Development of a novel meshless method - random integral quadrature (RIQ) method and its engineering application for solving integral equations

Zou, Hua

2013

Zou, H. (2013). Development of a novel meshless method - random integral quadrature (RIQ) method and its engineering application for solving integral equations. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/52045>

<https://doi.org/10.32657/10356/52045>

**DEVELOPMENT OF A NOVEL MESHLESS METHOD -
RANDOM INTEGRAL QUADRATURE (RIQ) METHOD
AND ITS ENGINEERING APPLICATION FOR SOLVING
INTEGRAL EQUATIONS**

ZOU HUA

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

2013

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisor, Assistant Professor Li Hua, for providing me with continuous encouragement, support, and sagacious guidance through my four-year Ph.D programme. His passion and enthusiasm in research have inspired me enormously, and it is a great learning experience for me to work under his supervision.

I would also like to thank Professors Ang Whye Teong, Sellakkutti Rajendran, Ng Teng Yong and Wong Chee How for their effort on my work.

Many thanks are conveyed to my colleagues and friends, Dr. Luo Rongmo, Dr. Shantanu Mulay, Dr. Venkataraman Pandurangan, Mr Ye Ting, Ms Tong Shanshan, Mr He Zhongjie, Mr Wu Tao, Ms Chen Lihong, Ms Li Bei, Ms Sheng Lingling, Ms Qu Liang, and Ms Ou Yanjing. I sincerely appreciate their help and friendship, for making my study in NTU more enjoyable and memorable.

I appreciate my family for their warm care and strong support. The constant love and encouragement of my parents and my young brother make my Ph.D years meaningful and happy. I owe special thanks to my beloved husband, Dr. Lai Fukun, who always takes care of me and encourages me during my stressful research period. Without his endless support, understanding and love, it is impossible for me to finish this thesis.

I wish to thank all the administrative staffs of School of Mechanical and Aerospace Engineering for their help in all the administrative matters. Last but not least, I am very

grateful to Nanyang Technological University for granting me the research scholarship and providing me such an exciting research environment.

Table of Content

ACKNOWLEDGEMENTS.....	I
TABLE OF CONTENT	III
SUMMARY.....	IX
LIST OF FIGURES	XII
LIST OF TABLES	XXIII
CHAPTER 1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 MESHLESS METHODS	3
1.2.1 Fundamental of meshless methods.....	3
1.2.2 General procedure for implementation of meshless method.....	3
1.3 INTEGRAL EQUATIONS AND THEIR CLASSIFICATION.....	5
1.3.1 Integral equations	5
1.3.2 Classifications of integral equations	6
1.3.2.1 Fredholm integral equations of the first and second kinds	6
1.3.2.2 Volterra integral equations of the first and second kinds.....	7
1.3.2.3 Singular integral equations	7
1.3.2.4 Integro-differential equations.....	8
1.3.2.5 Nonlinear integral equations	8
1.4 MOTIVATION AND OBJECTIVE	9
1.5 ORGANIZATION OF THE THESIS	9
CHAPTER 2 LITERATURE REVIEW.....	13

2.1	MESHLESS METHODS.....	13
2.1.1	<i>Strong-form methods</i>	13
2.1.2	<i>Weak-form methods</i>	15
2.2	SOLUTION APPROACHES TO INTEGRAL EQUATIONS.....	18
2.2.1	<i>First and second kinds of Fredholm integral equations</i>	19
2.2.1.1	Successive approximation method	19
2.2.1.2	Successive substitution method	20
2.2.1.3	Adomian decomposition method	20
2.2.1.4	Direct computational method	21
2.2.2	<i>First and second kinds of Volterra integral equations</i>	24
2.2.3	<i>Singular integral equations</i>	26
2.2.3.1	Cauchy singular integral equation (CSIE)	26
2.2.3.2	Weakly singular Volterra integral equation	27
2.2.4	<i>Integro-differential equations</i>	29
2.2.5	<i>Nonlinear integral equations</i>	30
CHAPTER 3 DEVELOPMENT OF RANDOM INTEGRAL QUADRATURE (RIQ) METHOD		33
3.1	INTRODUCTION	33
3.2	RANDOM INTEGRAL QUADRATURE (RIQ) METHOD	35
3.2.1	<i>Brief background</i>	35
3.2.2	<i>Formulation of the RIQ method</i>	36
3.2.2.1	Discretization of integral governing equation	37
3.2.2.2	Kriging interpolation	43
3.2.2.3	Finalization of discrete system.....	48
3.2.2.4	Solution of the final system	48
3.2.2.5	Implementation procedures of the RIQ method	49
3.2.3	<i>Stability analysis</i>	54

3.3	SUMMARY	55
CHAPTER 4 NUMERICAL VALIDATION OF THE RIQ METHOD		58
4.1	INTRODUCTION.....	58
4.2	THE SECOND KIND OF FREDHOLM INTEGRAL EQUATIONS	58
4.2.1	<i>One-dimensional problems</i>	59
4.2.1.1	Case one — $y(x) = (x^2 - 1/3) + \int_0^1 x \cdot y(t)dt$	59
4.2.1.2	Case two — $y(x) = 1/4 - x + \int_0^1 (3t - 6x^2)y(t)dt$	61
4.2.1.3	Case three — $y(x) = x + \int_0^1 k(x, t)y(t)dt, 0 \leq x \leq 1, k(x, t) = \begin{cases} x, & x \leq t \\ t, & t \leq x \end{cases}$	62
4.2.1.4	Case four — $y(x) = e^{2x+1/3} - (1/3) \int_0^1 e^{2x-(5/3)t} y(t)dt$	64
4.2.1.5	Case five — $y(x) = f(x) + \int_0^1 e^{x+t} y(t)dt, f(x)$ as a complex trigonometric function ...	66
4.2.1.6	Case six — $y(x) = 0.5x(1-x) + \int_0^1 K(x, t)y(t)dt, K(x, t) = \begin{cases} t(1-x), & 0 \leq t \leq x \\ x(1-t), & x \leq t \leq 1 \end{cases}$	68
4.2.2	<i>Two-dimensional problems</i>	69
4.2.2.1	Case one — $u(x, y) = \int_0^1 \int_0^1 xyu(\xi, \eta)d\xi d\eta + (3/4)xy$	70
4.2.2.2	Case two — $u(x, y) = e^{x+y} - e^{xy}(e-1)^2 + \int_0^1 \int_0^1 e^{xy}u(\xi, \eta)d\xi d\eta$	71
4.2.3	<i>Three-dimensional problem</i>	72
4.3	VOLTERRA INTEGRAL EQUATIONS OF THE SECOND KIND.....	73
4.3.1	<i>One-dimensional problems</i>	73
4.3.1.1	Case one — $u(x) = x^2 - (1/12)x^4 + \int_0^x (x-t)u(t)dt$	73
4.3.1.2	Case two — $u(x) = 1 + \int_0^x u(t)dt$	75
4.3.1.3	Case three — $u(x) = (1/4)x \cos(2x) + \sin(x) - (1/2)x + \int_0^x x \cos(t)u(t)dt$	76

4.3.2	Two-dimensional problems.....	78
4.3.2.1	Case one — $u(x, y) = -(1/4)x^2y^2 + xy + \int_0^x \int_0^y u(\xi, \eta) d\xi d\eta$	78
4.3.2.2	Case two — $u(x, y) = -1 + e^x + e^y + \int_0^x \int_0^y u(\xi, \eta) d\xi d\eta$	79
4.4	SUMMARY	80

CHAPTER 5 APPLICATION OF RIQ METHOD IN IRREGULAR DOMAINS AND FOR APPROXIMATE

SOLUTION OF NONLINEAR INTEGRAL EQUATIONS 106

5.1	INTRODUCTION	106
5.2	APPLICATION OF THE RIQ METHOD IN IRREGULAR INTEGRAL DOMAINS.....	107
5.2.1	<i>Solution methods</i>	107
5.2.1.1	Development of a strategy for nodes scattered in irregular domains	108
5.2.1.2	Discretization of the governing equations	109
5.2.1.3	Kriging interpolation	110
5.2.1.4	Solution of the final system	110
5.2.2	<i>Numerical experiments with validations</i>	110
5.2.2.1	Case one — $u(x, y) = xy - (1/8)\sin(x + y) + \int_{\Omega} \sin(x + y)u(\xi, \eta) d\xi d\eta$ defined on a triangular domain.....	111
5.2.2.2	Case two — $5u(x, y) + \int_{\Omega} u(\xi, \eta) d\xi d\eta = 5xy^2 - 1/40 + 2/15$ defined in a complex domain.....	111
5.3	APPLICATION OF THE RIQ METHOD FOR NONLINEAR INTEGRAL EQUATIONS	113
5.3.1	<i>Solution method</i>	113
5.3.1.1	Introduction of quasilinearization	113
5.3.1.2	Generalized quasilinearization.....	114
5.3.1.3	Development of a strategy combining quasilinearization and RIQ method for nonlinear integral equations	115
5.3.2	<i>Numerical experiments with validations</i>	116

5.3.2.1	Case one — $\phi(t) = (\sin(t) - e^{\sin(t)} + 1) + \int_0^t \cos(s)e^{\phi(s)} ds$	116
5.3.2.2	Case two — $\phi(t) = 2 - e^t + \int_0^t e^{t-s} \cos(s)e^{\phi(s)} ds$	117
5.3.2.3	Case three — $\phi(t) = e^{-t} + \int_0^t e^{s-t} (e^{-u(s)} + u(s)) ds$	118
5.3.2.4	Case four — $\phi(t) = 2t - (1/12)t^4 + 0.25 \int_0^t (t-s)\phi^2(s) ds, \quad 0 \leq t \leq 1$	119
5.3.3	<i>Numerical solution of RIQ method for Blasius boundary-layer problem</i>	120
5.3.3.1	RIQ solution	122
5.3.3.2	Fourth-order Runge-Kutta solution	123
5.3.3.3	Comparisons	123
5.4	SUMMARY	124
CHAPTER 6 APPLICATION OF RIQ METHOD FOR PERIDYNAMICS PROBLEMS		146
6.1	INTRODUCTION	146
6.2	FORMULATION OF PERIDYNAMICS THEORY	148
6.2.1	<i>Governing equation</i>	148
6.2.2	<i>Response function</i>	149
6.3	RIQ METHOD FOR PERIDYNAMICS	153
6.3.1	<i>Discretization of governing equation with the GIQ method</i>	154
6.3.2	<i>A numerical strategy – dynamic relaxation</i>	154
6.3.3	<i>Kriging interpolation</i>	158
6.3.4	<i>Solution of the final system</i>	158
6.4	NUMERICAL CASE STUDIES FOR COMPARISONS BETWEEN RIQ AND FE METHODS	159
6.4.1	<i>Beam subject to tension</i>	159
6.4.2	<i>Beam subject to bending</i>	162
6.4.3	<i>Rod subject to tension</i>	163
6.4.4	<i>Plate subject to tension</i>	164

6.4.5	<i>Plate with a central hole subject to tension</i>	166
6.5	SUMMARY	167
CHAPTER 7 CONCLUSIONS AND FUTURE WORKS		191
7.1	CONCLUSIONS.....	191
7.2	FUTURE WORKS.....	193
REFERENCES.....		196
PUBLICATIONS ARISING FROM THIS THESIS		209

Summary

As is well known, meshless methods are often accepted as an important numerical technique, and are increasingly studied in recent years. Objectives of developing a meshless method are to overcome some drawbacks in conventional numerical techniques, such as the finite element method. However, the main challenges we often encounter here includes the construction of an appropriate interpolation function, which is able to interpolate the field variables with uniformly or randomly distributed field nodes that are scattered in regular or irregular domains, in order to achieve more accurate computation.

The present work aims to develop a novel strong-form meshless technique, termed the random integral quadrature (RIQ) method, for solving generalized integral equations. By the RIQ method, the governing integral equations are discretized directly with randomly or uniformly distributed field nodes. This is achieved by discretizing the integral governing equations first through the generalized integral quadrature (GIQ) technique over a set of background virtual nodes, and then by interpolating the function values of the virtual nodes over a set of the field nodes through the Kriging interpolation technique. Using the interpolation function, the field nodes can be distributed either randomly or uniformly, regardless of the domain being regular or irregular.

In order to validate the RIQ method, several 1-D, 2-D and 3-D integral problems are solved via both the randomly and uniformly distributed field nodes. They include the two types of integral equations, namely the Fredholm integral equation of the second kind and the Volterra integral equation of the second kind. Corresponding convergence rate is

studied in detail for each problem. The numerical solutions of all the problems demonstrate that the RIQ method can achieve good computational accuracy even if only a few field nodes are scattered in the domain, and can also yield a good convergence rate. As expected, the numerical results get closer and closer to the analytical exact solutions as the number of the field nodes increases.

After the RIQ method is formulated and validated, it is performed to solve other complex integral equations, including the integral governing equations with irregular integral domains, and the nonlinear integral governing equations, respectively.

- For integral governing equations with irregular domains, a strategy is developed to distribute the virtual and field nodes in irregular domains. Several numerical experiments are carried out successfully by the RIQ method with good convergence rates.
- For nonlinear integral governing equations, a strategy is proposed for combining the quasilinearization technique with the RIQ method, where the generalized quasilinearization technique is implemented first to linearize the nonlinear integral equations, and then the RIQ method is performed to solve the linearized equations. For examination of the present strategy, several case studies are conducted and high computational efficiency and numerical accuracy are achieved. Specially, the Blasius boundary-layer problem is also solved by the present RIQ method, and the numerically computed results agree well with those by the forth-order Runge-Kutta technique.

Finally the RIQ method is employed to solve the peridynamic problem, which is based on a nonlocal continuum theory and the governing equations are in the form of the integro-differential equations to characterize the constitute relations between the displacements and forces. For numerical solutions of peridynamics, the RIQ method is coupled with an adaptive dynamic relaxation approach, in order to obtain the static solutions of the peridynamic problems. Several basic structures subjected to tensile and bending loads are solved by the present method, and good comparisons are achieved between the RIQ and FEM numerical results.

In summary, the novel meshless method called the RIQ method is developed in this thesis, in order to solve the integral governing equations for engineering applications. The formulation, convergence study and applications of this method are completed and discussed in detail. A number of numerical experiments are carried out with the RIQ method, and the numerically computed results are compared with the analytical solutions or numerical results obtained by other conventional numerical techniques, in which good convergences are achieved. Therefore, the present theoretical analysis together with the numerical case studies has proved the accuracy, efficiency and wide range of applications of the RIQ method.

List of Figures

Figure 1.1. General flowchart of implementation of meshless method.	12
Figure 3.1. Virtual and field nodes.	57
Figure 3.2. Schematic diagram of the GIQ method.	57
Figure 4.1. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.1 for 1-D Fredholm integral equation via 15 uniformly distributed field nodes.	93
Figure 4.2. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.1 for 1-D Fredholm integral equation via 15 randomly distributed field nodes.	93
Figure 4.3. Convergence rates of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by randomly and uniformly distributed field nodes.	94
Figure 4.4. Variations of the global errors of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by randomly or uniformly distributed field nodes.	94
Figure 4.5. Convergence rates of the example case shown in Section 4.2.1.4 for 1-D Fredholm integral equation by randomly and uniformly distributed field nodes.	95
Figure 4.6. Convergence rates of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation by randomly and uniformly distributed field nodes odes.	95

Figure 4.7. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation via 100 uniformly distributed field nodes.	96
Figure 4.8. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation via 100 randomly distributed field nodes.	96
Figure 4.9. Convergence rates of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation by randomly and uniformly distributed field nodes.	97
Figure 4.10. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation via 10×10 uniformly distributed field nodes.	97
Figure 4.11. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation via 10×10 randomly distributed field nodes.	98
Figure 4.12. Convergence rates of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation by randomly and uniformly distributed field nodes.	98
Figure 4.13. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation via 10×10 randomly distributed field nodes.	99
Figure 4.14. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation via 10×10 uniformly distributed field nodes.	99

Figure 4.15. Convergence rates of the example case shown in Section 4.2.3 for 3-D Fredholm integral equation by randomly and uniformly distributed field nodes.	100
Figure 4.16. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation via 15 randomly distributed field nodes.	100
Figure 4.17. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation via 15 uniformly distributed field nodes.	101
Figure 4.18. Convergence rates of the example case shown in Section 4.3.1.2 for 1-D Volterra integral equation by randomly and uniformly distributed field nodes.	101
Figure 4.19. Convergence rates of the example case shown in Section 4.3.1.3 for 1-D Volterra integral equation by randomly and uniformly distributed field nodes.	102
Figure 4.20. Convergence rates of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by randomly and uniformly distributed field nodes.	102
Figure 4.21. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by 15×15 randomly distributed field nodes.	103
Figure 4.22. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by 15×15 uniformly distributed field nodes.	103
Figure 4.23. Convergence rates of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by randomly and uniformly distributed field nodes.	104

Figure 4.24. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by 15×15 randomly distributed field nodes.	104
Figure 4.25. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by 15×15 uniformly distributed field nodes.	105
Figure 5.1. Irregular integral domain.	131
Figure 5.2. Distribution of the field and virtual nodes.	131
Figure 5.3. Variation of the relative errors with the number of the field nodes for the example case shown in Section 5.2.2.1.	132
Figure 5.4. Comparison between the numerical and analytical solutions with the field nodes for the example case shown in Section 5.2.2.1.	132
Figure 5.5. Distribution of the 549 virtual nodes for the example case shown in Section 5.2.2.2.	133
Figure 5.6. Distribution of the 93 field nodes for the example case shown in Section 5.2.2.2.	133
Figure 5.7. Variation of the relative errors with the number of the field nodes for the example case shown in Section 5.2.2.2.	134
Figure 5.8. Comparison between the analytical and numerical solutions with 42 field nodes for the example case shown in Section 5.2.2.2.	134
Figure 5.9. Comparison between the numerical and analytical solutions with 93 field nodes for the example case shown in Section 5.2.2.2.	135

Figure 5.10. Convergence of the iterative scheme over 45 virtual nodes to the analytical solution for the example case shown in Section 5.3.2.1.....	135
Figure 5.11. Variations of the global errors with the number of the virtual nodes for the example case shown in Section 5.3.2.1.	136
Figure 5.12. Convergence of the iterative scheme over 45 virtual nodes to the exact solution for the example case shown in Section 5.3.2.2.....	136
Figure 5.13. Variations of the global errors with the number of virtual nodes for the example case shown in Section 5.3.2.2.	137
Figure 5.14. Convergence of the iterative scheme over 45 virtual nodes to the exact solution for the example case shown in Section 5.3.2.3.....	137
Figure 5.15. Variations of the global errors with the number of virtual nodes for the example case shown in Section 5.3.2.3.	138
Figure 5.16. Convergence of the iterative scheme over 60 virtual nodes to the exact solution with lower solution as $v_0 = 0$ for the example case shown in Section 5.3.2.4.	138
Figure 5.17. Convergence of the iterative scheme over 60 virtual nodes to the exact solution with lower solution as $v_0 = -10$ for the example case shown in Section 5.3.2.4.	139
Figure 5.18. Convergence of the iterative scheme over 60 virtual nodes (iteration once only) to the exact solution with lower solution as $v_0 = 1 - e^{-t}$ for the example case shown in Section 5.3.2.4.....	139

Figure 5.19. Comparisons of the numerical results of y'' between the present RIQ method over 5 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.	140
Figure 5.20. Comparisons of the numerical results of y'' between the present RIQ method over 10 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.	140
Figure 5.21. Comparisons of the numerical results of y'' between the present RIQ method over 20 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.	141
Figure 5.22. Comparisons of the numerical results of y'' between the present RIQ method over 30 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem..	141
Figure 5.23. Comparisons of the numerical results of y' between the present RIQ method over 5 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.	142
Figure 5.24. Comparisons of the numerical results of y' between the present RIQ method over 10 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.	142
Figure 5.25. Comparisons of the numerical results of y' between the present RIQ method over 30 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.	143

Figure 5.26. Comparisons of the numerical results of y' between the present RIQ method over 50 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.	143
Figure 5.27. Comparisons of the numerical results of y between the present RIQ method over 5 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.....	144
Figure 5.28. Comparisons of the numerical results of y between the present RIQ method over 10 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.....	144
Figure 5.29. Comparisons of the numerical results of y between the present RIQ method over 30 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.....	145
Figure 5.30. Comparisons of the numerical results of y between the present RIQ method over 50 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.....	145
Figure 6.1. Interaction failure due to critical stretch.....	170
Figure 6.2. Rectangular beam subject to tension.	170
Figure 6.3. Variation of the velocities over the 51 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.....	171
Figure 6.4. Variation of the displacements over the 51 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.....	171

Figure 6.5. Comparison of the displacements between the FEM and RIQ method over the 51 virtual nodes for the example case shown in Section 6.4.1.....	172
Figure 6.6. Variation of the velocities over the 101 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.....	172
Figure 6.7. Variation of the displacements over 101 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.....	173
Figure 6.8. Comparison of the displacements between the FEM and RIQ method over 101 virtual nodes for the example case shown in Section 6.4.1.....	173
Figure 6.9. Variation of the velocities in the x -direction with the number of iterations over $51 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.1....	174
Figure 6.10. Variation of the displacements in the x -direction with the number of iterations over $51 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.1.	174
Figure 6.11. Comparison of the displacements in the x -direction between the FEM and RIQ method over $51 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.1.	175
Figure 6.12. Contour of the deformations for the example case shown in Section 6.4.1.	175
Figure 6.13. Beam subjected to bending.	176
Figure 6.14. Variation of the velocities in the x -direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2....	176
Figure 6.15. Variation of the velocities in the y -direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2....	177

Figure 6.16. Variation of the displacements in the x -direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2.	177
Figure 6.17. Variation of the displacements in the y -direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2.	178
Figure 6.18. Comparison of the vertical displacements between the RIQ method and FEM for the example case shown in Section 6.4.2.	178
Figure 6.19. Comparison of the horizontal displacements between the RIQ method and FEM for the example case shown in Section 6.4.2.	179
Figure 6.20. Rod subjected to tension.	179
Figure 6.21. Discretization of the y - z plane for the example case shown in Section 6.4.3.	180
Figure 6.22. Variation of the velocities over $51 \times 7 \times 7$ virtual nodes with the number of iterations for the example case shown in Section 6.4.3.	180
Figure 6.23. Variation of the displacements over $51 \times 7 \times 7$ virtual nodes with the number of iterations for the example case shown in Section 6.4.3.	181
Figure 6.24. Comparison of the displacements between the FEM and RIQ method over $51 \times 7 \times 7$ virtual nodes for the example case shown in Section 6.4.3.	181
Figure 6.25. Plate subjected to tension.	182
Figure 6.26. Variation of the velocities in the x -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.	182
Figure 6.27. Variation of the velocities in the y -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.	183

Figure 6.28. Variation of the displacements in the x -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.	183
Figure 6.29. Variation of the displacements in the y -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.	184
Figure 6.30. Comparison of the displacements between the FEM and RIQ method with 51×11 virtual nodes for the example case shown in Section 6.4.4, (a) along the horizontal central line; (b) along the vertical central line.	185
Figure 6.31. Comparison of the displacements between the FEM and RIQ method over 51×21 virtual nodes for the example case shown in Section 6.4.4, (a) along the horizontal central line; (b) along the vertical central line.	186
Figure 6.32. Rectangular plate with a central hole subjected to tension.	187
Figure 6.33. Nodal distribution for the plate with a central hole.	187
Figure 6.34. Variation of the velocities in the x -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5.	188
Figure 6.35. Variation of the velocities in the x -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5.	188
Figure 6.36. Variation of the displacements in the x -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5. .	189
Figure 6.37. Variation of the displacements in the y -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5. .	189

Figure 6.38. Comparison of the horizontal displacements between the RIQ method and
FEM for the example case shown in Section 6.4.5. 190

List of Tables

Table 4.1. Global errors of the example case shown in Section 4.2.2.1 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 50 virtual nodes.	82
Table 4.2. Global errors of the example case shown in Section 4.2.1.2 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 50 virtual nodes using linear basis.....	82
Table 4.3. Global errors of the example case shown in Section 4.2.1.2 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 50 virtual nodes using quadratic basis.	83
Table 4.4. Global errors of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 90 virtual nodes.	83
Table 4.5. Convergence rates of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by the random and uniform field node distributions.	84
Table 4.6. Global errors of the example case shown in Section 4.2.1.4 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes. ...	84
Table 4.7. Convergence rates of the example case shown in Section 4.2.1.4 for 1-D Fredholm integral equation by the random and uniform field node distributions.	84

Table 4.8. Global errors of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes. ...	85
Table 4.9. Convergence rates of example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation by the random and uniform field node distributions.	85
Table 4.10. Global errors of the example case shown in Section 4.2.1.6 for 1-D Fredholm integral equation by 5 uniformly and randomly distributed field nodes with variable virtual nodes.	86
Table 4.11. Global errors of the example case shown in Section 4.2.1.6 for 1-D Fredholm integral equation by 20 uniformly and randomly distributed field nodes with variable virtual nodes.	86
Table 4.12. Global errors of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.	87
Table 4.13. Convergence rates of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation by the random and uniform field node distributions.	87
Table 4.14. Global errors of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.	87
Table 4.15. Convergence rates of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation by random and uniform field node distributions.	88

Table 4.16. Global errors of the example case shown in Section 4.2.3 for 3-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 10×10×10 virtual nodes.	88
Table 4.17. Convergence rates of the example case shown in Section 4.2.3 for 3-D Fredholm integral equation by random and uniform field node distributions.	88
Table 4.18. Global errors of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes with 30 virtual nodes using linear basis.....	89
Table 4.19. Global errors of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes with 30 virtual nodes using quadratic basis.	89
Table 4.20. Global errors of the example case shown in Section 4.3.1.2 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes. ...	90
Table 4.21. Global errors of the example case shown in Section 4.3.1.3 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes. ...	90
Table 4.22. Convergence rates of the example case shown in Section 4.3.1.3 for 1-D Volterra integral equation by random and uniform field node distributions.	91
Table 4.23. Global errors of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.	91
Table 4.24. Convergence rates of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by random and uniform field node distributions.	91

Table 4.25. Global errors of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.	92
Table 4.26. Convergence rates of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by random and uniform field node distributions.	92
Table 5.1. Variation of the global errors for the example case shown in Section 5.2.2.1.	126
Table 5.2. Variation of the global errors for the example case shown in Section 5.2.2.2.	126
Table 5.3. Comparisons between the numerical and analytical solutions for the example case shown in Section 5.3.2.1 with the 30 virtual nodes and 10 field nodes.	127
Table 5.4. Comparisons between the numerical and analytical solutions for the example case shown in Section 5.3.2.2 with the 30 virtual nodes and 10 field nodes.	128
Table 5.5. Comparisons between the numerical and analytical solutions for the example case shown in Section 5.3.2.3 with the 45 virtual nodes and 10 field nodes.	129
Table 5.6. Numerical and analytical solutions for the example case shown in Section 5.3.2.4 with the 30 virtual nodes and 10 field nodes.	130

Chapter 1 Introduction

1.1 Background

In the past decades, a large number of numerical methods were developed successfully for various engineering problems, with the rapid development of high-performance computing technology. Among these developed methods, the finite element method (FEM) is one of the most well-known ones. However, it still encounters several limitations due to its dependence on meshes or elements, for example, the high computational cost in generating meshes, low accuracy in the derivatives of primary field variables, difficulties in implementation for adaptive analysis. Therefore, the idea of getting rid of the meshes and elements is a natural direction to be explored, resulting in a new class of numerical methods, termed meshless methods.

Equation Chapter 1 Section 1

Meshless methods eliminate some or all of the traditional mesh-based views of the computational domain, and rely on a particle view of field problem. One of the earliest meshless methods is smoothed particle hydrodynamics [1], which is often used for analysis of astrophysical phenomena without boundaries, such as exploding stars and dust clouds. Development of meshless methods became rapid from the early 1990s, when a weak form was introduced in the formulation. Since that, more and more effort was devoted to the study of meshless methods. So far several meshless methods has been proposed and developed, including the diffuse element method [2], the element-free

Galerkin method [3], the reproducing kernel particle method [4], the point interpolation method [5], the h - p clouds method [6], and the meshless local Petrov-Galerkin method [7] etc. These methods are widely implemented in solid and fluid mechanics. However, most of them are applied for the solution of ordinary/partial differential equations; only a few of meshless methods are applicable for solving integral equations.

As is well known, various integral equations have been studied since that Able first produced an integral equation in connection with the well-known tautochrone problem in 1825 [8], in which the time of a particle descending to the lowest position of a curve was described. Abel did not realize the importance of the integral equation in general, although he solved it. Since then, the development of analytical and numerical methods for the linear and nonlinear integral equations has attracted much attention from the scientific and engineering fields, and a number of solution techniques have been developed, for example, the classic Galerkin's method, Nystrom method, iteration method, and boundary integral method. Recently, several numerical methods have also been developed for nonlinear or singular integral equations [9-12]. In general, one of the important challenges for the numerical methods to solve integral equations is to approximate the field functions distributed over a set of randomly scattered field nodes.

In this thesis, a novel meshless method is developed, known as the random integral quadrature (RIQ) method, for solving integral equations and various engineering problems defined in a domain, which can be characterized by integral equations. By the RIQ method, an approximate solution of a field function for an integral equation can be obtained easily by a set of randomly or uniformly scatted field nodes in the

computational domain. Therefore, it is a powerful computational technique to solve different integral equations for simulations of various engineering and science problems.

1.2 Meshless methods

1.2.1 Fundamental of meshless methods

One of the most important issues in meshless methods is to create a connection for a set of scattering field nodes, in order to approximate a field variable function through interpolation approaches. The present interpolation or shape functions should have some essential properties, for example, the partition of unity, a certain order of consistency, and the Kronecker Delta functional property. If the shape function satisfies the delta function property, it is easy to impose essential boundary conditions. Otherwise, the boundary conditions could be imposed by some special arrangement.

1.2.2 General procedure for implementation of meshless method

As shown in Figure 1.1, the general flowchart for implementation of meshless methods can be decomposed into several steps.

Firstly, the computational domain is defined, in which real structures, components or domains are in general complex and have to be simplified to a manageable domain. There are many ways to create a properly simplified domain, depending on the algorithms used.

Secondly, nodes are created to discretize the domain, where randomly or uniformly distributed field nodes are arranged within the domain as well as across the boundaries.

The density of these scattered field nodes depends on the computational accuracy requirement of the results one wants to achieve. If there is a high gradient problem, more nodes are required to be distributed there for a desired accuracy of computing results.

Thirdly, the shape function is constructed. The construction of shape functions is the central issue in meshless methods. The nodes within a created local domain or the whole domain are used for the field variable interpolation at the concerned node by suitable interpolation function as

$$\phi_I^h(x, y) = \sum_{i=1}^n N_i(x_I, y_I) \times \phi_i \quad (1.1)$$

where $\phi_I^h(x, y)$ is the function value approximation at the I th field node, $N_i(x_I, y_I)$ is the shape function for the i th field node used for the interpolation, ϕ_i is the function value or nodal parameter at the i th node.

Fourthly, the governing equations are discretized by interpolation over the nodes. Basically there are three approaches to establish the simultaneous equations in meshless methods. The first is based on the principle of virtual work, such as Hamilton's principle, the minimum total potential energy principle and so on. The second is based on the residual methods and is more general. The third is based on Taylor series. The first two are termed the weak form, and the third is termed the strong form.

Fifthly, the global matrix is assembled and then the system equation is obtained at every field node after we applied the strong or weak form method. Here the equations are assembled together to form a final matrix and right-hand side vector. Boundary conditions are imposed at the same time, which is sometimes difficult if the interpolation

functions used don't have the Delta function property. Then some special arrangement is required to impose the boundary conditions.

Finally, the final system of equations is solved numerically. Here there are two main categories of methods: direct methods and iterative methods. Often-used direct methods include Gauss elimination and the matrix decomposition method, such as LU decomposition, LDL^T decomposition, and Cholesky factorization. Iterative methods include the Gauss-Jacobi method, the Gauss-Seidel method, generalized conjugate residual methods, the line relaxation method and so on.

1.3 Integral equations and their classification

1.3.1 Integral equations

Many natural phenomena are studied and described mathematically by integral equations. An integral equation is defined as an equation, in which the unknown function $\phi(\mathbf{x})$ appears under the integral sign. The study of integral equations is one of the most important research areas in both pure and applied mechanics. It has an enormously diverse range of applications in all kinds of engineering problems, such as the electromagnetic problem [13] and the peridynamic problem [14]. Many initial and boundary value problems can be transformed into integral equations. A typical form of integral equations in $\phi(\mathbf{x})$ is written as

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_{D(\mathbf{x})} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} \quad (1.2)$$

where $K(\mathbf{x}, \mathbf{y})$ is the kernel of the integral equation, $D(\mathbf{x})$ is the integral domain in the real space $\mathbf{R}^n$ and $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$. It is noted that the kernel $K(\mathbf{x}, \mathbf{y})$ and the function $f(\mathbf{x})$ in Eq. (1.2) are given, and λ is a constant parameter. The present RIQ method is devoted to determining the unknown function $\phi(\mathbf{x})$.

1.3.2 Classifications of integral equations

In general, the integral equations can be classified into several categories, depending on different criteria. For example, according to different types of integration domains, the integral equations can be classified into Fredholm integral equations with a constant integral domain, and Volterra integral equations which have the varied integral domain with independent variable [15]. In the following subsections, the definitions of all typical kinds of integral equations are given, where the unknown function is denoted by $\phi(\mathbf{x})$, the kernel function of the integral equation by $K(\mathbf{x}, \mathbf{y})$, and the free term function by $f(\mathbf{x})$.

1.3.2.1 Fredholm integral equations of the first and second kinds

The Fredholm equation has an integral domain independent of the field variable, as exemplified in Eqs. (1.3) and (1.4). This type of integral equations can be sorted further into two kinds of the integral equations, according to the existence of the known function $f(\mathbf{x})$. They are called the first kind of Fredholm integral equations as shown by Eq. (1.3), and the second kind of Fredholm integral equations as shown by Eq. (1.4).

$$\phi(\mathbf{x}) = \int_D K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} \quad (1.3)$$

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_D K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} \quad (1.4)$$

where $\lambda \neq 0$ is a parameter, D is a closed bound set in real space and it is independent of the field variable. The first kind of Fredholm integral equations is an ill-posed problem, since there is no bounded inverse for the integral operator in the operator form of the integral equations [16]. However, if the kernel function $K(\mathbf{x}, \mathbf{y})$ is singular and absolutely integrable, the ill-condition-ness is possibly manageable.

1.3.2.2 Volterra integral equations of the first and second kinds

The Volterra integral equation requires the integration over an unspecified interval in the general forms as follows,

$$\int_a^{b(\mathbf{x})} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} = \phi(\mathbf{x}) \quad (1.5)$$

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_a^{b(\mathbf{x})} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} \quad (1.6)$$

where $b(\mathbf{x})$ varies with the independent variable $\mathbf{x}$. Similarly, this type of integral equation can also be categorized into the first- and second- kinds Volterra integral equations, as shown in Eqs. (1.5) and (1.6), respectively. The difficulty for numerically solving the Volterra integral equation of the first kind also comes from their ill-conditioned, similarly to the Fredholm integral equations of the first kind.

1.3.2.3 Singular integral equations

An integral equation may have singular properties, where the singularity of the integral equations can be caused by the two conditions. One is that the integral domain of

the equation is unbounded, as shown in Eq. (1.7). The other is that the kernel of the integrals becomes unbounded at a certain point in the interval, such as in Eq. (1.8).

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_{-\infty}^{\infty} \phi(\mathbf{y}) d\mathbf{y} \quad (1.7)$$

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \int_0^x \frac{1}{(\mathbf{x}-\mathbf{y})^\alpha} \phi(\mathbf{y}) d\mathbf{y}, \quad 0 \leq \alpha \leq 1 \quad (1.8)$$

Solving the singular integral equations numerically is a mathematical challenge, especially due to such unbounded properties.

1.3.2.4 Integro-differential equations

When an integral operator is coupled with a differential one, an integro-differential equation results, where the unknown function $\phi(\mathbf{x})$ is appeared in both the integrated and differentiated terms in an equation. For example, Eq. (1.9) is the Volterra type of integro-differential equation, and Eq. (1.10) the Fredholm type of integro-differential equation.

$$\frac{\partial^2 \phi(x)}{\partial x^2} = f(x) + \lambda \int_0^x (x-y) \phi(y) dy, \quad \phi(0) = 0, \quad \frac{\partial \phi(0)}{\partial x} = 1, \quad (1.9)$$

$$\frac{\partial \phi(x)}{\partial x} = f(x) + \lambda \int_0^1 \phi(y) dy, \quad \phi(0) = 1. \quad (1.10)$$

1.3.2.5 Nonlinear integral equations

The Volterra and the Fredholm integral equations are also referred to the nonlinear integral equations, if the unknown function $\phi(\mathbf{x})$ appearing in the integral is in the function form $G(\phi(\mathbf{x}))$, and $G(\phi)$ is a nonlinear function of ϕ , i.e.

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_a^x K(\mathbf{x}, \mathbf{y}) G(\phi(\mathbf{y})) d\mathbf{y} \quad (1.11)$$

If $f(\mathbf{x}) = 0$, the resulting equation is then called a homogeneous integral equation, otherwise a non-homogeneous equation.

1.4 Motivation and objective

As is well known, the General Integral Quadrature (GIQ) method is an effective technique for integral calculations [17]. However, it always requires a regular domain of integration, where all nodes are required to lie along straight lines. Therefore, the main motivation and objective behind the present work is to overcome the shortcomings of the GIQ method, for extending the application of the GIQ method to irregular domains through scattering the field nodes either randomly or uniformly. A novel meshless technique is thus developed, termed the Random Integral Quadrature (RIQ) method.

In the RIQ method, the GIQ technique is used first to discretize the integral governing equation at the virtual nodes within the domain. Approximated function value at every virtual node is approximated further by Kriging interpolation function over the field nodes, where the Kriging function is employed to interpolate the function value at the virtual nodes by the field nodes. A system of linear algebraic equations is then obtained and finally solved numerically.

1.5 Organization of the thesis

This thesis is composed of seven chapters, each of which consists of several subsections. The arrangement of the thesis is as follows.

In Chapter 2, a literature review is presented on meshless methods and numerical techniques for integral equations. The review on meshless methods is mainly on two types of the meshless methods, namely the strong and weak forms. The review on the numerical techniques for integral equations is mainly on five types of integral equations, i.e., the Fredholm integral equation, the Volterra integral equation, singular integral equation, integro-differential equation, and nonlinear integral equation.

In Chapter 3, the RIQ method is developed for one-, two- and three-dimensional domains, based on the GIQ technique as well as the Kriging interpolation function, where the RIQ method is formulated in detail. The solution procedure of RIQ method is also described in a step by step manner.

In Chapter 4, a number of case studies are conducted for the validation of RIQ method. Corresponding convergence studies are also carried out. The case studies include the Fredholm integral equations of the second kind and the Volterra integral equations of the second kind in one-, two- and three-dimensional domains.

In Chapter 5, the RIQ method is applied for solving other complex integral equations, including integral equations in irregular domains and nonlinear integral equations. The detailed implementation and strategies required for applying the method are described. Numerical experiments are given to validate the accuracy and efficiency of the RIQ method for solving these two types of integral equations.

In Chapter 6, the RIQ method is utilized to solve peridynamic problems. Peridynamics is a nonlocal continuous theory to describe mechanical problems, in which the governing equation is of the form of integro-differential equations. The RIQ method is combined with the adaptive dynamic relaxation technique to obtain the static solutions

of these problems. Several mechanical problems are solved to investigate the applicability and computational accuracy of this method, by comparison of the numerical results with those obtained with a finite element method.

Finally, Chapter 7 presents a summary of the present along with recommendations for future work.

It is mentioned that MATLAB is employed to implement the RIQ method in this thesis. Specifically, the MATLAB 7.0 and a computer hardware of Intel 2 core cpu E8400, 4.00 GB ram with the windows operating system are used for all the computations.

Figure 1.1. General flowchart of implementation of meshless method.

Chapter 2 Literature Review

2.1 Meshless methods

Meshless methods have achieved great progress over the past few years. Various meshless methods are mainly categorized into two groups: the meshless methods based on the weak form of governing equations, and the meshless methods based on the strong form of governing equations. The literature survey is given below as per these two groups.

2.1.1 Strong-form methods

In order to approximate the strong form of a partial differential equation using a meshless method, the partial differential equation and boundary equations are usually discretized by a collocation technique. One of the famous meshless methods based on the strong form is the finite point method (FPM) [18], in which the weighted least-square interpolation technique is used for an interpolation of the data, and the point collocation technique is used to solve the governing differential equations. The Hermite cloud method (HCM) [19] is also considered as a strong-form meshless method, in which the Hermite interpolation function is combined with a point collocation method to discretize the governing differential equations, where the Hermite interpolation theorem is based on a fixed reproducing kernel particle method and it is constructed to directly calculate field variables and their first-order derivatives. This method is good for the local high

gradient problems, since the first-order derivative is calculated by directly solving the system of equations. The method has been applied successfully for simulation of submarine bed problems and a steady-state analysis of a pH-sensitive hydrogel. Liu *et al.* [20] proposed the gradient smoothing method (GSM), which was based on a gradient smoothing with a directional derivative technique to develop the first- and second-order derivative approximations for a node of interest. It was coupled with the collocation method to solve the strong form of governing partial differential equations.

Equation Chapter 2 Section 1

As one of the meshless methods, the differential quadrature (DQ) method was proposed by Bellman *et al.* [21] to solve the strong-form differential equations, in which the derivative at every grid point is approximated by the weighted sum of function values in the whole domain provided that all the grid points are always collinear. The most important task in the DQ method is to determinate the weighting coefficients. Bellman *et al.* [21] suggested two approaches. The first is to use the polynomial function as a test function, and the second is to choose the coordinates of the grid points as the roots of shift Legendre polynomials. However, when the order of the system of algebraic equations is very high, the resulting matrix is highly ill-conditioned. In order to overcome this disadvantage, Quan and Chang [22] proposed another approach, where the weighting coefficients are calculated by Lagrange interpolation polynomials. Shu *et al.* [23] proposed a general differential quadrature (GDQ) method, in which both the approaches proposed by Bellman [21] and Quan and Chang [22] were combined together. Based on this GDQ method, Mulay *et al.* [24] proposed a random differential quadrature

(RDQ) method to extend the applications of the GDQ method for solving partial differential equations and engineering problems.

Other strong-form meshless methods include the Hp-cloud method [6], radial point collocation method [25] and so on. Detailed discussions of these methods can be found in Ref. [26].

Meshless strong-form methods generally have some attractive advantages; for example, its simple algorithm, computational efficiency, and the fact that no background mesh is needed. However, meshless strong methods are usually unstable and less accurate, especially for problems with derivative boundary conditions. Thus another type of meshless methods has emerged, namely the meshless methods based on Galerkin weak forms. Compared with the strong form methods, the weak form methods are more stable.

2.1.2 Weak-form methods

In the weak form, the governing equations are integrated once and then solved in integral form. The first true meshless weak-form method was proposed by Lucy [1] and Gingold and Monaghan [27], named the smooth particle hydrodynamics (SPH), for simulation of large astro-physics problems such as exploding stars and dust clouds. The basic idea of this method consists in two parts. The first part is the “integral representation” or “kernel approximation” of the field function, in which the multiplication of the field and kernel functions is represented in the form of integration over a local domain. The second part is the “particle approximation”, in which the integral representation is approximated by summation over a local domain. As this

method involves smoothing a function by integration over a local domain using the particle approximation, it was named as the smooth-particle method. After SPH, numerous meshless weak-form methods were developed, especially from the early 1990s. Several significant works were published, for example, the diffuse element method (DEM) by Nayroles *et al.* [2] and the element-free Galerkin (EFG) method by Belytschko *et al.*[28]. Subsequently, the meshless methods based on the Galerkin weak forms have made significantly achievements.

In 1992 Nayroles *et al.* [2] proposed DEM to overcome the limitations of FEM, where FEM interpolation function was replaced by the local weighted least-square (WLS) interpolation functions. This WLS interpolation function is valid for a local domain surrounding the interest node, and vanishes outside the local domain. Due to this property, the nature in the local domain was reserved. Based on DEM, Belytschko *et al.* [28] developed the EFG method, in which the moving least-square (MLS) interpolation function is used with the Galerkin weak form. The EFG method is a refined form of DEM, and it is formed to give accurate results and to be stable [28, 29]. Braun and Sambridge [30] proposed another numerical method for solving partial differential equations, called the nature element method (NEM), in which the geometrical concept of “natural neighbors” is used, in order to solve large deformation problems and solid-fluid interactions on a deformation mesh as well as high-order partial differential equations, even with a highly irregular mesh. Another widely-used meshless method is the reproducing kernel particle method (RPKM) developed by Liu and co-workers [4]. The main idea of RPKM is to improve the SPH approximation to satisfy consistency requirement via a correction function. The method produces a smoother shape function,

and consequently provides higher accuracy for large-deformation problems. Later, Liu and his team proposed another famous meshless method, named the point interpolation method (PIM), based on the Galerkin weak form [5]. In this method, the shape functions were constructed based on a number of scattered nodes located in the support domain. There are two types of PIMs that use the polynomial basis and radial basis functions, respectively. The moment matrix in the polynomial-basis PIM may be singular, whereas the radial-basis PIM is very stable and robust for arbitrary nodal distributions. Melenk and Babuska [31] proposed the partition of unity finite element method (PUFEM) for improvement of FEM in a rough or oscillatory solution, when classical polynomial based on FEM fails. The basic idea is to construct conforming subspaces. By adding special functions, the standard finite element spaces are augmented, thus high-order function approximation is constructed. Atluri and Zhu [7] developed a meshless method, called the meshless local Petrov-Galerkin (MLPG) method, where MLS approximation is employed to construct the shape functions. In the Galerkin based FEM and EFGM, the Galerkin weak form is used over the whole domain, while the weak form is used in a local sub-domain in the MLFG method. By choosing the trial functions from different spaces, the local integration can be simplified. Li *et al.* [32] proposed the local Kriging (LoKring) method, where the Kriging technique was a generalized linear regression approach for formulation of an optimal estimator in a minimum mean square error sense. In this method, the Kriging interpolation is used to construct a function approximation, and then the weighted residual method is used to derive the local weak form of partial differential equations.

Other meshless weak-form methods include the boundary type of meshless approaches, which were derived by combining the boundary integral equation (BIE) with the meshless techniques. Zhu *et al.* [33] proposed the local boundary integral equation (LBIE) method, based on the MLS approximation in a weak unsymmetrical form. The domain boundary is discretized by the distributed nodes, and then the BIE is used locally to construct the system of equations at these nodes. Zhu *et al.* [33, 34] also successfully used this method for solving linear and nonlinear problems. Mukherjee and Mukherjee [35] proposed the boundary node method (BNM), in which the domain boundary of the problem is discretized by a group of nodes. The MLS approximation is used to discretize the BIEs by arbitrarily distributed boundary nodes. Mandal *et al.* applied this method for the three-dimensional problems of potential theory and linear elasticity [36, 37]. Liu and Gu also did some work on the boundary type of meshless methods, by developing the boundary point interpolation method (BPIM) [38] and the boundary radial point interpolation method (BRPIM) [39], based on the PIM and RPIM shape functions respectively.

2.2 Solution approaches to integral equations

There are a number of the solving techniques for integral equations, in which two important traditional methods are the successive approximation and successive substitution methods [8]. In addition, the series method and the direct computational approach are useful for some problems that involve integral equations. Recently, the Adomian decomposition method (ADM) and the modified decomposition method were developed for solving highly nonlinear integral equations [8]. Singular integral equations

encountered by the convolution type of the Able and Volterra integral equations were solved by the Laplace transform method. Following the classifications of integral equations in Section 1.3.2 above, the numerical methods for solving the five types of the integral equations are reviewed as below.

2.2.1 First and second kinds of Fredholm integral equations

There are several traditional methods for numerical solutions of the Fredholm integral equations; and these are discussed here.

2.2.1.1 Successive approximation method

The successive approximation method can easily be used for the basic Fredholm integral equation of the second kind,

$$\phi(x) = f(x) + \lambda \int_a^b K(x, t) \phi(t) dt \quad (2.1)$$

Let us consider the zero-th approximation of $\phi(x)$ as $\phi_0(x) = f(x)$, which can be selected as any real function, e.g. $\phi_0(x) = 0, 1, \dots$, or x . Accordingly the first approximation $\phi_1(x)$ is defined as

$$\phi_1(x) = f(x) + \lambda \int_a^b K(x, t) \phi_0(t) dt \quad (2.2)$$

The second approximation $\phi_2(x)$ is written via replacing $\phi_0(x)$ by $\phi_1(x)$, i.e.,

$$\phi_2(x) = f(x) + \lambda \int_a^b K(x, t) \phi_1(t) dt \quad (2.3)$$

In a similar manner, one can write the n th approximation as

$$\phi_n(x) = f(x) + \lambda \int_a^b K(x, t) \phi_{n-1}(t) dt \quad n \geq 1 \quad (2.4)$$

The final solution $\phi(x)$ is obtained as

$$\phi(x) = \lim_{n \rightarrow \infty} \phi_n(x) \quad (2.5)$$

Therefore, the final resulting solution $\phi(x)$ is independent of the choice of $\phi_0(x)$.

2.2.1.2 Successive substitution method

This method is similar to the successive approximation method in Section 2.2.1.1. By substituting $\phi(x)$ into the second term of Eq. (2.1), it is given by the equation itself, i.e.,

$$\phi(x) = f(x) + \lambda \int_a^b K(x, t) f(t) dt + \lambda^2 \int_a^b K(x, t) \int_a^b K(t, t_1) \phi(t_1) dt_1 dt \quad (2.6)$$

By repeating this, one can obtain

$$\begin{aligned} \phi(x) = & f(x) + \lambda \int_a^b K(x, t) f(t) dt \\ & + \lambda^2 \int_a^b K(x, t) \int_a^b K(t, t_1) \phi(t_1) dt_1 dt \\ & + \lambda^3 \int_a^b K(x, t) \int_a^b K(t, t_1) \int_a^b K(t_1, t_2) f(t_2) dt_2 dt_1 dt \\ & + \dots \end{aligned} \quad (2.7)$$

The series solution given by Eq. (2.7) converges uniformly in the interval $[a, b]$, if $\lambda M(b-a) < 1$, where $|K(x, t)| < M$.

2.2.1.3 Adomian decomposition method

By this method, the solution of Eq. (2.1) is usually expressed in a series form defined as

$$\phi(x) = \sum_{n=0}^{\infty} \phi_n(x) \quad (2.8)$$

Substituting Eq. (2.8) into both sides of Eq. (2.1) gives

$$\sum_{n=0}^{\infty} \phi_n(x) = f(x) + \lambda \int_a^b K(x, t) \sum_{n=0}^{\infty} \phi_n(x) dt \quad (2.9)$$

The components $\phi_0(x)$, $\phi_1(x)$, ..., and $\phi_n(x)$ are determined in the following manner:

$$\begin{aligned} \phi_0(x) &= f(x) \\ \phi_1(x) &= \lambda \int_a^b K(x, t) \phi_0(t) dt \\ &\vdots \\ \phi_n(x) &= \lambda \int_a^b K(x, t) \phi_{n-1}(t) dt \end{aligned} \quad (2.10)$$

The main idea is then to determine $\phi_0(x)$ by the known function $f(x)$. After $\phi_0(x)$ is determined, $\phi_1(x)$, $\phi_2(x)$, ..., $\phi_n(x)$, are determined. $\phi(x)$ is then obtained from Eq. (2.8).

2.2.1.4 Direct computational method

In this method, the kernel $K(x, t)$ is decomposed as

$$K(x, t) = \gamma(x) \psi(t) \quad (2.11)$$

The Fredholm equation is then written as

$$\phi(x) = f(x) + \lambda \gamma(x) \int_a^b \psi(t) \phi(t) dt \quad (2.12)$$

If

$$\int_a^b \psi(t) \phi(t) dt = \beta \quad (2.13)$$

Then Eq. (2.12) can be expressed as

$$\phi(x) = f(x) + \lambda \beta \gamma(x) \quad (2.14)$$

Apart from the above traditional methods, there are many other numerical approaches developed for Fredholm integral equations. For example, Allen and Wing [40] studied how to speed up the iterative methods for a class of Fredholm integral equations, wherein the synthetic approach based on the transport theory was extended to a large class of integral equations, finding the computational advantage over the conventional Neumann approach to be substantial. Javidi and Golbabai [41] used the homotopy perturbation method (HPM) to solve a system of Fredholm integral equations. Liu [42] employed the Chebyshev polynomial for approximation of the Fredholm integral equations. Golberg completed a series of studies for the Fredholm integral equations that resulted in three contributions [43-45]: (a) establishment of the equivalence between a family of Fredholm integral equations and a related Cauchy system of integro-differential equations, (b) development of an initial-value method equivalent to a classical initial value version by solving a set of linear equations, and (c) the initial value method for nonlinear Fredholm equations. It is noted that all the methods mentioned above are able to solve Fredholm integral equations of the first and second kinds. The numerical approaches, which were proposed for solving either the first or the second kind of Fredholm integral equations, are reviewed as below, respectively.

In order to solve Fredholm integral equations of the first kind, Maleknejad developed an efficient method, based on the Legendre wavelet approach[46], and also studied the convergence of the numerical solutions for the first kind of the Fredholm integral equations with degenerated kernel [16]. Yousefi and Banifatemi [47] presented a numerical solution of the Fredholm integral equations based on the cosine and sine (CAS) wavelet approximations. Hanson and Phillips [48] presented a procedure for numerically

solving the first kind of the linear Fredholm integral equations, where the approximate solution was expressed as a continuous piecewise linear function, and involved in a collocation, followed by the solution of an appropriately scaled stabilized linear algebraic system. Hsiao *et al.* [49] formulated a Galerkin collocation scheme for analysis of Fredholm integral equations of the first kind, which results in high accuracy due to the Galerkin principle and high speed due to the collocation methods. Numerical experiments validated both the high convergence rate and high accuracy. Another approximation function for the linear Fredholm integral equations is the hybrid Taylor and Block-Pulse function [50], by which good degree of computational accuracy is achieved through the numerical examples.

For Fredholm integral equations of the second kind, Abbasbandy proposed a numerical technique for approximately solving the second kind of the linear fuzzy Fredholm integral equations [51], where the linear fuzzy Fredholm integral equation was converted into two systems of integral equations of the second kind in the crisp case, and then solved by the Nystrom method. As a result, the solution for the linear fuzzy Fredholm integral equations was obtained approximately. Babolian [52] used the Adomian decomposition technique to solve a system of linear and nonlinear Fredholm integral equations of the second kind. After that, Babolian [53] proposed an approach for analysis of the second kind of Fredholm integral equations, in which a complementary pair of the orthogonal triangular function set derived from the block pulse function set was used. Farnoosh and Ebrahimi [54] used the Monte Carlo method for solution of the Fredholm integral equations of the second kind, based on the random sampling and the simulation of a continuous Markov chain. Golbabai and Keramati [55] constructed a

simple and approximate solution for the linear Fredholm integral equations of the second kind, through the Adomian decomposition technique. Kravchenko and Basarab [56] developed an approximation by the atomic function and numerical approaches for the Fredholm integral equations of the second kind. Long and Nelakanti [57] suggested a functional approximation to calculate the linear functional solution of Fredholm integral equations of the second kind with superconvergence, when applied to the collocation or the multi-projection approaches. The Wavelet Petrov-Galerkin (WPG) method based on discontinuous orthogonal multi-wavelets was used for solving Fredholm integral equations of the second kind [58]. Miel [59] analyzed the relationship among several numerical techniques for Fredholm integral equations of the second kind. In their paper, the weighted least-squares convergence of the orthogonal expansions was discussed, and the corresponding parallel algorithms were given. Rashidinia and Zarebnia [60] presented a collocation procedure for linear Fredholm integral equations of the second kind, using sine basis functions. Saberi-Nadjafi and Heidari [61] constructed an repeated modified trapezoid quadrature scheme for approximation of linear integral equations of the second kind. Vainikko *et al.* [62] proposed fast solvers for Fredholm integral equations of the second kind, based on a discretized wavelet Galerkin method with the Sloan improvement of the Galerkin solution.

2.2.2 First and second kinds of Volterra integral equations

There are a large number of numerical methods to deal with the Volterra integral equations. Volterra equations of the first and second kinds are expressed as follows,

$$\int_a^x K(x, y)\phi(y)dy = f(x) \quad (2.15)$$

$$\phi(x) = f(x) + \lambda \int_a^x K(x, y)\phi(y)dy \quad (2.16)$$

Some traditional methods mentioned above in Section 2.2.1, such as the successive approximations method, the successive substitutions method and the Adomian decomposition method, can also be used to solve Volterra integral equations.

Apart from these traditional numerical methods, other numerical techniques were also proposed for Volterra integral equations. A formula for the calculation of Taylor coefficients for solutions of Volterra integral equations was derived for both linear and nonlinear systems [63]. Babolian [53] used the Chebyshev collocation method to solve the functional integral equations of the first and second kinds by transforming the integral equations into a system of linear algebraic equations. Brunner *et. al.* [64] applied the collocation and iterated collocation methods, based on the mixed interpolation methods, to solve the Volterra integral equations. Conte and Prete [65] developed a fast discrete collocation method for the Hammerstein type of Volterra integral equations, in which the Laplace transform of the kernel was known as a priori, and multistep collocation methods were applied for Volterra integral equations thereafter [66]. Maleknejad *et. al.* [67] studied the numerical solution of the first kind of Volterra integral equations with wavelet basis, in which the collocation method was used as a projection method to convert the integral equation to a system of linear equations. Mckee *et al* [68] developed an Euler type method for solving the two-dimensional Volterra integral equations of the first kind. Messina *et al* [69] proposed a version of the trapezoidal direct quadrature method to solve Volterra integral equations with

discontinuous kernel. Oja and Saveljeva [70] studied cubic spline collocation for Volterra integral equations.

2.2.3 Singular integral equations

2.2.3.1 Cauchy singular integral equation (CSIE)

Several numerical methods are reviewed for solution of the Cauchy singular integral equations[71], including the Galerkin method, collocation method, quadrature method, spline based method, and a projector-spline based method. Quadrature methods for solution of CSIE extended the Nyström method for the singular integral equations of the second kind. The spline based methods were developed in view of the high computational cost of the polynomial based methods [72-74]. Cauchy singular equations over an interval have been studied in Ref. [75] in the weighted spaces of continuous functions as well as convergence rate for a general method. Subsequently, Junghanns and Muller [76] used a collocation method, with respect to Chebyshev nodes of the second kind together with a Newton iterative technique, to solve a class of singular nonlinear Cauchy integral equations, where the relationship was also discussed between the convergence of the Newton method and the stability of the respective collocation method. After that, they presented selective results of the numerical analysis of the one-dimensional Cauchy singular integral equation, especially the stability of operator sequences associated with different projection methods [77]. A numerical method including the discrete collocation was proposed for the approximate solution of some Fredholm and Cauchy integral equations [78]. A Nyström method was also presented to

approximate the solutions of Cauchy singular integral equations with constant coefficients having a negative index [11].

2.2.3.2 Weakly singular Volterra integral equation

In order to solve weakly singular Volterra integral equations, several research groups have made significant contributions which are reviewed as below.

Pedas and Vainikko [79] discussed the possibility to construct high-order numerical algorithms over uniformly or mildly graded grids for solving the second kind of weakly singular linear Volterra integral equations or other non-smooth kernels. They regularized the integral equation by introducing a suitable independent variable, and then solved the transformed equation by a piecewise polynomial collocation method on a set of mildly graded or uniform grid.

De Hoog and Weiss [80] developed high-order methods for a class of Volterra integral equations with weakly singular kernels. Schemes based on the product in integration analogue of Simpson's rule were proposed, which produced the convergence of order $h^{7/2}$. The numerical stability of the methods was also examined through asymptotic error estimates.

Brunner and Nørsett [81] studied the superconvergence of numerical solutions of Volterra integral equations of the second kind with continuous or weakly singular kernels by collocation methods in certain piecewise polynomial spaces, and concluded that the ordinary polynomial spline collocation cannot lead to high-order convergence. Therefore they developed a nonpolynomial spline collocation for the Volterra equations with weakly singular kernels [82], in which they introduced special nonpolynomial

spline spaces that were modeled after construction of these solutions near the points of non-smooth behavior. Collocation in these spaces leads to high-order methods.

Hairer *et al.* [83] obtained numerical solutions of Able-Volterra convolution equations of the first and second kinds by a fractional linear multistep method and the fast Fourier transform techniques to exploit the convolution structure. Frankel [84] discussed a classical nonlinear heat transfer problem using an equivalent integral formulation, in which the two numerical methods were used for solving the nonlinear singular Volterra integral equations of the first kind, including the global and local uses of singularity subreaction and product integration. These methods effectively overcame the numerical difficulties associated with the weakly singular kernel. A numerical procedure based on the trapezoidal integration and Newton-Raphson method was also reported to produce accurate results rapidly.

Tang [85] used a class of spline collocation methods for first-order Volterra integro-differential equations that contain a weakly singular kernel $(t-s)^{-\alpha}$, in which the superconvergence properties may be obtained by appropriate collocation parameters and graded meshes, and the grading exponents of the graded meshes used were not larger than a polynomial degree independent of α . This was in contrast to the theories of the spline collocation methods for Volterra and Fredholm integral equations of the second kind.

Jumarhon and McKee [86] used the product integration methods for solving a system of Volterra integral equations with a weakly singular kernel, in which the technique of removing singularities was used to derive the explicit and implicit product Euler schemes

with first-order convergence, and a product trapezoidal scheme with second-order convergence.

Orsi [87] introduced a numerical approach for solving Volterra integral equations of the second kind when the kernel contains a mild singularity. Lima and Diogo [88] considered the Volterra integral equation with a weakly singular kernel, and proved the existence of an asymptotic error expansion for the Euler method by introducing appropriate function spaces, such that an extrapolation method was developed for this type of integral equations.

Wazwaz and Khuri [89] developed a reliable technique for solving the second kind of the weakly singular Volterra integral equations, in which the Adomian decomposition method was discussed and the phenomenon of the self-canceling “noise” terms was considered, and the solution obtained was in the form of a convergent power series with computable terms. In addition, there are several other works for study of the weakly singular Volterra integral equations [90-98].

2.2.4 Integro-differential equations

One of the popular methods for solving the integro-differential equations is the Laplace decomposition method (LDM) [99], in which the Laplace transformation is used on both sides of the integro-differential equations, and the solution was assumed in the form of an infinite series. Avudainayagam and Vani [100] proposed the Wavelet-Galerkin method (WGM) for integro-differential equations, where the wavelet series and the Galerkin discretization scheme are employed to approximate the solution and to discretize the system of equations, respectively. He [101] developed the Homotopy

perturbation method (HPM), which couples a homotopy technique with a perturbation approach for solving the integro-differential equations [102]. In addition, there are several other methods that have been proposed for solving the integro-differential equations [64, 93, 103, 104].

2.2.5 Nonlinear integral equations

In general, there are three kinds of nonlinear integral equations, namely nonlinear Fredholm integral equations, the Hammerstein type of nonlinear integral equations, and nonlinear Volterra integral equations. Several research groups put much effort in this area, and they are reviewed below.

For the nonlinear Fredholm integral equations, Alipanah and Dehghan [105] developed a numerical method, based on the interpolation by the radial basis functions (RBF) for approximation. Biazar and Ghazvini applied the homotopy perturbation method (HPM) to solve the Fredholm type of the nonlinear integral equations [106]. Ghorbani and Saberi-Nadjafi modified HPM to solve the nonlinear integral equations [107].

For the Hammerstein type of the nonlinear integral equations with discontinuous kernel, Abdou *et al.* [108] discussed the existence and uniqueness of the solution, and used the Toeplitz matrix method to obtain a nonlinear system of algebraic equations. Another numerical method has been developed for solving the Fredholm-Volterra Hammerstein integral equations [109], where the unknown function is replaced by the truncated series of well-known Chebyshev expansion of functions. Hernandez *et al.* [110] proposed a one-parameter family of secant-type iterations for solving nonlinear

equations in Banach space, in which a semilocal convergence result is established for these iterations by means of a technique based on a new system of recurrence relations. The result was subsequently used to solve the Hammerstein type of the nonlinear integral equations. Maleknejad and Derili [111] combined a general kernel scheme based on Daubechies wavelets with a collocation method to solve the nonlinear integral equation. Later on they also proposed another method which combines the spline collocation method with Lagrange interpolation [112].

For the nonlinear Volterra integral equations, Katani and Shahmorad [113] developed a block-by-block method, in which the special starting procedure is avoided and at least fourth-order convergence was achieved. Javadi *et al.* [114] implemented the Adomian decomposition method for the second kind of the nonlinear Volterra integral equations. Capobianco *et al.* [115] applied a fast implicit and explicit Runge-Kutta methods for a system of Hammerstein type of Volterra integral equations. El-Kalla [116] deducted a reliable approach for analysis of the convergence of the Adomian method when it was used to solve a class of nonlinear Volterra integral equations.

Further literature on specific topics of interest will be reviewed in relative chapters. For example, the review of the generalized integral quadrature (GIQ) method is presented in Section 3.2.2.1, and the review of peridynamics is stated in Section 5.1.

Although there are plentiful meshless methods reviewed in Section 2.1, they are mostly proposed to solve the differential equations. The GIQ technique is one of the meshless techniques for solving integral equations. However, it always requires field nodes lying along a straight line, which restraint the range of application of this method. Thus, development of the RIQ method is of great significance.

Chapter 3 Development of Random Integral Quadrature (RIQ) Method

3.1 Introduction

In this chapter, a novel meshless method is developed, and it is called here the random integral quadrature (RIQ) method, for solution of various integral equations. The main idea of the RIQ method is to make the function approximated over a set of field nodes that are scattered randomly or uniformly within the domain as well as along the boundary. In the RIQ method, the GIQ technique is used first to discretize the integral terms in the governing integral equations over a set of virtual nodes, and then the Kriging interpolation function is employed to approximate the function values at the virtual nodes over a set of the field nodes. The system of linear algebraic equations is then constructed, and the values of the approximated functions at the field nodes are finally obtained numerically. [Equation Chapter 3 Section 1](#)

Before representing the formulation of the RIQ method, several integral quadrature techniques are introduced. Let the domain $[a, b]$ be finite and divided by N uniformly distributed points. A quadrature rule is then defined as

$$\int_a^b \phi(x) dx = \sum_{i=1}^N w_i \phi(x_i) \quad (3.1)$$

where w_i ($i=1, 2, \dots, N$) are the weighing coefficients, and x_i are the quadrature points.

A number of quadrature rules are derived to calculate the integration. Several frequently-used quadrature rules are given as below.

Rectangle rule:

$$\int_a^b \phi(x) dx = \frac{b-a}{N} \sum_{i=1}^N \phi(x_i) \quad (3.2)$$

Trapezoidal rule:

$$\int_a^b \phi(x) dx = \frac{b-a}{N} \sum_{i=2}^N \phi(x_i) + \frac{b-a}{2N} (\phi(a) + \phi(b)) \quad (3.3)$$

Simpson's rule:

$$\begin{aligned} \int_a^b \phi(x) dx = \frac{b-a}{3N} [& \phi(x_1) + 4\phi(x_2) + 2\phi(x_3) + 4\phi(x_4) \\ & + 2\phi(x_5) + \dots + 2\phi(x_{N-1}) + 4\phi(x_N) + \phi(x_{N+1})] \end{aligned} \quad (3.4)$$

Newton-Cotes rule:

$$\int_a^b \phi(x) dx = (b-a) \sum_{i=1}^N w_i \phi(x_i) \quad (3.5)$$

where $\sum_{i=1}^N w_i = 1$. The values of the weighting coefficients w_i are known [15].

Gaussian-Legendre Quadrature:

$$\int_a^b \phi(x) dx = \sum_{i=1}^{N+1} w_i \phi(x_i) \quad (3.6)$$

where $x_i = (a+b+(b-a)\xi_i)/2$, ξ_i and w_i ($i=1, 2, \dots, N$) are the Gaussian points and the weight coefficients, respectively.

Chebyshev rule:

$$\int_a^b \phi(x) dx = \frac{b-a}{N} \sum_{i=1}^N \phi(x_i) \quad (3.7)$$

where $x_i = (b-a)/2 + (b-a) \times \xi_i / 2$, and ξ_i ($i=1, 2, \dots, N$) are the Chebyshev points.

3.2 Random Integral Quadrature (RIQ) method

3.2.1 Brief background

As mentioned in Section 1.4, the motivation of developing the RIQ method is to overcome the limitation of the GIQ method, which requires all the field nodes being collinear. In order to achieve this, a group of background virtual nodes are created first in a linear manner, and then each of the virtual nodes is interpolated by the surrounding field nodes using the Kriging interpolation function. As a result, there are two key steps in the RIQ method. First, the governing integral equation is discretized at each virtual node with the GIQ technique, resulting in an initial system of linear algebraic equations. Second, the function value at each virtual node is approximated further using the Kriging interpolation technique over the field nodes within the whole domain. In other words, an algebraic equation is written at each virtual node first. By substituting the approximation into the initial system of equations over the virtual nodes, the final algebraic system is obtained over the field nodes. However, it should be remembered that the row size of the final system matrix is equal to the number of the virtual nodes in the domain, while the column size equal to the total number of the field nodes in the domain. Hence, a rectangular matrix of algebraic system is formed, which is then converted mathematically into a square matrix form, for which there are many solvers available,

including direct and indirect solvers, such as LU decomposing and iterative methods. After solving the final system of algebraic equations, the function value at each field node is finally obtained.

It should also be pointed out that the RIQ method uses two sets of nodes, the virtual and field nodes. As shown in Figure 3.1, the small circles denote the virtual nodes, and the solid dots indicate the field nodes. The virtual nodes are required to lie in a straight line by a cosine distribution for discretization of governing integral equations, and the field nodes are scattered either uniformly or randomly for interpolation of function values at virtual nodes.

3.2.2 Formulation of the RIQ method

In order to solve an integral equation, the most important implementation is to express the integral term approximately as the sum of the basic functions or function values at the nodes. In the RIQ method, the integral equation is approximated by the summation of the function values at the field nodes with weighting coefficients to control the proportion of each node. The RIQ method is to discretize the governing integral equations over the virtual nodes using the GIQ technique, and then to interpolate the approximation at each virtual node by the field nodes. We can also choose other quadrature methods to discretize the domain, such as Gaussian quadrature. However, there are some disadvantages in that technique. One of them is that the coordinates obtained may fall outside the region of interest, such that the coordinates should be checked before numerical integration is performed. Another drawback of Gaussian

quadrature is that the coordinates cannot be chosen arbitrarily. Therefore, the GIQ technique is chosen here to discretize the governing integral equations.

3.2.2.1 Discretization of integral governing equation

Let us define a one-dimensional integral domain as Ω , which is discretized with N field nodes, i.e., $x_1, x_2, \dots, x_N$. These field nodes can be distributed uniformly or randomly, according to the situation. As stated in Section 3.2.1, a set of the virtual nodes is created, i.e., $x^1, x^2, \dots, x^M$. After the virtual nodes have been created, the integral term can be discretized using the GIQ technique. If the whole integral domain Ω is decomposed into n sub-domains, the integral over the sub-domain Ω_i is discretized as follows,

$$\int_{\Omega_i} u(x) dx = \sum_{k=1}^M w_{ik} u(x^k) \quad (3.8)$$

where $k = 1, 2, \dots, M$, and $i = 1, 2, \dots, n$. $u(x)$ is the integral function, w_{ik} the weighting coefficients of the GIQ technique. The schematic diagram is shown in Figure 3.2. After the integral term over each sub-domain is determined, they can be summed together, and thus the integral over the whole domain is given by

$$\begin{aligned} \int_{\Omega} u(x) dx &= \int_{\Omega_1} u(x) dx + \int_{\Omega_2} u(x) dx + \dots + \int_{\Omega_n} u(x) dx \\ &= \sum_{i=1}^n \sum_{k=1}^M w_{ik} u(x^k) \end{aligned} \quad (3.9)$$

According to the GIQ method [117], the details of the calculation of the weighting coefficients are written as follows. The GIQ method was developed by following the same concept in the polynomial-based differential quadrature (PDQ) [117]. If the

function is sufficiently smooth in the entire domain of a problem, it can be approximated by a high-order polynomial or Fourier series. A polynomial-based approximation is usually adopted for simplicity and high accuracy. High-order polynomials should involve all the functional values in the whole domain. Hence, the integration of the function over the sub-domain can be calculated by integrating the approximated polynomial. The key step of this approach is to obtain the weighting coefficients. It is not easy to determine the weighting coefficients in integral quadrature. However, this can easily be achieved by first-order derivative in PDQ.

It is assumed that a function $\phi(x)$ is sufficiently continuous over the domain $[a, b]$. As stated before, the set of the virtual nodes $x^1, x^2, \dots, x^M$ are created, which can be distributed by the cosine scheme as follows,

$$x^i = \left(\frac{1 - \cos\left(\frac{i-1}{M-1}\pi\right)}{2} \right) \times (b-a) \quad (3.10)$$

where M is the number of the total virtual nodes in the domain and $i = 1, 2, \dots, M$. The function $\phi(x)$ is approximated by a $(M-1)$ th-order polynomial. In particular, when the functional values at the M grid points are known, $\phi(x)$ can be approximated by Lagrange interpolation polynomials that are related to the node values at all the virtual nodes. Hence, the integral of this approximated polynomial over a sub-domain $[x^i, x^j]$ may involve the function values over the whole domain. The integral of $\phi(x)$ over a sub-domain is then approximated as

$$\int_{x^i}^{x^j} \phi(x) dx = \sum_{k=1}^M w_k^{ij} \phi(x^k) \quad (3.11)$$

where x^i and x^j are the coordinates of the bounds of the integral, which can be altered when necessities. w_k^{ij} are the weighting coefficients, where $k=1, 2, \dots, M$. If the Lagrange interpolation polynomials, $r_k(x)$, are chosen as the basis polynomials, w_k^{ij} can be determined by

$$w_k^{ij} = \int_{x^i}^{x^j} r_k(x) dx \quad (3.12)$$

It is difficult to calculate w_k^{ij} from Eq. (3.12) directly. However, w_k^{ij} can be determined by the weighting coefficients of the first-order derivative in PDQ, if we define

$$\phi(x) \equiv \frac{du(x)}{dx} \quad (3.13)$$

It is noted that, if $\phi(x)$ is a $(M-1)$ th-order polynomial, $u(x)$ is an M th-order polynomial. Here $\phi(x)$ is approximated by

$$\phi(x) = a_0 + a_1 x + \dots + a_{M-1} x^{M-1} \quad (3.14)$$

where $a_0, a_1, \dots, a_{M-1}$ are constants. Integrating Eq. (3.14) from a constant c to x gives

$$u(x) = \int_c^x \phi(t) dt + u(c) = \psi(x, c) + u(c) \quad (3.15)$$

where

$$\psi(x, c) = x \left(a_0 + \frac{a_1}{2} x + \dots + \frac{a_{M-1}}{M} x^{M-1} \right) - c \left(a_0 + \frac{a_1}{2} c + \dots + \frac{a_{M-1}}{M} c^{M-1} \right) \quad (3.16)$$

It is known that $\psi(x, c)$ constitutes an M -dimensional linear polynomial vector space.

Let us choose a set of basis polynomials as

$$p_k(x) = (x - c)r_k(x) \quad k = 1, 2, \dots, M \quad (3.17)$$

where $r_k(x)$ is Lagrange interpolation polynomial. According to the PDQ [23],

$$\psi_x(x^i, c) = \sum_{j=1}^M a_{ij} \psi(x^j, c) \quad \text{for } i = 1, 2, \dots, M \quad (3.18)$$

where a_{ij} are the weighting coefficients for the PDQ method. Using the same method as in the PDQ method, we substitute Eq. (3.17) into Eq. (3.18), and then have

$$\begin{aligned} a_{ij} &= \frac{x^i - c}{x^j - c} b_{ij}, \quad \text{when } j \neq i \\ a_{ii} &= b_{ii} + \frac{1}{x^i - c} \end{aligned} \quad (3.19)$$

where b_{ij} are the weighting coefficients of the first-order derivative in the PDQ method, which can be expressed as

$$\begin{aligned} b_{ij} &= \frac{1}{x^j - x^i} \prod_{k=1, k \neq i, j}^M \left(\frac{x^i - x^k}{x^j - x^k} \right), \quad \text{when } j \neq i \\ b_{ii} &= - \sum_{j=1, j \neq i}^M b_{ij}^{(1)} \end{aligned} \quad (3.20)$$

Thus, the weighting coefficients in Eq. (3.18) are obtained from

$$\begin{aligned} a_{ij} &= \frac{x^i - c}{x^j - c} \times \frac{1}{x^j - x^i} \prod_{k=1, k \neq i, j}^M \left(\frac{x^i - x^k}{x^j - x^k} \right), \quad \text{when } j \neq i \\ a_{ii} &= - \sum_{j=1, j \neq i}^M w_{ij}^{(1)} + \frac{1}{x^i - c} \end{aligned} \quad (3.21)$$

In the above equations, c cannot be chosen as the coordinates of the virtual nodes x^i .

Eq. (3.18) can be written in matrix form,

$$[\psi_x] = [A][\psi] \quad (3.22)$$

where

$$\begin{aligned} [\psi] &= [\psi(x^1, c), \psi(x^2, c), \dots, \psi(x^M, c)]^T \\ [\psi_x] &= [\phi] = [\psi_x(x^1, c), \psi_x(x^2, c), \dots, \psi_x(x^M, c)]^T \end{aligned} \quad (3.23)$$

Substitute Eqs. (3.13) and (3.15) into Eq. (3.22) gives

$$[\phi] = [A][\psi] \quad (3.24)$$

By setting $[W] = [A]^{-1}$, Eq. (3.24) is rewritten as

$$\int_c^{x^i} \phi(x) dx = \sum_{k=1}^M w_{ik} \phi(x^k), \quad \text{for } i = 1, 2, \dots, M \quad (3.25)$$

As a result,

$$\int_{x^i}^{x^j} \phi(x) dx = \sum_{k=1}^M (w_{jk} - w_{ik}) \phi(x^k) \quad (3.26)$$

The weighting coefficients of the GIQ technique are thus given as follows,

$$w_k^{ij} = w_{jk} - w_{ik} \quad (3.27)$$

After the weighting coefficients are determined, the bridge is established between the integral term in the governing integral equation and the function values at the discrete virtual nodes.

The same approach can be extended to multi-dimensional cases. A two-dimensional case is considered as follows. Let the whole domain be $a \leq x \leq b$, $c \leq y \leq d$, which is decomposed with MX virtual nodes in the x -direction and MY virtual nodes in the y -direction. Let $\phi(x, y)$ be continuous in the entire domain, and consider an integral of

$\phi(x, y)$ over a sub-domain. By the same technique as the one-dimensional case, the formulation of the two-dimensional integral of $\phi(x, y)$ over the sub-domain, $x^i \leq x \leq x^j$ and $y^l \leq y \leq y^m$, is given as

$$\int_{x^i}^{x^j} \int_{y^l}^{y^m} \phi(x, y) dx dy = \sum_{n=1}^{MX} \sum_{k=1}^{MY} (w_{jn} - w_{in})(w_{mk} - w_{lk}) \phi(x^n, y^k) \quad (3.28)$$

where w_{jn} and w_{mk} are the weighting coefficients of the one-dimensional integral in the x - and y - directions respectively, and are given by

$$\begin{aligned} w_{jn} - w_{in} &= \int_{x^i}^{x^j} r_n(x) dx \quad i, j, n = 1, 2, \dots, MX \\ w_{mk} - w_{lk} &= \int_{y^l}^{y^m} s_k(y) dy \quad l, m, k = 1, 2, \dots, MY \end{aligned} \quad (3.29)$$

where $r_n(x)$ and $s_k(y)$ are the Lagrange interpolation polynomials in the x - and y - directions. As in the one-dimensional case, w_{jn} and w_{mk} can be determined by the PDQ approach.

Similarly, for the three-dimensional case, let the function $\phi(x, y, z)$ be continuous in the domain $a \leq x \leq b$, $c \leq y \leq d$ and $g \leq z \leq h$, which can be discretized with MX virtual nodes in the x -direction, MY virtual nodes in the y -direction, and MZ virtual nodes in the z -direction. The integral of $\phi(x, y, z)$ over the sub-domain, $x^i \leq x \leq x^j$, $y^l \leq y \leq y^m$ and $z_e \leq z \leq z_q$, can be approximated as

$$\begin{aligned} \int_{x^i}^{x^j} \int_{y^l}^{y^m} \int_{z_e}^{z_q} \phi(x, y, z) dx dy dz = \\ \sum_{n=1}^{MX} \sum_{k=1}^{MY} \sum_{p=1}^{MZ} (w_{jn} - w_{in})(w_{mk} - w_{lk})(w_{qp} - w_{ep}) \phi(x^n, y^k, z^p) \end{aligned} \quad (3.30)$$

where w_{qp} are the weighting coefficients of the one-dimensional integral in the z -direction, given by

$$w_{qp} - w_{ep} = \int_{z^e}^{z^q} t_p(z) dz \quad e, p, q = 1, 2, \dots, MZ \quad (3.31)$$

where $t_p(z)$ is the Lagrange interpolation polynomial in the z -direction.

So far the integral terms have been discretized over the virtual nodes. In order to express the integral terms by the summation of the approximate function values over the field nodes, the approximated expression at each virtual node is interpolated using the Kriging interpolation function over the field nodes.

3.2.2.2 Kriging interpolation

In the RIQ method, the Kriging function [32] is combined with the GIQ technique for interpolation. The Kriging interpolation technique is a generalized linear regression used to formulate an optimal estimator in a minimum mean square error sense, which was introduced by Krige for moving averages to avoid systematic errors in interpolation [32], with several advantages. The first is to compensate for the effects of data clustering, assigning individual points within a cluster less weight than the isolated data points. The second is to provide the estimation error along with an estimate of the variable itself. The third is that the availability of estimation error provides the basis for stochastic simulation of possible realizations of an estimator. So far the Kriging technique has become a fundamental tool in geostatistics and other fields. Due to the rigorous theoretical basis, the Kriging method can ensure the theoretical accuracy of the interpolation. In addition, the shape functions constructed have the delta properties as

they are a passing node interpolation, such that the boundary conditions can be easily imposed. Moreover, the computing algorithm of the Kriging function is quite simple, such that it is good to be used as an interpolation function.

Let $u(\mathbf{x})$ be an arbitrary function that is defined in the domain Ω discretized by a set of scattered field nodes $\mathbf{x}_i$ ($1 \leq i \leq N$), where N is the number of the total field nodes in the domain Ω . As per Kriging interpolation, the function approximation at each virtual node is given over all the field nodes by,

$$u^h(\mathbf{x}) = \sum_{i=1}^N \lambda_i u(\mathbf{x}_i) \quad (3.32)$$

where $u(\mathbf{x}_i)$ is the function value at the corresponding field node $\mathbf{x}_i$, λ_i is the weighting coefficient related to the neighborhood nodes, and determined by minimizing the squared variance of the estimation error $E \left\{ \left[u(\mathbf{x}) - u^h(\mathbf{x}) \right]^2 \right\}$.

If there is no bias in the estimation formula of Eq. (3.32), the expected values of $u(\mathbf{x})$ and $u^h(\mathbf{x})$ must be equal to each other,

$$E[u(\mathbf{x})] = E[u^h(\mathbf{x})] = \sum_{i=1}^N \lambda_i E[u(\mathbf{x}_i)] \quad (3.33)$$

The random field function $u(\mathbf{x})$ is decomposed into two parts,

$$u(\mathbf{x}) = R(\mathbf{x}) + m(\mathbf{x}) \quad (3.34)$$

where $R(\mathbf{x})$ is treated as a term with a stationary mean of zero and a stationary covariance with a function of lag h , i.e.,

$$E[R(\mathbf{x})] = 0 \quad (3.35)$$

$$\text{Cov}[R(\mathbf{x}), R(\mathbf{x}+\mathbf{h})] = E[R(\mathbf{x}) \cdot R(\mathbf{x}+\mathbf{h})] \quad (3.36)$$

$m(\mathbf{x}) = E[u(\mathbf{x})]$ is called the drift. The expected value of $u(\mathbf{x})$ is described as

$$m(\mathbf{x}) = \sum_{i=1}^n \lambda_i m(\mathbf{x}_i) \quad (3.37)$$

It is assumed here that the drift belongs to the linear subspace S , and is taken as a linear polynomial, Eq. (3.37) thus becomes

$$p_k(\mathbf{x}) = \sum_{i=1}^n \lambda_i p_k(\mathbf{x}_i) \quad 1 \leq k \leq l \quad (3.38)$$

where p is the basis in S . In one-dimensional space, the linear basis is given by

$$\mathbf{p}^T(\mathbf{x}) = [1 \quad x], \quad l=2 \quad (3.39)$$

In two-dimensional space, it is written as

$$\mathbf{p}^T(\mathbf{x}) = [1 \quad x \quad y], \quad l=3 \quad (3.40)$$

The squared variance of the estimation error can be computed by

$$\begin{aligned} E \left\{ \left[u(\mathbf{x}) - \sum_{i=1}^N \lambda_i u(\mathbf{x}_i) \right]^2 \right\} &= E[u(\mathbf{x})]^2 - 2 \sum_{i=1}^N \lambda_i E[u(\mathbf{x})u(\mathbf{x}_i)] \\ &\quad + \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j E[u(\mathbf{x}_i)u(\mathbf{x}_j)] \end{aligned} \quad (3.41)$$

By minimizing Eq. (3.41) with respect to the coefficients λ_i under l linear constraints that satisfy the no bias conditions, the solution is characterized by a linear system of $(N + l)$ equations related to unknowns $\lambda_1, \lambda_2, \dots, \lambda_N$ and $\mu_1, \mu_2, \dots, \mu_l$,

$$\begin{aligned} \sum_{j=1}^N \lambda_j E \left[u(\mathbf{x}_i) u(\mathbf{x}_j) \right] + \sum_{k=1}^l \mu_k p_k(\mathbf{x}_i) &= E \left[u(\mathbf{x}) u(\mathbf{x}_i) \right] \\ \sum_{j=1}^N \lambda_j p_k(\mathbf{x}_j) &= p_k(\mathbf{x}) \end{aligned} \quad (3.42)$$

where coefficients μ_k are the Lagrange multipliers associated with the constraints.

The covariance $E \left[u(\mathbf{x}) u(\mathbf{x}_i) \right]$ is replaced with the semivariogram $\gamma(h)$, where h is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_i$ and the Gaussian semivariogram model is employed as [32]

$$\gamma(h) = c_0 \left(1 - e^{-3 \left(\frac{\|h\|}{a_0} \right)^2} \right) \quad (3.43)$$

where c_0 is the sill representing the average variance of points at such a distance away from the point considered that there is no influence between the points. a_0 is the range representing the distance, at which there is no longer a correlation between the points, and it is taken as

$$a_0 = \alpha \cdot \beta \cdot d_{\min} \quad (3.44)$$

where α is a coefficient, β is a scaling parameter, and $d_{\min}$ is the shortest distance between the interpolation point and the neighboring points.

Thus, the Kriging system (3.42) is written in matrix form as

$$\mathbf{A}\mathbf{Y} = \mathbf{b} \quad (3.45)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \gamma(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \gamma(\mathbf{x}_1, \mathbf{x}_N) & p_1(\mathbf{x}_1) & \cdots & p_l(\mathbf{x}_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \gamma(\mathbf{x}_N, \mathbf{x}_1) & \cdots & \gamma(\mathbf{x}_N, \mathbf{x}_N) & p_1(\mathbf{x}_N) & \cdots & p_l(\mathbf{x}_N) \\ p_1(\mathbf{x}_1) & \cdots & p_1(\mathbf{x}_N) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p_l(\mathbf{x}_1) & \cdots & p_l(\mathbf{x}_N) & 0 & \cdots & 0 \end{bmatrix} \quad (3.46)$$

$$\mathbf{Y} = [\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_N \quad \mu_1 \quad \mu_2 \quad \cdots \quad \mu_N]^T \quad (3.47)$$

$$\mathbf{b} = [\gamma(\mathbf{x}) \quad \mathbf{p}(\mathbf{x})] = [\gamma(\mathbf{x}, \mathbf{x}_1) \quad \cdots \quad \gamma(\mathbf{x}, \mathbf{x}_N) \quad p_1(\mathbf{x}) \quad \cdots \quad p_l(\mathbf{x})]^T \quad (3.48)$$

By solving this system of linear equations, one can obtain λ_i . Substituting them into Eq. (3.32), the estimating value is obtained as

$$u^h(\mathbf{x}) = \phi(\mathbf{x})\mathbf{u} \quad (3.49)$$

where $\mathbf{u} = [\mathbf{u}(\mathbf{x}_1) \quad \mathbf{u}(\mathbf{x}_2) \quad \cdots \quad \mathbf{u}(\mathbf{x}_N)]^T$, $\phi(\mathbf{x})$ is the shape function matrix, and is calculated by

$$\begin{aligned} \phi(\mathbf{x}) &= \gamma^T \mathbf{S} + \mathbf{p}^T \mathbf{Q}, \\ \mathbf{S} &= \mathbf{R}^{-1} (\mathbf{I} - \mathbf{PQ}), \\ \mathbf{Q} &= (\mathbf{P}^T \mathbf{R}^{-1} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{R}^{-1} \end{aligned} \quad (3.50)$$

So far the Kriging interpolation has been formulated. It is seen that the commutating algorithm is simpler, compared with other methods, such as the moving least square method.

By the Kriging interpolation technique, the approximate function value over the k th virtual node is estimated as

$$u^h(\mathbf{x}^k) = \phi_{kp}(\mathbf{x})\mathbf{u}_p(\mathbf{x}) \quad (3.51)$$

where $u^h(\mathbf{x}^k)$ is the function approximation over the k th virtual node, and $u_p(\mathbf{x}) = [u_1(\mathbf{x}), u_2(\mathbf{x}), \dots, u_N(\mathbf{x})]$, which is, the function value over the field nodes. N is the number of the field nodes, $\phi_{kp}(\mathbf{x})$ is the shape function vector for the k th virtual node.

3.2.2.3 Finalization of discrete system

For construction of the shape functions, the matrix $\mathbf{Y}$ in Eq. (3.47) is computed first, and then the matrix $\mathbf{S}$ is calculated. The final step is that the shape function vector ϕ_k is filled up. It is found from Eq. (3.50) that the size of the matrix $\mathbf{Y}$ is $l \times N$ and that of $\mathbf{S}$ is $N \times N$, such that a $1 \times N$ shape function vector is obtained. Substituting Eq. (3.51) to Eq. (3.9) and writing the results in the matrix form, the RIQ formula is obtained,

$$\int_{\Omega} u(\mathbf{x}) d\mathbf{x} = [w_1, w_2, \dots, w_M]_{1 \times M} \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \cdots & \vdots & \phi_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ \phi_{M1} & \cdots & \phi_{M(N-1)} & \phi_{MN} \end{bmatrix}_{M \times N} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}_{N \times 1} \quad (3.52)$$

where w_k ($k = 1, 2, \dots, M$) are the GIQ weighting coefficients, and $w_k = \sum_{i=1}^M w_{ik}$.

After the integral term is expressed in the form of a summation of the approximation function values over the field nodes in the domain, the governing equation is discretized to a system of algebraic equations which can be easily solved.

3.2.2.4 Solution of the final system

It is obviously observed from the above section that the number of equations in the final discrete system is equal to the number of the virtual nodes M , while the number of

the unknown function values is equal to the number of the field nodes N . In order to obtain the unique and good numerical solution of this system, it is suggested that the number of the virtual nodes scattered is larger than the number of the field nodes. The least square approximation is employed here to solve the final system. Finally the approximate function values of the over-determined system are obtained. The solution of this algebraic system is the approximate function values over the field nodes due to the delta property of the Kriging interpolation.

3.2.2.5 Implementation procedures of the RIQ method

In this section, the implementation procedure of the RIQ method is described in detail, by taking the following Fredholm integral equation as an example,

$$u(x) = g(x) + \lambda \int_a^b k(x, t) u(t) dt \quad a \leq x, t \leq b \quad (3.53)$$

where $g(x)$ and $k(x, t)$ are known functions, λ is a constant, a and b are the bounds of the domain, $u(x)$ is the unknown function. The implementation of the RIQ method is described as detailed below.

Step 1: Arrangement of scattering field nodes

A set of field nodes is created in the computational domain, by either uniform or random distribution. As the field nodes are used to interpolate the approximation over the virtual nodes, it doesn't matter what fashion the nodes exist. Let the number of field nodes be N , the set of the field nodes are $x_i = x_1, x_2, \dots, x_N$.

Step 2: Arrangement of scattering virtual nodes

The virtual nodes x^j ($j = 1, 2, \dots, M$) are created in a straight linear manner, for example, using a cosine distribution. The governing equations are discretized over the virtual nodes using the GIQ technique, thus the virtual nodes should lie in a linear,

$$x^j = \left(\frac{1 - \cos\left(\frac{j-1}{M-1}\pi\right)}{2} \right) \times L \quad (3.54)$$

where $L = b - a$, M is the number of the total virtual nodes. Usually the number of the virtual nodes is suggested to be larger than the number of field nodes.

Step 3: Discretization of governing integral equation

The governing equation is discretized as

$$u(x^j) = g(x^j) + \lambda \sum_{k=1}^M w_j K(x^j, t^k) u(x^k) \quad (3.55)$$

where x^j , x^k and t^k are the virtual nodes. w_j is the GIQ weighting coefficient for the j th virtual node. In order to make the interpolation more convenient, the above equation (3.55) is rewritten in the following matrix form,

$$I - \lambda \begin{bmatrix} w_1 K(x^1, t^1) & w_1 K(x^1, t^2) & \dots & w_1 K(x^1, t^M) \\ w_2 K(x^2, t^1) & \dots & \ddots & \vdots \\ \vdots & \vdots & \ddots & w_{M-1} K(x^{M-1}, t^M) \\ w_M K(x^M, t^1) & \dots & w_M K(x^M, t^{M-1}) & w_M K(x^M, t^M) \end{bmatrix} u(x^j) = g(x^j) \quad (3.56)$$

In this step, the GIQ weighting coefficients are computed first as follows.

$$w_j = W_{Mj} - W_{1j} \quad (3.57)$$

where

$$[W] = [A]^{-1} \quad (3.58)$$

In Eq. (3.58), $[A]$ is a $M \times M$ matrix and its elements a_{ij} are calculated by

$$\begin{aligned} a_{ij} &= \frac{x^i - c}{x^j - c} \times b_{ij}, \quad \text{when } i \neq j \\ a_{ii} &= b_{ii} + \frac{1}{x^i - c} \end{aligned} \quad (3.59)$$

where c is a constant excluding the coordinates of the virtual nodes, and b_{ij} is the weighting coefficient of the first-order derivative in the PDQ technique [23] and is expressed as

$$\begin{aligned} b_{ij} &= \frac{1}{x^j - x^i} \prod_{k=1, k \neq i, j}^M \left(\frac{x^i - x^k}{x^j - x^k} \right), \quad \text{when } i \neq j \\ b_{ii} &= - \sum_{j=1, j \neq i}^M b_{ij} \end{aligned} \quad (3.60)$$

As a result,

$$\begin{aligned} a_{ij} &= \frac{x^i - c}{x^j - c} \times \frac{1}{x^j - x^i} \prod_{k=1, k \neq i, j}^M \left(\frac{x^i - x^k}{x^j - x^k} \right), \quad \text{when } i \neq j \\ a_{ii} &= - \sum_{j=1, j \neq i}^M \frac{1}{x^j - x^i} \prod_{k=1, k \neq i, j}^M \left(\frac{x^i - x^k}{x^j - x^k} \right) + \frac{1}{x^i - c} \end{aligned} \quad (3.61)$$

After the weighting coefficients are determined, the kernel function matrix $\mathbf{K}$ and the right-hand side vector $\mathbf{G}$ are formulated using the coordinates of the virtual nodes.

Thus Eq. (3.56) is simplified to

$$\mathbf{J}\mathbf{U} = \mathbf{G} \quad (3.62)$$

where $\mathbf{J}$ is a $M \times M$ matrix, $\mathbf{U}$ and $\mathbf{G}$ are $M \times 1$ vectors, i.e.,

$$\begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1M} \\ J_{21} & \cdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & J_{(M-1)M} \\ J_{M1} & \cdots & J_{M(M-1)} & J_{MM} \end{bmatrix} \begin{bmatrix} u(x^1) \\ u(x^2) \\ \vdots \\ u(x^M) \end{bmatrix} = \begin{bmatrix} g(x^1) \\ g(x^2) \\ \vdots \\ g(x^M) \end{bmatrix} \quad (3.63)$$

Step 4: Interpolation of the virtual nodes

The approximation function values over the virtual nodes are interpolated by the Kriging technique over the field nodes in the whole domain,

$$u^h(x^j) = \sum_{i=1}^N \phi_j(x_i) u(x_i) \quad (3.64)$$

where $u^h(x^j)$ is the approximation of the function value at the virtual node x^j . $\phi_j(x_i)$ is the shape function at the i th field node, and $u(x_i)$ is the function value at the i th field nodes. By Eq. (3.50), the shape function matrix is obtained. Eq. (3.64) is thus written in matrix form as

$$\begin{bmatrix} u^h(x^1) \\ u^h(x^2) \\ \vdots \\ u^h(x^M) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \cdots & \phi_1(x_N) \\ \phi_2(x_1) & \cdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \phi_{M-1}(x_N) \\ \phi_M(x_1) & \cdots & \phi_M(x_{N-1}) & \phi_M(x_N) \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_N) \end{bmatrix} \quad (3.65)$$

Step 5: Finalization of discrete system

The final system of the algebraic equations is obtained by substituting Eq. (3.65) into Eq. (3.63), and is written as

$$\mathbf{CU} = \mathbf{G} \quad (3.66)$$

It is noted that the size of the matrix $\mathbf{J}$ is $M \times M$, and that of the shape function matrix is $M \times N$. As a result, the size of the final coefficient matrix $\mathbf{C}$ is $M \times N$, and here $\mathbf{U} = [u(x_1), u(x_2), \dots, u(x_N)]$.

Step 6: Computation of the algebraic system

Various techniques can be used to solve the system of linear algebraic equations, for example, the LU decomposition, and iterative methods. As mentioned in Section 3.2.2.4, the least-square approximation is used here to solve Eq. (3.66), by which the rectangular matrix $\mathbf{C}$ is converted into a square matrix by multiplying $\mathbf{C}^T$ on both sides of the system. As such, the residual error $\mathbf{R} = \mathbf{CU} - \mathbf{G}$ is minimized. After solving the system of equations, the approximate function at the field nodes is determined.

Step 7: Estimation of the convergence rate

The order of the convergence rate is referred to the order of error for discretization methods [118], which is written as

$$\varepsilon = O(h^p) \quad (3.67)$$

where ε is the error, h is the nodal spacing, and p is the order of error, i.e., order of the rate of convergence. Using log function on both sides results in

$$\ln(\varepsilon) = p \ln(h) \quad (3.68)$$

and then,

$$p = \frac{\ln(\varepsilon)}{\ln(h)} \quad (3.69)$$

Here the global error is employed to calculate the convergence rate, which is written as [35],

$$\varepsilon = \frac{1}{|u|_{\max}} \sqrt{\frac{1}{N} \sum_{i=1}^N (u_i - u_i^h)^2} \quad (3.70)$$

where ε is defined as the global error for the numerical solution, N the number of the total field nodes. u_i and u_i^h are the analytical exact solution and the numerical solution at the i^{th} field node respectively.

3.2.3 Stability analysis

During the computation, it is noted that the transformed moment matrix $\mathbf{P}^T \mathbf{R}^{-1} \mathbf{P}$ in Eq. (3.50) and transformed weighting coefficient matrix $\mathbf{A}$ in Eq. (3.58) may not be invertible and the whole procedure will fail. The examination of this possible situation is thus conducted to analyze the stability of the RIQ method.

First the matrix $\mathbf{A}$ in Eq. (3.58) is studied. The matrix $\mathbf{A}$ is actually the weighting coefficient matrix of the polynomial-based differential quadrature (PDQ) method [117]. According to the PDQ method, the determinant of matrix $\mathbf{A}$ won't be equal to zero if the discretized governing equation is applied at more than $(m-n)$ virtual nodes, where m is the size of the matrix, and n is the differential order of the function. Otherwise the resulting weighting coefficient matrix $\mathbf{A}$ will be singular. According to Section 3.2.2.1, the weighting coefficients of the GIQ method are obtained by using the first order differential of PDQ method, thus $n=1$ and $m=M$ in this thesis. In other words, the number of used virtual nodes in this thesis is actually $(M-1)$. As the number of field nodes cannot be more than that of field nodes (otherwise the number of unknown

variables are more than the number of equations in Eq. (3.66)), the number of field nodes should be no more than $(M - 1)$, i.e., $N \leq M - 1$.

For the transformed moment matrix $\mathbf{P}^T \mathbf{R}^{-1} \mathbf{P}$ in Eq. (3.50), the singularity analysis is conducted as follows. Because $\mathbf{R}$ is symmetric and invertible (with a full rank), the transformed moment matrix $\mathbf{P}^T \mathbf{R}^{-1} \mathbf{P}$ can be made at least symmetric. If the columns in $\mathbf{P}$ are independent, then $\mathbf{P}^T \mathbf{R}^{-1} \mathbf{P}$ is easily proved to be invertible by invoking the full rank property of $\mathbf{R}$. To have all the columns of $\mathbf{P}$ could be a problem in theory, but it is very easy to achieve in practice. It is noted that the size of matrix $\mathbf{P}$ is $N \times l$, where N is the number of the field nodes, and l is the number of polynomial base. According to the method proposed by Liu [26], a strategy of having $N > l$ would be a workable method to avoid the singularity issue. The minimum l in this thesis is equal to 2.

In conclusion, the number of field nodes should be more than 2, and no more than $(M - 1)$, i.e., $2 < N \leq M - 1$. As the least square approximation is employed to solve the final discrete system, the number of virtual nodes is thus much greater than that of field nodes in order to obtain accurate results in the practical computing.

3.3 summary

In this chapter, the RIQ method is developed based on the GIQ and Kriging techniques. In the RIQ method, there are two sets of nodes scatted, the virtual and field nodes. The GIQ technique is used for discretization of the governing equation over the virtual nodes, while the Kriging interpolation technique is used for interpolation of the approximate function values at the virtual nodes. After the discretization of the

governing integral equations by the GIQ technique over the virtual nodes, the approximation function values at the virtual nodes are interpolated by the Kriging technique over the field nodes. The GIQ technique and the Kriging interpolation are thus combined to discretize the integral equation at the randomly or uniformly distributed field nodes. In Sections 3.2.2.1 and 3.2.2.2, the discretizing and interpolating techniques are discussed in detail. Finally, the whole numerical procedure is given in detail by taking the Fredholm integral equation of the second kind as an example.

Since the RIQ method is developed based on the GIQ method, it is of interest to compare the performance of these two methods. As is well known, the time complexity is usually considered to examine the efficiency of an algorithm. The time complexity of an algorithm is commonly expressed using big O notation, which excludes coefficients and lower order terms. It is seen from Eq. (3.56) that the time complexity $T(M) = O(M^3)$ for the GIQ technique, as there is a matrix inversion to obtain the GIQ solution, where M is the input size, and is equal to the number of virtual nodes in this case. For the RIQ method, the time complexity is also equal to $O(M^3)$ because there is a matrix multiply together with a matrix inversion, seen from Eq. (3.66). Hence, the time complexities for these two methods are the same.

Compared with the GIQ method, advantage of the RIQ method is that this method can be applied with the field nodes distributed either randomly or uniformly, and the integral domain being regular or irregular. In other words, the RIQ method has a wider range application compared with the GIQ method.

Figure 3.1. Virtual and field nodes.

Figure 3.2. Schematic diagram of the GIQ method.

Chapter 4 Numerical Validation of the RIQ Method

4.1 Introduction

In Chapter 3, the formulation of the RIQ method is detailed. In order to validate this method, several standard integral equations are taken as examples and solved here for numerical convergence analysis, including Fredholm integral equations of the second kind and Volterra integral equations of the second kind, in the 1-D, 2-D and 3-D domains, respectively. The purpose of the convergence analysis is to examine the robustness of the RIQ algorithms. The same order of monomial as the monomial order of the exact solution is employed as the polynomial basis when the shape function is constructed, such that the field function is fully reproduced for satisfactory approximation of numerical computation. [Equation Chapter \(Next\) Section 1](#)

4.2 The second kind of Fredholm integral equations

As mentioned in Section 1.3.2.1, according to the location of the unknown function, the Fredholm integral equations are classified into two kinds of the equations, namely Fredholm integral equations of the first and second kinds, respectively. Here the Fredholm integral equation of the second kind is taken as an example for numerical validation of the RIQ method, which is given as follows,

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_D K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} \quad (4.1)$$

where $\lambda \neq 0$ is a parameter, D is a closed bound set in real space and independent of the field variable. $K(\mathbf{x}, \mathbf{y})$ is the integrable kernel function, and $f(\mathbf{x})$ a known function.

4.2.1 One-dimensional problems

4.2.1.1 Case one — $y(x) = (x^2 - 1/3) + \int_0^1 x \cdot y(t) dt$

In this example, the kernel function is defined as x , $f(x)$ as $(x^2 - 1/3)$, and the integral domain as $[0, 1]$,

$$y(x) = (x^2 - \frac{1}{3}) + \int_0^1 x \cdot y(t) dt \quad (4.2)$$

with the exact solution $y(x) = x^2$.

In this problem, the exact solution is a second-order polynomial, and thus the second-order monomial term is included in the polynomial basis when the shape function is calculated, in order to make the field function exactly reproduced by the RIQ method.

Two types of field node distribution are imposed in the computational domain: uniformly distributed field nodes with the 50 virtual nodes, and randomly distributed field nodes with the same number of the virtual nodes. The global error analysis of the numerical results is shown in Table 4.1 for the 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with the 50 virtual nodes. The corresponding comparisons between the numerical results and analytical exact solutions are shown in Figures 4.1 and 4.2 for the 1-D Fredholm integral equation via the 15 uniformly and randomly distributed field nodes, respectively.

It is observed from Table 4.1 that the RIQ method can achieve good computing accuracy directly for solution of the second kind of the Fredholm integral equation. It is also found that the global errors remain of the order of 10^{-3} for different numbers of field nodes, even if just a few field nodes are used, such as 3 field nodes. Probably the reason is that the analytical solution is included in the polynomial basis, such that accurate results can be obtained with a few field nodes. Such phenomenon is very important to illustrate that the RIQ method is able to make the field variables reproduced precisely by a small number of field nodes. In other words, the accurate results can be achieved with a really low number of the field nodes, if the complete order monomial is used in the polynomial basis. In the present example, a global error of 1.3529×10^{-3} is achieved with 3 uniformly distributed field nodes, and a global error of 2.2702×10^{-3} with 3 randomly distributed field nodes.

In order to obtain the minimum number of the field nodes necessary to be considered for the RIQ method, the number of the field nodes for this example is reduced to 2 and 1 respectively. When 2 field nodes are employed, the function values at the field nodes can still be calculated, and the global error is 0.0448 for this case. However, when the number of the field nodes is reduced to 1, the global error is infinite. According to the stability analysis in Section 3.2.3, the number of field nodes should be more than 2 to retain the stability of this method. Hence, the minimum number of field nodes is suggested to be 3 in the practical computation.

By comparing the numerical results in Table 4.1, it is found that the same order of global errors is obtained for both the randomly and uniformly distributed field nodes. It is an interesting case with 5 field nodes, where the better result is obtained by the

randomly than uniformly distributed field nodes. In brief, the presently proposed RIQ method is capable of handling randomly distributed field nodes as well as uniformly distributed nodes.

4.2.1.2 Case two — $y(x) = 1/4 - x + \int_0^1 (3t - 6x^2)y(t)dt$

Here the kernel function is defined as $(3t - 6x^2)$ with a nonlinear term, $f(\mathbf{x})$ as $(1/4 - x)$, and the integral domain as $[0, 1]$,

$$y(x) = \frac{1}{4} - x + \int_0^1 (3t - 6x^2)y(t)dt \quad (4.3)$$

with the exact solution is $y(x) = x^2 - x$.

In this problem, the exact solution is a polynomial including first-order and second-order monomials. At first, the linear basis is employed when the shape function is calculated, with the variations of the uniformly or randomly distributed field nodes and the 50 virtual nodes. The global errors of the numerical results are listed in Table 4.2 for the linear basis. Then the same problem is solved again by employing the quadratic basis when the shape function is constructed, with the same distributions of both the field and virtual nodes. The global errors of numerical results are listed in Table 4.3 for the quadratic basis.

In Table 4.2, the linear basis is employed as the polynomial basis when the shape function is established. It is seen that the computational results are quite good. Even if a few field nodes such as 3 field nodes are employed, the global error is as small as 2.1829×10^{-3} via the uniformly distributed field nodes, and 4.0140×10^{-3} via the

randomly distributed field nodes. As the number of the field nodes increases, the global errors become smaller. In Table 4.3, the monomials with up to second order are employed when the shape function is constructed. The results are almost the same as when the linear basis is used. The reason for this phenomenon may be in the Kriging interpolation function, which is able to reproduce the field function precisely even if the linear basis is used. It is thus concluded that both the linear and quadratic bases can be used when the shape function is constructed for a problem, in which the exact solution includes the monomials with up to second order. This is a quite good conclusion because it makes the choice of the polynomial basis flexible when the shape function is established. Hence, the linear basis is employed in the following problems.

4.2.1.3 Case three — $y(x) = x + \int_0^1 k(x,t)y(t)dt, 0 \leq x \leq 1, k(x,t) = \begin{cases} x, x \leq t \\ t, t \leq x \end{cases}$

In this case study, the kernel function is defined as a piecewise function, $f(x)$ as x , and the integral domain as $[0, 1]$,

$$y(x) = x + \int_0^1 k(x,t)y(t)dt, 0 \leq x \leq 1 \quad (4.4)$$

where the kernel function is

$$k(x,t) = \begin{cases} x, x \leq t \\ t, t \leq x \end{cases} \quad (4.5)$$

with the exact solution $y(x) = \sec(1)\sin x$.

In order to solve this problem, the linear basis is used as the polynomial basis of the shape function. Since the exact solution of this example is a trigonometric function, it is interesting to see how the numerical solutions converge to the analytical solution as the

number of the field nodes increases by the linear basis. The equation is solved via both the uniformly and randomly distributed field nodes with the 90 virtual nodes. The global errors of numerical results for both the uniformly and randomly distributed field nodes are shown in Table 4.4. The convergence curves are also plotted, with 90 virtual nodes and 3, 5, 10, 20, 50, 60 uniformly and randomly distributed field nodes respectively, which are shown in Figure 4.3. The convergence rates for both the uniformly and randomly distributed nodes are also calculated and listed in Table 4.5.

It is observed from Table 4.4 that the global errors of the numerical results improve rapidly as the number of the field nodes increases, for both the uniformly and randomly distributed field nodes. This is a desired behavior for convergence study. The global errors reduce to 1.4888×10^{-5} and 1.4291×10^{-5} with 60 randomly and uniformly distributed field nodes, respectively. When the number of the field nodes is below 10, the results are improved really fast. If the number of field nodes is larger than 10, the results are still improved but not very fast, especially when the number of the field nodes is more than 20, the results are almost constant. It is seen that, with the RIQ method, good accurate results can be obtained with even a few field nodes.

The numerical results via the randomly and uniformly distributed field nodes are almost the same for this problem. When the number of the field nodes is 3, the result by the uniformly distributed field nodes is much better than that by the randomly distributed field nodes. However, the results by the randomly distributed field nodes improve rapidly with the number of the field nodes increases. Finally the global errors for these two kinds of the field node distributions are almost the same. This phenomenon is seen clearly from the convergence curves, as shown in Figure 4.3. The convergence curve for

the global error of the randomly distributed nodes descends very fast as the node space becomes smaller, especially when the number of nodes is small. At last the two curves almost coincide with each other. It is thus understood that the RIQ method can work well for randomly distributed field nodes and uniformly distributed field nodes with the desired computational accuracy.

Convergence rates for the uniformly and randomly distributed field nodes are computed and listed in Table 4.5, based on the results listed in Table 4.4 and the convergence curves in Figure 4.3. It is observed that the convergence rates are good enough for both the uniformly and the randomly distributed field nodes. However, the convergence rate for the randomly distributed nodes is even better than that for the uniformly distributed nodes.

Figure 4.4 shows how the global error reduces as the field nodes increases. It is seen from Figure 4.4 that regardless of the uniformly or the randomly distributed field nodes, the steady decrease in the global error is achieved as the number of the field nodes increases, though at the beginning the global error for the randomly distributed field nodes is larger than that for the uniformly distributed field nodes.

4.2.1.4 Case four — $y(x) = e^{2x+1/3} - (1/3) \int_0^1 e^{2x-(5/3)t} y(t) dt$

In this problem, the kernel function is defined as $\exp(2x-5t/3)$, $f(x)$ as $\exp(2x+1/3)$, and the integral domain as $[0, 1]$,

$$y(x) = e^{2x+\frac{1}{3}} - \frac{1}{3} \int_0^1 \left(e^{2x-\frac{5}{3}t} \right) y(t) dt \quad (4.6)$$

with the exact solution $y(x) = e^{2x}$.

Since the exact solution is a highly nonlinear function, it is interesting to see how it converges when the linear basis is used as the polynomial basis of the shape function. The problem is solved via two kinds of the field node distributions, i.e., randomly and uniformly distributed field nodes. Field nodes up to 100 and the virtual nodes up to 180 are employed. The global errors of numerical results are listed in Table 4.6. The convergence curves for both the uniformly and randomly distributed field nodes are plotted in Figure 4.5. The convergence rates are also calculated for both kinds of field nodes and the results are listed in Table 4.7.

It is observed from Table 4.6 that the global errors for both the uniformly and randomly distributed field nodes are not satisfactory at very few field nodes. For example, the global error at the 3 randomly distributed field nodes is 1.5500×10^{-2} and for the 3 uniformly distributed field nodes is 1.2674×10^{-2} . However, they are improved with the increase of field nodes, especially when the number of the field nodes is below ten. It is observed that the global errors reduce to 1.3300×10^{-4} and 5.4387×10^{-5} for the randomly and uniformly distributed field nodes respectively, when the number of the field nodes is 10. Finally the global errors reduce to 1.3308×10^{-5} and 1.1400×10^{-5} for the randomly and uniformly distributed field nodes respectively, when the number of the field nodes is 100. It is thus understood that the RIQ method can achieve good numerical results with a few properly nodes and also can converge rapidly as the field nodes increases.

The convergence curves are plotted for the case of 3, 5, 10, 20, 40, 60, 80 and 100 field nodes with both the random and uniform distributions as shown in Figure 4.5. The computed result by the uniformly distributed field nodes is slightly better than that by the randomly distributed field nodes. The descending tendencies of the two curves are similar, and they approach the steady convergence. Convergence rate for the randomly distributed field nodes is 1.4451, which is slightly larger than 1.4098 for the uniformly distributed field nodes. Obviously the RIQ method can solve the exponential problems well, even if the linear basis is used as the polynomial basis when the shape function is constructed.

4.2.1.5 Case five — $y(x) = f(x) + \int_0^1 e^{x+t} y(t) dt$, $f(x)$ as a complex trigonometric function

The present kernel function is defined as e^{x+t} , $f(x)$ as a complex trigonometric function, and the integral domain as $[0, 1]$,

$$y(x) = f(x) + \int_0^1 e^{x+t} y(t) dt \quad (4.7)$$

where

$$f(x) = -\frac{e^x (N\pi - eN\pi \cos(N\pi) + e \sin(N\pi))}{1 + N^2 \pi^2} + \sin(N\pi x)$$

with exact $y(x) = \sin(N\pi x)$, $N = 1, 2, \dots$.

The present exact solution is a periodic trigonometric function, and $N = 4$ when the example is solved. Two kinds of scatted field nodes are used: randomly and uniformly distributed field nodes. Similarly, the linear basis is employed as the polynomial basis to

construct the shape function. Field nodes up to 100 and virtual nodes up to 300 are used, and the computational results are listed in Table 4.8. The convergence curves are plotted for the cases of the 10, 20, 40, 60, 80 and 100 field nodes, as shown in Figure 4.6. The convergence rates are also calculated, and listed in Table 4.9. Figures 4.7 and 4.8 show the comparison between the numerical and analytical solutions using the uniformly and randomly distributed field nodes, respectively.

It is found from Table 4.8 that the global errors are not sufficient when the number of the field nodes is below ten. The reason may be that the present problem contains high gradients so that the exact solution may not be caught with a few nodes. When the number of the field nodes is more than ten, the results are much improved. Due to the possible reason explained above, we give up the first two results listed in the table when the convergence rate is calculated. The convergence curves are plotted in Figure 4.6, in which the tendency of the global error descends as the number of the field nodes increases. The numerical results by the uniformly distributed field nodes are better than those by the randomly distributed field nodes in this problem. The convergence rate for the randomly distributed field nodes listed in Table 4.9 is slightly better than that for the uniformly distributed field nodes, though the small global error is obtained for the uniformly field nodes.

Figures 4.7 and 4.8 demonstrate the comparisons between the numerical results and analytical solutions by the 100 field nodes of uniform and random distributions respectively. From the figures it is observed that the numerical solution coincides with the analytical solution very well for both uniformly and randomly distributed field nodes, regardless of the crest or trough of the function.

4.2.1.6 Case six — $y(x) = 0.5x(1-x) + \int_0^1 K(x,t)y(t)dt$, $K(x,t) = \begin{cases} t(1-x), 0 \leq t \leq x \\ x(1-t), x \leq t \leq 1 \end{cases}$

In this problem, the kernel function is defined as Green function, $f(x)$ as $0.5x(1-x)$, and the integral domain as $[0, 1]$,

$$y(x) = 0.5x(1-x) + \int_0^1 K(x,t)y(t)dt \quad (4.8)$$

with a symmetrical Green function kernel as,

$$K(x,t) = \begin{cases} t(1-x), 0 \leq t \leq x \\ x(1-t), x \leq t \leq 1 \end{cases} \quad (4.9)$$

The corresponding exact solution is

$$y = A \sin(x) + B \cos(x) - 1 \quad (4.10)$$

where $A = \tan(0.5)$ and $B = 1$.

This example is taken to study how the global errors of the numerical results varies when the number of field nodes is fixed while that of the virtual nodes changes. Firstly 5 field nodes are given and the number of the virtual nodes ranges from 10 to 50. The computational results by 5 randomly and uniformly distributed field nodes are listed in Table 4.10 respectively. In succession, 20 uniformly and randomly distributed field nodes are given, and the number of the virtual nodes ranges from 20 to 200. The computational results by the 20 randomly and uniformly distributed field nodes are listed in Table 4.11 respectively.

It is seen from Table 4.10 that, as the number of the virtual nodes increases, the global errors for both the uniformly and randomly distributed field nodes reduce first and then

increase. This is quite an interesting phenomenon. When the number of the virtual nodes is 20, the global errors for both the uniformly and randomly distributed field nodes reach the smallest values. As shown in Table 4.11 however, with 20 field nodes and increasing the virtual nodes, the global errors for both the uniformly and randomly distributed field nodes reduce first and then remain almost a constant.

Probably the above phenomenon may be explained as follows. When the number of the field nodes is small, it is easy for large errors to emerge when solving the problem. The proper number of the virtual nodes thus needs to be chosen. However, when the number of the field nodes is large enough, the influence of the number of virtual nodes is no longer significant. In such a situation, the number of the virtual nodes is no longer important.

4.2.2 Two-dimensional problems

As mentioned in Section 3.2.2.1, the two dimensional integral term can be discretized as

$$\begin{aligned}
 & \int_{x_1}^{x_N} \int_{y_1}^{y_M} f(x, y) dx dy = \\
 & \left[\begin{aligned}
 & (w_{N,1}^I - w_{1,1}^I) \times (\bar{w}_{M,1}^I - \bar{w}_{1,1}^I), (w_{N,2}^I - w_{1,2}^I) \times (\bar{w}_{M,1}^I - \bar{w}_{1,1}^I), \dots, (w_{N,N}^I - w_{1,N}^I) \times (\bar{w}_{M,1}^I - \bar{w}_{1,1}^I), \\
 & (w_{N,1}^I - w_{1,1}^I) \times (\bar{w}_{M,2}^I - \bar{w}_{1,2}^I), (w_{N,2}^I - w_{1,2}^I) \times (\bar{w}_{M,2}^I - \bar{w}_{1,2}^I), \dots, (w_{N,N}^I - w_{1,N}^I) \times (\bar{w}_{M,2}^I - \bar{w}_{1,2}^I), \\
 & \vdots \\
 & (w_{N,1}^I - w_{1,1}^I) \times (\bar{w}_{M,M}^I - \bar{w}_{1,M}^I), (w_{N,2}^I - w_{1,2}^I) \times (\bar{w}_{M,M}^I - \bar{w}_{1,M}^I), \dots, (w_{N,N}^I - w_{1,N}^I) \times (\bar{w}_{M,M}^I - \bar{w}_{1,M}^I)
 \end{aligned} \right] \\
 & \times [f(x_1, y_1), f(x_2, y_1), \dots, f(x_N, y_1), f(x_1, y_2), f(x_2, y_2), \dots, f(x_N, y_2), \dots, f(x_N, y_M)]^T
 \end{aligned} \tag{4.11}$$

4.2.2.1 Case one — $u(x, y) = \int_0^1 \int_0^1 xyu(\xi, \eta) d\xi d\eta + (3/4)xy$

The kernel function of this example is defined as xy , $f(\mathbf{x})$ as $(3xy/4)$, and the integral domain as $0 \leq x$ and $y \leq 1$,

$$u(x, y) = \int_0^1 \int_0^1 xyu(\xi, \eta) d\xi d\eta + \frac{3}{4}xy \quad (4.12)$$

with the exact solution is $u(x, y) = xy$.

The problem is solved first using uniformly distributed field nodes of 3×3 , 6×6 , 9×9 , 12×12 and 15×15 . Then it is solved again using the same numbers of randomly distributed field nodes. The numerical results are listed in Table 4.12. The convergence curves for both the uniformly and randomly distributed field nodes are plotted in Figure 4.9, and the corresponding convergence rates are also computed in Table 4.13. Figures 4.10 and 4.11 show the comparisons between numerical and analytical exact solutions at 10×10 field nodes for uniformly and randomly distribution field nodes, respectively.

It is observed from Table 4.12 that the global error slightly increases when the number of field nodes is small, such as 3×3 field nodes. The global error is 5.9444×10^{-2} for randomly distributed field nodes and is 3.5443×10^{-3} for uniformly distributed field nodes. However, the global errors for both uniformly and randomly field nodes reduce rapidly as the number of the field nodes increases.

Figure 4.9 is plotted for analysis of the convergence rate for both uniformly and randomly distributed field nodes. At the beginning, the global error for the randomly distributed field nodes is larger than that for the uniformly distributed field nodes. Then the results turn to the opposite direction as the number of the field nodes increases. At

6×6 and 9×9 field nodes, the global errors for randomly distributed field nodes are even better than those for the uniformly distributed field nodes.

The convergence rates for both kinds of the field nodes are listed in Table 4.13. It is seen that the convergence rate for randomly distributed field nodes is better than that for uniformly distributed field nodes, such that the proposed RIQ method can achieve better convergence using the randomly distributed field nodes than uniformly distributed field nodes for this 2-D problem.

4.2.2.2 Case two — $u(x, y) = e^{x+y} - e^{xy}(e-1)^2 + \int_0^1 \int_0^1 e^{xy} u(\xi, \eta) d\xi d\eta$

The present kernel function is defined as e^{xy} , $f(\mathbf{x})$ as $e^{x+y} - e^{xy}(e-1)^2$, and the integral domain as $0 \leq x, y \leq 1$,

$$u(x, y) = e^{x+y} - e^{xy}(e-1)^2 + \int_0^1 \int_0^1 e^{xy} u(\xi, \eta) d\xi d\eta \quad (4.13)$$

with the exact solution $u(x, y) = e^{x+y}$.

The present exact solution is a 2-D exponential function. As the exact solution is a highly nonlinear function, it is valuable to study how it converges when the 2-D linear basis is employed as the polynomial basis to construct the shape functions. The problem is solved via 3×3, 6×6, 9×9, 12×12 and 15×15 randomly and uniformly distributed field nodes respectively, with the given 20×20 virtual nodes. The global errors of the numerical results are listed in Table 4.14. Using the results listed in Table 4.14, the convergence curves are plotted for both uniformly and randomly distributed field nodes, which are shown in Figure 4.12. The convergence rates for both uniformly and randomly distributed field nodes are also calculated and listed in Table 4.15. Figures 4.13 and 4.14

are plotted for comparisons between the numerical and analytical exact solutions by 10×10 field nodes with the random and uniform distributions respectively.

It is understood from Table 4.14 that the global error is small even if a few field nodes are employed for both the uniformly and randomly distributed field nodes. As the number of field nodes increases, the global errors become smaller. From the convergence curves shown in Figure 4.12, it is found that the computed result for the uniformly distributed field nodes is slightly better than that for the randomly distributed field nodes in this problem. It is obviously seen from Table 4.15 that the convergence rate for the uniformly distributed field nodes is almost the same as that for the randomly distributed field nodes, and both rates are relatively small compared with the previous example.

4.2.3 Three-dimensional problem

In this 3-D problem, the kernel function is defined as xyz , $f(\mathbf{x})$ as $7xyz/8$, and the integral domain as $0 \leq x, y$ and $z \leq 1$,

$$f(x, y, z) = \int_0^1 \int_0^1 \int_0^1 xyz f(\xi, \eta, \gamma) d\xi d\eta d\gamma + \frac{7}{8}xyz \quad (4.14)$$

with the exact solution $f(x, y, z) = xyz$.

This problem is solved by $3 \times 3 \times 3$, $6 \times 6 \times 6$ and $9 \times 9 \times 9$ uniformly and randomly distributed field nodes respectively, with the given $10 \times 10 \times 10$ virtual nodes. The global errors of the numerical results are listed in Table 4.16. The convergence curves are plotted in Figure 4.15 for both the uniformly and randomly distributed field nodes. The convergence rates are also calculated and listed in Table 4.17 for both the uniformly and randomly distributed field nodes.

It is demonstrated from Table 4.16 that the global errors are small for both uniformly and randomly distributed field nodes, even if a few field nodes, such as $3 \times 3 \times 3$ distributions are employed. In the case of $3 \times 3 \times 3$ distributed field nodes, the global error for the randomly distributed field nodes is smaller than that for uniformly distributed field nodes. From Table 4.17 and Figure 4.15, it is shown that the convergence rate for the uniformly distributed field nodes is better than that for randomly distributed field nodes in this problem.

4.3 Volterra integral equations of the second kind

As introduced in the first chapter, the general form for Volterra integral equations of the second kind is given by

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_{D(\mathbf{x})} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} \quad (4.15)$$

where $\lambda \neq 0$ is a numerical parameter, $D(\mathbf{x})$ a variation domain, $K(\mathbf{x}, \mathbf{y})$ the integrable kernel function, and $f(\mathbf{x})$ a known function.

4.3.1 One-dimensional problems

4.3.1.1 Case one — $u(x) = x^2 - (1/12)x^4 + \int_0^x (x-t)u(t)dt$

In this case study, the kernel function is defined as $(x-t)$, $f(\mathbf{x})$ as $(x^2 - x^4/12)$, and the integral domain as $[0, x]$,

$$u(x) = x^2 - \frac{1}{12}x^4 + \int_0^x (x-t)u(t)dt \quad (4.16)$$

where $0 < x < 2$. The corresponding exact solution is $u(x) = x^2$.

The linear basis is employed first as the polynomial basis for construction of the shape function. The problem is solved by both uniformly and randomly distributed field nodes respectively, with 30 virtual nodes. The global errors of the numerical results are listed in Table 4.18. Then the problem is solved again using the quadratic basis as the polynomial basis to construct the shape function. The computational results for both uniformly and randomly distributed field nodes and a fixed number of virtual nodes are obtained and listed in Table 4.19. Figures 4.16 and 4.17 are plotted for the comparisons between the numerical results and analytical solutions for the randomly and uniformly distributed field nodes respectively.

As illustrated in Table 4.18, the global error is small for uniformly distributed field nodes as well as for randomly distributed field nodes. Even with 3 field nodes, the global error reaches the order of 10^{-3} for both uniformly and randomly distributed field nodes, meaning that the RIQ method can produce very accurate results just with a few field nodes for the second kind of Volterra integral equations. As the number of field nodes increases, the global errors reduce for both uniformly and randomly distributed field nodes, showing that the numerical results converge with an increasing number of field nodes, or smaller node space. The computed results by the quadratic basis are presented in Table 4.19, in which the same performing behaviors exist as described in Table 4.18.

Comparing the results obtained between linear and quadratic bases, it is shown that, with a few field nodes such as 3 field nodes, the results with the quadratic basis are slightly better than those with the linear basis. The global errors with the linear basis are

1.1588×10^{-3} and 1.6773×10^{-3} for uniformly and randomly distributed field nodes, respectively, while the global errors with the quadratic basis are 2.6071×10^{-4} and 5.3622×10^{-4} for uniformly and randomly distributed field nodes respectively. However, as the number of the field nodes increases, the global errors of the numerical results with the linear and quadratic bases become almost the same. For instance, at 15 field nodes, the global errors with the linear basis for uniformly and randomly distributed field nodes are 2.7467×10^{-4} and 3.0015×10^{-4} respectively, while the global errors with the quadratic basis are 2.6851×10^{-4} and 2.8370×10^{-4} for uniformly and randomly distributed field nodes respectively. From these results, it is concluded that both linear and quadratic bases can be used as the polynomial basis to construct the shape functions. Figures 4.16 and 4.17 also show that the numerical solutions coincide well with the analytical exact solutions for the randomly and uniformly distributed field nodes respectively.

4.3.1.2 Case two — $u(x) = 1 + \int_0^x u(t) dt$

In this problem, the kernel function is defined as a constant, $f(\mathbf{x})$ as 1, and the integral domain as $[0, x]$,

$$u(x) = 1 + \int_0^x u(t) dt \quad (4.17)$$

where $0 < x \leq 2$, and the exact solution is $u(x) = e^x$.

The exact solution of this example case is an exponential function. As stated before, the linear basis is employed as the polynomial basis for constructing the shape functions.

Both the uniformly and randomly distributed field nodes are employed to solve this problem, with the number of the virtual nodes up to 150 and the number of the field nodes ranges from 5 to 70. The global errors of the numerical results are listed in Table 4.20. The convergence curves are plotted in Figure 4.18 for both the uniformly and randomly distributed field nodes.

It is observed from Table 4.20 that global errors are small for both uniformly and randomly distributed field nodes even if a few field nodes are used. The global errors for both randomly and uniformly distributed field nodes reduce with the increase of the field nodes.

Figure 4.18 shows the convergence tendencies of the numerical results for both uniformly and randomly distributed field nodes. At first it converges slowly for both kinds of field nodes distributions. However, with the number of field nodes getting larger, the function values converge rapidly for both the uniformly and randomly distributed field nodes. This phenomenon is different from that we discovered when Fredholm integral equation of the second kind is solved. In that kind of problem the numerical results converge rapidly first as the number of the field nodes increases on a small scale, and then converge slowly with large number of field nodes.

4.3.1.3 Case three — $u(x) = (1/4)x \cos(2x) + \sin(x) - (1/2)x + \int_0^x x \cos(t)u(t)dt$

Here the kernel function is defined as x , $f(x)$ as $\left(\frac{1}{4}x \cos(2x) + \sin(x) - \frac{1}{2}x\right)$,

and the integral domain as $[0, x]$,

$$u(x) = \frac{1}{4}x \cos(2x) + \sin(x) - \frac{1}{2}x + \int_0^x x \cos(t) u(t) dt \quad (4.18)$$

where $0 < x \leq \pi$, and the corresponding exact solution is $u(x) = \sin(x)$.

In this problem, the analytical solution is a trigonometric function. Similarly, the linear basis is employed to construct the shape function. Two kinds of field node distributions are used to solve this example problem, i.e., uniformly and randomly distributed field nodes. The number of field nodes range from 5 to 70, while the number of virtual nodes is up to 200. The global errors of the numerical results are listed in Table 4.21. The convergence curves for both the uniformly and randomly distributed field nodes are shown in Figure 4.19, and the convergence rates are also calculated and listed in Table 4.22.

It is seen from Table 4.21 that the global errors are small for both the uniformly and randomly distributed field nodes even if a few field nodes are employed. As the number of the field nodes increases, the global errors reduce slightly at first, and then they reduce rapidly when the number of the field nodes becomes larger. Field nodes of 5, 10, 40 and 70 are used to plot the convergence curves in Figure 4.19. It is seen that the numerical results converge slowly when the numbers of the field nodes are small, and they converge fast as the numbers of the field nodes increase largely. This is similar to the example in Section 4.3.1.2. From the convergence rates displayed in Table 4.22, it is obvious that the convergence rate for the randomly distributed field nodes is larger than that for the uniformly distributed field nodes.

4.3.2 Two-dimensional problems

4.3.2.1 Case one — $u(x, y) = -(1/4)x^2y^2 + xy + \int_0^x \int_0^y u(\xi, \eta) d\xi d\eta$

This example is used to illustrate the numerical solution when the kernel function is defined as a constant, $f(\mathbf{x})$ as $(-x^2y^2/4 + xy)$ and the integral domain as $0 \leq \xi \leq x$ and $0 \leq \eta \leq y$,

$$u(x, y) = -\frac{1}{4}x^2y^2 + xy + \int_0^x \int_0^y u(\xi, \eta) d\xi d\eta \quad (4.19)$$

where $0 < x, y < 2$. Here the exact solution $u(x, y) = xy$.

A 2-D linear basis is employed to solve this equation, in which both uniformly and randomly distributed field nodes are used with fixed 20×20 virtual nodes and the number of field nodes is given as 3×3 , 6×6 , 9×9 , 12×12 and 15×15 . The global errors of the numerical results are listed in Table 4.23. The convergence curves for both the uniformly and randomly distributed field nodes are demonstrated in Figure 4.20. The convergence rates are also calculated and given in Table 4.24.

It is clear from Table 4.23 that the global errors for the uniformly distributed field nodes are smaller than those for the randomly distributed field nodes when the numbers of the field nodes are small. However, both the global errors reduce rapidly with increasing the number of the field nodes, and finally remain more or less a constant. From the convergence curves shown in Figure 4.20, it is seen that the numerical results converge rapidly first and then slowly. The results for the uniformly distributed field nodes are slightly better than those for the randomly distributed field nodes. On contrast

the convergence rate for the randomly distributed field nodes is larger than that for the uniformly distributed field nodes, as shown in Table 4.24.

Figures 4.21 and 4.22 make the comparisons between the numerical and analytical solutions for the randomly and uniformly distributed field nodes, respectively. It is found that the numerical solutions coincide well with the analytical solutions.

4.3.2.2 Case two — $u(x, y) = -1 + e^x + e^y + \int_0^x \int_0^y u(\xi, \eta) d\xi d\eta$

In order to demonstrate how the numerical solution captures the exact solution which is a nonlinear function, this case study is carried out, in which the kernel function is defined as a constant, $f(\mathbf{x})$ as $(-1 + e^x + e^y)$, and the integral domain as $0 \leq \xi \leq x$ and $0 \leq \eta \leq y$,

$$u(x, y) = -1 + e^x + e^y + \int_0^x \int_0^y u(\xi, \eta) d\xi d\eta \quad (4.20)$$

where $0 < x$ and $y < 2$. The corresponding exact solution is $u(x, y) = e^{x+y}$.

The exact solution of this problem is a 2-D exponential function. It is of interest to study how the computed results converge with 2-D linear basis employed as the polynomial basis for constructing the shape functions. Two kinds of the field node distributions are employed to solve this equation, which are randomly and uniformly distributed field nodes. The numbers of field nodes are set as 3×3 , 6×6 , 9×9 , 12×12 and 15×15 , while the number of virtual nodes is fixed at 20×20 . The global errors of the numerical results are listed in Table 4.25. The convergence curves for both uniformly and randomly distributed field nodes are plotted in Figure 4.23, and the convergence rates are computed and displayed in Table 4.26. Figures 4.24 and 4.25 show the

comparisons between the numerical and analytical solutions by the randomly and uniformly distributed field nodes respectively.

It is observed from Table 4.25 that global errors for both uniformly and randomly distributed field nodes are small even if a few field nodes are employed. At a small number of field nodes such as 3×3 , the global error for the uniformly distributed field nodes is much smaller than that for the randomly distributed field nodes. The global errors for both the randomly and uniformly distributed field nodes reduce with increasing the field nodes. Finally, the global errors for the two kinds of node distributions are almost equal to each other. Figure 4.23 presents how the numerical results converge with the node spacing getting smaller for both the uniformly and randomly distributed field nodes. It is seen that the global errors for uniformly distributed field nodes are slightly smaller than those for the randomly distributed field nodes. However, the disparity becomes smaller as the node space decreases. This phenomenon is seen even more clearly from the convergence rates listed in Table 4.26. The convergence rate for the randomly distributed field nodes is equal to 1.0424, which is larger than 0.3271 for the uniformly distributed field nodes.

4.4 Summary

In this chapter, the presently developed RIQ method has been validated by solving in total 14 different problems defined in 1-D, 2-D and 3-D domains. They are 9 Fredholm integral equations of the second kind and 5 Volterra integral equations of the second kind, with both randomly and uniformly distributed field nodes. Based on the comparisons and discussions of the numerically computational results, it is concluded that the RIQ method

is a powerful tool for approximate solutions of various integral equations, by scattering the random and uniform distributive field nodes. In this chapter, the convergence rates are also studied for both the uniformly and randomly distributed field nodes by studying the 9 examples (Sections 4.2.1.3~4.2.1.5, Sections 4.2.2.1~4.2.2.2, Section 4.2.3, Section 4.3.1.3 and Sections 4.3.2.1~4.3.2.2). Four of the 9 examples achieve larger convergence rates via the randomly distributed field nodes if compared with those via the uniformly distributed field nodes, other four example cases result in the contrast results. The remaining example achieves the convergence rate by the randomly distributed field nodes which are almost the same as that by the uniformly distributed field nodes (diversity factor $< 3\%$). It is thus understood that the RIQ method can handle the randomly distributed field nodes equally as to the uniformly distributed field nodes.

Table 4.1. Global errors of the example case shown in Section 4.2.2.1 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 50 virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3	50	2.2702E-3	1.3529E-3
5	50	1.0846E-3	1.4562E-3
10	50	1.4548E-3	1.4003E-3
15	50	1.3989E-3	1.3962E-3
20	50	1.4838E-3	1.3738E-3
25	50	1.5191E-3	1.0080E-3
30	50	2.2718E-3	1.1853E-3
35	50	1.4426E-3	1.3830E-3
40	50	5.3381E-3	1.3830E-3

Table 4.2. Global errors of the example case shown in Section 4.2.1.2 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 50 virtual nodes using linear basis.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3	50	4.0140E-3	2.1829E-3
5	50	3.2534E-3	2.8336E-3
10	50	2.8324E-3	2.7771E-3
15	50	2.6085E-3	2.6023E-3
20	50	2.6072E-3	2.5626E-3
25	50	2.4416E-3	2.5315E-3
30	50	2.4767E-3	2.5324E-3
35	50	2.5216E-3	2.5538E-3
40	50	2.3102E-3	2.7772E-3

Table 4.3. Global errors of the example case shown in Section 4.2.1.2 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 50 virtual nodes using quadratic basis.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3	50	4.1423E-3	3.1259E-3
5	50	3.7843E-3	2.8364E-3
10	50	2.8310E-3	2.7043E-3
15	50	2.6182E-3	2.6110E-3
20	50	2.5733E-3	2.5939E-3
25	50	2.4391E-3	2.5724E-3
30	50	2.2385E-3	2.5660E-3
35	50	2.5541E-3	2.5562E-3
40	50	2.5815E-3	2.5528E-3

Table 4.4. Global errors of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 90 virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3	90	6.9601E-3	3.8285E-3
5	90	1.5180E-3	1.5057E-4
10	90	4.9509E-5	7.1407E-5
20	90	1.2783E-5	1.5181E-5
30	90	1.8673E-4	2.3372E-5
40	90	2.2188E-5	1.4963E-5
50	90	1.6014E-5	1.5078E-5
60	90	1.4888E-5	1.4291E-5

Table 4.5. Convergence rates of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by the random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Random	1.4561
Uniform	1.9055

Table 4.6. Global errors of the example case shown in Section 4.2.1.4 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3	90	1.5500E-2	1.2674E-2
5	90	2.7721E-4	1.9560E-4
10	90	1.3300E-4	5.4387E-5
20	90	2.6991E-5	2.4431E-5
40	90	3.2718E-5	2.6658E-5
60	90	2.4774E-5	1.9140E-5
80	180	2.0331E-5	1.3602E-5
100	180	1.3308E-5	1.1400E-5

Table 4.7. Convergence rates of the example case shown in Section 4.2.1.4 for 1-D Fredholm integral equation by the random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Uniform	1.4098
Random	1.4451

Table 4.8. Global errors of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation by the uniformly and randomly distributed field nodes.

Field nodes	Virtual nodes	Error for randomly distributed real nodes	Error for uniformly distributed real nodes
6	90	1.3143E-1	2.1923E-1
8	90	1.4785E-2	1.6230E-2
10	90	5.2204E-4	2.5252E-4
20	90	4.3019E-4	1.3222E-4
40	90	1.8046E-4	1.3167E-4
60	90	1.6329E-4	7.0355E-5
80	300	1.3363E-4	4.0760E-5
100	300	1.0836E-4	3.4080E-5

Table 4.9. Convergence rates of example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation by the random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Random	0.7996
Uniform	0.6874

Table 4.10. Global errors of the example case shown in Section 4.2.1.6 for 1-D Fredholm integral equation by 5 uniformly and randomly distributed field nodes with variable virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
5	10	2.1375E-3	1.9133E-3
5	15	2.0614E-3	6.6884E-4
5	20	2.5354E-4	3.6553E-4
5	25	8.8093E-4	1.1514E-3
5	30	4.0675E-4	1.1154E-3
5	40	1.4515E-3	1.0358E-2
5	50	2.7511E-3	3.4671E-2

Table 4.11. Global errors of the example case shown in Section 4.2.1.6 for 1-D Fredholm integral equation by 20 uniformly and randomly distributed field nodes with variable virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
20	20	7.4129E-4	1.2501E-3
20	30	4.3503E-4	4.2010E-4
20	40	1.4924E-4	1.3326E-4
20	50	3.1007E-4	3.2728E-4
20	60	7.0579E-5	8.4985E-5
20	70	3.7972E-4	3.9422E-4
20	80	2.7600E-5	3.1069E-5
20	90	3.0224E-5	3.9271E-5
20	140	3.2138E-5	2.7395E-5
20	200	2.7237E-5	3.0366E-5

Table 4.12. Global errors of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3×3	20×20	5.9444E-2	3.5443E-3
6×6	20×20	1.2672E-3	2.0806E-3
9×9	20×20	6.3762E-4	1.0983E-3
12×12	20×20	5.1084E-4	1.2543E-4
15×15	20×20	2.0752E-4	1.1966E-4

Table 4.13. Convergence rates of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation by the random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Uniform	1.8428
Random	2.7684

Table 4.14. Global errors of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3×3	20×20	2.7653E-3	1.5418E-3
6×6	20×20	1.2600E-3	7.5165E-4
9×9	20×20	1.1479E-3	1.0447E-3
12×12	20×20	1.1227E-3	1.0047E-3
15×15	20×20	1.0546E-3	3.7855E-4

Table 4.15. Convergence rates of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation by random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Uniform	0.4902
Random	0.4641

Table 4.16. Global errors of the example case shown in Section 4.2.3 for 3-D Fredholm integral equation by the uniformly and randomly distributed field nodes with $10 \times 10 \times 10$ virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed virtual nodes
$3 \times 3 \times 3$	$10 \times 10 \times 10$	7.6079E-3	1.2466E-2
$6 \times 6 \times 6$	$10 \times 10 \times 10$	3.2746E-3	1.4743E-3
$9 \times 9 \times 9$	$10 \times 10 \times 10$	2.1764E-3	6.3426E-4

Table 4.17. Convergence rates of the example case shown in Section 4.2.3 for 3-D Fredholm integral equation by random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Uniform	2.1733
Random	0.9051

Table 4.18. Global errors of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes with 30 virtual nodes using linear basis.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3	30	1.6773E-3	1.1588E-3
6	30	8.8068E-4	2.3844E-4
9	30	3.7266E-4	3.6267E-4
12	30	2.6234E-4	2.9177E-4
15	30	3.0015E-4	2.7467E-4
18	30	6.7438E-4	2.7932E-4
21	30	2.1954E-4	2.7775E-4

Table 4.19. Global errors of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes with 30 virtual nodes using quadratic basis.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3	30	5.3622E-4	2.6071E-4
6	30	2.2849E-4	2.3485E-4
9	30	2.1164E-4	2.5580E-4
12	30	1.9686E-4	2.6673E-4
15	30	2.8370E-4	2.6851E-4
18	30	2.3502E-4	2.7242E-4
21	30	3.1695E-4	2.8079E-4

Table 4.20. Global errors of the example case shown in Section 4.3.1.2 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
5	100	4.1050E-3	2.1961E-3
10	100	2.8451E-3	2.3239E-3
15	100	2.4841E-3	2.2501E-3
20	100	2.8193E-3	2.3127E-3
30	100	2.5200E-3	2.2855E-3
40	100	2.4646E-3	2.2794E-3
50	100	2.3019E-3	2.2757E-3
60	150	1.4611E-3	7.0915E-4
70	150	7.6030E-4	6.5830E-4

Table 4.21. Global errors of the example case shown in Section 4.3.1.3 for 1-D Volterra integral equation by the uniformly and randomly distributed field nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
5	100	2.4302E-3	1.6462E-3
10	100	1.3339E-3	1.3825E-3
20	100	2.3483E-3	1.9721E-3
30	100	2.5174E-3	1.6331E-3
40	100	1.4112E-3	1.5528E-3
50	100	2.8461E-3	1.4941E-3
70	200	2.5470E-4	5.8470E-4

Table 4.22. Convergence rates of the example case shown in Section 4.3.1.3 for 1-D Volterra integral equation by random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Random	0.6033
Uniform	0.2606

Table 4.23. Global errors of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3×3	20×20	3.3118E-2	1.0345E-2
6×6	20×20	7.9787E-3	7.5447E-3
9×9	20×20	7.8908E-3	7.2573E-3
12×12	20×20	7.6517E-3	7.1753E-3
15×15	20×20	7.5415E-3	7.1345E-3

Table 4.24. Convergence rates of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Random	0.1908
Uniform	0.7442

Table 4.25. Global errors of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by the uniformly and randomly distributed field nodes with 20×20 virtual nodes.

Field nodes	Virtual nodes	Error with randomly distributed real nodes	Error with uniformly distributed real nodes
3×3	20×20	4.2832E-2	1.0000E-2
6×6	20×20	8.0018E-3	5.3233E-3
9×9	20×20	6.1152E-3	5.2155E-3
12×12	20×20	5.8476E-3	5.1781E-3
15×15	20×20	5.7093E-3	5.2837E-3

Table 4.26. Convergence rates of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by random and uniform field node distributions.

Distributive type of field nodes	Convergence rate
Random	1.0424
Uniform	0.3271

Figure 4.1. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.1 for 1-D Fredholm integral equation via 15 uniformly distributed field nodes.

Figure 4.2. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.1 for 1-D Fredholm integral equation via 15 randomly distributed field nodes.

Figure 4.3. Convergence rates of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by randomly and uniformly distributed field nodes.

Figure 4.4. Variations of the global errors of the example case shown in Section 4.2.1.3 for 1-D Fredholm integral equation by randomly or uniformly distributed field nodes.

Figure 4.5. Convergence rates of the example case shown in Section 4.2.1.4 for 1-D Fredholm integral equation by randomly and uniformly distributed field nodes.

Figure 4.6. Convergence rates of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation by randomly and uniformly distributed field nodes.

Figure 4.7. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation via 100 uniformly distributed field nodes.

Figure 4.8. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.1.5 for 1-D Fredholm integral equation via 100 randomly distributed field nodes.

Figure 4.9. Convergence rates of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation by randomly and uniformly distributed field nodes.

Figure 4.10. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation via 10×10 uniformly distributed field nodes.

Figure 4.11. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.1 for 2-D Fredholm integral equation via 10×10 randomly distributed field nodes.

Figure 4.12. Convergence rates of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation by randomly and uniformly distributed field nodes.

Figure 4.13. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation via 10×10 randomly distributed field nodes.

Figure 4.14. Comparison between the numerical and exact solutions of the example case shown in Section 4.2.2.2 for 2-D Fredholm integral equation via 10×10 uniformly distributed field nodes.

Figure 4.15. Convergence rates of the example case shown in Section 4.2.3 for 3-D Fredholm integral equation by randomly and uniformly distributed field nodes.

Figure 4.16. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation via 15 randomly distributed field nodes.

Figure 4.17. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.1.1 for 1-D Volterra integral equation via 15 uniformly distributed field nodes.

Figure 4.18. Convergence rates of the example case shown in Section 4.3.1.2 for 1-D Volterra integral equation by randomly and uniformly distributed field nodes.

Figure 4.19. Convergence rates of the example case shown in Section 4.3.1.3 for 1-D Volterra integral equation by randomly and uniformly distributed field nodes.

Figure 4.20. Convergence rates of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by randomly and uniformly distributed field nodes.

Figure 4.21. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by 15×15 randomly distributed field nodes.

Figure 4.22. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.1 for 2-D Volterra integral equation by 15×15 uniformly distributed field nodes.

Figure 4.23. Convergence rates of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by randomly and uniformly distributed field nodes.

Figure 4.24. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by 15×15 randomly distributed field nodes.

Figure 4.25. Comparison between the numerical and exact solutions of the example case shown in Section 4.3.2.2 for 2-D Volterra integral equation by 15×15 uniformly distributed field nodes.

Chapter 5 Application of RIQ Method in Irregular Domains and for Approximate Solution of Nonlinear Integral Equations

5.1 Introduction

In Chapter 3, the formulation of the random integral quadrature (RIQ) method is presented. The method is validated in Chapter 4, by solving the second kind of Fredholm integral equations and the second kind of Volterra integral equations, respectively. In this chapter, the RIQ method is examined by solving integral equations on irregular integral domains as well as nonlinear integral equations. [Equation Chapter \(Next\) Section 1](#)

As mentioned in Chapter 3, the RIQ method is developed for extension of the generalized integral quadrature (GIQ) approach, as the GIQ approach requires the integration nodes lying in a straight line with the regular integral domain. For partial differential equations, irregular domains may be transformed to regular domains by coordinate transformation [119]. However, it is difficult to do so for integral equations. Therefore, it is necessary to investigate the numerical method for solving integral equations on irregular integral domains. The RIQ method is considered here to solve this kind of integral equation, as detailed in Section 5.2, where the detailed procedure is given in Section 5.2.1 for the integral equation on irregular domains, and the numerical experiments are conducted in Section 5.2.2.

In addition, nonlinear integral equations as another type of complex integral equations are also investigated in this chapter, since they play a very important role in characterizing physics, elasticity, viscodynamics fluid, and mixed mechanics problems in continuous media. In general, it is difficult to obtain analytical solutions of nonlinear integral equations, such that many researchers focus on various numerical solutions. Depending on whether the integral domain is fixed or variable, nonlinear integral equations can be categorized into two groups, namely nonlinear Fredholm integral equations with the fixed integral domain, and nonlinear Volterra integral equations with the variable integral domain respectively. In this chapter, the nonlinear Volterra integral equation is studied, and written as follows,

$$\phi(t) = f(t) + \int_0^t K(t, s, \phi(s)) ds \quad (5.1)$$

where K is a kernel function, and f is a known function.

The numerical RIQ method for solving the nonlinear Volterra integral equations is described in Section 5.3.1, and numerical experiments are presented in Section 5.3.2, in order to validate this method. In Section 5.3.3 this method is applied to the Blasius boundary-layer problem.

5.2 Application of the RIQ method in irregular integral domains

5.2.1 Solution methods

A typical form of integral equation is expressed as

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \lambda \int_{D(\mathbf{x})} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) d\mathbf{y} \quad (5.2)$$

where $K(\mathbf{x}, \mathbf{y})$ is the kernel of the integral equation, $D(\mathbf{x})$ is the integral domain, $f(\mathbf{x})$ is a given function, and λ is a constant parameter. The present integral domain is taken to be irregularity. In other words, the integral domain is not formed by several straight lines along the directions of system coordinate axes, but formed by several curves, such as the domain shown in Figure 5.1.

5.2.1.1 Development of a strategy for nodes scattered in irregular domains

As stated before, two sets of the nodes are required in the RIQ method, virtual nodes and the field nodes. The virtual nodes are scattered in the form of

$$x^i = \frac{L1}{2} \left[1 - \cos \left(\frac{i-1}{M1-1} \pi \right) \right] \quad (5.3)$$

and

$$y^j = \frac{L2}{2} \left[1 - \cos \left(\frac{j-1}{M2-1} \pi \right) \right] \quad (5.4)$$

where $L1$ and $L2$ are the domain maximum lengths in the x - and y - directions respectively. $M1$ and $M2$ are the numbers of virtual nodes in the x - and y - directions respectively. Hence, a discretized rectangular domain is created which includes the irregular domain with totally $M1 \times M2$ virtual nodes, as shown in Figure 5.2. It is seen that the irregular domain is fully discretized by part of the linearly distributed virtual nodes. As such, only the virtual nodes inside the integral domain are employed. In Figure 5.2, the solid nodes represent the virtual nodes employed along a straight line. If M is the number of virtual nodes given inside the integral domain, the virtual nodes are given as $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^M$.

The field nodes are distributed either uniformly or randomly. However, since the virtual nodes cannot be chosen arbitrarily, the boundary of the domain is not accurately captured. It is thus suggested that the field nodes are distributed on the boundary of the domain as well as within the domain. In the RIQ method, the field nodes are distributed within the integral domain as well as on the boundary. The function value at each virtual node is approximated by the summation of the weighting coefficient multiplying by the function value at each field node. The field nodes on the boundary take the same effect as the nodes within the domain during the computation. Hence, there is no guideline how many nodes are needed to be placed on the boundary exactly. They are shown by the circle nodes in Figure 5.2. If totally N field nodes are employed, they are given as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$.

5.2.1.2 Discretization of the governing equations

Eq. (5.2) is discretized first over the virtual nodes as

$$\phi(\mathbf{x}^i) = f(\mathbf{x}^i) + \lambda \sum_{j=1}^M w^j K(\mathbf{x}^i, \mathbf{y}^j) \phi(\mathbf{y}^j) \quad (5.5)$$

where w^j is the weighting coefficient for the j th virtual node. Since the domain is two-dimensional, the weighting coefficient is calculated by $w^j = w_x^j \times w_y^j$, where w_x^j and w_y^j are the GIQ weighting coefficients for the j th node in the x - and y - directions, respectively. It is remarkable that w_x and w_y are calculated by $M1$ virtual nodes in the x -direction and $M2$ virtual nodes in the y -direction. However, only the corresponding w^j for the j th virtual node that falls inside the integral domain is employed.

5.2.1.3 Kriging interpolation

After the integral equation is discretized by the virtual nodes, the Kriging interpolation function is employed to approximate the function values at the virtual nodes, which is given in Section 3.2.2.2. The final algebraic equation system is then obtained.

5.2.1.4 Solution of the final system

Once the final algebraic system is formulated, the least-square approximation is used to solve this system. The detailed implementation is described in Section 3.2.2.4.

5.2.2 Numerical experiments with validations

Let us consider the following two numerical examples with irregular integral domains, by the presented RIQ method with different numbers of the virtual and field nodes respectively. The numerical results are summarized in the tables, and the relative errors are computed as

$$\varepsilon = \frac{\bar{e}}{\phi_{\max}^a} \quad (5.6)$$

where $\phi_{\max}^a$ is the maximum value of the analytical solution, $\bar{e}$ is the mean average error, defined as

$$\bar{e} = \sum_{i=1}^N |\phi_i^a - \phi_i^n| / N \quad (5.7)$$

where ϕ_i^a and ϕ_i^n are the analytical and numerical solutions at the i th field node, and N is the number of total field nodes.

5.2.2.1 Case one — $u(x, y) = xy - (1/8)\sin(x + y) + \int_{\Omega} \sin(x + y)u(\xi, \eta)d\xi d\eta$ defined on a triangular domain

The first example is given as

$$u(x, y) = xy - \frac{1}{8}\sin(x + y) + \int_{\Omega} \sin(x + y)u(\xi, \eta)d\xi d\eta \quad (5.8)$$

where $\Omega = \{0 \leq \xi \leq 1 \text{ and } 0 \leq \eta \leq \xi\}$. The corresponding exact solution is $u(x, y) = xy$.

In this example, the integral domain is defined as a triangle. In order to solve this problem, the virtual nodes are scattered along the straight lines in the x - or y -directions, by following the cosine distribution given by Eqs. (5.3) and (5.4), where only the nodes inside the integral domain are employed. The field nodes are scattered uniformly inside the domain and along the boundary. The obtained numerical results are summarized in Table 5.1, where the relative errors are calculated with the different numbers of the virtual and field nodes. The variations of the relative error with the number of the field nodes are plotted in Figure 5.3. The comparisons of the function values between the numerical and analytical solutions are plotted in Figure 5.4.

It is observed that the global error for a given number of the virtual nodes decreases quite fast as the number of the field nodes increases. In other words, the numerical solution converges to the analytical solution very quickly with the increase of the number of field nodes.

5.2.2.2 Case two — $5u(x, y) + \int_{\Omega} u(\xi, \eta)d\xi d\eta = 5xy^2 - 1/40 + 2/15$ defined in a complex domain

The second example is given as

$$5u(x, y) + \int_{\Omega} u(\xi, \eta) d\xi d\eta = 5xy^2 - 1/40 + 2/15 \quad (5.9)$$

where the integral domain is $\Omega = \{-1 \leq \eta \leq 1 \text{ and } \varphi_1(\eta) \leq \xi \leq \varphi_2(\eta)\}$, in which

$$\varphi_1(\eta) = \begin{cases} 0, & -1 \leq \eta \leq 0, \\ -\sqrt{\eta - \eta^2}, & 0 \leq \eta \leq 1, \end{cases} \quad (5.10)$$

and

$$\varphi_2(\eta) = \sqrt{1 - \eta^2}. \quad (5.11)$$

The corresponding exact solution is $u(x, y) = xy^2$.

In order to solve this integral equation, the virtual nodes are scattered uniformly along the straight lines covering the whole integral domain. However, only the virtual nodes inside the integral domain are employed. The field nodes are scattered in a radioactivity since the integral domain is composed of two hemicycles. Figures 5.5 and 5.6 illustrate the distributions of 549 virtual nodes and 93 field nodes, respectively. As stated before, the number of virtual nodes is required to be larger than that of the field nodes. In this example, the 549 virtual nodes are given, while the number of field nodes ranges from 16 to 93, and the numerical results are listed in Table 5.2. It is observed that the relative error becomes smaller and smaller, as the number of the field nodes increases, as shown in Figure 5.7. The comparisons between the analytical and numerical results are demonstrated in Figures 5.8 and 5.9, for 42 and 93 field nodes, where the solid dots represent the analytical solutions at the virtual nodes, and the circles and stars represent the numerical and analytical solutions at the field nodes respectively. It is found that the

RIQ method can capture the approximate function values well not only inside the integral domain, but also along the irregular boundary.

5.3 Application of the RIQ method for nonlinear integral equations

5.3.1 Solution method

In this section, the RIQ method is explored for numerical solution of nonlinear integral equations, which is based on a quasilinearization technique and the combination of the RIQ method.

5.3.1.1 Introduction of quasilinearization

As is well known, the quasilinearization technique offers an approach for approximate solutions of nonlinear differential and integral equations. When K in Eq. (5.1) is non-decreasing and satisfies the Lipschitz condition, the successive approximations method [35] yields a monotonic sequence that converges uniformly to the solution of Eq. (5.1). The quasilinearization technique was pioneered by Bellman and Kalaba [120], then generalized [121] and later extended [122]. It is often used for solving nonlinear differential equations, which offers two monotonic sequences of linear iterations converging uniformly or quadratically to the unique solution of the initial value problem,

$$\phi'(t) = h(\phi, t), \quad \text{and} \quad \phi(t_0) = c \quad (5.12)$$

This technique was extended to solve the nonlinear integral equation (5.1) by Pandit [123], when K is non-decreasing and convex in u . The iterative scheme is given as [123]

$$\phi_p(t) = f(t) + \int_0^t \left(K(t, s, \phi_{p-1}(s)) + K_\phi(t, s, \phi_{p-1}(s))(\phi_p(s) - \phi_{p-1}(s)) \right) ds, \quad p = 1, 2, \dots, \quad (5.13)$$

where $\phi_0(t)$ is a lower solution of Eq. (5.1), K is the kernel function, and K_ϕ is the first-order derivative of K over ϕ . Since the initial lower solution is given, Eq. (5.13) is actually a linear Volterra integral equation of the second kind related to ϕ_p , which can be easily solved by the random integral quadrature (RIQ) method.

5.3.1.2 Generalized quasilinearization

For $T \in \mathbb{R}$ and $T > 0$, let $J = [0, T]$ and $D = \{(t, s) \in J \times J : s < t\}$. Here we consider a nonlinear Volterra integral equation of the form

$$\phi(t) = f(t) + \int_0^t K(t, s, \phi(s)) ds \quad (5.14)$$

where $f \in C(J, \mathbb{R})$ and $K \in C(D, \mathbb{R})$.

A function $v(t)$ is an upper solution of Eq. (5.14) if

$$v(t) \geq f(t) + \int_0^t K(t, s, v(s)) ds \quad (5.15)$$

and a lower solution if the inequality sign is inversed.

The upper and lower solutions can be employed to develop a quadratically convergent iterative scheme for Eq.(5.14) [123], if the kernel function K is non-decreasing and convex in ϕ of Eq. (5.14) and satisfies the one-side Lipschitz condition

$$K(t, s, \alpha) - K(t, s, \beta) \leq L(\alpha - \beta), \quad \alpha \leq \beta, \quad L > 0 \quad (5.16)$$

Then if $v_0(0) \leq w_0(0)$, an unique solution $\phi(t)$ exists for Eq.(5.14), and

$$v_0(0) \leq u(t) \leq w_0(0) \quad (5.17)$$

Defining two iterative schemes using the lower and upper solutions gives two linear integral equations as

$$v_p(t) = f(t) + \int_0^t \left(K(t, s, v_{p-1}(s)) + K_\phi(t, s, v_{p-1}(s)) (v_p(s) - v_{p-1}(s)) \right) ds \quad (5.18)$$

and

$$w_p(t) = f(t) + \int_0^t \left(K(t, s, w_{p-1}(s)) + K_\phi(t, s, w_{p-1}(s)) (w_p(s) - w_{p-1}(s)) \right) ds \quad (5.19)$$

where $v_0(t)$ and $w_0(t) \in [J, \mathbb{R}]$ are the lower and upper solutions of Eq. (5.14) respectively, and K_ϕ is the first-order derivative of the kernel function K over ϕ . It is also known that the sequences (5.18) and (5.19) converge quadratically to the exact solution of Eq. (5.14) [123].

5.3.1.3 Development of a strategy combining quasilinearization and RIQ method for nonlinear integral equations

It is obviously observed that the iterative schemes (5.18) and (5.19) are linear Volterra integral equations of the second kind. Let us take Eq. (5.18) as an example, and rewrite it as

$$v_p(t) = F(t) + \int_0^t K_\phi(t, s) v_p(s) ds \quad (5.20)$$

where

$$F(t) = f(t) + \int_0^t \left(K(t, s, v_{p-1}(s)) + K_\phi(t, s, v_{p-1}(s)) v_{p-1}(s) \right) ds \quad (5.21)$$

In order to solve Eq. (5.20), the random integral quadrature (RIQ) method is employed, as formulated in Chapter 3. It is noticed that, during the computation, the iterative procedure is carried on after Eq. (5.21) is discretized by the generalized integral

quadrature method in order to reduce the rounding error. In other words, the Kriging interpolation, the finalization of the discrete system, and the solution of the algebraic system are implemented after the iterative procedure is finished.

5.3.2 Numerical experiments with validations

5.3.2.1 Case one — $\phi(t) = (\sin(t) - e^{\sin(t)} + 1) + \int_0^t \cos(s) e^{\phi(s)} ds$

The following nonlinear Volterra integral equation is taken below as the first example for numerical experiment [124]

$$\phi(t) = (\sin(t) - e^{\sin(t)} + 1) + \int_0^t \cos(s) e^{\phi(s)} ds \quad (5.22)$$

where $0 \leq t \leq 1$. The corresponding exact solution is $\phi(t) = \sin(t)$.

The present strategy is suitable for this nonlinear integral equation, since the kernel function K is convex and non-decreasing in ϕ , which satisfies the conditions of quasilinearization technique. The lower solution is given as $v_0(t) = 0$, and the iterative scheme is written as

$$v_p(t) = (\sin(t) - e^{\sin(t)} + 1) + \int_0^t \cos(s) e^{v_{p-1}(s)} (1 - v_{p-1}(s)) ds + \int_0^t \cos(s) e^{v_{p-1}(s)} v_p(s) ds \quad (5.23)$$

Eq. (5.23) is solved by the present strategy via 20, 30, 45 and 60 virtual nodes respectively, which are distributed in a cosine pattern. In order to reduce the rounding error, the Kriging interpolation procedure is carried out until the iteration stops. In this example, the numerical solutions converge to the exact solution after the scheme has performed two iterative steps. Figure 5.10 shows how the numerical solutions converge to the analytical solutions by the iterative scheme (5.23) with the 45 virtual nodes. It is

obvious that the iterative scheme makes the convergence to the exact solution very fast with high accuracy.

As the iterative procedure is carried out over the virtual nodes, the number of virtual nodes becomes important for the final results. Different numbers of virtual nodes are thus employed to solve this problem. For each number of the virtual nodes, the global error is calculated by Eq. (3.70). The variations of the global errors with the numbers of the virtual nodes are plotted in Figure 5.11. It is found that the global errors decrease as the number of the virtual nodes increases. After the iterative procedure stops, the 10 field nodes are distributed with the 30 virtual nodes. Table 5.3 makes the comparisons between the numerical and analytical solutions at these 10 field nodes, where the global error is equal to 0.0079 calculated by Eq. (3.70).

5.3.2.2 Case two — $\phi(t) = 2 - e^t + \int_0^t e^{t-s} \cos(s) e^{\phi(s)} ds$

The second example is taken as [124]

$$\phi(t) = 2 - e^t + \int_0^t e^{t-s} \cos(s) e^{\phi(s)} ds \quad (5.24)$$

where $0 \leq t \leq 1$. The corresponding exact solution of Eq. (5.24) is given as $\phi(t) = 1$. By the lower solution $v_0(t) = 1 - e^t$, the iterative scheme is written as

$$v_p(t) = 2 - e^t + \int_0^t e^{t-s} (v_{p-1}(s))^2 ds + \int_0^t e^{t-s} (v_{p-1}(s))^2 v_p(s) ds \quad (5.25)$$

Similarly, Eq. (5.25) is solved by the RIQ method via 20, 30, 45 and 60 virtual nodes respectively, with 10 field nodes. It is found that the numerical solutions converge to the analytical solution after 4 iterations, which is bigger than the iterative times in case one.

Probably the reason is that the exact solution for this case is a linear function, while a nonlinear function is chosen as the lower solution. Figure 5.12 shows the convergence procedure of the numerical results to the analytical solutions. It is clear that the nonlinear lower solution approaches to the linear exact solution at the fourth iteration. The global errors for different numbers of the virtual nodes have also been calculated, and they are plotted in Figure 5.13. As the number of the virtual nodes increases, the global errors decrease very fast at the beginning, and then slowly decrease to a small value. Table 5.4 shows the comparison between the numerical and analytical solutions for 10 field nodes, where the global error 0.0683 is obtained for this case.

5.3.2.3 Case three — $\phi(t) = e^{-t} + \int_0^t e^{s-t} (e^{-u(s)} + u(s)) ds$

As the third example, the integral equation is given as

$$\phi(t) = e^{-t} + \int_0^t e^{s-t} (e^{-u(s)} + u(s)) ds \quad (5.26)$$

with the exact solution $\phi(t) = \ln(t+e)$, where $0 \leq t \leq 1$. The lower solution is given by $v_0(t) = 0$, such that the iterative scheme is developed as

$$v_p = e^{-t} + \int_0^t e^{s-t} e^{-v_{p-1}(s)} (1 + v_{p-1}(s)) ds + \int_0^t e^{s-t} (-e^{-v_{p-1}(s)} + 1) v_p(s) ds \quad (5.27)$$

Eq. (5.27) is solved similarly by the RIQ method via 30, 45, 60 and 90 virtual nodes with 10 field nodes, respectively. It takes 3 iterations for the numerical results to converge to the exact solution, and the convergence procedure with 45 virtual nodes is plotted in Figure 5.14. The global errors with different virtual nodes are calculated, and they are plotted in Figure 5.15. The comparisons between the numerical and analytical

results for 10 field nodes are listed in Table 5.5. It is observed that the global error decreases with increasing the number of the virtual nodes, where it reaches 10^{-3} when the number of the virtual nodes becomes more than 45. The global error is equal to 0.0170 for this case.

5.3.2.4 Case four — $\phi(t) = 2t - (1/12)t^4 + 0.25\int_0^t (t-s)\phi^2(s)ds, \quad 0 \leq t \leq 1$

In this example, the influence of the selection of lower solutions is investigated. Consider the nonlinear Volterra integral equation

$$\phi(t) = 2t - (1/12)t^4 + 0.25\int_0^t (t-s)\phi^2(s)ds, \quad 0 \leq t \leq 1 \quad (5.28)$$

with the exact solution $\phi(t) = 2t$. The lower solutions are selected as $v_0(t) = 0$, $v_0(t) = -10$, and $v_0(t) = 1 - e^{-t}$ respectively. The iterative scheme is developed as

$$v_p = 2t - (1/12)t^4 - 0.25\int_0^t (t-s)v_{p-1}^2 ds + 0.5\int_0^t (t-s)v_{p-1}(s)v_p(s)ds \quad (5.29)$$

Similarly, Eq. (5.29) is solved numerically with the RIQ method via 60 virtual nodes with the three different lower solutions respectively. The iterative procedures for the lower solutions $v_0(t) = 0$, $v_0(t) = -10$ and $v_0(t) = 1 - e^{-t}$ are plotted in Figures 5.16, 5.17 and 5.18, respectively.

In Figure 5.16, the converged solutions are obtained after 3 iterations. The numerical results are almost the same at the second and third iterations. The numerical solutions are accurate when t is small, but not satisfactory if t becomes large. Obviously the lower solution $v_0(t) = 0$ is not satisfactory for this problem. In Figure 5.17, it takes 4 iterations through the scheme for converging to the analytical solution. The results are better than

the previous one, but still need more improvement. In Figure 5.18, it is found that the numerical results agree with the analytical solutions exactly, and it only requires one iteration to reach it. It is thus concluded that, though different lower solutions can be selected to reach the exact solution, they should be selected carefully to achieve better results and faster convergence. The comparison between the numerical and analytical results is listed in Table 5.6, with $v_0(t) = 1 - e^{-t}$ as the initial lower solution. It is noticed that the numerical results are very accurate, when compared with the exact solution, and the global error 0.0014 is achieved.

5.3.3 Numerical solution of RIQ method for Blasius boundary-layer problem

As well known, the Blasius boundary layer specifies a steady two-dimensional boundary layer that forms on a semi-infinite plate, and is held parallel to a constant unidirectional flow. In this problem the pressure is constant, and the boundary-layer equations of continuity and momentum become [125]

$$\begin{aligned}\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} &= 0 \\ u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} &= \nu \frac{\partial^2 u}{\partial z^2}\end{aligned}\tag{5.30}$$

with the boundary conditions $u = w = 0$ at $z = 0$, and $u = u_e$ at $z = \infty$. For the continuity equation, a stream function ψ exists such that

$$u = \frac{\partial \psi}{\partial z}, \quad w = -\frac{\partial \psi}{\partial x}\tag{5.31}$$

Since there is no characteristic geometric length to determine the flow, the velocity profiles for different values of the coordinates x are similar, and can be made identically by imposing the scaling factors on z , which are functions of x only. Let us introduce a non-dimensional variable t as

$$t = \sqrt{\frac{u_e}{\nu x}} z \quad (5.32)$$

and a dimensionless function $y(t)$ to make

$$\psi = \sqrt{\nu u_e x} y(t) \quad (5.33)$$

In terms of the new variables, the velocity components become

$$u = u_e y' \quad (5.34)$$

$$w = \frac{1}{2} \left(\frac{\nu u_e}{x} \right)^{1/2} (ty' - y) \quad (5.35)$$

By substituting Eqs. (5.34) and (5.35) into Eq.(5.30), the momentum equation is reduced to

$$\frac{1}{2} y \cdot y'' + y''' = 0 \quad (5.36)$$

with the boundary conditions $y(0) = y'(0) = 0$, and $y'(\infty) = 2$. As a result, the two partial differential equations given by (5.30) in x and z are reduced to a single nonlinear ordinary differential equation in t . Here we solve this problem with initial conditions [126]

$$y''' + 2yy'' = 0, \quad y(0) = y'(0) = 0, \quad y''(0) = 1 \quad (5.37)$$

5.3.3.1 RIQ solution

Let $\phi = -\ln y''$, and Eq. (5.37) is then converted into a Volterra integral equation [123]

$$\phi(t) = \int_0^t K(t, s, \phi(s)) ds \quad (5.38)$$

where the kernel function is

$$K(t, s, \phi(s)) = \begin{cases} (t-s)^2 \exp(-\phi) & \phi \geq 0 \\ (t-s)^2 & \phi < 0 \end{cases} \quad (5.39)$$

Since the kernel function is non-increasing and convex in u , the present approach can be used to solve this problem. The lower solution is given as $v_0(t) = 0$. When $\phi \geq 0$, the iterative scheme is given by

$$v_p(t) = \int_0^t (t-s)^2 \exp(-v_{p-1}(s))(1-v_{p-1}(s)) ds - \int_0^t (t-s)^2 \exp(-v_{p-1}(s)) v_p(s) ds \quad (5.40)$$

If $\phi < 0$, the iterative scheme is written as

$$v_p(t) = t^3 / 3 \quad (5.41)$$

It is observed that the iterative scheme expressed is related to the variable t when $\phi < 0$, such that we just consider the case when $\phi \geq 0$. After ϕ is obtained, we have

$$y'' = e^{-u} \quad (5.42)$$

Integrating Eq. (5.42) and imposing the initial conditions easily result in

$$y' = \int_0^t e^{-u} dt \quad (5.43)$$

By solving Eq. (5.43) via the RIQ method, y' is obtained, and the numerical results of y are computed in a similar way.

5.3.3.2 Fourth-order Runge-Kutta solution

In order to validate the results obtained by the presented method, another numerical technique is employed to solve the same boundary-layer problem, in which the high-order nonlinear differential equation is converted into a system of first-order differential equations, and Eq. (5.37) is thus rewritten as

$$\begin{aligned}u &= y', \quad u(0) = 0; \\v &= u', \quad v(0) = 1; \\v' + 2yv &= 0, \quad y(0) = 0.\end{aligned}\tag{5.44}$$

Eq. (5.44) is solved here by the fourth-order Runge-Kutta method.

5.3.3.3 Comparisons

In this section, the numerical results of y'' , y' and y are compared obtained between the present method and the fourth-order Runge-Kutta approach. The comparisons of y'' with 5, 10, 20 and 30 virtual nodes between the two methods are plotted in Figures 5.19 to 5.22 respectively. It is observed from these four figures that the numerical results of the present RIQ method agree well with those of the fourth-order Runge-Kutta method, even if the numbers of the virtual nodes are small. As the number of virtual nodes increases, the numerical results coincide with those by the fourth-order Runge-Kutta method.

The numerical results of y' are calculated by Eq. (5.43). The comparisons of y' with 5, 10, 30 and 50 virtual nodes between the two methods are plotted in Figures 5.23 to 5.26. Similarly, the numerical results of y are also computed, and the comparisons of y between the two methods are plotted in Figures 5.27 to 5.30. In the comparison of y'

and y , it is seen that the numerical results of the present RIQ method are not good if the number of virtual nodes is small such as 5. However, as the number of the virtual nodes increases, the numerical results of the present method agree well with those of the fourth-order Runge-Kutta method.

5.4 Summary

In this chapter, the random integral quadrature (RIQ) method is applied for the approximate numerical solution of the integral equations with irregular domains and for nonlinear integral equations. For implementation of the RIQ method in integral equations with irregular domains, the virtual and field nodes should be chosen carefully. The integral equation is discretized into a system of algebraic equations over all the field nodes, which are scattered within the irregular integral domain as well as along the boundary. The numerical examples reveal that the presented RIQ method can solve the integral equation defined in an irregular domain with high computational accuracy and good convergence.

The RIQ method is also applied for solving the nonlinear Volterra integral equation with the non-decreasing and convex kernel function, by the combination of the RIQ method with the quasilinearization technique. In other words, the nonlinear integral equation is converted to a linear iterative scheme through the quasilinearization technique first, and then solved by the RIQ method. Several numerical examples are investigated in detail. It is shown that the present numerical solutions converge to the analytical solution with fast convergence and good accuracy. This approach is also applied to analysis of the numerical solution of Blasius boundary-layer problem, by

converting the nonlinear differential governing equation into a Volterra nonlinear integral equation. The numerical results are examined by comparison with those via the explicit Runge-Kutta technique, and good agreement is achieved.

Table 5.1. Variation of the global errors for the example case shown in Section 5.2.2.1.

Number of the virtual nodes (M)	Number of the field nodes (N)	Global error (ε)
465	15	2.550E-2
465	36	1.320E-2
465	55	1.180E-2
465	91	8.800E-3
465	120	8.700E-3

Table 5.2. Variation of the global errors for the example case shown in Section 5.2.2.2.

Number of the virtual nodes (M)	Number of the field nodes (N)	Global errors(ε)
549	16	5.687E-2
549	32	3.712E-2
549	42	2.270E-2
549	66	1.175E-2
549	79	8.514E-3
549	93	5.753E-3

Table 5.3. Comparisons between the numerical and analytical solutions for the example case shown in Section 5.3.2.1 with the 30 virtual nodes and 10 field nodes.

t_i	Exact solution	Numerical solution	Relative error
0.0000	0.0000	0.0000	0.0000
0.1529	0.1523	0.1579	0.0360
0.3987	0.3882	0.3897	0.0015
0.6287	0.5881	0.5927	0.0078
0.7906	0.7108	0.7169	0.0086
1.0276	0.8561	0.8691	0.0152
1.2355	0.9443	0.9286	0.0166
1.3926	0.9842	0.9932	0.0091
1.4825	0.9961	0.9909	0.0052
1.5	0.9975	0.9984	0.0009

Table 5.4. Comparisons between the numerical and analytical solutions for the example case shown in Section 5.3.2.2 with the 30 virtual nodes and 10 field nodes.

t_i	Exact solution	Numerical solution	Relative error
0.0000	1.0000	0.9980	0.0020
0.1020	1.0000	0.9922	0.0078
0.2194	1.0000	0.9827	0.0173
0.3149	1.0000	0.9749	0.0251
0.4191	1.0000	0.9789	0.0211
0.5271	1.0000	0.9988	0.0012
0.6338	1.0000	0.9946	0.0054
0.7342	1.0000	1.0068	0.0068
0.8237	1.0000	1.0078	0.0078
1.0000	1.0000	1.0311	0.0311

Table 5.5. Comparisons between the numerical and analytical solutions for the example case shown in Section 5.3.2.3 with the 45 virtual nodes and 10 field nodes.

t_i	Exact solution	Numerical solution	Relative error
0.0000	1.0000	1.0011	0.0011
0.1496	1.0536	1.0554	0.0017
0.3005	1.1049	1.1083	0.0031
0.4879	1.1651	1.1723	0.0062
0.5906	1.1966	1.1214	0.0062
0.7500	1.2437	1.2511	0.0059
1.0121	1.3165	1.3223	0.0044
1.1995	1.3655	1.3751	0.0070
1.4083	1.4174	1.4236	0.0044
1.5000	1.4394	1.4429	0.0024

Table 5.6. Numerical and analytical solutions for the example case shown in Section 5.3.2.4 with the 30 virtual nodes and 10 field nodes.

t_i	Exact solution	Numerical solution	Relative error
0.0000	0.0000	0.0000	0.0000
0.1020	0.2040	0.2039	0.0005
0.2194	0.4388	0.4388	0.0000
0.3149	0.6298	0.6300	0.0003
0.4191	0.8382	0.8384	0.0002
0.5271	1.0542	1.0544	0.0001
0.6338	1.2676	1.2677	0.0001
0.7342	1.4684	1.4678	0.0004
0.8237	1.6474	1.6456	0.0011
1.0000	2.0000	1.9915	0.0043

Figure 5.1. Irregular integral domain.

Figure 5.2. Distribution of the field and virtual nodes.

Figure 5.3. Variation of the relative errors with the number of the field nodes for the example case shown in Section 5.2.2.1.

Figure 5.4. Comparison between the numerical and analytical solutions with the 120 field nodes for the example case shown in Section 5.2.2.1.

Figure 5.5. Distribution of the 549 virtual nodes for the example case shown in Section 5.2.2.2.

Figure 5.6. Distribution of the 93 field nodes for the example case shown in Section 5.2.2.2.

Figure 5.7. Variation of the relative errors with the number of the field nodes for the example case shown in Section 5.2.2.2.

Figure 5.8. Comparison between the analytical and numerical solutions with 42 field nodes for the example case shown in Section 5.2.2.2.

Figure 5.9. Comparison between the numerical and analytical solutions with 93 field nodes for the example case shown in Section 5.2.2.2.

Figure 5.10. Convergence of the iterative scheme over 45 virtual nodes to the analytical solution for the example case shown in Section 5.3.2.1.

Figure 5.11. Variations of the global errors with the number of the virtual nodes for the example case shown in Section 5.3.2.1.

Figure 5.12. Convergence of the iterative scheme over 45 virtual nodes to the exact solution for the example case shown in Section 5.3.2.2.

Figure 5.13. Variations of the global errors with the number of virtual nodes for the example case shown in Section 5.3.2.2.

Figure 5.14. Convergence of the iterative scheme over 45 virtual nodes to the exact solution for the example case shown in Section 5.3.2.3.

Figure 5.15. Variations of the global errors with the number of virtual nodes for the example case shown in Section 5.3.2.3.

Figure 5.16. Convergence of the iterative scheme over 60 virtual nodes to the exact solution with lower solution as $v_0 = 0$ for the example case shown in Section 5.3.2.4.

Figure 5.17. Convergence of the iterative scheme over 60 virtual nodes to the exact solution with lower solution as $v_0 = -10$ for the example case shown in Section 5.3.2.4.

Figure 5.18. Convergence of the iterative scheme over 60 virtual nodes (iteration once only) to the exact solution with lower solution as $v_0 = 1 - e^{-t}$ for the example case shown in Section 5.3.2.4.

Figure 5.19. Comparisons of the numerical results of y'' between the present RIQ method over 5 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.20. Comparisons of the numerical results of y'' between the present RIQ method over 10 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.21. Comparisons of the numerical results of y'' between the present RIQ method over 20 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.22. Comparisons of the numerical results of y'' between the present RIQ method over 30 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem..

Figure 5.23. Comparisons of the numerical results of y' between the present RIQ method over 5 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.24. Comparisons of the numerical results of y' between the present RIQ method over 10 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.25. Comparisons of the numerical results of y' between the present RIQ method over 30 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.26. Comparisons of the numerical results of y' between the present RIQ method over 50 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.27. Comparisons of the numerical results of y between the present RIQ method over 5 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.28. Comparisons of the numerical results of y between the present RIQ method over 10 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.29. Comparisons of the numerical results of y between the present RIQ method over 30 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Figure 5.30. Comparisons of the numerical results of y between the present RIQ method over 50 virtual nodes and the Runge-Kutta technique for Blasius boundary-layer problem.

Chapter 6 Application of RIQ Method for Peridynamics

Problems

6.1 Introduction

The classical theory of solid mechanics is generally based on the assumption that mass is continuously distributed within a body, and that all internal forces are contact forces that act across zero distance. Fundamental physical laws such as the conservation of mass, momentum and energy are applicable to such models, in order to derive governing differential equations describing the behavior of such objects. The classical theory has been validated to be accurate when the length scale is much larger than the inter-atomic distance. However, if the developed technology involves the design and fabrication of devices at smaller and smaller length scales, or when cracks and other discontinuities occur, the partial differential governing equations of the classical theory cannot be employed directly. Therefore, a formulation of continuum mechanics called peridynamics is proposed to treat these discontinuities such as fractures. The governing equations of the peridynamic theory are in the form of integro-differential equations, which can be solved by the RIQ method.

A number of works have been published to investigate various aspects of peridynamic theory. Silling and his group investigated the deformation of a peridynamic bar in 2003 [14], in which a linear Fredholm integral equation was used to describe an infinite bar

subjected to a self-equilibrated load, and it was solved by the Fourier transform method. Later on, they proposed a meshless method to solve the peridynamic model of solid mechanics [127], where the dynamic problems were studied with the prototype microelastic brittle (PMB) material model. Apart from the PMB material, they also applied the peridynamic theory to model membranes and fibers [128], as well as viscoplasticity [129]. The convergence of peridynamic theory to the classical elasticity theory was discussed [130], showing that the peridynamic stress tensor converged in the limit to a Piola-Kirchhoff stress tensor as the length scale approached to zero. Silling and his group also made other contributions in this area [131-134].

Peridynamics constantly attracts attention for analysis of fractures and cracks. Ha and Bobaru [135] studied dynamic crack propagation and crack branching with peridynamics in brittle materials, and investigated the convergence upon uniform grid refinement and upon decreasing the peridynamic horizon. In 2009, Bobaru studied the convergence, the adaptive refinement and the scaling in peridynamics using a one-dimensional self-equilibrant bar [136]. Three types of numerical convergence for peridynamics were discussed, and the uniform convergence to the classical solutions of static and dynamic elasticity problems were obtained in 1-D case with the horizon approaching to zero. The extreme load on the reinforced concrete structures using peridynamics were considered by Paul [137], in which the applications of peridynamics in composite materials, fluids and explosives were also discussed. Kikic applied the peridynamics for analysis of the elastic stability and failure of simple structures to determine buckling characteristics [138].

Peridynamics was also explored by several classical numerical methods. For example, a molecular dynamics code was developed to simulate on mesoscopic or even macroscopic length and time scales [139]. The finite element analysis (FEA) was also employed to analyze peridynamic models by Macek and Silling [140], where the results were compared with those by a meshless code called EMU. In addition, the discontinuous Galerkin method was reformulated and applied to peridynamic theory by Aksoy and Senocak [141].

6.2 Formulation of Peridynamics theory

6.2.1 Governing equation

As stated in Section 6.1, the peridynamic theory is developed to meet the needs of those problems, which attempt to treat the evolution of discontinuities according to the same governing equations for the continuous deformations. The main difference of peridynamic theory with classical continuous mechanics is that the governing equation is formulated in terms of integral equations, instead of partial differential equations including derivatives of the displacement components. This allows the peridynamic formulation to hold everywhere whether the discontinuity occurs or not. The basic governing equation of the peridynamics at a reference configuration $\mathbf{x}$ and time t is given by [142]

$$\rho(\mathbf{x}) \frac{\partial^2 \mathbf{u}(\mathbf{x}, t)}{\partial t^2} = \int_{R(\mathbf{x})} \mathbf{f}(\mathbf{\eta}, \xi) dV_{\mathbf{x}} + \mathbf{b}(\mathbf{x}, t) \quad (6.1)$$

where $\mathbf{x}$ is a point in the body, t is the time, $\mathbf{u}(\mathbf{x}, t)$ is the displacement field vector, ρ is the mass density in the un-deformed body, $R(\mathbf{x})$ is the integral domain, $dV_{\mathbf{x}'}$ is the integration variable that indicates an infinitesimal domain located at the point $\mathbf{x}'$, and $\mathbf{b}$ is the prescribed loading force density, which represents the external force per unit volume. In addition, ξ and η are defined as

$$\xi = \mathbf{x}' - \mathbf{x}, \quad \eta = \mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t) \quad (6.2)$$

$\mathbf{f}(\eta, \xi)$ in Eq. (6.1) is called the response function, and it represents the force density that $\mathbf{x}'$ exerts on $\mathbf{x}$, which contains all the constitutive / material-dependent properties.

6.2.2 Response function

The particles interact on each other directly through the response function. Therefore, all constitutive information is contained in the function $\mathbf{f}$. As per Newton's third law, an important restriction is imposed on $\mathbf{f}$ as [142]

$$\mathbf{f}(-\eta, -\xi) = -\mathbf{f}(\eta, \xi) \quad \forall \eta, \xi \quad (6.3)$$

which is called the linear admissibility condition on $\mathbf{f}$. Another angular admissibility condition arising from the conservation of angular momentum is written as

$$(\xi + \eta) \times \mathbf{f}(\eta, \xi) = 0 \quad \forall \xi, \eta \quad (6.4)$$

The response function is influenced by the relative position and the relative displacement, such that it is convenient to focus on the forces arising from the surrounding body in a finite radius. In order to describe the influence area more precisely, it is assumed that a horizon δ exists, such that $\mathbf{f}(\eta, \xi) = 0$ for any η whenever $|\xi| > \delta$.

There is a connection between peridynamics and classical elasticity theory by considering a special form of the response function $\mathbf{f}$ [14],

$$f(\xi, \eta) = -2\sigma\left(\frac{\eta}{\xi}\right) \frac{d\delta_l(\xi)}{d\xi} \quad (6.5)$$

where σ is a continuously differentiable function in $(-\infty, \infty)$. δ_l represents a generalized Dirac delta function parameterized by l . Let us consider the one-dimensional peridynamic equation as

$$\rho \frac{\partial^2 u(x, t)}{\partial t^2} = \int_{-\infty}^{\infty} dx' f(\xi, \eta) \quad (6.6)$$

By substituting Eq. (6.5) into Eq.(6.6), the peridynamic theory is reduced to the classical theory of elasticity as

$$\begin{aligned} \rho \frac{\partial^2 u(x, t)}{\partial t^2} &= -2 \lim_{l \rightarrow 0} \int_{-\infty}^{\infty} dx' \sigma\left(\frac{\eta}{\xi}\right) \frac{d\delta_l(\xi)}{d\xi} \\ &= -2 \lim_{l \rightarrow 0} \int_{-\infty}^{\infty} d\xi \sigma\left(\frac{u(x+\xi, t) - u(x, t)}{\xi}\right) \frac{d\delta_l(\xi)}{d\xi} \\ &= 2 \lim_{l \rightarrow 0} \frac{d}{d\xi} \sigma\left(\frac{u(x+\xi, t) - u(x, t)}{\xi}\right) = \frac{d\sigma(du/dx)}{du} \frac{d^2 u}{dx^2} = \frac{d\sigma(\varepsilon)}{d\varepsilon} \end{aligned} \quad (6.7)$$

where σ and ε are the stress and strain fields respectively. The subscript l in Eq. (6.5) represents a length scale, and the classical theory of elasticity is then recovered by the vanishing of the length scale. The parameter l is called the internal length here.

All the material information is contained in the response function, which relates the particle forces with the relative positions and displacements of the particles. As per the method proposed by Silling and Askari [128], the response function of a microelastic material model is derived via a scalar micropotential w as

$$f(\eta, \xi) = \frac{\partial w}{\partial \eta}(\eta, \xi) \quad \forall \xi, \eta \quad (6.8)$$

The micropotential is the strain energy in a single bond with the dimensions of energy per unit volume squared. For a given node $\mathbf{x}$, a bond means the physical interaction (without specified means) between the particle $\mathbf{x}$ and the neighborhood particle $\mathbf{x}'$ inside the horizon δ . The energy density at a given node in the body is then written by [128]

$$W = \frac{1}{2} \int_R w(\eta, \xi) dV_\xi \quad (6.9)$$

By equating the strain energy density of the classical continuum theory to the corresponding energy in peridynamic theory, the response function derived for the microelastic material is given as [128]

$$\mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \begin{cases} \frac{\boldsymbol{\xi} + \boldsymbol{\eta}}{|\boldsymbol{\xi} + \boldsymbol{\eta}|} \mu(t, \boldsymbol{\xi}) \frac{18k}{\pi \delta^4} s(\boldsymbol{\xi}, \boldsymbol{\eta}) & \text{for } |\boldsymbol{\xi}| \leq \delta \\ 0 & \text{for } |\boldsymbol{\xi}| > \delta \end{cases} \quad (6.10)$$

where s is the stretch, and is defined as

$$s(\boldsymbol{\xi}, \boldsymbol{\eta}) = \frac{|\boldsymbol{\xi} + \boldsymbol{\eta}| - |\boldsymbol{\xi}|}{|\boldsymbol{\xi}|} \quad (6.11)$$

and the function $\mu(t, \boldsymbol{\xi})$ is a function, defined as

$$\mu(t, \boldsymbol{\xi}) = \begin{cases} 1 & \text{if } s(t', \boldsymbol{\xi}) < s_0 \quad \text{for all } 0 \leq t' \leq t \\ 0 & \text{otherwise} \end{cases} \quad (6.12)$$

in which s_0 is the critical stretch for bond failure as shown in Figure 6.1, and the bond force becomes zero if the bond fails.

During the body deformation, some bonds may break when the stretch between two nodes exceeds the critical stretch, and failure occurs. The two nodes then cease to

interact with each other. When the broken interactions align themselves along a surface, a crack has formed. However, the governing integral equation is still valid for the crack but is discontinuous. Therefore, the damage is treated as part of the constitutive model in the peridynamic theory though the irreversible breakage of interactions exists, which is different from traditional fracture mechanics. As described by Silling and Askari [127], the introduction of failure at bond level leads to an unambiguous notion of local damage at a point, which is defined as

$$\varphi(\mathbf{x}, t) = 1 - \frac{\int_{R(\mathbf{x})} \mu(\mathbf{x}, t, \xi) dV_{\mathbf{x}'}}{\int_{R(\mathbf{x})} dV_{\mathbf{x}'}} \quad (6.13)$$

It is obvious that $0 \leq \varphi \leq 1$, and the point $\mathbf{x}$ is fully connected with other points at $\varphi = 0$, while the point $\mathbf{x}$ is completely disconnected with the other points at $\varphi = 1$.

Another response function is proposed in a similar way to represent the effect of thermal loading and the distance between the points as [138]

$$\mathbf{f}(\boldsymbol{\eta}, \xi) = \frac{\xi + \boldsymbol{\eta}}{|\xi + \boldsymbol{\eta}|} \mu(t, \xi) \frac{9k}{\pi l^4} e^{-(|\xi|/l)^2} (s - \alpha\theta) \quad (6.14)$$

where θ is the instantaneous temperature difference between the material points $\mathbf{x}'$ and $\mathbf{x}$. k is the bulk modulus of the material, which is expressed in terms of the elastic modulus E and Poisson's ratio ν as

$$k = \frac{E}{3(1-2\nu)} \quad (6.15)$$

α in Eq. (6.14) denotes the coefficient of thermal expansion of the material, and l in Eq. (6.14) is the internal length, which is the measure of the nonlocal behavior. The

interactions become more local as l decreases. Since Eq. (6.14) includes the effect of thermal loading, the function μ is modified to [138]

$$\mu(t, \xi) = \begin{cases} 1 & \text{if } s(t', \xi) - \alpha\theta < s_0 \quad \text{for all } 0 \leq t' \leq t \\ 0 & \text{otherwise} \end{cases} \quad (6.16)$$

where s_0 still is the critical stretch when failure occurs. The weighted local damage introduced in Eq. (6.13) is also modified as [138]

$$\varphi(\mathbf{x}, t) = 1 - \frac{\int_{R(\mathbf{x})} \mu(t, \xi) e^{-\left(\frac{|\xi|}{l}\right)^2} dV_{\mathbf{x}'}}{\int_{R(\mathbf{x})} e^{-\left(\frac{|\xi|}{l}\right)^2} dV_{\mathbf{x}'}} \quad (1.17)$$

Since there are various response functions for the governing equations of peridynamics, it is specified that Eq. (6.14) is employed as the present response function in this study.

6.3 RIQ method for Peridynamics

The peridynamic theory is characterized by integral equations, it can therefore be solved by the random integral quadrature (RIQ) method. As mentioned in Eq.(6.1), the governing equation is given as

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \int_R \mathbf{f}(\mathbf{u}(\mathbf{x}, t), \mathbf{u}(\mathbf{x}', t), \mathbf{x}, \mathbf{x}', t) dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}, t) \quad (6.18)$$

where $\mathbf{u}$ is the displacement vector, R is the integral domain, $\mathbf{b}$ is a prescribed body force density field, and ρ is the mass density. The response function $\mathbf{f}$ is defined as Eq. (6.14) given by Kilic [138]. In order to solve Eq. (6.18), the RIQ method is performed as follows.

6.3.1 Discretization of governing equation with the GIQ method

If we have NP virtual nodes that are distributed in the whole domain and represented by $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{NP}$, the integral governing equation is discretized as

$$\rho \ddot{\mathbf{u}}(\mathbf{x}^i, t) = \sum_{j=1}^{NP} w^j f(\mathbf{u}(\mathbf{x}^i, t), \mathbf{u}(\mathbf{x}^j, t), \mathbf{x}^i, \mathbf{x}^j, t) + \mathbf{b}(\mathbf{x}^i, t) \quad (6.19)$$

where $\mathbf{x}^i$ is the position vector at the i^{th} virtual node, $\mathbf{x}^j$ is the j^{th} virtual node around the node $\mathbf{x}^i$, and w^j is the weighting coefficient for the generalized integral quadrature (GIQ) method associated with the node $\mathbf{x}^j$. In order to solve Eq. (6.19), an implicit time integration scheme is employed. However, it may cause difficulty in the simulations subjected to quasi-static conditions as well as stability. An extension of the dynamic relaxation method is thus introduced to obtain the steady-state solutions of peridynamic equations.

6.3.2 A numerical strategy – dynamic relaxation

In order to analyze various nonlinear engineering problems, a stable and efficient numerical approach named the dynamic relaxation (DR) has been developed [143], which is based on the fact that a system undergoing the damped vibration ultimately comes to the static equilibrium position.

This method can be interpreted in both mathematical and physical ways. Physically, the procedure is similar to obtaining the steady state of a dynamical system. In order to obtain the static solution of the dynamic problem characterized by the differential equation associated with time, the differential equation is transferred to a fictitious

dynamic space, with the solution as $t \rightarrow \infty$. As such, an artificial damping is introduced to guide the solution into the steady-state regime as soon as possible. Mathematically, the DR method can be generated by the second-order Richardson rule.

In the first decade of the twentieth century, the dynamic relaxation method was used as an iterative procedure to solve large system of equations, derived from the finite difference method [144]. Underwood reviewed the dynamic relaxation techniques and proposed the adaptive dynamic relaxation method [145], in which the integration parameter may be changed via step by step, such as the pseudo-time step size as well as the mass and damping coefficients. In this chapter, this adaptive dynamic relaxation method is employed to solve Eq. (6.19).

By adding the artificial inertia and damping forces, Eq. (6.19) is transformed to the artificial dynamic space as follows,

$$\mathbf{M}\ddot{\mathbf{U}}(\mathbf{X},t) + c\mathbf{M}\dot{\mathbf{U}}(\mathbf{X},t) = \mathbf{F}(\mathbf{U}, \mathbf{U}', \mathbf{X}, \mathbf{X}') \quad (6.20)$$

where $\mathbf{M}$ is called the pseudomass, which is a fictitious diagonal density matrix. c is the damping coefficient, whose value is determined through Greschgorin's theorem and Reyleigh's quotient. $\mathbf{X}$ and $\mathbf{U}$ are the vectors of positions and relative displacements, and given as

$$\mathbf{X}^T = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{NP}] \quad (6.21)$$

and

$$\mathbf{U}^T = [\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^{NP}] \quad (6.22)$$

where NP is the number of total virtual nodes. $\mathbf{F}$ in Eq. (6.20) is the summation of the internal and external forces given by $[\mathbf{F}^1, \mathbf{F}^2, \dots, \mathbf{F}^{NP}]$, and the i^{th} component of $\mathbf{F}$ is written as

$$\mathbf{F}^i = \sum_{j=1}^{NP} w^j f(\mathbf{u}(\mathbf{x}^i, t), \mathbf{u}(\mathbf{x}^j, t), \mathbf{x}^i, \mathbf{x}^j, t) + b(\mathbf{x}^i, t) \quad (6.23)$$

Eq. (6.20) is treated with the half station form of the central difference integrator as follows [138],

$$\begin{aligned} \dot{\mathbf{U}}^{k+1/2} &= \dot{\mathbf{U}}^{k-1/2} + \Delta t \ddot{\mathbf{U}}, \\ \dot{\mathbf{U}}^{k+1} &= \dot{\mathbf{U}}^k + \Delta t \dot{\mathbf{U}}^{k+1/2} \end{aligned} \quad (6.24)$$

By inserting Eq. (6.24) into Eq.(6.20), the following schemes are obtained,

$$\dot{\mathbf{U}}^{k+1/2} = \frac{2-c\Delta t}{2+c\Delta t} \dot{\mathbf{U}}^{k-1/2} - \frac{2\Delta t}{2+c\Delta t} \mathbf{M}^{-1} \mathbf{F}^k \quad (6.25)$$

and

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \Delta t \dot{\mathbf{U}}^{k+1/2} \quad (6.26)$$

where the superscript k indicates the k^{th} iteration. However, the iteration cannot start directly via Eqs. (6.25) and (6.26), since the velocity field at $t = -1/2$ is unknown. Therefore, the starting procedure requires some arrangements. The initial conditions are given by $\mathbf{U}^0 \neq 0$ and $\dot{\mathbf{U}}^0 = 0$, and the velocities in the scheme (6.25) are rewritten as

$$\dot{\mathbf{U}}^{1/2} = \Delta t \mathbf{M}^{-1} \mathbf{F}^0 / 2 \quad (6.27)$$

The formulation of the DR scheme is now complete. Only the summation of the external and internal forces $\mathbf{F}$ is the physical term in this iterative scheme. For the practical implementation of this method, the density matrix $\mathbf{M}$, the damping coefficient c

and the time step size Δt need to be chosen carefully to ensure stability and faster convergence.

There are several ways to identify the density matrix, and the most efficient one is given by Underwood [145], and written as [143]

$$\mathbf{M}_{ii} \geq \frac{1}{4} \Delta t^2 \sum_j K_{ij} \quad (6.28)$$

where K_{ij} is the elements of the stiffness matrix $\mathbf{K}$, and the absolute sum of the rows for the stiffness matrix is written as

$$\sum_j |K_{ij}| = \max \left(\sum_{j=1}^{NP} \text{abs}(\xi) w^j \frac{18k}{\pi l^4} e^{-(|\xi|/l)^2} \frac{1}{|\xi|^2} \right) \quad (6.29)$$

where $\max(\cdot)$ produces the maximum value of the three-dimensional vector. As a result, the density matrix may be changed during the process of the scheme to achieve faster convergence.

Another important factor is the damping coefficient, which is to guarantee the convergence and stability of the analysis procedure. Underwood suggested the use of the lowest frequency of the system [145], which is calculated by utilizing Rayleigh theory as

$$c_k = 2 \sqrt{\left((\mathbf{U}^k)^T \mathbf{K}^k \mathbf{U}^k \right) / \left((\mathbf{U}^k)^T \mathbf{U}^k \right)} \quad (6.30)$$

where c_k is the damping coefficient at the k^{th} time step. $\mathbf{K}^k$ is the diagonal local stiffness matrix given by

$$K_{ii}^k = - \left(F_i^k / M_{ii} - F_i^{k-1} / M_{ii} \right) / \left(\Delta t \dot{u}_i^{k-1/2} \right) \quad (6.31)$$

It is obvious from this formula that the damping coefficient is fixed at each time step. Since the velocity is involved in the division term in Eq. (6.31), it is important to ensure

the nonzero value of the velocity, such that the solution starts with zero displacement field to avoid division by zero. The velocity is also monitored during the iterative procedure. The local stiffness matrix is set to be equal to zero once the velocity is found to be zero.

The time step has a significant effect on the numerical stability and the convergence rate of the dynamic relaxation procedure. Underwood has suggested the value of 1 or 1.1 for the time step Δt [145]. In fact, using a constant slightly larger than 1 can ensure the stability and convergence.

6.3.3 Kriging interpolation

After the dynamic relaxation scheme is performed to obtain the numerical solution over the virtual nodes, the Kriging interpolation function is employed to approximate the function values at the virtual nodes. The Kriging interpolation function may also be employed during the iterative procedure of the dynamic relaxation. However, the computational error may increase because of the large iterative times.

6.3.4 Solution of the final system

A system of algebraic equations is achieved after the Kriging interpolation. Solving this system by the method described in Section 3.2.2.4 gives the final numerical solutions.

6.4 Numerical case studies for comparisons between RIQ and FE methods

As mentioned in the section 6.3, the random integral quadrature method is combined with the dynamic relaxation in the appropriate procedures to solve peridynamic problems. The effectiveness of the present method is validated by the following case studies of peridynamic problems, where the numerical results are compared with those by the finite element method.

6.4.1 Beam subject to tension

A rectangular beam subjected to tension is considered in this case. The dimensions of the beam are given as $L=10$ (in), $w=0.2$ (in) and $h=0.2$ (in), as shown in Figure 6.2. The origin of the coordinate system is located at the center of the beam. The beam is subjected to an axial force of 1000 lb in the x -direction. The isotropic material properties of the beam are assumed with Yong's modulus $E=10^7$ (psi) and Poisson's ratio $\nu=0.25$, respectively.

The peridynamic model is constructed as

$$\rho \ddot{\mathbf{u}} = \int_{\mathbf{R}} \mathbf{f}(\mathbf{u}(\mathbf{x}, t), \mathbf{u}(\mathbf{x}', t), \mathbf{x}, \mathbf{x}', t) dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x}, t) \quad (6.32)$$

where $\mathbf{u}$ and $\mathbf{x}$ are the vectors of the displacement and coordinate of the point within the beam, respectively. The beam can be discretized as a one- or three-dimensional problem, such that both one- and three-dimensional models are constructed, respectively.

Let us scatter NP virtual nodes along the x -direction according to the cosine distribution, i.e.

$$x^i = \left(1 - \cos\left(\frac{i-1}{NP-1}\pi\right)\right) \times L/2 + a \quad (6.33)$$

where $i = 1, 2, \dots, NP$. L is the length, and a is the coordinate of the left end of the beam.

In the one-dimensional case, the peridynamic model is therefore discretized as

$$\rho \ddot{u}(x^i, t) = A \sum_{j=1}^{NP} w^j f(u(x^i, t), u(x^j, t), x^i, x^j, t) + b(x^i, t) \quad (6.34)$$

where w^j is the one-dimensional weighting coefficient calculated by the technique introduced in Section 3.2.2.1. Since the integral term is based on the volume in Eq. (6.32), the weighting coefficient should be multiplied by the area A when the integral term is calculated. A is taken to be 0.04 in this example. The tensile load cannot be imposed directly in peridynamics, such that the uniaxial tension is loaded as the body force over small volumes at both ends of the beam. The volume is defined as the multiplication of the cross-section area by the distance of the two nodes at the edge. For example, the value of the body force in this beam case is equal to $1000 / (A \times (x^2 - x^1))$, where x^2 and x^1 are the first and second virtual nodes distributed in the x -direction.

In this study, 51 virtual nodes are distributed in the x -direction of the beam. The corresponding body force is equal to 2.5339×10^6 (lb/in³). $l = 0.43$ is chosen. The variations of the velocities and displacements at $x = 0$, 2.4008 (in) and 4.9606 (in) are plotted in Figures 6.3 and 6.4 respectively. It is seen that it takes about 600 iterations to reach the static state with 51 virtual nodes. The numerical results are compared with those by the finite element method (FEM) in Figure 6.5. It is obvious that the numerical results of the RIQ method agree well with those by FEM. However, the numerical results

near the boundary are slightly inaccurate. The possible reason may be that the nodes near the boundary are much denser than those in the middle by using the cosine distribution. As a result, the response function cannot catch the force accurately near the boundary.

The same beam is also discretized by 101 virtual nodes. The corresponding body force is 1.0133×10^7 (lb/in³). The variations of the velocities and displacements at $x = 4.9778$ (in), 2.5452 (in) and 0 are plotted in Figures 6.6 and 6.7. It is found that the numerical results converge well after 800 iterations. After comparison with FEM, it is seen from Figure 6.8 that the numerical results agree well with those obtained with FEM, and they are more accurate than those with 51 virtual nodes.

The three-dimensional peridynamic model is also solved by the RIQ method, where the NPX , NPY and NPZ virtual nodes are distributed in x -, y - and z -directions, and Eq. (6.32) at the node (x^i, y^j, z^k) is discretized as

$$\rho \ddot{\mathbf{u}}(x^i, y^j, z^k, t) = \sum_{n=1}^{NPX} \sum_{m=1}^{NPY} \sum_{p=1}^{NPZ} w^n w^m w^p \mathbf{f}(\mathbf{u}^{nmp}, \mathbf{u}^{ijk}, \mathbf{x}^{nmp}, \mathbf{x}^{ijk}, t) + \mathbf{b}(\mathbf{x}^{ijk}, t) \quad (6.35)$$

where

$$\begin{aligned} \mathbf{u}^{nmp} &= \mathbf{u}(x^n, y^m, z^p, t); & \mathbf{u}^{ijk} &= \mathbf{u}(x^i, y^j, z^k, t); \\ \mathbf{x}^{nmp} &= \mathbf{x}(x^n, y^m, z^p); & \mathbf{x}^{ijk} &= \mathbf{x}(x^i, y^j, z^k); \end{aligned} \quad (6.36)$$

w^n , w^m and w^p in Eq. (6.35) are the weighting coefficients in the x -, y - and z -directions, respectively. The problem is discretized with 51, 3 and 3 virtual nodes in x -, y - and z -directions, respectively.

Figures 6.9 and 6.10 demonstrate the variations of the velocities and displacements in x -direction with the number of the iterations at the three nodes $x = 4.9778$ (in),

2.5452 (in) and 0, respectively. It is observed that the numerical results converge to the static state after about 150 iterations, which is faster than that of the one-dimensional model. Figure 6.11 displays the comparisons between the RIQ and FEM results, showing that the numerical results of the RIQ method agree well with the FEM results. Figure 6.12 is plotted for the corresponding contour of the RIQ results in the 3D model, which clearly demonstrates the deformations of the beam subjected to the tensile loading.

6.4.2 Beam subject to bending

In this case study, a cantilever beam is considered with a vertical loading at the free end. This is a bending problem, and the transmutation of the beam is nonlinear. The dimensions of the beam are given as: $L=10$ (in) , $w=0.2$ (in) and $h=0.2$ (in). The vertical force $F=20$ (lb) is imposed at the free end of the beam, as shown in Figure 6.13.

This is treated as a three-dimensional problem and discretized by 41, 3 and 3 virtual nodes in x -, y - and z -directions, respectively. Since the external force cannot be applied directly in peridynamics, it is converted into a body force in a small volume at the free end of the beam. The small volume is specified as 0.0616 (in) $\times$ 0.04 (in²) = 0.002464 (in³) , where 0.0616 (in) is the distance between the two nodes at the free end of the beam, and 0.04 (in²) is the area of the cross section. The left end of the beam is fixed, such that the boundary condition of the beam is represented by $\mathbf{U}^i = [0,0]$, where $\mathbf{U}^i$ is the vector of the displacement of the node on the left cross section. $l=0.3$ is chosen in this case study.

The variations of the velocities in x - and y - directions are plotted in Figures 6.14 and 6.15 respectively. It is seen that the velocities converge to zero at around 7×10^4 iterations, which is much larger than that for the beam subjected to tension. This is because the bending problem is nonlinear, and the system is heavily over-damped with the dynamic relaxation technique. As such, the convergence procedure for the present bending case is much slower than that for the tensile problem. The variations of the displacements in x - and y - directions are plotted in Figures 6.16 and 6.17. The comparisons of the vertical and horizontal displacements between the RIQ method and the FEM are plotted in Figures 6.18 and 6.19. It is observed that the differences of the two numerical results become slightly large for this nonlinear case. Probably the reason is that the Poisson ratio of 0.25 cannot be captured exactly for thin structures.

6.4.3 Rod subject to tension

A rod subjected to a tensile load in the x -direction is considered as a three-dimensional peridynamic model with a circular cross-sectional area. The radius of the circular cross section is $r = 0.25$ (in), the rod length is $L = 10$ (in), and the tensile load is $F = 1000$ (lb), as shown in Figure 6.20. The isotropic material properties of the rod is given by the Young's modules $E = 10^7$ (psi) and Poisson's ratio $\nu = 0.25$, respectively.

The three-dimensional peridynamic model is discretized by Eqs. (6.35) and (6.36). For the circular cross-section area, the nodes are selected as stated in Section 5.2.1.1. The 7 nodes in the y -direction and 7 nodes in the z -direction are scattered to discretize the rectangular area that covers the circle, in which only the nodes fallen inside the circular area are selected. As a result, 25 nodes are used in total to discretize the circular cross-

section area, as shown in Figure 6.21. The domain in the x -direction is discretized with 51 nodes. Since the external force cannot be applied to the peridynamic model directly, it is imposed as the body force at the ends of the rod, and calculated by $b_x = F/((x^2 - x^1) \times A)$. In this case $b_x = 5.1619 \times 10^5$ (lb/in³).

The problem is solved by the present method with $l = 0.35$. The variations of the velocity and displacement are plotted in Figures 6.22 and 6.23 with the number of iterations. It is found that the numerical results converge to the static state after less than 150 iterations. The comparison of the numerical results obtained by the present RIQ method with those by FEM is plotted in Figure 6.24, in which the two sets of numerical results reach very good agreement.

6.4.4 Plate subject to tension

A two-dimensional plate is considered in the peridynamic model and solved by the RIQ method. The length $L = 10$ (in), width $w = 4$ (in) and thickness $h = 0.2$ (in) are given, as shown in Figure 6.25. The plate is subjected to a uniaxial force of 1.5×10^4 (lb) in x -direction. However, the axial force cannot be applied to the plate directly in peridynamic model, because of the integration term over the volume. Therefore, the tensile load is applied to the plate as the body force, calculated by $b_x = F/((x^2 - x^1) \times A)$. In this case study A is equal to $4 \times 0.2 = 0.8$ (in²), and thus b_x is equal to 1.3489×10^6 (lb/in³).

The isotropic plate has Young's modulus $E = 10^7$ (psi) and Poisson's ratio $\nu = 0.25$ respectively. If the numbers of the virtual nodes are NPX and NPY in x - and y -directions respectively, the cosine law gives the distribution of the virtual nodes as

$$x^i = \left(1 - \cos \left(\frac{i-1}{NPX-1} \pi \right) \right) \times 10 / 2 + (-5) \quad (6.37)$$

and

$$y^j = \left(1 - \cos \left(\frac{j-1}{NPY-1} \pi \right) \right) \times 4 / 2 + (-2) \quad (6.38)$$

The discretized peridynamic model of the plate is thus constructed as

$$\rho \ddot{\mathbf{u}}(x^i, y^j, t) = \sum_{n=1}^{NPX} \sum_{m=1}^{NPY} w^n w^m \mathbf{f}(\mathbf{u}^{nm}, \mathbf{u}^{ij}, \mathbf{x}^{nm}, \mathbf{x}^{ij}, t) + \mathbf{b}(\mathbf{x}^{ij}, t) \quad (6.39)$$

Eq. (6.39) is solved with $NPX = 51$ and $NPY = 11$ respectively, and $l = 0.86$ is chosen. The variations of the velocities in x - and y -directions are plotted in Figures 6.26 and 6.27, respectively. It is found that the numerical results converge to the static state after 200 iterations.

The variations of the displacements with the number of iterations in x - and y -directions are plotted in Figures 6.28 and 6.29, respectively. The comparisons of the displacements between the RIQ method and FEM in the x - and y -directions are plotted in Figure 6.30. It is seen that the numerical results of the RIQ method agree well with those of FEM in the middle of the beam, whereas slightly differ at the boundary, probably because of the cosine distribution of the virtual nodes. The density of the virtual nodes near the boundary is very high, which is good for capturing the solutions near the boundary conditions. However, the response function in peridynamics possesses a very

high gradient for the dense distribution of nodes. The numerical results of the RIQ method in the y -direction are poorer than those in the x -direction, probably because the number of nodes in y -direction is much less than that in the x -direction. In order to improve the numerical results, the number of the nodes in y -direction is increased to 21, and another pattern of nodal distribution is employed as follows,

$$x^i = \left\{ -\cos \left[\frac{(2i-1)\pi}{2N} \right] / \left[2\cos(\pi/2N) \right] + \frac{1}{2} \right\} \times L + \left(-\frac{L}{2} \right) \quad (6.40)$$

and

$$y^j = \left\{ -\cos \left[\frac{(2j-1)\pi}{2N} \right] / \left[2\cos(\pi/2N) \right] + \frac{1}{2} \right\} \times h + \left(-\frac{h}{2} \right) \quad (6.41)$$

The numerical results of the RIQ method and FEM are compared in Figure 6.31, in which the results near the boundary are much improved, compared with those in Figure 6.30.

6.4.5 Plate with a central hole subject to tension

In this case study, a plate with a hole at the center is considered. The dimensions of the plate are given as follows, the length $L = 9$ (in), the height $h = 3$ (in), the thickness $w = 0.2$ (in), and the radius of the central hole $r = 0.5$ (in), as shown in Figure 6.32. The plate is subjected to a uniaxial force of 6000 lb. The isotropic material properties of the plate include the elastic modulus $E = 10^7$ (psi) and Poisson's ratio $\nu = 0.25$ respectively.

The plate is discretized by 51 and 11 virtual nodes in x - and y -directions, following the distributions given in Eqs. (6.40) and (6.41), respectively. In order to capture the boundary of the hole, the nodes in the area $-0.5 \leq x \leq 0.5$ are replaced by $x = \pm 0.5 \cos(t)$,

and the nodes in the area $-0.5 \leq y \leq 0.5$ are replaced by $y = \pm \sin(t)$. As such, in total 51 nodes in the x -direction and 11 nodes in the y -direction are employed to discretize the plate and calculate the weighting coefficients, as shown in Figure 6.33. The field nodes are distributed in the same manner but less than the virtual nodes, which are 31 and 9 in x - and y - directions.

The problem is solved with $l = 0.35$. The variations of the velocities in x -direction at the two points (0.5, 0) and (4.5, 0) are shown in Figure 6.34. The variations of the velocities in y -direction at the two points (0, 0.5) and (0, 1.5) are shown in Figure 6.35. It is found that the velocities converge to zero at around 350 iterations. The variations of the displacements in x -direction at the two points (0.5, 0) and (4.5, 0) are plotted in Figure 6.36. The variations of the displacements in the y -direction at the two points (0, 0.5) and (0, 1.5) are plotted in Figure 6.37. The comparison of the horizontal displacements between the RIQ method and FEM are plotted in Figure 6.38. It is obvious that the numerical results by the RIQ method agree well with those by the FEM, and the displacements become slightly nonlinear due to the cutout in the center of the plate.

6.5 Summary

In this chapter, the RIQ method is employed to achieve static numerical solutions of peridynamic problems. As a new theory of continuum mechanics, peridynamics takes the integral equation as the governing equation, which can be conveniently discretized and solved by the RIQ method. In order to achieve the static numerical results, the RIQ method is combined with the dynamic relaxation technique together. The governing equation is discretized first by the RIQ method, and then transformed to an artificial

dynamic space by adding the artificial inertia and damping forces. After sufficient iterations of the dynamic relaxation scheme, the system undergoing damped vibration ultimately comes to the static equilibrium position. As a result, the final static solution for the problem is obtained. Several peridynamic problems are solved by the present RIQ method, including a beam subjected to tension, a beam subjected to bending, a rod subjected to tension, a rectangular plate subjected to tension, and a plate with a central hole subjected to tension. For the beam subjected to tension, both the one- and three-dimensional models are investigated with good results. The beam subjected to bending is used to study the problem with nonlinear transmutation. It is concluded that the RIQ method can capture the nonlinear deformation well. However, the number of iterative times is quite large due to the system being over-damped. The rod under tension and the plate with a central hole under tension reveal the peridynamic problems with non-rectangular domains. Also, both the plate under tension and the plate with a central hole under tension are studied by two-dimensional models. The numerical results are compared with those by the finite element method to prove the accuracy and efficiency of the present RIQ method. Based on the case studies mentioned above, it is thus concluded that the RIQ method can solve the peridynamic problems well for basic structures subjected to tensile and bending loads.

The computational cost for the present technique, which is a combination of the RIQ method and the dynamic relaxation technique, is relatively large than the FEM, due to the iterative procedure in the dynamic relaxation technique. The results appear slightly inferior at boundaries compared to the FEM, may be due to the cosine distribution of the virtual nodes for the GIQ technique. The density of the virtual nodes on the boundary is

very large, causing the nodal spacing between the virtual nodes near the boundary being too small. Thus the weighting matrix for the GIQ technique is near singular. In order to overcome this drawback, the RIQ method may be localized in the future. In other words, small local domains may be employed as the interpolation domain, instead of the whole domain employed in the present method.

Figure 6.1. Interaction failure due to critical stretch.

Figure 6.2. Rectangular beam subject to tension.

Figure 6.3. Variation of the velocities over the 51 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.

Figure 6.4. Variation of the displacements over the 51 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.

Figure 6.5. Comparison of the displacements between the FEM and RIQ method over the 51 virtual nodes for the example case shown in Section 6.4.1.

Figure 6.6. Variation of the velocities over the 101 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.

Figure 6.7. Variation of the displacements over 101 virtual nodes with the number of iterations for the example case shown in Section 6.4.1.

Figure 6.8. Comparison of the displacements between the FEM and RIQ method over 101 virtual nodes for the example case shown in Section 6.4.1.

Figure 6.9. Variation of the velocities in the x -direction with the number of iterations over $51 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.1.

Figure 6.10. Variation of the displacements in the x -direction with the number of iterations over $51 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.1.

Figure 6.11. Comparison of the displacements in the x -direction between the FEM and RIQ method over $51 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.1.

Figure 6.12. Contour of the deformations for the example case shown in Section 6.4.1.

Figure 6.13. Beam subjected to bending.

Figure 6.14. Variation of the velocities in the x -direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2.

Figure 6.15. Variation of the velocities in the y-direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2.

Figure 6.16. Variation of the displacements in the x-direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2.

Figure 6.17. Variation of the displacements in the y-direction with the number of iterations over $41 \times 3 \times 3$ virtual nodes for the example case shown in Section 6.4.2.

Figure 6.18. Comparison of the vertical displacements between the RIQ method and FEM for the example case shown in Section 6.4.2.

Figure 6.19. Comparison of the horizontal displacements between the RIQ method and FEM for the example case shown in Section 6.4.2.

Figure 6.20. Rod subjected to tension.

Figure 6.21. Discretization of the y - z plane for the example case shown in Section 6.4.3.

Figure 6.22. Variation of the velocities over $51 \times 7 \times 7$ virtual nodes with the number of iterations for the example case shown in Section 6.4.3.

Figure 6.23. Variation of the displacements over $51 \times 7 \times 7$ virtual nodes with the number of iterations for the example case shown in Section 6.4.3.

Figure 6.24. Comparison of the displacements between the FEM and RIQ method over $51 \times 7 \times 7$ virtual nodes for the example case shown in Section 6.4.3.

Figure 6.25. Plate subjected to tension.

Figure 6.26. Variation of the velocities in the x -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.

Figure 6.27. Variation of the velocities in the y -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.

Figure 6.28. Variation of the displacements in the x -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.

Figure 6.29. Variation of the displacements in the y -direction with the number of iterations over 51×11 virtual nodes for the example case shown in Section 6.4.4.

Figure 6.30. Comparison of the displacements between the FEM and RIQ method with 51×11 virtual nodes for the example case shown in Section 6.4.4, (a) along the horizontal central line; (b) along the vertical central line.

Figure 6.31. Comparison of the displacements between the FEM and RIQ method over 51×21 virtual nodes for the example case shown in Section 6.4.4, (a) along the horizontal central line; (b) along the vertical central line.

Figure 6.32. Rectangular plate with a central hole subjected to tension.

Figure 6.33. Nodal distribution for the plate with a central hole.

Figure 6.34. Variation of the velocities in the x -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5.

Figure 6.35. Variation of the velocities in the x -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5.

Figure 6.36. Variation of the displacements in the x -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5.

Figure 6.37. Variation of the displacements in the y -direction over 51×13 virtual nodes with the number of iterations for the example case shown in Section 6.4.5.

Figure 6.38. Comparison of the horizontal displacements between the RIQ method and FEM for the example case shown in Section 6.4.5.

Chapter 7 Conclusions and Future Works

7.1 Conclusions

In this thesis, the novel meshless method, termed the random integral quadrature (RIQ) method, is developed for solving the integral governing equations for engineering applications. As a strong-form meshless technique, the RIQ method combines the generalized integral quadrature (GIQ) technique with the Kriging interpolation technique. Although the GIQ technique is a remarkable quadrature method for integral equations, it always requires regular domains formed by straight lines and the distributed nodes lying on straight lines inside the domain. As such, the Kriging interpolation technique is required to interpolate the field variables generated by the GIQ method, via the randomly distributed field nodes scattered in irregular domains. The two sets of nodes distributed in the integral domain are thus required by the RIQ method, namely virtual and field nodes. The virtual nodes are distributed along straight lines, and the field nodes can be distributed randomly or uniformly. The integral equation is discretized first over the virtual nodes by the GIQ strategy, and then the function value at each virtual node is approximated by the Kriging interpolation technique over all of the field nodes. As a result, a system of algebraic equations is established, and the numerical results at the field nodes are finally obtained by solving this algebraic system.

The RIQ method is validated by solving the Fredholm integral equations of the second kind and the Volterra integral equations of the second kind. For the Fredholm

integral equations of the second kind, the RIQ method is performed in one-, two- and three-dimensional case studies, with the field nodes distributed randomly or uniformly. The results are quite accurate even with a few field nodes. For Volterra integral equations of the second kind, the RIQ method is employed for one- and two-dimensional case studies, with both randomly and uniformly distributed field nodes. The results are compared with the analytical solutions, and good agreement is achieved.

The convergence characteristics of the RIQ method are studied through Fredholm integral equations of the second kind and Volterra integral equations of the second kind in multi-dimensional domains. In the numerical experiments, the number of virtual nodes is often fixed, while the number of the field nodes varies in a certain range. It is suggested that the number of the virtual nodes is generally larger than that of the field nodes. It is also found that the numerical results converge to the analytical solution well as the number of field nodes increases. The convergence rates are calculated, and they are quite large for most of the case studies, meaning the numerical results converge to the analytical solutions very fast.

In addition, the RIQ method is implemented to solve other types of integral equations, including integral equations with irregular domains and nonlinear integral equations respectively. For the integral governing equations with irregular domains, the distributions of the virtual and field nodes are arranged by a developed strategy. A rectangular domain is created firstly, which covers the irregular domain, and the virtual nodes are distributed within the rectangular domain to obtain the weighting coefficients of the generalized integral quadrature (GIQ) technique. However, only the virtual nodes inside the irregular domain are selected to discretize the integral equation. In order to

capture the boundary of the irregular domain, the field nodes are distributed along the boundary and inside the domain. Through this strategy, the RIQ method is successfully used to solve the integral equations over the irregular domains, where good results are obtained. For nonlinear integral problems, the RIQ method is combined with a quasilinearization technique to solve this kind of problems. The nonlinear integral equation is linearized first by the quasilinearization technique. An iterative scheme is then formulated in the form of the Volterra integral equation of the second kind, for which the RIQ method is easily implemented. Several examples and the Blasius boundary-layer problem are solved numerically.

The RIQ method is also applied to solve a kind of the elastic engineering problems, called peridynamics. Peridynamic theory is a reformulation of the continuum theory. As a kind of nonlocal problem, the governing equations are in the form of integro-differential equations. The RIQ method is combined with a dynamic relaxation technique to obtain the static solution of the solid mechanics problems, such as the beam subjected to tension, the beam subjected to bending, the plate subjected to tension, and so on. The numerical results are compared with those by FEM, and they agree very well each other.

7.2 Future works

In this section, several possible interesting studies are recommended below for further research work.

- (1) The RIQ method is explored to solve the peridynamic problems associated with the fracture and discontinuity. Since the governing equation of peridynamics is of integro-differential form, there is no derivative of the displacement component. As

such, it is valid everywhere whether or not displacement discontinuities occur. The RIQ method may be a good approach to solve this kind of problems, with simple implementation and high accuracy.

- (2) The singular integral equation has enormous engineering applications in the fluid mechanics, bio-mechanics and electromagnetic theory. It is a challenge for numerical solutions, because the kernel matrix becomes singular when the problems are discretized. The RIQ method is recommended to solve singular integral equations, where the main challenge may be how to treat the singularity of the kernel matrix appropriately.
- (3) In the present work, only the Fredholm integral equations and the Volterra integral equations both of the second kind are solved for examination of the RIQ method. Both the Fredholm integral equations and the Volterra integral equations of the first kind are not considered here. It is recalled that these two types of equations are ill-conditioned problems [16] and not easily solved by the conventional numerical techniques. The RIQ method should be tried for these problems.
- (4) As an efficient meshless method, the RIQ method is possibly combined with other numerical methods, such as the finite element method (FEM), to solve multi-scale problems. For different parts of the system, the governing equations may be discretized with the RIQ method and FEM respectively.
- (5) In a meshless method, the function approximation affects the computational accuracy of the solution directly. The shape function in the present RIQ method is constructed by employing all the field nodes in the whole domain. However, the localized variation of function values may be more efficient and accurate than that

of the whole domain for some cases, such as high-gradient problems. Hence, the local domains may be created to approximate function values at the virtual nodes. Only the field nodes inside the local domain are employed to construct the shape functions. In other words, the RIQ method may be localized for improvement of computational efficiency and accuracy.

References

- [1] L. B. Lucy, A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 1977. **82**(12): p. 1013-1024.
- [2] B. Nayroles, G. Touzot, and P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, 1992. **10**: p. 307-318.
- [3] T. Belytschko, Y. Y. Lu, and L. Gu, Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 1994. **37**(2): p. 229-256.
- [4] W. K. Liu, J. Sukky, and Z. Yi Fei, Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 1995. **20**(8-9): p. 1081-1106.
- [5] G. R. Liu and Y. T. Gu, A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering*, 2001. **50**(4): p. 937-951.
- [6] T. J. Liszka, C. A. M. Duarte, and W. W. Tworzydło, hp-Meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, 1996. **139**(1-4): p. 263-288.
- [7] S. N. Atluri and T. Zhu, A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 1998. **22**(2): p. 117-127.
- [8] M. Rahman, *Integral equations and their applications*. 2007, Southampton: WIT Press.
- [9] M. Y. Rahimi-Ardabili and S. Shahmorad, Iterative numerical solution of non-linear integro-differential equations by the Tau method. *Applied Mathematics and Computation*, 2007. **193**(2): p. 514-522.
- [10] K. Maleknejad and A. Arzhang, Numerical solution of the Fredholm singular integro-differential equation with Cauchy kernel by using Taylor-series expansion and Galerkin method. *Applied Mathematics and Computation*, 2006. **182**(1): p. 888-897.

- [11] M. C. De Bonis and C. Laurita, Nystrom method for Cauchy singular integral equations with negative index. *Journal of Computational and Applied Mathematics*, 2009. **232**(2): p. 523-538.
- [12] Y. Pelekh, Numerical methods of solving nonlinear integral equations of volterra type. *Journal of Mathematical Sciences*, 1998. **90**(5): p. 2431-2435.
- [13] S. Hatamzadeh-Varmazyar and Z. Masouri, Numerical method for analysis of one- and two-dimensional electromagnetic scattering based on using linear Fredholm integral equation models. *Mathematical and Computer Modelling*, 2011. **54**(9-10): p. 2199-2210.
- [14] S. A. Silling, M. Zimmermann, and R. Abeyaratne, Deformation of a Peridynamic Bar. *Journal of Elasticity*, 2003. **73**(1): p. 173-190.
- [15] P. K. Kythe and P. Puri, *Computational Methods for Linear Integral Equations*. 2002, Boston: Birkhauser.
- [16] K. Maleknejad, R. Mollapourasl, and K. Nouri, Convergence of numerical solution of the Fredholm integral equation of the first kind with degenerate kernel. *Applied Mathematics and Computation*, 2006. **181**(2): p. 1000-1007.
- [17] C. Shu, et al., Numerical studies of unsteady boundary layer flows past an impulsively started circular cylinder by GDQ and GIQ approaches. *Computer Methods in Applied Mechanics and Engineering*, 1996. **135**(3-4): p. 229-241.
- [18] E. Oñate, et al., A finite point method in computational mechanics and its applications to overcom transport and fluid flow. *International Journal for Numerical Methods in Engineering*, 1996. **39**(22): p. 3839-3866.
- [19] H. Li, et al., Hermite–Cloud: a novel true meshless method. *Computational Mechanics*, 2003. **33**(1): p. 30-41.
- [20] G. R. Liu, et al., A gradient smoothing method (GSM) with directional correction for solid mechanics problems. *Computational Mechanics*, 2008. **41**(3): p. 457-472.
- [21] R. Bellman, B.G. Kashef, and J. Casti, Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations. *Journal of Computational Physics*, 1972. **10**: p. 40-52.
- [22] J. R. Quan and C. T. Chang, New insights in solving distributed system equations by the quadrature method-II. *Numerical experiments. Computers and Chemical Engineering*, 1989. **13**(9): p. 1017-1024.

- [23] C. Shu, B. C. Khoo, and K. S. Yeo, Numerical solutions of incompressible Navier-Stokes equations by generalized differential quadrature. *Finite Elements in Analysis and Design*, 1994. **18**(1-3): p. 83-97.
- [24] S. S. Mulay, H. Li, and 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*, 2010: p. 1-27.
- [25] G. R. Liu and Y. T. Gu, A local radial point interpolation method (LRPIM) for free vibration analyses of 2-D solids. *Journal of Sound and Vibration*, 2001. **246**(1): p. 29-46.
- [26] G. R. Liu, *Mesh free methods: moving beyond the finite element method*. 2003, Boca Raton: CRC press.
- [27] R. A. Gingold and J. J. Monaghan, smooth particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 1977. **181**(2): p. 375-389.
- [28] T. Belytschko, Y. Y. Lu, and L. Gu, Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 1994. **37**(2): p. 229-256.
- [29] T. Belytschko, L. Gu, and Y. Y. Lu, Fracture and crack growth by element free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering*, 1994. **2**(3A): p. 519.
- [30] J. Braun and M. Sambridge, A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature*, 1995. **376**: p. 655-660.
- [31] J. M. Melenk and I. Babuska, The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 1996. **139**(1-4): p. 289-314.
- [32] H. Li, Q. X. Wang, and K. Y. Lam, Development of a novel meshless Local Kriging (LoKriging) method for structural dynamic analysis. *Computer Methods in Applied Mechanics and Engineering*, 2004. **193**(23-26): p. 2599-2619.
- [33] T. Zhu, J. D. Zhang, and S. N. Atluri, A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. *Computational Mechanics*, 1998. **21**(3): p. 223-235.

- [34] T. Zhu, J. Zhang, and S. N. Atluri, A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. *Computational Mechanics*, 1998. **22**(2): p. 174-186.
- [35] Y. X. Mukherjee and S. Mukherjee, On boundary conditions in the element-free Galerkin method. *Computational Mechanics*, 1997. **19**(4): p. 264-270.
- [36] K. C. Mandal, M. Subrata, and X. M. Yu, The boundary node method for three-dimensional linear elasticity. *International Journal for Numerical Methods in Engineering*, 1999. **46**(8): p. 1163-1184.
- [37] K. C. Mandal and M. Subrata, The boundary node method for three-dimensional problems in potential theory. *International Journal for Numerical Methods in Engineering*, 2000. **47**(9): p. 1523-1547.
- [38] Y. T. Gu and G. R. Liu, A boundary point interpolation method for stress analysis of solids. *Computational Mechanics*, 2002. **28**(1): p. 47-54.
- [39] Y. T. Gu and G. R. Liu, A boundary radial point interpolation method (BRPIM) for 2-D structural analysis *Structural Engineering and mechanics*, 2003. **15**(5): p. 530-550.
- [40] R. C. Allen and G. M. Wing, A method for accelerating the iterative solution of a class of Fredholm integral equations. *Journal of Optimization Theory and Applications*, 1978. **24**(1): p. 5-27.
- [41] M. Javidi and A. Golbabai, A numerical solution for solving system of Fredholm integral equations by using homotopy perturbation method. *Applied Mathematics and Computation*, 2007. **189**(2): p. 1921-1928.
- [42] Y. Liu, Application of the Chebyshev polynomial in solving Fredholm integral equations. *Mathematical and Computer Modelling*, 2009. **50**(3-4): p. 465-469.
- [43] M. A. Golberg, Initial value methods in the theory of Fredholm integral equations. *Journal of Optimization Theory and Applications*, 1972. **9**(2): p. 112-119.
- [44] M. Golberg, Initial-value methods in the theory of Fredholm integral equations: II. *Journal of Optimization Theory and Applications*, 1972. **9**(6): p. 426-432.
- [45] M. A. Golberg, Initial value methods in the theory of Fredholm integral equations III. The nonlinear eigenvalue problem. *Applied Mathematics and Computation*, 1975. **1**(4): p. 281-286.

- [46] K. Maleknejad and S. Sohrabi, Numerical solution of Fredholm integral equations of the first kind by using Legendre wavelets. *Applied Mathematics and Computation*, 2007. **186**(1): p. 836-843.
- [47] S. Yousefi and A. Banifatemi, Numerical solution of Fredholm integral equations by using CAS wavelets. *Applied Mathematics and Computation*, 2006. **183**(1): p. 458-463.
- [48] R. J. Hanson and J. L. Phillips, An adaptive numerical method for solving linear Fredholm integral equations of the first kind. *Numerische Mathematik*, 1975. **24**(4): p. 291-307.
- [49] G. Hsiao, P. Kopp, and W. Wendland, A Galerkin collocation method for some integral equations of the first kind. *Computing*, 1980. **25**(2): p. 89-130.
- [50] K. Maleknejad and Y. Mahmoudi, Numerical solution of linear Fredholm integral equation by using hybrid Taylor and Block-Pulse functions. *Applied Mathematics and Computation*, 2004. **149**(3): p. 799-806.
- [51] S. Abbasbandy, E. Babolian, and M. Alavi, Numerical method for solving linear Fredholm fuzzy integral equations of the second kind. *Chaos, Solitons & Fractals*, 2007. **31**(1): p. 138-146.
- [52] E. Babolian, J. Biazar, and A. R. Vahidi, The decomposition method applied to systems of Fredholm integral equations of the second kind. *Applied Mathematics and Computation*, 2004. **148**(2): p. 443-452.
- [53] E. Babolian, S. Abbasbandy, and F. Fattahzadeh, A numerical method for solving a class of functional and two dimensional integral equations. *Applied Mathematics and Computation*, 2008. **198**(1): p. 35-43.
- [54] R. Farnoosh and M. Ebrahimi, Monte Carlo method for solving Fredholm integral equations of the second kind. *Applied Mathematics and Computation*, 2008. **195**(1): p. 309-315.
- [55] A. Golbabai and B. Keramati, Easy computational approach to solution of system of linear Fredholm integral equations. *Chaos, Solitons & Fractals*, 2008. **38**(2): p. 568-574.
- [56] V. F. Kravchenko and M. A. Basarab, Approximations by Atomic Functions and Numerical Methods for Fredholm Integral Equations of the Second Kind. *Differential Equations*, 2001. **37**(10): p. 1480-1490.

- [57] G. Long and G. Nelakanti, Superconvergence of functional approximation methods for integral equations. *Applied Mathematics Letters*, 2009. **22**(3): p. 401-405.
- [58] K. Maleknejad and M. Karami, Using the WPG method for solving integral equations of the second kind. *Applied Mathematics and Computation*, 2005. **166**(1): p. 123-130.
- [59] G. Miel, Parallel solution of Fredholm integral equations of the second kind by orthogonal polynomial expansions. *Applied Numerical Mathematics*, 1989. **5**(4): p. 345-361.
- [60] J. Rashidinia and M. Zarebnia, Numerical solution of linear integral equations by using Sinc-collocation method. *Applied Mathematics and Computation*, 2005. **168**(2): p. 806-822.
- [61] J. Saberi-Nadjafi and M. Heidari, Solving linear integral equations of the second kind with repeated modified trapezoid quadrature method. *Applied Mathematics and Computation*, 2007. **189**(1): p. 980-985.
- [62] G. Vainikko, A. Kivinukk, and J. Lippus, Fast solvers of integral equations of the second kind: wavelet methods. *Journal of Complexity*, 2005. **21**(2): p. 243-273.
- [63] S. A. Belbas, On series solutions of Volterra equations. *Applied Mathematics and Computation*, 2006. **181**(2): p. 1287-1304.
- [64] H. Brunner, A. Makroglou, and R. K. Miller, Mixed interpolation collocation methods for first and second order Volterra integro-differential equations with periodic solution. *Applied Numerical Mathematics*, 1997. **23**(4): p. 381-402.
- [65] D. Conte and I. D. Prete, Fast collocation methods for Volterra integral equations of convolution type. *Journal of Computational and Applied Mathematics*, 2006. **196**(2): p. 652-663.
- [66] D. Conte and B. Paternoster, Multistep collocation methods for Volterra Integral Equations. *Applied Numerical Mathematics*, 2009. **59**(8): p. 1721-1736.
- [67] K. Maleknejad, T. Lotfi, and K. Mahdiani, Numerical solution of first kind Fredholm integral equations with wavelets-Galerkin method (WGM) and wavelets precondition. *Applied Mathematics and Computation*, 2007. **186**(1): p. 794-800.
- [68] S. Mckee, T. Tang, and T. Diogo, An Euler-type method for two-dimensional Volterra integral equations of the first kind *IMA Journal of Numerical Analysis*, 2000. **20**: p. 423-440.

- [69] E. Messina, E. Russo, and A. Vecchio, A stable numerical method for Volterra integral equations with discontinuous kernel. *Journal of Mathematical Analysis and Applications*, 2008. **337**(2): p. 1383-1393.
- [70] P. Oja and D. Saveljeva, Cubic Spline Collocation for Volterra Integral Equations. *Computing*, 2002. **69**(4): p. 319-337.
- [71] L. Gori, A review of some numerical methods for the solution of Cauchy singular integral equations. *Milan Journal of Mathematics*, 1995. **65**(1): p. 373-391.
- [72] P. Rabinowitz, Application of approximating splines for the solution of Cauchy singular integral equations. *Applied Numerical Mathematics*, 1994. **15**(2): p. 285-297.
- [73] S. Prosdorf and J. Elschner, Finite element methods for singular integral equations on an interval. *Engineering Analysis*, 1984. **1**(2): p. 83-87.
- [74] S. Prosdorf and A. Rathsfeld, On spline Galerkin methods for singular integral equations with piecewise continuous coefficients. *Numerische Mathematik*, 1986. **48**(1): p. 99-118.
- [75] P. Junghanns and U. Luther, Cauchy singular integral equations in spaces of continuous functions and methods for their numerical solution. *Journal of Computational and Applied Mathematics*, 1997. **77**(1-2): p. 201-237.
- [76] P. Junghanns and K. Muller, A collocation method for nonlinear Cauchy singular integral equations. *Journal of Computational and Applied Mathematics*, 2000. **115**(1-2): p. 283-300.
- [77] P. Junghanns and B. Silbermann, Numerical analysis for one-dimensional Cauchy singular integral equations. *Journal of Computational and Applied Mathematics*, 2000. **125**(1-2): p. 395-421.
- [78] M. C. De Bonis and G. Mastroianni, Projection Methods and Condition Numbers in Uniform Norm for Fredholm and Cauchy Singular Integral Equations. *SIAM Journal on Numerical Analysis*, 2006. **44**(4): p. 1351-1374.
- [79] A. Pedas and G. Vainikko, Smoothing Transformation and Piecewise Polynomial Collocation for Weakly Singular Volterra Integral Equations. *Computing*, 2004. **73**(3): p. 271-293.
- [80] F. de Hoog and R. Weiss, High Order Methods for a Class of Volterra Integral Equations with Weakly Singular Kernels. *SIAM Journal on Numerical Analysis*, 1974. **11**(6): p. 1166-1180.

- [81] H. Brunner and S. P. Nørsett, Superconvergence of collocation methods for Volterra and Abel integral equations of the second kind. *Numerische Mathematik*, 1981. **36**(4): p. 347-358.
- [82] H. Brunner, Nonpolynomial Spline Collocation for Volterra Equations with Weakly Singular Kernels. *SIAM Journal on Numerical Analysis*, 1983. **20**(6): p. 1106-1119.
- [83] E. Hairer, C. Lubich, and M. Schlichte, Fast numerical solution of weakly singular Volterra integral equations. *Journal of Computational and Applied Mathematics*, 1988. **23**(1): p. 87-98.
- [84] J. I. Frankel, A nonlinear heat transfer problem: solution of nonlinear, weakly singular Volterra integral equations of the second kind. *Engineering Analysis with Boundary Elements*, 1991. **8**(5): p. 231-238.
- [85] T. Tang, Superconvergence of numerical solutions to weakly singular Volterra integro-differential equations. *Numerische Mathematik*, 1992. **61**(1): p. 373-382.
- [86] B. Jumarhon and S. McKee, Product integration methods for solving a system of nonlinear Volterra integral equations. *Journal of Computational and Applied Mathematics*, 1996. **69**(2): p. 285-301.
- [87] A. P. Orsi, Product integration for volterra integral equations of the second kind with weakly singular kernels. *Mathematics of computation*, 1996. **65**: p. 1201-1212.
- [88] P. Lima and T. Diogo, An extrapolation method for a Volterra integral equation with weakly singular kernel. *Applied Numerical Mathematics*, 1997. **24**(2-3): p. 131-148.
- [89] A. M. Wazwaz and S. A. Khuri, A reliable technique for solving the weakly singular second-kind Volterra-type integral equations. *Applied Mathematics and Computation*, 1996. **80**(2-3): p. 287-299.
- [90] P. Baratella, A Nyström interpolant for some weakly singular linear Volterra integral equations. *Journal of Computational and Applied Mathematics*, 2009. **231**(2): p. 725-734.
- [91] G. Capobianco, D. Conte, and I. Del Prete, High performance parallel numerical methods for Volterra equations with weakly singular kernels. *Journal of Computational and Applied Mathematics*, 2009. **228**(2): p. 571-579.

- [92] T. Diogo, Collocation and iterated collocation methods for a class of weakly singular Volterra integral equations. *Journal of Computational and Applied Mathematics*, 2009. **229**(2): p. 363-372.
- [93] E. Babolian and A. S. Shamloo, Numerical solution of Volterra integral and integro-differential equations of convolution type by using operational matrices of piecewise constant orthogonal functions. *Journal of Computational and Applied Mathematics*, 2008. **214**(2): p. 495-508.
- [94] L. Tao and H. Yong, Extrapolation method for solving weakly singular nonlinear Volterra integral equations of the second kind. *Journal of Mathematical Analysis and Applications*, 2006. **324**(1): p. 225-237.
- [95] Y. Cao, T. Herdman, and Y. Xu, A Hybrid Collocation Method for Volterra Integral Equations with Weakly Singular Kernels. *SIAM Journal on Numerical Analysis*, 2003. **41**(1): p. 364-381.
- [96] P. Lima and T. Diogo, Numerical solution of a nonuniquely solvable Volterra integral equation using extrapolation methods. *Journal of Computational and Applied Mathematics*, 2002. **140**(1-2): p. 537-557.
- [97] E. A. Galperin, et al., Variable transformations in the numerical solution of second kind Volterra integral equations with continuous and weakly singular kernels; extensions to Fredholm integral equations. *Journal of Computational and Applied Mathematics*, 2000. **115**(1-2): p. 193-211.
- [98] J. P. Kauthen, A survey of singularly perturbed Volterra equations. *Applied Numerical Mathematics*, 1997. **24**(2-3): p. 95-114.
- [99] D. Bahuguna, A. Ujlayan, and D. N. Pandey, A comparative study of numerical methods for solving an integro-differential equation. *Computers & Mathematics with Applications*, 2009. **57**(9): p. 1485-1493.
- [100] A. Avudainayagam and C. Vani, Wavelet-Galerkin method for integro-differential equations. *Applied Numerical Mathematics*, 2000. **32**(3): p. 247-254.
- [101] J.-H. He, A coupling method of a homotopy technique and a perturbation technique for non-linear problems. *International Journal of Non-Linear Mechanics*, 2000. **35**(1): p. 37-43.
- [102] J. Biazar, H. Ghazvini, and M. Eslami, He's homotopy perturbation method for systems of integro-differential equations. *Chaos, Solitons & Fractals*, 2009. **39**(3): p. 1253-1258.

- [103] M. R. Capobianco, G. Criscuolo, and P. Junghanns, A fast algorithm for Prandtl's integro-differential equation. *Journal of Computational and Applied Mathematics*, 1997. **77**(1-2): p. 103-128.
- [104] Z. He and X. He, Monotone iterative technique for impulsive integro-differential equations with periodic boundary conditions. *Computers & Mathematics with Applications*, 2004. **48**(1-2): p. 73-84.
- [105] A. Alipanah and M. Dehghan, Numerical solution of the nonlinear Fredholm integral equations by positive definite functions. *Applied Mathematics and Computation*, 2007. **190**(2): p. 1754-1761.
- [106] J. Biazar and H. Ghazvini, Numerical solution for special non-linear Fredholm integral equation by HPM. *Applied Mathematics and Computation*, 2008. **195**(2): p. 681-687.
- [107] A. Ghorbani and J. Saberi-Nadjafi, Exact solutions for nonlinear integral equations by a modified homotopy perturbation method. *Computers & Mathematics with Applications*, 2008. **56**(4): p. 1032-1039.
- [108] M. A. Abdou, M. M. El-Borai, and M. M. El-Kojok, Toeplitz matrix method and nonlinear integral equation of Hammerstein type. *Journal of Computational and Applied Mathematics*, 2009. **223**(2): p. 765-776.
- [109] E. Babolian, F. Fattahzadeh, and E. G. Raboky, A Chebyshev approximation for solving nonlinear integral equations of Hammerstein type. *Applied Mathematics and Computation*, 2007. **189**(1): p. 641-646.
- [110] M. A. Hernandez, M. J. Rubio, and J. A. Ezquerro, Secant-like methods for solving nonlinear integral equations of the Hammerstein type. *Journal of Computational and Applied Mathematics*, 2000. **115**(1-2): p. 245-254.
- [111] K. Maleknejad and H. Derili, The collocation method for Hammerstein equations by Daubechies wavelets. *Applied Mathematics and Computation*, 2006. **172**(2): p. 846-864.
- [112] K. Maleknejad and H. Derili, Numerical solution of Hammerstein integral equations by using combination of spline-collocation method and Lagrange interpolation. *Applied Mathematics and Computation*, 2007. **190**(2): p. 1557-1562.
- [113] R. Katani and S. Shahmorad, Block by block method for the systems of nonlinear Volterra integral equations. *Applied Mathematical Modelling*, 2010. **34**(2): p. 400-406

- [114] S. H. Javadi, A. Davari, and E. Babolian, Numerical implementation of the Adomian decomposition method for nonlinear Volterra integral equations of the second kind. *International Journal of Computer Mathematics*, 2007. **84**(1): p. 75 - 79.
- [115] G. Capobianco, et al., Fast Runge–Kutta methods for nonlinear convolution systems of Volterra integral equations. *BIT Numerical Mathematics*, 2007. **47**(2): p. 259-275.
- [116] I. L. El-Kalla, Convergence of the Adomian method applied to a class of nonlinear integral equations. *Applied Mathematics Letters*, 2008. **21**(4): p. 372-376.
- [117] C. Shu, *Differential Quadrature and Its Application in Engineering*. 2000, London: Springer. 340.
- [118] W. Gautschi, *Numerical Analysis: An Introduction*. 1997: Springer.
- [119] Y. L. Wu, et al., Implementation of multi-grid approach in domain-free discretization method to speed up convergence. *Computer Methods in Applied Mechanics and Engineering*, 2003. **192**(20-21): p. 2425-2438.
- [120] R. Bellman and K. RE, *Quasilinearization and nonlinear boundary value problems*. 1965, New York: American Ensevier.
- [121] V. Lakshmikantham. and S. Malek., *Generalized quasilinearization*. *Nonlinear world*, 1994. **1**: p. 59-63.
- [122] V. Lakshmikantham, An extension of the method of quasilinearization. *Journal of Optimization Theory and Applications*, 1994. **82**(2): p. 315-321.
- [123] S. G. Pandit, Quadratically converging iterative schemes for nonlinear Volterra integral equations and an application. *Journal of Applied Mathematics and Stochastic Analysis*, 1997. **10**(2): p. 169-178.
- [124] K. Maleknejad and E. Najafi, Numerical solution of nonlinear volterra integral equations using the idea of quasilinearization. *Communications in Nonlinear Science and Numerical Simulation*, 2011. **16**(1): p. 93-100.
- [125] A. D. Young, *Boundary layers*. BSP Professional Books. 1989, London: British library.
- [126] H. Weyl, On the Differential Equations of the Simplest Boundary-Layer Problems. *The Annals of Mathematics*, 1942. **43**(2): p. 381-407.

- [127] S. A. Silling and E. Askari, A meshfree method based on the peridynamic model of solid mechanics. *Computers & Structures*, 2005. **83**(17-18): p. 1526-1535.
- [128] S. A. Silling and F. Bobaru, Peridynamic modeling of membranes and fibers. *International Journal of Non-Linear Mechanics*. **40**(2-3): p. 395-409.
- [129] J. T. Foster and et al., Viscoplasticity using peridynamics. *International Journal for Numerical Methods in Engineering*, 2009: p. n/a.
- [130] S.A. Silling and R. B. Lehoucq, Convergence of peridynamics to classical elasticity. *Journal of Elasticity*, 2008. **93**: p. 13-37.
- [131] S. Silling, et al., Peridynamic States and Constitutive Modeling. *Journal of Elasticity*, 2007. **88**(2): p. 151-184.
- [132] S. Silling, Linearized Theory of Peridynamic States. *Journal of Elasticity*, 2010. **99**(1): p. 85-111.
- [133] T. L. Warren, et al., A non-ordinary state-based peridynamic method to model solid material deformation and fracture. *International Journal of Solids and Structures*, 2009. **46**(5): p. 1186-1195.
- [134] R. B. Lehoucq and S. A. Silling, Force flux and the peridynamic stress tensor. *Journal of the Mechanics and Physics of Solids*, 2008. **56**(4): p. 1566-1577.
- [135] Y. D. Ha and F. Bobaru, Studies of dynamic crack propagation and crack branching with peridynamics. *International Journal of Fracture*, 2010. **162**(1-2): p. 229-244.
- [136] F. Bobaru, et al., Convergence, adaptive refinement, and scaling in 1D peridynamics. *International Journal for Numerical Methods in Engineering*, 2009. **77**(6): p. 852-877.
- [137] P. N. Demmie and S. A. Silling, An approach to modeling extreme loading of structures using peridynamics. *Journal of mechanics of materials and structures*, 2007. **2**(10): p. 1921-1945.
- [138] B. Kilic and E. Madenci, Structural stability and failure analysis using peridynamic theory. *International Journal of Non-Linear Mechanics*, 2009. **44**(8): p. 845-854.
- [139] M. L. Parks, et al., Implementing peridynamics within a molecular dynamics code. *Computer Physics Communications*, 2008. **179**(11): p. 777-783.

- [140] R. W. Macek and S. A. Silling, Peridynamics via finite element analysis. *Finite Elements in Analysis and Design*, 2007. **43**(15): p. 1169-1178.
- [141] H. G. Aksoy and E. Senocak, Discontinuous Galerkin method based on peridynamic theory, in *9th World Congress on Computational Mechanics and 4th Asian Pacific Congress on Computational Mechanics*.
- [142] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 2000. **48**(1): p. 175-209.
- [143] M. R. Pajand and M. T. Hakkak, Nonlinear Analysis of Truss Structures Using Dynamic Relaxation. *International Journal of Engineering*, 2006. **19**: p. 11-22.
- [144] G. M. Campbell and J. T. Day, The numerical solution of nonlinear Volterra integral equations. *BIT Numerical Mathematics*, 1970. **10**(1): p. 10-19.
- [145] P. Underwood, Dynamic Relaxation. *computational Methods for Transient Analysis*, 1983. **1**: p. 245-265.

Publications arising from this thesis

Journal papers

- [1] **Hua Zou**, Hua Li, A novel meshless method for solving the second kind of Fredholm integral equations. *Computer Modeling in Engineering and Sciences* 67 (2010) 55-77.
- [2] Hua Li, **Hua Zou**, A random integral quadrature method for numerical analysis of the second kind of Volterra integral equations. *Journal of Computational and Applied Mathematics* 237 (2013) 35-42.
- [3] **Hua Zou**, Hua Li, A random integral quadrature method for numerical computation of the Fredholm integral equations with irregular domains. *Applied Mathematics and Computation* (submitted).
- [4] Hua Li, **Hua Zou**, A numerical approach to nonlinear Volterra integral equation by combination of the RIQ method with quasilinearization for solving boundary-layer problems. *International Journal for Numerical Methods in Engineering* (submitted).
- [5] **Hua Zou**, Hua Li, A new meshless method for the peridynamic theory. *Computer Methods in Applied Mechanics and Engineering* (submitted).

Conference papers

- [1] **Hua Zou**, Hua Li, A novel meshless method for solving integral equations. *International Conference on Applied Mathematics 2012 (ICAM2012)*, Hong Kong, 28 May – 1 Jun 2012.
- [2] **Hua Zou**, Hua Li, A Transient simulation for analysis of volume transition of pH-sensitive hydrogel in micro-channel. *International Conference on Materials for Advanced Technologies 2009 (ICMAT 2009)*, Singapore, 28 June – 3 July 2009.