for the discretization of

the governing equation at the concerned virtual node. The governing equation is then discretized by the DQ method applied over the local DQ domain about the concerned virtual node. The value of function at every virtual node is approximated by the surrounding field nodes with the concepts of natural neighbor and iso-parametric interpolation, as explained earlier. The terms corresponding to the approximation of function value at virtual node in the discretized equation are replaced by the corresponding expression of iso-parametric interpolation. The simplification of the discretized governing equation results in a simple algebraic expression. Similar expressions are obtained at all the virtual nodes, and assembled together to form a global stiffness matrix $AX = B$. This global matrix can be solved by direct or iterative methods. This is the broad concept of the proposed meshless method.

7.3 Summary

In summary, several important conclusions are made from the convergence, consistency, stability and the adaptive analyses of the RDQ method. The studies of the microswitches for the pull-in instability and the 2-D simulation of pH-sensitive hydrogel highlight the applicability of the RDQ method in handling the complex problems with the nonlinear force fields as well as the nonlinear governing equations. Numerous suggestions are made, including a novel concept of the meshless method, for the possible future research work.

REFERENCES

- [1] Y. Krongauz, T. Belytschko, Enforcement of essential boundary conditions in meshless approximations using finite elements, *Computer Methods in Applied Mechanics and Engineering*. 131 (1996) 133-145.
- [2] L.B. Lucy, A numerical approach to testing the fission hypothesis, *Astronomical Journal*. 82 (1977) 1013-1024.
- [3] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics - Theory and application to non-spherical stars, *Monthly Notices of the Royal Astronomical Society*. 181 (1977) 375-389.
- [4] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: Diffuse approximation and diffuse elements, *Computational Mechanics*. 10 (1992) 307-318.
- [5] T. Belytschko, L. Gu, Y.Y. Lu, Fracture and crack growth by element free Galerkin methods, *Modelling and Simulation in Materials Science and Engineering*. 2 (1994) 519-534.
- [6] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods, *International Journal for Numerical Methods in Engineering*. 37 (1994) 229 - 256.
- [7] J. Braun, M. Sambridge, A numerical method for solving partial differential equations on highly irregular evolving grids, *Nature*. 376 (1995) 655-660.
- [8] W.K. Liu, Y. Chen, R.A. Uras, C.T. Chang, Generalized multiple scale reproducing kernel particle methods, *Computer Methods in Applied Mechanics and Engineering*. 139 (1996) 91-157.
- [9] W.K. Liu, S. Jun, Multiple-scale reproducing kernel particle methods for large deformation problems, *International Journal for Numerical Methods in Engineering*. 41 (1998) 1339-1362.
- [10] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, *International Journal for Numerical Methods in Fluids*. 20 (1995) 1081 - 1106.
- [11] J.M. Melenk, I. Babuška, The partition of unity finite element method: Basic theory and applications, *Computer Methods in Applied Mechanics and Engineering*. 139 (1996) 289-314.

- [12] S.N. Atluri, T. Zhu, A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics, *Computational Mechanics*. 22 (1998) 117-127.
- [13] S.N. Atluri, J. Sladek, V. Sladek, T. Zhu, Local boundary integral equation (LBIE) and it's meshless implementation for linear elasticity, *Computational Mechanics*. 25 (2000) 180-198.
- [14] T. Zhu, J.D. Zhang, S.N. Atluri, A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach, *Computational Mechanics*. 21 (1998) 223-235.
- [15] G.R. Liu, Y.T. Gu, A point interpolation method for two-dimensional solids, *International Journal for Numerical Methods in Engineering*. 50 (2001) 937-951.
- [16] H. Li, Q.X. Wang, K.Y. Lam, Development of a novel meshless Local Kriging (LoKriging) method for structural dynamic analysis, *Computer Methods in Applied Mechanics and Engineering*. 193 (2004) 2599-2619.
- [17] E. Oñate, S. Idelsohn, O.C. Zienkiewicz, 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 (1996) 3839-3866.
- [18] E. Oñate, S. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, C. Sacco, A stabilized finite point method for analysis of fluid mechanics problems, *Computer Methods in Applied Mechanics and Engineering*. 139 (1996) 315-346.
- [19] H. Li, T.Y. Ng, J.Q. Cheng, K.Y. Lam, Hermite-Cloud: A novel true meshless method, *Computational Mechanics*. 33 (2003) 30-41.
- [20] G.R. Liu, J. Zhang, K.Y. Lam, H. Li, G. Xu, Z.H. Zhong, G.Y. Li, X. Han, A gradient smoothing method (GSM) with directional correction for solid mechanics problems, *Computational Mechanics*. 41 (2008) 457-472.
- [21] S.N. Atluri, J.Y. Cho, H.G. Kim, Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations, *Computational Mechanics*. 24 (1999) 334-347.
- [22] S.N. Atluri, H.G. Kim, J.Y. Cho, Critical assessment of the truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods, *Computational Mechanics*. 24 (1999) 348-372.

- [23] D.F. Gilhooley, J.R. Xiao, R.C. Batra, M.A. McCarthy, J.W. Gillespie Jr, Two-dimensional stress analysis of functionally graded solids using the MLPG method with radial basis functions, *Computational Materials Science*. 41 (2008) 467-481.
- [24] S.R. Idelsohn, E. Onate, N. Calvo, F. Del Pin, The meshless finite element method, *International Journal for Numerical Methods in Engineering*. 58 (2003) 893-912.
- [25] H. Li, Y.K. Yew, T.Y. Ng, K.Y. Lam, Meshless steady-state analysis of chemo-electro-mechanical coupling behavior of pH-sensitive hydrogel in buffered solution, *Journal of Electroanalytical Chemistry*. 580 (2005) 161-172.
- [26] G.R. Liu, J. Zhang, H. Li, K.Y. Lam, B.B.T. Kee, Radial point interpolation based finite difference method for mechanics problems, *International Journal for Numerical Methods in Engineering*. 68 (2006) 728-754.
- [27] N. Sukumar, B. Moran, T. Belytschko, The natural element method in solid mechanics, *International Journal for Numerical Methods in Engineering*. 43 (1998) 839-887.
- [28] N. Sukumar, B. Moran, A.Y. Semenov, V.V. Belikov, Natural neighbour Galerkin methods, *International Journal for Numerical Methods in Engineering*. 50 (2001) 1-27.
- [29] J. Bonet, T.S.L. Lok, Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations, *Computer Methods in Applied Mechanics and Engineering*. 180 (1999) 97-115.
- [30] S. Wong, Y. Shie, Large deformation analysis with galerkin based smoothed particle hydrodynamics, *CMES - Computer Modeling in Engineering and Sciences*. 36 (2008) 97-117.
- [31] C.P. Wu, K.H. Chiu, Y.M. Wang, A differential reproducing kernel particle method for the analysis of multilayered elastic and piezoelectric plates, *CMES - Computer Modeling in Engineering and Sciences*. 27 (2008) 163-186.
- [32] L. Li, S. Liu, H. Wang, Meshless analysis of ductile failure, *CMES - Computer Modeling in Engineering and Sciences*. 36 (2008) 173-191.

- [33] R. Bellman, B.G. Kashef, J. Casti, Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations, *Journal of Computational Physics*. 10 (1972) 40-52.
- [34] J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature method-II. Numerical experiments, *Computers and Chemical Engineering*. 13 (1989) 1017-1024.
- [35] J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature method-I. Analysis, *Computers and Chemical Engineering*. 13 (1989) 779-788.
- [36] C. Shu, B.C. Khoo, K.S. Yeo, Numerical solutions of incompressible Navier-Stokes equations by generalized differential quadrature, *Finite Elements in Analysis and Design*. 18 (1994) 83-97.
- [37] C. Shu, *Differential Quadrature and its application in Engineering*, Springer-Verlag, London, 2000.
- [38] Y.Y. Shan, C. Shu, Z.L. Lu, Application of local MQ-DQ method to solve 3D incompressible viscous flows with curved boundary, *CMES - Computer Modeling in Engineering and Sciences*. 25 (2008) 99-113.
- [39] K.M. Liew, J.Z. Zhang, T.Y. Ng, S.A. Meguid, Three-dimensional modelling of elastic bonding in composite laminates using layerwise differential quadrature, *International Journal of Solids and Structures*. 40 (2003) 1745-1764.
- [40] K.M. Liew, Y.Q. Huang, J.N. Reddy, Moving least squares differential quadrature method and its application to the analysis of shear deformable plates, *International Journal for Numerical Methods in Engineering*. 56 (2003) 2331-2351.
- [41] K.M. Liew, Y.Q. Huang, J.N. Reddy, Analysis of general shaped thin plates by the moving least-squares differential quadrature method, *Finite Elements in Analysis and Design*. 40 (2004) 1453-1474.
- [42] G. Naadimuthu, R. Bellman, K.M. Wang, E.S. Lee, Differential quadrature and partial differential equations: Some numerical results, *Journal of Mathematical Analysis and Applications*. 98 (1984) 220-235.
- [43] H. Ding, C. Shu, K.S. Yeo, D. Xu, Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form by local multiquadric

- differential quadrature method, *Computer Methods in Applied Mechanics and Engineering*. 195 (2006) 516-533.
- [44] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, *Computer Methods in Applied Mechanics and Engineering*. 139 (1996) 3-47.
 - [45] T. Belytschko, Y.Y. Lu, L. Gu, M. Tabbara, Element-free galerkin methods for static and dynamic fracture, *International Journal of Solids and Structures*. 32 (1995) 2547-2570.
 - [46] T. Belytschko, Y.Y. Lu, L. Gu, Crack propagation by element-free Galerkin methods, *Engineering Fracture Mechanics*. 51 (1995) 295-315.
 - [47] N.R. Aluru, G. Li, Finite cloud method: A true meshless technique based on fixed reproducing kernel approximation, *International Journal for Numerical Methods in Engineering*. 50 (2001) 2373-2410.
 - [48] N.R. Aluru, A point collocation method based on reproducing kernel approximations, *International Journal for Numerical Methods in Engineering*. 47 (2000) 1083-1121.
 - [49] I. Babuška, J.M. Melenk, The partition of unity method, *International Journal for Numerical Methods in Engineering*. 40 (1997) 727-758.
 - [50] S.N. Atluri, The meshless method (MLPG) for domain & BIE discretizations, Tech Science Press, Forsyth, 2004.
 - [51] S.N. Atluri, S. Shen, The meshless local Petrov-Galerkin method, Tech Science Press, Encino, 2002.
 - [52] S.N. Atluri, Methods of computer modeling in engineering & the sciences, Tech Science Press, Forsyth, 2005.
 - [53] S.N. Atluri, The meshless method (MLPG) for domain & BIE discretizations, Tech Science Press, Forsyth, 2004.
 - [54] S.N. Atluri, H.T. Liu, Z.D. Han, Meshless Local Petrov-Galerkin (MLPG) mixed finite difference method for solid mechanics, *CMES - Computer Modeling in Engineering and Sciences*. 15 (2006) 1-16.

- [55] S.N. Atluri, Z.D. Han, A.M. Rajendran, A new implementation of the meshless finite volume method, through the MLPG "mixed" approach, CMES - Computer Modeling in Engineering and Sciences. 6 (2004) 491-513.
- [56] S.N. Atluri, H.T. Liu, Z.D. Han, Meshless Local Petrov-Galerkin (MLPG) mixed collocation method for elasticity problems, CMES - Computer Modeling in Engineering and Sciences. 14 (2006) 141-152.
- [57] Z.D. Han, A.M. Rajendran, S.N. Atluri, Meshless Local Petrov-Galerkin (MLPG) Approaches for Solving Nonlinear Problems with Large Deformations and Rotations, CMES: Computer Modeling in Engineering and Sciences. 10 (2005) 1-12.
- [58] Z.D. Han, H.T. Liu, A.M. Rajendran, S.N. Atluri, The applications of Meshless Local Petrov-Galerkin (MLPG) approaches in high-speed impact, penetration and perforation problems, CMES - Computer Modeling in Engineering and Sciences. 14 (2006) 119-128.
- [59] X. Wu, S. Shen, W. Tao, Meshless local Petrov-Galerkin collocation method for two-dimensional heat conduction problems, CMES - Computer Modeling in Engineering and Sciences. 22 (2007) 65-76.
- [60] J. Sladek, V. Sladek, C. Zhang, P. Sulek, L. Starek, Fracture analyses in continuously nonhomogeneous piezoelectric solids by the MLPG, CMES - Computer Modeling in Engineering and Sciences. 19 (2007) 247-262.
- [61] J. Sladek, V. Sladek, C. Zhang, P. Sulek, Application of the MLPG to thermo-piezoelectricity, CMES - Computer Modeling in Engineering and Sciences. 22 (2007) 217-233.
- [62] T. Jarak, J. Sorić, J. Hoster, Analysis of shell deformation responses by the Meshless Local Petrov-Galerkin (MLPG) approach, CMES - Computer Modeling in Engineering and Sciences. 18 (2007) 235-246.
- [63] Y.C. Cai, H.H. Zhu, A local meshless shepard and least square interpolation method based on local weak form, CMES - Computer Modeling in Engineering and Sciences. 34 (2008) 179-204.

- [64] Y.H. Liu, S.S. Chen, J. Li, Z.Z. Cen, A meshless local natural neighbour interpolation method applied to structural dynamic analysis, CMES - Computer Modeling in Engineering and Sciences. 31 (2008) 145-156.
- [65] A. Arefmanesh, M. Najafi, H. Abdi, Meshless local Petrov-Galerkin method with unity test function for non-isothermal fluid flow, CMES - Computer Modeling in Engineering and Sciences. 25 (2008) 9-22.
- [66] S.S. Chen, Y.H. Liu, Z.Z. Cen, A combined approach of the MLPG method and nonlinear programming for lower-bound limit analysis, CMES - Computer Modeling in Engineering and Sciences. 28 (2008) 39-55.
- [67] T.D. Dane, B.V. Sankar, Meshless local Petrov-Galerkin micromechanical analysis of periodic composites including shear loadings, CMES - Computer Modeling in Engineering and Sciences. 26 (2008) 169-187.
- [68] S. Li, S.N. Atluri, Topology-optimization of structures based on the MLPG mixed collocation method, CMES - Computer Modeling in Engineering and Sciences. 26 (2008) 61-74.
- [69] J. Zheng, S. Long, Y. Xiong, G. Li, A finite volume meshless local petrov-galerkin method for topology optimization design of the continuum structures, CMES - Computer Modeling in Engineering and Sciences. 42 (2009) 19-34.
- [70] S.Y. Long, K.Y. Liu, G.Y. Li, An analysis for the elasto-plastic fracture problem by the meshless local Petrov-Galerkin method, CMES - Computer Modeling in Engineering and Sciences. 28 (2008) 203-216.
- [71] J. Sladek, V. Sladek, C.L. Tan, S.N. Atluri, Analysis of transient heat conduction in 3D anisotropic functionally graded solids, by the MLPG method, CMES - Computer Modeling in Engineering and Sciences. 32 (2008) 161-174.
- [72] J. Sladek, V. Sladek, P. Sulek, S.N. Atluri, Modeling of intelligent material systems by the MLPG, CMES - Computer Modeling in Engineering and Sciences. 34 (2008) 273-300.
- [73] J. Sladek, V. Sladek, P. Sulek, P.H. Wen, Thermal bending of Reissner-Mindlin plates by the MLPG, CMES - Computer Modeling in Engineering and Sciences. 28 (2008) 57-76.

- [74] G. Pini, A. Mazzia, F. Sartoretto, Accurate MLPG solution of 3D potential problems, CMES - Computer Modeling in Engineering and Sciences. 36 (2008) 43-63.
- [75] G.R. Liu, Mesh free methods: moving beyond the finite element method, CRC Press 2003.
- [76] G. Dahlquist, A. Björck. 4.3.4. Equidistant Interpolation and the Runge Phenomenon In: Anderson N (ed) Numerical Methods, Dover Publications, New York, pp 101-103
- [77] D.G. Krige. A review of the development of geostatistics in South Africa In: Guarascio M, David M, Huijbregts C (eds) Advanced Geostatistics in the Mining Industry D. Reidel Publishing Company, Holland
- [78] H. Li, J.Q. Cheng, T.Y. Ng, J. Chen, K.Y. Lam, A meshless Hermite-Cloud method for nonlinear fluid-structure analysis of near-bed submarine pipelines under current, Engineering Structures. 26 (2004) 531-542.
- [79] X. Jin, G. Li, N.R. Aluru, On the equivalence between least-squares and kernel approximations in meshless methods, CMES - Computer Modeling in Engineering and Sciences. 2 (2001) 447-462.
- [80] S.S. Mulay, H. Li, S. See, On the random differential quadrature (RDQ) method: Consistency analysis and application in elasticity problems, Computational Mechanics. 44 (2009) 563-590.
- [81] Y.X. Mukherjee, S. Mukherjee, On boundary conditions in the element-free Galerkin method, Computational Mechanics. 19 (1997) 264-270.
- [82] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, The finite element method: its basis and fundamentals, Elsevier Butterworth-Heinemann, 2005.
- [83] G.R. Liu, S.S. Quek, The finite element method: A practical course, Butterworth-Heinemann, 2003.
- [84] U. Christoph, Numerical computation: methods, software, and analysis, Springer-Verlag, Germany, 1997.
- [85] S.A. Sarra, Chebyshev Interpolation: An Interactive Tour, Journal of Online Mathematics and Its Applications. 6 (2006) 1-13.

- [86] E. Scheinerman, C++ for Mathematicians: An Introduction for Students and Professionals, CRC Press, USA, 2006.
- [87] S.P. Timoshenko, J.N. Goodier, Theory of Elasticity, McGraw-Hill, New York, 1970.
- [88] Z. Xu, Applied Elasticity (Eng), New Age International (p) Ltd, India, 1992.
- [89] E.P. Popov, Engineering Mechanics of Solids, Prentice Hall, New Jersey, 1990.
- [90] Q.X. Wang, H. Li, K.Y. Lam, Analysis of microelectromechanical systems (MEMS) devices by the meshless point weighted least-squares method, Computational Mechanics. 40 (2007) 1-11.
- [91] H. Li, Q.X. Wang, K.Y. Lam, A variation of local point interpolation method (vLPIM) for analysis of microelectromechanical systems (MEMS) device, Engineering Analysis with Boundary Elements. 28 (2004) 1261-1270.
- [92] J. Strikwerda, Finite difference schemes and partial differential equations, Wadsworth and Brooks, California, 1989.
- [93] D.S. Balsara, Von Neumann stability analysis of Smoothed Particle Hydrodynamics - Suggestions for optimal algorithms, Journal of Computational Physics. 121 (1995) 357-372.
- [94] S. Børve, M. Omang, J. Trulsen, Two-dimensional MHD smoothed particle hydrodynamics stability analysis, Astrophysical Journal, Supplement Series. 153 (2004) 447-462.
- [95] J.W. Swegle, D.L. Hicks, S.W. Attaway, Smoothed particle hydrodynamics stability analysis, Journal of Computational Physics. 116 (1995) 123-134.
- [96] L.D.G. Sigalotti, H. López, Adaptive kernel estimation and SPH tensile instability, Computers and Mathematics with Applications. 55 (2008) 23-50.
- [97] R. Di Lisio, E. Grenier, M. Pulvirenti, The convergence of the SPH method, Computers and Mathematics with Applications. 35 (1998) 95-102.
- [98] T. Belytschko, Y. Guo, W.K. Liu, S.P. Xiao, A unified stability analysis of meshless particle methods, International Journal for Numerical Methods in Engineering. 48 (2000) 1359-1400.

- [99] T.C. Fung, Stability and accuracy of differential quadrature method in solving dynamic problems, *Computer Methods in Applied Mechanics and Engineering*. 191 (2002) 1311-1331.
- [100] S. Tomasiello, Stability and accuracy of the iterative differential quadrature method, *International Journal for Numerical Methods in Engineering*. 58 (2003) 1277-1296.
- [101] L. Aceto, D. Trigiante, The stability problem for linear multistep methods: Old and new results, *Journal of Computational and Applied Mathematics*. 210 (2007) 2-12.
- [102] R. Ata, A. Soulaïmani, A stabilized SPH method for inviscid shallow water flows, *International Journal for Numerical Methods in Fluids*. 47 (2005) 139-159.
- [103] J.J.H. Miller, On the Location of Zeros of Certain Classes of Polynomials with Applications to Numerical Analysis, *IMA J Appl Math*. 8 (1971) 397-406.
- [104] J.N. Reddy, *An introduction to the finite element method*, McGraw-Hill Inc1993.
- [105] D.D.J. Chandar, M. Damodaran. Computational fluid-structure interaction of a flapping wing in free flight using overlapping grids (27th AIAA Applied Aerodynamics Conference), San Antonio, TX
- [106] H. Li, S.S. Mulay, S. See, On the convergence of random differential quadrature (RDQ) method and its application in solving nonlinear differential equations in mechanics, *CMES - Computer Modeling in Engineering and Sciences*. 48 (2009) 43-82.
- [107] M. Ainsworth, J.T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley & Sons, Inc2000.
- [108] E. Stein, *Error-controlled Adaptive Finite Elements in Solid Mechanics*, John Wiley & Sons Ltd, England, 2003.
- [109] I. Babuska, *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, John Wiley & Sons, Inc1986.
- [110] R.E. Bank, A. Weiser, Some A Posteriori Error Estimators for Elliptic Partial Differential Equations, *Mathematics of Computation*. 44 (1985) 283-301.

- [111] I. Babuska, W.C. Rheinboldt, *A-posteriori* error estimates for the finite element method, International Journal for Numerical Methods in Engineering. 12 (1978) 1597-1615.
- [112] L. Diening, C. Kreuzer, Linear convergence of an adaptive finite element method for the p-laplacian equation, SIAM Journal on Numerical Analysis. 46 (2008) 614-638.
- [113] O.C. Zienkiewicz, J.Z. Zhu, Simple error estimator and adaptive procedure for practical engineering analysis, International Journal for Numerical Methods in Engineering. 24 (1987) 337-357.
- [114] I. Babuska, A. Miller, Post-processing approach in the finite element method - Part 3: A posteriori error estimates and adaptive mesh selection, International Journal for Numerical Methods in Engineering. 20 (1984) 2311-2324.
- [115] I. Babuska, A. Miller, Post-processing approach in the finite element method - Part 2: The calculation of stress intensity factors, International Journal for Numerical Methods in Engineering. 20 (1984) 1111-1129.
- [116] I. Babuska, A. Miller, Post-processing approach in the finite element method - Part 1: Calculation of displacements, stresses and other higher derivatives of the displacements, International Journal for Numerical Methods in Engineering. 20 (1984) 1085-1109.
- [117] O.C. Zienkiewicz, J.Z. Zhu, Superconvergent patch recovery and a posteriori error estimates. Part 1: the recovery technique, International Journal for Numerical Methods in Engineering. 33 (1992) 1331-1364.
- [118] O.C. Zienkiewicz, J.Z. Zhu, Superconvergent patch recovery and a posteriori error estimates. Part 2: error estimates and adaptivity, International Journal for Numerical Methods in Engineering. 33 (1992) 1365-1382.
- [119] C.A. Duarte, J.T. Oden, An h-p adaptive method using clouds, Computer Methods in Applied Mechanics and Engineering. 139 (1996) 237-262.
- [120] L. Gavete, S. Falcón, A. Ruiz, An error indicator for the element free Galerkin method, European Journal of Mechanics, A/Solids. 20 (2001) 327-341.

- [121] W. Han, X. Meng, Error analysis of the reproducing kernel particle method, *Computer Methods in Applied Mechanics and Engineering*. 190 (2001) 6157-6181.
- [122] T.J. Liszka, C.A.M. Duarte, W.W. Tworzydło, hp-Meshless cloud method, *Computer Methods in Applied Mechanics and Engineering*. 139 (1996) 263-288.
- [123] M. Rüter, R. Stenberg, Error-controlled adaptive mixed finite element methods for second-order elliptic equations, *Computational Mechanics*. 42 (2008) 447-456.
- [124] C.K. Lee, Y.Y. Shuai, On adaptive refinement analysis for the coupled boundary element method - Reproducing kernel particle method, *International Journal of Computational Methods in Engineering Science and Mechanics*. 8 (2007) 263-272.
- [125] C.K. Lee, Y.Y. Shuai, An automatic adaptive refinement procedure for the reproducing kernel particle method. Part II: Adaptive refinement, *Computational Mechanics*. 40 (2007) 415-427.
- [126] C.K. Lee, Y.Y. Shuai, An automatic adaptive refinement procedure for the reproducing kernel particle method. Part I: Stress recovery and a posteriori error estimation, *Computational Mechanics*. 40 (2007) 399-413.
- [127] B.B.T. Kee, G.R. Liu, C. Lu, A regularized least-squares radial point collocation method (RLS-RPCM) for adaptive analysis, *Computational Mechanics*. 40 (2007) 837-853.
- [128] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, *Computers and Structures*. 11 (1980) 83-95.
- [129] J. Zhang, G.R. Liu, K.Y. Lam, H. Li, G. Xu, A gradient smoothing method (GSM) based on strong form governing equation for adaptive analysis of solid mechanics problems, *Finite Elem Anal Des*. 44 (2008) 889-909.
- [130] Y. Kondratyuk, R. Stevenson, An optimal adaptive finite element method for the stokes problem, *SIAM Journal on Numerical Analysis*. 46 (2008) 747-775.
- [131] N. Roquet, P. Saramito, An adaptive finite element method for viscoplastic flows in a square pipe with stick-slip at the wall, *Journal of Non-Newtonian Fluid Mechanics*. 155 (2008) 101-115.

- [132] C. Carstensen, Convergence of adaptive finite element methods in computational mechanics, *Applied Numerical Mathematics*. 59 (2009) 2119-2130.
- [133] L. Demkowicz, J.T. Oden, T. Strouboulis, Adaptive finite elements for flow problems with moving boundaries. part I: Variational principles and a posteriori estimates, *Computer Methods in Applied Mechanics and Engineering*. 46 (1984) 217-251.
- [134] H. Blum, A. Rademacher, A. Schröder, Space adaptive finite element methods for dynamic Signorini problems, *Computational Mechanics*. 44 (2009) 481-491.
- [135] L. Champaney, P.A. Boucard, S. Guinard, Adaptive multi-analysis strategy for contact problems with friction: Application to aerospace bolted joints, *Computational Mechanics*. 42 (2008) 305-315.
- [136] Y.F. Nie, S.N. Atluri, C.W. Zuo, The optimal radius of the support of radial weights used in moving least squares approximation, *CMES - Computer Modeling in Engineering and Sciences*. 12 (2006) 137-147.
- [137] P.J. Flory, J. Rehner Jr, Statistical mechanics of cross-linked polymer networks II. Swelling, *The Journal of Chemical Physics*. 11 (1943) 521-526.
- [138] P.J. Flory, J. Rehner Jr, Statistical mechanics of cross-linked polymer networks I. Rubberlike elasticity, *The Journal of Chemical Physics*. 11 (1943) 512-520.
- [139] Y.C. Hon, M.W. Lu, W.M. Xue, X. Zhou, New formulation and computation of the triphasic model for mechano-electrochemical mixtures, *Computational Mechanics*. 24 (1999) 155-165.
- [140] S.K. De, N.R. Aluru, A chemo-electro-mechanical mathematical model for simulation of pH sensitive hydrogels, *Mechanics of Materials*. 36 (2004) 395-410.
- [141] H. Li, T.Y. Ng, Y.K. Yew, K.Y. Lam, Modeling and simulation of the swelling behavior of pH-stimulus-responsive hydrogels, *Biomacromolecules*. 6 (2005) 109-120.
- [142] H. Li, X. Wang, G. Yan, K.Y. Lam, S. Cheng, T. Zou, R. Zhuo, A novel multiphysic model for simulation of swelling equilibrium of ionized thermal-stimulus responsive hydrogels, *Chemical Physics*. 309 (2005) 201-208.

- [143] W. Hong, X. Zhao, J. Zhou, Z. Suo, A theory of coupled diffusion and large deformation in polymeric gels, *Journal of the Mechanics and Physics of Solids*. 56 (2008) 1779-1793.
- [144] J. Zhang, X. Zhao, Z. Suo, H. Jiang, A finite element method for transient analysis of concurrent large deformation and mass transport in gels, *Journal of Applied Physics*. 105 (2009)
- [145] T. Wallmersperger, D. Ballhause, B. Kröplin, M. Günther, Z. Shi, G. Gerlach. Coupled chemo-electro-mechanical simulation of polyelectrolyte gels as actuators and sensors (Electroactive Polymer Actuators and Devices (EAPAD) 2008), San Diego, CA
- [146] T. Wallmersperger, B. Kröplin, J. Holdenried, R.W. Gülch. A coupled multi-field-formulation for ionic polymer gels in electric fields In: Bar-Cohen Y (ed) (Electroactive Polymer, Actuators and Devices-Smart Structures and Materials 2001-), Newport Beach, CA, pp 264-275
- [147] T. Wallmersperger, D. Ballhause, Coupled chemo-electro-mechanical finite element simulation of hydrogels: II. Electrical stimulation, *Smart Materials and Structures*. 17 (2008) 045012.
- [148] T. Wallmersperger, D. Ballhause, B. Kraplin, M. Nther, G. Gerlach, Coupled multi-field formulation in space and time for the simulation of intelligent hydrogels, *Journal of Intelligent Material Systems and Structures*. 20 (2009) 1483-1492.
- [149] E. Samson, J. Marchand, J.L. Robert, J.P. Bournazel, Modelling ion diffusion mechanisms in porous media, *International Journal for Numerical Methods in Engineering*. 46 (1999) 2043-2060.
- [150] H. Li, Kinetics of smart hydrogels responding to electric field: A transient deformation analysis, *International Journal of Solids and Structures*. 46 (2009) 1326-1333.
- [151] H. Li, J. Chen, K.Y. Lam, Multiphysical Modeling and Meshless Simulation of Electric-Sensitive Hydrogels, *Journal of Polymer Science, Part B: Polymer Physics*. 42 (2004) 1514-1531.

- [152] H. Li, R. Luo, Modeling and characterization of glucose-sensitive hydrogel: Effect of Young's modulus, *Biosensors and Bioelectronics*. 24 (2009) 3630-3636.
- [153] H. Li, R. Luo, E. Birgersson, K.Y. Lam, A chemo-electro-mechanical model for simulation of responsive deformation of glucose-sensitive hydrogels with the effect of enzyme catalysis, *Journal of the Mechanics and Physics of Solids*. 57 (2009) 369-382.
- [154] H. Li, R. Luo, K.Y. Lam, Modeling of ionic transport in electric-stimulus-responsive hydrogels, *Journal of Membrane Science*. 289 (2007) 284-296.
- [155] H. Li, T.Y. Ng, Y.K. Yew, K.Y. Lam, Meshless modeling of pH-sensitive hydrogels subjected to coupled pH and electric field stimuli: Young modulus effects and case studies, *Macromolecular Chemistry and Physics*. 208 (2007) 1137-1146.
- [156] H. Li, Y.K. Yew, Simulation of soft smart hydrogels responsive to pH stimulus: Ionic strength effect and case studies, *Materials Science and Engineering C*. 29 (2009) 2261-2269.
- [157] H. Li, Y.K. Yew, K.Y. Lam, T.Y. Ng, Numerical simulation of pH-stimuli responsive hydrogel in buffer solutions, *Colloids and Surfaces A: Physicochemical and Engineering Aspects*. 249 (2004) 149-154.
- [158] H. Li, Z. Yuan, K.Y. Lam, H.P. Lee, J. Chen, J. Hanes, J. Fu, Model development and numerical simulation of electric-stimulus-responsive hydrogels subject to an externally applied electric field, *Biosensors and Bioelectronics*. 19 (2004) 1097-1107.
- [159] R. Luo, H. Li, Simulation analysis of effect of ionic strength on physiochemical and mechanical characteristics of glucose-sensitive hydrogels, *Journal of Electroanalytical Chemistry*. 635 (2009) 83-92.
- [160] R. Luo, H. Li, E. Birgersson, Y.L. Khin, Modeling of electric-stimulus-responsive hydrogels immersed in different bathing solutions, *Journal of Biomedical Materials Research - Part A*. 85 (2008) 248-257.
- [161] R. Luo, H. Li, K.Y. Lam, Modeling and simulation of chemo-electro-mechanical behavior of pH-electric-sensitive hydrogel, *Analytical and Bioanalytical Chemistry*. 389 (2007) 863-873.

- [162] D. Ballhause, T. Wallmersperger, Coupled chemo-electro-mechanical finite element simulation of hydrogels: I. Chemical stimulation, *Smart Materials and Structures*. 17 (2008)
- [163] E. Birgersson, H. Li, S. Wu, Transient analysis of temperature-sensitive neutral hydrogels, *Journal of the Mechanics and Physics of Solids*. 56 (2008) 444-466.
- [164] D. Brock, W. Lee, Dynamic model of a linear actuator based on polymer hydrogel, *Journal of Intelligent Material Systems and Structures*. 5 (1994) 764-771.
- [165] A.N. Chatterjee, Q. Yu, J.S. Moore, N.R. Aluru, Mathematical modeling and simulation of dissolvable hydrogels, *Journal of Aerospace Engineering*. 16 (2003) 55-64.
- [166] J. Chen, H. Li, K.Y. Lam, Transient simulation for kinetic responsive behaviors of electric-sensitive hydrogels subject to applied electric field, *Materials Science and Engineering C*. 25 (2005) 710-712.
- [167] K.S. De, N.R. Aluru, B. Johnson, W.C. Crone, D.J. Beebe, J. Moore, Equilibrium swelling and kinetics of pH-responsive hydrogels: Models, experiments, and simulations, *Journal of Microelectromechanical Systems*. 11 (2002) 544-555.
- [168] S.K. De, R.R. Ohs, N.R. Aluru. Modeling of hydrogel swelling in buffered solutions In: Bar-Cohen Y (ed) (Electroactive Polymer, Actuators and Devices-Smart Structures and Materials 2001-), Newport Beach, CA, pp 285-291
- [169] R.W. Guelch, J. Holdenried, A. Weible, T. Wallmersperger, B. Kroeplin, Polyelectrolyte gels in electric fields - a theoretical and experimental approach, *Proceedings of SPIE - The International Society for Optical Engineering*. 3987 (2000) 193-202.
- [170] A.L. Iordanskii, M.M. Feldstein, V.S. Markin, J. Hadgraft, N.A. Plate, Modeling of the drug delivery from a hydrophilic transdermal therapeutic system across polymer membrane, *European Journal of Pharmaceutics and Biopharmaceutics*. 49 (2000) 287-293.
- [171] B. Kang, Y.d. Dai, X.h. Shen, D. Chen, Dynamical modeling and experimental evidence on the swelling/deswelling behaviors of pH sensitive hydrogels, *Materials Letters*. 62 (2008) 3444-3446.

- [172] M. Kvarnström, A. Westergård, N. Lorén, M. Nydén, Brownian dynamics simulations in hydrogels using an adaptive time-stepping algorithm, *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*. 79 (2009)
- [173] W.M. Lai, J.S. Hou, V.C. Mow, A triphasic theory for the swelling and deformation behaviors of articular cartilage, *Journal of Biomechanical Engineering*. 113 (1991) 245-258.
- [174] K.Y. Lam, H. Li, T.Y. Ng, R. Luo, Modeling and simulation of the deformation of multi-state hydrogels subjected to electrical stimuli, *Engineering Analysis with Boundary Elements*. 30 (2006) 1011-1017.
- [175] K. Lebedev, S. Mafé, P. Stroeve, Convection, diffusion and reaction in a surface-based biosensor: Modeling of cooperativity and binding site competition on the surface and in the hydrogel, *Journal of Colloid and Interface Science*. 296 (2006) 527-537.
- [176] B.A. Mann, K. Kremer, C. Holm, The swelling behavior of charged hydrogels, *Macromolecular Symposia*. 237 (2006) 90-107.
- [177] J. Ostroha, M. Pong, A. Lowman, N. Dan, Controlling the collapse/swelling transition in charged hydrogels, *Biomaterials*. 25 (2004) 4345-4353.
- [178] D. Šnita, M. Pačes, J. Lindner, J. Kosek, M. Marek, Nonlinear behaviour of simple ionic systems in hydrogel in an electric field, *Faraday Discussions*. 120 (2001) 53-66.
- [179] K.J. Suthar, M.K. Ghantasala, D.C. Mancini. Simulation of hydrogel micro-actuation (Microelectronics: Design, Technology, and Packaging III), Canberra, ACT
- [180] T. Traitel, J. Kost, S.A. Lapidot, Modeling ionic hydrogels swelling: Characterization of the non-steady state, *Biotechnology and Bioengineering*. 84 (2003) 20-28.
- [181] Q.X. Wang, H. Li, K.Y. Lam, Meshless simulation of equilibrium swelling/deswelling of PH-sensitive hydrogels, *Journal of Polymer Science, Part B: Polymer Physics*. 44 (2006) 326-337.
- [182] S. Wu, H. Li, J.P. Chen, K.Y. Lam, Modeling Investigation of Hydrogel Volume Transition, *Macromolecular Theory and Simulations*. 13 (2004) 13-29.

- [183] Y. Wu, S. Joseph, N.R. Aluru, Effect of cross-linking on the diffusion of water, ions, and small molecules in hydrogels, *Journal of Physical Chemistry B*. 113 (2009) 3512-3520.
- [184] Y.K. Yew, T.Y. Ng, H. Li, K.Y. Lam, Analysis of pH and electrically controlled swelling of hydrogel-based micro-sensors/actuators, *Biomedical microdevices*. 9 (2007) 487-499.
- [185] Y.G. Yu, Y. Xu, H. Ning, S.S. Zhang, Swelling behaviors of thermoresponsive hydrogels cross-linked with acryloyloxyethylaminopolysuccinimide, *Colloid and Polymer Science*. 286 (2008) 1165-1171.
- [186] Y.Q. Yu, Z.Z. Li, H.J. Tian, S.S. Zhang, P.K. Ouyang, Synthesis and characterization of thermoresponsive hydrogels cross-linked with acryloyloxyethylaminopolysuccinimide, *Colloid and Polymer Science*. 285 (2007) 1553-1560.
- [187] Z. Yuan, L. Yin, H. Jiang. Numerical simulation of transient nonlinear behaviors of electric-sensitive hydrogel membrane under an external electric field (Microfluidics, BioMEMS, and Medical Microsystems V), San Jose, CA
- [188] X. Zhao, Z. Suo, Electromechanical instability in semicrystalline polymers, *Applied Physics Letters*. 95 (2009)
- [189] J. Zhou, W. Hong, X. Zhao, Z. Zhang, Z. Suo, Propagation of instability in dielectric elastomers, *International Journal of Solids and Structures*. 45 (2008) 3739-3750.
- [190] X. Zhou, Y.C. Hon, S. Sun, A.F.T. Mak, Numerical simulation of the steady-state deformation of a smart hydrogel under an external electric field, *Smart Materials and Structures*. 11 (2002) 459-467.
- [191] H. Li, S.S. Mulay, S. See, On the location of zeroes of polynomials from the stability analysis of novel strong-form meshless random differential quadrature method, *CMES - Computer Modeling in Engineering and Sciences*. 54 (2009) 147-199.
- [192] S.S. Mulay, H. Li. Analysis of microelectromechanical systems using the meshless random differential quadrature method (International Conference on Materials for Advanced Technologies, ICMAT 2009), Singapore, pp 29-32

- [193] S.S. Mulay, H. Li, S. See, On the development of adaptive random differential quadrature method with an error recovery technique and its application in the locally high gradient problems, *Computational Mechanics*. 45 (2010) 467-493.

Appendix A: Derivation of characteristic polynomial $\phi(z)$

Following procedure is pursued to derive the characteristic polynomial $\phi(z)$ from the discretized form of the governing PDE. Here the procedure is explained by the central time and space with multi-step scheme, but the similar procedure is followed for rest of the schemes. From Eq. (4.1)

$$\begin{Bmatrix} a_{j,j-1} & a_{j,j} & a_{j,j+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_{j-1}) \\ \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i-1} & a_{i,i} & a_{i,i+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_{i-1}, t_j) \\ \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \end{Bmatrix} \quad (\text{A.1})$$

The weighting coefficients from Eq. (A.1) are computed by Shu's general approach to get

$$\phi_m^{j+1} = \phi_m^{j-1} + r \left[-\phi_{m-1}^j + \phi_{m+1}^j \right] \quad (\text{A.2})$$

where $m=i$ and $r=(at/h)$, where t and h are the spacings of the time and space domains, respectively. Taking Fourier inverse on both sides of Eq. (A.2) results in

$$\begin{aligned} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{ihm\xi} \hat{v}^{n+1}(\xi) d\xi &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{ihm\xi} \hat{v}^{n-1}(\xi) d\xi + \\ \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} r \left(-e^{ih(m-1)\xi} \hat{v}^n(\xi) d\xi + e^{ih(m+1)\xi} \hat{v}^n(\xi) d\xi \right) \end{aligned} \quad (\text{A.3})$$

where $n=j$. Rearranging Eq. (A.3) results in

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{ihm\xi} \left\{ \hat{v}^{n+1}(\xi) - \hat{v}^{n-1}(\xi) + r[e^{-ih\xi} - e^{ih\xi}] \hat{v}^n(\xi) \right\} d\xi = 0 \quad (\text{A.4})$$

In order to satisfy the consistency

$$\hat{v}^{n+1}(\xi) - \hat{v}^{n-1}(\xi) + r[\cos(h\xi) - i\sin(h\xi) - \cos(h\xi) - i\sin(h\xi)] \hat{v}^n(\xi) = 0 \quad (\text{A.5})$$

where $e^{i\theta} = \cos(\theta) + i\sin(\theta)$ relation is used. Substitute $\hat{v}^n(\xi) = \phi^n(z)$ and $h\xi = \theta$ in Eq. (A.5) to get

$$\phi(z) = z^2 - [2r i \sin(\theta)] z - 1, \text{ taking out } \phi^{n-1}(z) \text{ as common} \quad (\text{A.6})$$

Eq. (A.6) is same as Eq. (4.2). Similar procedure is followed to derive all the characteristic polynomials referred in Chapter 4.

Appendix B: Definition of reduced polynomial $\phi_1(z)$

If $\phi(z)$ is any general nonzero polynomial of degree n , as given in Eq. (4.3), and $\phi^*(z)$ is its complex conjugate, as given in Eq. (4.5), the reduced polynomial $\phi_1(z)$ of at most the degree $n-1$ can be defined as [103]

$$\phi_1(z) = \frac{\phi_0^*(0) \phi(z) - \phi(0) \phi_0^*(z)}{z} \quad (\text{B.1})$$

The reduced polynomial $\phi_1(z)$ gives immediate criterion for the self inversiveness of the polynomial $\phi(z)$. As a result, $\phi(z)$ is self inversive polynomial if and only if $\phi_1(z) = 0$. The proof can be referred in [103].

Appendix C: Derivation of discretization equation by Taylor series

In this appendix, the derivation of the consistent equation from the discretized equation by Taylor series is explained for Scheme 1. Identical procedure is followed to obtain the rest of the discretized equations by Taylor series [80]. From Eq. (4.1)

$$\Rightarrow \begin{Bmatrix} a_{j,j-1} & a_{j,j} & a_{j,j+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_i, t_{j-1}) \\ \phi(x_i, t_j) \\ \phi(x_i, t_{j+1}) \end{Bmatrix} = a \begin{Bmatrix} a_{i,i-1} & a_{i,i} & a_{i,i+1} \end{Bmatrix} \begin{Bmatrix} \phi(x_{i-1}, t_j) \\ \phi(x_i, t_j) \\ \phi(x_{i+1}, t_j) \end{Bmatrix} \quad (\text{C.1})$$

The weighting coefficients $a_{i,j}$ are computed by Shu's general approach [37] to get

$$a_{m,m-1} = \frac{-1}{2h}, \quad a_{m,m+1} = \frac{1}{2h}, \quad a_{m,m} = 0 \text{ and} \quad (\text{C.2})$$

$$a_{n,n-1} = \frac{-1}{2t}, \quad a_{n,n+1} = \frac{1}{2t}, \quad a_{n,n} = 0 \quad (\text{C.3})$$

where $m = j$ and $n = i$. Substituting Eqns. (C.2) and (C.3) into Eq. (C.1) leads to

$$\phi_m^{n+1} = \phi_m^{n-1} + r \left[-\phi_{m-1}^n + \phi_{m+1}^n \right] \quad (\text{C.4})$$

The terms ϕ from Eq. (C.4) are replaced by Taylor series expansion as

$$\phi_{m-1}^n = \phi(x_m - h) = \phi - h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} - O(h)^3 \quad (\text{C.5})$$

$$\phi_{m+1}^n = \phi(x_m + h) = \phi + h\phi_{,x} + \frac{h^2}{2}\phi_{,xx} + O(h)^3 \quad (\text{C.6})$$

$$\phi_m^{n-1} = \phi(t_n - t) = \phi - t\phi_{,t} + \frac{t^2}{2}\phi_{,tt} - O(t)^3 \quad (\text{C.7})$$

$$\phi_m^{n+1} = \phi(t_n + t) = \phi + t\phi_{,t} + \frac{t^2}{2}\phi_{,tt} + O(t)^3 \quad (\text{C.8})$$

where $\phi = \phi_m^n$. Substituting Eqns. (C.5) to (C.8) in Eq. (C.4) and simplifying it to get

$$\phi_{,t} - a \phi_{,x} = \phi_{,xxx} \left[\left(\frac{a h^2}{6} \right) - \left(\frac{t^2 a^3}{6} \right) \right] \quad (\text{C.9})$$

where $\phi_{,ttt} = a^3 \phi_{,xxx}$. Therefore, the consistent equation in Eq. (C.9) is same as Eq. (4.34).

Appendix D: Derivation of the ratio of successive amplitude reduction values for the fixed-fixed beam using the explicit and implicit approaches

In this appendix, the derivation steps of the successive amplitude reduction ratio are given for the fixed-fixed beam with the explicit and implicit approaches.

The ODE $\psi(t)$ obtained by the explicit approach is given in Eq. (4.71) as

$$\frac{1}{\psi} \left[k \frac{\partial^3 \psi}{\partial t^3} + \frac{\partial^2 \psi}{\partial t^2} \right] = -\omega_n^2 \quad (\text{D.1})$$

The roots of Eq. (D.1) are computed by substituting $\psi = e^{st}$, and solving $k s^3 + s^2 + \omega_n^2 = 0$ in MATLAB to get the general solution as

$$\psi(t) = C_1 e^{\psi_1 t} + C_2 e^{\psi_2 t} + C_3 e^{\psi_3 t} \quad (\text{D.2})$$

where

$$\begin{aligned} \psi_1 &= \frac{1}{6k} \left[\beta + \frac{4}{\beta} - 2 \right], \quad \psi_2 = \frac{1}{12k} \left[-\beta - \frac{4}{\beta} - 4 \right] + \frac{i\sqrt{3}}{24k} \left[2\beta - \frac{8}{\beta} \right] \text{ and} \\ \psi_3 &= \frac{1}{12k} \left[-\beta - \frac{4}{\beta} - 4 \right] - \frac{i\sqrt{3}}{24k} \left[2\beta - \frac{8}{\beta} \right] \end{aligned} \quad (\text{D.3})$$

where $\beta = \sqrt[3]{-108 \omega_n^2 k^2 - 8 + 12\sqrt{3} \left(\sqrt{27 \omega_n^2 k^2 + 4} \right) \omega_n k}$. The first root ψ_1 is neglected as it grows with time. The remaining roots, ψ_2 and ψ_3 , are complex conjugate of each other. As a result, it is given from ψ_2 and ψ_3 that

$$\nu_1 = \frac{1}{12k} \left[\beta + \frac{4}{\beta} + 4 \right] \quad \text{and} \quad \nu_2 = \frac{\sqrt{3}}{24k} \left[2\beta - \frac{8}{\beta} \right] \quad (\text{D.4})$$

Therefore, the solutions from Eq. (D.2) are given as

$$sol_1 = e^{(-\nu_1 + i\nu_2) t} \quad \text{and} \quad sol_2 = e^{(-\nu_1 - i\nu_2) t} \quad (\text{D.5})$$

Such that

$$sol_1 = e^{-\nu_1 t} [C_1 \cos(\nu_2 t) + C_3 i \sin(\nu_2 t)] \text{ and } sol_2 = e^{-\nu_1 t} [C_4 \cos(\nu_2 t) - i C_5 \sin(\nu_2 t)] \quad (D.6)$$

The amplitudes from Eq. (D.6) are obtained as

$$amp_1 = e^{-\nu_1 t} C_1 \cos(\nu_2 t) \text{ and } amp_2 = e^{-\nu_1 t} C_4 \cos(\nu_2 t) \quad (D.7)$$

The total amplitude from the general solution of characteristic vibration is given as

$$A(t) = e^{-\nu_1 t} [C_1 \cos(\nu_2 t) + C_2 \sin(\nu_2 t)] \quad (D.8)$$

$A(0) = 0$ at $t = 0$, as a result $C_1 = 0$. Therefore, Eq. (D.8) for the time t_1 and $[t_1 + (2\pi / \omega_d)]$ is given as

$$A(t_1) = e^{-\nu_1 t_1} C_2 \sin(\nu_2 t_1), \text{ and } A\left(t_1 + \frac{2\pi}{\omega_d}\right) = e^{-\nu_1 \left[t_1 + \frac{2\pi}{\omega_d}\right]} C_2 \sin\left[\nu_2 \left(t_1 + \frac{2\pi}{\omega_d}\right)\right] \quad (D.9)$$

Eq. (D.9) is simplified as

$$A(t_1) = e^{-\nu_1 t_1} C_2 \sin(\nu_2 t_1), \text{ and } A(t_2) = A\left(t_1 + \frac{2\pi}{\omega_d}\right) = e^{-\nu_1 t_1} e^{\left(-\nu_1 \frac{2\pi}{\omega_d}\right)} C_2 \sin(\nu_2 t_1) \quad (D.10)$$

As a result, the ratio $A(t_2) / A(t_1)$ is given as

$$\frac{A(t_2)}{A(t_1)} = e^{-\nu_1 \left[\frac{2\pi}{\omega_d}\right]} \quad \because \omega_d = \nu_2 \quad (D.11)$$

It is verified that Eq. (D.11) is same as Eq. (4.77). Similar procedure is adopted to compute the successive amplitude reduction ratios by the implicit approach.

List of publications arising from the thesis

Journal papers

Hua Li, **Shantanu S. Mulay**, Simon See, *On the Convergence of Random Differential Quadrature (RDQ) Method and Its Application in Solving Nonlinear Differential Equations in Mechanics*, CMES Computer Modeling in Engineering and Sciences, 48:43-82, **2009**.

Hua Li, **Shantanu S. Mulay**, Simon See, *On the Location of Zeroes of Polynomials from the Stability Analysis of Novel Strong-Form Meshless Random Differential Quadrature Method*, CMES Computer Modeling in Engineering and Sciences, 54:147-199, **2009**.

Shantanu S. Mulay, Hua Li, Simon See, *On the random differential quadrature (RDQ) method: Consistency analysis and application in elasticity problems*, Computational Mechanics, 44:563-590, **2009**.

Shantanu S. Mulay, Hua Li, Simon See, *On the development of adaptive random differential quadrature method with an error recovery technique and its application in the locally high gradient problems*, Computational Mechanics, 45:467-493, **2010**.

Hua Li, **Shantanu S. Mulay**, *2D simulation of pH-sensitive hydrogels by novel strong-form meshless random differential quadrature method*, Journal of Computational Physics (submitted).

Shantanu S. Mulay, Hua Li, *Influence of Young's modulus and geometrical shapes on the 2-D simulation of pH-sensitive hydrogels by the meshless random differential quadrature method*, Computer Methods in Applied Mechanics and Engineering (submitted).

Conference paper

Shantanu Mulay, Hua Li, *Analysis of Microelectromechanical Systems Using the Meshless Random Differential Quadrature Method*, Symposium L: NEMS/MEMS Technology and Devices, International Conference on Materials for Advanced Technologies (ICMAT 2009), Singapore, 28 June – 3 July **2009**.