



**The radial integration boundary integral and
integro-differential equation methods for
numerical solution of problems with variable
coefficients**

A thesis submitted for the degree of
Doctor of Philosophy

by

Majeed Ahmed Weli AL-Jawary

B.Sc., M.Sc.

Department of Mechanical Engineering
School of Engineering and Design
Brunel University, London, UK

May 2012

Abstract

The boundary element method (BEM) has become a powerful method for the numerical solution of boundary-value problems (BVPs), due to its ability (at least for problems with constant coefficients) of reducing a BVP for a linear partial differential equation (PDE) defined in a domain to an integral equation defined on the boundary, leading to a simplified discretisation process with boundary elements only. On the other hand, the coefficients in the mathematical model of a physical problem typically correspond to the material parameters of the problem. In many physical problems, the governing equation is likely to involve variable coefficients. The application of the BEM to these equations is hampered by the difficulty of finding a fundamental solution.

The first part of this thesis will focus on the derivation of the boundary integral equation (BIE) for the Laplace equation, and numerical results are presented for some examples using constant elements. Then, the formulations of the boundary-domain integral or integro-differential equation (BDIE or BDIDE) for heat conduction problems with variable coefficients are presented using a parametrix (Levi function), which is usually available.

The second part of this thesis deals with the extension of the BDIE and BDIDE formulations to the treatment of the two-dimensional Helmholtz equation with variable coefficients. Four possible cases are investigated, first of all when both material parameters and wave number are constant, in which case the zero-order Bessel function of the second kind is used as fundamental solution. Moreover, when the material parameters are variable (with constant or variable wave number), a parametrix is adopted to reduce the Helmholtz equation to a BDIE or a BDIDE. Finally, when material parameters are constant (with variable wave number), the standard fundamental solution for the Laplace equation is used in the formulation.

In the third part, the radial integration method (RIM) is introduced and discussed in detail. Modifications are introduced to the RIM, particularly the fact that the radial integral is calculated by using a pure boundary-only integral which relaxes the “star-shaped” requirement of the RIM. Then, the RIM is used to convert the domain integrals appearing in both BDIE and BDIDE for heat conduction and Helmholtz equations to equivalent boundary integrals. For domain integrals consisting of known functions the transformation is straightforward, while for domain integrals that include unknown variables the transformation is accomplished with the use of augmented radial basis functions (RBFs).

The most attractive feature of the method is that the transformations are very simple and have similar forms for both 2D and 3D problems.

Finally, the application of the RIM is discussed for the diffusion equation, in which the parabolic PDE is initially reformulated as a BDIE or a BDIDE and the RIM is used to convert the resulting domain integrals to equivalent boundary integrals. Three cases have been investigated, for homogenous, non-homogeneous and variable coefficient diffusion problems.

Declaration of own work

I hereby declare that the work presented in this thesis is my own research and has not been presented for a higher degree at any other university or institute. Any material that could be construed as the work of others is fully cited and appears in the references.

Majeed Ahmed Weli AL-Jawary

To my parents

Acknowledgements

First and foremost, I wish to thank my supervisor Professor Luiz Wrobel for all the time he devoted to give advice on numerous problems. Without him, this thesis would have never been possible. Furthermore, I am grateful for his patience, motivation, enthusiasm, and immense knowledge. I would like to express my sincere gratitude for his useful advice and encouragement, it has been a privilege to work with him.

Special thanks and deepest gratitude also to Dr. Matthias Maischak for his encouragement and personal guidance that provided a good basis for the present thesis, specially during my first year of the PhD study.

I would like to thank the staff at Brunel University who have made my time at the university enjoyable and stimulating.

Last but not least, I would like to thank my family for the great support during my PhD study. I want to thank my wife for the encouragement and support she has provided over the years.

Author's Publications

Journal Papers

1. M.A. AL-Jawary and L.C. Wrobel. Numerical solution of two-dimensional mixed problems with variable coefficients by the boundary-domain integral and integro-differential equation methods, *Engineering Analysis with Boundary Elements*, 35:1279-1287; 2011.
2. M.A. AL-Jawary and L.C. Wrobel. Radial integration boundary integral and integro-differential equation methods for two-dimensional heat conduction problems with variable coefficients, *Engineering Analysis with Boundary Elements*, 36:685-695; 2012.
3. M.A. AL-Jawary and L.C. Wrobel. Numerical solution of the two-dimensional Helmholtz equation with variable coefficients by the radial integration boundary integral and integro-differential equation methods. *International Journal of Computer Mathematics*, DOI:10.1080/00207160.2012.667087; 2012.
4. M.A. AL-Jawary, J. Ravnik, L.C. Wrobel and L. Škerget. Boundary element formulations for numerical solution of two-dimensional diffusion problems with variable coefficients. *Computers and Mathematics with Applications*. (In review); 2012.
5. J. Ravnik, M.A. AL-Jawary, L.C. Wrobel and L. Škerget. Domain decomposition technique for three-dimensional heat conduction problems with variable coefficients. (In preparation).

Conference Papers

1. M.A. AL-Jawary, L.C. Wrobel, M. Maischak. Numerical solution of a Dirichlet problem with variable coefficients by the boundary-domain integro-differential equation

-
- method. In: Lesnic D., editor. Eighth UK Conference on Boundary Integral Methods, University of Leeds Press; 2011, pp. 25-32. ISBN 9780853162957.
2. M.A. AL-Jawary, L.C. Wrobel, M. Maischak. Numerical solution of a Neumann problem with variable coefficients by the boundary-domain integral equation method. In: Lesnic D., editor. Eighth UK Conference on Boundary Integral Methods, University of Leeds Press; 2011, pp. 33-40. ISBN 9780853162957.
 3. M.A. AL-Jawary and L.C. Wrobel. Numerical solution of a mixed problem with variable coefficients by the boundary-domain integral and integro-differential equation methods. The International Association for Boundary Element Methods, IABEM 2011, Brescia, Italy, University of Brescia Press; 2011, pp.19-26.
 4. M.A. AL-Jawary, J. Ravnik, L.C. Wrobel and L. Škerget. Novel BEM formulations for numerical solution of two-dimensional diffusion problems with variable coefficients. In: Eds.: A. Nowak, R.A. Bialecki. ECCOMAS Special Interest Conference, Numerical Heat Transfer, 4-6 September 2012, Gliwice-Wroclaw, Poland, (accepted for presentation and publication).

Contents

Abstract	ii
Declaration of own work	iv
Acknowledgements	vi
Author's Publications	vii
1 Introduction	1
1.1 Aims and Objectives	2
1.2 Outline of contributions of this thesis	2
1.3 Summary of the thesis	3
2 The boundary element method	5
2.1 Two-dimensional Laplace's equation	6
2.2 Two-dimensional boundary integral equation	7
2.3 The boundary element method	12
2.4 Numerical solutions using Matlab	16
2.5 Examples	17
2.5.1 Example 1:	17
2.5.2 Example 2:	19
2.5.3 Example 3:	21
2.6 Concluding remarks	23
3 Heat conduction problems with variable coefficients	24
3.1 Introduction	24
3.2 Reduction of the BVP to a BDIE/BDIDE	26
3.3 Boundary-domain integral/integro-differential equations (BDIE/BDIDE)	29

3.3.1	Dirichlet problem	29
3.3.2	Neumann problem	30
3.3.3	Mixed problem	32
3.4	Discretisation of the BDIE/BDIDE	33
3.4.1	Dirichlet problem	34
3.4.2	Neumann problem	35
3.4.3	Mixed problem	36
3.5	Assembling the system matrix and right-hand side for BDIE/BDIDE	37
3.6	Numerical results	41
3.6.1	Laplace's equation with mixed boundary conditions	41
3.6.2	Poisson's equation with mixed boundary conditions	41
3.6.3	Test 1	42
3.6.4	Test 2	42
3.6.5	Test 3	42
3.6.6	Test 4	42
3.6.7	Test 5	43
3.6.8	Test 6	43
3.6.9	Dirichlet problem	43
3.6.10	Neumann problem	48
3.6.11	Mixed problem	52
3.7	Concluding remarks	54
4	Helmholtz equation with variable coefficients	56
4.1	Introduction	56
4.2	Reduction of the Helmholtz equation to an integral equation	57
4.3	Boundary-domain integral/integro-differential equations (BDIE/BDIDE)	58
4.4	Discretisation of the BDIE/BDIDE	59
4.4.1	Discretisation of the BDIE	59
4.4.2	Discretisation of the BDIDE	60
4.5	Numerical results	61
4.5.1	Numerical results when both $a(x)$ and $k(x)$ constant	61
4.5.2	BDIE/BDIDE for higher wave numbers	72
4.6	Numerical results for variable coefficients	73
4.6.1	Numerical results when $a(x)$ variable and $k(x)$ constant	74

4.6.2	Numerical results when $a(x)$ constant and $k(x)$ variable	77
4.6.3	Numerical results when both $a(x)$ and $k(x)$ variable	80
4.7	Conclusion	83
5	Radial integration method	84
5.1	Introduction	84
5.2	Transformation of domain integrals to boundary using RIM	87
5.2.1	RIM formulation for domain integrals with known integrand	87
5.2.2	RIM formulation for domain integrals with unknown integrand	90
5.3	Applications of the RIM	95
5.3.1	The RIM for transforming domain integrals with known integrand	95
5.3.2	The RIM for transforming domain integrals with unknown integrand	97
5.4	Conclusion	98
6	Radial integration method for heat conduction with variable coefficients	100
6.1	Introduction	100
6.2	Integral equation for heat conduction with variable coefficients	101
6.2.1	Boundary-domain integral equation (BDIE)	102
6.2.2	Boundary-domain integro-differential equation (BDIDE)	102
6.3	Transformation of domain integrals to boundary using RIM	103
6.3.1	Transformation of heat source domain integral to the boundary	103
6.3.2	RIM formulation for domain integrals with unknown integrand	103
6.4	The radial integration boundary integral and integro-differential equations	105
6.4.1	The radial integration boundary integral equation (RIBIE)	105
6.4.2	The radial integration boundary integro-differential equation (RIBIDE)	105
6.5	Discretisation of the RIBIE and RIBIDE	106
6.5.1	Discretisation of the RIBIE	106
6.5.2	Discretisation of the RIBIDE	107
6.6	Numerical results	108
6.6.1	Poisson's equation with mixed boundary conditions	108
6.6.2	Test 1	110
6.6.3	Test 2	111
6.6.4	Test 3	113
6.7	Conclusion	115

7	RIBIE and RIBIDE for Helmholtz equation with variable coefficients	116
7.1	Introduction	116
7.2	Integral equation for the Helmholtz equation with variable coefficients . . .	117
7.2.1	Boundary-domain integral equation (BDIE)	117
7.2.2	Boundary-domain integro-differential equation (BDIDE)	118
7.3	Transformation of domain integrals to boundary using RIM	118
7.3.1	Transformation of right-hand side domain integral to the boundary .	118
7.3.2	RIM formulation for domain integrals with unknown integrand . . .	118
7.4	The radial integration boundary integral and integro-differential equations .	120
7.4.1	The radial integration boundary integral equation (RIBIE)	120
7.4.2	The radial integration boundary integro-differential equation (RIBIDE)	121
7.5	Discretisation of the RIBIE and RIBIDE	121
7.5.1	Discretisation of the RIBIE	121
7.5.2	Discretisation of the RIBIDE	122
7.6	Numerical results	123
7.6.1	Numerical results when $a(x)$ variable and $k(x)$ constant	124
7.6.2	Numerical results when $a(x)$ constant and $k(x)$ variable	128
7.6.3	Numerical results when both $a(x)$ and $k(x)$ variable	131
7.7	Conclusion	134
8	RIBIE and RIBIDE for diffusion equation with variable coefficients	135
8.1	Introduction	135
8.2	Reduction of diffusion equation to a BDIE/BDIDE	136
8.2.1	Boundary-domain integral equation (BDIE)	140
8.2.2	Boundary-domain integro-differential equation (BDIDE)	140
8.3	Transformation of domain integrals to the boundary using RIM	141
8.3.1	RIM formulation for domain integrals with known integrand	141
8.3.2	RIM formulation for domain integrals with unknown integrand . . .	142
8.4	The radial integration boundary integral and integro-differential equations .	143
8.4.1	The radial integration boundary integral equation (RIBIE)	144
8.4.2	The radial integration boundary integro-differential equation (RIBIDE)	144
8.4.3	Discretisation of the RIBIE	145
8.4.4	Discretisation of the RIBIDE	146
8.5	Implementation and time marching solution scheme	146

8.5.1	Assembling the system for RIBIE for heat conduction with variable coefficients	147
8.5.2	Assembling the system for RIBIE for diffusion with variable coefficients	150
8.5.3	Assembling the system for RIBIDE for diffusion with variable coefficients	151
8.6	Numerical results	152
8.6.1	Numerical results for homogeneous diffusion equation with constant $a(x)$	152
8.6.2	Numerical results for non-homogeneous diffusion equation with constant $a(x)$	157
8.6.3	Numerical results for non-homogeneous diffusion equation with variable $a(x)$	159
8.7	Concluding remarks	165
9	Conclusions and Future Work	167
9.1	Conclusions	167
9.2	Future work	169
	References	171
	Appendix A	179
A	Matlab codes for the three examples in chapter 2	179
A.1	Main program for example 1	179
A.2	Main program for example 2	184
A.3	Main program for example 3	187
	Appendix B	191
B	Numerical implementation for chapters 3 and 4	192
B.1	Computation of matrix A	192
B.1.1	Computation of sub-matrix Q'	192
B.2	Mapping the reference triangle to an arbitrary triangle	195
B.2.1	Computation of sub-matrix K'	196
B.3	Numerical integration of boundary integrals	196

B.3.1	Mapping reference interval to arbitrary interval	196
B.4	Numerical integration of domain integrals	197
B.4.1	Symmetric Gauss quadrature formula for unit right triangle	197
B.4.2	Gauss Legendre quadrature formula for unit right triangle	198
B.4.3	Other useful quadrature formulas	201
B.5	Treatment of weak singularity for domain integrals using Duffy transformation	202
 Appendix C		204
C	Matlab codes for calculating of the radial integral in chapter 5	205
 Appendix D		211
D	Numerical results for RIBIE for heat conduction	212
D.1	Laplace's equation with mixed boundary conditions	212
D.2	Poisson's equation with mixed boundary conditions	213
D.3	Variable coefficients	213

List of Figures

2.1	Laplace's equation in two dimensions for a bounded region Ω	6
2.2	Domain which excludes a circle of radius ϵ centred at source point y	8
2.3	Semi-circle of radius ϵ centred at source point y	10
2.4	Illustration of internal angle α at source point y	11
2.5	Curve approximated by boundary elements $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_N$	12
2.6	Linear element definitions	15
2.7	Domain plot for Example 1	18
2.8	Surface plot of solution for Example 1	19
2.9	Domain plot for Example 2	20
2.10	Surface plot of solution for Example 2	21
2.11	Domain plot for Example 3	22
2.12	Surface plot of solution for Example 3	22
3.1	Discretization of the domain Ω	33
3.2	Simple mesh	38
3.3	Relative error for Laplace's equation; when $J=1089$, $r(1089) \approx 1.62 \times 10^{-11}$	44
3.4	Relative error for Poisson's equation; when $J=1089$, $r(1089) \approx 1.00 \times 10^{-4}$	44
3.5	Relative error for Test 1; when $J=1089$, $r(1089) \approx 4.84 \times 10^{-10}$	44
3.6	Relative error for Test 2; when $J=1089$, $r(1089) \approx 1.42 \times 10^{-4}$	44
3.7	Relative error for Test 3; when $J=656$, $r(656) \approx 1.13 \times 10^{-4}$	45
3.8	Relative error for Test 4; when $J=925$, $r(925) \approx 1.05 \times 10^{-7}$	45
3.9	Relative error for Test 5; when $J=1089$, $r(1089) \approx 3.22 \times 10^{-9}$	45
3.10	Relative error for Test 6; when $J=1089$, $r(1089) \approx 1.60 \times 10^{-4}$	45
3.11	Relative error for Laplace's equation; when $J=1089$, $r(1089) \approx 4.09 \times 10^{-7}$	46
3.12	Relative error for Poisson's equation; when $J=1089$, $r(1089) \approx 1.22 \times 10^{-6}$	46
3.13	Relative error Test 1; when $J=1089$, $r(1089) \approx 4.25 \times 10^{-7}$	46

3.14	Relative error Test 2; when $J=1089$, $r(1089) \approx 2.60 \times 10^{-6}$	46
3.15	Relative error for Test 3; when $J=656$, $r(656) \approx 3.29 \times 10^{-4}$	47
3.16	Relative error for Test 4; when $J=925$, $r(925) \approx 2.20 \times 10^{-4}$	47
3.17	Relative error for Test 5; when $J=1089$, $r(1089) \approx 5.00 \times 10^{-7}$	47
3.18	Relative error for Test 6; when $J=1089$, $r(1089) \approx 7.50 \times 10^{-6}$	47
3.19	Relative error for Laplace's equation; when $J=1089$, $r(1089) \approx 5.9 \times 10^{-6}$	49
3.20	Relative error for Poisson's equation; when $J=1089$, $r(1089) \approx 5.09 \times 10^{-5}$	49
3.21	Relative error for Test 1; when $J=1089$, $r(1089) \approx 6.36 \times 10^{-6}$	50
3.22	Relative error for Test 2; when $J=1089$, $r(1089) \approx 1.33 \times 10^{-4}$	50
3.23	Relative error for Test 3; when $J=656$, $r(656) \approx 1.67 \times 10^{-3}$	51
3.24	Relative error for Test 4; when $J=925$, $r(925) \approx 2.06 \times 10^{-3}$	51
3.25	Relative error for Test 5; when $J=1089$, $r(1089) \approx 6.60 \times 10^{-6}$	51
3.26	Relative error for Test 6; when $J=1089$, $r(1089) \approx 2.00 \times 10^{-4}$	51
3.27	Relative errors for Laplace's equation	53
3.28	Relative errors for Poisson's equation	53
3.29	Relative errors for Test 1	53
3.30	Relative errors for Test 2	53
3.31	Relative errors for Test 3	54
3.32	Relative errors for Test 4	54
3.33	Relative errors for Test 5	54
3.34	Relative errors for Test 6	54
4.1	Relative error for test 1, when $J=1089$, $r(1089) \approx 4.00 \times 10^{-4}$	62
4.2	Relative error for test 2, when $J=925$, $r(925) \approx 1.10 \times 10^{-3}$	62
4.3	Relative error for test 1, when $J=1089$, $r(1089) \approx 2.64 \times 10^{-6}$	63
4.4	Relative error for test 2, when $J=925$, $r(925) \approx 2.00 \times 10^{-4}$	63
4.5	Relative error for test 1, when $J=1089$, $r(1089) \approx 2.00 \times 10^{-4}$	64
4.6	Relative error for test 2, when $J=925$, $r(925) \approx 2.00 \times 10^{-3}$	64
4.7	Relative error for test 1, when $J=1089$, $r(1089) \approx 4.00 \times 10^{-4}$	64
4.8	Relative error for test 2, when $J=925$, $r(925) \approx 1.10 \times 10^{-3}$	64
4.9	Relative error for test 1 with E-NC; when $J=1089$, $r(1089) \approx 1.57 \times 10^{-5}$	65
4.10	Relative error for test 2 with M-NC; when $J=925$, $r(925) \approx 1.00 \times 10^{-3}$	65
4.11	Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-9}$	67
4.12	Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$	67

4.13	Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$. . . 68
4.14	Surface plot of solution for test 4, when $J=925$, $r(925) \approx 1.00 \times 10^{-4}$. . . 68
4.15	Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 1.23 \times 10^{-6}$. . . 68
4.16	Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 2.43 \times 10^{-6}$. . . 68
4.17	Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 2.90 \times 10^{-6}$. . . 69
4.18	Surface plot of solution for test 4, when $J=925$, $r(925) \approx 1.00 \times 10^{-4}$. . . 69
4.19	Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 2.85 \times 10^{-6}$. . . 69
4.20	Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$. . . 69
4.21	Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 3.00 \times 10^{-4}$. . . 70
4.22	Surface plot of solution for test 4, when $J=925$, $r(925) \approx 3.10 \times 10^{-3}$. . . 70
4.23	Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 1.68 \times 10^{-6}$. . . 70
4.24	Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 5.00 \times 10^{-4}$. . . 70
4.25	Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$. . . 71
4.26	Surface plot of solution for test 4, when $J=925$, $r(925) \approx 7.00 \times 10^{-4}$. . . 71
4.27	Surface plot of solution for test 1 with E-NC, $r(1089) \approx 3.36 \times 10^{-6}$ 72
4.28	Surface plot of solution for test 2 with E-NC, $r(1089) \approx 4.11 \times 10^{-5}$ 72
4.29	Surface plot of solution for test 3 with E-NC, $r(1089) \approx 2.30 \times 10^{-5}$ 72
4.30	Surface plot of solution for test 4 with M-NC, $r(925) \approx 1.00 \times 10^{-4}$ 72
4.31	Relative errors for test 1 using BDIDE 73
4.32	Relative errors for test 1 using BDIE 73
4.33	Acoustic pressure distribution along the line $x_2 = 2.875$ in test 1 75
4.34	Relative error for BDIDE and BDIE for test 1 75
4.35	Relative error for BDIDE and BDIE for test 2 76
4.36	Acoustic pressure distribution along the line $x_2 = 1.875$ in test 3 77
4.37	Relative error for BDIDE and BDIE for test 3 77
4.38	Acoustic pressure distribution along the line $x_2 = 0.875$ in test 4 78
4.39	Relative error for BDIDE and BDIE for test 4 78
4.40	Relative error for BDIDE and BDIE for test 5 79
4.41	Acoustic pressure distribution along the line $x_2 = 2.875$ in test 6 80
4.42	Relative error for BDIDE and BDIE for test 6 80
4.43	Acoustic pressure distribution along the line $x_2 = 1.875$ in test 7 81
4.44	Relative error for BDIDE and BDIE for test 7 81
4.45	Acoustic pressure distribution along the line $x_2 = 1.875$ in test 8 82

4.46	Relative error for BDIDE and BDIE for test 8	82
5.1	Relationship between differential elements $r d\theta$ and $d\Gamma$	87
5.2	Integration along radial direction r	89
5.3	Relationship between distances	93
6.1	Temperature distribution along the line $x_2 = 2.875$	109
6.2	Relative and RMS errors for RIBIDE method for Poisson's equation	110
6.3	Relative and RMS errors for RIBIE method for Poisson's equation	110
6.4	Temperature distribution along the line $x_2 = 2.875$	111
6.5	Relative and RMS errors for RIBIDE method for test 1	112
6.6	Relative and RMS errors for RIBIE method for test 1	112
6.7	Temperature distribution along the line $x_2 = 1.875$	112
6.8	Relative and RMS errors for RIBIDE method for test 2	113
6.9	Relative and RMS errors for RIBIE method for test 2	113
6.10	Temperature distribution along the line $x_2 = 2.875$	114
6.11	Relative and RMS errors for RIBIDE method for test 3	114
6.12	Relative and RMS errors for RIBIE method for test 3	114
7.1	Acoustic pressure distribution along the line $x_2 = 2.875$	124
7.2	Relative and RMS errors for RIBIDE method for test 1	125
7.3	Relative and RMS errors for RIBIE method for test 1	125
7.4	Acoustic pressure distribution along the line $x_2 = 1.875$	126
7.5	Relative and RMS errors for RIBIDE method for test 2	126
7.6	Relative and RMS errors for RIBIE method for test 2	126
7.7	Acoustic pressure distribution along the line $x_2 = 2.875$	127
7.8	Relative and RMS errors for RIBIDE method for test 3	128
7.9	Relative and RMS errors for RIBIE method for test 3	128
7.10	Acoustic pressure distribution along the line $x_2 = 0.875$	129
7.11	Relative and RMS errors for RIBIDE method for test 4	129
7.12	Relative and RMS errors for RIBIE method for test 4	129
7.13	Acoustic pressure distribution along the line $x_2 = 2.875$	130
7.14	Relative and RMS errors for RIBIDE method for test 5	131
7.15	Relative and RMS errors for RIBIE method for test 5	131
7.16	Acoustic pressure distribution along the line $x_2 = 1.875$	132

7.17	Relative and RMS errors for RIBIDE method for test 6	132
7.18	Relative and RMS errors for RIBIE method for test 6	132
7.19	Acoustic pressure distribution along the line $x_2 = 1.875$	133
7.20	Relative and RMS errors for RIBIDE method for test 7	134
7.21	Relative and RMS errors for RIBIE method for test 7	134
8.1	Simple mesh with \circ for u and \diamond for flux q	147
8.2	Relative and RMS errors for RIBIE method for test 1 with $\Delta t = 0.1$	153
8.3	Relative and RMS errors for RIBIDE method for test 1 with $\Delta t = 0.1$	153
8.4	Temperature distribution along the line $x_2 = 0.5$ with $x_1 = 0.25$	154
8.5	Temperature distribution along the line $x_2 = 0.5$ with $x_1 = 0.5$	154
8.6	Temperature distribution along the line $x_2 = 0.5$ with $x_1 = 0.75$	154
8.7	Relative and RMS errors for RIBIE method for test 4 with $\Delta t = 0.1$	158
8.8	Relative and RMS errors for RIBIDE method for test 4 with $\Delta t = 0.1$	158
8.9	Relative and RMS errors for RIBIE method for test 5 with $\Delta t = 0.1$	159
8.10	Relative and RMS errors for RIBIDE method for test 5 with $\Delta t = 0.1$	159
8.11	Relative and RMS errors for RIBIE method for test 5 with $\Delta t = 0.01$	159
8.12	Relative and RMS errors for RIBIDE method for test 5 with $\Delta t = 0.01$	159
8.13	Relative and RMS errors for RIBIE method for test 6 with $\Delta t = 0.1$	160
8.14	Relative and RMS errors for RIBIDE method for test 6 with $\Delta t = 0.1$	160
8.15	Relative and RMS errors for each time step using RIBIE for test 7	161
8.16	Relative and RMS errors for each time step using RIBIDE for test 7	161
8.17	Relative and RMS errors for final time step using RIBIE for test 7	162
8.18	Relative and RMS errors for final time step using RIBIDE for test 7	162
8.19	Relative and RMS errors for each time step using RIBIE for test 8	163
8.20	Relative and RMS errors for each time step using RIBIDE for test 8	163
8.21	Relative and RMS errors for final time step using RIBIE for test 8	164
8.22	Relative and RMS errors for final time step using RIBIDE for test 8	164
8.23	Relative and RMS errors for each time step using RIBIE for test 9	164
8.24	Relative and RMS errors for each time step using RIBIDE for test 9	165
8.25	Relative and RMS errors for final time step using RIBIE for test 9	165
8.26	Relative and RMS errors for final time step using RIBIDE for test 9	165
B.1	Mapping reference triangle to arbitrary triangle	195

B.2	Mapping reference interval to arbitrary interval	196
B.3	Mapping unit right triangle to unit square	198
D.1	Relative and RMS errors for BIE method for Laplace's equation	213
D.2	Relative and RMS errors for RIBIE method for Poisson's equation	213
D.3	Relative and RMS errors for RIBIE method for variable coefficients	214

List of Tables

2.1	Computed temperatures along line $x_2 = 0.8$	18
2.2	Convergence of the approximate solution of Example 1	19
2.3	Convergence of the approximate solution of Example 2	20
2.4	Convergence of the approximate solution of Example 3	23
3.1	Relative errors for all tests by using BDIE and BDIDE	52
4.1	Computed acoustic pressure along line $x_2 = 2.5$	74
4.2	Computed acoustic pressure for some selected interior points	75
4.3	Computed acoustic pressure along line $x_2 = 1.5$	76
4.4	Computed acoustic pressure along line of $x_2 = 0.5$	78
4.5	Computed acoustic pressure for some selected interior points	79
4.6	Computed acoustic pressure along line $x_2 = 2.5$	80
4.7	Computed acoustic pressure along line $x_2 = 1.5$	81
4.8	Computed acoustic pressure along line $x_2 = 1.5$	82
5.1	Commonly used radial basis functions $\phi(R)$	91
5.2	Computational results of the RIM for the integral in test 1	96
5.3	Computational results for non-augmented RBFs	97
5.4	Computational results for augmented RBFs by LP without interior nodes	97
5.5	Computational results for augmented RBFs by LP with interior nodes	98
5.6	Computational results for augmented RBFs by QP with interior nodes	98
6.1	Computed temperatures along the line $x_2 = 2.5$	109
6.2	Computed temperatures along the line $x_2 = 2.5$	111
6.3	Computed temperatures along the line $x_2 = 1.5$	112
6.4	Computed temperatures along the line $x_2 = 2.5$	113

7.1	Computed acoustic pressure along line of $x_2 = 2.5$	124
7.2	Computed acoustic pressure along line of $x_2 = 1.5$	125
7.3	Computed acoustic pressure along line $x_2 = 2.5$	127
7.4	Computed acoustic pressure along line $x_2 = 0.5$	128
7.5	Computed acoustic pressure along line of $x_2 = 2.5$	130
7.6	Computed acoustic pressure along line of $x_2 = 1.5$	131
7.7	Computed acoustic pressure along line of $x_2 = 1.5$	133
8.1	Computed temperatures using RIBIE along line $x_2 = 0.5$ with $x_1 = 0.25$.	155
8.2	Computed temperatures using RIBIDE along line $x_2 = 0.5$ with $x_1 = 0.25$	155
8.3	Computed temperatures using RIBIE along line $x_2 = 0.5$ with $x_1 = 0.5$. .	156
8.4	Computed temperatures using RIBIDE along line $x_2 = 0.5$ with $x_1 = 0.5$.	156
8.5	Computed temperatures using RIBIE along line $x_2 = 0.5$ with $x_1 = 0.75$.	156
8.6	Computed temperatures using RIBIDE along line $x_2 = 0.5$ with $x_1 = 0.75$	157
B.1	Symmetric Gauss quadrature formula for unit right triangle	199
B.2	7-points quadrature formula	201
B.3	16-points quadrature formula	202
B.4	16-points Stroud quadrature formula	203

Nomenclature

LIST OF SYMBOLS

Ω : domain of the problem

$\partial\Omega$: boundary of the domain

$\partial_D\Omega$: Dirichlet boundary

$\partial_N\Omega$: Neumann boundary

∇ : gradient operator

$\nabla^2 = \Delta$: the Laplace operator

$x = (x_1, x_2)$: the cartesian coordinates of integration point (field point)

$y = (y_1, y_2)$: the cartesian coordinates of collocation point (source point)

$n(x)$: the external normal vector to the boundary $\partial\Omega$

$F(x, y)$: fundamental solution

r, θ : polar coordinate system

q : heat flux

J : total number of nodes

$r(J)$: relative error

$a(x)$: known variable coefficients

$k(x)$: a known variable wave number

Y_0 : the zero order Bessel function of the second kind

Δt : time step

LIST OF ABBREVIATIONS

AEM	Analog Equation Method
BDIE	Boundary Domain Integral Equation
BDIDE	Boundary Domain Integro-Differential Equation
BDIEM	Boundary Domain Integral Equation
BDIDEM	Boundary Bomain Integro-Differential Equation
BEM	Boundary Element Method
BIE	Boundary Integral Equation
BVP	Boundary-Value Problem
DRM	Dual Reciprocity Method
DRBEM	Dual Reciprocity Boundary Element Method
DRM-MD	Dual Reciprocity-Multi-Domain
FDM	Finite Difference Method
FEM	Finite Element Method
PDE	Partial Differential Equation
RBFs	Radial Basis Functions
RIM	Radial Integration Method
RIBIE	Radial Integration Boundary Integral Equation
RIBIDE	Radial Integration Boundary Integro-Differential Equation
RMS	Root Mean Square
2D	Two-Dimensional
3D	Three-Dimensional

Chapter 1

Introduction

The boundary element method (BEM) has become an efficient and popular alternative to the finite element method (FEM) because of its ability, at least for some problems with constant coefficients, of reducing a boundary-value problem (BVP) for a linear partial differential equation (PDE) defined in a domain to an integral equation defined on the boundary, leading to a simplified discretisation process with boundary elements only. The main requirement for the reduction of the PDE to a boundary integral equation (BIE) is that a fundamental solution to the PDE must be available. Such fundamental solutions are well known for many PDEs with constant coefficients, see [1–6], but are not generally available when the coefficients of the original PDE are variable. The solution of PDEs with variable coefficients is important in many practical engineering problems and there is an effort to develop BEM formulations to treat these problems, see [7].

In the last three decades, many researchers proposed formulations for the application of the BEM for PDEs with variable coefficients. Although some success has been achieved, this has been limited to specific forms of the coefficients for some engineering problems. However, this is hampered by the difficulty of finding appropriate fundamental solutions for general forms of the coefficients.

In this thesis, we investigate mathematical formulations leading to a boundary-domain integral and integro-differential equations (BDIE and BDIDE) based on the use of a parametrix (Levi function), which is usually available, for three important problems in heat conduction, wave propagation and diffusion.

1.1 Aims and Objectives

The motivation of this research is to derive BDIE and BDIDE formulations to represent PDEs with variable coefficients. The radial integration method (RIM) is then used to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals. This is achieved through studying three problems, based on Laplace, Helmholtz and diffusion equations with variable coefficients. These approaches allowed to derive boundary-only integral equations for such problems. Numerical examples are presented for several simple problems with square and circular domains, for which exact solutions are available. It is shown that the present methods produce accurate results even with coarse meshes. The numerical results also show that satisfactory results and convergence are obtained with mesh refinement.

1.2 Outline of contributions of this thesis

The main contributions to knowledge of this thesis can be summarised in the following points:

- 1) Derivation of BDIE and BDIDE formulations for stationary heat transfer in isotropic media associated with Dirichlet, Neumann and mixed boundary conditions.
- 2) Extension of the BDIE and BDIDE formulations to the treatment of the two-dimensional Helmholtz equation with variable coefficients.
- 3) Modifications have been introduced to the RIM particularly the fact that the radial integral is calculated by using a pure boundary-only formulation which relaxes the “star-shaped” requirement of the RIM, as the straight path from the source point to any field point will always exist.
- 4) A new type of boundary-only integral equation technique is developed for the non-homogeneous heat conduction problems with variable coefficients based on the use of a parametrix. The RIM is used to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals.
- 5) A new type of boundary-only integral equation technique is developed for the Helmholtz equation when the material parameters and wave number vary within the medium. The RIM is used to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals.
- 6) Derivation of the BDIE and BDIDE formulations for non-homogeneous diffusion equa-

tion with variable coefficients. The RIM is used to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals.

1.3 Summary of the thesis

The present chapter is the first introductory part of the thesis. It gives some preliminary background information about the subject, presents the aims and objectives of the research, outlines the contributions of the present thesis and provides the structure for the remaining eight chapters.

Chapter 2 presents the boundary integral equation formulation for solving two-dimensional BVPs with mixed boundary conditions. The BEM is then applied to solve the corresponding BIE, with the boundary discretised by using constant elements.

The third chapter provides the BDIE and BDIDE formulations for stationary heat transfer in isotropic materials with variable coefficients associated with Dirichlet, Neumann and mixed boundary conditions. The Neumann BVP is not unconditionally and uniquely solvable and neither is the corresponding integral equation. Therefore, the resulting system of linear algebraic equations will be either singular or ill-conditioned. This problem is treated using a perturbation technique.

Chapter 4 aims to extend the BDIE and BDIDE formulations to the treatment of the two-dimensional Helmholtz equation with variable coefficients. Four possible cases are investigated, first of all when both material parameters and wave number are constant, in which case the zero-order Bessel function of the second kind is used as fundamental solution. Moreover, when the material parameters are variable (with constant or variable wave number), a parametrix is adopted to reduce the Helmholtz equation to a BDIE or BDIDE. However, when material parameters are constant (with variable wave number), the standard fundamental solution for the Laplace equation is used in the formulation. In order to show the accuracy of these methods, some test examples with square and circular domains are given.

In Chapter 5, the RIM is introduced and discussed in detail. For domain integrals consisting of known functions the transformation into boundary integrals is straightforward, while for domain integrals that include unknown variables the transformation is accomplished with the use of radial basis functions (RBFs) augmented by polynomials to approximate the unknown quantities. The most attractive feature of the method is that the transfor-

mations are very simple and have similar forms for both 2D and 3D problems. It can also remove various singularities appearing in the domain integrals, and treat different types of domain integrals in a unified way since it does not resort to particular solutions as in the dual reciprocity method (DRM). Some numerical examples are given to demonstrate the efficiency of the presented method.

Chapter 6 presents the application of the RIM to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals. Then, a new type of boundary-only integral equation technique is developed for the non-homogeneous heat conduction problems with variable coefficients.

In Chapter 7 the RIM is used again to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals. Then, a new type of boundary-only integral equation technique is developed for the non-homogeneous Helmholtz equation when the material parameters and wave number vary within the medium.

Chapter 8 presents the derivation of BDIE and BDIDE formulations for the parabolic time dependent diffusion equation. Then, the RIM is used to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals. Moreover, three possible cases are investigated, homogenous, non-homogeneous and variable coefficient diffusion equations.

Finally, Chapter 9, summarises the conclusions drawn as a result of the research work presented herein. This chapter also discusses some suggestions for future work.

Chapter 2

The boundary element method

The boundary element method (BEM) is an important computational tool used by researchers in many fields in the physical and engineering sciences [4].

The BEM is one of the methods for the numerical solution of boundary integral equations. It presents many advantages, first of all, only the boundary of the domain needs to be discretised. Especially in two dimensions, where the boundary is just a curve, this allows very simple data input and storage methods. Also, the smaller computer time due to a lesser number of nodes and elements (but a fully populated matrix). Moreover, high accuracy and the stability of numerical computations because of the utilization of fundamental solutions. Furthermore, the BEM method has another important advantage in the case of problems with infinite or semi-infinite domains. Only the finite surface of the infinite domain has to be discretised for these problems, and the solution at any arbitrary point in the domain can be found after determining the unknown boundary data, see [5].

In order to reduce a boundary-value problem (BVP) for a partial differential equation (PDE) to a boundary-integral equation useful for numerical implementation, the necessary condition is the knowledge of the fundamental solution of the governing differential operator. Such fundamental solutions are well known for many PDEs with constant coefficients, see [1–6].

The boundary is discretised by elements, where the continuous function and its normal derivative along the boundary are approximated using interpolation functions. Many types of interpolation functions (constant, linear and quadratic) have been implemented to arrive at a system of linear algebraic equations, but this system unlike FEM is fully populated [1–6].

In the present chapter, the boundary integral formulation for the Laplace equation will be derived, and then the BEM will be implemented. Numerical examples are presented for three problems with different geometries, for which exact solutions are available, to demonstrate the accuracy of the BEM.

2.1 Two-dimensional Laplace's equation

Perhaps a good starting point for introducing the BEM is through solving BVPs governed by Laplace's equation for a two-dimensional bounded body Ω , with prescribed temperature $\bar{u}(x)$ on part $\partial_D\Omega$ of the boundary $\partial\Omega$ and prescribed heat flux $\bar{t}(x)$ on the remaining $\partial_N\Omega$ part of $\partial\Omega$:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = 0, \quad x \in \Omega. \quad (2.1)$$

subject to boundary conditions:

$$u(x) = \bar{u}(x), \quad x \in \partial_D\Omega, \quad (2.2)$$

$$q = \frac{\partial u}{\partial n}(x) = \bar{t}(x), \quad x \in \partial_N\Omega, \quad (2.3)$$

where $\nabla^2 = \nabla \cdot \nabla$ is the Laplace operator, u is the dependent variable, x_1, x_2 are cartesian coordinates of the point $x = (x_1, x_2)$, $n(x)$ is the external normal vector to the boundary $\partial\Omega$, and $\bar{u}(x)$ and $\bar{t}(x)$ are known functions on the Dirichlet part $\partial_D\Omega$ and the Neumann part $\partial_N\Omega$, respectively.

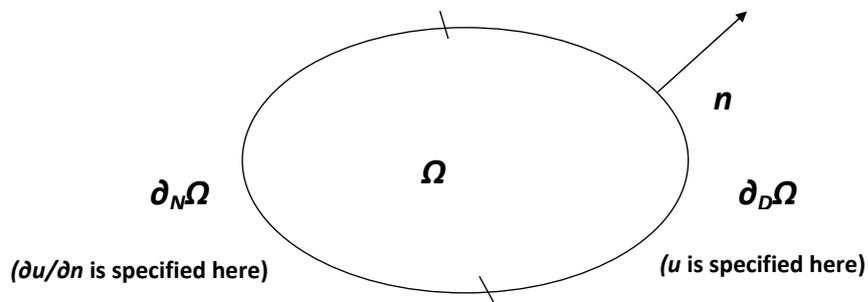


Figure 2.1: Laplace's equation in two dimensions for a bounded region Ω

2.2 Two-dimensional boundary integral equation

The fundamental solution of the two-dimensional Laplace equation with source point at $y = (y_1, y_2)$ is [6, 8],

$$F(x, y) = \frac{1}{2\pi} \ln r, \quad (2.4)$$

where $r = |x - y| = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$.

In the BEM formulation, the problem is reduced from within the domain to its boundary.

We use the well-known Green's second identity to do this, see e.g. [8–10],

$$\int_{\Omega} (u \nabla^2 v - v \nabla^2 u) d\Omega = \int_{\partial\Omega} \left(u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right) d\Gamma \quad (2.5)$$

Here v, u have continuous first and second derivatives.

Let u satisfy Eq.(2.1), i.e. $\nabla^2 u = 0$, everywhere in the solution domain Ω . Then, take $v(x)$ as the fundamental solution $F(x, y)$ which satisfies $\nabla_x^2 F(x, y) = 0$ everywhere except at the point $x = y$, where $F(x, y)$ is singular. If y does not lie in the region $\bar{\Omega} = \Omega \cup \partial\Omega$, then

$$\int_{\partial\Omega} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x) = 0, \quad y \notin \Omega \cup \partial\Omega. \quad (2.6)$$

In order to use Eq.(2.5) when the point y lies in the region $\Omega \cup \partial\Omega$, we must deal with the singularity at this point. Let us deal first with the case in which y lies in the interior of Ω , [5].

Let us surround y with a small circle of radius ϵ , and then examine the solution as $\epsilon \rightarrow 0$.

The new domain is $\Omega \setminus \Omega_\epsilon$ and the boundary is $\partial\Omega \cup \partial\Omega_\epsilon$, so:

$$\int_{\Omega \setminus \Omega_\epsilon} (u(x) \nabla_x^2 F(x, y) - F(x, y) \nabla^2 u(x)) d\Omega(x) = \int_{\partial\Omega \cup \partial\Omega_\epsilon} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x). \quad (2.7)$$

Within the domain $\Omega \setminus \Omega_\epsilon$, $\nabla^2 u = 0$ and $\nabla_x^2 F(x, y) = 0$. So, the left-hand side of Eq.(2.7) is zero and the right-hand side is now:

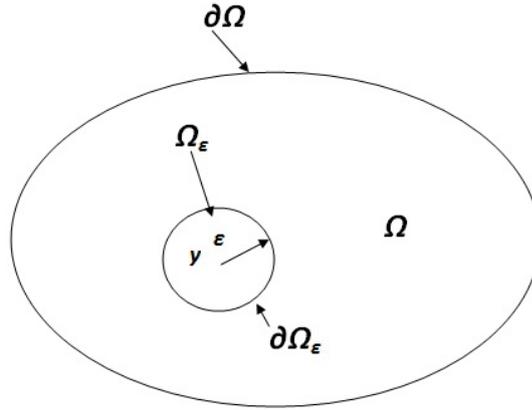


Figure 2.2: Domain which excludes a circle of radius ϵ centred at source point y

$$0 = \int_{\partial\Omega} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x) + \int_{\partial\Omega_\epsilon} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x). \quad (2.8)$$

Let us first analyse the first integral on the second term in Eq.(2.8). Initially, the value of u at the source point, $u(y)$, is subtracted from and added to the value of $u(x)$ to give

$$\int_{\partial\Omega_\epsilon} u(x) \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x) = \int_{\partial\Omega_\epsilon} [u(x) - u(y)] \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x) + u(y) \int_{\partial\Omega_\epsilon} \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x) \quad (2.9)$$

with

$$\frac{\partial F(x, y)}{\partial n(x)} = \sum_{i=1}^2 \frac{\partial F(x, y)}{\partial x_i} n_i$$

From Eq.(2.4) we get,

$$\frac{\partial F(x, y)}{\partial x_i} = \frac{1}{2\pi r} \frac{\partial r}{\partial x_i}$$

and

$$\frac{\partial r}{\partial x_i} = \frac{\partial \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}}{\partial x_i} = \frac{1}{2} \frac{2(x_i - y_i)}{r} = \frac{x_i - y_i}{r}$$

so that,

$$\frac{\partial F(x, y)}{\partial x_i} = \frac{1}{2\pi r} \frac{x_i - y_i}{r}$$

and then,

$$\frac{\partial F(x, y)}{\partial n(x)} = \sum_{i=1}^2 \frac{\partial F(x, y)}{\partial x_i} n_i = \sum_{i=1}^2 \frac{1}{2\pi r} (-n_i) \cdot (n_i) = -\frac{1}{2\pi r}$$

since

$$n_i = -\frac{x_i - y_i}{r} \quad \text{and} \quad n_i \cdot n_i = 1.$$

Now, substituting the expression of $\frac{\partial F(x, y)}{\partial n(x)}$ in the second integral of Eq.(2.9), and writing the integral in polar coordinates, in which case $d\Gamma = \epsilon d\theta$, we get

$$\lim_{\epsilon \rightarrow 0} \int_{\partial\Omega_\epsilon} \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x) = -\lim_{\epsilon \rightarrow 0} \int_0^{2\pi} \frac{1}{2\pi\epsilon} \epsilon d\theta = -1. \quad (2.10)$$

We still need to evaluate the limit of the first integral on the right-hand side of Eq.(2.9). Using the same reasoning as before, and assuming the function u is continuous at y , we achieve,

$$\lim_{\epsilon \rightarrow 0} \int_{\partial\Omega_\epsilon} [u(x) - u(y)] \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x) = 0. \quad (2.11)$$

Thus, using Eq.(2.10) and Eq.(2.11),

$$\int_{\partial\Omega_\epsilon} u(x) \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x) = -u(y). \quad (2.12)$$

The same ideas can be applied to evaluate the limit for the second integral of the second term in Eq.(2.8), this gives

$$\lim_{\epsilon \rightarrow 0} \int_{\partial\Omega_\epsilon} \frac{\partial u(x)}{\partial n(x)} F(x, y) d\Gamma(x) = -\lim_{\epsilon \rightarrow 0} \int_0^{2\pi} \frac{1}{2\pi} \ln(\epsilon) \frac{\partial u(x)}{\partial n(x)} \epsilon d\theta = 0. \quad (2.13)$$

Substituting both values of the limit in Eq.(2.12) and Eq.(2.13) in the integral over $\partial\Omega_\epsilon$

in Eq.(2.8), we have

$$\lim_{\epsilon \rightarrow 0} \int_{\partial\Omega_\epsilon} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x) = -u(y). \quad (2.14)$$

The following integral equation is obtained from Eq.(2.7):

$$u(y) = \int_{\partial\Omega} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x), \quad y \in \Omega. \quad (2.15)$$

This equation is known as Green's third identity.

To obtain a boundary integral equation when y lies on the boundary $\partial\Omega$, we can follow the same procedure as before by excluding the point y by a semi-circle when y belongs to a smooth part of the boundary as in Figure 2.3:

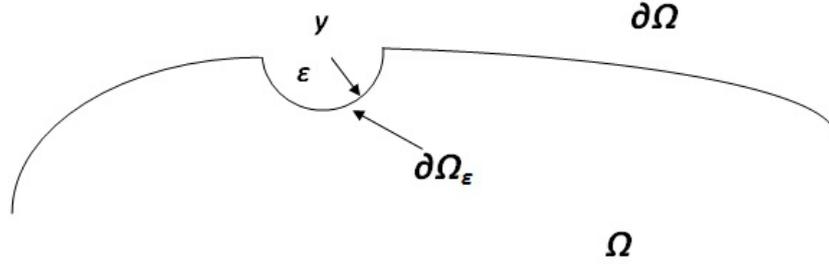


Figure 2.3: Semi-circle of radius ϵ centred at source point y

The only difference from the previous case is on the upper integration limit in Eq.(2.10) and Eq.(2.13), which is now π rather than 2π . Taking the limit when $\epsilon \rightarrow 0$ gives, for a point y on a smooth boundary,

$$\lim_{\epsilon \rightarrow 0} \int_{\partial\Omega_\epsilon} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x) = -\frac{1}{2}u(y) \quad (2.16)$$

and the following boundary integral equation is obtained:

$$\frac{1}{2}u(y) = \int_{\partial\Omega} \left[u(x) \frac{\partial F(x, y)}{\partial n(x)} - F(x, y) \frac{\partial u(x)}{\partial n(x)} \right] d\Gamma(x) \quad (2.17)$$

for every point y on a smooth part of the boundary.

The above is true if the point y is at a smooth point (i.e., a point with a unique tangent) on the boundary of Ω . If y happens to lie at some nonsmooth point, the coefficient will be changed, e.g. at a corner, the coefficient $\frac{1}{2}$ is replaced by $\frac{\alpha}{2\pi}$, where α is the internal angle at y , Figure 2.4, see e.g [4].

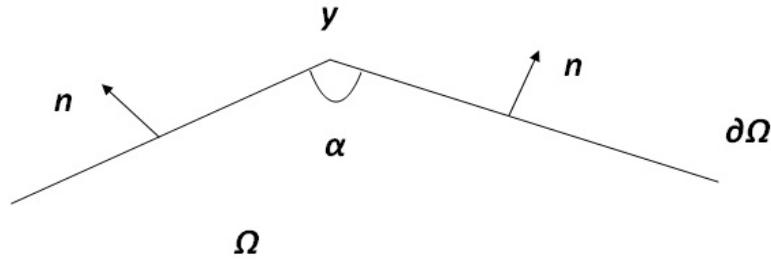


Figure 2.4: Illustration of internal angle α at source point y .

For convenience, we can write Eqs.(2.6), (2.15) and (2.17) as a single equation given by:

$$c(y)u(y) = \int_{\partial\Omega} u(x) \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x) - \int_{\partial\Omega} F(x, y) \frac{\partial u}{\partial n} d\Gamma(x) \quad (2.18)$$

$$c(y) = \begin{cases} 1, & \text{if } y \in \Omega \\ \frac{1}{2}, & \text{if } y \in \partial\Omega \text{ and } \partial\Omega \text{ smooth at } y \\ \frac{\alpha}{2\pi}, & \text{if } y \in \partial\Omega \text{ and } \partial\Omega \text{ not smooth at } y \text{ (corner with internal angle } \alpha) \\ 0, & \text{if } y \notin \Omega \cup \partial\Omega \end{cases}$$

The unknowns in Eq.(2.18) are the boundary values of u or $\frac{\partial u}{\partial n}$. We shall consider the Laplace equation with the following boundary conditions:

- 1) Dirichlet problem: u is given at every point y on the boundary.
- 2) Neumann problem: $\frac{\partial u}{\partial n}$ is given at every point y on the boundary.
- 3) Mixed case: Either Dirichlet or Neumann conditions are given at every point y on the boundary.

Substituting the boundary conditions 1) or 2) or 3) above in the Green identity (2.18) and applying it for $y \in \partial\Omega$, we arrive at a direct boundary-integral equation [6, 8, 9].

2.3 The boundary element method

In this section, we shall show how a simple numerical implementation of the BEM may be obtained to find approximations to the solution of the BVP for the Laplace equation.

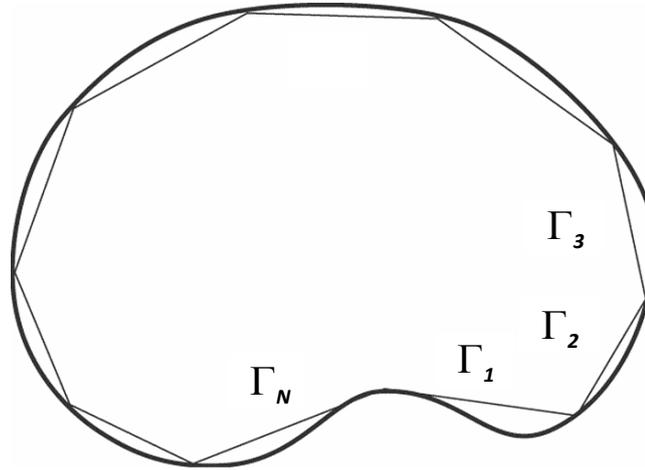


Figure 2.5: Curve approximated by boundary elements $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_N$

The first step of the BEM is to approximate the boundary $\partial\Omega$ by a polygon with N straight-line segments. That is, we make $\partial\Omega \simeq \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \dots \cup \Gamma_N$, as illustrated in Figure 2.5. We construct these segments by choosing N points, say $x^k = (x_1^k, x_2^k)$, $k = 1, 2, \dots, N$, on the boundary, and we define $x^{N+1} = x^1$. Then, Γ_k is the straight line joining x^k to x^{k+1} . These straight line segments Γ_k for $k = 1, 2, \dots, N$, are known as **boundary elements**. There are different types of boundary elements.

Case 1 (Constant elements):

We approximate the values of u and $\frac{\partial u}{\partial n}$ by constants over each boundary element. More specifically, we take the values at the midpoint of each element to be the values over the whole element. That is, for $k = 1, 2, \dots, N$, $u \approx u_k$ and $\frac{\partial u}{\partial n} \approx q_k$, where u_k and q_k are the values of u and $\frac{\partial u}{\partial n}$ at the midpoint of boundary element Γ_k .

Using these values, we can write (2.18) as an approximation, see e.g [2, 9]:

$$c(y)u(y) = \sum_{k=1}^N (u_k H_k(y) - q_k G_k(y)), \quad (2.19)$$

where

$$G_k(y) = \int_{\Gamma_k} F(x, y) d\Gamma(x), \quad (2.20)$$

$$H_k(y) = \int_{\Gamma_k} \frac{\partial F(x, y)}{\partial n(x)} d\Gamma(x). \quad (2.21)$$

If we know u_k and q_k , then we can use equation (2.19) to find the values of $u(y)$ for $y \in \Omega$. However, in a properly posed boundary value problem, either u_k or q_k (but not both) is known on any given portion of the boundary.

By using the collocation method, we collocate y in (2.19) at the midpoints of $\Gamma_1, \Gamma_2, \dots, \Gamma_N$. Therefore, one can use equation (2.19), firstly to find approximations to these unknown boundary values, before using it again to approximate the values at the interior points, so we get,

$$\frac{1}{2}u_m = \sum_{k=1}^N \left[u_k H_k(\bar{x}_m, \bar{y}_m) - q_k G_k(\bar{x}_m, \bar{y}_m) \right], \quad \text{for } m = 1, 2, \dots, N, \quad (2.22)$$

where (\bar{x}_m, \bar{y}_m) is the midpoint of Γ_m , and when y is on Γ_k , $c(y) = 1/2$.

It is convenient to write Eq.(2.22) as a system of equations in the form $\mathbf{Az}=\mathbf{b}$. The unknowns in \mathbf{z} are either u_k or q_k . The right hand side \mathbf{b} is made up of all the known values. In other words, for each $m=1,2,\dots,N$, we obtain:

$$\sum_{k=1}^N A_{mk} z_k = b_m, \quad \text{where } b_m = \sum_{k=1}^N \tilde{b}_{mk}. \quad (2.23)$$

We construct \mathbf{A} and $\tilde{\mathbf{b}}_{mk}$ by considering the two cases, namely (a) when u is given (Dirichlet condition) and (b) when $q = \frac{\partial u}{\partial n}$ is given (Neumann condition) over the boundary Γ_k , for each equation $m = 1, 2, \dots, N$ in Eq.(2.22).

So, we have the following cases for the construction of matrix \mathbf{A} and vector $\tilde{\mathbf{b}}_{mk}$,

$$A_{mk} = \begin{cases} -G_k(\bar{x}_m, \bar{y}_m), & \text{if } u \text{ given over } \Gamma_k; \\ H_k(\bar{x}_m, \bar{y}_m), & \text{if } q \text{ given over } \Gamma_k \text{ and } k \neq m, \\ H_k(\bar{x}_m, \bar{y}_m) - \frac{1}{2}, & \text{if } q \text{ given over } \Gamma_k \text{ and } k=m. \end{cases} \quad (2.24)$$

$$\tilde{b}_{mk} = \begin{cases} q_k G_k(\bar{x}_m, \bar{y}_m), & \text{if } q \text{ given over } \Gamma_k; \\ -u_k H_k(\bar{x}_m, \bar{y}_m), & \text{if } u \text{ given over } \Gamma_k \text{ and } k \neq m, \\ u_k (-H_k(\bar{x}_m, \bar{y}_m) + \frac{1}{2}), & \text{if } u \text{ given over } \Gamma_k \text{ and } k=m. \end{cases} \quad (2.25)$$

Therefore, after \mathbf{A} and \mathbf{b} are formed, we can solve for \mathbf{z} .

When all the values on the boundary are obtained, we can then use equation (2.19) to obtain values at any interior point in the domain Ω .

Case 2 (Linear elements):

The values of u and $q = \frac{\partial u}{\partial n}$ at any point on the element can be defined in terms of their nodal values and two linear interpolation functions $\Psi^1(t)$ and $\Psi^2(t)$. They can be represented by a reference coordinate t , see e.g [4], as shown in Figure 2.6. In order to write the explicit form for the coordinates of a point placed somewhere along the element as a function of the coordinates of the end points of this element, we use the reference coordinate t in the following way,

$$x_i(t) = \Psi^j(t)x_i^j, \quad (2.26)$$

where the repetition of the index j implies a summation; t is the reference coordinate along the element, which takes the values $-1, +1$, at the edges; x_i is the coordinate i ($i=1,2$ in $2D$) at the intermediate point defined by t ; Ψ^j is called the interpolation function associated to j , x_i^j represents the i coordinate of node j .

The two linear interpolation functions are,

$$\Psi^1(t) = \frac{1}{2}(1 - t); \quad \Psi^2(t) = \frac{1}{2}(1 + t). \quad (2.27)$$

Then (2.26) yields

$$x_1(t) = \Psi^1(t)x_1^1 + \Psi^2(t)x_1^2 = \frac{x_1^2 - x_1^1}{2}t + \frac{x_1^2 + x_1^1}{2},$$

$$x_2(t) = \Psi^1(t)x_2^1 + \Psi^2(t)x_2^2 = \frac{x_2^2 - x_2^1}{2}t + \frac{x_2^2 + x_2^1}{2}.$$

Also,

$$ds_k = \sqrt{dx_1^2 + dx_2^2} = \frac{L_k}{2}dt,$$

where L_k represents the length of the element Γ_k .

Now we are going to represent the functions which are defined along the elements (the

functions u and q), which can be represented in the following way,

$$u(t) = \Psi^1(t)u^1 + \Psi^2(t)u^2 = [\Psi^1(t) \ \Psi^2(t)] \begin{Bmatrix} u^1 \\ u^2 \end{Bmatrix}, \quad (2.28)$$

$$q(t) = \Psi^1(t)q^1 + \Psi^2(t)q^2 = [\Psi^1(t) \ \Psi^2(t)] \begin{Bmatrix} q^1 \\ q^2 \end{Bmatrix}, \quad (2.29)$$

where u^i, q^i , ($i = 1, 2$) are the nodal values.

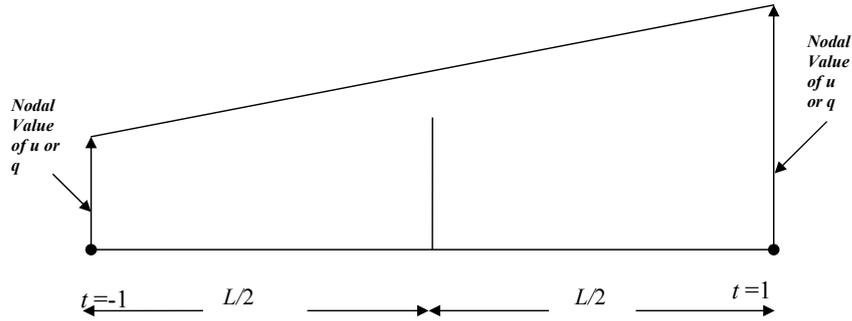


Figure 2.6: Linear element definitions

By using representations such as (2.27), (2.28), (2.29), we can write equation (2.18) as an approximation given by:

$$c(y)u(y) = \sum_{k=1}^N \left[\int_{-1}^1 [\Psi^1(t) \ \Psi^2(t)] \begin{Bmatrix} u^1 \\ u^2 \end{Bmatrix} \frac{\partial F(t, y)}{\partial n(x)} \frac{L_k}{2} dt - \int_{-1}^1 F(t, y) [\Psi^1(t) \ \Psi^2(t)] \begin{Bmatrix} q^1 \\ q^2 \end{Bmatrix} \frac{L_k}{2} dt \right]. \quad (2.30)$$

We can write the first and second integrals in equation (2.30) in a compact form as,

$$\int_{-1}^1 F(t, y) [\Psi^1(t) \ \Psi^2(t)] \begin{Bmatrix} q^1 \\ q^2 \end{Bmatrix} \frac{L_k}{2} dt = [g_1 \ g_2] \begin{Bmatrix} q^1 \\ q^2 \end{Bmatrix} \quad (2.31)$$

and

$$\int_{-1}^1 [\Psi^1(t) \ \Psi^2(t)] \begin{Bmatrix} u^1 \\ u^2 \end{Bmatrix} \frac{\partial F(t, y)}{\partial n(x)} \frac{L_k}{2} dt = [h_1 \ h_2] \begin{Bmatrix} u^1 \\ u^2 \end{Bmatrix}, \quad (2.32)$$

where:

$$g_1 = \int_{-1}^1 F(t, y) \Psi^1(t) \frac{L_k}{2} dt, \text{ and } g_2 = \int_{-1}^1 F(t, y) \Psi^2(t) \frac{L_k}{2} dt,$$

$$h_1 = \int_{-1}^1 \Psi^1(t) \frac{\partial F(t, y)}{\partial n(x)} \frac{L_k}{2} dt, \text{ and } h_2 = \int_{-1}^1 \Psi^2(t) \frac{\partial F(t, y)}{\partial n(x)} \frac{L_k}{2} dt.$$

2.4 Numerical solutions using Matlab

It is clear from the previous discussion that a method such as the BEM would best be implemented with a tool that can perform matrix computations and numerical integration efficiently and produce graphical output. Matlab is one such tool that can perform all these tasks with a simple code.

For simplicity and convenience, we shall solve the boundary integral equations using constant elements.

It is very useful to divide the Matlab program into three stages [9]:

First stage:

During this stage, the program creates a uniform discretisation on the boundary of the domain. It also computes the coordinates of all the midpoints and the lengths of each boundary element, as well as the unit normal vector to each element. This information will be needed for the next stage.

Second stage:

During this stage, the matrix \mathbf{A} and column vector \mathbf{b} are constructed, so that the system for \mathbf{z} can be solved to find the approximate solutions to the unknown boundary values.

There are two integrals to be computed. These are,

$$\int_{\Gamma_k} F(x, y) ds(x) \text{ and } \int_{\Gamma_k} \frac{\partial F(x, y)}{\partial n(x)} ds(x).$$

When $k = m$, the integrals can be evaluated analytically to give, see e.g [2, 9],

$$\int_{\Gamma_k} F(x, y) ds(x) = \frac{L_k}{2\pi} \left[\ln\left(\frac{L_k}{2}\right) - 1 \right], \quad (2.33)$$

$$\int_{\Gamma_k} \frac{\partial F(x, y)}{\partial n(x)} ds(x) = 0. \quad (2.34)$$

When $k \neq m$, we can evaluate the integrals using numerical methods such as Gauss quadrature because the analytic calculation of the integrals is more complicated. Then, we use the Gauss elimination method to solve the final algebraic system of equations.

Third stage:

In this stage, we essentially use Eq.(2.19) to calculate the solutions at the interior points. We also calculate the exact solution and the difference between the exact and approximate solutions. We can draw and generate a surface plot for the exact and approximate solutions and the difference between them, also the relative error is calculated as

$$r(J) = \frac{\max_{1 \leq j \leq J} |u_{approx}(x^j) - u_{exact}(x^j)|}{\max_{1 \leq j \leq J} |u_{exact}(x^j)|}, \quad (2.35)$$

where u_{approx} , u_{exact} are the numerical and exact solutions, respectively.

2.5 Examples

In this section, we shall examine some test examples to assess the performance of the BEM Matlab program. In order to verify the convergence of the method, we applied the BEM to some test problems with different domains as discussed below. The surface plots of the numerical solutions were obtained with the most refined mesh in each example. The graph of relative error has the number of nodes on the horizontal axis and the relative error on the vertical axis.

2.5.1 Example 1:

Let us solve the following problem, see Figure 2.7:

$$\nabla^2 u = 0, \quad \text{for } 0 < x_1 < 1, \quad 0 < x_2 < 1,$$

with the boundary conditions:

$$u=0 \quad \text{when } x_1=0, \quad \text{for } 0 < x_2 < 1,$$

$$u=\cos(\pi x_2) \quad \text{when } x_1=1, \quad \text{for } 0 < x_2 < 1,$$

$$q=\frac{\partial u}{\partial n} = 0, \quad \text{when } x_2=0 \quad \text{and} \quad x_2=1, \quad \text{for } 0 < x_1 < 1.$$

The exact solution for this problem is

$$u_{exact}(x_1, x_2) = \frac{\sinh(\pi x_1) \cos(\pi x_2)}{\sinh(\pi)}.$$

A surface plot of the exact and approximate solutions and the difference between them is shown in Figure 2.8.

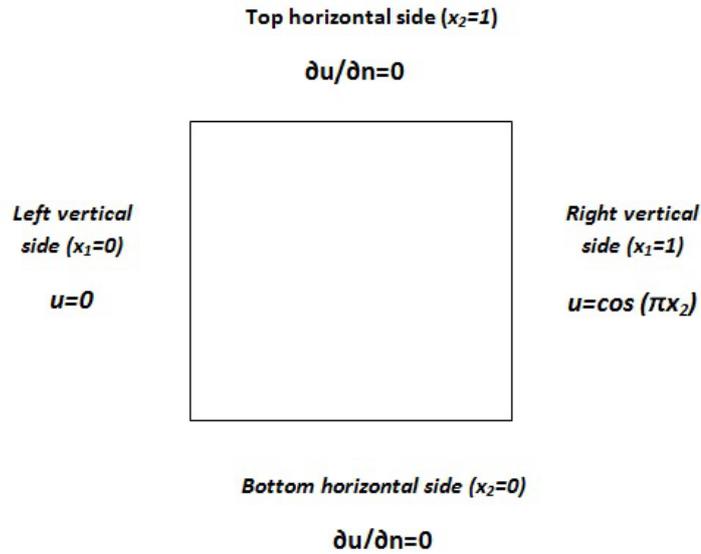


Figure 2.7: Domain plot for Example 1

Table 2.1: Computed temperatures along line $x_2 = 0.8$

x_1	32 elements	64 elements	128 elements	256 elements	Exact
0.10	-0.022383	-0.022375	-0.022373	-0.022372	-0.022371
0.20	-0.047005	-0.046981	-0.046973	-0.046970	-0.046969
0.30	-0.076310	-0.076263	-0.076248	-0.076243	-0.076241
0.40	-0.113207	-0.113134	-0.113110	-0.113103	-0.113099
0.50	-0.161361	-0.161261	-0.161228	-0.161217	-0.161212
0.60	-0.225558	-0.225431	-0.225388	-0.225373	-0.225367
0.70	-0.312171	-0.312024	-0.311973	-0.311956	-0.311948
0.80	-0.429788	-0.429648	-0.429597	-0.429579	-0.429571
0.90	-0.590047	-0.589986	-0.589958	-0.589947	-0.589941

Table 2.1 lists the computed values of $u(x)$ along the line $x_2 = 0.8$. The values at interior points are obtained by using 32 to 256 boundary elements, compared with the exact solution. The improvement in the accuracy of the numerical results can be seen clearly when the number of boundary elements is increased from 32 to 256.

Number of elements	Relative error
32	0.0437
64	0.0145
128	0.0030
256	0.0007

Table 2.2: Convergence of the approximate solution of Example 1

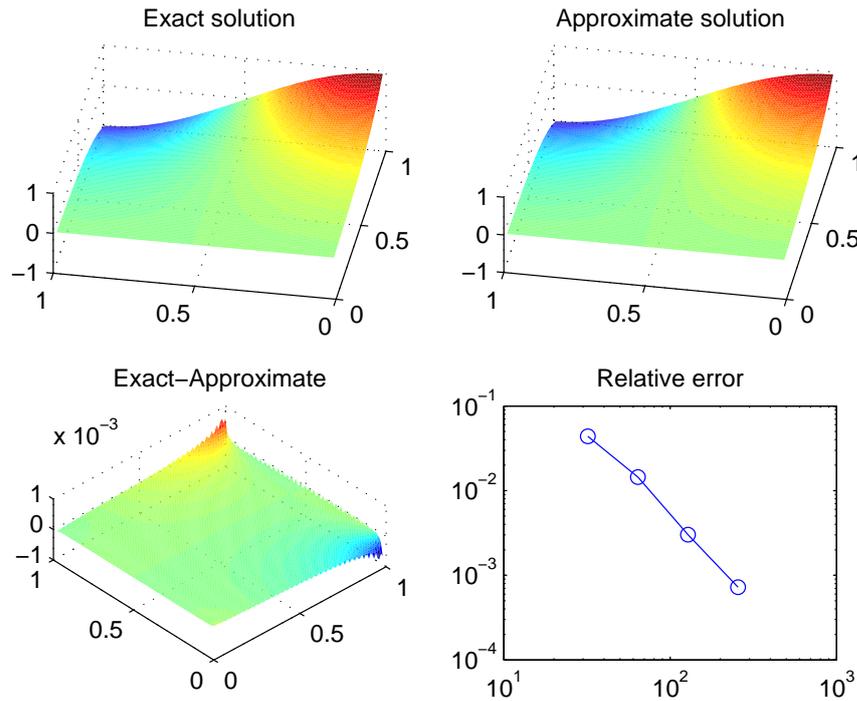


Figure 2.8: Surface plot of solution for Example 1

Table 2.2 provides the relative error as the number of boundary elements is increased. It can be clearly seen that the relative error is reduced by increasing the number of boundary elements, demonstrating the convergence of the solution.

2.5.2 Example 2:

Let us solve the following problem, see Figure 2.9:

$$\nabla^2 u = 0 \text{ for } x_1 > 0, x_2 > 0 \text{ and } x_1^2 + x_2^2 < 1$$

with the boundary conditions:

$$u = x_2 \text{ when } x_1 = 0, \text{ for } 0 < x_2 < 1,$$

$$u = x_1 + x_2 \text{ when for } x_1^2 + x_2^2 = 1,$$

$q = \frac{\partial u}{\partial n} = -1$, when $x_2 = 0$, for $0 < x_1 < 1$.

The exact solution for this problem is $u_{exact}(x_1, x_2) = x_1 + x_2$.

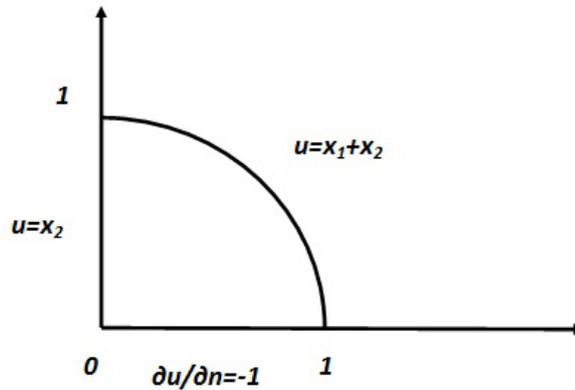


Figure 2.9: Domain plot for Example 2

Let us discretise each straight side of the boundary into N_0 elements and the arc on $(x_1^2 + x_2^2 = 1)$ into $2N_0$ elements, so the total number of elements for the domain is $N = 4N_0$.

By applying the Matlab program (Appendix A), a surface plot of the exact and approximate solutions and the difference between them is obtained, see Figure 2.10. Table 2.3 provides the relative error for an increasing number of the boundary elements.

Number of elements	Relative error
64	0.0067
128	0.0019
256	0.0006

Table 2.3: Convergence of the approximate solution of Example 2

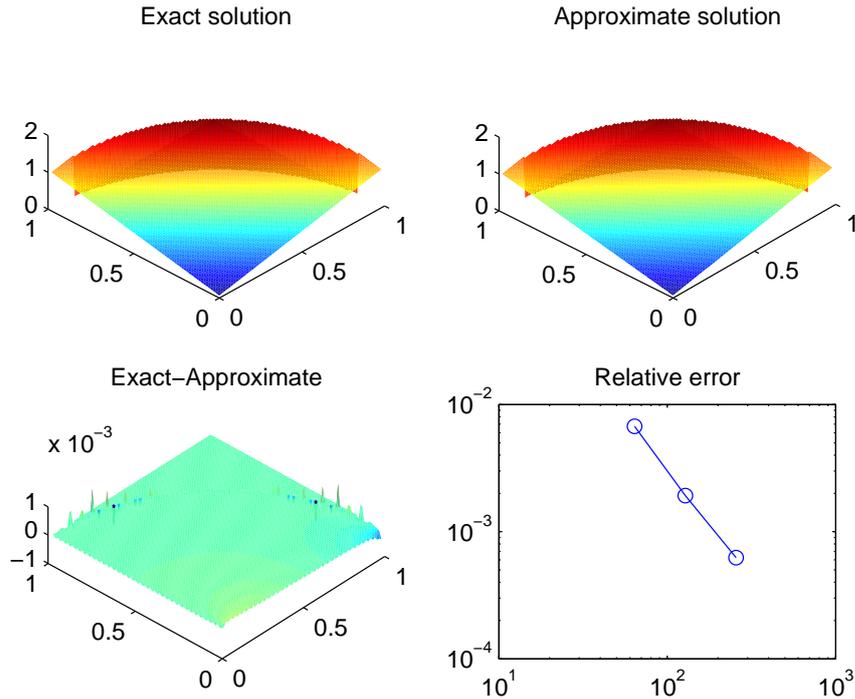


Figure 2.10: Surface plot of solution for Example 2

2.5.3 Example 3:

Let us solve the following problem, see Figure 2.11:

$$\nabla^2 u = 0 \text{ for } 1 < x_1 < 2, 1 < x_2 < 2 \text{ and } 1 < x_1^2 + x_2^2 < 4.$$

The domain is bounded by the circles $x_1^2 + x_2^2 < 1$ and $x_1^2 + x_2^2 < 4$ and the boundary conditions are :

$$\frac{\partial u}{\partial n} = 0 \text{ on the straight side } x_1 = 0, 1 < x_2 < 2,$$

$$\frac{\partial u}{\partial n} = 0 \text{ on the straight side } x_2 = 0, 1 < x_1 < 2,$$

$$u = \cos(4\arctan(\frac{x_2}{x_1})) \text{ on the arc } x_1^2 + x_2^2 = 1, x_1 > 0, x_2 > 0,$$

$$u = 3 \cos(4\arctan(\frac{x_2}{x_1})) \text{ on the arc } x_1^2 + x_2^2 = 4, x_1 > 0, x_2 > 0.$$

The exact solution for this problem is

$$u_{exact}(x_1, x_2) = \left[\frac{16}{85} \left([x_1^2 + x_2^2]^2 - \frac{1}{[x_1^2 + x_2^2]^2} \right) - \frac{16}{255} \left(\frac{[x_1^2 + x_2^2]^2}{16} - \frac{16}{[x_1^2 + x_2^2]^2} \right) \right] \cos(4\arctan(\frac{x_2}{x_1})).$$

Let us discretise each of the straight sides of the boundary into N_0 elements and the arcs on $(x_1^2 + x_2^2 = 1)$ and $(x_1^2 + x_2^2 = 4)$ into $2N_0$ and $4N_0$ elements, respectively, so the total number of elements for the domain is $N = 8N_0$.

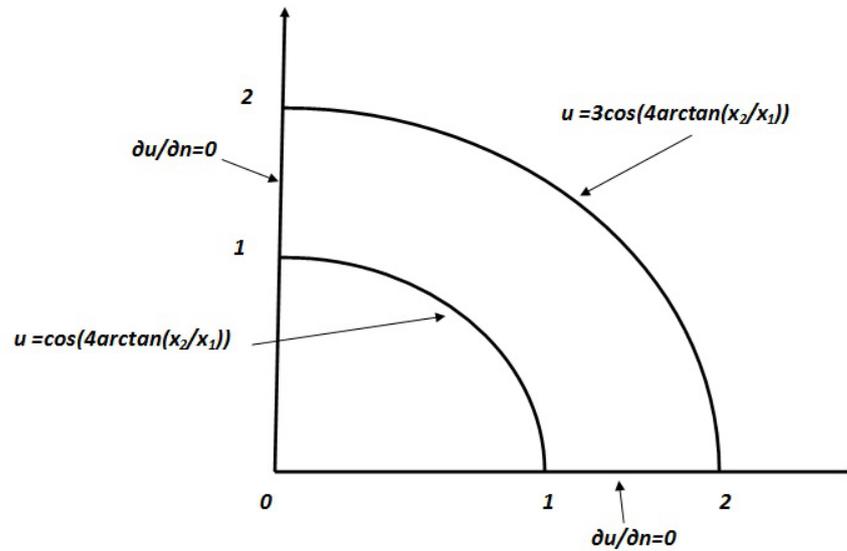


Figure 2.11: Domain plot for Example 3

By applying the Matlab program (Appendix A), a surface plot of the exact and approximate solutions and the difference between them is obtained, see Figure 2.12. Table 2.4 shows the values of the relative error with different BEM discretisations.

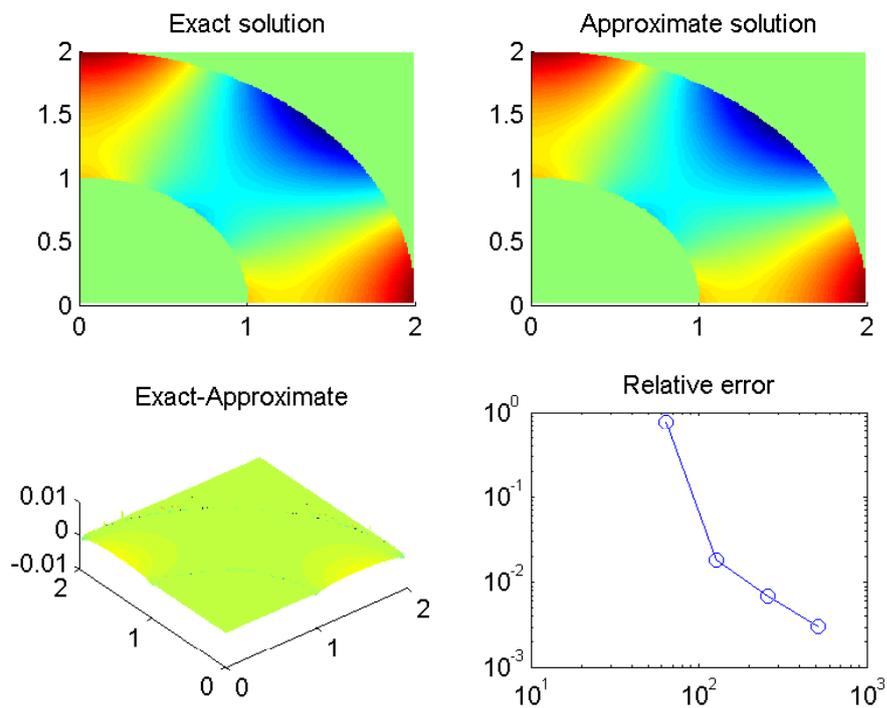


Figure 2.12: Surface plot of solution for Example 3

Number of elements	Relative error
64	0.7638
128	0.0182
256	0.0068
512	0.0031

Table 2.4: Convergence of the approximate solution of Example 3

2.6 Concluding remarks

In this chapter, the BIE formulation is presented for solving the two-dimensional interior boundary-value problem defined by Eq.(2.1) with mixed boundary conditions. The BEM is then applied to solve the corresponding BIE. The boundary is discretised by straight line elements, and the function u and its normal derivative q are approximated as constants over each boundary element.

The following remarks apply to the present chapter:

- The function u and its normal derivative q are approximated as constants over each boundary element, therefore the boundary integrals can be easily evaluated and the discontinuity of the normal derivative at corner points is avoided;
- The main features which render the BEM advantageous with respect to the finite element method (FEM) and the finite difference method (FDM) are, therefore, the reduction of the problem dimensions by one and the fact that no discretisation of the computational domain is required;
- Convergence studies with mesh refinement show that the BEM possesses acceptable rates of convergence for the problems studied.

Chapter 3

Heat conduction problems with variable coefficients

3.1 Introduction

As discussed in the previous chapters, the main requirement for the reduction of the PDE to a BIE is that a fundamental solution to the PDE must be available. Such fundamental solutions are well-known for many PDEs with constant coefficients, see [1–7], but are not generally available when the coefficients of the original PDE are variable.

The coefficients in the mathematical model of a physical problem typically correspond to the material parameters of the problem. In heterogeneous media the material parameters may vary with position and/or time. For this reason, there is a demand on the development of accurate and efficient numerical methods able to deal with the spatial variations of material coefficients [7, 11, 12].

Finding the fundamental solution for PDEs with variable coefficients has generally been limited to specific forms of the coefficients. Clements [13] derived a BEM formulation for a 2D Laplace equation with variable conductivity in one direction. Shaw [14] and Shaw et al. [15] developed a BIE formulation for a Poisson equation with a linearly layered conductivity, using 2D and 3D Green's functions for this heterogeneous medium problems, respectively. Ang et al. [16] have extended the method of Clements [13] for more complicated variable coefficients, which can be written in separate forms but only for such special case. The main drawbacks of the analytic methods to find the fundamental solution presented in [13–16] are first, that they only work for special cases of variable coefficients.

Second, even for such simple cases, the mathematical procedures are very complicated [12]. Several BEM techniques have been proposed to treat problems with variable coefficients, e.g. by cell discretisation [11] or the dual reciprocity method (DRM) [12]. In the present chapter, a parametrix (Levi function) is adopted, which is usually available [8, 17, 18]. This allows a reduction of the mathematical problem to a boundary-domain integral or integro-differential equation (BDIE or BDIDE) [8, 19]. A BDIE and a BDIDE formulations to solve problems with variable coefficients are presented in [8] using specially constructed localised parametrices to reduce a BVP with variable coefficients to a localised boundary-domain integral or integro-differential equation (LBDIE or LBDIDE). The use of specially constructed localised parametrices leads to sparsely populated systems of linear algebraic equations. An implementation of the LBDIE method for the numerical solution of a second-order linear elliptic PDE with variable coefficients is presented in [19], although the formulation is restricted to Neumann boundary-value problems.

Boundary-domain integral equation (BDIE) methods have also been developed by Skerget et al. [20–22] for the solution of non-linear fluid mechanics problems described by the Navier-Stokes equations. Skerget's formulation treats all the nonlinear terms as body forces, which are included in the boundary integral equations as a domain integral evaluated by discretising the body into cells. Another related formulation developed by Popov and Power [23], named the dual reciprocity-multi-domain (DRM-MD) approach, combines the DRM with domain decomposition, leading to substantial improvements in the accuracy and convergence of the DRM formulation for complex problems. The DRM-MD formulation has been applied to the solution of the Navier-Stokes equations [24] and to flow and solute transport in fractured porous media [25].

A further BDIE technique is the Analog Equation Method (AEM) of Katsikadelis [26], which has been applied to the solution of several elasticity problems, mostly related to plate bending. Katsikadelis and Nerantzaki [27] extended the AEM to a boundary-only method which decomposes the solution into a homogeneous part and a particular solution of the non-homogeneous one, and then obtained the particular solution via a radial basis function expansion of the domain term.

In the present chapter, the BDIE and BDIDE formulations proposed by Mikhailov [8, 19] are extended to the treatment of mixed BVPs. The numerical algorithms developed here do not use the concept of localisation as in [8, 19], but rather use global mesh-based discretisations. The chapter also discusses different techniques to deal with the discontinuity of

the flux at corners, by testing different positions of the collocation points. The implementation of the BDIE and BDIDE formulations in the present chapter for stationary heat transfer in isotropic materials with variable coefficients associated with Dirichlet, Neumann and mixed boundary conditions has been published in [28–31]. Numerical solutions of several test examples are included to validate the methods.

3.2 Reduction of the BVP to a BDIE/BDIDE

Let us consider the following stationary heat transfer BVP in an isotropic inhomogeneous medium for a two-dimensional bounded body Ω , with prescribed temperature $\bar{u}(x)$ on part $\partial_D\Omega$ of the boundary $\partial\Omega$ and prescribed heat flux $\bar{t}(x)$ on the remaining $\partial_N\Omega$ part of $\partial\Omega$, i.e. we consider the second-order linear elliptic PDE [8, 19, 28–31],

$$(Lu)(x) := \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[a(x) \frac{\partial u(x)}{\partial x_i} \right] = f(x), \quad x \in \Omega. \quad (3.1)$$

Let us consider three types of problems:

• **Dirichlet problem:**

$$u(x) = \bar{u}(x), \quad x \in \partial\Omega. \quad (3.2)$$

•• **Neumann problem:**

$$[Tu](x) = a(x) \frac{\partial u(x)}{\partial n(x)} = \bar{t}(x), \quad x \in \partial\Omega. \quad (3.3)$$

••• **Mixed problem:**

$$u(x) = \bar{u}(x), \quad x \in \partial_D\Omega, \quad (3.4)$$

$$Tu(x) = \bar{t}(x), \quad x \in \partial_N\Omega, \quad (3.5)$$

where $\bar{u}(x)$ and $\bar{t}(x)$ are known functions on the Dirichlet part $\partial_D\Omega$ and the Neumann part $\partial_N\Omega$, respectively.

In equations (3.1) to (3.5), Ω is a bounded domain, $u(x)$ the temperature, $a(x)$ a known variable thermal conductivity coefficient, $f(x)$ a known heat source, T a surface flux operator, $[Tu](x) := a(x) \frac{\partial u}{\partial n}(x)$, $n(x)$ the external normal vector to the boundary $\partial\Omega$, and

$\bar{u}(x)$ and $\bar{t}(x)$ are known functions. The BVP (3.1-3.5) appears when modelling stationary heat transfer, elastostatics, electrostatics, and diffusion problems for functionally graded materials, as well as in flow in porous media.

The Green formula for the differential operator L has the form

$$\int_{\Omega} [uLv - vLu]d\Omega = \int_{\partial\Omega} [uTv - vTu]d\Gamma, \quad (3.6)$$

where u and v are arbitrary twice differentiable functions.

Let L be a linear operator and $F(x, y)$ be its fundamental solution, i.e.

$$L_x F(x, y) = \delta(x - y),$$

where δ is the Dirac delta function. Then one could take $v(x) = F(x, y)$, identify $u(x)$ with a solution of Eq.(3.1), and thus arrive at the third Green identity

$$c(y)u(y) - \int_{\partial\Omega} [u(x)T_x F(x, y) - F(x, y)Tu(x)]d\Gamma(x) = \int_{\Omega} F(x, y)f(x)d\Omega(x), \quad (3.7)$$

where $c(y)$ is given by Eq.(2.18). Substituting the boundary condition in the Green identity Eq.(3.7) and applying it for $y \in \partial\Omega$, we arrive at a direct BIE [1–6].

For partial differential operators with variable coefficients, like L in Eq.(3.1), a fundamental solution is generally not available in explicit form. However, a parametrix is often available, which is a function $P(x, y)$ satisfying the equation [8, 19, 28–31],

$$L_x P(x, y) = \delta(x - y) + R(x, y). \quad (3.8)$$

The fundamental solution of the operator with frozen coefficients $a(x) = a(y)$ corresponding to the operator L defined in Eq.(3.1), can be used as a parametrix, in the two-dimensional case [8, 19, 28–31],

$$P(x, y) = \frac{1}{2\pi a(y)} \ln |x - y|. \quad (3.9)$$

Substituting Eq.(3.9) in Eq.(3.8), we obtain,

$$\sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[a(x) \frac{\partial}{\partial x_i} \left[\frac{1}{2\pi a(y)} \ln |x - y| \right] \right] = \delta(x - y) + R(x, y).$$

By applying the product rule for differentiation, we get:

$$\sum_{i=1}^2 \left[\frac{1}{a(y)} \cdot \frac{\partial a(x)}{\partial x_i} \cdot \frac{\partial}{\partial x_i} \left[\frac{1}{2\pi} \ln|x-y| \right] + \frac{a(x)}{a(y)} \cdot \frac{\partial^2}{\partial x_i^2} \left[\frac{1}{2\pi} \ln|x-y| \right] \right] = \delta(x-y) + R(x,y).$$

Now, since $a(x) = a(y)$ and $\sum_{i=1}^2 \frac{\partial^2}{\partial x_i^2} \left[\frac{1}{2\pi} \ln|x-y| \right] = \delta(x-y)$, we have

$$\delta(x-y) = \left[\frac{a(x)}{a(y)} \right] \cdot \sum_{i=1}^2 \frac{\partial^2}{\partial x_i^2} \left[\frac{1}{2\pi} \ln|x-y| \right]$$

and

$$R(x,y) = \sum_{i=1}^2 \frac{1}{a(y)} \cdot \frac{\partial a(x)}{\partial x_i} \cdot \frac{\partial}{\partial x_i} \left[\frac{1}{2\pi} \ln|x-y| \right].$$

since

$$\frac{\partial}{\partial x_i} \left[\frac{1}{2\pi} \ln|x-y| \right] = \frac{1}{2\pi r} \frac{\partial r}{\partial x_i} = \frac{x_i - y_i}{2\pi r^2}, \quad r = |x-y|,$$

The remainder $R(x,y)$ will then be,

$$R(x,y) = \sum_{i=1}^2 \frac{x_i - y_i}{2\pi a(y) |x-y|^2} \frac{\partial a(x)}{\partial x_i}, \quad x, y \in \mathbb{R}^2, \quad (3.10)$$

which has only a weak singularity at $x = y$.

Substituting $P(x,y)$ for $v(x)$ in Eq.(3.6) and taking $u(x)$ as a solution to Eq.(3.1), we obtain the integral equation,

$$\begin{aligned} c(y)u(y) - \int_{\partial\Omega} [u(x)T_x P(x,y) - P(x,y)T u(x)] d\Gamma(x) + \\ + \int_{\Omega} R(x,y)u(x) d\Omega(x) = \int_{\Omega} P(x,y)f(x) d\Omega(x). \end{aligned} \quad (3.11)$$

Identity (3.11) can be used for formulating either a BDIE or a BDIDE, with respect to u and its derivatives. Let us consider the two forms below.

3.3 Boundary-domain integral/integro-differential equations (BDIE/BDIDE)

3.3.1 Dirichlet problem

System of boundary-domain integral equations (BDIEs)

Substituting the boundary condition (3.2) into (3.11), introducing a new variable $t(x) = Tu(x)$ for the unknown $Tu(x)$ on $\partial\Omega$, we can reduce the BVP (3.1)-(3.2) to the following system of BDIEs for $u(x)$ at $x \in \Omega$, and $t(x)$ at $x \in \partial\Omega$,

$$\begin{aligned} \int_{\partial\Omega} P(x, y)t(x)d\Gamma(x) + \int_{\Omega} R(x, y)u(x)d\Omega(x) &= -c(y)\bar{u}(y) + \\ + \int_{\partial\Omega} \bar{u}(x)T_xP(x, y)d\Gamma(x) + \int_{\Omega} P(x, y)f(x)d\Omega(x), \quad y \in \partial\Omega, \end{aligned} \quad (3.12)$$

$$\begin{aligned} u(y) + \int_{\partial\Omega} P(x, y)t(x)d\Gamma(x) + \int_{\Omega} R(x, y)u(x)d\Omega(x) &= \\ = \int_{\partial\Omega} \bar{u}(x)T_xP(x, y)d\Gamma(x) + \int_{\Omega} P(x, y)f(x)d\Omega(x), \quad y \in \Omega. \end{aligned} \quad (3.13)$$

Boundary-domain integro-differential equations (BDIDEs)

Using another approach, we can substitute the boundary condition (3.2) into (3.11) but leave T as a differential operator acting on u on the boundary. We then arrive at the following BDIDE for $u(x)$ at $x \in \bar{\Omega}$ with given boundary values of u ,

$$u(y) = \bar{u}(y), \quad y \in \partial\Omega, \quad (3.14)$$

$$\begin{aligned} u(y) + \int_{\partial\Omega} P(x, y)Tu(x)d\Gamma(x) + \int_{\Omega} R(x, y)u(x)d\Omega(x) &= \\ = \int_{\partial\Omega} \bar{u}(x)T_xP(x, y)d\Gamma(x) + \int_{\Omega} P(x, y)f(x)d\Omega(x), \quad y \in \Omega. \end{aligned} \quad (3.15)$$

As the last term in the left hand side of Eqs.(3.12), (3.13) and (3.15) includes the unknown values of u over the whole domain Ω , this BDIE does not lead to a BIE as in the case when the parametrix is a fundamental solution. The two systems of equations (3.12)-(3.13) and

(3.14)-(3.15) will lead, after discretisation, to fully populated systems of linear algebraic equations.

3.3.2 Neumann problem

To ensure a solution exists for the BVP (3.1), (3.3), we assume that the functions $\bar{t}(x)$ and $f(x)$ satisfy the compatibility condition [10, 19, 32],

$$\int_{\partial\Omega} \bar{t}(x) d\Gamma(x) - \int_{\Omega} f(x) d\Omega(x) = 0.$$

Substituting the boundary condition (3.3) into (3.11), but leaving $u(x)$ for the unknown on $\partial\Omega$, we can reduce the BVP (3.1), (3.3) to the following system of BDIEs for $u(x)$ at $x \in \Omega \cup \partial\Omega$,

$$\begin{aligned} c(y)u(y) - \int_{\partial\Omega} u(x)T_x P(x, y) d\Gamma(x) + \int_{\Omega} R(x, y)u(x) d\Omega(x) + \\ = \int_{\Omega} P(x, y)f(x) d\Omega(x) - \int_{\partial\Omega} P(x, y)\bar{t}(x) d\Gamma(x), \quad y \in \partial\Omega, \end{aligned} \quad (3.16)$$

$$\begin{aligned} u(y) - \int_{\partial\Omega} u(x)T_x P(x, y) d\Gamma(x) + \int_{\Omega} R(x, y)u(x) d\Omega(x) \\ = \int_{\Omega} P(x, y)f(x) d\Omega(x) - \int_{\partial\Omega} P(x, y)\bar{t}(x) d\Gamma(x), \quad y \in \Omega. \end{aligned} \quad (3.17)$$

The Neumann BVP (3.1), (3.3) is not unconditionally and uniquely solvable, see e.g. [10, 19, 32–34], and neither is the integral equation. Therefore, the resulting system of linear algebraic equations will be either singular or ill-conditioned. The problem can be solved by the approach described in [10, 19, 32, 33]. The BVP (3.1), (3.3) is only unique up to an additive constant. So, if u is a solution of (3.1), (3.3) then, for any $c \in \mathbb{R}$, $\tilde{u}(x) = u(x) + c$ for $x \in \mathbb{R}$ is also a solution of Neumann BVP(3.1), (3.3). In order to fix the constant $c \in \mathbb{R}$ and make it uniquely determined, one can perturb the BDIE (3.16) by adding the operator

$$[\overset{o}{K} u](x) = \frac{1}{|\partial\Omega|} \int_{\partial\Omega} u(x) d\Gamma(x), \quad (3.18)$$

where $|\partial\Omega|$ denotes the length of the boundary $\partial\Omega$, to the left-hand side of the BDIE

(3.16), obtaining the equations:

$$c(y)u(y) - \int_{\partial\Omega} u(x)T_x P(x,y)d\Gamma(x) + \int_{\Omega} R(x,y)u(x)d\Omega(x) + \frac{1}{|\partial\Omega|} \int_{\partial\Omega} u(x)d\Gamma(x) = \int_{\Omega} P(x,y)f(x)d\Omega(x) - \int_{\partial\Omega} P(x,y)\bar{t}(x)d\Gamma(x), \quad y \in \partial\Omega, \quad (3.19)$$

$$\begin{aligned} & u(y) - \int_{\partial\Omega} u(x)T_x P(x,y)d\Gamma(x) + \int_{\Omega} R(x,y)u(x)d\Omega(x) \\ &= \int_{\Omega} P(x,y)f(x)d\Omega(x) - \int_{\partial\Omega} P(x,y)\bar{t}(x)d\Gamma(x), \quad y \in \Omega. \end{aligned} \quad (3.20)$$

Using the operator in Eq.(3.18), one can prove that the BVP (3.1), (3.3) has a unique solution up to an additive constant.

Let $\tilde{u}_1(x) = u(x)$, then

$$\frac{1}{|\partial\Omega|} \int_{\partial\Omega} \tilde{u}_1(x)d\Gamma(x) = a, \quad a \in \mathbb{R}. \quad (3.21)$$

Let us define a new function $\tilde{u}_2(x) = \tilde{u}_1(x) + c$, which is also a solution of the Neumann problem,

$$\frac{1}{|\partial\Omega|} \int_{\partial\Omega} \tilde{u}_2(x)d\Gamma(x) = b, \quad b \in \mathbb{R}. \quad (3.22)$$

Moreover, let $u_1(x) = \tilde{u}_1(x) - a$, and hence,

$$\int_{\partial\Omega} u_1(x)d\Gamma(x) = \int_{\partial\Omega} [\tilde{u}_1(x) - a]d\Gamma(x).$$

Using Eq.(3.21) and the fact that $\int_{\partial\Omega} d\Gamma(x) = |\partial\Omega|$, we get,

$$\int_{\partial\Omega} \left[\tilde{u}_1(x) - \frac{1}{|\partial\Omega|} \int_{\partial\Omega} \tilde{u}_1(x)d\Gamma(x) \right] d\Gamma(x) = 0.$$

Therefore,

$$\int_{\partial\Omega} u_1(x)d\Gamma(x) = 0.$$

Let us now define $u_2(x) = \tilde{u}_2(x) - b$, following the same steps as before, we can prove that

$$\int_{\partial\Omega} u_2(x) d\Gamma(x) = 0.$$

Now, our goal is to prove that $u_1(x) = u_2(x)$. Using Eq.(3.22), we have

$$b = \frac{1}{|\partial\Omega|} \int_{\partial\Omega} \tilde{u}_2(x) d\Gamma(x).$$

Since $\tilde{u}_2(x) = \tilde{u}_1(x) + c$ and $\int_{\partial\Omega} d\Gamma(x) = |\partial\Omega|$, we get,

$$b = \frac{1}{|\partial\Omega|} \int_{\partial\Omega} [\tilde{u}_1(x) + c] d\Gamma(x) = a + c.$$

Therefore, $b = a + c$. On the other hand, $u_2(x) = \tilde{u}_2(x) - b = (\tilde{u}_1(x) + c) - (a + c) = \tilde{u}_1(x) - a = u_1(x)$. \square

3.3.3 Mixed problem

Boundary-domain integral equation (BDIE)

Substituting the boundary conditions (3.4) and (3.5) into (3.11), introducing a new variable $t(x)=Tu(x)$ for the unknown flux on $\partial_D\Omega$ and using Eq.(3.11) at $y \in \Omega \cup \partial\Omega$ reduces the BVP (3.1) with (3.4)-(3.5) to the following BDIE for $u(x)$ at $x \in \Omega \cup \partial_N\Omega$ and $t(x)$ at $x \in \partial_D\Omega$,

$$\begin{aligned} c^0(y)u(y) - \int_{\partial_N\Omega} u(x)T_xP(x,y)d\Gamma(x) + \int_{\partial_D\Omega} P(x,y)t(x)d\Gamma(x) + \\ + \int_{\Omega} R(x,y)u(x)d\Omega(x) = \Psi^0(y), \quad y \in \Omega \cup \partial\Omega, \end{aligned} \quad (3.23)$$

where

$$\Psi^0(y) := [c^0(y) - c(y)]\bar{u}(y) + \Psi(y), \quad (3.24)$$

$$\Psi(y) := \int_{\partial_D\Omega} \bar{u}(x)T_xP(x,y)d\Gamma(x) - \int_{\partial_N\Omega} P(x,y)\bar{t}(x)d\Gamma(x) + \int_{\Omega} P(x,y)f(x)d\Omega(x) \quad (3.25)$$

and

$$c^0(y) = \begin{cases} 0 & \text{if } y \in \partial_D\Omega \\ c(y) & \text{if } y \in \Omega \cup \partial_N\Omega \end{cases} \quad (3.26)$$

Boundary-domain integro-differential equation (BDIDE)

Using another approach, we can substitute the boundary conditions (3.4) and (3.5) into (3.11) but leave T as a differential flux operator acting on u on the Dirichlet boundary $\partial_D\Omega$ and use the following BDIDE

$$\begin{aligned} c(y)u(y) - \int_{\partial_N\Omega} u(x)T_xP(x,y)d\Gamma(x) + \int_{\partial_D\Omega} P(x,y)Tu(x)d\Gamma(x) + \\ + \int_{\Omega} R(x,y)u(x)d\Omega(x) = \Psi(y), \quad y \in \Omega \cup \partial_N\Omega. \end{aligned} \quad (3.27)$$

The two equations Eq.(3.23) and Eq.(3.27) will lead, after discretisation, to fully populated systems of linear algebraic equations.

3.4 Discretisation of the BDIE/BDIDE

Let us discretise the domain Ω into a mesh of triangular elements T_k , $k = 1, 2, \dots, N$, $T_h \cap T_m = \emptyset$, $h \neq m$. Let J be the total number of nodes x^i , $i = 1, \dots, J$, at the vertices of triangles, from which there are J_D nodes on $\partial\Omega$, see Figure 3.1.

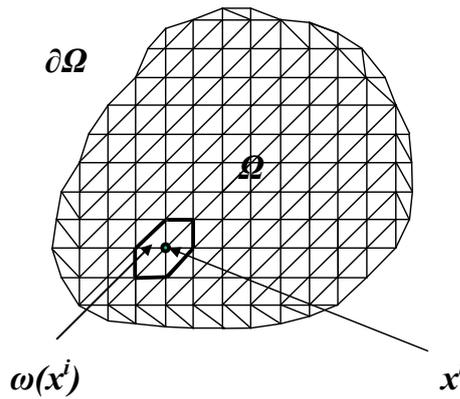


Figure 3.1: Discretization of the domain Ω

In the present chapter equations (3.12)-(3.13), (3.14)-(3.15), (3.21)-(3.22), (3.23) and (3.27) will be solved. To obtain a system of linear algebraic equations by the collocation method, we collocate at the nodes x^i , $i = 1, \dots, J$, and substitute an interpolation of

$u(x)$ of the form

$$u(x) \approx \sum_{\bar{\omega}_j \ni x} u(x^j) \Phi_j(x), \quad \Phi_j(x) = \begin{cases} \phi_{kj}(x) & \text{if } x, x^j \in \bar{T}_k \\ 0 & \text{otherwise,} \end{cases} \quad (3.28)$$

where $\bar{\omega}_j$ is the support of $\Phi_j(x)$, which consists of all triangular elements that have x^j as a vertex; $\phi_{kj}(x)$ are the shape functions localized on an element T_k , and associated with the node x^j . For the triangular elements, $\phi_{kj}(x)$ can be chosen as linear functions. We can also use an interpolation of $t(x) = (Tu)(x^j)$ along only boundary nodes

$$t(x) = \sum_{x^j \in \partial\Omega} t(x^j) v_j(x), \quad x \in \partial\Omega. \quad (3.29)$$

Here, $v_j(x)$ are boundary shape functions, taken now as constant. Therefore, $v_j(x)$ will be equal 1 at $x_j \in \Gamma_j$ and 0 elsewhere, and Γ_j are boundary elements.

3.4.1 Dirichlet problem

System of boundary-domain integral equations (BDIEs)

Substituting the interpolations (3.28) and (3.29) in BDIE (3.12) and applying the collocation method, we arrive at the following system of J_D linear algebraic equations,

$$\sum_{x^j \in \Omega} K'_{ij} u(x^j) + \sum_{x^j \in \partial\Omega} Q_{ij} t(x^j) = -c(x^i) \bar{u}(x^i) - \sum_{x^j \in \partial\Omega} K'_{ij} \bar{u}(x^j) + \Psi D(x^i), \quad x^i \in \partial\Omega. \quad (3.30)$$

Discretising the BDIE (3.13), by substitution of interpolations (3.28) and (3.29), we arrive at the following system of $J - J_D$ linear algebraic equations,

$$u(x^i) + \sum_{x^j \in \Omega} K'_{ij} u(x^j) + \sum_{x^j \in \partial\Omega} Q_{ij} t(x^j) = - \sum_{x^j \in \partial\Omega} K'_{ij} \bar{u}(x^j) + \Psi D(x^i), \quad x^i \in \Omega. \quad (3.31)$$

Here,

$$K'_{ij} = \int_{\omega_j} \Phi_j(x) R(x, x^i) d\Omega(x), \quad (3.32)$$

$$\Psi D(x^i) = \int_{\partial\Omega} \bar{u}(x) T_x P(x, x^i) d\Gamma(x) + \int_{\Omega} f(x) P(x, x^i) d\Omega(x), \quad (3.33)$$

$$Q_{ij}(x^i) = \int_{\partial\Omega \cap \bar{\omega}_j} P(x, x^i) v_j(x) d\Gamma(x). \quad (3.34)$$

Boundary-domain integro-differential equations (BDIDEs)

From (3.14), we have J_D known values of u , since Dirichlet boundary conditions are given,

$$u(x^j) = \bar{u}(x^j), \quad x^j \in \partial\Omega. \quad (3.35)$$

Discretising (3.15) by substitution of interpolations (3.28) and applying the collocation method, we arrive at the following system of $J - J_D$ linear algebraic equations,

$$u(x^i) + \sum_{x^j \in \Omega} K'_{ij} u(x^j) + \sum_{x^j \in \Omega} Q'_{ij} u(x^j) = - \sum_{x^j \in \partial\Omega} K'_{ij} \bar{u}(x^j) - \sum_{x^j \in \partial\Omega} Q'_{ij} \bar{u}(x^j) + \Psi D(x^i), \quad x^i \in \Omega. \quad (3.36)$$

Here,

$$Q'_{ij} = \int_{\partial\Omega \cap \bar{\omega}_j} P(x, x^i) T \Phi_j(x) d\Gamma(x), \quad (3.37)$$

and K'_{ij} , $\Psi D(x^i)$ are given by Eqs. (3.32) and (3.33), respectively. Equation (3.36) is used in order to obtain the numerical values of $u(x)$ for the BVP (3.1)-(3.2) at interior points.

3.4.2 Neumann problem

System of boundary-domain integral equations (BDIEs)

To obtain a system of linear algebraic equations from the BDIEs (3.19) and (3.20) by the collocation method, we apply (3.19) at the nodes x^i , $i = 1, \dots, J$, and by substituting the interpolations (3.28), we arrive at the following system of J_D linear algebraic equations,

$$c(x^i)u(x^i) - \sum_{x^j \in \bar{\Omega}} Q_{ij} u(x^j) + \sum_{x^j \in \bar{\Omega}} K'_{ij} u(x^j) + \sum_{x^j \in \partial\Omega} \overset{\circ}{K}_j u(x^j) = \Psi N(x^i), \quad x^i \in \partial\Omega, \quad (3.38)$$

where $\overset{\circ}{K}_j := \frac{1}{|\partial\Omega|} \int_{\partial\Omega} \Phi_j(x) d\Gamma(x)$.

Discretising the BDIE (3.20), by substituting the interpolations (3.28) and applying the

collocation method, we arrive at the following system of $J - J_D$ linear algebraic equations:

$$u(x^i) - \sum_{x^j \in \bar{\Omega}} QN_{ij}u(x^j) + \sum_{x^j \in \bar{\Omega}} K'_{ij}u(x^j) = \Psi N(x^i), \quad x^i \in \Omega, \quad (3.39)$$

where

$$\Psi N(x^i) = \int_{\Omega} f(x)P(x, x^i)d\Omega(x) - \int_{\partial\Omega} P(x, x^i)\bar{t}(x)d\Gamma(x), \quad (3.40)$$

$$QN_{ij} = \int_{\partial\Omega \cap \bar{\omega}_j} \Phi_j(x)T_x P(x, x^i)d\Gamma(x). \quad (3.41)$$

3.4.3 Mixed problem

We can also use an interpolation of $t(x) = (Tu)(x^j)$ along the boundary nodes belonging to $\bar{\omega}(x^j) \cap \partial_D\Omega$

$$t(x) = \sum_{x^j \in \bar{\omega}(x^j) \cap \partial_D\Omega} t(x^j)v_j(x), \quad x \in \bar{\omega}(x^j) \cap \partial_D\Omega. \quad (3.42)$$

Here, $v_j(x)$ are boundary shape functions, taken now as constant. Therefore, $v_j(x)$ will be equal 1 at $x_j \in \bar{\omega}(x^j) \cap \partial_D\Omega$ and $v_j(x) = 0$ if $x^j \notin \bar{\omega}(x^j) \cap \partial_D\Omega$.

Boundary-domain integral equation (BDIE)

Substituting the interpolations (3.28) and (3.42) in BDIE (3.23) and applying the collocation method, we arrive at the following system of J linear algebraic equations for J unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N\Omega$ and $t(x^j) = (Tu)(x^j)$, $x^j \in \partial_D\Omega$,

$$\begin{aligned} c^0(x^i)u(x^i) + \sum_{x^j \in \Omega \cup \partial_N\Omega} KM_{ij}u(x^j) + \sum_{x^j \in \partial_D\Omega} QM'_{ij}t(x^j) &= \Psi^0(x^i) - \\ - \sum_{x^j \in \partial_D\Omega} KM_{ij}\bar{u}(x^j), \quad x^i \in \Omega \cup \partial\Omega, \quad i = 1, \dots, J, \text{ no sum in } i, \end{aligned} \quad (3.43)$$

where $\Psi^0(x^i) = [c^0(y) - c(y)]\bar{u}(y) + \Psi M(y)$, and

$$\begin{aligned} \Psi M(x^i) &= \int_{\bar{\omega}(x^i) \cap \partial_D\Omega} \bar{u}(x)T_x P(x, x^i)d\Gamma(x) - \int_{\bar{\omega}(x^i) \cap \partial_N\Omega} P(x, x^i)\bar{t}(x)d\Gamma(x) + \\ &+ \int_{\omega(x^i) \cap \Omega} f(x)P(x, x^i)d\Omega(x), \end{aligned} \quad (3.44)$$

$$KM_{ij} = \int_{\omega_j} \Phi_j(x)R(x, x^i)d\Omega(x) - \int_{\bar{\omega}(x^i) \cap \partial_N \Omega} \Phi_j(x)T_x P(x, x^i)d\Gamma(x), \quad (3.45)$$

$$QM'_{ij} = \int_{\bar{\omega}(x^i) \cap \partial_D \Omega} P(x, x^i)v_j(x)d\Gamma(x). \quad (3.46)$$

Boundary-domain integro-differential equation (BDIDE)

To obtain a system of linear algebraic equations from the BDIDE (3.27) by the collocation method, we collocate at the nodes x^i , $i = 1, \dots, J$, arriving at a system of $J - J_D$ algebraic equations for $J - J_D$ unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N \Omega$. Substituting interpolation formulae (3.28) into the BDIDE (3.27) leads to the following system:

$$c(x^i)u(x^i) + \sum_{x^j \in \Omega \cup \partial_N \Omega} KM'_{ij}u(x^j) = \Psi M(x^i) - \sum_{x^j \in \partial_D \Omega} KM'_{ij}\bar{u}(x^j),$$

$$x^i \in \Omega \cup \partial_N \Omega, \quad \text{no sum in } i, \quad (3.47)$$

where

$$KM'_{ij} = \int_{\omega(x^i) \cap \Omega} \Phi_j(x)R(x, x^i)d\Omega(x) + \int_{\bar{\omega}(x^i) \cap \partial_D \Omega} P(x, x^i)T\Phi_j(x)d\Gamma(x) -$$

$$- \int_{\bar{\omega}(x^i) \cap \partial_N \Omega} \Phi_j(x)T_x P(x, x^i)d\Gamma(x). \quad (3.48)$$

The details of the calculations of the boundary and domain integrals and the treatment of the weak singularity using Duffy transformation is given in Appendix B. Also, some literature about numerical integration techniques for domain integrals is listed in Appendix B.

3.5 Assembling the system matrix and right-hand side for BDIE/BDIDE

We shall discuss in this section the assembling of matrix A and right-hand side b for the case of mixed problems, with the Dirichlet and Neumann problems forming special cases. Let us start with a Laplace equation with a mesh of eight boundary elements, nine nodes and eight triangular cells, as shown in Figure 3.2.

For the BDIE method, the system of algebraic equations resulting from Eq.(3.43) has two unknown variables t and u , i.e. t on Dirichlet boundaries and u on Neumann boundaries, in addition to interior nodes.

In this chapter, we present two implementations using mixed boundary elements with linear variation of u and constant t , to avoid the discontinuity of t at corner points. In the first case, the collocation nodes for calculating t on Dirichlet boundaries are taken at the mid point of the boundary elements, while in the second case the collocation points are at the end nodes. Therefore, for the first case, the system $Ax = b$ is given by:

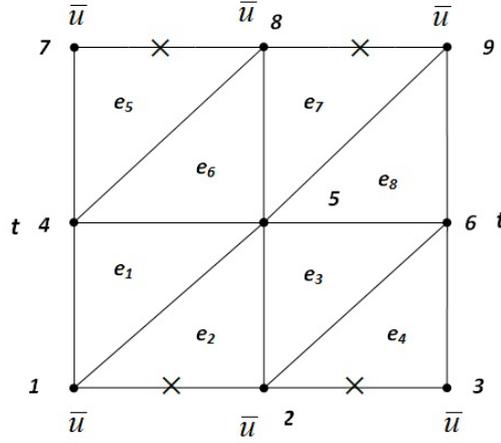


Figure 3.2: Simple mesh

$$A = \begin{bmatrix} A11 & A12 & A13 \\ A21 & D + A22 & A23 \\ A31 & A32 & I + A33 \end{bmatrix}_{7 \times 7}, \quad x = \begin{bmatrix} t \text{ on } \partial_D \Omega \\ u \text{ on } \partial_N \Omega \\ u \text{ on } \Omega \end{bmatrix}_{7 \times 1},$$

$$b = \begin{bmatrix} cvec + b11 + b12 + b13 \\ b21 + b22 + b23 \\ b31 + b32 + b33 \end{bmatrix}_{7 \times 1},$$

where $[A11]_{4 \times 4}$, $[A21]_{2 \times 4}$, $[A31]_{1 \times 4}$ are the integrals in Eq.(3.46) with the collocation nodes belonging to $\partial_D \Omega$, $\partial_N \Omega$ and interior nodes in Ω and integration nodes x^j belonging to $\partial_D \Omega$, respectively. Also, $[A12]_{4 \times 2}$, $[A22]_{2 \times 2}$, $[A32]_{1 \times 2}$ are the second integral in Eq.(3.45) with the collocation nodes x^i belonging to $\partial_D \Omega$, $\partial_N \Omega$ and interior nodes in Ω and integration nodes x^j belonging to $\partial_N \Omega$, respectively. Moreover, $[A13]_{4 \times 1}$, $[A23]_{2 \times 1}$, $[A33]_{1 \times 1}$ are the second integral in Eq.(3.45) with the collocation nodes x^i belonging to $\partial_D \Omega$, $\partial_N \Omega$ and interior nodes in Ω and integration nodes x^j belonging to Ω , respectively. In a general mesh the dimensions of the matrix are $[A]_{J-2 \times J-2}$, and right-hand side $[b]_{J-2 \times 1}$, where J

is the total number of nodes.

The matrix D in this simple case is given by

$$D = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}_{2 \times 2}, \text{ and the matrix } I \text{ is the identity matrix, in this case just equal } 1.$$

The right-hand side can be assembled in the same way as matrix A , where $[b11]_{4 \times 1}$, $[b21]_{2 \times 1}$, $[b31]_{1 \times 1}$ are the first integral in Eq.(3.44) with the collocation nodes belonging to $\partial_D \Omega$, $\partial_N \Omega$ and interior nodes in Ω , respectively. Also, $[b12]_{4 \times 1}$, $[b22]_{2 \times 1}$, $[b32]_{1 \times 1}$ are the second integral in Eq.(3.44) with the collocation nodes belonging to $\partial_D \Omega$, $\partial_N \Omega$ and interior nodes in Ω , respectively.

Moreover, $[b13]_{4 \times 1}$, $[b23]_{2 \times 1}$ and $[b33]_{1 \times 1}$ are equal to $-\sum_{x^j \in \partial_D \Omega} K_{ij} \bar{u}(x^j)$ (with only the second integral in Eq.(3.45), since $R = 0$ for the Laplace equation, therefore, the first integral disappears), with the collocation nodes belonging to $\partial_D \Omega$, $\partial_N \Omega$ and interior nodes in Ω , respectively. In addition, $[cvec]_{4 \times 1}$ is a vector equal to $-c(x^i) \bar{u}(x^i)$, with the collocation nodes belonging to $\partial_D \Omega$ and the values of c given in Eq.(2.18).

In the second case, where the collocation nodes for calculating t in Dirichlet boundaries are taken at the end points of the elements, the system is given by:

$Ax = b$, where $[A]_{m \times n}$ and $m \geq n$. This system can be solved in the least square sense by solving the system $A^T Ax = A^T b$, where for our simple mesh the matrices are given by:

$$A = \begin{bmatrix} A11 & A12 & A13 \\ A21 & D + A22 & A23 \\ A31 & A32 & I + A33 \end{bmatrix}_{9 \times 7}, \quad x = \begin{bmatrix} t \in \partial_D \Omega \\ u \in \partial_N \Omega \\ u \in \Omega \end{bmatrix}_{7 \times 1},$$

$$b = \begin{bmatrix} cvec + b11 + b12 + b13 \\ b21 + b22 + b23 \\ b31 + b32 + b33 \end{bmatrix}_{9 \times 1},$$

where $[A11]_{6 \times 4}$, $[A21]_{2 \times 4}$, $[A31]_{1 \times 4}$, $[A12]_{6 \times 2}$, $[A22]_{2 \times 2}$, $[A32]_{1 \times 2}$, $[A13]_{6 \times 1}$, $[A23]_{2 \times 1}$, $[A33]_{1 \times 1}$ and the matrices D , I are the same as before.

By applying the least squares technique, the final system will be $Cx = d$:

$[C]_{7 \times 7} = [A^T]_{7 \times 9} [A]_{9 \times 7}$, and $[d]_{7 \times 1} = [A^T]_{7 \times 9} [b]_{9 \times 1}$. Also, the right-hand side b can be calculated like in the previous case but with different dimensions for the sub-vectors, i.e., $[b11]_{6 \times 1}$, $[b12]_{6 \times 1}$, $[b13]_{6 \times 1}$, $[cvec]_{6 \times 1}$ and the other sub-vectors are the same as before.

For the BDIDE method, the system of algebraic equations in Eq.(3.47) has only one un-

known variable u , i.e. u in Neumann parts in addition to interior nodes. In this case, the assembling of matrix A and vector b is much easier than in the BDIE, i.e. by just adding the sub-matrices or sub-vectors which have the same dimension $J \times J$ for matrix A_{full} or vector b_{full} . The matrix A_{full} and vector b_{full} only have coefficients on positions x^i , $x^j \in \Omega \cup \partial_N \Omega$ and zero elsewhere. So, for the simple mesh, we can construct matrix A and vector b from A_{full} , b_{full} respectively, and the system $Ax = b$ is given by:

$$A = \begin{bmatrix} A1 + A2 + A3 \end{bmatrix}_{k \times k}, \quad x = \begin{bmatrix} u \in \partial_N \Omega \\ u \in \Omega \end{bmatrix}_{k \times 1}, \quad b = \begin{bmatrix} b1 + b2 + b3 + b4 \end{bmatrix}_{k \times 1},$$

where $k = J - r$, J is the total number of nodes and r is the number of nodes on Dirichlet boundaries. The matrix $A1$ in this simple case is,

$$A1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}_{3 \times 3}, \quad \text{the matrix } [A2]_{3 \times 3} \text{ is the second integral in Eq.(3.48) and } [A3]_{3 \times 3} \text{ is the third integral in Eq.(3.48).}$$

Finally, the right-hand side b can be calculated by adding the sub-vectors $b1, b2, b3, b4$, where $b1, b2$ are the first and the second integrals in Eq.(3.44), respectively. The vector $b3$ and $b4$ are defined as:

$$b3 = - \sum_{x^j \in \partial_D \Omega} K1_{ij} \bar{u}(x^j), \quad b4 = - \sum_{x^j \in \partial_D \Omega} K2_{ij} \bar{u}(x^j),$$

where

$$K1_{ij} = \int_{\bar{\omega}(x^i) \cap \partial_D \Omega} P(x, x^i) T \Phi_j(x) d\Gamma(x),$$

$$K2_{ij} = - \int_{\bar{\omega}(x^i) \cap \partial_N \Omega} \Phi_j(x) T_x P(x, x^i) d\Gamma(x).$$

Remark 3.1: In order to assemble the system of algebraic equations for the Poisson equation, where the third integral in Eq.(3.44) appears, one can follow the same steps as before for both BDIE and BDIDE in addition to the domain integral on the right-hand side b . However, to solve Eqs.(3.1)-(3.5), which have been re-formulated to BDIE or BDIDE in Eqs.(3.43), (3.47), an extra domain integral appears (first integral in Eqs.(3.45) and (3.48)). Therefore, extra sub-matrices will be added to the matrix A for both BDIE and

BDIDE.

3.6 Numerical results

In this section, we shall examine some test examples to assess the performance of the BDIE/ BDIDE formulations with either Dirichlet, Neumann or mixed boundary conditions. To verify the convergence of the methods, we applied them to some test problems on square and circular domains, for which an exact analytical solution, u_{exact} , is available. Also, the relative error was calculated as in chapter 2, Eq.(2.35).

3.6.1 Laplace's equation with mixed boundary conditions

The starting point for testing the BDIE/ BDIDE formulations is to consider Laplace's equation with mixed boundary conditions. In this case there is no domain integral, i.e. $f = 0$ and $a(x) = 1$ in our original BVP (3.1)-(3.5),

$$\nabla^2 u := \Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = 0$$

This simple test involves a square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, with boundary conditions:

(3.2) is $\bar{u}(x) = x_1 + x_2$, (3.3) is $\bar{t}(x) = n_1(x) + n_2(x)$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $\{x_2 = 2; x_2 = 3; 2 \leq x_1 \leq 3\}$; and (3.3) for $\{x_1 = 2; x_1 = 3; 2 \leq x_2 \leq 3\}$.

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

3.6.2 Poisson's equation with mixed boundary conditions

The next test considers Poisson's equation, in which case there is a domain integral coming from $f \neq 0$; we still consider $a(x) = 1$ in the original BVP (3.1)-(3.5), and assume a square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, $f(x) = 4$ for $x \in \bar{\Omega}$, with boundary conditions, (3.2) is $\bar{u}(x) = x_1^2 + x_2^2$, (3.3) is $\bar{t}(x) = 2(x_1 n_1(x) + x_2 n_2(x))$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $\{x_2 = 1; x_2 = 2; 1 \leq x_1 \leq 2\}$; and (3.3) for $\{x_1 = 1; x_1 = 2; 1 \leq x_2 \leq 2\}$.

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

The next series of tests with variable coefficients involve simple square or circular geometries with increasing degree of complexity of the variation of both the material parameter coefficients and the body force term f . The exact solutions of the problems range from linear to cubic, and will be used to verify the convergence of the numerical solutions.

3.6.3 Test 1

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = 2(x_1 + x_2)$, $f(x) = 4$ for $x \in \bar{\Omega}$, with boundary conditions:

(3.2) is $\bar{u}(x) = x_1 + x_2$, (3.3) is $\bar{t}(x) = 2(x_1 + x_2)(n_1(x) + n_2(x))$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $\{x_2 = 2; x_2 = 3; 2 \leq x_1 \leq 3\}$; and (3.3) for $\{x_1 = 2; x_1 = 3; 2 \leq x_2 \leq 3\}$.

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

3.6.4 Test 2

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, $a(x) = x_1^2 + x_2^2$, $f(x) = 8(x_1^2 + x_2^2)$ for $x \in \bar{\Omega}$, with boundary conditions:

(3.2) is $\bar{u}(x) = x_1^2 + x_2^2$, (3.3) is $\bar{t}(x) = 2(x_1^2 + x_2^2)(x_1 n_1(x) + x_2 n_2(x))$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $\{x_2 = 2; x_2 = 3; 2 \leq x_1 \leq 3\}$; and (3.3) for $\{x_1 = 2; x_1 = 3; 2 \leq x_2 \leq 3\}$.

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

3.6.5 Test 3

Circular domain $\bar{\Omega} = \{(x_1, x_2) : (x_1 - 1.5)^2 + (x_2 - 1.5)^2 \leq 0.25\}$, $a(x) = x_1^2 + x_2^2$, $f(x) = 8(x_1^2 + x_2^2)$ for $x \in \bar{\Omega}$, with boundary conditions:

(3.2) is $\bar{u}(x) = x_1^2 + x_2^2$, (3.3) is $\bar{t}(x) = 2(x_1^2 + x_2^2)(x_1 n_1(x) + x_2 n_2(x))$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $x_2 = \sqrt{0.25 - (x_1 - 1.5)^2} + 1.5$; and (3.3) for $x_2 = -\sqrt{0.25 - (x_1 - 1.5)^2} + 1.5$.

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

3.6.6 Test 4

Circular domain $\bar{\Omega} = \{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}$, $a(x) = x_1 + x_2 + 4$, $f(x) = 2$ for $x \in \bar{\Omega}$, with boundary conditions:

(3.2) is $\bar{u}(x) = x_1^2 + x_2^2$, (3.3) is $\bar{t}(x) = (x_1 + x_2 + 4)(n_1(x) + n_2(x))$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $x_2 = \sqrt{1 - x_1^2}$; and (3.3) for $x_2 = -\sqrt{1 - x_1^2}$.

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

3.6.7 Test 5

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = 2 \exp(x_1 + x_2)$ for $x \in \bar{\Omega}$, with boundary conditions:

(3.2) is $\bar{u}(x) = x_1 + x_2$, (3.3) is $\bar{t}(x) = \exp(x_1 + x_2)(n_1(x) + n_2(x))$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $\{x_2 = 2; x_2 = 3; 2 \leq x_1 \leq 3\}$; and (3.3) for $\{x_1 = 2; x_1 = 3; 2 \leq x_2 \leq 3\}$.

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

3.6.8 Test 6

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = \exp(x_1 + x_2)(6x_1 + 3x_1^2 + 6x_2 + 3x_2^2)$ for $x \in \bar{\Omega}$, with boundary conditions:

(3.2) is $\bar{u}(x) = x_1^3 + x_2^3$, (3.3) is $\bar{t}(x) = \exp(x_1 + x_2)(3x_1^2 n_1(x) + 3x_2^2 n_2(x))$, and for the mixed boundary conditions (3.4), (3.5), we have (3.2) for $\{x_2 = 2; x_2 = 3; 2 \leq x_1 \leq 3\}$; and (3.3) for $\{x_1 = 2; x_1 = 3; 2 \leq x_2 \leq 3\}$.

The exact solution for this problem is $u_{exact}(x) = x_1^3 + x_2^3$, $x \in \bar{\Omega}$.

The surface plots of the numerical solutions were obtained with the most refined mesh in each example. The graph of relative error has the number of nodes on the horizontal axis and the relative error on the vertical axis. Both boundary and domain integrals appearing in the formulation have a weak singularity. To calculate the boundary integrals we used a standard Gaussian quadrature rule. For the domain integrals, we have implemented a Gaussian quadrature rule for two dimensions with Duffy transformation, by mapping the triangles into squares and eliminating the weak singularity, see Appendix B.

3.6.9 Dirichlet problem

Numerical results for BDIDE

The results in Figures 3.3 and 3.4 are for the Laplace and Poisson tests, respectively. In these tests, the domain is square and the exact solutions are linear and quadratic,

respectively. Since we are using linear basis functions, there is no interpolation error for Laplace’s equation test, but there is interpolation error for Poisson’s equation as the exact solution is quadratic. In addition, other errors come either from discretisation of the domain into triangles or from calculating the boundary and domain integrals numerically; very good results and high rates of convergence are obtained for both tests.

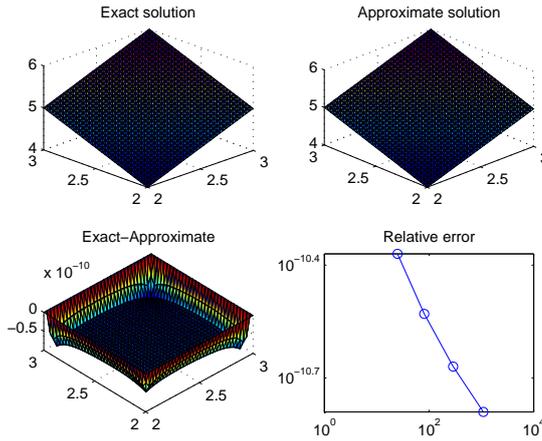


Figure 3.3: Relative error for Laplace’s equation; when $J=1089$, $r(1089) \approx 1.62 \times 10^{-11}$

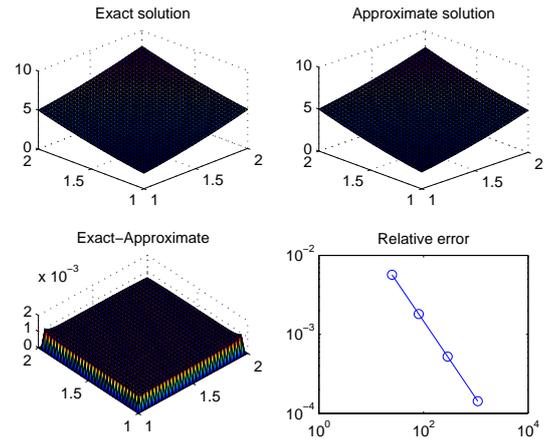


Figure 3.4: Relative error for Poisson’s equation; when $J=1089$, $r(1089) \approx 1.00 \times 10^{-4}$

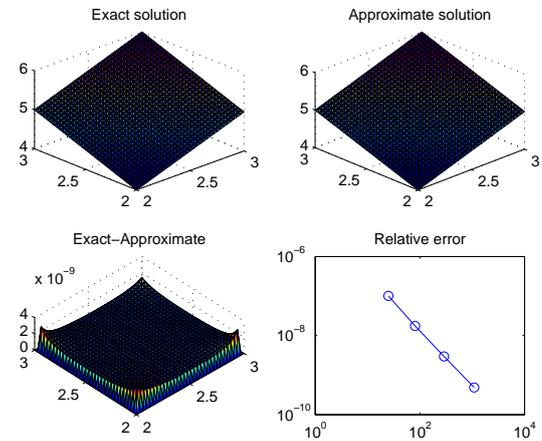


Figure 3.5: Relative error for Test 1; when $J=1089$, $r(1089) \approx 4.84 \times 10^{-10}$

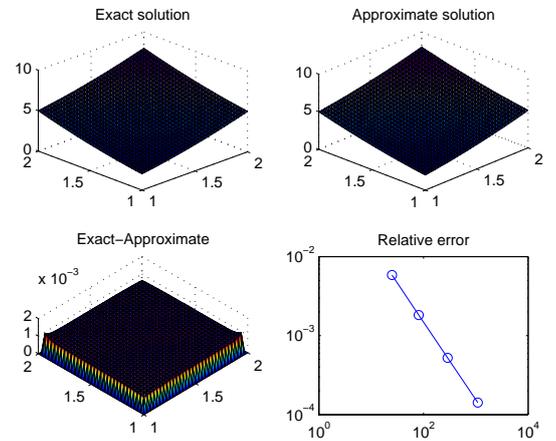


Figure 3.6: Relative error for Test 2; when $J=1089$, $r(1089) \approx 1.42 \times 10^{-4}$

Tests 1 and 2 analyse problems with variable coefficients, so there is one more domain integral coming from the remainder, i.e. $R \neq 0$. Therefore, there are discretisation and numerical integration errors for test 1. Also, there is interpolation error for test 2, as the exact solution is quadratic. It can be seen clearly from Figures 3.5 and 3.6 the convergence of the solution by increasing the number of elements, for both tests.

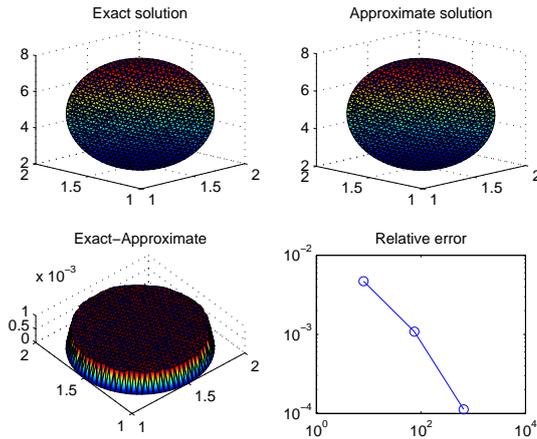


Figure 3.7: Relative error for Test 3; when $J=656$, $r(656) \approx 1.13 \times 10^{-4}$

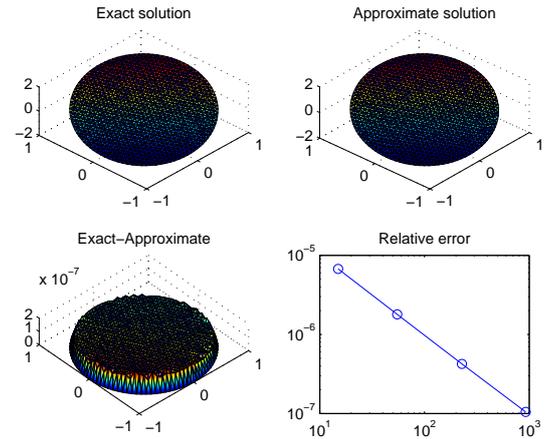


Figure 3.8: Relative error for Test 4; when $J=925$, $r(925) \approx 1.05 \times 10^{-7}$

In addition, for the circular domain in tests 3 and 4, an extra approximation error has been added which comes from approximating the boundary curve by polygons. It can be seen clearly from Figures 3.7 and 3.8 that test 4 gives better results than test 3 in this case; a possible reason is that there is interpolation error for test 3, as the exact solution is quadratic.

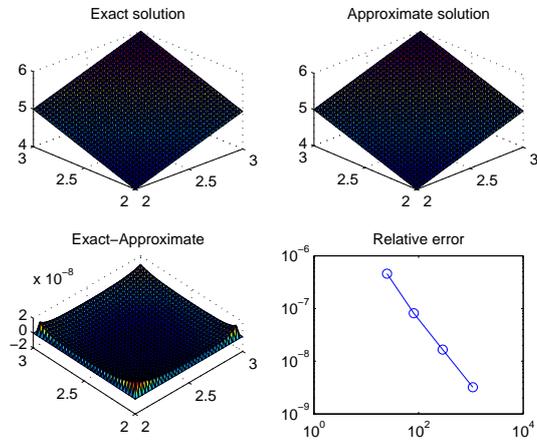


Figure 3.9: Relative error for Test 5; when $J=1089$, $r(1089) \approx 3.22 \times 10^{-9}$

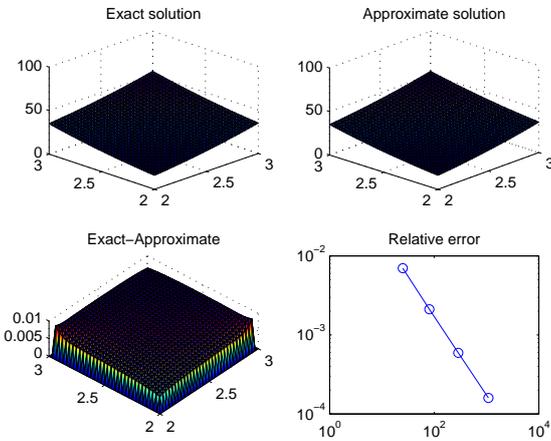


Figure 3.10: Relative error for Test 6; when $J=1089$, $r(1089) \approx 1.60 \times 10^{-4}$

The exact solutions for tests 5 and 6 are linear and cubic, respectively. It can be seen clearly from Figures 3.9 and 3.10 that the results for test 5 are still better than test 6, as there is interpolation error for test 6 since the exact solution is cubic.

Numerical results for BDIE

As both u and t along the boundary are calculated in the BDIE method, we implemented mixed boundary elements with linear u and constant t to avoid the discontinuities of t at corner points.

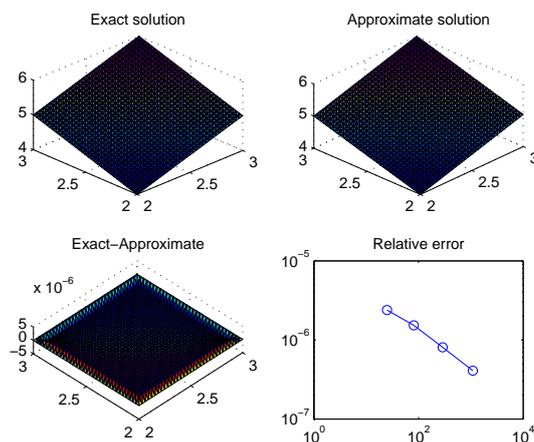


Figure 3.11: Relative error for Laplace's equation; when $J=1089$, $r(1089) \approx 4.09 \times 10^{-7}$

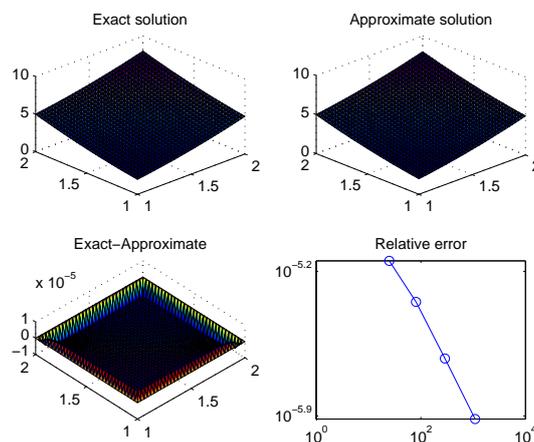


Figure 3.12: Relative error for Poisson's equation; when $J=1089$, $r(1089) \approx 1.22 \times 10^{-6}$

From Figures 3.11-3.14 the results for the Laplace and Poisson equations and for tests 1 and 2 are very good, with high rates of convergence. Moreover, it is clear that the BDIE results for the Poisson equation and test 2 are better than the BDIDE results for the same problems.

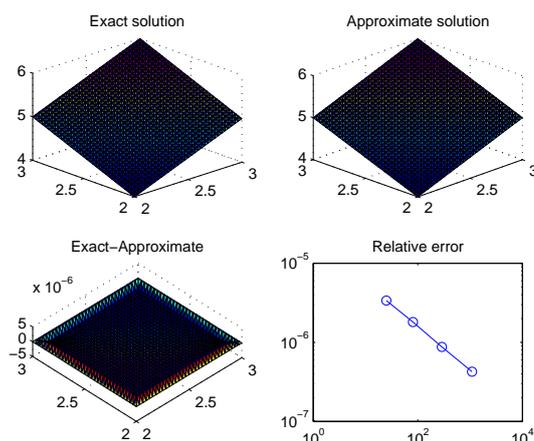


Figure 3.13: Relative error Test 1; when $J=1089$, $r(1089) \approx 4.25 \times 10^{-7}$

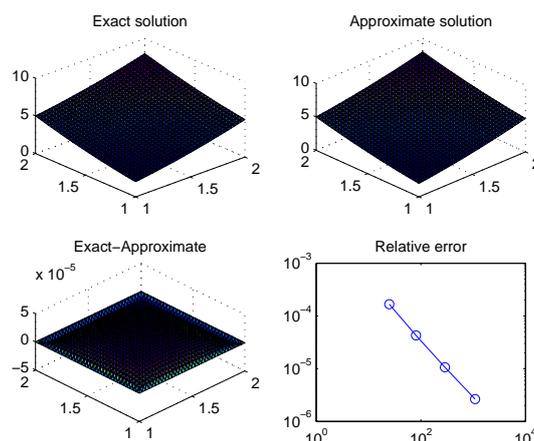


Figure 3.14: Relative error Test 2; when $J=1089$, $r(1089) \approx 2.60 \times 10^{-6}$

Moreover, in tests 3 and 4 the exact solutions are quadratic and linear, respectively. Since both tests have a circular domain, an extra approximation error has been added which

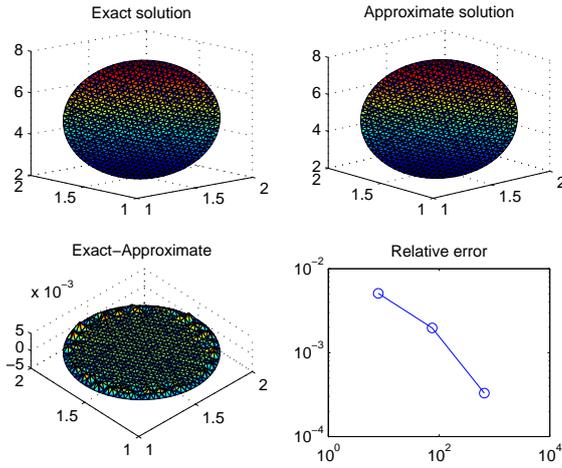


Figure 3.15: Relative error for Test 3; when $J=656$, $r(656) \approx 3.29 \times 10^{-4}$

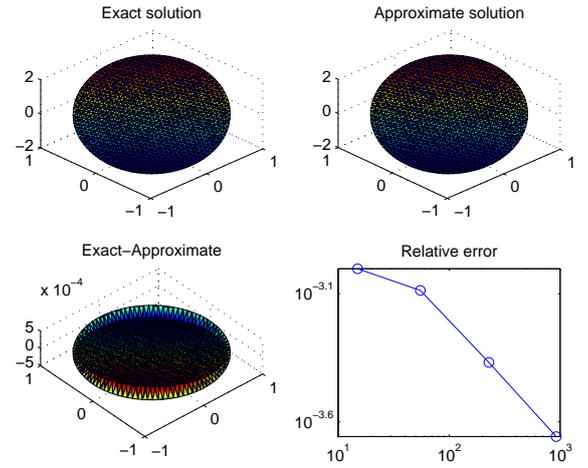


Figure 3.16: Relative error for Test 4; when $J=925$, $r(925) \approx 2.20 \times 10^{-4}$

comes from approximating the boundary curve by polygons. Therefore, lower accuracy is obtained (see Figures 3.15 and 3.16) in comparison with tests 1 and 2. It can be seen also that the BDIDE gives better results than BDIE in this case; a possible reason is the approximate calculation of the value of c for BDIE which is avoided for the BDIDE.

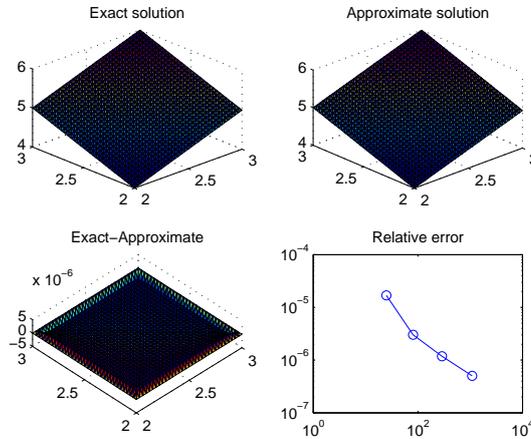


Figure 3.17: Relative error for Test 5; when $J=1089$, $r(1089) \approx 5.00 \times 10^{-7}$

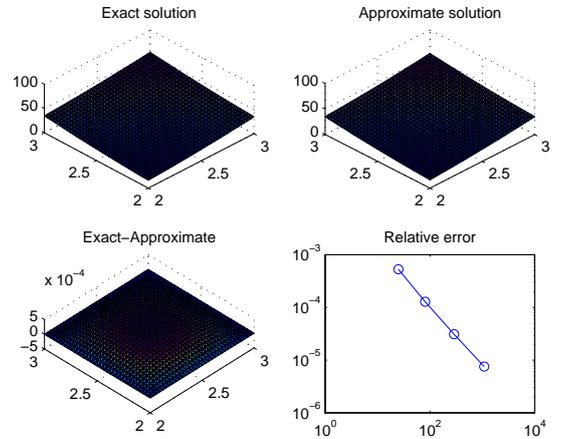


Figure 3.18: Relative error for Test 6; when $J=1089$, $r(1089) \approx 7.50 \times 10^{-6}$

Finally, in tests 5 and 6, good rates of convergence have been achieved for both tests, see Figures 3.17 and 3.18. The results for test 6 for the BDIE are better than those for the BDIDE.

3.6.10 Neumann problem

Numerical results for BDIE

As discussed in the subsection 3.3.2, the perturbation operator ensures that the solution of the Neumann problem has **zero mean integral** on the boundary, i.e.

$$\int_{\partial\Omega} u_{approx} d\Gamma(x) = 0.$$

The exact solution of course has also to be constructed such that the same property holds

$$\int_{\partial\Omega} u_{exact} d\Gamma(x) = 0.$$

But our exact solution did not have this property, therefore we have to change it by a constant \tilde{c} , i.e. $u_{exact} - \tilde{c}$, we define $u_{exact} = \tilde{u} - \tilde{c}$ with $\int_{\partial\Omega} \tilde{u} d\Gamma(x) \neq 0$. As

$$0 = \int_{\partial\Omega} u_{exact} = \int_{\partial\Omega} (\tilde{u} - \tilde{c}) = \int_{\partial\Omega} \tilde{u} - |\partial\Omega|\tilde{c}$$

we could deduce

$$\tilde{c} = \frac{1}{|\partial\Omega|} \int_{\partial\Omega} \tilde{u}.$$

The value of $\tilde{c} = \frac{1}{|\partial\Omega|} \int_{\partial\Omega} u(x) d\Gamma(x)$ for Laplace's equation can be calculated in the parameterisation equation for the boundary curve through the following steps:

$$\frac{1}{|\partial\Omega|} \int_{\partial\Omega} u(x) d\Gamma(x) = \frac{1}{|\partial\Omega|} \left[\int_{\Gamma_1} u(x) d\Gamma_1(x) + \int_{\Gamma_2} u(x) d\Gamma_2(x) + \int_{\Gamma_3} u(x) d\Gamma_3(x) + \int_{\Gamma_4} u(x) d\Gamma_4(x) \right],$$

where $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ are the boundary segments starting from point $(2, 2)$ in anti-clockwise direction.

So,

$$\Gamma_1 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} + t \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad 0 \leq t \leq 1$$

$d\Gamma_1(x) = |\dot{x}(t)| dt = 1$, since $\dot{x}(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, therefore,

$$\int_{\Gamma_1} (x_1 + x_2) d\Gamma_1(x) = \int_0^1 [(2+t) + (2+0)] \cdot 1 d\Gamma_1(x) = 4.5$$

and similarly,

$$\Gamma_2 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} + t \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad 0 \leq t \leq 1,$$

$$\int_{\Gamma_2} (x_1 + x_2) d\Gamma_2(x) = 5.5,$$

$$\Gamma_3 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} + t \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad 0 \leq t \leq 1,$$

$$\int_{\Gamma_3} (x_1 + x_2) d\Gamma_3(x) = 5.5,$$

$$\Gamma_4 = \begin{pmatrix} x_1 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + t \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad 0 \leq t \leq 1,$$

$$\int_{\Gamma_4} (x_1 + x_2) d\Gamma_4(x) = 4.5,$$

and then, $\tilde{c} = \frac{1}{|\partial\Omega|} \cdot 20 = \frac{1}{4} \cdot 20 = 5 \Rightarrow u_{exact} = x_1 + x_2 - 5$.

For Poisson's equation the value of \tilde{c} can be calculated with the same steps above and then we get the value $\tilde{c} = \frac{58}{12}$, so the exact solution was taken as $u_{exact} = x_1^2 + x_2^2 - \frac{58}{12}$.

The results in Figures 3.19 and 3.20 demonstrate that the BDIE method is able to generate accurate solutions for both Laplace and Poisson equations.

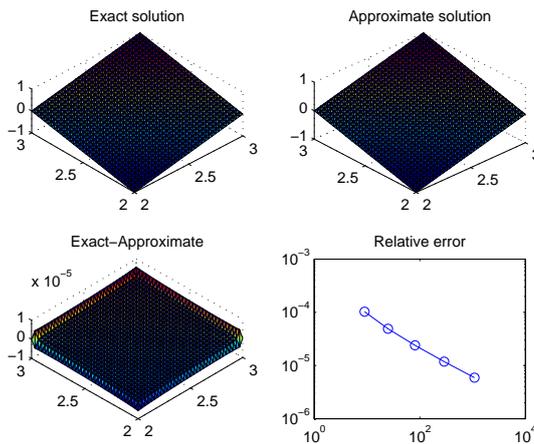


Figure 3.19: Relative error for Laplace's equation; when $J=1089$, $r(1089) \approx 5.9 \times 10^{-6}$

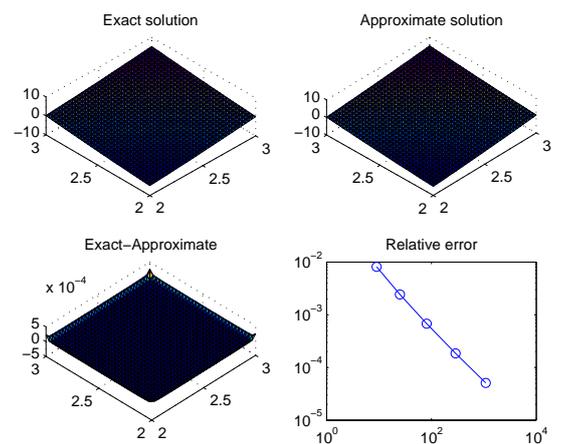


Figure 3.20: Relative error for Poisson's equation; when $J=1089$, $r(1089) \approx 5.09 \times 10^{-5}$

In test 1, the value of \tilde{c} can be calculated with exactly the same steps as in the Laplace's

equation test and then, $u_{exact} = x_1 + x_2 - 5$. Also, the exact solution for test 2 was taken as $u_{exact} = x_1^2 + x_2^2 - \frac{58}{12}$.

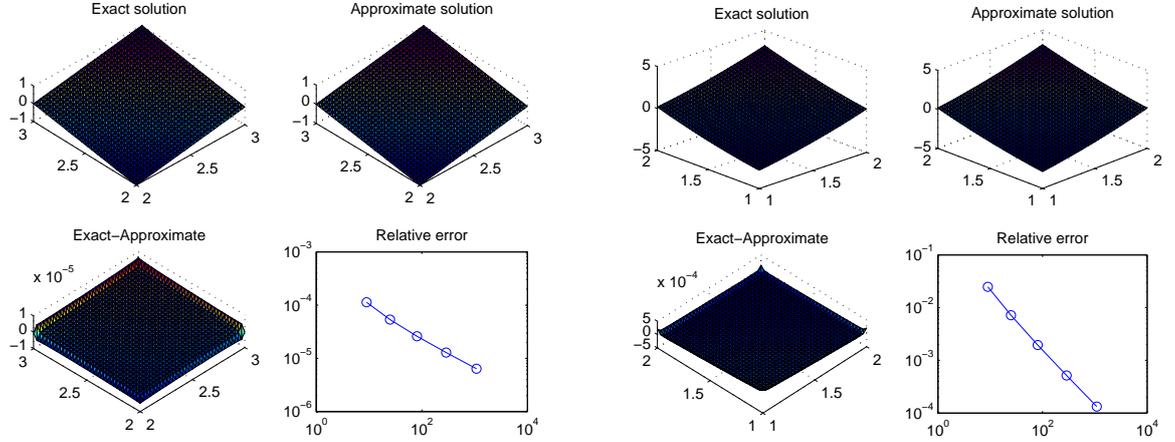


Figure 3.21: Relative error for Test 1; when $J=1089$, $r(1089) \approx 6.36 \times 10^{-6}$

Figure 3.22: Relative error for Test 2; when $J=1089$, $r(1089) \approx 1.33 \times 10^{-4}$

In tests 3 and 4, the domain is a circle, and the value of \tilde{c} can also be calculated by the parameterisation equation for the boundary curve but slightly different than in examples (1) and (2), since the general equation of the circle with center (h, k) is

$$(x - h)^2 + (y - k)^2 = r^2$$

Then,

$$\Gamma = \begin{pmatrix} r \cos(t) + h \\ r \sin(t) + k \end{pmatrix}, \quad 0 \leq t \leq 2\pi$$

where r is the radius of the circle, equal to 0.5, and $h = k = 1.5$, $d\Gamma(x) = |\dot{x}(t)|dt = 0.5$, since $\dot{x}(t) = \begin{pmatrix} -\frac{1}{2} \sin(t) \\ \frac{1}{2} \cos(t) \end{pmatrix}$, therefore,

$$\frac{1}{|\partial\Omega|} \int_{\partial\Omega} u(x) d\Gamma(x) = \frac{1}{|\partial\Omega|} \int_{\Gamma} u(x) d\Gamma(x) = \frac{19}{4},$$

where $\frac{1}{|\partial\Omega|} = \pi$, so the exact solution was taken as $u_{exact} = x_1^2 + x_2^2 - \frac{19}{4}$. The exact solution for test 4 was taken to be $u_{exact}(x) = x_1 + x_2$ as $\tilde{c} = 0$ in this case. The numerical results for both tests are given in Figures 3.23 and 3.24.

Remark 3.2:

The values of $c(x^i) = \frac{\alpha(x^i)}{2\pi}$, where $\alpha(x^i)$ is the interior angle at a corner point x^i of the

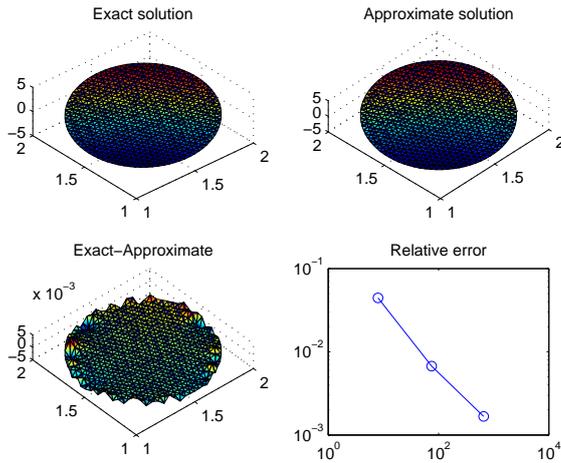


Figure 3.23: Relative error for Test 3; when $J=656$, $r(656) \approx 1.67 \times 10^{-3}$

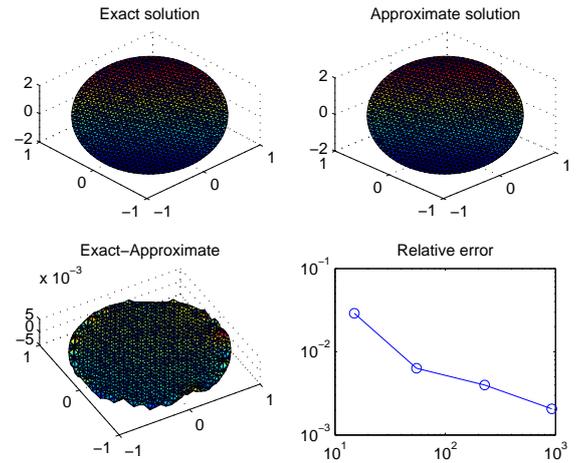


Figure 3.24: Relative error for Test 4; when $J=925$, $r(925) \approx 2.06 \times 10^{-3}$

boundary $\partial\Omega$ in Eq.(2.18). By using the fact that the sum of the interior angles of a regular polygon with n sides is $(n - 2) \times 180$, the interior angle of a regular polygon with n sides is just $(n - 2) \times 180$ divided by the number of sides n .

Finally, by following the same procedure for calculating the value of \tilde{c} the exact solution for test 5 is $u_{exact}(x) = x_1 + x_2 - 5$, and for test 6 we have $u_{exact}(x) = x_1^3 + x_2^3 - \frac{135}{4}$. The numerical results for both tests are given in Figures 3.25 and 3.26.

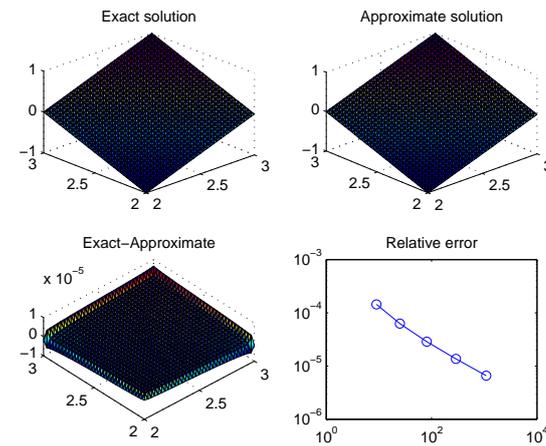


Figure 3.25: Relative error for Test 5; when $J=1089$, $r(1089) \approx 6.60 \times 10^{-6}$

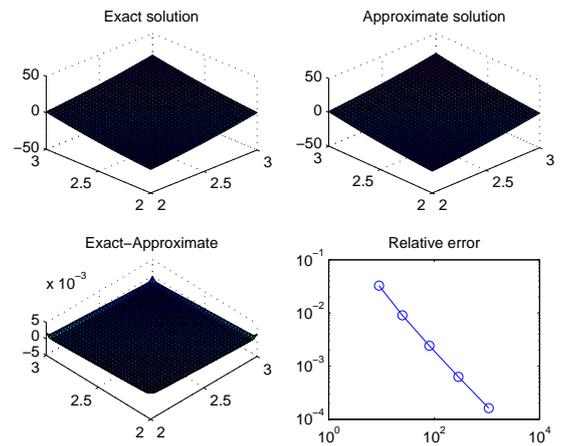


Figure 3.26: Relative error for Test 6; when $J=1089$, $r(1089) \approx 2.00 \times 10^{-4}$

3.6.11 Mixed problem

Numerical results for BDIE and BDIDE

The results in Figures 3.27 to 3.34 demonstrate that both the BDIE and BDIDE methods are able to generate accurate and convergent solutions for the BVP (3.1) with mixed boundary conditions (3.4) and (3.5). When comparing the solutions obtained by using both methods, it can be seen that the BDIDE method produced better results for the Laplace equation and for tests 1 and 3-5. However, more accurate results were obtained for the BDIE method for the Poisson equation and for tests 2 and 6. A possible explanation is the approximation of the flux t in the BDIDE method using linear basis functions for u living on triangles; thus, $T\Phi_j(x)$ is constant within each triangle. These approximations are appropriate for the Laplace equation and for tests 1, 4 and 5 as the solution to these tests are all linear, while the solution to the Poisson equation and tests 2, 3 and 6 are quadratic or cubic. The accuracy of the BDIE for tests 3 and 4 is also reduced by the approximation of the flux t at the boundary nodes, as there is a slight flux discontinuity at these points which is avoided in the BDIDE method.

Table 3.1: Relative errors for all tests by using BDIE and BDIDE

No. of nodes	Tests	BDIE (mid-node)	BDIE (end-node)	BDIDE
1089	Laplace	2.48×10^{-5}	1.01×10^{-6}	5.80×10^{-7}
1089	Poisson	4.64×10^{-5}	4.40×10^{-6}	1.00×10^{-4}
1089	Test 1	1.62×10^{-4}	1.38×10^{-6}	6.10×10^{-7}
1089	Test 2	5.00×10^{-4}	1.40×10^{-5}	1.00×10^{-4}
3715	Test 3	2.63×10^{-4}	4.01×10^{-4}	4.56×10^{-5}
3715	Test 4	1.00×10^{-5}	2.00×10^{-5}	5.00×10^{-7}
1089	Test 5	8.00×10^{-4}	2.70×10^{-6}	6.80×10^{-7}
1089	Test 6	9.00×10^{-4}	2.26×10^{-5}	2.00×10^{-4}

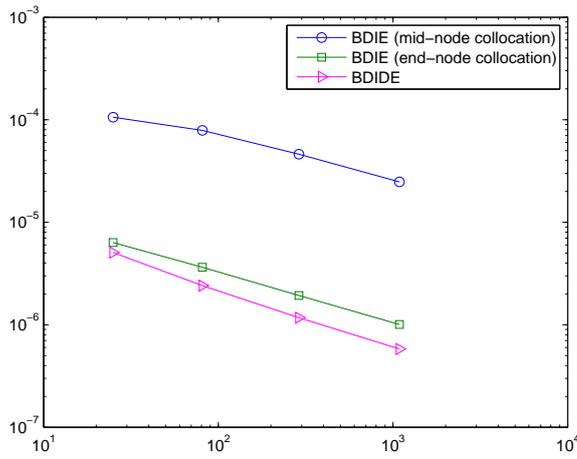


Figure 3.27: Relative errors for Laplace's equation

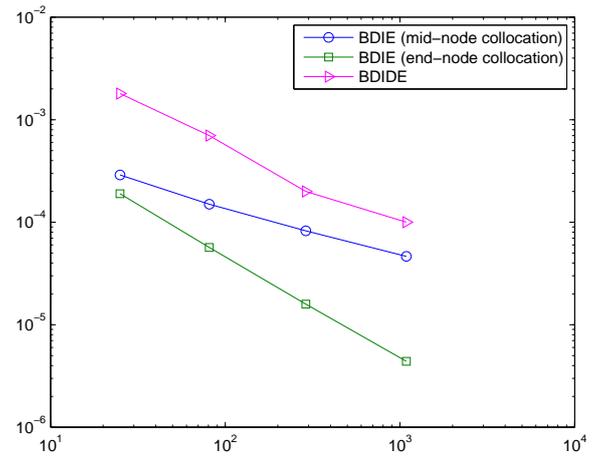


Figure 3.28: Relative errors for Poisson's equation

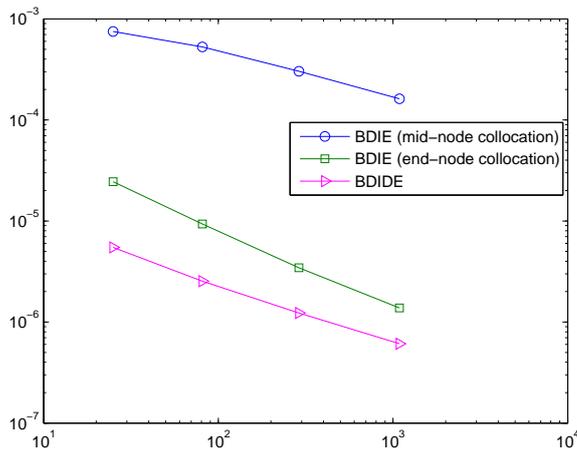


Figure 3.29: Relative errors for Test 1

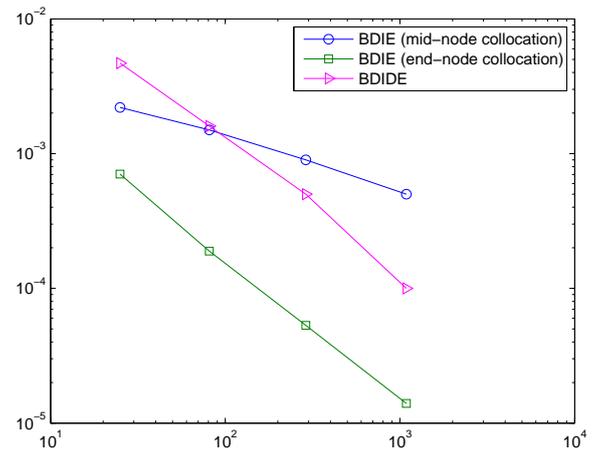


Figure 3.30: Relative errors for Test 2

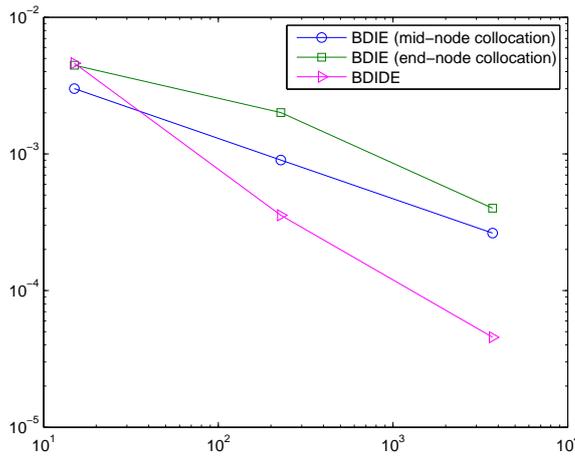


Figure 3.31: Relative errors for Test 3

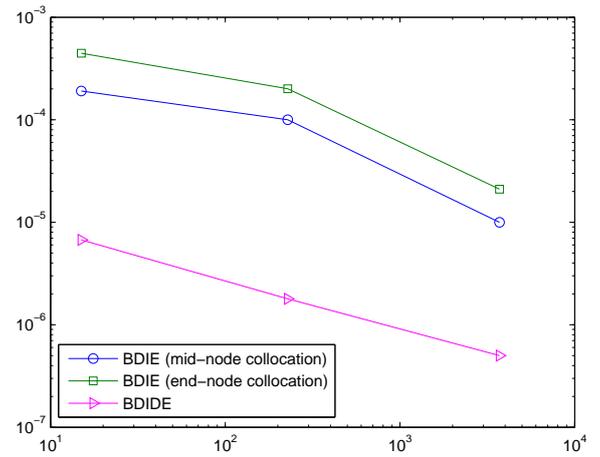


Figure 3.32: Relative errors for Test 4

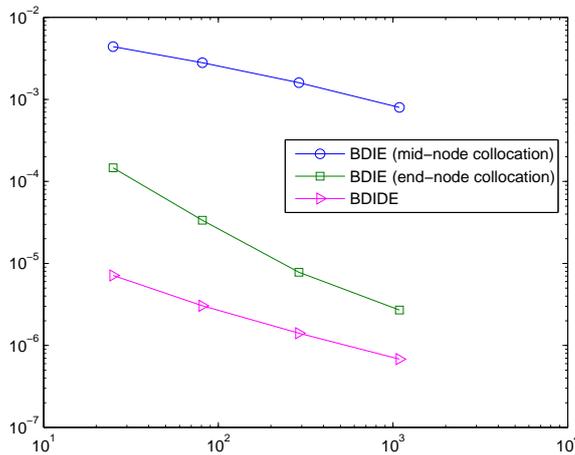


Figure 3.33: Relative errors for Test 5

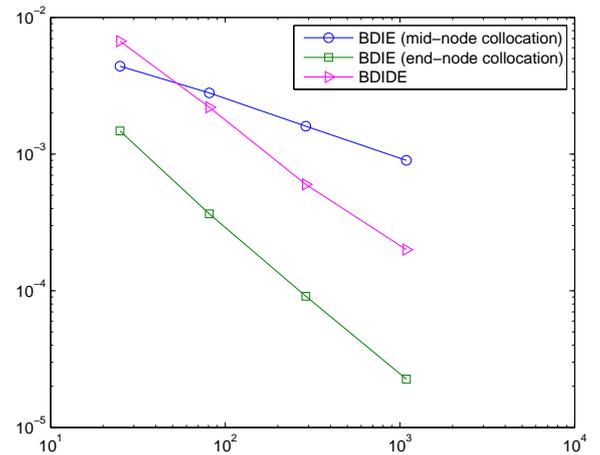


Figure 3.34: Relative errors for Test 6

3.7 Concluding remarks

In this chapter, the BDIE and BDIDE methods are developed and implemented for solving two-dimensional second-order linear elliptic mixed problems with variable coefficients with either Dirichlet, Neumann or mixed boundary conditions. Convergence studies with mesh refinement show that the present methods produce accurate results. The boundary and domain integrals appearing in the formulations have a weak singularity. To calculate the boundary integrals we used a standard Gaussian quadrature rule. For the domain integrals, we have implemented a Gaussian quadrature rule with Duffy transformation by mapping the triangles into squares and eliminating the weak singularity, which is discussed

in more details in Appendix B. The following remarks apply to the present approach:

- A parametrix (Levi function), which is available for equations with variable coefficients, is used as a test function;
- The values of the unknown variables are obtained accurately with the present methods;
- Unlike in the standard BEM, the unknown function u is approximated using linear basis functions living on triangles for both BDIE and BDIDE methods, allowing to obtain the values of u at interior points directly;
- As both u and t along the boundary are calculated in the BDIE method, we implemented mixed boundary elements with linear u and constant t to avoid the discontinuities of t at corner points. In this case, collocation was tested at the mid and end points of each boundary element. It was shown that end-node collocation generally provides higher accuracy than mid-node collocation;
- The only boundary variable in the BDIDE method is u along Neumann boundaries, thus there is no need for collocation along Dirichlet boundaries. Thus, the problem caused by the discontinuity of the normal derivative at corner points is avoided. This feature will save memory and computational time when we apply the BDIDE for practical problems;
- The generation, assembly and solution of the system of linear equations for the BDIE method are more complicated and thus take longer than that for the BDIDE method.

Chapter 4

Helmholtz equation with variable coefficients

4.1 Introduction

The Helmholtz equation is widely used to model many problems in physics and mechanics. If the material is homogeneous and there are no source/sink terms, then the governing equation is the homogeneous Helmholtz equation [35,36]. When source terms are present, however, a non-homogeneous Helmholtz equation must be considered. Numerical solutions of these problems, in either the homogeneous or the non-homogeneous case, have been obtained by means of the finite element method (FEM) and the finite difference method (FDM).

Rangogni [36] presented a BEM formulation for the non-homogeneous Helmholtz equation with harmonic source terms, and the domain integral transformed to a boundary integral using Green's formula. The BEM for a non-homogeneous Helmholtz equation with variable coefficients is discussed in [35]. The authors used the fundamental solution for the Laplace equation to transform the non-homogeneous Helmholtz equation to a boundary integral equation, and then an iteration method was used to solve the BIE. A comparative study of BEM and FEM for the Helmholtz equation in two dimensions is performed in [38]; the numerical investigations showed that the BEM is generally more accurate than the FEM when the size of the finite elements is comparable to that of the boundary elements, especially for the Dirichlet problem. As in previous chapters, in order to apply the BEM for Helmholtz equation the fundamental solution must be known. For Helmholtz equation

with constant coefficients the fundamental solution is well-known [5, 6, 38]. However, for Helmholtz equations with variable coefficients, fundamental solutions are only available for simple forms of coefficients, see [38]. Recently, Marin et al. [39, 40] successfully applied the dual reciprocity boundary element method (DRBEM) for a Helmholtz equation with variable coefficients. In this chapter, the boundary-domain integral or integro-differential equation (BDIE or BDIDE) formulations developed in chapter three for heat conduction are extended to the treatment of the two-dimensional Helmholtz equation with variable coefficients. Four possible cases are investigated, first of all when both material parameters and wave number are constant, in which case the zero-order Bessel function of the second kind is used as fundamental solution.

Moreover, when the material parameters are variable (with constant or variable wave number), a parametrix is adopted to reduce the Helmholtz equation to a BDIE or BDIDE. However, when material parameters are constant (with variable wave number), the standard fundamental solution for the Laplace equation is used in the formulation. In order to show the accuracy of these methods, some test examples with square and circular domains are given.

4.2 Reduction of the Helmholtz equation to an integral equation

Let us consider the following non-homogeneous Helmholtz equation with variable coefficients for a two-dimensional body Ω . In the direct problem formulation, the acoustic pressure $\bar{u}(x)$ is prescribed on part $\partial_D\Omega$ of the boundary $\partial\Omega$ and the normal velocity $\bar{t}(x)$ on the remaining $\partial_N\Omega$ part of $\partial\Omega$, see [6, 39],

$$\sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[a(x) \frac{\partial u(x)}{\partial x_i} \right] + k(x)u(x) = f(x), \quad x \in \Omega, \quad (4.1)$$

with the mixed boundary conditions

$$u(x) = \bar{u}(x), \quad x \in \partial_D\Omega, \quad (4.2)$$

$$Tu(x) = \bar{t}(x), \quad x \in \partial_N\Omega, \quad (4.3)$$

where Ω is a bounded domain, $a(x)$ is a known variable material coefficient, $f(x)$ is a given function; $x = (x_1, x_2)$; $k(x)$ is a known variable wave number, $[Tu](x) := a(x)\frac{\partial u}{\partial n}(x)$, $n(x)$ is the external normal vector to the boundary $\partial\Omega$, and $\bar{u}(x)$ and $\bar{t}(x)$ are known functions. The non-homogeneous Helmholtz problem (4.1)-(4.3) becomes a pure Neumann problem if $\partial_D\Omega = \emptyset$, and a pure Dirichlet problem if $\partial_N\Omega = \emptyset$. We assume that $k(x)$ is not an eigenvalue for the homogeneous form of the mixed problem (4.1)-(4.3).

Eqs. (4.1)-(4.3) when both $a(x)$ and $k(x)$ are constant can be expressed as the integral equation (3.7) where the fundamental solution in this case is given by [6, 37],

$$F(x, y) = \frac{1}{4}Y_0(kr), \quad (4.4)$$

where Y_0 is the zero order Bessel function of the second kind.

For partial differential operators with variable coefficients, like L in Eq.(4.1) a fundamental solution is generally not available in explicit form. However, a parametrix is often available instead, which is discussed in detail in chapter 3. By following the same procedure, we obtain the integral equality

$$\begin{aligned} & c(y)u(y) - \int_{\partial\Omega} [u(x)T_xP(x, y) - P(x, y)Tu(x)]d\Gamma(x) + \\ & + \int_{\Omega} [R(x, y) + k(x)P(x, y)]u(x)d\Omega(x) = \int_{\Omega} P(x, y)f(x)d\Omega(x), \end{aligned} \quad (4.5)$$

where the parametrix $P(x, y)$ and $R(x, y)$ are given in Eqs.(3.9) and (3.10), respectively. Identity (4.5) can be used for formulating either a BDIE or a BDIDE, with respect to u and its derivatives. Let us consider both forms below.

4.3 Boundary-domain integral/integro-differential equations (BDIE/BDIDE)

Boundary-domain integral equation (BDIE)

Substituting the boundary conditions (4.2) and (4.3) into (4.5), introducing a new variable $t(x)=Tu(x)$ for the unknown normal velocity on $\partial_D\Omega$ and using Eq.(4.5) at $y \in \Omega \cup \partial\Omega$ reduces the BVP (4.1)-(4.3) to the following BDIE for $u(x)$ at $x \in \Omega \cup \partial_N\Omega$ and $t(x)$ at

$x \in \partial_D \Omega$,

$$\begin{aligned} c^0(y)u(y) - \int_{\partial_N \Omega} u(x)T_x P(x, y)d\Gamma(x) + \int_{\partial_D \Omega} P(x, y)t(x)d\Gamma(x) + \\ + \int_{\Omega} [R(x, y) + k(x)P(x, y)]u(x)d\Omega(x) = \Psi^0(y), \quad y \in \Omega \cup \partial\Omega \end{aligned} \quad (4.6)$$

where $\Psi^0(y)$ is given by Eq. (3.24).

Boundary-domain integro-differential equation (BDIDE)

Using another approach, we can substitute the boundary conditions (4.2) and (4.3) into (4.5) but leave T as a differential operator acting on u on the Dirichlet boundary $\partial_D \Omega$ and use the following BDIDE at $y \in \Omega \cup \partial_N \Omega$,

$$\begin{aligned} a(y)c(y)u(y) - \int_{\partial_N \Omega} u(x)T_x P(x, y)d\Gamma(x) + \int_{\partial_D \Omega} P(x, y)Tu(x)d\Gamma(x) + \\ + \int_{\Omega} [R(x, y) + k(x)P(x, y)]u(x)d\Omega(x) = \Psi(y), \quad y \in \Omega \cup \partial_N \Omega, \end{aligned} \quad (4.7)$$

where $\Psi(y)$ is given by Eq.(3.25). As we will see below, this approach can lead, after discretisation, to a system with a reduced number of linear algebraic equations.

4.4 Discretisation of the BDIE/BDIDE

By following the same procedure in chapter 3, and using the interpolations of $u(x)$ and $t(x)$ in Eqs.(3.28) and (3.42), we can obtain a system of linear algebraic equations by the collocation method for equations (4.6) and (4.7).

4.4.1 Discretisation of the BDIE

Substituting the interpolations (3.28) and (3.42) in BDIE (4.6) and applying the collocation method, we arrive at the following system of J linear algebraic equations for J

unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N \Omega$ and $t(x^j) = (Tu)(x^j)$, $x^j \in \partial_D \Omega$,

$$\begin{aligned} & c^0(x^i)u(x^i) + \sum_{x^j \in \Omega \cup \partial_N \Omega} M_{ij}u(x^j) + \sum_{x^j \in \Omega \cup \partial_N \Omega} DK_{ij}u(x^j) + \\ & + \sum_{x^j \in \partial_D \Omega} M'_{ij}t(x^j) = \Psi^0(x^i) - \sum_{x^j \in \partial_D \Omega} M_{ij}\bar{u}(x^j) - \sum_{x^j \in \partial_D \Omega} DK_{ij}\bar{u}(x^j), \\ & x^i \in \Omega \cup \partial \Omega, \quad i = 1, \dots, J, \text{ no sum in } i. \end{aligned} \quad (4.8)$$

where $\Psi^0(x^i)$ is calculated from Eq.(3.24), and

$$\begin{aligned} \Psi(x^i) = & \int_{\bar{\omega}(x^i) \cap \partial_D \Omega} \bar{u}(x)T_x P(x, x^i)d\Gamma(x) - \int_{\bar{\omega}(x^i) \cap \partial_N \Omega} P(x, x^i)\bar{t}(x)d\Gamma(x) + \\ & + \int_{\Omega} P(x, x^i)f(x)d\Omega(x), \end{aligned} \quad (4.9)$$

$$M_{ij} = - \int_{\bar{\omega}(x^i) \cap \partial_N \Omega} \Phi_j(x)T_x P(x, x^i)d\Gamma(x) \quad (4.10)$$

$$M'_{ij} = \int_{\bar{\omega}(x^i) \cap \partial_D \Omega} P(x, x^i)v_j(x)d\Gamma(x), \quad (4.11)$$

$$DK_{ij} = \int_{\omega(x^i) \cap \Omega} \Phi_j(x)[R(x, x^i) + k(x)P(x, x^i)]d\Omega(x). \quad (4.12)$$

4.4.2 Discretisation of the BDIDE

To obtain a system of linear algebraic equations from the BDIDE (4.7) by the collocation method, we collocate at the nodes x^i , $i = 1, \dots, J$, arriving at a system of $J - J_D$ algebraic equations for $J - J_D$ unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N \Omega$. Substituting interpolation formulae (3.28) into the BDIDE (4.7) leads to the following system:

$$\begin{aligned} & c(x^i)u(x^i) + \sum_{x^j \in \Omega \cup \partial_N \Omega} M''_{ij}u(x^j) + \sum_{x^j \in \Omega \cup \partial_N \Omega} DK_{ij}u(x^j) = \Psi(x^i) \\ & - \sum_{x^j \in \partial_D \Omega} M''_{ij}\bar{u}(x^j) - \sum_{x^j \in \partial_D \Omega} DK_{ij}\bar{u}(x^j), \quad x^i \in \Omega \cup \partial_N \Omega, \quad \text{no sum in } i, \end{aligned} \quad (4.13)$$

where $\Psi(x^i)$ and DK_{ij} are given by Eqs.(4.9) and (4.12), respectively, and

$$M''_{ij} = \int_{\bar{\omega}(x^i) \cap \partial_D \Omega} P(x, x^i) T \Phi_j(x) d\Gamma(x) - \int_{\bar{\omega}(x^i) \cap \partial_N \Omega} \Phi_j(x) T_x P(x, x^i) d\Gamma(x). \quad (4.14)$$

4.5 Numerical results

In this section, we shall examine some test examples to assess the performance of the BDIE and BDIDE formulations for the non-homogeneous Helmholtz equation for four cases. Firstly, when both material parameter $a(x)$ and wave number k are constant. Secondly, when the material parameter $a(x)$ is variable and the wave number k is constant. Thirdly, when the material parameter $a(x)$ is constant and the wave number $k(x)$ is variable. Fourthly, when both the material parameter and wave number are variable. We applied the BDIE and BDIDE methods to some test problems on square and circular domains, for which an exact analytical solution, u_{exact} , is available. Also, the relative error was calculated as in chapter 2, Eq.(2.35). The relative error has been calculated for $J= 25, 81, 289$ and 1089 in all test examples.

4.5.1 Numerical results when both $a(x)$ and $k(x)$ constant

Numerical results for homogeneous Helmholtz equation

We shall initially test the BDIE and BDIDE formulations for the homogeneous Helmholtz equation, i.e. $f(x) = 0$. In this case, both BDIE and BDIDE reduce to BIE and BIDE with no domain integrals.

Test 1 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 0 \leq x_1, x_2 \leq 1\}$, $k = \frac{\sqrt{2}\pi}{4}$, for $x \in \bar{\Omega}$. The exact solution for this problem is $u_{exact}(x) = \sin(\frac{\pi x_1}{4}) \cos(\frac{\pi x_2}{4})$, $x \in \bar{\Omega}$.

Test 2 :

Circular domain $\bar{\Omega} = \{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}$, $k = \frac{\sqrt{2}\pi}{4}$, for $x \in \bar{\Omega}$. The exact solution for this problem is $u_{exact}(x) = \sin(\frac{\pi x_1}{4}) \cos(\frac{\pi x_2}{4})$, $x \in \bar{\Omega}$.

Both tests will be run with the following boundary conditions:

- **Dirichlet boundary conditions:**

$$\bar{u}(x) = \sin\left(\frac{\pi x_1}{4}\right) \cos\left(\frac{\pi x_2}{4}\right), x \in \bar{\Omega}.$$

•• Neumann boundary conditions:

$$\bar{t}(x) = \frac{1}{4} \cos\left(\frac{1}{4}\pi x_1\right)\pi \cos\left(\frac{1}{4}\pi x_2\right)n_1(x) - \frac{1}{4} \sin\left(\frac{1}{4}\pi x_1\right)\pi \sin\left(\frac{1}{4}\pi x_2\right)n_2(x).$$

••• Mixed boundary conditions:

$$\bar{u}(x) = \sin\left(\frac{\pi x_1}{4}\right) \cos\left(\frac{\pi x_2}{4}\right) \text{ for } \{x_2 = 0; x_2 = 1; 0 \leq x_1 \leq 1\}; \text{ or } x_2 = \sqrt{1 - x_1^2}.$$

$$\bar{t}(x) = \frac{1}{4} \cos\left(\frac{1}{4}\pi x_1\right)\pi \cos\left(\frac{1}{4}\pi x_2\right)n_1(x) - \frac{1}{4} \sin\left(\frac{1}{4}\pi x_1\right)\pi \sin\left(\frac{1}{4}\pi x_2\right)n_2(x) \text{ for } \{x_1 = 0; x_1 = 1; 0 \leq x_2 \leq 1\}; \text{ or } x_2 = -\sqrt{1 - x_1^2}.$$

The algorithm was implemented in computer programs by using Matlab. The surface plots of the numerical solutions were obtained with the most refined mesh in each example. The graph of relative error has the number of nodes on the horizontal axis and the relative error on the vertical axis.

Dirichlet problem

Boundary Integro-Differential Equation Method

The BIDE formulation for the Dirichlet problem only calculates the values of the function u at internal points as the normal derivative $\frac{\partial u}{\partial n}$ is approximated through linear basis functions living on triangles. The main numerical errors come from the resulting domain discretisation and the numerical evaluation of the boundary integrals. Good results and rates of convergence are obtained for both tests, as can be seen in Figures 4.1 and 4.2. It can also be seen that the results for test 2 have lower order of accuracy as there is an extra error coming from approximating the circular boundary by polygons.

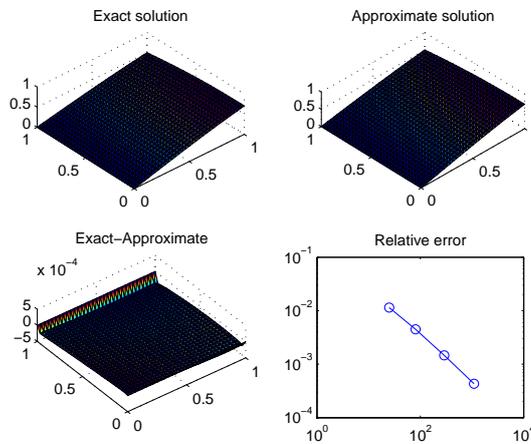


Figure 4.1: Relative error for test 1, when $J=1089$, $r(1089) \approx 4.00 \times 10^{-4}$

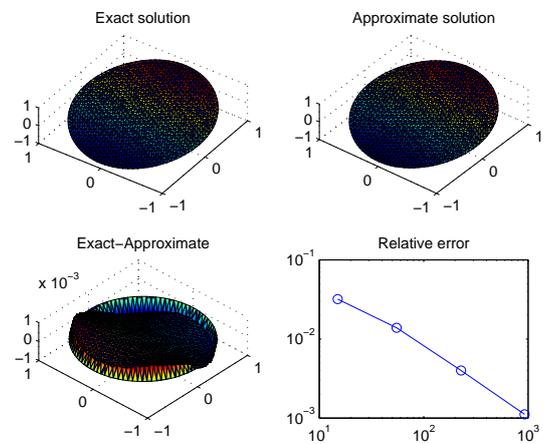


Figure 4.2: Relative error for test 2, when $J=925$, $r(925) \approx 1.10 \times 10^{-3}$

Boundary Integral Equation Method

The BIE method for the homogeneous Helmholtz equation reduces to the standard BEM. The normal derivative $\frac{\partial u}{\partial n}$ is assumed to be constant within each boundary element to take into account its discontinuities at corner points. The results for tests 1 and 2, presented in Figures 4.3 and 4.4, respectively, show that the BIE method produces more accurate results than the BIDE method for Dirichlet problems.

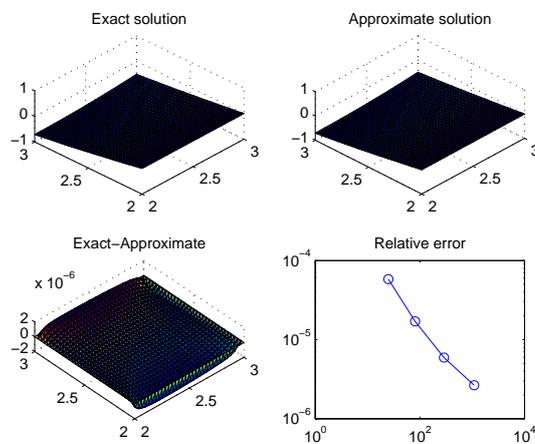


Figure 4.3: Relative error for test 1, when $J=1089$, $r(1089) \approx 2.64 \times 10^{-6}$

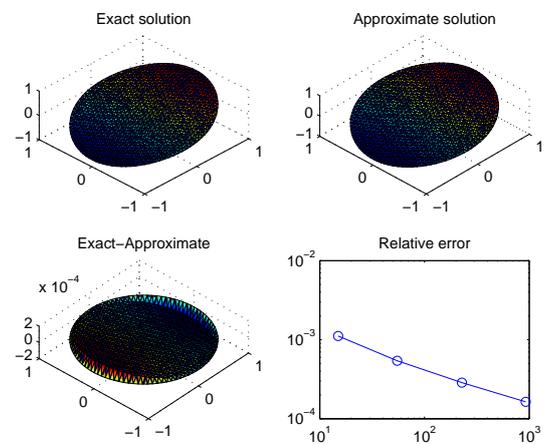


Figure 4.4: Relative error for test 2, when $J=925$, $r(925) \approx 2.00 \times 10^{-4}$

Neumann problem

Only the BIE formulation is applicable for Neumann problems. Tests 1 and 2 with Neumann boundary conditions are solved using linear interpolation for the function u along each boundary element. The results for both tests in Figures 4.5 and 4.6, respectively, demonstrate the convergence of the solutions.

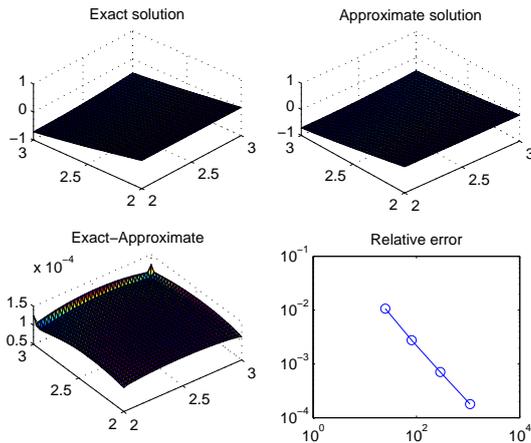


Figure 4.5: Relative error for test 1, when $J=1089$, $r(1089) \approx 2.00 \times 10^{-4}$

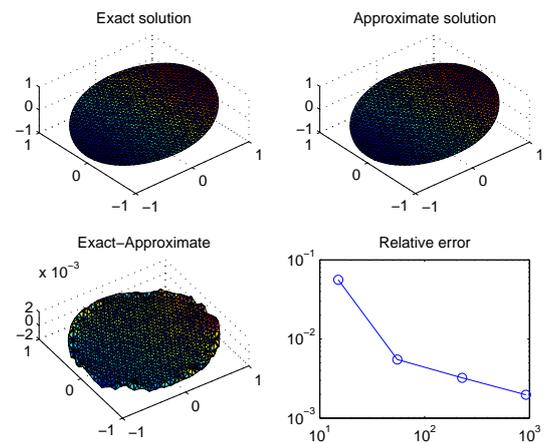


Figure 4.6: Relative error for test 2, when $J=925$, $r(925) \approx 2.00 \times 10^{-3}$

Mixed problem

Boundary Integro-Differential Equation Method

The BIDE formulation for mixed problems does not employ collocation points on the Dirichlet boundaries; therefore, unlike the standard BEM, the values of t on Dirichlet boundaries are not calculated. The main numerical errors come from the use of linear basis functions living on triangles for approximating the values of t , the resulting domain discretisation and the numerical evaluation of the boundary and domain integrals. Good results and rates of convergence are obtained for both tests, as can be seen in Figures 4.7 and 4.8.

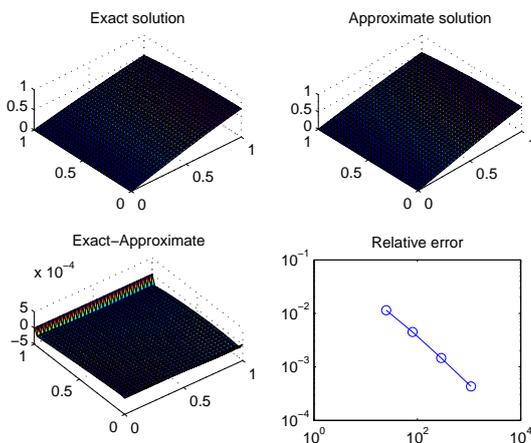


Figure 4.7: Relative error for test 1, when $J=1089$, $r(1089) \approx 4.00 \times 10^{-4}$

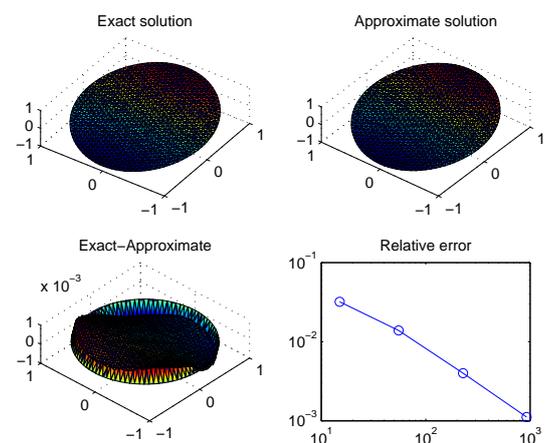


Figure 4.8: Relative error for test 2, when $J=925$, $r(925) \approx 1.10 \times 10^{-3}$

Boundary Integral Equation Method

The treatment of corner points in the BIE method for two-dimensional mixed BVP for a second-order linear elliptic PDE with variable coefficients, by using mixed boundary elements with linear interpolation for u and constant for t , with mid-node or end-node collocation (M-NC or E-NC), is discussed in detail in chapter three. Figures 4.9 and 4.10 below show the results for tests 1 and 2 with mixed boundary conditions. Once more, the BIE method produced more accurate results than the BDIE method.

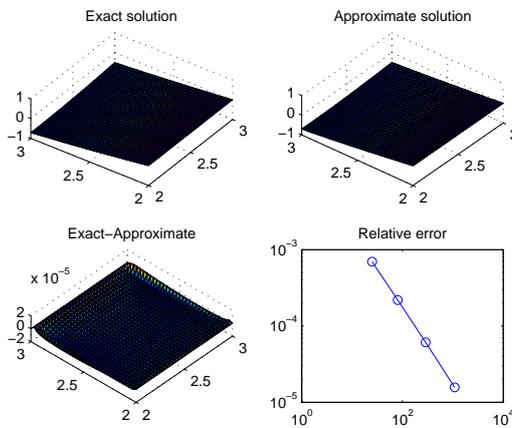


Figure 4.9: Relative error for test 1 with E-NC; when $J=1089$, $r(1089) \approx 1.57 \times 10^{-5}$

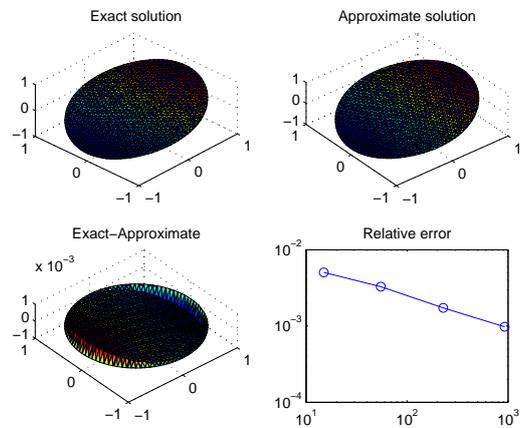


Figure 4.10: Relative error for test 2 with M-NC; when $J=925$, $r(925) \approx 1.00 \times 10^{-3}$

Numerical results for the non-homogeneous Helmholtz equation

We shall test the BDIE and BDIDE formulations for the non-homogeneous Helmholtz equation, i.e. $f(x) \neq 0$. Initially, all tests will assume $k = 1$. For the domain integrals which appear in the non-homogeneous Helmholtz equation, we have implemented the cell-integration technique exactly like when treating the heat source domain integral in chapter three.

In the cell-integration technique, the weak singularities which appear when using triangular elements (where collocation is at the vertex of the integration element) have been solved by using Gaussian quadrature rules for two dimensions (with eight points in each direction), in addition to Duffy transformation, by mapping the triangles into squares and eliminating the weak singularity in the process for each of the three vertices of the integration element. However, for elements with no collocation points, Gaussian quadrature

rules for two dimensions are enough to achieve the desired accuracy.

Test 1 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 0 \leq x_1, x_2 \leq 1\}$, for $x \in \bar{\Omega}$, $f(x) = x_1 + x_2$ and the boundary conditions: (4.2) is $\bar{u}(x) = x_1 + x_2$, (4.3) is $\bar{t}(x) = n_1(x) + n_2(x)$ and for mixed boundary conditions (4.2), (4.3), we have (4.2) for $\{x_2 = 0; x_2 = 1; 0 \leq x_1 \leq 1\}$; and (4.3) for $\{x_1 = 0; x_1 = 1; 0 \leq x_2 \leq 1\}$. The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

Test 2:

Square domain $\bar{\Omega} = \{(x_1, x_2) : 0 \leq x_1, x_2 \leq 1\}$, for $x \in \bar{\Omega}$, $f(x) = 4 + x_1^2 + x_2^2$ and the boundary conditions: (4.2) is $\bar{u}(x) = x_1^2 + x_2^2$, (4.3) is $\bar{t}(x) = 2x_1n_1(x) + 2x_2n_2(x)$ and for mixed boundary conditions (4.2), (4.3), we have (4.2) for $\{x_2 = 0; x_2 = 1; 0 \leq x_1 \leq 1\}$; and (4.3) for $\{x_1 = 0; x_1 = 1; 0 \leq x_2 \leq 1\}$. The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

Test 3 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 0 \leq x_1, x_2 \leq 1\}$, for $x \in \bar{\Omega}$, $f(x) = -\cos(x_1)\sin(x_2)$ and the boundary conditions: (4.2) is $\bar{u}(x) = \cos(x_1)\sin(x_2)$, (4.3) is $\bar{t}(x) = -\sin(x_1)\sin(x_2)n_1(x) + \cos(x_1)\cos(x_2)n_2(x)$ and for mixed boundary conditions (4.2), (4.3), we have (4.2) for $\{x_2 = 0; x_2 = 1; 0 \leq x_1 \leq 1\}$; and (4.3) for $\{x_1 = 0; x_1 = 1; 0 \leq x_2 \leq 1\}$. The exact solution for this problem is $u_{exact}(x) = \cos(x_1)\sin(x_2)$, $x \in \bar{\Omega}$.

Test 4 :

Circular domain $\bar{\Omega} = \{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}$, for $x \in \bar{\Omega}$, $f(x) = -\cos(x_1)\sin(x_2)$. and the boundary conditions: (4.2) is $\bar{u}(x) = \cos(x_1)\sin(x_2)$, (4.3) is $\bar{t}(x) = -\sin(x_1)\sin(x_2)n_1(x) + \cos(x_1)\cos(x_2)n_2(x)$ and for mixed boundary conditions (4.2), (4.3), we have (4.2) for $x_2 = \sqrt{1 - x_1^2}$; and (4.3) for $x_2 = -\sqrt{1 - x_1^2}$. The exact solution for this problem is $u_{exact}(x) = \cos(x_1)\sin(x_2)$, $x \in \bar{\Omega}$.

The Matlab programs for homogeneous problems have been modified by adding the calculation of the domain integral for the new tests 1-4 above.

Dirichlet problem

The Boundary-Domain Integro-Differential Equation Method

In test 1 the exact solution is linear and we are using linear basis functions living on triangles, therefore, there is no interpolation error. So, the errors come either from discretisation of the domain into triangles or from calculating the boundary and domain integrals numerically. However, for test 2, the exact solution is quadratic, therefore, there is interpolation error in addition to the other numerical errors in test 1. The results for both tests in Figures 4.11 and 4.12 are good, with satisfactory convergence.

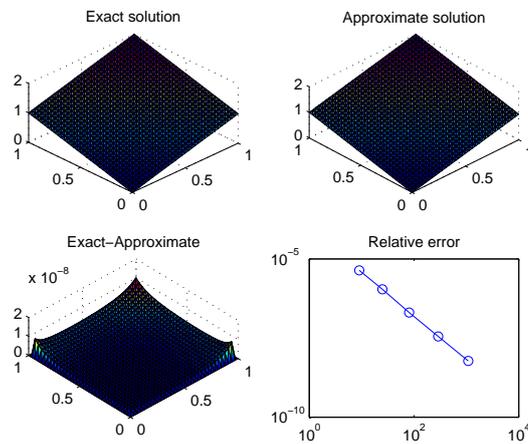


Figure 4.11: Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-9}$

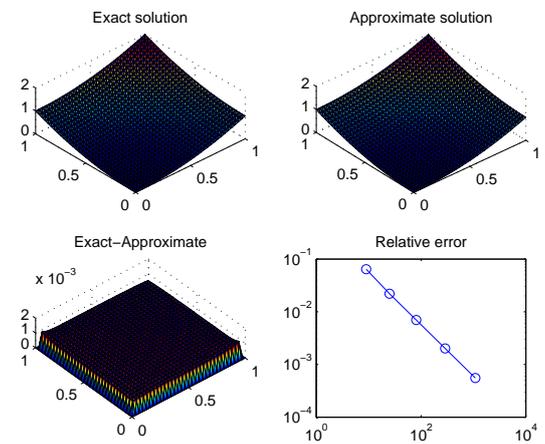


Figure 4.12: Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$

The exact solutions for tests 3 and 4 are trigonometric functions, therefore, interpolation errors are present for both tests. Also, there are discretisation and numerical integration errors, and an extra error has been added for test 4 which comes from approximating the boundary curve by polygons, but the results are still good with satisfactory results and convergence, as can be seen in Figures 4.13 and 4.14.

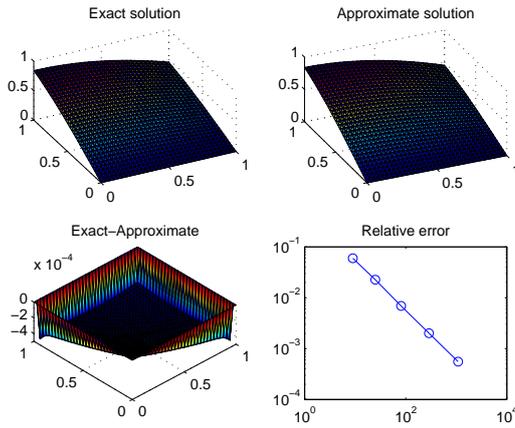


Figure 4.13: Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$

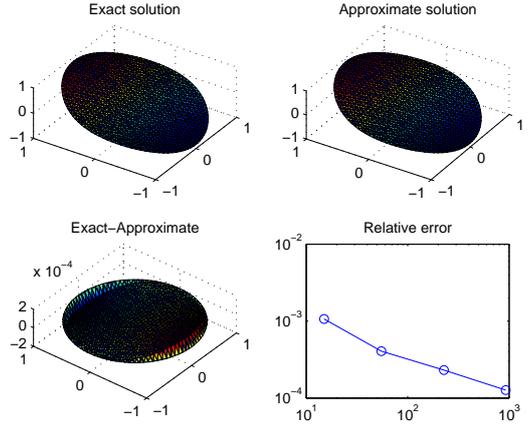


Figure 4.14: Surface plot of solution for test 4, when $J=925$, $r(925) \approx 1.00 \times 10^{-4}$

The Boundary-Domain Integral Equation Method

We will now test the BDIE for tests 1-4, using the same procedure as for the homogenous case, in which t is assumed to be constant on the boundary to deal with the discontinuity of the normal derivative at corner points.

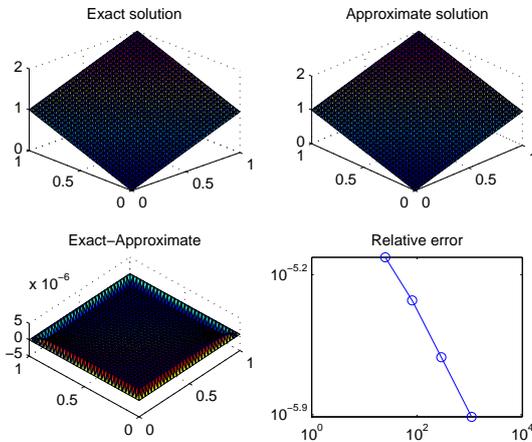


Figure 4.15: Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 1.23 \times 10^{-6}$

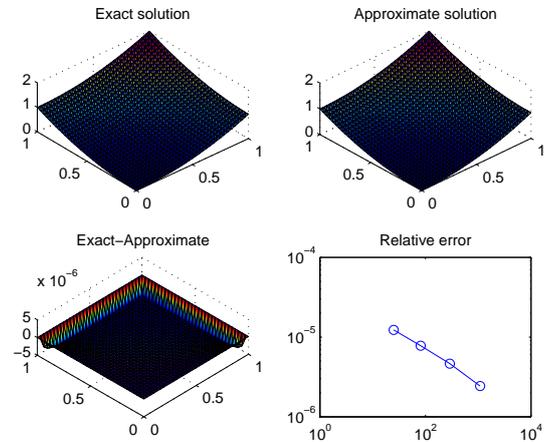


Figure 4.16: Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 2.43 \times 10^{-6}$

The results in Figures 4.15-4.18 demonstrate that the BIE method is able to generate accurate solutions with high rates of convergence for the BVP (4.1)-(4.2). When comparing the solutions obtained by using BIE/BIDE methods, it can be seen that the BIDE method produced better results for test 1. However, more accurate results were obtained for the BIE method for tests 2, 3 and 4. A possible explanation is the approximation of the flux t in the BIDE method using linear basis functions for u living on triangles; thus, $T\Phi_j(x)$

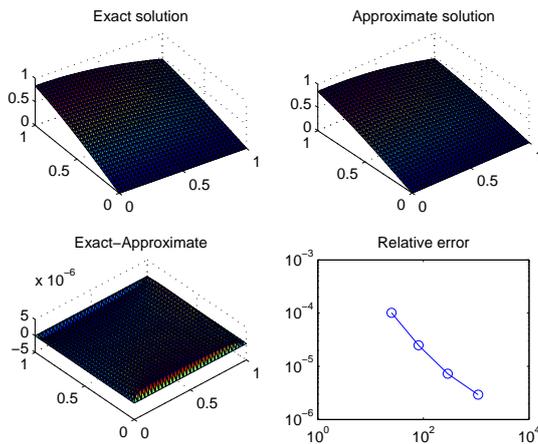


Figure 4.17: Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 2.90 \times 10^{-6}$

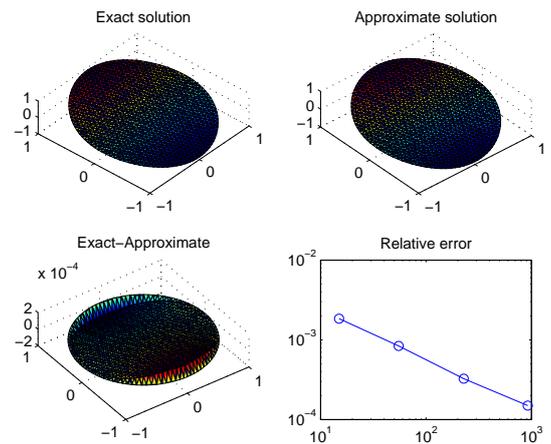


Figure 4.18: Surface plot of solution for test 4, when $J=925$, $r(925) \approx 1.00 \times 10^{-4}$

is constant within each triangle. These approximations are appropriate for test 1, as the solution is linear, while the solution to the tests 2, 3 and 4 are not linear. The accuracy of the BIE for tests 1, 2 and 3 is also reduced by the approximation of the flux t at the boundary nodes, as there is a slight flux discontinuity at these points which is avoided in the BIDE method.

Neumann problem

The solution of the Neumann problem will provide values of u on the boundary and domain (different from Dirichlet problems, where the values of u on the boundary are specified), therefore, lower accuracy should be expected.

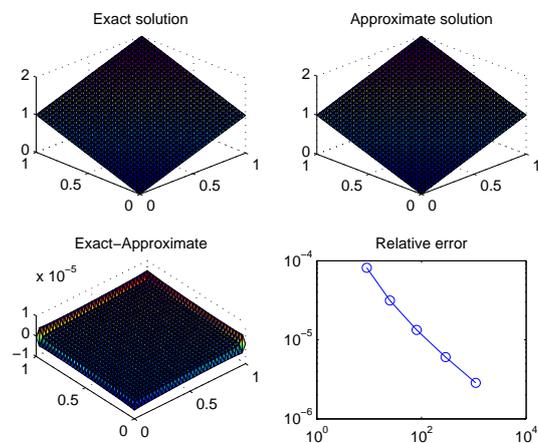


Figure 4.19: Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 2.85 \times 10^{-6}$

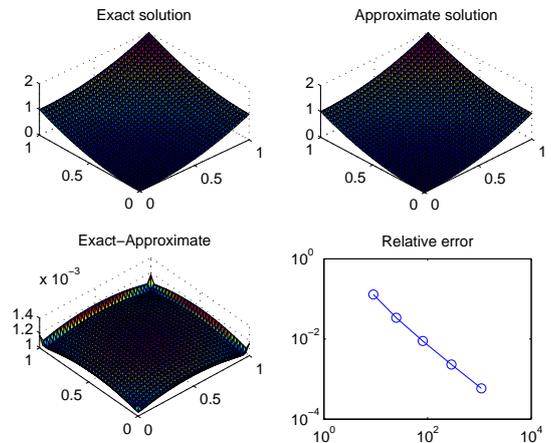


Figure 4.20: Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$

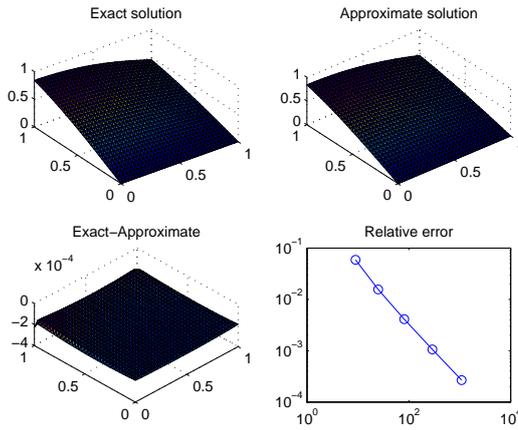


Figure 4.21: Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 3.00 \times 10^{-4}$

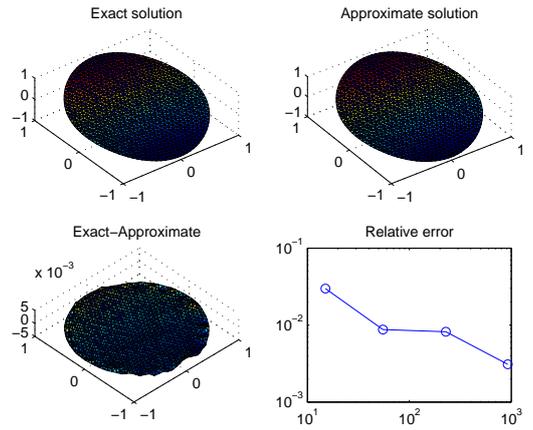


Figure 4.22: Surface plot of solution for test 4, when $J=925$, $r(925) \approx 3.10 \times 10^{-3}$

Mixed problem

The Boundary-Domain Integro-Differential Equation Method

BDIDE will be tested for tests 1-4 with mixed boundary conditions. It is important to notice that there are no collocation points on Dirichlet boundaries, therefore, the only unknown is u along Neumann boundaries. Thus, the problem caused by the discontinuity of the normal derivative at corner points is avoided in all tests.

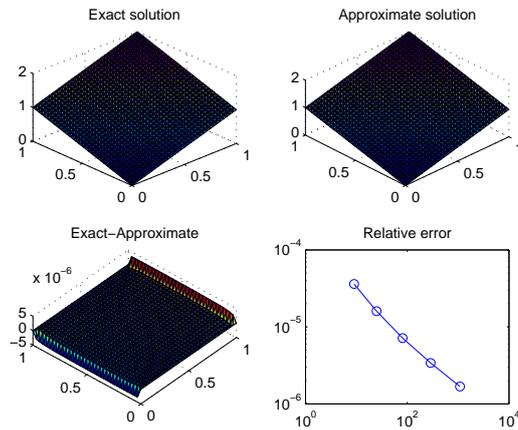


Figure 4.23: Surface plot of solution for test 1, when $J=1089$, $r(1089) \approx 1.68 \times 10^{-6}$

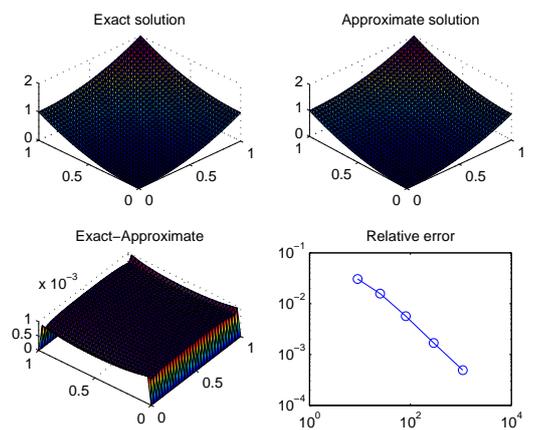


Figure 4.24: Surface plot of solution for test 2, when $J=1089$, $r(1089) \approx 5.00 \times 10^{-4}$

For tests 2, 3 and 4, lower accuracy is obtained in comparison with test 1. A possible explanation is the approximation of the flux t in the BIDE method using linear basis functions for u living on triangles; thus, $T\Phi_j(x)$ is constant within each triangle. These

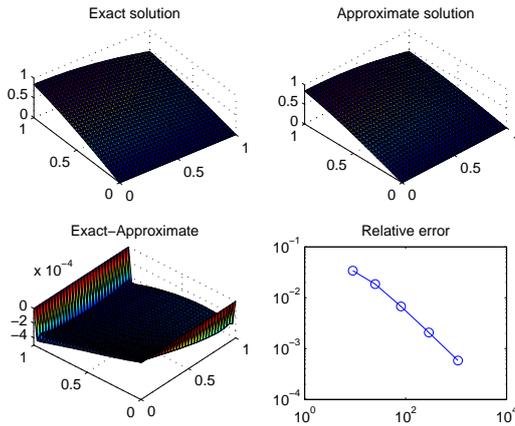


Figure 4.25: Surface plot of solution for test 3, when $J=1089$, $r(1089) \approx 6.00 \times 10^{-4}$

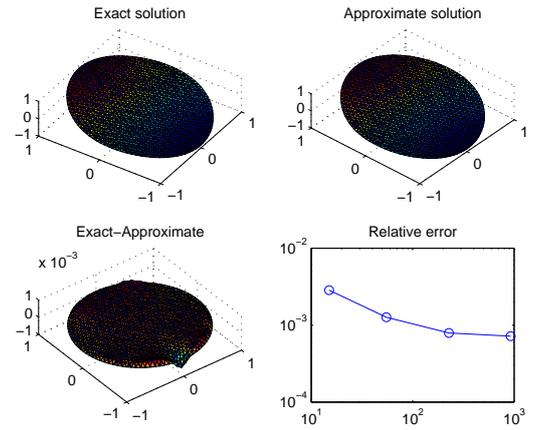


Figure 4.26: Surface plot of solution for test 4, when $J=925$, $r(925) \approx 7.00 \times 10^{-4}$

approximations are appropriate for test 1, as the solution to this test is linear, while the solution to tests 2, 3 and 4 are quadratic and trigonometric functions.

The Boundary-Domain Integral Equation Method

Finally, BDIE will be tested with mixed boundary conditions. In the BDIE formulation, both u and t along the boundary are calculated. We implemented mixed boundary elements with linear u and constant t to avoid the discontinuities of t at corner points. In this case, collocation was tested at the mid and end points of each boundary element. It was shown in chapter three that end-node collocation generally provides higher accuracy than mid-node collocation. Therefore, the end-node collocation is adopted for the square domain in tests 1-3, while mid-node collocation is adopted for the circular domain as it gives slightly better results than end-node collocation as shown in chapter three (using mid-node collocation has an advantage for curved domains as the value of c is always $\frac{1}{2}$). Good rates of convergence are obtained for all tests, as can be seen in Figures 4.27-4.30.

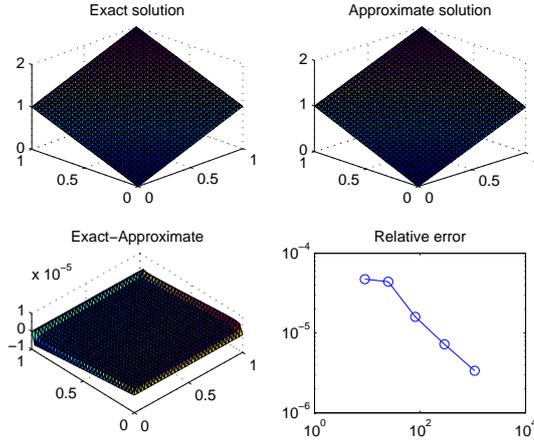


Figure 4.27: Surface plot of solution for test 1 with E-NC, $r(1089) \approx 3.36 \times 10^{-6}$

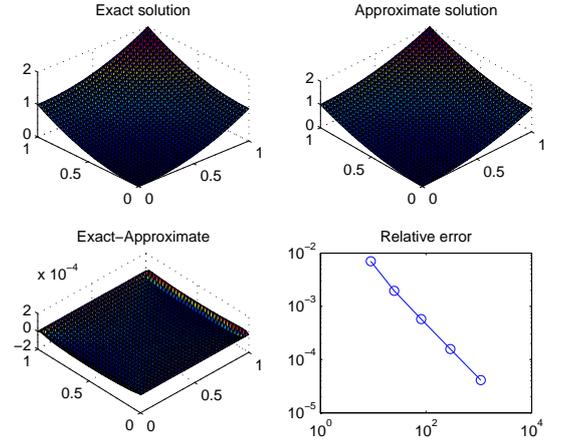


Figure 4.28: Surface plot of solution for test 2 with E-NC, $r(1089) \approx 4.11 \times 10^{-5}$

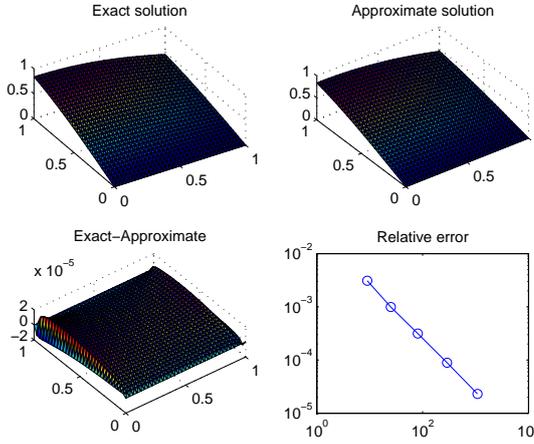


Figure 4.29: Surface plot of solution for test 3 with E-NC, $r(1089) \approx 2.30 \times 10^{-5}$

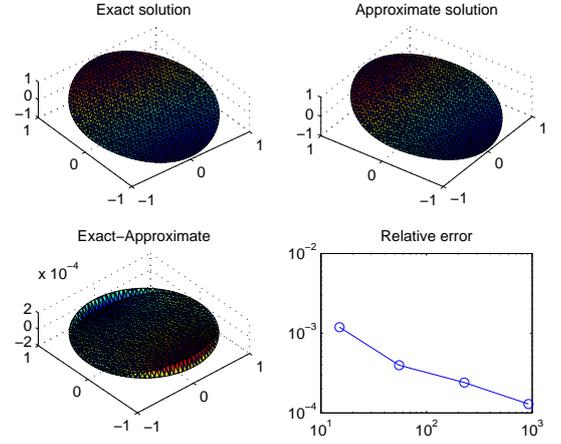


Figure 4.30: Surface plot of solution for test 4 with M-NC, $r(925) \approx 1.00 \times 10^{-4}$

4.5.2 BDIE/BDIDE for higher wave numbers

In this section, we assess the performance of the BDIE and BDIDE methods for the non-homogeneous Helmholtz equation with higher wave numbers. It is well known that the proper resolution of wave propagation and scattering problems for high wave numbers requires the use of fine meshes [41]. Let us resolve Test 1 in page 66, with mixed boundary conditions using BDIE and BDIDE with $k > 1$, where: $f(x) = k^2(x_1 + x_2)$, the exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

It can be clearly seen from Figures 4.31 and 4.32 that lower accuracy is obtained as k is increased. This behaviour is similar to other methods such as the standard BEM, FEM and FDM for solving the Helmholtz equation.

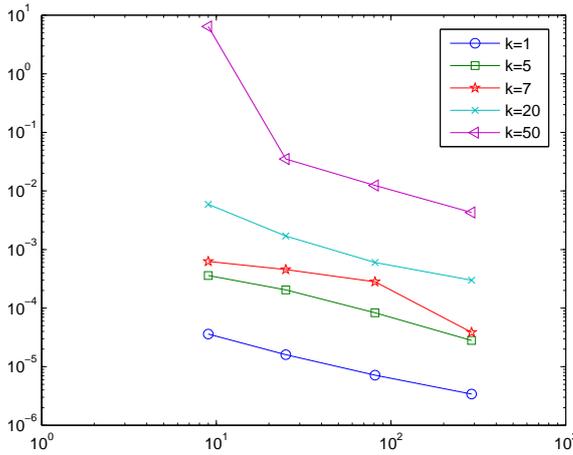


Figure 4.31: Relative errors for test 1 using BDIDE

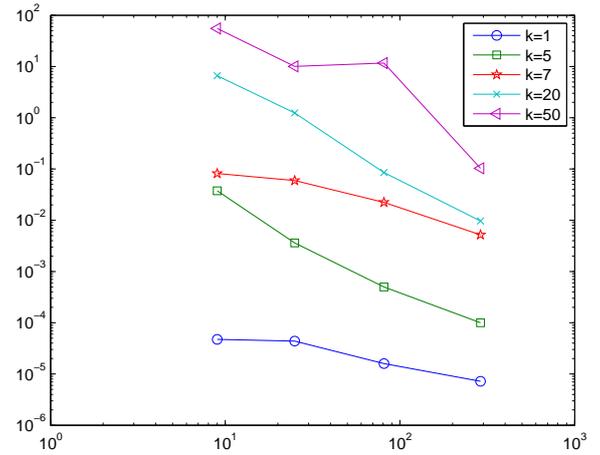


Figure 4.32: Relative errors for test 1 using BDIE

4.6 Numerical results for variable coefficients

In this section, we shall examine some test examples to assess the performance of the BDIE/ BDIDE formulations with mixed boundary conditions for three possible cases, when $a(x)$ variable and $k(x)$ constant, when $a(x)$ constant and $k(x)$ variable and when both $a(x)$ and $k(x)$ variable. To verify the convergence of the methods, we applied them to some test problems on square and circular domains, for which an exact analytical solution, u_{exact} , is available. Also, the relative error was calculated for fixed $J=9, 25, 81, 289$ and 1089 as in chapter two, Eq.(2.35). For the square domain, the top and bottom sides have known acoustic pressure u (Dirichlet boundary condition), while the left and right are imposed with normal velocity t (Neumann boundary condition). In order to compute the acoustic pressure $u(x)$ along some lines of the plates, the total number of nodes is fixed to 81 (32 on the boundary plus 49 in interior).

For the circular domain, the upper-half semi-circle has known acoustic pressure u , while the lower-half semi-circle has known normal velocity t . In order to compute the acoustic pressure $u(x)$ for some selected interior points, the total number of nodes is fixed to 55 (23 on the boundary plus 32 in interior) for test 2 and 75 (27 on the boundary plus 48 in interior) for test 5.

4.6.1 Numerical results when $a(x)$ variable and $k(x)$ constant

In this case, when the material parameter $a(x)$ is variable and the wave number $k(x)$ is constant, the parametrization in Eq.(3.9) is adopted.

Test 1 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, where $k(x) = 1$, for $x \in \bar{\Omega}$, $a(x) = 2(x_1 + x_2)$, $f(x) = 4 + x_1 + x_2$ and the boundary conditions:

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 3 + x_1, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = 2(x_1 + x_2)(n_1(x) + n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

The first test example to be considered is a square domain. Table 4.1 lists the computed acoustic pressure $u(x)$ along the line $x_2 = 2.5$ of the plate using BDIDE and BDIE, and Figures 4.33 and 4.34 represent the plot of the results and relative errors, respectively.

Table 4.1: Computed acoustic pressure along line $x_2 = 2.5$

x_1	BDIDE	BDIE	Exact
2	4.50001261	4.50001472	4.50000000
2.125	4.62500159	4.62500359	4.62500000
2.25	4.75000069	4.75000233	4.75000000
2.375	4.87500002	4.87500117	4.87500000
2.5	4.99999943	5.00000008	5.00000000
2.625	5.12499878	5.12499896	5.12500000
2.750	5.24999796	5.24999770	5.25000000
2.875	5.37499684	5.37499625	5.37500000
3	5.49998557	5.49998486	5.50000000

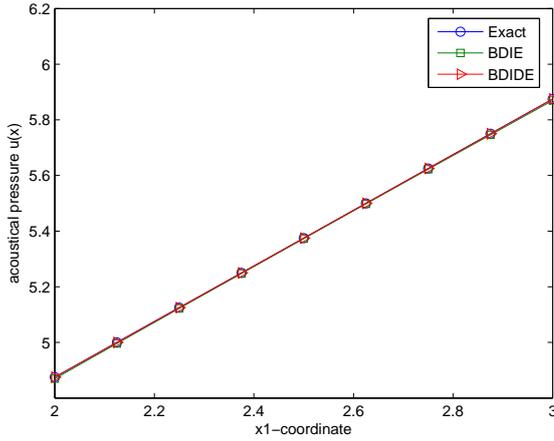


Figure 4.33: Acoustic pressure distribution along the line $x_2 = 2.875$ in test 1

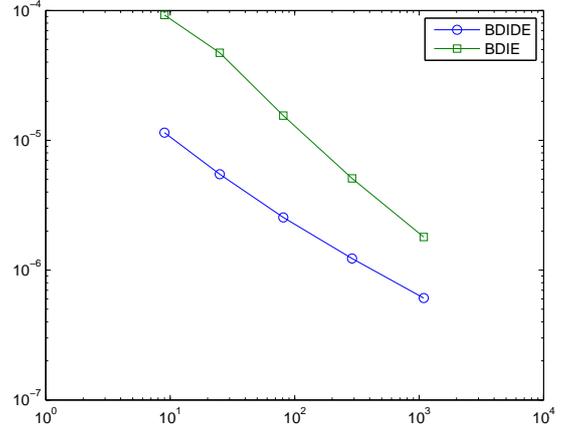


Figure 4.34: Relative error for BDIDE and BDIE for test 1

Test 2 :

Circular domain $\bar{\Omega} = \{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}$, where $k(x) = 1$, for $x \in \bar{\Omega}$, $a(x) = x_1 + x_2 + 4$, $f(x) = 2 + x_1 + x_2$ and the boundary conditions:

$$\bar{u}(x) = x_1 + x_2, \text{ for } x_2 = \sqrt{1 - x_1^2},$$

$$\bar{t}(x) = (x_1 + x_2 + 4)(n_1(x) + n_2(x)), \text{ for } x_2 = -\sqrt{1 - x_1^2}.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2, x \in \bar{\Omega}$.

The second test example to be considered is a circular domain. Table 4.2 lists the computed acoustic pressure $u(x)$ for some selected interior points using BDIDE and BDIE, and Fig.4.35 represents the plot of the relative errors.

Table 4.2: Computed acoustic pressure for some selected interior points

(x_1, x_2)	BDIDE	BDIE	Exact
(0.00000000, -0.72045733)	-0.72040690	-0.72032692	-0.72045733
(0.13034379, -0.42081664)	-0.29045666	-0.29052234	-0.29047285
(0.12923797, 0.15630183)	0.28555418	0.28554375	0.28553981
(0.26820818, 0.68285186)	0.95109415	0.95116850	0.95106003
(0.00000001, 0.73368609)	0.73366884	0.73386992	0.73368609
(0.28400244, -0.66580702)	-0.38179397	-0.34945629	-0.38180459

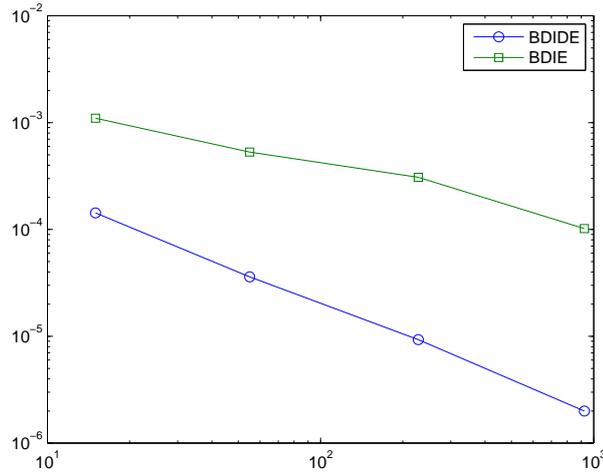


Figure 4.35: Relative error for BDIDE and BDIE for test 2

Test 3:

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, where $k(x) = 1$, for $x \in \bar{\Omega}$, $a(x) = x_1^2 + x_2^2$, $f(x) = 9(x_1^2 + x_2^2)$ and the boundary conditions:

$$\bar{u}(x) = 1 + x_1^2, \text{ for } x_2 = 1; \quad 1 \leq x_1 \leq 2,$$

$$\bar{u}(x) = 4 + x_1^2, \text{ for } x_2 = 2; \quad 1 \leq x_1 \leq 2,$$

$$\bar{t}(x) = 2(x_1^2 + x_2^2)(x_1 n_1(x) + x_2 n_2(x)), \text{ for } x_1 = 1 \text{ or } x_1 = 2; \quad 1 \leq x_2 \leq 2.$$

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

The third test example to be considered is a square domain. Table 4.3 lists the computed acoustic pressure $u(x)$ along the line $x_2 = 1.5$ of the plate using BDIDE and BDIE, and Figs. 4.36 and 4.37 represent the plot of the results and relative errors, respectively.

Table 4.3: Computed acoustic pressure along line $x_2 = 1.5$

x_1	BDIDE	BDIE	Exact
1	3.23904254	3.25001788	3.25000000
1.125	3.50524145	3.51695736	3.51562500
1.25	3.80173933	3.81393990	3.81250000
1.375	4.12971330	4.14223153	4.14062500
1.5	4.48916298	4.50185631	4.50000000
1.625	4.88007916	4.89283775	4.89062500
1.750	5.30243537	5.31520029	5.31250000
1.875	5.75620027	5.76895964	5.76562500
2	6.23998789	6.25274089	6.25000000

It can be seen from Tables 4.1-4.3 and Figs.4.33-4.37 that both the BDIE and BDIDE methods are able to generate accurate solutions with good convergence for non-homogeneous

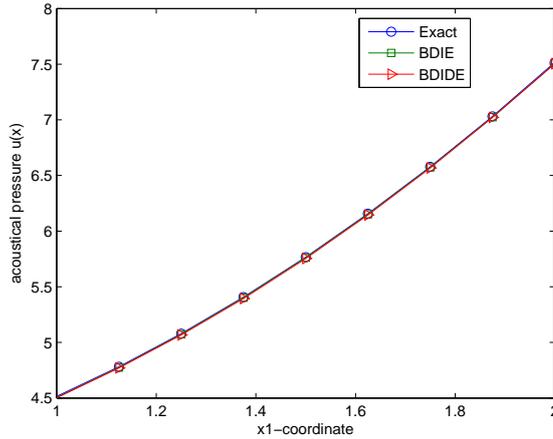


Figure 4.36: Acoustic pressure distribution along the line $x_2 = 1.875$ in test 3

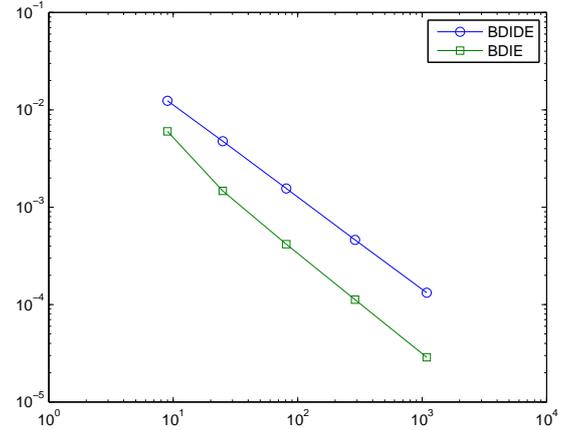


Figure 4.37: Relative error for BDIDE and BDIE for test 3

Helmholtz equations with variable material parameter $a(x)$ and constant wave number k , by using a parametrix to derive the formulations. The BDIDE formulation for mixed problems does not employ collocation points on the Dirichlet boundaries; therefore, unlike the standard BEM, the values of t on Dirichlet boundaries are not calculated. The treatment of corner points in the BDIE method, by using mixed boundary elements with linear interpolation for u and constant for t , with mid-node or end-node collocation, is discussed in detail in the previous chapter. We adopted the end-node collocation in BDIE as the results are much better than for mid-node collocation.

4.6.2 Numerical results when $a(x)$ constant and $k(x)$ variable

In this case, when the material parameter $a(x) = 1$ constant, the remainder $R(x, y)$ in Eq.(3.10) will be zero. The parametrix in Eq.(3.9) is exactly the same as the fundamental solution for the Laplace equation.

Test 4 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 0 \leq x_1, x_2 \leq 1\}$, where $k(x) = x_1^3 + x_2^3$, for $x \in \bar{\Omega}$, $a(x) = 1$, $f(x) = (x_1^3 + x_2^3)(x_1 + x_2)$ and the boundary conditions:

$$\bar{u}(x) = x_1, \text{ for } x_2 = 0; \quad 0 \leq x_1 \leq 1,$$

$$\bar{u}(x) = 1 + x_1, \text{ for } x_2 = 1; \quad 0 \leq x_1 \leq 1,$$

$$\bar{t}(x) = n_1(x) + n_2(x), \text{ for } x_1 = 0 \text{ or } x_1 = 1; \quad 0 \leq x_2 \leq 1.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

Table 4.4 lists the computed acoustic pressure $u(x)$ along the line $x_2 = 0.5$ of the plate using BDIDE and BDIE, and Figures 4.38 and 4.39 represent the plot of the results and relative errors, respectively.

Table 4.4: Computed acoustic pressure along line of $x_2 = 0.5$

x_1	BDIDE	BDIE	Exact
0	0.50001354	0.50001418	0.50000000
0.125	0.62500238	0.62500297	0.62500000
0.25	0.75000135	0.75000181	0.75000000
0.375	0.87500056	0.87500085	0.87500000
0.5	0.99999989	0.99999998	1.00000000
0.625	1.12499922	1.12499910	1.12500000
0.750	1.24999839	1.24999810	1.25000000
0.875	1.37499729	1.37499688	1.37500000
1	1.49998607	1.49998561	1.50000000

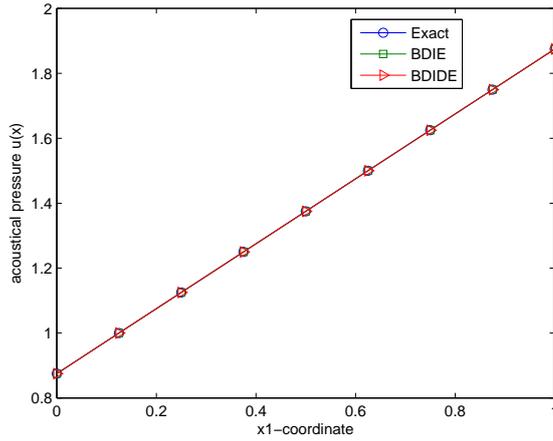


Figure 4.38: Acoustic pressure distribution along the line $x_2 = 0.875$ in test 4

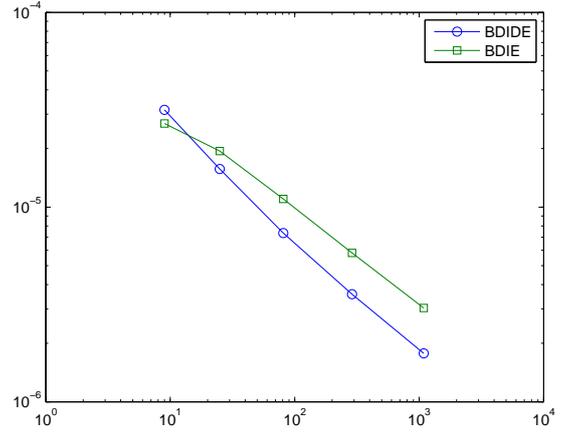


Figure 4.39: Relative error for BDIDE and BDIE for test 4

Test 5:

Circular domain $\bar{\Omega} = \{(x_1, x_2) : (x_1 - 1.5)^2 + (x_2 - 1.5)^2 \leq 0.25\}$, $k(x) = x_1^2 + x_2^2$, $a(x) = 1$,

$f(x) = 4 + (x_1^2 + x_2^2)^2$ for $x \in \bar{\Omega}$, and the boundary conditions:

$$\bar{u}(x) = x_1^2 + x_2^2, \text{ for } x_2 = \sqrt{0.25 - (x_1 - 1.5)^2} + 1.5,$$

$$\bar{t}(x) = 2(x_1 n_1(x) + x_2 n_2(x)), \text{ for } x_2 = -\sqrt{0.25 - (x_1 - 1.5)^2} + 1.5.$$

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

The second test example to be considered is a circular domain. Table 4.5 lists the computed acoustic pressure $u(x)$ for some selected interior points using BDIDE and BDIE, and Figure

4.40 represents the plot of the relative errors.

Table 4.5: Computed acoustic pressure for some selected interior points

(x_1, x_2)	BDIDE	BDIE	Exact
(1.25387206, 1.20879066)	3.02577130	3.03266203	3.03336100
(1.14867828, 1.37015320)	3.18952596	3.19618628	3.19678159
(1.24740029, 1.44618959)	3.64013856	3.64768562	3.64747180
(1.21390452, 1.56002889)	3.90026471	3.90758951	3.90725431
(1.27109495, 1.67196906)	4.40409550	4.41203385	4.41116290
(1.21064845, 1.77910151)	4.62408247	4.63144832	4.63087188

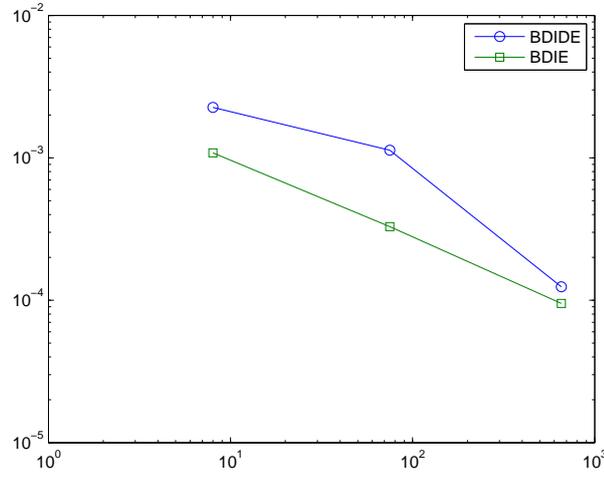


Figure 4.40: Relative error for BDIDE and BDIE for test 5

Test 6 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, where $k(x) = \cos(x_1) + \cos(x_2)$, for $x \in \bar{\Omega}$,

$a(x) = 1$, $f(x) = (\cos(x_1) + \cos(x_2))(x_1 + x_2)$ and the boundary conditions:

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 3 + x_1, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = (n_1(x) + n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

The third test example to be considered is a square domain. Table 4.6 lists the computed acoustic pressure $u(x)$ along the line $x_2 = 2.5$ of the plate using BDIDE and BDIE, and Figures 4.41 and 4.42 represent the plot of the results and relative errors, respectively.

From Tables 4.5 and 4.6 and Figs.4.38-4.42, it can be seen that both BDIDE and BDIE results are very close to the exact solution. Moreover, from relative error plots, it can

Table 4.6: Computed acoustic pressure along line $x_2 = 2.5$

x_1	BDIDE	BDIE	Exact
2	4.50001317	4.50001361	4.50000000
2.125	4.62500210	4.62500249	4.62500000
2.25	4.75000120	4.75000149	4.75000000
2.375	4.87500055	4.87500069	4.87500000
2.5	5.00000005	5.00000000	5.00000000
2.625	5.12499955	5.12499932	5.12500000
2.750	5.24999893	5.24999854	5.25000000
2.875	5.37499806	5.37499757	5.37500000
3	5.49998701	5.49998648	5.50000000

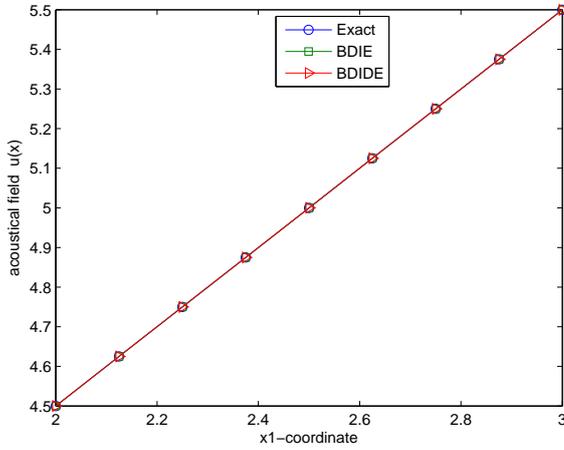
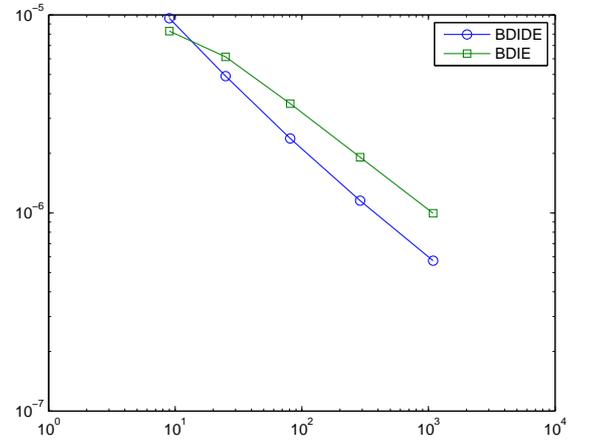
Figure 4.41: Acoustic pressure distribution along the line $x_2 = 2.875$ in test 6

Figure 4.42: Relative error for BDIDE and BDIE for test 6

be noticed that by increasing the number of nodes, good convergence is achieved. This demonstrates that the proposed formulations for both BDIDE and BDIE, and their numerical implementation, are correct.

4.6.3 Numerical results when both $a(x)$ and $k(x)$ variable

In this final case, when both the material parameter $a(x)$ and the wave number $k(x)$ are variable, the parametrix in Eq.(3.9) is adopted.

Test 7 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, where $k(x) = x_1 + x_2$, for $x \in \bar{\Omega}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = 2(\exp(x_1 + x_2)) + (x_1 + x_2)^2$ and the boundary conditions: $\bar{u}(x) = 1 + x_1$, for $x_2 = 1$; $1 \leq x_1 \leq 2$,

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 1 \leq x_1 \leq 2,$$

$$\bar{t}(x) = (\exp(x_1 + x_2))(n_1(x) + n_2(x)), \text{ for } x_1 = 1 \text{ or } x_1 = 2; \quad 1 \leq x_2 \leq 2.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2, x \in \bar{\Omega}$.

The first test example to be considered is a square domain. Table 4.7 lists the computed acoustic pressure $u(x)$ along the line $x_2 = 1.5$ of the plate using BDIDE and BDIE, and Figs.4.43 and 4.44 represent the plot of the results and relative errors, respectively.

Table 4.7: Computed acoustic pressure along line $x_2 = 1.5$

x_1	BDIDE	BDIE	Exact
1	2.50000827	2.49975098	2.50000000
1.125	2.62499815	2.62485102	2.62500000
1.25	2.74999800	2.74992766	2.75000000
1.375	2.87499777	2.87499185	2.87500000
1.5	2.99999737	3.00004969	3.00000000
1.625	3.12499670	3.12510806	3.12500000
1.750	3.24999566	3.25017286	3.25000000
1.875	3.37499422	3.37524777	3.37500000
2	3.49998273	3.50034354	3.50000000

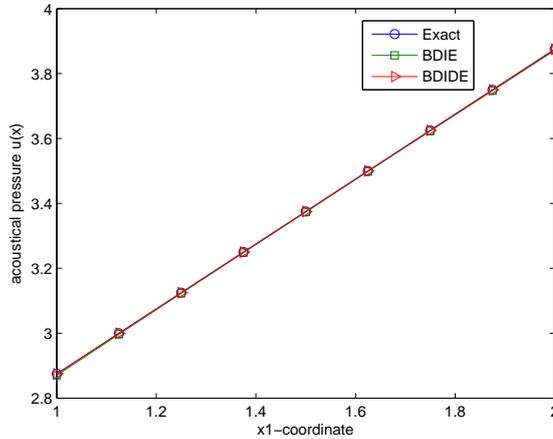


Figure 4.43: Acoustic pressure distribution along the line $x_2 = 1.875$ in test 7

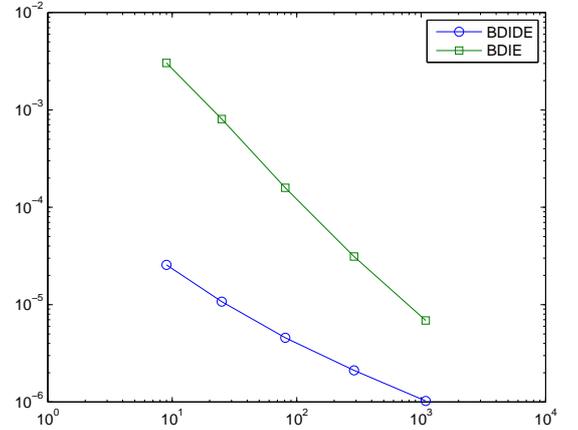


Figure 4.44: Relative error for BDIDE and BDIE for test 7

Test 8 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, where $k(x) = \sin(x_1) + \sin(x_2)$, for $x \in \bar{\Omega}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = (2(\exp(x_1 + x_2))(2 + x_1 + x_2)) + (\sin(x_1) + \sin(x_2))(x_1^2 + x_2^2)$ and the boundary conditions:

$$\bar{u}(x) = 1 + x_1, \text{ for } x_2 = 1; \quad 1 \leq x_1 \leq 2,$$

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 1 \leq x_1 \leq 2,$$

$$\bar{t}(x) = 2(\exp(x_1 + x_2))(x_1 n_1(x) + x_2 n_2(x)), \text{ for } x_1 = 1 \text{ or } x_1 = 2; \quad 1 \leq x_2 \leq 2.$$

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2, x \in \bar{\Omega}$.

The second test example to be considered is a square domain. Table 4.8 lists the computed acoustic pressure $u(x)$ along the line $x_2 = 1.5$ of the plate using BDIDE and BDIE, and Figs.4.45 and 4.46 represent the plot of the results and relative errors, respectively.

Table 4.8: Computed acoustic pressure along line $x_2 = 1.5$

x_1	BDIDE	BDIE	Exact
1	3.23907298	3.24875134	3.25000000
1.125	3.50510973	3.51593634	3.51562500
1.25	3.80153097	3.81314959	3.81250000
1.375	4.12948095	4.14165412	4.14062500
1.5	4.48895221	4.50149612	4.50000000
1.625	4.87992777	4.89272494	4.89062500
1.750	5.30237464	5.31538789	5.31250000
1.875	5.75625542	5.76951472	5.76562500
2	6.24013358	6.25374388	6.25000000

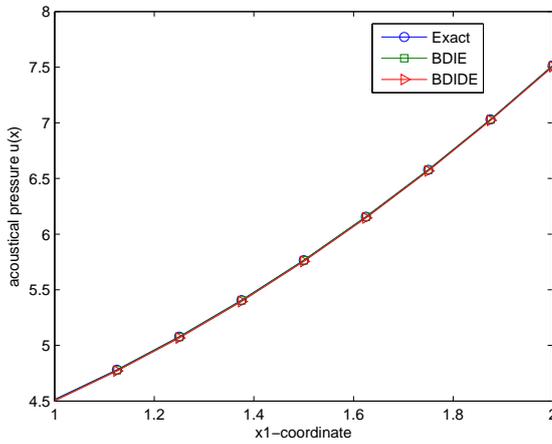


Figure 4.45: Acoustic pressure distribution along the line $x_2 = 1.875$ in test 8

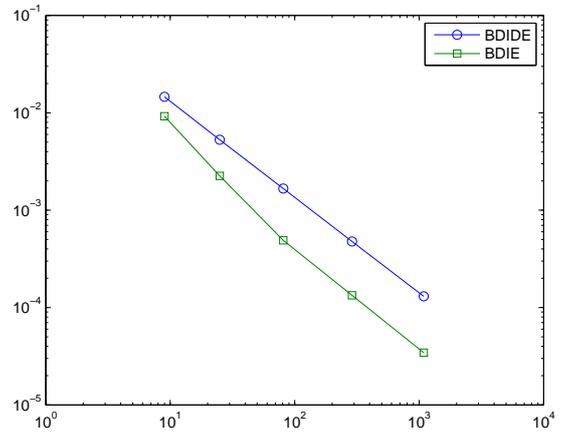


Figure 4.46: Relative error for BDIDE and BDIE for test 8

From Tables 4.7 and 4.8 and Figs.4.43-4.46, the results are acceptable with good rates of convergence.

In general, for all test 1-8, it can be seen that the BDIDE method produced better results for tests 1, 2, 4, 6, and 7. However, more accurate results were obtained for the BDIE method for the tests 3, 5 and 8. A possible explanation is the approximation of the flux t

in the BDIDE method using linear basis functions for u living on triangles; thus, $T\Phi_j(x)$ is constant within each triangle. These approximations are appropriate for the tests 1, 2, 4, 6, and 7 as the solution to these tests are all linear, while the solution to the tests 3, 5 and 8 is quadratic. The accuracy of the BDIE for tests 3, 5 and 8 is also reduced by the approximation of the flux t at the boundary nodes, as there is a slight flux discontinuity at these points which is avoided in the BDIDE method, as discussed in the previous chapter.

4.7 Conclusion

In this chapter, the BDIE and BDIDE formulations are derived and implemented for solving the two-dimensional Helmholtz equation with constant and variable coefficients. Four possible cases are investigated, first of all when both material parameters and wave number are constant.

Then, the zero-order Bessel function of the second kind is used when the material parameters are variable (with constant or variable wave number), and a parametrix is adopted to reduce the Helmholtz equation to a BDIE or BDIDE. However, when material parameters are constant (with variable wave number), the standard fundamental solution for the Laplace equation is used in the formulation.

Numerical test examples show that accurate computational results can be achieved using both BDIE and BDIDE methods. The boundary and domain integrals in the formulations have a weak singularity. To calculate the boundary integrals we used a standard Gaussian quadrature rule. For the domain integrals, we have implemented a Gaussian quadrature rule with Duffy transformation by mapping the triangles into squares and eliminating the weak singularity.

Chapter 5

Radial integration method

5.1 Introduction

The BDIE and BDIDE formulations have been developed and successfully implemented for heat conduction and Helmholtz problems with variable coefficients in chapters 3 and 4, respectively. However, these formulations require a domain integral.

Several methodologies have been proposed in order to overcome this difficulty. One possible technique is to find a fundamental solution for the problem with variable coefficients which can provide a pure boundary integral equation. Unfortunately, these fundamental solutions are only available for some very special cases as discussed in previous chapters. A robust methodology was developed by Kassab and Divo [7] in which generalised fundamental solutions are used to achieve boundary integral equations for heat conduction problems with spatially varying conductivity. Kassab and Divo's technique is based on developing a generalised forcing function rather than using the Dirac delta function in the derivation of the fundamental solutions. This technique can solve many problems, although issues have been raised by some researchers [42, 43]. In addition, as this technique has been developed for homogeneous problems, a domain integral will still appear when dealing with heat sources.

An alternative methodology for solving PDEs with variable coefficients with the BEM without domain discretisation involves the transformation of the domain integrals appearing in the integral equation, derived by using fundamental solutions for linear homogeneous problems, into equivalent boundary integrals. There are several methods available in the literature, such as the Galerkin vector technique [2, 12] applied to convert the domain

integral coming from a heat source or from a known function. For some special cases (e.g. a harmonic function), the Galerkin vector technique can be used to transform the domain integral to the boundary based on the second Green identity and a particular solution. The drawbacks from this approach are that it can be only applied for some simple cases of known functions as it requires to calculate a particular solution. Also, it can not be applied for domain integral with unknown functions (as for some of the domain integrals in previous chapters, coming from the remainder $R(x, y)$).

Nowak and Brebbia [44] developed an alternative technique called the multiple reciprocity method (MRM) to solve Poisson and Helmholtz equations. The MRM can be viewed as a generalisation of the Galerkin vector approach. Instead of using one higher-order fundamental solution as in the Galerkin vector, to convert the remaining domain integrals to equivalent boundary integrals a series of higher-order fundamental solutions is used. This method is very powerful, but it may be difficult to calculate the primitives in the recurrence formula governed by the Laplace operator [45]. The principal limitation of the MRM appears to be its lack of generality. In particular, it does not appear that the MRM can be applied to a general variable coefficient PDE [12].

The dual reciprocity method (DRM) was developed by Nardini and Brebbia [46]. In this method, the transformation is carried out by approximating the body force term with a series of basis functions and by using their particular solutions. A detailed description and practical applications of this method can be found in the book of Partridge et al. [2]. The drawback of this technique is that the particular solutions may be difficult to obtain for some complicated problems, depending on the radial basis function (RBF) adopted. In addition, even for known body forces, the method still requires an approximation of the known function using RBFs [45].

More recently, a new transformation technique, the radial integration method (RIM), has been developed by Gao [45, 47]. The RIM can transform any complicated domain integral to the boundary, while also removing various singularities appearing in the domain integrals. The main feature of the RIM is that it can treat different types of domain integrals in a unified way since it does not resort to particular solutions as in the DRM. The RIM was implemented in [47] for the analysis of elastoplastic problems, in which case strong and weak singularities were removed by transforming the domain integrals to the boundary. The RIM was also applied to thermoelastic problems in [48], in which case the domain integrals included in both displacement and internal stress integral equations

were transformed into equivalent boundary integrals. The Green's function for Laplace's equation was used to derive a boundary-domain integral equation for heat conduction problems with heat generation and spatially varying conductivity in [49]. Then, the RIM was adopted to convert the domain integrals for both heat generation and variable thermal conductivities to boundary integrals. Albuquerque et al. [50] extended the RIM to transform domain integrals into boundary integrals in a BEM formulation for anisotropic plate bending problems. Numerical results showed that the RIM, although more time consuming, presents some advantages over the DRM in terms of accuracy and the absence of particular solutions in the formulation for static and dynamic problems. Gao et al. [51] also implemented the RIM to solve elastic problems with nonlinearly-varying material parameters, such as for functionally graded materials and damage mechanics problems. Recent work by Yang et al. [52] presented analytic integrations for the RIM for heat conduction problems with variable coefficients, which can reduce the time needed for computing the radial integrals.

In this chapter, the RIM will be introduced and discussed in detail. For domain integrals consisting of known functions the transformation into boundary integrals is straightforward, while for domain integrals that include unknown variables the transformation is accomplished with the use of RBFs augmented by polynomials to approximate the unknown quantities as in the DRM. The most attractive feature of the method is that the transformations are very simple and have similar forms for both 2D and 3D problems. It can also remove various singularities appearing in the domain integrals, and treat different types of domain integrals in a unified way since it does not resort to particular solutions as in the DRM.

Modifications have been introduced to the RIM in its application to the BDIE and BDIDE formulations, particularly the fact that the radial integral is calculated by using a transformation proposed by Fata [53] which produces a pure boundary-only formulation and relaxes the "star-shaped" requirement of the RIM as the straight path from the source point to any field point will always exist. Some numerical examples are given to demonstrate the efficiency of the proposed methods.

5.2 Transformation of domain integrals to boundary using RIM

In this section, the RIM will be discussed to transform the domain integrals into boundary integrals for two possible cases. The first one is when the integrand is known as in a heat source domain integral, and the second case when the domain integral has an unknown integrand.

5.2.1 RIM formulation for domain integrals with known integrand

A domain integral with known integrand function $f(x)$, $x = (x_1, x_2)$, can be transformed into an equivalent boundary integral by the procedure described below [47].

Given a two-dimensional domain Ω bounded by a boundary Γ , define a Cartesian coordinate system (x_1, x_2) and a polar coordinate system (r, θ) with origin at the source point $y = (y_1, y_2)$. The relationships between the Cartesian and polar coordinate systems are:

$$r_1 = x_1 - y_1 = r \cos(\theta), \quad r_2 = x_2 - y_2 = r \sin(\theta), \quad (5.1)$$

where $0 \leq \theta \leq 2\pi$ and r is the distance between the source point y and a field point x .

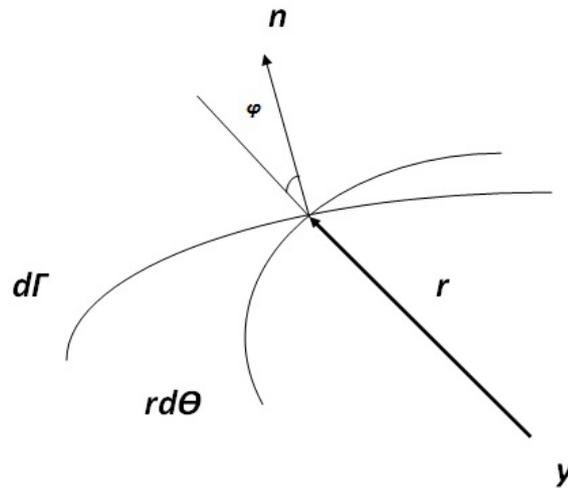


Figure 5.1: Relationship between differential elements $r d\theta$ and $d\Gamma$

The relationship between a differential domain in the Cartesian system and the polar

system is given by:

$$d\Omega = dx_1 dx_2 = J dr d\theta = r dr d\theta, \quad (5.2)$$

where J is the Jacobian given by:

$$J = \left| \frac{\partial(x_1, x_2)}{\partial(r, \theta)} \right| = \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_1}{\partial \theta} \\ \frac{\partial x_2}{\partial r} & \frac{\partial x_2}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -r \sin(\theta) \\ \sin(\theta) & r \cos(\theta) \end{vmatrix} = r.$$

From Fig. 5.1, when the field point is located on the boundary, we can obtain the following relation [47],

$$r d\theta = d\Gamma \cos \varphi = d\Gamma \frac{r_i n_i}{r}, \quad (5.3)$$

where φ is the angle between the normals of the differential arc $r d\theta$ with radius r and the differential boundary $d\Gamma$ with outward normal n_i , and the summation subscript i takes values 1 to 2.

Substituting Eq.(5.3) in Eq.(5.2) and re-arranging, we obtain:

$$d\Omega = r dr ds, \quad (5.4)$$

where

$$ds = \frac{1}{r} \frac{\partial r}{\partial n} d\Gamma, \quad (5.5)$$

$$\frac{\partial r}{\partial n} = r_{,i} n_i, \quad (5.6)$$

$$r_{,i} = \frac{\partial r}{\partial x_i} = \frac{r_i}{r}. \quad (5.7)$$

Now, a function in Cartesian coordinates can be written in polar coordinates and integrated as follows:

$$\int_{\Omega} f(x) d\Omega = \int_{\Gamma} \left\{ \int_0^{r(x)} f(x) r^\alpha dr \right\} d\Gamma(x) = \int_{\Gamma} F(x) d\Gamma(x) \quad (5.8)$$

where

$$F(x) = \int_0^{r(x)} f(x) r^\alpha dr. \quad (5.9)$$

In Eqs.(5.8) and (5.9), $\alpha = 1$ for the two-dimensional case and $\alpha = 2$ for the three-dimensional case. The symbol $r(x)$ means the variable r takes values on the boundary Γ , see Fig.(5.2).

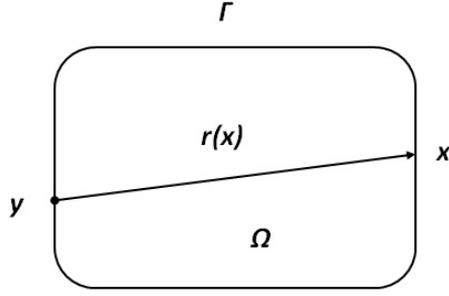


Figure 5.2: Integration along radial direction r

Substituting Eq.(5.5) into Eq.(5.8), we get:

$$\int_{\Omega} f(x) d\Omega = \int_{\partial\Omega} \frac{1}{r^{\alpha}} \frac{\partial r}{\partial n} F(x) d\Gamma(x). \quad (5.10)$$

The following remarks are important for the RIM:

- The most attractive feature of the RIM is that the transformation (5.9) is very simple and has similar forms for both 2D and 3D. It can remove various singularities appearing in domain integrals since r^{α} is included in the radial integral.

In order to transform a domain integral to a boundary integral, the main task is to calculate the radial integral in Eq.(5.9), which can be done analytically for simple kernels. We have written a simple Matlab code for analytic integration of Eq.(5.9) which can integrate many given functions $f(x)$, see Appendix C; however, for complicated functions, numerical integration techniques are required [45,47] which can be easily done in Matlab, see Appendix C.

- In order to evaluate the radial integral in Eq.(5.9), the coordinates x_1, x_2 in $f(x)$ need to be expressed in terms of the distance r using:

$$x_i = y_i + r_{,i} r \quad i = 1, 2, \quad (5.11)$$

where the quantities y_i and $r_{,i}$ are constant for the radial integral in Eq.(5.9).

- Following the idea presented in [53], we can introduce the change of variable:

$$r = t|x - y|, \quad t \in [0, 1] \quad (5.12)$$

and substitute the new transformation in the straight-line radial integral in Eq.(5.9), leading to:

$$F(x) = \int_0^1 f(y_1 + r_{,1}rt, y_2 + r_{,2}rt)r^2tdt. \quad (5.13)$$

The representation (5.13) makes it unnecessary to define a variable transformation as in [49] to treat the radial integral in Eq.(5.9), adding an attractive feature to the RIM as Eq.(5.13) is now a pure boundary integral. Moreover, the star-shaped requirement for the integral in Eq.(5.9) can be relaxed as the straight path from the source point y to any field point x always exists [53].

5.2.2 RIM formulation for domain integrals with unknown integrand

If the domain integral has an unknown function $u(x)$, the RIM in Eqs.(5.10) and (5.13) cannot be directly used. Therefore, similar to the Dual Reciprocity Method (DRM), $u(x)$ has to be approximated by RBFs [2, 54, 55]. We adopt an augmented RBF, as discussed in [2, 54, 55].

Let us write the unknown function $u(x)$ in the following way, assuming augmentation by a linear function:

$$u(x) = \sum_{k=1}^M \alpha_k \phi_k(R) + c_1x_1 + c_2x_2 + c_3, \quad (5.14)$$

where $M = N_b + N_I$ and N_b, N_I are the number of boundary and interior nodes, respectively. Also, $R = \|x - a\|$ is the distance from the application point a to the field point x . Normally, the application points a consist of all boundary nodes and some selected interior nodes. The most commonly used radial basis functions $\phi(R)$ are given in Table 5.1.

The following equilibrium conditions have to be satisfied [55]:

$$\sum_{k=1}^M \alpha_k = \sum_{k=1}^M \alpha_k x_{1k} = \sum_{k=1}^M \alpha_k x_{2k} = 0. \tag{5.15}$$

Table 5.1: Commonly used radial basis functions $\phi(R)$

R	$R + 1$	$\ R^3$	$\ 1 + R^2 + R^3$	$\ R^2 \log R$
-----	---------	----------	--------------------	-----------------

The unknown coefficients α_k , c_1 , c_2 and c_3 can be calculated by applying Eqs.(5.14) and (5.15) at the application points a , which can be written in the following matrix form [55]:

$$\left(B \right)_{(M+3) \times (M+3)} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \alpha_M \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}_{(M+3) \times 1} = \begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_M \\ 0 \\ 0 \\ 0 \end{pmatrix}_{(M+3) \times 1} \tag{5.16}$$

with

$$B = \begin{pmatrix} B1 & B2^T \\ B2 & \bar{0} \end{pmatrix}, \tag{5.17}$$

where $B1$ represents the matrix formed by the RBF values, i.e. the distance between application points a to the field points x , as written below:

$$B1 = \begin{pmatrix} \phi_{1,1} \dots \phi_{1,M} \\ \phi_{2,1} \dots \phi_{2,M} \\ \dots \\ \dots \\ \phi_{M,1} \dots \phi_{M,M} \end{pmatrix}_{M \times M} .$$

Also, $B2$ represents the matrix of linear polynomial terms as shown below,

$$B2 = \begin{pmatrix} 1 & 1 \dots \dots \dots 1 \\ x_{11} & x_{12} \dots \dots \dots x_{1M} \\ x_{21} & x_{22} \dots \dots \dots x_{2M} \end{pmatrix}_{3 \times M} .$$

Moreover, $\bar{0}$ represents a 3×3 zero matrix.

Substituting Eq.(5.14) into the following domain integral with unknown function $u(x)$, we obtain

$$\int_{\Omega} u(x) d\Omega(x) = \sum_{k=1}^M \alpha_k \int_{\Omega} \phi_k(R) d\Omega(x) + c_1 \int_{\Omega} x_1 d\Omega(x) + c_2 \int_{\Omega} x_2 d\Omega(x) + c_3 \int_{\Omega} d\Omega(x). \tag{5.18}$$

It is very important before applying the RIM using Eqs.(5.9)-(5.10) and (5.12)-(5.13), that the coordinates x_1 and x_2 appearing in each domain integral in Eq.(5.18) are expressed in terms of the distance r using Eq.(5.11).

Now, applying the RIM to each domain integral in Eq.(5.18) leads to

$$\int_{\Omega} u(x) d\Omega(x) = \sum_{k=1}^M \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F1(x) d\Gamma(x) + c_1 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F2(x) d\Gamma(x) + c_2 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F3(x) d\Gamma(x) + c_3 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F4(x) d\Gamma(x), \tag{5.19}$$

where

$$F1(x) = \int_0^1 \phi(R) r^2 t dt, \tag{5.20a}$$

$$F2(x) = \int_0^1 (y_1 + r_{,1} r t) r^2 t dt, \tag{5.20b}$$

$$F3(x) = \int_0^1 (y_2 + r_{,2} r t) r^2 t dt, \tag{5.20c}$$

$$F4(x) = \int_0^1 r^2 t dt. \tag{5.20d}$$

The four integrals in Eqs.(5.20a-5.20d) can be easily integrated numerically in Matlab.

Since $\phi(R)$ is function of the distance R , see Table 5.1, $\phi(R)$ needs to be expressed in term of the distance r . Gao [45, 47], referring to Fig. 5.3, defined three vectors $\vec{a}\vec{y}$ with length \bar{R} , $\vec{a}\vec{x}$ with length R and $\vec{y}\vec{x}$ with length r . From elementary calculus, we have the following identity:

$$\vec{a}\vec{x} = \vec{a}\vec{y} + \vec{y}\vec{x}.$$

Therefore,

$$|\vec{a}\vec{x}|^2 = (\vec{a}\vec{y} + \vec{y}\vec{x}) \cdot (\vec{a}\vec{y} + \vec{y}\vec{x}) = |\vec{a}\vec{y}|^2 + 2\vec{a}\vec{y} \cdot \vec{y}\vec{x} + |\vec{y}\vec{x}|^2.$$

Then,

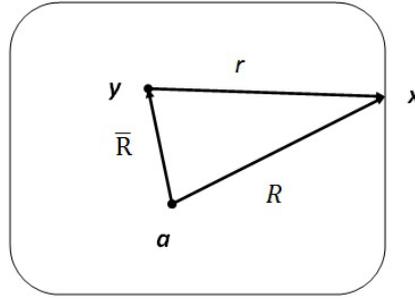


Figure 5.3: Relationship between distances

$$R = \sqrt{\bar{R}^2 + sr + r^2}, \quad (5.21)$$

where $s = 2 \left(\frac{(x-y) \cdot (y-a)}{r} \right)$.

We use a simpler procedure that leads to exactly the same results as in Eq.(5.21), in which we express $\phi(R) = R$ in terms of r as follows:

$$R = \sqrt{(a_1 - x_1)^2 + (a_2 - x_2)^2} = |\vec{a}\vec{x}|.$$

Then, using Eqs.(5.11) and (5.12), we get

$$R = \sqrt{(a_1 - (y_1 + r_1 r t))^2 + (a_2 - (y_2 + r_2 r t))^2}.$$

After numerical integration, if no two nodes share the same co-ordinates, the matrix B in Eq.(5.17) is invertible, then the unknown coefficients $\alpha_k, k = 1, \dots, M, c_1, c_2$ and c_3 , can

be calculated by:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \alpha_M \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = (B)^{-1} \begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_m \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

and the global matrix expressing the domain integral for the function u at all nodes can be written in the following way:

$$(A)_{M \times (M+3)} (B)_{(M+3) \times (M+3)}^{-1} (C)_{(M+3) \times M} \begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_M \end{pmatrix}_{M \times 1}, \quad (5.22)$$

where

$$(A)_{M \times (M+3)} = ((A1)_{M \times M} \quad (A2)_{M \times 1} \quad (A3)_{M \times 1} \quad (A4)_{M \times 1}),$$

and Ai ; $i = 1, \dots, 4$, are the matrices coming from the four integrals in Eq.(5.20),

$$(C)_{(M+3) \times M} = \begin{pmatrix} (I)_{M \times M} \\ (0)_{3 \times M} \end{pmatrix},$$

where I , 0 are the identity and zero matrices, respectively.

Remark 5.1:

It is important to point out that the formulation discussed in this section is augmented by a linear polynomial. In order to modify the procedure for augmentation by a quadratic polynomial the number three appearing in all dimensions of sub-matrices is increased to six as there are six coefficients in this case. Also, the four integrals in Eq.(5.20) will increase to seven as well.

5.3 Applications of the RIM

In the previous section, we have discussed the formulation of the RIM for transforming domain integrals with known and unknown functions to boundary integrals. It is important to point out that the procedure for unknown functions will also work for domain integrals with known functions. However, the direct RIM discussed in subsection 5.2.1 is recommended for domain integrals with known functions as there is an exact transformation in this case.

In this section, simple domain integrals (with known functions) will be tested using the RIM for both known and unknown procedures to convert the domain integrals to the boundary. For unknown procedures the non-augmented (local radial basis functions) and augmented cases by linear and quadratic polynomials with the different types of RBFs in Table 5.1 will be investigated.

5.3.1 The RIM for transforming domain integrals with known integrand

Test 1

Let us test the following domain integral,

$$\int_{\Omega} f(x) d\Omega(x) \quad (5.23)$$

on a square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$ with the following types of functions:

case 1: $f(x) = 2$,

case 2: $f(x) = x_1 + x_2$,

case 3: $f(x) = x_1^2 + x_2^2$,

case 4: $f(x) = x_1^3 + x_2^3$,

case 5: $f(x) = \exp(x_1 + x_2)$,

case 6: $f(x) = \exp(x_1 + x_2) + \cos(x_1 + x_2) + \sin(x_1 + x_2) + \log(x_1 + x_2)$, for $x \in \bar{\Omega}$.

Following the procedure described in subsection 5.2.1, the domain integral in Eq.(5.23) can be converted as follows:

$$\int_{\Omega} f(x) d\Omega(x) = \int_{\partial\Omega} \frac{1}{r^\alpha} \frac{\partial r}{\partial n} F(x) d\Gamma(x), \quad (5.24)$$

where

$$F(x) = \int_0^1 f(y_1 + r_1 r t, y_2 + r_2 r t) r^2 t dt. \quad (5.25)$$

The main task in the RIM is to calculate the radial integral in either Eq.(5.9) or Eq.(5.25). We have written Matlab codes in Appendix C (Radialintegral1, Radialintegral2) to calculate the radial integral in both Eq.(5.9) and Eq.(5.25) analytically. After calculating the radial integral, it can be put back in Eq.(5.24) and follow standard BEM procedures for the boundary integral as in the Matlab code in Appendix C.

In the present thesis, the radial integral in Eq.(5.25) will be adopted as it is a pure boundary integral. After running the (Radialintegral2) code in Appendix C, the radial integral $F(x)$ in Eq.(5.25) will be for cases (a-f):

a) $F(x) = r^2,$

b) $F(x) = (r^2 * (2 * x_1 + 2 * x_2 + y_1 + y_2))/6,$

c) $F(x) = (r^2 * x_1^2)/4 + (r^2 * x_2^2)/4 + (r^2 * y_1^2)/12 + (r^2 * y_2^2)/12 + (r^2 * x_1 * y_1)/6 + (r^2 * x_2 * y_2)/6,$

d) $F(x) = (r^2 * x_1^3)/5 + (r^2 * x_2^3)/5 + (r^2 * y_1^3)/20 + (r^2 * y_2^3)/20 + (r^2 * x_1 * y_1^2)/10 + (3 * r^2 * x_1^2 * y_1)/20 + (r^2 * x_2 * y_2^2)/10 + (3 * r^2 * x_2^2 * y_2)/20,$

e) $F(x) = \exp(y_1 + y_2) / ((x_1 - y_1)/r + (x_2 - y_2)/r)^2 - (\exp(x_1 + x_2) * (y_1 - x_2 - x_1 + y_2 + 1)) / ((x_1 - y_1)/r + (x_2 - y_2)/r)^2,$

f) $F(x) = \text{Warning: Explicit integral could not be found.}$

We can conclude from case f) that the analytic calculation of the radial integral in Eq.(5.25) will not always be possible, therefore numerical integration is recommended for complicated functions.

The results obtained by applying the Matlab code in Appendix C with 8 boundary elements are presented in Table 5.2. The radial integral in Eq.(5.25) is calculated numerically, see Appendix C.

$f(x)$	RIM	Exact
case 1	2	2
case 2	3	3
case 3	4.6667	4.6667
case 4	7.5	7.5
case 5	21.8161	21.8161
case 6	22.1248	22.1248

Table 5.2: Computational results of the RIM for the integral in test 1

It can be seen from Table 5.2 that the results are exact to the number of decimal places

displayed, despite the simple BEM discretisation.

5.3.2 The RIM for transforming domain integrals with unknown integrand

In this subsection, the domain integrals in test 1 with function cases (1-6) will be tested again using the RIM procedure for unknown functions to convert the domain integrals to the boundary. The non-augmented RBFs and augmented RBFs by linear and quadratic polynomials (LP and QP) with total number of nodes equal to 25 (16 on the boundary plus 9 in interior), will be investigated with different types of RBFs in Table 5.1.

$f(x)$	R	$R + 1$	R^3	$1 + R^2 + R^3$	$R^2 \log R$	Exact
case 1	1.9877	1.9941	2.0047	2.0036	2.0207	2
case 2	2.9815	2.9911	3.0070	3.0054	3.0310	3
case 3	4.6470	4.6637	4.6768	4.6736	4.7174	4.6667
case 4	7.4948	7.5265	7.5142	7.5070	7.5886	7.5
case 5	21.7810	21.8698	21.8558	21.8361	22.0663	21.8161
case 6	22.0908	22.1811	22.1651	22.1450	22.3791	22.1248

Table 5.3: Computational results for non-augmented RBFs

It can be seen from Tables 5.3-5.6 that the results for augmented RBFs are much better than for non-augmented. Also, the results for RBFs augmented by a linear polynomial are exact for constant and linear functions, while the results for RBFs augmented by a quadratic polynomial are exact for constant, linear, quadratic and cubic functions. These results are similar to those in [54, 55] in which augmented RBFs have been implemented in the DRM. It is very easy to change the type of RBFs in the RIM by changing this in both the $B1$ matrix and in the integral in Eq.(5.20a) only, which is different from the DRM where the particular solution and the corresponding matrices all must be changed.

$f(x)$	R	$R + 1$	R^3	$1 + R^2 + R^3$	$R^2 \log R$	Exact
case 1	2	2	2	2	2	2
case 2	3	3	3	3	3	3
case 3	4.7714	4.7714	4.6680	4.6680	4.7049	4.6667
case 4	7.9711	7.9711	7.5059	7.5059	7.6719	7.5
case 5	22.8886	22.8886	21.6868	21.6868	22.1166	21.8161
case 6	23.2349	23.2349	22.0020	22.0020	22.4429	22.1248

Table 5.4: Computational results for augmented RBFs by LP without interior nodes

It can be clearly seen from Tables 5.4 and 5.5 that the augmented RBFs by a linear

polynomial with interior nodes produce better results in comparison to the case without interior nodes. Such behaviour is again similar to the DRM [2, 54, 55].

$f(x)$	R	$R + 1$	R^3	$1 + R^2 + R^3$	$R^2 \log R$	Exact
case 1	2	2	2	2	2	2
case 2	3	3	3	3	3	3
case 3	4.6790	4.6790	4.6678	4.6678	4.6706	4.6667
case 4	7.5556	7.5556	7.5049	7.5049	7.5175	7.5
case 5	21.9512	21.9512	21.8271	21.8271	21.8583	21.8161
case 6	22.2640	22.2640	22.1361	22.1361	22.1683	22.1248

Table 5.5: Computational results for augmented RBFs by LP with interior nodes

Moreover, the results in Table 5.6 for RBFs augmented by a quadratic polynomial are better than for linear augmentation. However, they are more time consuming as the four integrals in Eq.(5.20) become seven integrals in this case.

$f(x)$	R	$R + 1$	R^3	$1 + R^2 + R^3$	$R^2 \log R$	Exact
case 1	2	2	2	2	2	2
case 2	3	3	3	3	3	3
case 3	4.6667	4.6667	4.6667	4.6667	4.6667	4.6667
case 4	7.5	7.5	7.5	7.5	7.5	7.5
case 5	21.8138	21.8138	21.8144	21.8144	21.8142	21.8161
case 6	22.1226	22.1226	22.1231	22.1231	22.1230	22.1248

Table 5.6: Computational results for augmented RBFs by QP with interior nodes

5.4 Conclusion

In this chapter, the radial integration method (RIM) is presented to transform domain integrals into equivalent boundary integrals. The most attractive feature of the method is that the transformations are very simple and have similar forms for both 2D and 3D problems. They can also remove various singularities appearing in the domain integrals, and treat different types of domain integrals in a unified way since the RIM does not resort to particular solutions as in the DRM.

Moreover, for domain integrals with known functions the transformation is straightforward, while for domain integrals that include unknown variables the transformation is accomplished with the use of RBFs augmented by polynomials to approximate the unknown quantities as in the DRM. Several cases have been investigated for domain integrals with

unknown functions using different types of non-augmented RBFs, and with RBFs augmented by linear and quadratic polynomials. The numerical results showed that RBFs augmented by a quadratic polynomial produce the best result, however they are more time consuming. In the next two chapters the RIM will be implemented to convert the domain integrals appearing in both BDIE and BDIDE formulations for heat conduction and Helmholtz equations discussed in chapters 3 and 4, respectively.

Chapter 6

Radial integration method for heat conduction with variable coefficients

6.1 Introduction

In this chapter, a new type of boundary-only integral equation analysis technique is developed for non-homogeneous heat conduction problems with variable coefficients based on the use of a parametrix (Levi function). Unlike the existing method discussed in chapter 3 and in our published work [41], where the BDIE and BDIDE are solved using the cell-integration technique, the RIM is adopted and used to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals. For the domain integrals consisting of known functions the transformation is direct, while for domain integrals that include unknown variables the transformation is accomplished with the use of a RBF augmented by polynomials to approximate the unknown quantities as in the DRM. The most attractive feature of the method is that the transformations are very simple and have similar forms for both 2D and 3D problems. The proposed methodology calculates the radial integral as a pure boundary integral and relaxes the “star-shaped” requirement of the RIM. Some numerical examples are given to demonstrate the efficiency of the proposed methods. The present formulations and numerical results of the radial integration BIE and BIDE for heat conduction problems with variable coefficients, associated with mixed boundary conditions, have been published in [56].

6.2 Integral equation for heat conduction with variable coefficients

The use of Green's third identity in the form of Eq.(3.11) for heat conduction problems with variable coefficients has two main drawbacks. The first is that it contains the variable coefficient $a(x)$ in each integrand, which makes the implementation of the formulation and the resulting computer code less general and difficult to develop as a unified code, as the function $a(x)$ will change for different problems. The second main drawback is that there are two domain integrals with known integrand on the right-hand side (coming from heat source effects), and the second domain integral with unknown integrand on the left hand-side coming from the remainder $R(x, y)$.

As shown in chapter 3, the BVP (3.1)-(3.3) can be expressed as the following integral equation,

$$c(y)u(y) - \int_{\partial\Omega} [u(x)T_x P(x, y) - P(x, y)Tu(x)]d\Gamma(x) + \int_{\Omega} R(x, y)u(x)d\Omega(x) = \int_{\Omega} P(x, y)f(x)d\Omega(x). \quad (6.1)$$

Now, we can multiply both sides of Eq.(6.1) by $a(y)$ to obtain:

$$a(y)c(y)u(y) - \int_{\partial\Omega} [u(x)T_x \tilde{P}(x, y) - \tilde{P}(x, y)Tu(x)]d\Gamma(x) + \int_{\Omega} \tilde{R}(x, y)u(x)d\Omega(x) = \int_{\Omega} \tilde{P}(x, y)f(x)d\Omega(x), \quad (6.2)$$

where

$$\tilde{P}(x, y) = a(y)P(x, y) = \frac{1}{2\pi} \ln |x - y|, \quad (6.3)$$

$$\tilde{R}(x, y) = a(y)R(x, y) = \sum_{i=1}^2 \frac{x_i - y_i}{2\pi|x - y|^2} \frac{\partial a(x)}{\partial x_i}. \quad (6.4)$$

Differently from [8, 19, 28–31, 56], the parametrix in identity (6.2) is the fundamental solution to the Laplace equation, which is much easier to implement in a unified code. Also, identity (6.2) can be used for formulating either a BDIE or a BDIDE, with respect

to u and its derivatives. Following the same procedure as in chapter 3, the new modified BDIE and BDIDE can be written as follows.

6.2.1 Boundary-domain integral equation (BDIE)

$$c^0(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x,y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x,y)t(x)d\Gamma(x) + \int_{\Omega} \tilde{R}(x,y)u(x)d\Omega(x) = \Psi^0(y), \quad y \in \Omega \cup \partial\Omega, \quad (6.5)$$

where

$$\Psi^0(y) := [c^0(y) - a(y)c(y)]\bar{u}(y) + \Psi(y), \quad (6.6)$$

$$\Psi(y) := \int_{\partial_D\Omega} \bar{u}(x)T_x\tilde{P}(x,y)d\Gamma(x) - \int_{\partial_N\Omega} \tilde{P}(x,y)\bar{t}(x)d\Gamma(x) + \int_{\Omega} \tilde{P}(x,y)f(x)d\Omega(x) \quad (6.7)$$

and $c^0(y)$ is given by

$$c^0(y) = \begin{cases} 0 & \text{if } y \in \partial_D\Omega \\ a(y)c(y) & \text{if } y \in \Omega \cup \partial_N\Omega \end{cases} \quad (6.8)$$

6.2.2 Boundary-domain integro-differential equation (BDIDE)

$$a(y)c(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x,y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x,y)Tu(x)d\Gamma(x) + \int_{\Omega} \tilde{R}(x,y)u(x)d\Omega(x) = \Psi(y), \quad y \in \Omega \cup \partial_N\Omega, \quad (6.9)$$

where $\Psi(y)$ is given by Eq.(6.7).

6.3 Transformation of domain integrals to boundary using RIM

In this section, the RIM discussed in the previous chapter is used to transform the domain integrals appearing in equations (6.5) and (6.9) into boundary integrals.

6.3.1 Transformation of heat source domain integral to the boundary

Both BDIE in Eq.(6.5) and BDIDE in Eq.(6.9) have domain integrals coming from the known heat source $f(x)$ that appears in Eq.(6.7). The RIM can be directly used to convert these domain integrals to the boundary, as discussed in subsection 5.2.1. This leads to

$$\int_{\Omega} \tilde{P}(x, y) f(x) d\Omega(x) = \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x) d\Gamma(x), \quad (6.10)$$

where

$$F(x) = \int_0^1 \tilde{P}(x, y) f(y_1 + r_{,1}rt, y_2 + r_{,2}rt) r^2 dt. \quad (6.11)$$

The integral in Eq.(6.11) can be calculated analytically for many heat sources, and numerically without the need to define a transformation as in [49], as discussed in detail in the previous chapter. Also, due to the radial integral in Eq.(6.11), the weak singularity coming from the fundamental solution is removed.

6.3.2 RIM formulation for domain integrals with unknown integrand

As the last domain integrals on the left-hand side of Eqs. (6.5) and (6.9) have the unknown temperature $u(x)$, the RIM cannot be directly used. Following exactly the same procedure discussed in subsection 5.2.2 (the only difference being that the term $\tilde{R}(x, y)$ in Eq.(6.4)

now appears inside the domain integral) leads to:

$$\begin{aligned} \int_{\Omega} \tilde{R}(x, y)u(x)d\Omega(x) &= \sum_{k=1}^M \alpha_k \int_{\Omega} \tilde{R}(x, y)\phi_k(R)d\Omega(x) + c_1 \int_{\Omega} \tilde{R}(x, y)x_1d\Omega(x) \\ &+ c_2 \int_{\Omega} \tilde{R}(x, y)x_2d\Omega(x) + c_3 \int_{\Omega} \tilde{R}(x, y)d\Omega(x). \end{aligned} \quad (6.12)$$

Let $r_{,1} = \frac{x_1-y_1}{r}$ and $r_{,2} = \frac{x_2-y_2}{r}$, then $\tilde{R}(x, y)$ in Eq.(6.4) can be written as:

$$\tilde{R}(x, y) = \frac{1}{2\pi} \left(\frac{r_{,1}}{r} \frac{\partial a(x)}{\partial x_1} + \frac{r_{,2}}{r} \frac{\partial a(x)}{\partial x_2} \right). \quad (6.13)$$

It is very important before applying the RIM that the coordinates x_1 and x_2 appearing in Eqs.(6.12) and (6.13) are expressed in terms of the distance r using Eq.(5.11).

Now, applying the RIM in section 5.2 to each domain integral in Eq.(6.12) leads to

$$\int_{\Omega} \tilde{R}(x, y)u(x)d\Omega(x) = \int_{\partial\Omega} h(x)d\Gamma(x)$$

and

$$\begin{aligned} \int_{\partial\Omega} h(x)d\Gamma(x) &= \sum_{k=1}^M \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F1(x)d\Gamma(x) + c_1 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F2(x)d\Gamma(x) \\ &+ c_2 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F3(x)d\Gamma(x) + c_3 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F4(x)d\Gamma(x), \end{aligned} \quad (6.14)$$

where

$$F1(x) = \int_0^1 \tilde{R}(x, y)\phi(R)r^2tdt, \quad (6.15a)$$

$$F2(x) = \int_0^1 \tilde{R}(x, y)(y_1 + r_{,1}rt)r^2tdt, \quad (6.15b)$$

$$F3(x) = \int_0^1 \tilde{R}(x, y)(y_2 + r_{,2}rt)r^2tdt, \quad (6.15c)$$

$$F4(x) = \int_0^1 \tilde{R}(x, y)r^2tdt. \quad (6.15d)$$

The calculations of the four integrals in Eq.(6.15) are discussed in detail in section 5.2.

6.4 The radial integration boundary integral and integro-differential equations

Eqs.(6.10)-(6.11) and (6.14)-(6.15) can now be substituted in both BDIE in Eq.(6.5) and BDIDE in Eq.(6.9), leading to the expressions in the next subsections.

6.4.1 The radial integration boundary integral equation (RI-BIE)

$$c^0(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x,y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x,y)t(x)d\Gamma(x) + \int_{\partial\Omega} h(x)d\Gamma(x) = \Psi^0(y), \quad y \in \Omega \cup \partial\Omega, \quad (6.16)$$

where

$$\Psi^0(y) := [c^0(y) - a(y)c(y)]\bar{u}(y) + \tilde{\Psi}(y), \quad (6.17)$$

$$\tilde{\Psi}(y) := \int_{\partial_D\Omega} \bar{u}(x)T_x\tilde{P}(x,y)d\Gamma(x) - \int_{\partial_N\Omega} \tilde{P}(x,y)\bar{t}(x)d\Gamma(x) + \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x)d\Gamma(x), \quad (6.18)$$

and $c^0(y)$, $F(x)$ and $\int_{\partial\Omega} h(x)d\Gamma(x)$ are given in Eqs.(6.8), (6.11) and (6.14)-(6.15), respectively.

6.4.2 The radial integration boundary integro-differential equation (RIBIDE)

$$a(y)c(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x,y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x,y)Tu(x)d\Gamma(x) + \int_{\partial\Omega} h(x)d\Gamma(x) = \tilde{\Psi}(y), \quad y \in \Omega \cup \partial_N\Omega, \quad (6.19)$$

where $\int_{\partial\Omega} h(x)d\Gamma(x)$ and $\tilde{\Psi}(y)$ are given in Eqs.(6.14)-(6.15) and (6.18), respectively.

It can be clearly seen from both RIBIE in Eq.(6.16) and RIBIDE in Eq.(6.19) that all integrations are now carried out only on the boundary, with no domain integrals.

6.5 Discretisation of the RIBIE and RIBIDE

6.5.1 Discretisation of the RIBIE

The RIBIE formulation employs mixed boundary elements with linear u and constant t to avoid the discontinuities of t at corner points. In this case, collocation was taken at the end points of each boundary element, since our previous results in chapters three and four have shown that end-node collocation generally provides higher accuracy than mid-node collocation.

Let J be the total number of nodes x^i , $i = 1, \dots, J$, at the end points of elements, from which there are J_D nodes on $\partial_D\Omega$. Thus, the values of u at any point on the element can be defined in terms of their nodal values and two linear interpolation functions $\Psi^1(t)$ and $\Psi^2(t)$ given by Eq.(2.27).

To obtain a system of linear algebraic equations from the RIBIE (6.16), we collocate at the nodes x^i , $i = 1, \dots, J$. We can also use an interpolation of $t(x) = (Tu)(x^j)$ along boundary nodes belonging to $x^j \in \partial_D\Omega$

$$t(x) = \sum_{x^j \in \partial_D\Omega} t(x^j)v_j(x), \quad x \in \partial_D\Omega. \quad (6.20)$$

Here, $v_j(x)$ are boundary shape functions, taken now as constant. Therefore, $v_j(x)$ will be equal to 1 at $x^j \in \partial_D\Omega$ and $v_j(x) = 0$ if $x^j \notin \partial_D\Omega$. Substituting the interpolations (2.27) and (6.20) in RIBIE (6.16) and applying the collocation method, we arrive at the following system of J linear algebraic equations for J unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N\Omega$ and $t(x^j) = (Tu)(x^j)$, $x^j \in \partial_D\Omega$,

$$\begin{aligned} c^0(x^i)u(x^i) + \sum_{x^j \in \Omega \cup \partial_N\Omega} K_{ij}u(x^j) + \sum_{x^j \in \partial_D\Omega} Q'_{ij}t(x^j) &= \Psi^0(x^i) - \\ - \sum_{x^j \in \partial_D\Omega} K_{ij}\bar{u}(x^j), \quad x^i \in \Omega \cup \partial\Omega, \quad i = 1, \dots, J, \text{ no sum in } i, & \end{aligned} \quad (6.21)$$

where $\Psi^0(x^i)$ is calculated from Eq.(6.17), and

$$\tilde{\Psi}(x^i) = \int_{\partial_D\Omega} \bar{u}(x) T_x \tilde{P}(x, x^i) d\Gamma(x) - \int_{\partial_N\Omega} \tilde{P}(x, x^i) \bar{t}(x) d\Gamma(x) + \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x) d\Gamma(x), \quad (6.22)$$

$$K_{ij} = \int_{\partial\Omega} h(x) d\Gamma(x) - \int_{\partial_N\Omega} [\Psi^1, \Psi^2] T_x \tilde{P}(x, x^i) d\Gamma(x), \quad (6.23)$$

$$Q'_{ij} = \int_{\partial_D\Omega} \tilde{P}(x, x^i) v_j(x) d\Gamma(x). \quad (6.24)$$

6.5.2 Discretisation of the RIBIDE

To obtain a system of linear algebraic equations from the RIBIDE (6.19), we collocate at the nodes x^i , $i = 1, \dots, J$, and substitute an interpolation of $u(x)$ of the form

$$u(x) \approx \sum_{S_j \ni x} u(x^j) \Phi_j(x), \quad \Phi_j(x) = \begin{cases} \phi_{kj}(x) & \text{if } x, x^j \in \bar{T}_k \\ 0 & \text{otherwise,} \end{cases} \quad (6.25)$$

where S_j in this case is the set of collocation points in $\partial_D\Omega$ and some selected interior nodes near the boundary segments; $\phi_{kj}(x)$ are the shape functions which can be constructed from the distance between the two end nodes of each segments and the selected interior nodes, and associated with the node x^j as the normal derivative requires that the function is known perpendicular to the boundary. In this work, $\phi_{kj}(x)$ are chosen as piecewise linear functions. It is important to point out here that such formulation is unlike the standard BEM where u and t on the boundary are independent variables.

We then arrive at a system of $J - J_D$ algebraic equations for $J - J_D$ unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N\Omega$. Substituting interpolation formulae (6.25) into the RIBIDE (6.19) leads to the following system of equations:

$$a(x^i) c(x^i) u(x^i) + \sum_{x^j \in \Omega \cup \partial_N\Omega} K'_{ij} u(x^j) = \tilde{\Psi}(x^i) - \sum_{x^j \in \partial_D\Omega} K'_{ij} \bar{u}(x^j), \quad (6.26)$$

$x^i \in \Omega \cup \partial_N\Omega$, no sum in i ,

where

$$K'_{ij} = K_{ij} + \int_{\partial_D\Omega} \tilde{P}(x, x^i) T \Phi_j(x) d\Gamma(x) \quad (6.27)$$

and $\tilde{\Psi}(x^j)$ and K_{ij} are given in Eqs.(6.22) and (6.23), respectively.

The details of the calculations of the boundary and domain integrals are given in Appendix B. The interior nodes near the Dirichlet boundary are used only to define the Jacobian (see Appendix B) but not as collocation points. The advantages of the RIBIDE technique are that the only boundary variables are those of u along Neumann boundaries, as there is no need for collocation along Dirichlet boundaries. Thus, the problem caused by the discontinuity of the normal derivative at corner points is avoided. Second, the system of linear equations is smaller than the one for RIBIE. This feature will save memory and computational time when we apply the RIBIDE for practical problems. Finally, the assembling of matrix A and vector b is much easier than in the RIBIE, as discussed in chapters 3 and 4.

6.6 Numerical results

In this section, we shall examine some test examples to assess the performance of the RIBIE/ RIBIDE formulations. To verify the convergence of the methods, we applied the methods to some test problems on a square domain, for which an exact analytical solution, u_{exact} , is available. For comparison, the problems are also computed using both BDIDE and BDIE given in chapter 3. Also, the relative error was calculated as in chapter 2, as given by Eq.(2.35) and the Root Mean Square (RMS) is defined as

$$RMS(J) = \left(\frac{\sum_{j=1}^J (u_{approx,j} - u_{exact,j})^2}{\sum_{j=1}^J u_{exact,j}^2} \right)^{1/2}, \quad (6.28)$$

where u_{approx} is the numerical solution and J is the number of nodes in the computational mesh. These errors have been calculated for $J= 25, 81, 289$ and 1089 in all test examples.

6.6.1 Poisson's equation with mixed boundary conditions

The starting point for testing the RIBIE/ RIBIDE formulations is to consider Poisson's equation, in which case there is a domain integral coming from $f \neq 0$; we assume a square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = 1$, $f(x) = 4$ for $x \in \bar{\Omega}$, with boundary conditions,

$$\bar{u}(x) = 4 + x_1^2, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 9 + x_1^2, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$\bar{t}(x) = 2(x_1 n_1(x) + x_2 n_2(x))$, for $x_1 = 2$ or $x_1 = 3$; $2 \leq x_2 \leq 3$.

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

Table 6.1 lists the computed temperatures $u(x)$ along the middle line of the plate using RIBIDE and RIBIE, while Fig. 6.1 plots the results along the line $x_2 = 2.875$. The total number of boundary nodes is 32 in both cases. In addition, 16 interior nodes are used for the RIBIDE method, 8 near the top face and 8 near the bottom face of the plate where the Dirichlet boundary conditions are defined. No interior nodes are necessary for the RIBIE technique. It can be seen that the RIBIDE and RIBIE results are very close to the BDIDE and BDIE results, respectively.

Table 6.1: Computed temperatures along the line $x_2 = 2.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
2	10.23978745	10.23978851	10.25004510	10.25004675	10.25000000
2.125	10.75602601	10.75602813	10.76664511	10.76664784	10.76562500
2.25	11.30245688	11.30245871	11.31330624	11.31330869	11.31250000
2.375	11.88030341	11.88030507	11.89129850	11.89130078	11.89062500
2.5	12.48958075	12.48958235	12.50062670	12.50062893	12.50000000
2.625	13.13029723	13.13029889	13.14129031	13.14129260	13.14062500
2.750	13.80244312	13.80244496	13.81328878	13.81329123	13.81250000
2.875	14.50600215	14.50600426	14.51661639	14.51661911	14.51562500
3	15.23965207	15.23965314	15.24990440	15.24990605	15.25000000

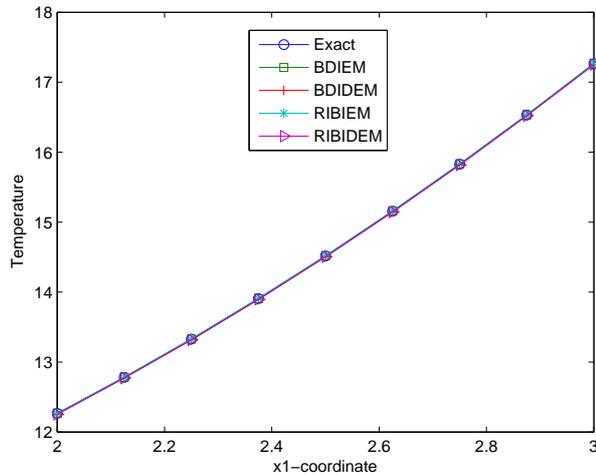


Figure 6.1: Temperature distribution along the line $x_2 = 2.875$

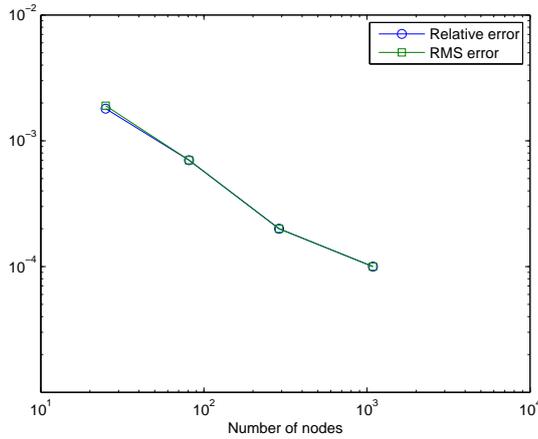


Figure 6.2: Relative and RMS errors for RBIDE method for Poisson's equation

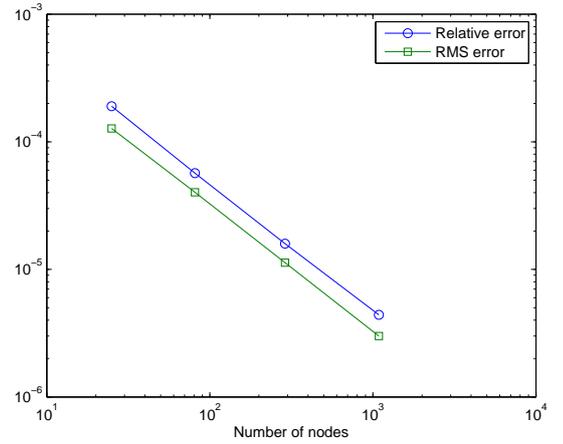


Figure 6.3: Relative and RMS errors for RBIE method for Poisson's equation

In the next tests we will consider problems with variable coefficients with increasing degree of complexity of the variation of both the material coefficients and the body force term f . The exact solutions of the problems range from linear to cubic, and will be used to verify the convergence of the numerical solutions. Moreover, $\phi(R) = R^3$ in chapter 5 (Table 5.1) is adopted in the next test examples.

6.6.2 Test 1

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = 2(x_1 + x_2)$, $f(x) = 4$ for $x \in \bar{\Omega}$, with boundary conditions:

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 3 + x_1, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

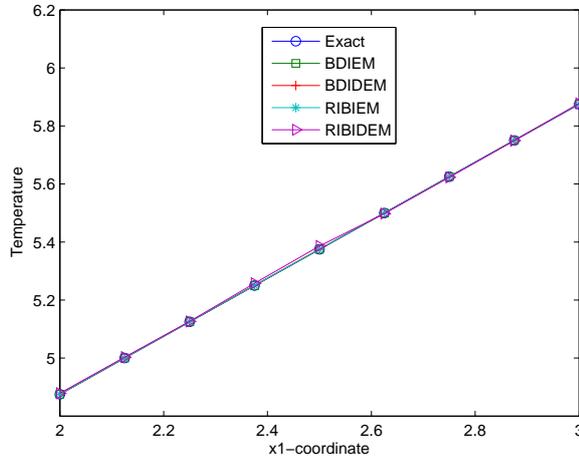
$$\bar{t}(x) = 2(x_1 + x_2)(n_1(x) + n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

Table 6.2 lists the computed temperatures $u(x)$ along the middle line of the plate using RBIDE and RBIE, while Fig. 6.4 shows the temperatures along the line $x_2 = 2.875$.

Table 6.2: Computed temperatures along the line $x_2 = 2.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
2	4.50001258	4.50372426	4.50001416	4.50001436	4.50000000
2.125	4.62500157	4.62847379	4.62500308	4.62500356	4.62500000
2.25	4.75000068	4.75263200	4.75000196	4.75000220	4.75000000
2.375	4.87500001	4.87690243	4.87500097	4.87500119	4.87500000
2.5	4.99999943	5.00094713	5.00000004	5.00000025	5.00000000
2.625	5.12499880	5.12496950	5.12499907	5.12499928	5.12500000
2.750	5.24999798	5.24920965	5.24999795	5.24999817	5.25000000
2.875	5.37499687	5.37339011	5.37499662	5.37499667	5.37500000
3	5.49998561	5.49836636	5.49998526	5.49998541	5.50000000

Figure 6.4: Temperature distribution along the line $x_2 = 2.875$

6.6.3 Test 2

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, $a(x) = x_1^2 + x_2^2$, $f(x) = 8(x_1^2 + x_2^2)$ for $x \in \bar{\Omega}$, with boundary conditions:

$$\bar{u}(x) = 1 + x_1^2, \text{ for } x_2 = 1; \quad 1 \leq x_1 \leq 2,$$

$$\bar{u}(x) = 4 + x_1^2, \text{ for } x_2 = 2; \quad 1 \leq x_1 \leq 2,$$

$$\bar{t}(x) = 2(x_1^2 + x_2^2)(x_1 n_1(x) + x_2 n_2(x)), \text{ for } x_1 = 1 \text{ or } x_1 = 2; \quad 1 \leq x_2 \leq 2.$$

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

Table 6.3 lists the computed temperatures along the middle line of the plate, while Fig. 6.7 shows the temperatures along the line $x_2 = 1.875$.

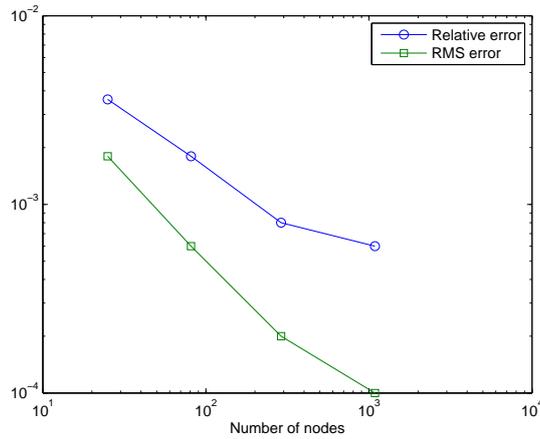


Figure 6.5: Relative and RMS errors for RIBIDE method for test 1

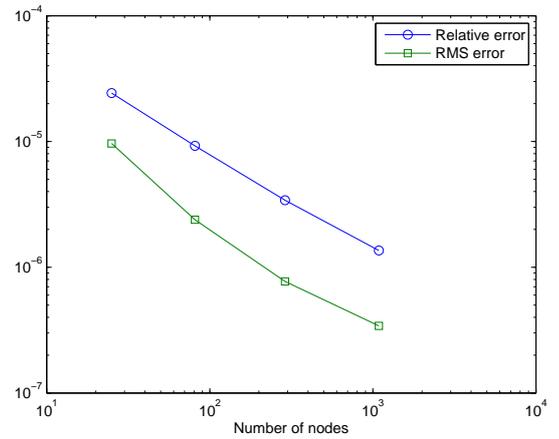


Figure 6.6: Relative and RMS errors for RIBIE method for test 1

Table 6.3: Computed temperatures along the line $x_2 = 1.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
1	3.23920748	3.27464013	3.24851271	3.25003483	3.25000000
1.125	3.50540314	3.53538109	3.51542087	3.51661121	3.51562500
1.25	3.80189587	3.82668952	3.81238259	3.81327508	3.81250000
1.375	4.12986266	4.15003618	4.14064478	4.14128510	4.14062500
1.5	4.48930354	4.50444549	4.50022930	4.50063728	4.50000000
1.625	4.88021029	4.88998845	4.89115469	4.89132522	4.89062500
1.750	5.30255791	5.30736330	5.31343854	5.31334249	5.31250000
1.875	5.75631681	5.75642826	5.76708929	5.76667600	5.76562500
2	6.24010302	6.23934447	6.25069986	6.24993669	6.25000000

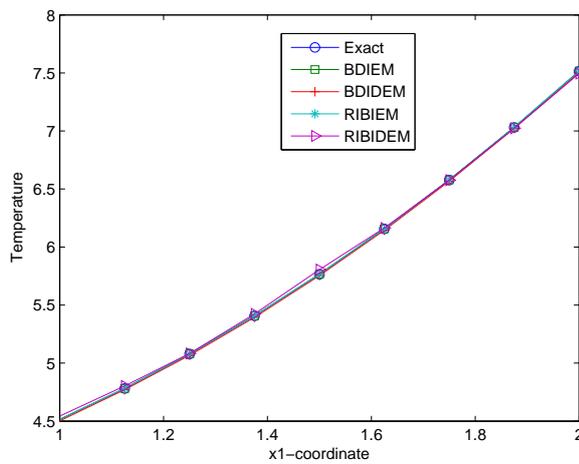


Figure 6.7: Temperature distribution along the line $x_2 = 1.875$

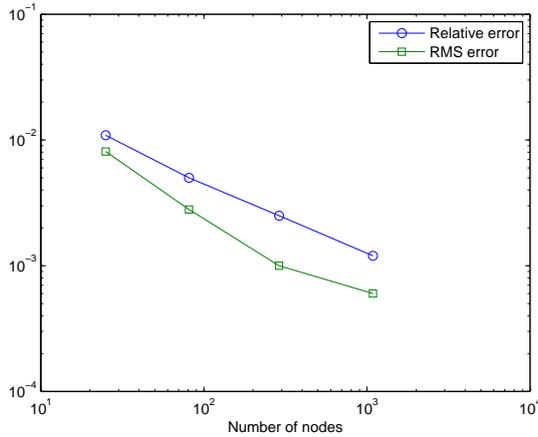


Figure 6.8: Relative and RMS errors for RIBIDE method for test 2

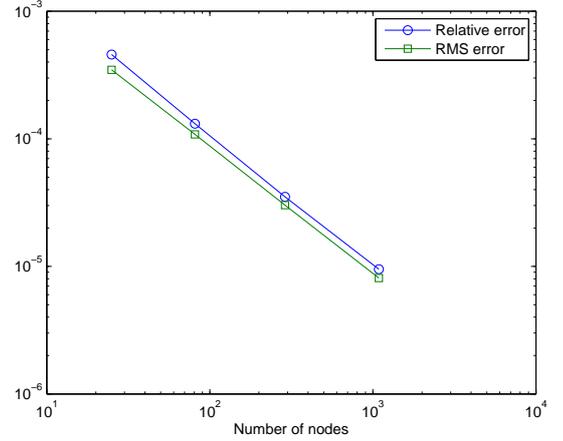


Figure 6.9: Relative and RMS errors for RIBIE method for test 2

6.6.4 Test 3

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = \exp(x_1 + x_2)(6x_1 + 3x_1^2 + 6x_2 + 3x_2^2)$ for $x \in \bar{\Omega}$, with boundary conditions:

$$\bar{u}(x) = 8 + x_1^3, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 27 + x_1^3, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = \exp(x_1 + x_2)(3x_1^2n_1(x) + 3x_2^2n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1^3 + x_2^3$, $x \in \bar{\Omega}$.

Table 6.4 lists the computed temperatures $u(x)$ along the middle line of the plate using RBIDE and RBIE, while Fig. 6.10 shows the temperatures along the line $x_2 = 2.875$.

Table 6.4: Computed temperatures along the line $x_2 = 2.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
2	23.54232032	24.08630891	23.60527950	23.62380015	23.62500000
2.125	25.14010914	25.58333193	25.21156793	25.22695034	25.22070313
2.25	26.93073535	27.28210016	27.00804008	27.02042258	27.01562500
2.375	28.93461607	29.20987807	29.01570850	29.02566367	29.02148438
2.5	31.16341970	31.36593610	31.24659571	31.25423100	31.25000000
2.625	33.62872621	33.76025149	33.71270058	33.71778096	33.71289063
2.750	36.34199348	36.40918430	36.42599323	36.42801377	36.42187500
2.875	39.31464962	39.31893069	39.39832517	39.39627288	39.38867188
3	42.54761895	42.52963849	42.63055914	42.62478750	42.62500000

It can be seen from Tables 6.1-6.4 and Figs. 6.1, 6.4, 6.7 and 6.10 that both the RIBIE and RIBIDE methods are able to generate accurate solutions in good agreement with BDIE

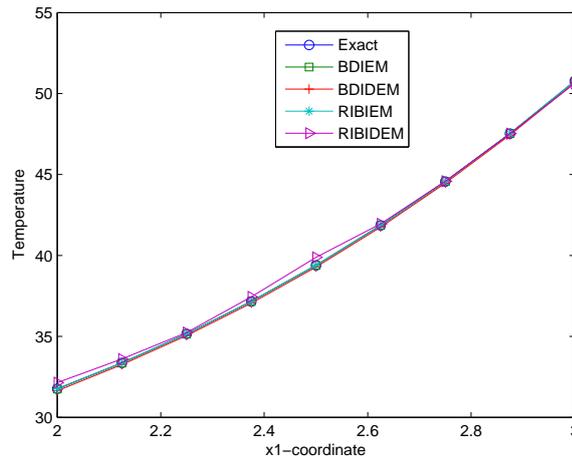
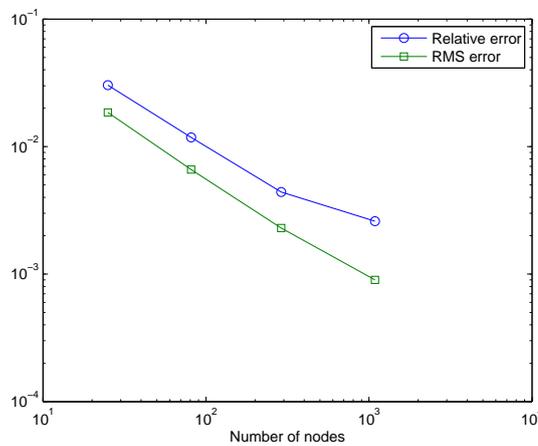
Figure 6.10: Temperature distribution along the line $x_2 = 2.875$ 

Figure 6.11: Relative and RMS errors for RIBIDE method for test 3

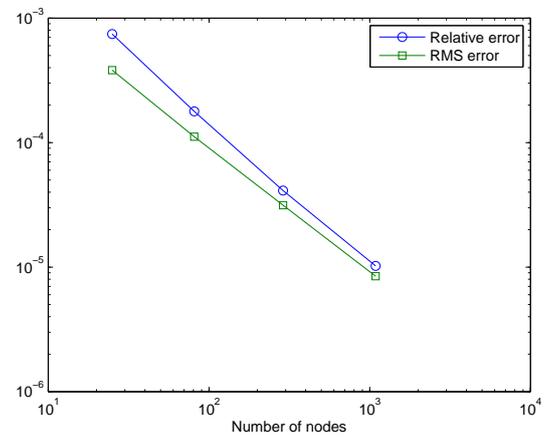


Figure 6.12: Relative and RMS errors for RIBIE method for test 3

and BDIDE results. It is important to point out that the numerical integration in Matlab is very fast and can save a substantial amount of computational time in comparison to both BDIDE and BDIE. It is noticed that the RIBIE produces better results than RIBIDE in all tests. Moreover, the relative and RMS errors in Figs. 6.2-6.3, 6.5-6.6, 6.8-6.9 and 6.11-6.12 show that both the RIBIE and RIBIDE methods are convergent with mesh refinement, and in general the RMS error is lower than the relative error as expected.

6.7 Conclusion

In this chapter, a pure BEM formulation is presented for solving heat conduction problems with variable conductivities. The radial integration method is used to transform the domain integrals appearing in both the boundary-domain integral or integro-differential equation formulations (discussed in chapter 3) to equivalent boundary integrals. Moreover, the RIM removes the weak singularities appearing in both domain integrals, simplifying and speeding up the calculation of the integrals.

Chapter 7

RIBIE and RIBIDE for Helmholtz equation with variable coefficients

7.1 Introduction

In this chapter, a new type of boundary-only integral equation analysis technique is developed for non-homogeneous two-dimensional Helmholtz equation based on the use of a parametrix (Levi function). Unlike chapter 4 where the BDIE and BDIDE are solved using the cell-integration technique, the RIM is adopted and used to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals. For the domain integrals consisting of known functions the transformation is direct, while for domain integrals that include unknown variables the transformation is accomplished with the use of a RBF augmented by polynomials to approximate the unknown quantities as in the DRM. The most attractive feature of the method is that the transformation formulations are very simple and have similar forms for both 2D and 3D problems. Some numerical examples are given to demonstrate the efficiency of the proposed methods. The present formulations and numerical results of the radial integration BIE and BIDE for the Helmholtz equation with variable coefficients, associated with mixed boundary conditions, have been published in [57].

7.2 Integral equation for the Helmholtz equation with variable coefficients

The third Green identity as developed in chapter 4, Eq.(4.5), has two main drawbacks. The first is that it contains the variable coefficient $a(x)$ in each integrand, which makes the implementation of the formulation and the resulting computer code less general and difficult to develop as a unified code, as the function $a(x)$ will change for different problems. The second main drawback is that, there are two domain integrals with known integrand on the right-hand side (coming from source effects), and the second domain integral with unknown integrand on the left hand-side coming from the remainder $R(x, y) + k(x)P(x, y)$. As shown in chapter 4, the BVP (4.1)-(4.3) can be expressed as the integral equation (4.5). Now, we can multiply both sides of Eq.(4.5) by $a(y)$ to obtain

$$a(y)c(y)u(y) - \int_{\partial\Omega} [u(x)T_x\tilde{P}(x, y) - \tilde{P}(x, y)Tu(x)]d\Gamma(x) + \int_{\Omega} [\tilde{R}(x, y) + k(x)\tilde{P}(x, y)]u(x)d\Omega(x) = \int_{\Omega} \tilde{P}(x, y)f(x)d\Omega(x), \quad (7.1)$$

where $\tilde{P}(x, y)$ and $\tilde{R}(x, y)$ are given by Eqs. (6.3) and (6.4), respectively.

The parametrix in identity (7.1) is the fundamental solution for the Laplace equation, which is much easier to implement in a unified code. Also, identity (7.1) can be used for formulating either a BDIE or BDIDE, with respect to u and its derivatives. Following the same procedure as in chapter 4, the new modified BDIE and BDIDE can be written as follows.

7.2.1 Boundary-domain integral equation (BDIE)

$$c^0(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x, y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x, y)t(x)d\Gamma(x) + \int_{\Omega} [\tilde{R}(x, y) + k(x)\tilde{P}(x, y)]u(x)d\Omega(x) = \Psi^0(y), \quad y \in \Omega \cup \partial\Omega, \quad (7.2)$$

where $\Psi^0(y)$ is given by Eq. (6.6).

7.2.2 Boundary-domain integro-differential equation (BDIDE)

$$\begin{aligned}
 a(y)c(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x,y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x,y)Tu(x)d\Gamma(x) + \\
 + \int_{\Omega} [\tilde{R}(x,y) + k(x)\tilde{P}(x,y)]u(x)d\Omega(x) = \Psi(y), \quad y \in \Omega \cup \partial_N\Omega, \quad (7.3)
 \end{aligned}$$

where $\Psi(y)$ is given by Eq.(6.7).

7.3 Transformation of domain integrals to boundary using RIM

In this section, the RIM discussed in chapter 5 is used to transform the domain integrals appearing in equations (7.2) and (7.3) into boundary integrals.

7.3.1 Transformation of right-hand side domain integral to the boundary

Both Eq.(7.2) and Eq.(7.3) have domain integrals coming from the known function $f(x)$. The RIM can be directly used to convert these domain integrals to the boundary, as discussed in subsection 5.2.1. This leads to Eq.(6.10)

The integral in Eq.(6.11) can be calculated analytically for many different functions, and numerically without the need to define a transformation as in [49], as discussed in previous chapter.

7.3.2 RIM formulation for domain integrals with unknown integrand

As the last domain integrals on the left-hand side of Eqs. (7.2) and (7.3) have the unknown function $u(x)$, the RIM cannot be directly used. Following exactly the same procedure discussed in subsection 5.2.2 (the only difference being that the term $\tilde{R}(x,y) + k(x)P(x,y)$)

in Eq.(6.4) now appears inside the domain integral) leads to:

$$\begin{aligned} \int_{\Omega} \{\tilde{R}(x, y) + k(x)\tilde{P}(x, y)\}u(x)d\Omega(x) &= \sum_{k=1}^M \alpha_k \int_{\Omega} \{\tilde{R}(x, y) + k(x)\tilde{P}(x, y)\}\phi_k(R)d\Omega(x) + \\ &+ d_1 \int_{\Omega} \{\tilde{R}(x, y) + k(x)\tilde{P}(x, y)\}x_1d\Omega(x) + d_2 \int_{\Omega} \{\tilde{R}(x, y) + k(x)\tilde{P}(x, y)\}x_2d\Omega(x) + \\ &+ d_3 \int_{\Omega} \{\tilde{R}(x, y) + k(x)\tilde{P}(x, y)\}d\Omega(x). \end{aligned} \quad (7.4)$$

Let $r_{,1} = \frac{x_1-y_1}{r}$ and $r_{,2} = \frac{x_2-y_2}{r}$, then we can write:

$$\{\tilde{R}(x, y) + k(x)\tilde{P}(x, y)\} = \frac{1}{2\pi} \left(\frac{r_{,1}}{r} \frac{\partial a(x)}{\partial x_1} + \frac{r_{,2}}{r} \frac{\partial a(x)}{\partial x_2} + k(x)\tilde{P}(x, y) \right). \quad (7.5)$$

It is very important before applying the RIM the coordinates that x_1 and x_2 appearing in Eqs.(7.4) and (7.5) are expressed in terms of the distance r using Eq.(5.11).

Now, applying the RIM in section 5.2 to each domain integral in Eq.(7.4) leads to

$$\int_{\Omega} \{\tilde{R}(x, y) + k(x)\tilde{P}(x, y)\}u(x)d\Omega(x) = \int_{\partial\Omega} z(x)d\Gamma(x)$$

and

$$\begin{aligned} \int_{\partial\Omega} z(x)d\Gamma(x) &= \sum_{k=1}^M \alpha_k \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F1(x)d\Gamma(x) + d_1 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F2(x)d\Gamma(x) + \\ &+ d_2 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F3(x)d\Gamma(x) + d_3 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F4(x)d\Gamma(x), \end{aligned} \quad (7.6)$$

where

$$F1(x) = \int_0^1 \{\tilde{R}(x, y) + k(y_1 + r_{,1}rt, y_2 + r_{,2}rt)\tilde{P}(x, y)\}\phi(R)r^2tdt, \quad (7.7a)$$

$$F2(x) = \int_0^1 \{\tilde{R}(x, y) + k(y_1 + r_{,1}rt, y_2 + r_{,2}rt)\tilde{P}(x, y)\}(y_1 + r_{,1}rt)r^2tdt, \quad (7.7b)$$

$$F3(x) = \int_0^1 \{\tilde{R}(x, y) + k(y_1 + r_{,1}rt, y_2 + r_{,2}rt)\tilde{P}(x, y)\}(y_2 + r_{,2}rt)r^2tdt, \quad (7.7c)$$

$$F4(x) = \int_0^1 \{\tilde{R}(x, y) + k(y_1 + r_{,1}rt, y_2 + r_{,2}rt)\tilde{P}(x, y)\}r^2tdt. \quad (7.7d)$$

The calculations of the four integrals in Eq.(7.7) are discussed in detail in section 5.2.

7.4 The radial integration boundary integral and integro-differential equations

Eqs.(6.10)-(6.11) and (7.6)-(7.7) can now be substituted in both BDIE in Eq.(7.2) and BDIDE in Eq.(7.3), and this leads to the following expressions.

7.4.1 The radial integration boundary integral equation (RI-BIE)

$$c^0(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x, y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x, y)t(x)d\Gamma(x) + \int_{\partial\Omega} z(x)d\Gamma(x) = \Psi^0(y), \quad y \in \Omega \cup \partial\Omega \quad (7.8)$$

where

$$\Psi^0(y) := [c^0(y) - a(y)c(y)]\bar{u}(y) + \tilde{\Psi}(y), \quad (7.9)$$

$$\tilde{\Psi}(y) := \int_{\partial_D\Omega} \bar{u}(x)T_x\tilde{P}(x, y)d\Gamma(x) - \int_{\partial_N\Omega} \tilde{P}(x, y)\bar{t}(x)d\Gamma(x) + \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x)d\Gamma(x), \quad (7.10)$$

where $c^0(y)$, $F(x)$ and $\int_{\partial\Omega} z(x)d\Gamma(x)$ are given in Eqs.(6.8), (6.11) and (7.6)-(7.7), respectively.

7.4.2 The radial integration boundary integro-differential equation (RIBIDE)

$$a(y)c(y)u(y) - \int_{\partial_N\Omega} u(x)T_x\tilde{P}(x,y)d\Gamma(x) + \int_{\partial_D\Omega} \tilde{P}(x,y)Tu(x)d\Gamma(x) + \int_{\partial\Omega} z(x)d\Gamma(x) = \tilde{\Psi}(y), \quad y \in \Omega \cup \partial_N\Omega, \quad (7.11)$$

where $\int_{\partial\Omega} z(x)d\Gamma(x)$ and $\tilde{\Psi}(y)$ are given in Eqs.(7.6)-(7.7) and (7.10), respectively.

It can be seen clearly from both RIBIE in Eq.(7.8) and RIBIDE in Eq.(7.11) that all integrations are now carried out only the boundary, with no domain integrals.

7.5 Discretisation of the RIBIE and RIBIDE

7.5.1 Discretisation of the RIBIE

The RIBIE formulation employs mixed boundary elements with linear u and constant t to avoid the discontinuities of t at corner points. In this case, collocation was taken at the end points of each boundary element, since our previous chapters have shown that end-node collocation generally provides higher accuracy than mid-node collocation.

Let J be the total number of nodes x^i , $i = 1, \dots, J$, at the end points of elements, from which there are J_D nodes on $\partial_D\Omega$. Thus, the values of u at any point on the element can be defined in terms of their nodal values and two linear interpolation functions $\Psi^1(t)$ and $\Psi^2(t)$, given in Eq. (2.27).

To obtain a system of linear algebraic equations from the RIBIE (7.8), we collocate at the nodes x^i , $i = 1, \dots, J$. We can also use an interpolation of $t(x) = (Tu)(x^j)$ along boundary nodes belonging to $x^j \in \partial_D\Omega$

$$t(x) = \sum_{x^j \in \partial_D\Omega} t(x^j)v_j(x), \quad x \in \partial_D\Omega. \quad (7.12)$$

Here, $v_j(x)$ are boundary shape functions, taken now as constant. Therefore, $v_j(x)$ will be equal to 1 at $x^j \in \partial_D\Omega$ and $v_j(x) = 0$ if $x^j \notin \partial_D\Omega$. Substituting the interpolations (2.27) and (7.12) in the RBIE (7.8) and applying the collocation method, we arrive at the following system of J linear algebraic equations for J unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N\Omega$ and $t(x^j) = (Tu)(x^j)$, $x^j \in \partial_D\Omega$,

$$\begin{aligned} c^0(x^i)u(x^i) + \sum_{x^j \in \Omega \cup \partial_N\Omega} K_{ij}u(x^j) + \sum_{x^j \in \partial_D\Omega} Q'_{ij}t(x^j) &= \Psi^0(x^i) \\ - \sum_{x^j \in \partial_D\Omega} K_{ij}\bar{u}(x^j), \quad x^i \in \Omega \cup \partial\Omega, \quad i = 1, \dots, J, \text{ no sum in } i, \end{aligned} \quad (7.13)$$

where $\Psi^0(x^i)$ is calculated from Eq.(7.9), and

$$\Psi(x^i) = \int_{\partial_D\Omega} \bar{u}(x)T_x\tilde{P}(x, x^i)d\Gamma(x) - \int_{\partial_N\Omega} \tilde{P}(x, x^i)\bar{t}(x)d\Gamma(x) + \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x)d\Gamma(x), \quad (7.14)$$

$$K_{ij} = \int_{\partial\Omega} z(x)d\Gamma(x) - \int_{\partial_N\Omega} [\Psi^1, \Psi^2]T_x\tilde{P}(x, x^i)d\Gamma(x), \quad (7.15)$$

$$Q'_{ij} = \int_{\partial_D\Omega} \tilde{P}(x, x^i)v_j(x)d\Gamma(x), \quad (7.16)$$

where $F(x)$ is given in Eq.(6.11) and $\int_{\partial\Omega} z(x)d\Gamma(x)$ is given in Eqs.(7.6)-(7.7).

7.5.2 Discretisation of the RIBIDE

To obtain a system of linear algebraic equations from the RIBIDE (7.11), we collocate at the nodes x^i , $i = 1, \dots, J$, and substitute an interpolation in Eq.(6.25), we then arrive at a system of $J - J_D$ algebraic equations for $J - J_D$ unknowns $u(x^j)$, $x^j \in \Omega \cup \partial_N\Omega$. Substituting interpolation formulae (6.25) into the RIBIDE (7.11) leads to the following system of equations:

$$\begin{aligned} a(x^i)c(x^i)u(x^i) + \sum_{x^j \in \Omega \cup \partial_N\Omega} K'_{ij}u(x^j) &= \Psi(x^i) - \sum_{x^j \in \partial_D\Omega} K'_{ij}\bar{u}(x^j), \\ x^i \in \Omega \cup \partial_N\Omega, \quad \text{no sum in } i, \end{aligned} \quad (7.17)$$

where

$$K'_{ij} = K_{ij} + \int_{\partial D \Omega} \tilde{P}(x, x^i) T \Phi_j(x) d\Gamma(x) \quad (7.18)$$

and $\Psi(x^i)$ and K_{ij} are given in Eqs.(7.14) and (7.15), respectively.

The advantages of the RIBIDE technique are that the only boundary variables are those of u along Neumann boundaries, as there is no need for collocation along Dirichlet boundaries. Thus, the problem caused by the discontinuity of the normal derivative at corner points is avoided. Second, the system of linear equations is smaller than the one for RIBIE. This feature will save memory and computational time when we apply the RIBIDE for practical problems. Finally, the assembly of matrix A and vector b is much easier than in the RIBIE, as discussed in previous chapters.

7.6 Numerical results

In this section, we shall examine some test examples to assess the performance of the RIBIDE/RIBIE formulations for the non-homogeneous Helmholtz equation with variable coefficients for three cases. Firstly, when the parameter $a(x)$ is variable and the wave number k is constant. Secondly, when the parameter $a(x)$ is constant and the wave number $k(x)$ is variable. Thirdly, when both the parameter $a(x)$ and the wave number $k(x)$ are variable. For comparison, the problems are also computed using both BDIDE and BDIE.

We applied the RIBIDE/RIBIE and BDIE/BDIDE methods to some test problems on a square domain, for which an exact analytical solution, u_{exact} , is available. Computer programs were developed by using Matlab. The exact solutions of the problems range from linear to cubic, and will be used to verify the convergence of the numerical solutions. Moreover, $\phi(R) = R^3$ is adopted in the test examples. The total number of nodes is 81 (32 on the boundary plus 49 in the interior). The top and bottom sides of the plates for all tests examples have prescribed acoustic pressure u (Dirichlet boundary conditions), while the left and right are imposed with normal velocity t (Neumann boundary conditions). Also, the relative error and Root Mean Square (RMS) error are calculated as given by Eqs.(2.35) and (6.28) to check the convergence of the proposed methods. These errors have been calculated for $J= 25, 81, 289$ and 1089 in all test examples.

7.6.1 Numerical results when $a(x)$ variable and $k(x)$ constant

Test 1 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, where $k(x) = 1$, for $x \in \bar{\Omega}$, $a(x) = 2(x_1 + x_2)$, $f(x) = 4 + x_1 + x_2$ and the boundary conditions:

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 3 + x_1, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = 2(x_1 + x_2)(n_1(x) + n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

Table 7.1: Computed acoustic pressure along line of $x_2 = 2.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
2	4.50001261	4.50410518	4.50001472	4.50001407	4.50000000
2.125	4.62500159	4.62799548	4.62500359	4.62500309	4.62500000
2.25	4.75000069	4.75269790	4.75000233	4.75000179	4.75000000
2.375	4.87500002	4.87623517	4.87500117	4.87500083	4.87500000
2.5	4.99999943	5.00087173	5.00000008	4.99999980	5.00000000
2.625	5.12499878	5.12429503	5.12499896	5.12499891	5.12500000
2.750	5.24999796	5.24910682	5.24999770	5.24999772	5.25000000
2.875	5.37499684	5.37283038	5.37499625	5.37499615	5.37500000
3	5.49998557	5.49839621	5.49998486	5.49998504	5.50000000

Table 7.1 lists the computed values of $u(x)$ along the middle line of the plate using RIBIDE, RIBIE, BDIDE and BDIE, while Fig.7.1 shows the variation of $u(x)$ along the line $x_2 = 2.875$.

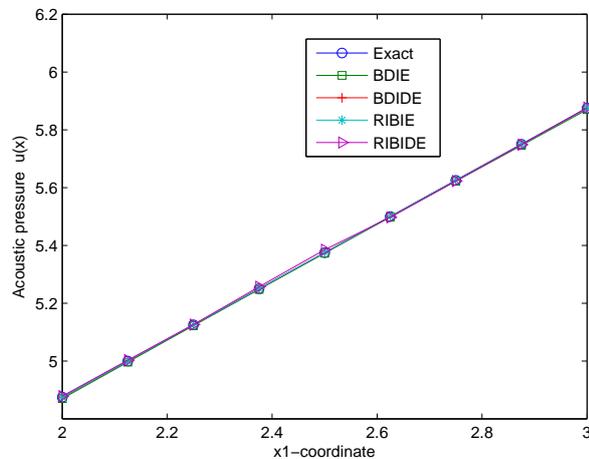


Figure 7.1: Acoustic pressure distribution along the line $x_2 = 2.875$

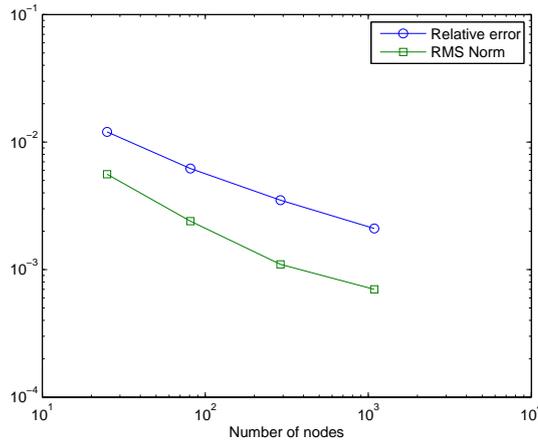


Figure 7.2: Relative and RMS errors for RIBIDE method for test 1

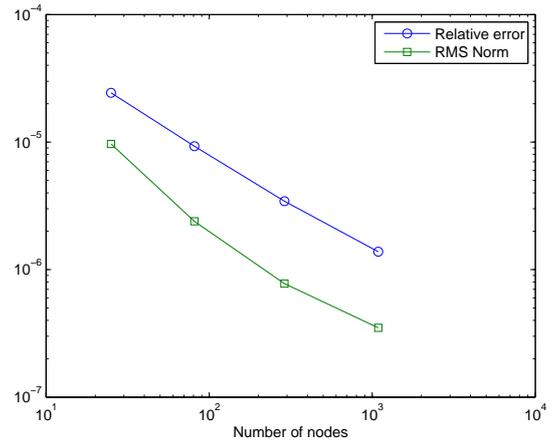


Figure 7.3: Relative and RMS errors for RIBIE method for test 1

Test 2:

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, where $k(x) = 1$, for $x \in \bar{\Omega}$, $a(x) = x_1^2 + x_2^2$, $f(x) = 9(x_1^2 + x_2^2)$ and the boundary conditions:

$$\bar{u}(x) = 1 + x_1^2, \text{ for } x_2 = 1; \quad 1 \leq x_1 \leq 2,$$

$$\bar{u}(x) = 4 + x_1^2, \text{ for } x_2 = 2; \quad 1 \leq x_1 \leq 2,$$

$$\bar{t}(x) = 2(x_1^2 + x_2^2)(x_1 n_1(x) + x_2 n_2(x)), \text{ for } x_1 = 1 \text{ or } x_1 = 2; \quad 1 \leq x_2 \leq 2.$$

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

Table 7.2 lists the computed values of $u(x)$ along the middle line of the plate using RIBIDE, RIBIE, BDIDE and BDIE, while Fig.7.4 shows the variation of $u(x)$ along the line $x_2 = 2.875$.

Table 7.2: Computed acoustic pressure along line of $x_2 = 1.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
1	3.23904254	3.27576532	3.25001788	3.25005523	3.25000000
1.125	3.50524145	3.53417482	3.51695736	3.51663265	3.51562500
1.25	3.80173933	3.82657272	3.81393990	3.81329415	3.81250000
1.375	4.12971330	4.14846598	4.14223153	4.14130378	4.14062500
1.5	4.48916298	4.50394266	4.50185631	4.50065229	4.50000000
1.625	4.88007916	4.88811964	4.89283775	4.89134235	4.89062500
1.750	5.30243537	5.30636671	5.31520029	5.31335853	5.31250000
1.875	5.75620027	5.75461679	5.76895964	5.76669239	5.76562500
2	6.23998789	6.23825471	6.25274089	6.24995223	6.25000000

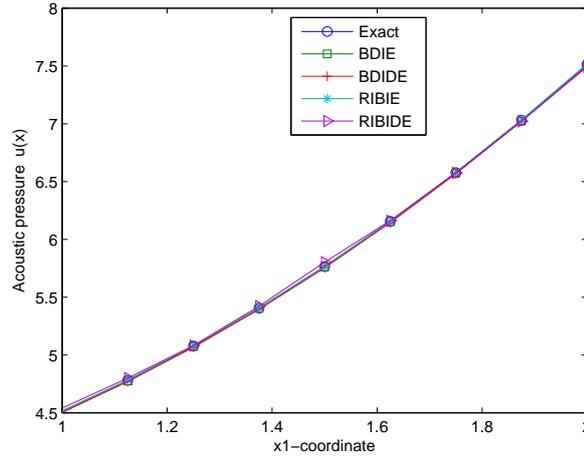
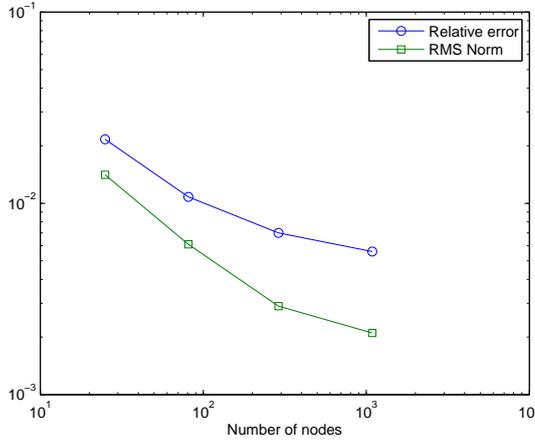
Figure 7.4: Acoustic pressure distribution along the line $x_2 = 1.875$ 

Figure 7.5: Relative and RMS errors for RIBIDE method for test 2

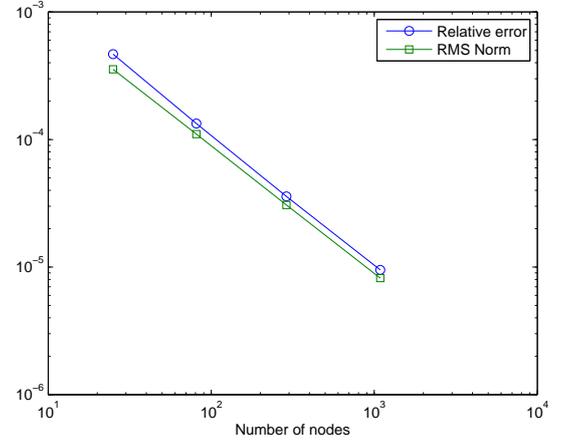


Figure 7.6: Relative and RMS errors for RIBIE method for test 2

Test 3 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, where $k(x) = 1$, for $x \in \bar{\Omega}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = \exp(x_1 + x_2)(6x_1 + 3x_1^2 + 6x_2 + 3x_2^2) + x_1^3 + x_2^3$ and the boundary conditions:

$$\bar{u}(x) = 8 + x_1^3, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 27 + x_1^3, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = \exp(x_1 + x_2)(3x_1^2 n_1(x) + 3x_2^2 n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

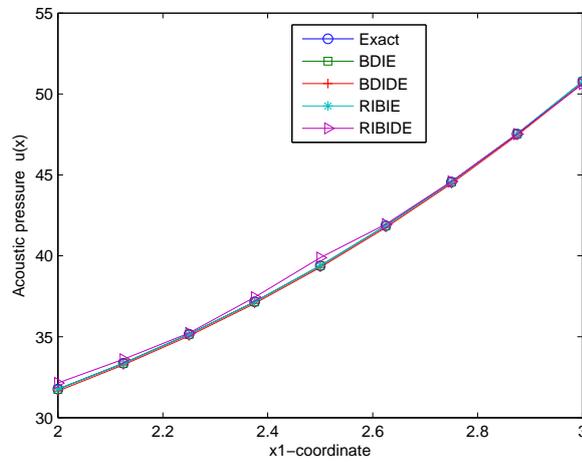
The exact solution for this problem is $u_{exact}(x) = x_1^3 + x_2^3$, $x \in \bar{\Omega}$.

Table 7.3 lists the computed values of $u(x)$ along the middle line of the plate using RIBIDE, RIBIE, BDIDE and BDIE, while Fig.7.7 shows the variation of $u(x)$ along the line $x_2 = 2.875$.

It can be seen from tables 7.1-7.3 and Figs. 7.1, 7.4 and 7.7 that both the RIBIE and

Table 7.3: Computed acoustic pressure along line $x_2 = 2.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
2	23.54227795	24.08753950	23.60035796	23.62472805	23.62500000
2.125	25.14006793	25.58360246	25.20769297	25.22737334	25.22070313
2.25	26.93069620	27.28226866	27.00518929	27.02088546	27.01562500
2.375	28.93457973	29.20969252	29.01338124	29.02608159	29.02148438
2.5	31.16338664	31.36617250	31.24439482	31.25465503	31.25000000
2.625	33.62869650	33.75998894	33.71028780	33.71823012	33.71289063
2.750	36.34196670	36.40889990	36.42304776	36.42847247	36.42187500
2.875	39.31462484	39.31873001	39.39451701	39.39688883	39.38867188
3	42.54759473	42.52953357	42.62601667	42.62475880	42.62500000

Figure 7.7: Acoustic pressure distribution along the line $x_2 = 2.875$

RIBIDE methods are able to generate accurate solutions in good agreement with BDIE and BDIDE results.

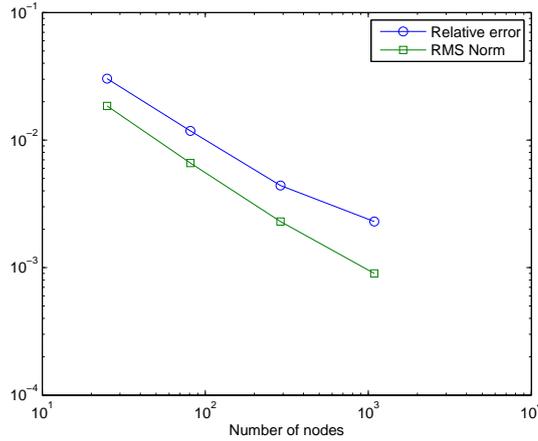


Figure 7.8: Relative and RMS errors for RIBIDE method for test 3

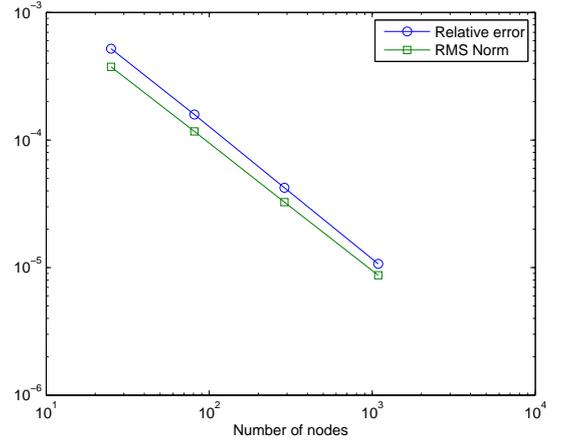


Figure 7.9: Relative and RMS errors for RIBIE method for test 3

7.6.2 Numerical results when $a(x)$ constant and $k(x)$ variable

In this case, when the parameter $a(x)$ is constant, the remainder $\tilde{R}(x, y)$ in Eq.(7.1) will be zero.

Test 4 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 0 \leq x_1, x_2 \leq 1\}$, where $k(x) = x_1^3 + x_2^3$, for $x \in \bar{\Omega}$, $a(x) = 1$, $f(x) = (x_1^3 + x_2^3)(x_1 + x_2)$ and the boundary conditions:

$$\bar{u}(x) = x_1, \text{ for } x_2 = 0; \quad 0 \leq x_1 \leq 1,$$

$$\bar{u}(x) = 1 + x_1, \text{ for } x_2 = 1; \quad 0 \leq x_1 \leq 1,$$

$$\bar{t}(x) = n_1(x) + n_2(x), \text{ for } x_1 = 0 \text{ or } x_1 = 1; \quad 0 \leq x_2 \leq 1.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

Table 7.4: Computed acoustic pressure along line $x_2 = 0.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
0	0.50001354	0.49808573	0.50001417	0.50001295	0.50000000
0.125	0.62500238	0.62295805	0.62500297	0.62499982	0.62500000
0.25	0.75000135	0.74742745	0.75000181	0.74999816	0.75000000
0.375	0.87500056	0.87244433	0.87500085	0.87499711	0.87500000
0.5	0.99999989	0.99766609	0.99999998	0.99999501	1.00000000
0.625	1.12499922	1.12187828	1.12499910	1.12499501	1.12500000
0.750	1.24999839	1.24604751	1.24999810	1.24999412	1.25000000
0.875	1.37499729	1.37172154	1.37499688	1.37499191	1.37500000
1	1.49998607	1.49586330	1.49998561	1.49998264	1.50000000

Table 7.4 lists the computed values of $u(x)$ along the middle line of the plate using RIBIDE,

RIBIE, BDIDE and BDIE, while Fig.7.10 shows the variation of $u(x)$ along the line $x_2 = 0.875$.

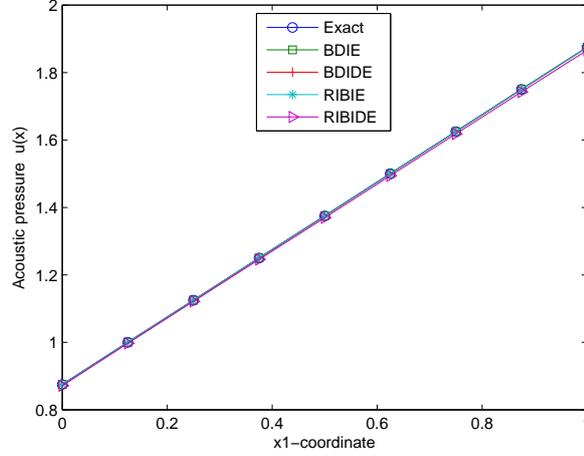


Figure 7.10: Acoustic pressure distribution along the line $x_2 = 0.875$

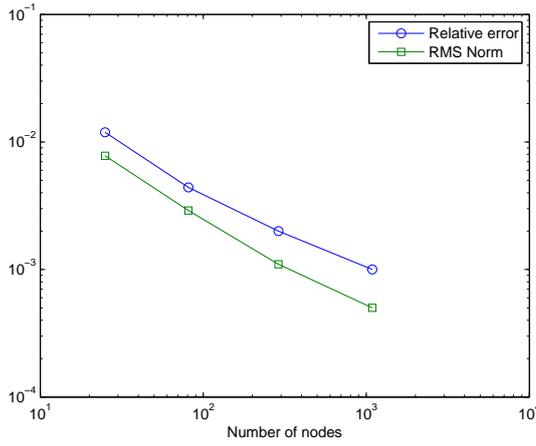


Figure 7.11: Relative and RMS errors for RIBIDE method for test 4

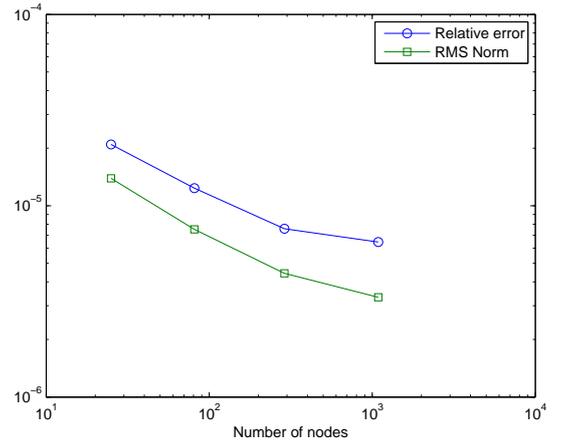


Figure 7.12: Relative and RMS errors for RIBIE method for test 4

Test 5 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, where $k(x) = \cos(x_1) + \cos(x_2)$, for $x \in \bar{\Omega}$,

$a(x) = 1$, $f(x) = (\cos(x_1) + \cos(x_2))(x_1 + x_2)$ and the boundary conditions:

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 3 + x_1, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = n_1(x) + n_2(x), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

Table 7.5 lists the computed values of $u(x)$ along the middle line of the plate using RIBIDE, RIBIE, BDIDE and BDIE, while Fig.7.13 shows the variation of $u(x)$ along the line $x_2 = 2.875$.

Table 7.5: Computed acoustic pressure along line of $x_2 = 2.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
2	4.50001317	4.50314478	4.50001361	4.50001702	4.50000000
2.125	4.62500210	4.63140975	4.62500249	4.62500705	4.62500000
2.25	4.75000120	4.76005830	4.75000149	4.75000504	4.75000000
2.375	4.87500055	4.88111739	4.87500069	4.87500317	4.87500000
2.5	5.00000005	5.00489097	4.99999999	5.00000137	5.00000000
2.625	5.12499955	5.13117889	5.12499932	5.12500185	5.12500000
2.750	5.24999893	5.25665463	5.24999854	5.25000135	5.25000000
2.875	5.37499806	5.37992272	5.37499757	5.37500057	5.37500000
3	5.49998701	5.49941772	5.49998648	5.49998770	5.50000000

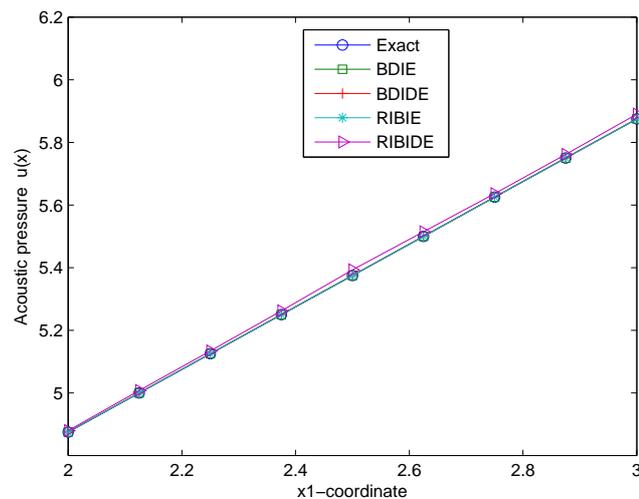


Figure 7.13: Acoustic pressure distribution along the line $x_2 = 2.875$

From tables 7.4-7.5 and Figs. 7.10 and 7.13, it is clear that both the RIBIE and RIBIDE methods are able to generate accurate solutions in good agreement with BDIE and BDIDE results.

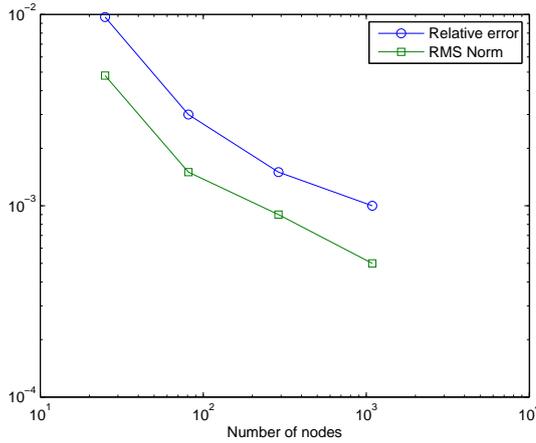


Figure 7.14: Relative and RMS errors for RIBIDE method for test 5

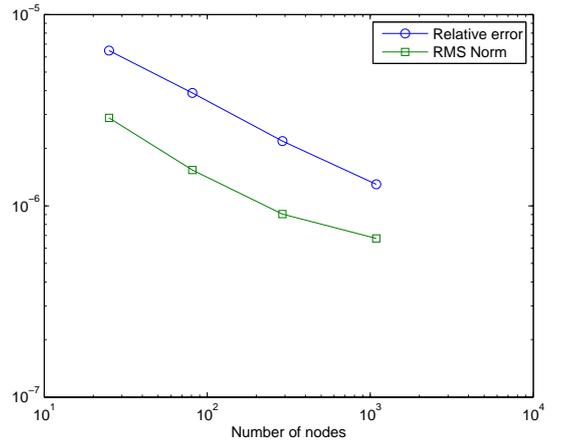


Figure 7.15: Relative and RMS errors for RIBIE method for test 5

7.6.3 Numerical results when both $a(x)$ and $k(x)$ variable

In this final case, when both the material parameter $a(x)$ and wave number $k(x)$ are variable, the parametrix is adopted. Let us consider some test examples to assess the accuracy of the RIBIDE/RIBIE and BDIDE/BDIE methods.

Test 6 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, where $k(x) = x_1 + x_2$, for $x \in \bar{\Omega}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = 2(\exp(x_1 + x_2)) + (x_1 + x_2)^2$ and the boundary conditions:

$$\bar{u}(x) = 1 + x_1, \text{ for } x_2 = 1; \quad 1 \leq x_1 \leq 2,$$

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 1 \leq x_1 \leq 2,$$

$$\bar{t}(x) = (\exp(x_1 + x_2))(n_1(x) + n_2(x)), \text{ for } x_1 = 1 \text{ or } x_1 = 2; \quad 1 \leq x_2 \leq 2.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

Table 7.6: Computed acoustic pressure along line of $x_2 = 1.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
1	2.50000827	2.53200759	2.49975098	2.49997915	2.50000000
1.125	2.62499815	2.65179178	2.62485102	2.62497797	2.62500000
1.25	2.74999800	2.77092058	2.74992766	2.74998578	2.75000000
1.375	2.87499777	2.89109656	2.87499185	2.87499340	2.87500000
1.5	2.99999737	3.01111090	3.00004969	3.00000009	3.00000000
1.625	3.12499670	3.13157705	3.12510806	3.12500611	3.12500000
1.750	3.24999566	3.25284233	3.25017286	3.25001177	3.25000000
1.875	3.37499422	3.37410114	3.37524777	3.37501589	3.37500000
2	3.49998273	3.49815802	3.50034354	3.50000900	3.50000000

Table 7.6 lists the computed values of $u(x)$ along the middle line of the plate using RIBIDE, RIBIE, BDIDE and BDIE, while Fig.7.16 shows the variation of $u(x)$ along the line $x_2 = 1.875$.

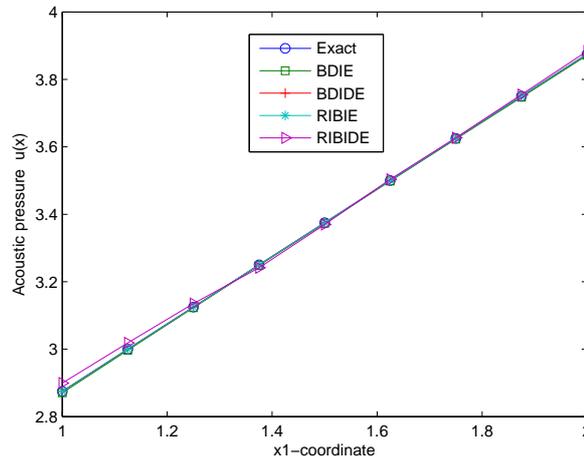


Figure 7.16: Acoustic pressure distribution along the line $x_2 = 1.875$

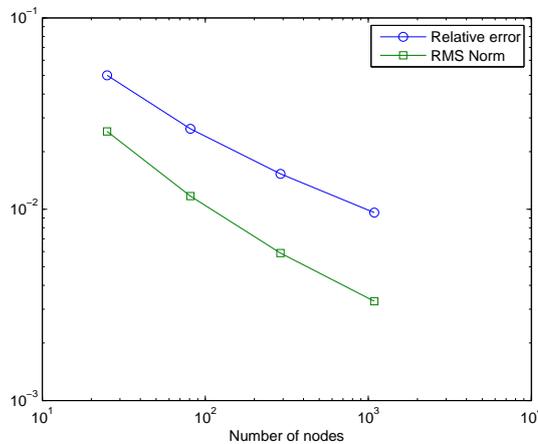


Figure 7.17: Relative and RMS errors for RIBIDE method for test 6

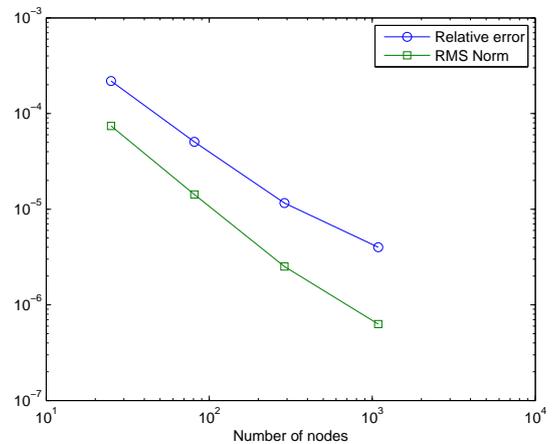


Figure 7.18: Relative and RMS errors for RIBIE method for test 6

Test 7 :

Square domain $\bar{\Omega} = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 2\}$, where $k(x) = \sin(x_1) + \sin(x_2)$, for $x \in \bar{\Omega}$, $a(x) = \exp(x_1 + x_2)$, $f(x) = (2(\exp(x_1 + x_2))(2 + x_1 + x_2)) + (\sin(x_1) + \sin(x_2))(x_1^2 + x_2^2)$ and the boundary conditions:

$$\bar{u}(x) = 1 + x_1, \text{ for } x_2 = 1; \quad 1 \leq x_1 \leq 2,$$

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 1 \leq x_1 \leq 2,$$

$\bar{t}(x) = 2(\exp(x_1 + x_2))(x_1 n_1(x) + x_2 n_2(x))$, for $x_1 = 1$ or $x_1 = 2$; $1 \leq x_2 \leq 2$.

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

Table 7.7 lists the computed values of $u(x)$ along the middle line of the plate using RIBIDE, RIBIE, BDIDE and BDIE, while Fig.7.19 shows the variation of $u(x)$ along the line $x_2 = 1.875$.

Table 7.7: Computed acoustic pressure along line of $x_2 = 1.5$

x_1	BDIDE	RIBIDE	BDIE	RIBIE	Exact
1	3.23907298	3.32246311	3.24875134	3.24996383	3.25000000
1.125	3.50510973	3.57286266	3.51593634	3.51652295	3.51562500
1.25	3.80153097	3.85513479	3.81314959	3.81320922	3.81250000
1.375	4.12948095	4.17133413	4.14165412	4.14125138	4.14062500
1.5	4.48895221	4.51957143	4.50149612	4.50063760	4.50000000
1.625	4.87992777	4.89980552	4.89272494	4.89135856	4.89062500
1.750	5.30237464	5.31223044	5.31538789	5.31340484	5.31250000
1.875	5.75625542	5.75675965	5.76951472	5.76675756	5.76562500
2	6.24013358	6.23692919	6.25374388	6.25000002	6.25000000

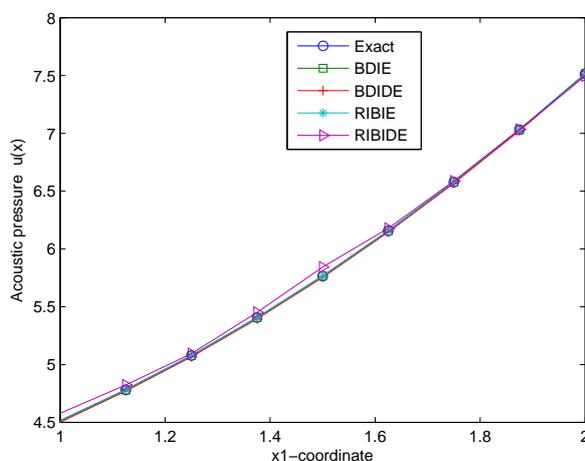


Figure 7.19: Acoustic pressure distribution along the line $x_2 = 1.875$

It can be seen from tables 7.6 and 7.7 and Figs. 7.16-7.19 that both the RIBIE and RIBIDE methods are able to generate accurate solutions in good agreement with the BDIE and BDIDE results. It is important to point out that the numerical integration of the RIM in Matlab is very fast and can save a substantial amount of computational time in comparison to both BDIDE and BDIE. It is noticed that the RIBIE produces better results than RIBIDE in all tests. Moreover, the relative and RMS errors in tests 1-7 show that both the RIBIE and RIBIDE methods are convergent with mesh refinement

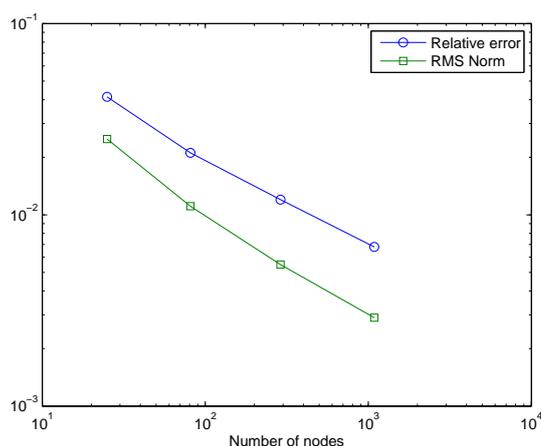


Figure 7.20: Relative and RMS errors for RIBIDE method for test 7

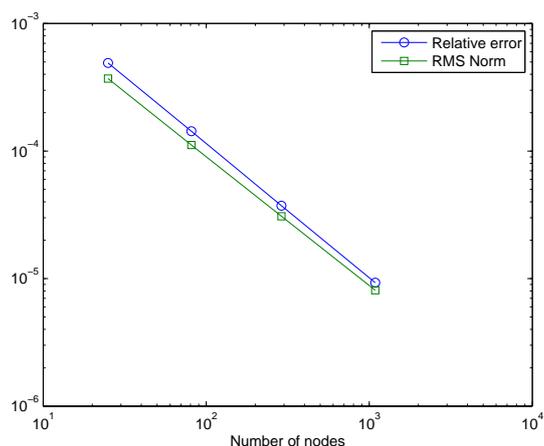


Figure 7.21: Relative and RMS errors for RIBIE method for test 7

and, in general, the RMS error is lower than the relative error as expected. It can also be clearly seen that the convergence of RIBIE is much faster than RIBIDE, which is a similar behaviour to the BDIE and BDIDE presented in chapter four in which cell-integration is used.

7.7 Conclusion

In this chapter, the RIBIE/RIBIDE formulations are derived and implemented for solving the two-dimensional Helmholtz equation with variable coefficients. Three different cases have been solved; when the parameter $a(x)$ is variable (with constant or variable wave number k), a parametrization is adopted in the formulation. However, when the parameter is constant (with variable wave number), the standard fundamental solution for the Laplace equation is used.

Using the RIM, it is possible to transform the domain integrals that appear in both BDIE and BDIDE methods derived and implemented in chapter 4, into equivalent boundary integrals, thus retaining the boundary-only character of the standard BEM. Moreover, the RIM removes the weak singularities appearing in both domain integrals, simplifying and speeding up the calculation of the integrals. Numerical results showed that both the RIBIE and RIBIDE methods are able to generate accurate solutions in good agreement with BDIE and BDIDE results.

Chapter 8

RIBIE and RIBIDE for diffusion equation with variable coefficients

8.1 Introduction

The diffusion equation models transient heat transfer and other similar problems [7, 58]. Several numerical techniques have been proposed to generate boundary integral representations for the diffusion equation. The dual reciprocity boundary element method (DRBEM), initially applied to transient heat conduction problems by Wrobel et al. [59], interprets the time derivative in the diffusion equation as a body force and employs the fundamental solution to Laplace's equation to generate a boundary integral equation. The BEM and the DRBEM are discussed in detail in [2]. Also, Wrobel and Brebbia [60] presented an extension of the DRBEM to deal with nonlinear diffusion problems in which the thermal conductivity, specific heat, and density coefficients are all functions of temperature.

In [61] the authors applied a meshless method as a boundary-only formulation for transient heat conduction with a heat source. The transformation of the domain integral into the corresponding boundary integral is carried out using Green's theorem.

Recent work by Yang and Gao [58] adopted the Green's function for the Laplace equation in deriving normalized boundary-domain integral equations for time-dependent problems with varying heat conductivities. The authors argued that, unlike the standard BEM, considering the product of variable coefficients by the unknown functions as a new variable can provide accurate results. Then, the RIM is employed to convert the resulting domain

integrals into equivalent boundary integrals. However, due to the way the inner radial integral was calculated, the formulations still required to calculate the integral inside the domain and are restricted to star-shaped domains.

In this chapter, a new type of boundary-only integral equation technique is developed for non-homogeneous transient heat conduction problems with variable coefficients based on the use of a parametrix. It is shown in this chapter that both the parametrix and the standard fundamental solution for Laplace's equation lead to the same form of Green's third identity when the variable coefficients appear in the second-order linear parabolic PDE. Moreover, in the last decade, the mathematical theory and equivalence of the BDIE and BDIDE to the steady-state heat conduction with variable coefficients using parametrix was proved along with their solvability, solution uniqueness, and the operator invertibility in appropriate spaces [62, 63].

The RIM is used in this chapter to convert the domain integrals appearing in both BDIE and BDIDE to equivalent boundary integrals. Moreover, the radial integral is calculated along the boundary only. For domain integrals consisting of known functions the transformation is straightforward, while for domain integrals that include unknown variables the transformation is accomplished with the use of augmented RBFs, as in chapters 6 and 7. The most attractive feature of the method is that the transformations are very simple and have similar forms for both 2D and 3D problems.

8.2 Reduction of diffusion equation to a BDIE/BDIDE

Let us consider the following diffusion equation in an isotropic non-homogeneous medium for a two-dimensional body Ω , with prescribed temperature $\bar{u}(x)$ on part $\partial_D\Omega$ of the boundary $\partial\Omega$ and prescribed heat flux $\bar{q}(x)$ on the remaining $\partial_N\Omega$ part of $\partial\Omega$, i.e. we consider the second-order linear parabolic PDE,

$$\sum_{i=1}^2 \frac{\partial}{\partial x_i} \left[a(x) \frac{\partial u(x,t)}{\partial x_i} \right] = f(x,t) + D(x,t) \frac{\partial u(x,t)}{\partial t}, \quad x \in \Omega \quad (8.1)$$

with the initial-boundary conditions

$$u(x_1, x_2, 0) = u_0(x), \quad \text{for } x = (x_1, x_2) \in \Omega, \quad (8.2)$$

$$u(x_1, x_2, t) = \bar{u}(x_1, x_2, t), \quad \text{for } (x_1, x_2) \in \partial_D \Omega, \quad t > 0, \quad (8.3)$$

$$Tu(x_1, x_2, t) = \bar{q}(x_1, x_2, t), \quad \text{for } (x_1, x_2) \in \partial_N \Omega, \quad t > 0, \quad (8.4)$$

where Ω is a bounded domain, $u(x, t)$ the temperature, $a(x)$ a known variable thermoconductivity coefficient, $f(x, t)$ a known heat source, T a surface flux operator, $x = (x_1, x_2)$, $[Tu](x, t) := a(x) \frac{\partial u(x, t)}{\partial n(x)}$, $n(x)$ the external normal vector to the boundary $\partial\Omega$, u_0 , \bar{u} and \bar{q} are known functions, and $D(x, t) = \rho C$, ρ is the mass density and C is the specific heat. The Green formula for the differential operator L has the form (3.6).

Let L be a linear operator and $G(x, y)$ its fundamental solution, i.e.

$$L_x G(x, y) = \delta(x - y),$$

where δ is the Dirac delta function. Also, the fundamental solution used is the same as for the Laplace's equation, given by:

$$G(x, y) = \frac{1}{2\pi} \ln |x - y| = \tilde{P}(x, y). \quad (8.5)$$

Then, one could take $v(x) = G(x, y)$, identify $u(x, t)$ with a solution of Eq.(8.1), and thus arrive at the third Green identity

$$\begin{aligned} c(y)u(y, t) - \int_{\partial\Omega} [u(x, t)T_x G(x, y) - G(x, y)Tu(x, t)]d\Gamma(x) = \\ = \int_{\Omega} G(x, y)f(x, t)d\Omega(x) + \int_{\Omega} D(x, t)G(x, y)\frac{\partial u(x, t)}{\partial t}d\Omega(x), \end{aligned} \quad (8.6)$$

where $c(y)$ is given by Eq. (2.18).

Substituting the boundary condition in the Green identity Eq.(8.6) and applying it for $y \in \partial\Omega$, we arrive at a direct BIE [2, 6].

For partial differential operators with variable coefficients, like L in Eq.(8.1), a fundamental solution is generally not available in explicit form. In order to get the third Green identity corresponding to the variable coefficients operator L in Eq.(8.1), there are two approaches available in literature.

The first approach is to use a parametrix $P(x, y)$, which is often available, as discussed in previous chapters, which for two-dimensional problems will be given by Eq. (3.9).

The remainder $R(x, y)$ will then be given by Eq. (3.10).

Substituting $P(x, y)$ for $v(x)$ in Eq.(3.6) and taking $u(x, t)$ as a solution to Eq.(8.1), we obtain the integral equality,

$$\begin{aligned} c(y)u(y, t) - \int_{\partial\Omega} [u(x, t)T_x P(x, y) - P(x, y)Tu(x, t)]d\Gamma(x) + \int_{\Omega} R(x, y)u(x, t)d\Omega(x) + \\ = \int_{\Omega} P(x, y)f(x, t)d\Omega(x) + \int_{\Omega} D(x, t)P(x, y)\frac{\partial u(x, t)}{\partial t}d\Omega(x). \end{aligned} \quad (8.7)$$

Now, we can multiply both sides of Eq.(8.7) by $a(y)$ to obtain:

$$\begin{aligned} a(y)c(y)u(y, t) - \int_{\partial\Omega} [u(x, t)T_x G(x, y) - G(x, y)Tu(x, t)]d\Gamma(x) + \int_{\Omega} \tilde{R}(x, y)u(x, t)d\Omega(x) = \\ = \int_{\Omega} G(x, y)f(x, t)d\Omega(x) + \int_{\Omega} D(x, t)G(x, y)\frac{\partial u(x, t)}{\partial t}d\Omega(x), \end{aligned} \quad (8.8)$$

where

$$G(x, y) = a(y)P(x, y) = \frac{1}{2\pi} \ln |x - y|,$$

$$\tilde{R}(x, y) = a(y)R(x, y) = \sum_{i=1}^2 \frac{x_i - y_i}{2\pi|x-y|^2} \frac{\partial a(x)}{\partial x_i}.$$

The second alternative approach is to use the fundamental solution for the Laplace equation given in Eq.(8.5) as a weighting function to Eq.(8.1), leading to

$$\begin{aligned} \int_{\Omega} \frac{\partial}{\partial x_i} \left[a(x) \frac{\partial u(x, t)}{\partial x_i} \right] G(x, y) d\Omega(x) = \int_{\Omega} f(x, t) G(x, y) d\Omega(x) + \\ + \int_{\Omega} D(x, t) \frac{\partial u(x, t)}{\partial t} G(x, y) d\Omega(x), \quad x \in \Omega. \end{aligned} \quad (8.9)$$

Using integration by parts, the first domain integral in Eq.(8.9) can be written as,

$$\begin{aligned} \int_{\Omega} \frac{\partial}{\partial x_i} \left[a(x) \frac{\partial u(x, t)}{\partial x_i} \right] G(x, y) d\Omega(x) = \int_{\Omega} \frac{\partial}{\partial x_i} \left[G(x, y) a(x) \frac{\partial u(x, t)}{\partial x_i} \right] d\Omega(x) - \\ - \int_{\Omega} \frac{\partial G(x, y)}{\partial x_i} a(x) \frac{\partial u(x, t)}{\partial x_i} d\Omega(x). \end{aligned} \quad (8.10)$$

The Gauss' divergence theorem can now be applied to the first domain integral on the

right-hand side, leading to

$$\begin{aligned} \int_{\Omega} \frac{\partial}{\partial x_i} \left[G(x, y) a(x) \frac{\partial u(x, t)}{\partial x_i} \right] d\Omega(x) &= \int_{\partial\Omega} G(x, y) a(x) \frac{\partial u(x, t)}{\partial x_i} n_i(x) d\Gamma(x) = \\ &= \int_{\partial\Omega} G(x, y) \frac{\partial u(x, t)}{\partial n(x)} d\Gamma(x). \end{aligned} \quad (8.11)$$

The second domain integral on the right-hand side of Eq.(8.10) can be manipulated as,

$$\begin{aligned} - \int_{\Omega} \frac{\partial G(x, y)}{\partial x_i} \left[a(x) \frac{\partial u(x, t)}{\partial x_i} \right] d\Omega(x) &= - \int_{\Omega} \frac{\partial G(x, y)}{\partial x_i} \frac{\partial}{\partial x_i} [a(x)u(x, t)] d\Omega(x) + \\ &+ \int_{\Omega} \frac{\partial G(x, y)}{\partial x_i} \frac{\partial a(x)}{\partial x_i} u(x, t) d\Omega(x). \end{aligned} \quad (8.12)$$

The first domain integral on the right-hand side of Eq.(8.12) can be manipulated as before,

$$\begin{aligned} - \int_{\Omega} \frac{\partial G(x, y)}{\partial x_i} \frac{\partial}{\partial x_i} [a(x)u(x, t)] d\Omega(x) &= - \int_{\Omega} \frac{\partial}{\partial x_i} \left[\frac{\partial G(x, y)}{\partial x_i} a(x)u(x, t) \right] d\Omega(x) \\ &+ \int_{\Omega} \frac{\partial}{\partial x_i} \left[\frac{\partial G(x, y)}{\partial x_i} \right] a(x)u(x, t) d\Omega(x). \end{aligned} \quad (8.13)$$

Due to the fact that the fundamental solution is singular, the source point y is surrounded by a small circle of radius ϵ , and then the integrals are examined in the limit as $\epsilon \rightarrow 0$, as discussed in detail in section 2.2. The second domain integral on the right-hand side of Eq.(8.13) vanishes, since the source point is excluded from the integration domain. During the limit procedure the first domain integral on the right-hand side of Eq.(8.13) produces an additional term $a(y)c(y)u(y)$.

The Gauss' divergence theorem can now be applied to the first domain integral on the right-hand side, leading to

$$\begin{aligned} - \int_{\Omega} \frac{\partial}{\partial x_i} \left[\frac{\partial G(x, y)}{\partial x_i} a(x)u(x, t) \right] d\Omega(x) &= - \int_{\partial\Omega} \frac{\partial G(x, y)}{\partial x_i} n_i(x) a(x)u(x, t) d\Gamma(x) = \\ &= a(y)c(y)u(y) - \int_{\partial\Omega} \frac{\partial G(x, y)}{\partial n(x)} u(x, t) d\Gamma(x). \end{aligned} \quad (8.14)$$

Substituting Eqs.(8.10-8.14) in Eq.(8.9), we get the same third Green identity as in Eq.(8.8).

Remark 8.1: It is important to point out, that using both a parametrix and the funda-

mental solution for the Laplace equation for the second-order linear parabolic PDE with variable coefficients leads to the same third Green identity. This statement is also valid for the previous problems of heat conduction and wave propagation.

The identity (8.8) can be used for formulating either a BDIE or BDIDE, with respect to u and its derivatives. We still prefer to call the new formulations in the next sections BDIE and BDIDE even if the identity (8.8) has a new domain integral with a time derivative for simplicity. Let us consider the two forms below.

8.2.1 Boundary-domain integral equation (BDIE)

Substituting the boundary conditions (8.3) and (8.4) into (8.8), introducing a new variable $q(x, t) = Tu(x, t)$ for the unknown flux on $\partial_D\Omega$ and using Eq.(8.8) at $y \in \Omega \cup \partial\Omega$ reduces the Eq.(8.1) to the following BDIE for $u(x, t)$ at $x \in \Omega \cup \partial_N\Omega$ and $q(x, t)$ at $x \in \partial_D\Omega$,

$$\begin{aligned} c^0(y)u(y, t) - \int_{\partial_N\Omega} u(x, t)T_x G(x, y)d\Gamma(x) + \int_{\partial_D\Omega} G(x, y)q(x, t)d\Gamma(x) + \\ + \int_{\Omega} \tilde{R}(x, y)u(x, t)d\Omega(x) = \Psi^0(y, t), \quad y \in \Omega \cup \partial\Omega \end{aligned} \quad (8.15)$$

$$\Psi^0(y, t) := [c^0(y) - a(y)c(y)]\bar{u}(y, t) + \Psi(y, t), \quad (8.16)$$

$$\begin{aligned} \Psi(y, t) := \int_{\partial_D\Omega} \bar{u}(x, t)T_x G(x, y)d\Gamma(x) - \int_{\partial_N\Omega} G(x, y)\bar{q}(x, t)d\Gamma(x) + \\ + \int_{\Omega} G(x, y)f(x, t)d\Omega(x) + \int_{\Omega} D(x, t)\frac{\partial u(x, t)}{\partial t}G(x, y)d\Omega(x), \end{aligned} \quad (8.17)$$

and $c^0(y)$ is given by (6.8).

8.2.2 Boundary-domain integro-differential equation (BDIDE)

Using another approach, we can substitute the boundary conditions (8.3) and (8.4) into (8.8) but leave T as a differential flux operator acting on u on the Dirichlet boundary $\partial_D\Omega$

8.3. Transformation of domain integrals to the boundary using RIM 141

and use the following BDIDE at $y \in \Omega \cup \partial_N \Omega$,

$$a(y)c(y)u(y, t) - \int_{\partial_N \Omega} u(x, t)T_x G(x, y)d\Gamma(x) + \int_{\partial_D \Omega} G(x, y)Tu(x, t)d\Gamma(x) + \int_{\Omega} \tilde{R}(x, y)u(x, t)d\Omega(x) = \Psi(y, t), \quad y \in \Omega \cup \partial_N \Omega \quad (8.18)$$

where $\Psi(y, t)$ is given by Eq.(8.17).

8.3 Transformation of domain integrals to the boundary using RIM

In this section, the RIM discussed in chapter five is used to transform the domain integrals appearing in equations (8.15) and (8.18) into boundary integrals.

8.3.1 RIM formulation for domain integrals with known integrand

Both Eq.(8.15) and Eq.(8.18) have domain integrals coming from the known function $f(x, t)$. The RIM can be directly used to convert these domain integrals to the boundary, as discussed in subsection 5.2.1. This leads to

$$\int_{\Omega} G(x, y)f(x, t)d\Omega(x) = \int_{\partial \Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x, t)d\Gamma(x) \quad (8.19)$$

where

$$F(x, t) = \int_0^1 G(x, y)f(y_1 + r_{,1}rs, y_2 + r_{,2}rs, t)r^2sds \quad (8.20)$$

The integral in Eq.(8.20) can be calculated analytically for many different functions, and numerically as discussed in previous chapters. Also, due to the radial integral in Eq.(8.20), the weak singularity coming from the fundamental solution is removed.

8.3.2 RIM formulation for domain integrals with unknown integrand

Both Eq.(8.15) and Eq.(8.18) have two domain integrals with unknown integrands. The first domain integral on the left-hand side comes from the term $\tilde{R}(x, y)$ and can be treated exactly as in chapter 6 (subsection 6.3.1). However, the right-hand side domain integral in Eqs.(8.15) and (8.18) has the unknown time-derivative $\frac{\partial u(x,t)}{\partial t}$ in the last domain integral of Eq. (8.17), and the RIM cannot be directly used. We adopt a similar procedure as in previous chapters (5, 6 and 7), as follows:

Let us approximate the variation of $u(x, t)$ in the following way:

$$\frac{\partial u(x, t)}{\partial t} = \sum_{k=1}^M \beta_k \phi_k(R) + c_1 x_1 + c_2 x_2 + c_3. \quad (8.21)$$

The following equilibrium conditions have to be satisfied , as before:

$$\sum_{k=1}^M \beta_k = \sum_{k=1}^M \beta_k x_{1k} = \sum_{k=1}^M \beta_k x_{2k} = 0 \quad (8.22)$$

Substituting Eq.(8.21) into the last domain integrals of Eq. (8.17), we obtain:

$$\begin{aligned} \int_{\Omega} D(x, t) \frac{\partial u(x, t)}{\partial t} G(x, y) d\Omega(x) &= \sum_{k=1}^M \beta_k \int_{\Omega} D(x, t) G(x, y) \phi_k(R) d\Omega(x) + \\ &+ c_1 \int_{\Omega} D(x, t) G(x, y) x_1 d\Omega(x) + c_2 \int_{\Omega} D(x, t) G(x, y) x_2 d\Omega(x) + \\ &+ c_3 \int_{\Omega} D(x, t) G(x, y) d\Omega(x). \end{aligned} \quad (8.23)$$

It is very important before applying the RIM that the coordinates x_1 and x_2 appearing in Eq.(8.23) are expressed in terms of the distance r using Eq.(5.11). Now, applying the RIM to each domain integral in Eq.(8.23) leads to

$$\int_{\Omega} D(x, t) \frac{\partial u(x)}{\partial t} G(x, y) d\Omega(x) = \int_{\partial\Omega} w(x, t) d\Gamma(x)$$

where

$$\int_{\partial\Omega} w(x, t) d\Gamma(x) = \sum_{k=1}^M \beta_k \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} \acute{F}1(x, t) d\Gamma(x) + c_1 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} \acute{F}2(x, t) d\Gamma(x) + c_2 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} \acute{F}3(x, t) d\Gamma(x) + c_3 \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} \acute{F}4(x, t) d\Gamma(x), \quad (8.24)$$

where

$$\acute{F}1(x, t) = \int_0^1 D(x, t) G(x, y) \phi(R) r^2 s ds \quad (8.25a)$$

$$\acute{F}2(x, t) = \int_0^1 D(x, t) G(x, y) (y_1 + r_{,1} r s) r^2 s ds \quad (8.25b)$$

$$\acute{F}3(x, t) = \int_0^1 D(x, t) G(x, y) (y_2 + r_{,2} r s) r^2 s ds \quad (8.25c)$$

$$\acute{F}4(x, t) = \int_0^1 D(x, t) G(x, y) r^2 s ds \quad (8.25d)$$

After numerical integration, the unknown coefficients $\beta_k, k = 1, \dots, M, c_1, c_2$ and c_3 , can be calculated following the procedures discussed in section 5.2, in which the $\alpha_k, k = 1, \dots, M$, need to be replaced by $\beta_k, k = 1, \dots, M$ and u by $\frac{\partial u}{\partial t}$.

8.4 The radial integration boundary integral and integro-differential equations

Eqs. (6.14)-(6.15), (8.19)-(8.20) and (8.24)-(8.25) can now be substituted in both BDIE in Eq.(8.15) and BDIDE in Eq.(8.18), and this leads to the following expressions.

8.4.1 The radial integration boundary integral equation (RI-BIE)

$$c^0(y)u(y, t) - \int_{\partial_N\Omega} u(x, t)T_x G(x, y)d\Gamma(x) + \int_{\partial_D\Omega} G(x, y)q(x, t)d\Gamma(x) + \int_{\partial\Omega} h(x)d\Gamma(x) = \Psi^0(y, t), \quad y \in \Omega \cup \partial\Omega \quad (8.26)$$

$$\Psi^0(y, t) := [c^0(y) - a(y)c(y)]\bar{u}(y, t) + \tilde{\Psi}(y, t), \quad (8.27)$$

$$\tilde{\Psi}(y, t) := \int_{\partial_D\Omega} \bar{u}(x, t)T_x G(x, y)d\Gamma(x) - \int_{\partial_N\Omega} G(x, y)\bar{q}(x, t)d\Gamma(x) + \int_{\partial\Omega} w(x, t)d\Gamma(x) + \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x, t)d\Gamma(x) \quad (8.28)$$

where $c^0(y)$, $F(x, t)$, $\int_{\partial\Omega} h(x)d\Gamma(x)$ and $\int_{\partial\Omega} w(x, t)d\Gamma(x)$ are given in Eqs.(6.8), (8.20), (6.14)-(6.15), and (8.24)-(8.25), respectively.

8.4.2 The radial integration boundary integro-differential equation (RIBIDE)

$$a(y)c(y)u(y, t) - \int_{\partial_N\Omega} u(x, t)T_x G(x, y)d\Gamma(x) + \int_{\partial_D\Omega} G(x, y)Tu(x, t)d\Gamma(x) + \int_{\partial\Omega} h(x)d\Gamma(x) = \tilde{\Psi}(y, t), \quad y \in \Omega \cup \partial_N\Omega \quad (8.29)$$

where $\tilde{\Psi}(y, t)$ is given in Eq. (8.28). It can be seen clearly from both RIBIE in Eq.(8.26) and RIBIDE in Eq.(8.29) that all integrations are now carried out only on the boundary, with no domain integrals.

8.4.3 Discretisation of the RIBIE

The RIBIE formulation employs mixed boundary elements with linear u and constant q to avoid the discontinuities of q at corner points. In this case, collocation was taken at the end points of each boundary element, since our previous chapters have shown that end-node collocation generally provides higher accuracy than mid-node collocation.

Let J be the total number of nodes x^i , $i = 1, \dots, J$, at the end points of elements, from which there are J_D nodes on $\partial_D\Omega$. Thus, the values of u at any point on the element can be defined in terms of their nodal values and two linear interpolation functions $\Psi^1(t)$ and $\Psi^2(t)$ given in Eq. (2.27).

To obtain a system of linear algebraic equations from the RIBIE (8.26), we collocate at the nodes x^i , $i = 1, \dots, J$. We can also use an interpolation of $q(x, t) = (Tu)(x^j, t)$ along boundary nodes belonging to $x^j \in \partial_D\Omega$

$$q(x, t) = \sum_{x^j \in \partial_D\Omega} q(x^j, t)v_j(x), \quad x \in \partial_D\Omega \quad (8.30)$$

Here, $v_j(x)$ are boundary shape functions, taken now as constant. Therefore, $v_j(x)$ will be equal to 1 at $x^j \in \partial_D\Omega$ and $v_j(x) = 0$ if $x^j \notin \partial_D\Omega$. Substituting the interpolations (8.30) and (2.27) in the RIBIE (8.26) and applying the collocation method, we arrive at the following system of J linear algebraic equations for J unknowns $u(x^j, t)$, $x^j \in \Omega \cup \partial_N\Omega$ and $q(x^j, t) = (Tu)(x^j, t)$, $x^j \in \partial_D\Omega$,

$$\begin{aligned} c^0(x^i)u(x^i, t) + \sum_{x^j \in \Omega \cup \partial_N\Omega} K_{ij}u(x^j, t) + \sum_{x^j \in \partial_D\Omega} Q'_{ij}q(x^j, t) = \Psi^0(x^i, t) - \\ - \sum_{x^j \in \partial_D\Omega} K_{ij}\bar{u}(x^j, t), \quad x^i \in \Omega \cup \partial\Omega, \quad i = 1, \dots, J, \text{ no sum in } i, \end{aligned} \quad (8.31)$$

where $\Psi^0(x^i, t)$ is calculated from Eq.(8.27), and

$$\begin{aligned} \tilde{\Psi}(x^i, t) = \int_{\partial_D\Omega} \bar{u}(x, t)T_x G(x, x^i)d\Gamma(x) - \int_{\partial_N\Omega} G(x, x^i)\bar{q}(x, t)d\Gamma(x) + \\ + \int_{\partial\Omega} w(x, t)d\Gamma(x) + \int_{\partial\Omega} \frac{1}{r} \frac{\partial r}{\partial n} F(x, t)d\Gamma(x) \end{aligned} \quad (8.32)$$

$$K_{ij} = \int_{\partial\Omega} h(x) d\Gamma(x) - \int_{\partial_N\Omega} [\Psi^1, \Psi^2] T_x G(x, x^i) d\Gamma(x) \quad (8.33)$$

$$Q'_{ij} = \int_{\partial_D\Omega} G(x, x^i) v_j(x) d\Gamma(x) \quad (8.34)$$

8.4.4 Discretisation of the RIBIDE

To obtain a system of linear algebraic equations from the RIBIDE (8.29), we collocate at the nodes x^i , $i = 1, \dots, J$, and substitute an interpolation of $u(x, t)$ in Eq.(6.25), we then arrive at a system of $J - J_D$ algebraic equations for $J - J_D$ unknowns $u(x^j, t)$, $x^j \in \Omega \cup \partial_N\Omega$. Substituting interpolation formulae (6.25) into the RIBIDE (8.29) leads to the following system of equations:

$$a(x^i)c(x^i)u(x^i, t) + \sum_{x^j \in \Omega \cup \partial_N\Omega} K'_{ij}u(x^j, t) = \tilde{\Psi}(x^i) - \sum_{x^j \in \partial_D\Omega} K'_{ij}\bar{u}(x^j, t),$$

$$x^i \in \Omega \cup \partial_N\Omega, \quad \text{no sum in } i, \quad (8.35)$$

where

$$K'_{ij} = K_{ij} + \int_{\partial_D\Omega} G(x, x^i) T\Phi_j(x) d\Gamma(x) \quad (8.36)$$

and $\tilde{\Psi}(x^i)$ and K_{ij} are given in Eqs.(8.32) and (8.33), respectively.

8.5 Implementation and time marching solution scheme

The assembly of the system matrix A and right-hand side b for the BDIE in chapter 3 and the RIBIE in chapters 6 and 7 is suitable for steady-state heat conduction and the Helmholtz equation. However, this needs to be modified for the diffusion equation. We will start with steady-state heat conduction before moving to the diffusion problems.

8.5.1 Assembling the system for RIBIE for heat conduction with variable coefficients

Let us recall the RIBIE in chapter 6,

$$\begin{aligned}
 a(y)c(y)u(y) - \int_{\partial\Omega} u(x)a(x)\frac{\partial\tilde{P}(x,y)}{\partial n(x)}d\Gamma(x) + \int_{\partial\Omega} \tilde{P}(x,y)q(x)d\Gamma(x) + \\
 + \int_{\partial\Omega} h(x)d\Gamma(x) = \int_{\partial\Omega} \frac{1}{r}\frac{\partial r}{\partial n}F(x)d\Gamma(x)
 \end{aligned}
 \tag{8.37}$$

where $\tilde{P}(x,y)$ and $\tilde{R}(x,y)$, $F(x)$ and $\int_{\partial\Omega} h(x)d\Gamma(x)$ are given in Eqs. (6.3), (6.4), (6.11) and (6.14)-(6.15), respectively. Let us start with a mesh of eight boundary elements and nine nodes (including an interior node), as shown in Fig. 8.1.

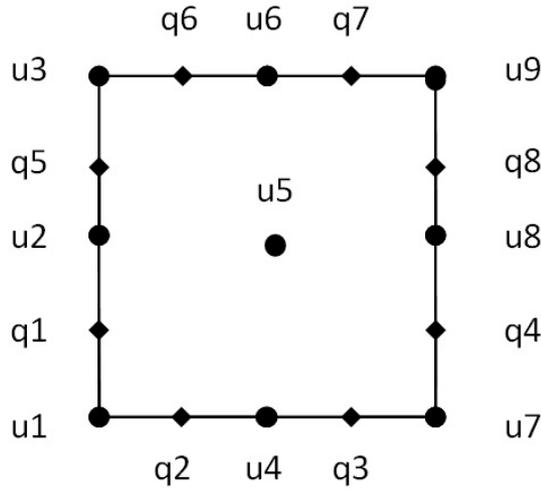


Figure 8.1: Simple mesh with \circ for u and \diamond for flux q

For the RIBIE method, the system of algebraic equations resulting from Eq.(8.37) has two unknown variables q and u , i.e. q in Dirichlet boundaries and u in Neumann boundaries, in addition to u at interior nodes. After doing the numerical integrations for all boundary integrals in Eq.(8.37), we obtain the following system for the simple mesh in Fig. 8.1,

$$(\text{Hmatrix}+\text{Rmatrix})_{9\times 9} * u_{9\times 1} + (\text{Gmatrix})_{9\times 8} * q_{8\times 1} = \text{fvector}_{9\times 1},$$

where $\text{Hmatrix} = a(y)c(y)u(y) - \int_{\partial\Omega} u(x)a(x)\frac{\partial\tilde{P}(x,y)}{\partial n(x)}d\Gamma(x)$, Rmatrix is the last boundary integral on the left-hand side of Eq.(8.37), Gmatrix is the second boundary integral in Eq.(8.37) and fvector is the right-hand heat source vector.

Now we can define the boundary conditions (assuming in Fig. 8.1, the top and bottom of the plate have Dirichlet boundary conditions, left and right have Neumann boundary conditions). Referring to Fig. 8.1, we have six known values of u , namely $u_1, u_4, u_7, u_3, u_6, u_9$ and four known values of q , namely q_1, q_5, q_4, q_8 ; also, we have three unknown values of u , namely u_2, u_5, u_8 and four unknown values of q , namely q_2, q_3, q_6, q_7 . By moving the columns corresponding to the known u and q values to the right-hand side and re-arranging the system, we get

$$A_{9 \times 7} * x_{7 \times 1} = B_{9 \times 10} * y_{10 \times 1} + \text{fvector}_{9 \times 1} = b_{9 \times 1},$$

where the vector y contains the known six values of u and four values of q (coming from the boundary conditions).

In previous chapters, the least squares technique has been applied in the following way, the final system being $Cx = d$:

$[C]_{7 \times 7} = [A^T]_{7 \times 9}[A]_{9 \times 7}$, and $[d]_{7 \times 1} = [A^T]_{7 \times 9}[b]_{9 \times 1}$. The disadvantages of setting up the normal equations ($A^T A$) are discussed in [64–66]:

- 1) Normal equations square the condition number of the original system of equations.
- 2) Setting up the normal equations can lead to loss of accuracy.
- 3) Setting up the normal equations is costly in terms of cpu time and memory storage.
- 4) The possible sparsity of the original set of equations is lost with the normal equations (in our work, matrix A is already dense).

Regarding points 2 and 3, there is a well known example available in [64, 66]:

$$\begin{pmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{pmatrix}, \quad A^T A = \begin{pmatrix} 1 + \epsilon^2 & 1 & 1 \\ 1 & 1 + \epsilon^2 & 1 \\ 1 & 1 & 1 + \epsilon^2 \end{pmatrix}.$$

If only 6 decimal digits are used on a machine and ϵ equals 10^{-3} , $A^T A$ will result in a matrix where all coefficients are equal to one and all information in the last three rows of A is lost, the matrix $A^T A$ is then exactly singular and its inverse does not exist. Furthermore, the sparsity of matrix A is completely lost after the multiplication. In order to avoid all these difficulties, MATLAB avoids calculating the normal equations. There are several ways to compute the unknown quantities x :

1) The backslash operator not only solves square, nonsingular systems, but it also computes the least squares solution to rectangular, overdetermined systems:

$$x = A \backslash b.$$

The computation is done by an orthogonalization algorithm known as the QR factorization. There are two versions of the QR factorization in MATLAB: the first one is the ‘full’ QR decomposition, for a given general matrix $A_{m \times n}$ with $m > n$. In MATLAB, $[Q, R] = qr(A)$ returns the ‘full’ QR decomposition, with square, orthogonal $Q_{m \times m}$, i.e. $Q^T Q = I$, and $R_{m \times n}$ upper triangular. The second is the ‘economy’ QR decomposition, in which $[Q, R] = qr(A, 0)$, and $Q_{m \times n}$, $R_{n \times n}$ then:

$$x = R \backslash Q^T b \text{ which is equivalent to } x = A \backslash b \text{ and } x = lsqr(A, b).$$

The Gram-Schmidt process and Householder reflections described in many linear algebra texts are used to get both Q and R , see [67, 68], but the Gram-Schmidt process is numerically less satisfactory. Therefore, Householder reflections are adopted in the built-in MATLAB functions \backslash , qr and $lsqr$ that are numerically stable.

2) Statistics Toolbox functions like `regress` and `regstats` call the MATLAB backslash operator to perform linear regression. The QR decomposition is also used for an efficient computation, see [69, 70]:

$$x = \text{regress}(b, A).$$

Remark 8.2: In order to assemble the system of algebraic equations for the Laplace equation we follow exactly the same procedure in which the `Rmatrix` and `fvector` are removed, and for the Poisson equation only the `Rmatrix` is removed. This procedure leads to the same results as in chapter 6, see Appendix D, with the advantages that the assembly is much simpler and the code is much faster, but more memory is used in comparison to the previous technique.

8.5.2 Assembling the system for RIBIE for diffusion with variable coefficients

By following the same procedure as for steady-state heat conduction in the previous subsection, we get

$$\begin{aligned} (\text{Hmatrix}+\text{Rmatrix})_{9\times 9} * u_{9\times 1} + (\text{Gmatrix})_{9\times 8} * q_{8\times 1} = \text{fvector}_{9\times 1} + \\ + (\text{Ematrix})_{9\times 9} * \dot{u}_{9\times 1}, \end{aligned} \quad (8.38)$$

where the Ematrix results from the boundary integral $\int_{\partial\Omega} w(x, t) d\Gamma(x)$ given in Eqs. (8.24)-(8.25). To solve the equation Eq.(8.38), we adopt a time marching scheme [2, 71, 72]. A finite difference approximation for the time derivative term is given by:

$$\dot{u} = \frac{u^{k+1} - u^k}{\Delta t}, \quad (8.39)$$

$$u = (1 - \theta_u)u^k + \theta_u u^{k+1}, \quad (8.40)$$

$$q = (1 - \theta_q)q^k + \theta_q q^{k+1}, \quad (8.41)$$

$$b = (1 - \theta_b)b^k + \theta_b b^{k+1}, \quad (8.42)$$

where Δt is the time step, u^k and q^k are the temperature and flux at the k th time step, and θ is a real parameter that determines if the method is explicit ($\theta_u, \theta_q, \theta_b = 0$) or implicit ($\theta_u, \theta_q, \theta_b = 1$). The special choice of ($\theta_u, \theta_q, \theta_b = \frac{1}{2}$) is known as the Crank-Nicolson scheme. Several tests were done here to choose the best values for θ and we selected $\theta_u = \frac{1}{2}$, $\theta_q = 1$ and $\theta_b = 1$.

By applying the values of $\theta_u, \theta_q, \theta_b$ and Eqs. (8.39-8.41) in Eq.(8.38), we get:

$$\begin{aligned} \left[\frac{1}{2}(\text{Hmatrix}+\text{Rmatrix})_{9\times 9} - \frac{1}{\Delta t}\text{Ematrix} \right] * u_{9\times 1}^{k+1} + (\text{Gmatrix})_{9\times 8} * q_{8\times 1}^{k+1} = \\ = \text{fvector}_{9\times 1} + \left[-\frac{1}{2}(\text{Hmatrix}+\text{Rmatrix})_{9\times 9} - \frac{1}{\Delta t}\text{Ematrix} \right]_{9\times 9} * u_{9\times 1}^k. \end{aligned} \quad (8.43)$$

Now we can define the boundary conditions with $\theta_b = 1$, follow the same procedure as in

the previous subsection and re-arrange the system to end up with:

$$A_{9 \times 7} * x_{7 \times 1}^{k+1} = b_{9 \times 1}^{k+1} + \left[-\frac{1}{2}(\text{Hmatrix} + \text{Rmatrix})_{9 \times 9} - \frac{1}{\Delta t} \text{Ematrix} \right]_{9 \times 9} * u_{9 \times 1}^k. \quad (8.44)$$

In the time marching computation, the unknown quantities x are updated at each time step by the new values obtained after solving equation (8.44) (using the same solver as in the previous subsection). At the first time step, the temperature and heat flux at all boundary and internal points are specified with initial values. The computation ends when all time steps are fulfilled [58] or a steady state is reached.

8.5.3 Assembling the system for RIBIDE for diffusion with variable coefficients

By following exactly the same procedure as for steady-state heat conduction in chapters 3 and 6 for the integro-differential equation method, our unknown is only u in Neumann boundaries, in addition to interior nodes, namely u_2, u_5, u_8 . Therefore, the implementation here is much simpler than the RIBIE. Several tests were done here again to choose the best values for θ and we concluded the values to be the same as for RIBIE, i.e. $\theta_u = \frac{1}{2}$ and $\theta_b = 1$, we then get:

$$\begin{aligned} & \left[\frac{1}{2}(\text{Hmatrix} + \text{Rmatrix} + \text{Gmatrix})_{3 \times 3} - \frac{1}{\Delta t} \text{Ematrix} \right] * u_{3 \times 1}^{k+1} = b_{3 \times 1}^{k+1} + \\ & + \left[-\frac{1}{2}(\text{Hmatrix} + \text{Rmatrix} + \text{Gmatrix})_{3 \times 3} - \frac{1}{\Delta t} \text{Ematrix} \right]_{3 \times 3} * u_{3 \times 1}^k. \end{aligned} \quad (8.45)$$

Then, the unknown quantities u can be obtained by:

$$u_{3 \times 1}^{k+1} = M \setminus S,$$

where $M = \left[\frac{1}{2}(\text{Hmatrix} + \text{Rmatrix} + \text{Gmatrix})_{3 \times 3} - \frac{1}{\Delta t} \text{Ematrix} \right]$,

$S = b_{3 \times 1}^{k+1} + \left[-\frac{1}{2}(\text{Hmatrix} + \text{Rmatrix} + \text{Gmatrix})_{3 \times 3} - \frac{1}{\Delta t} \text{Ematrix} \right]_{3 \times 3} * u_{3 \times 1}^k$.

In the time marching computation, the unknown quantities u are updated at each time step by the new values obtained after solving equation (8.45) (using the same solver as in the previous subsection, since the system is square the backslash operator uses the Gauss elimination method to solve for the unknown u). At the first time step, the temperature at all Neumann boundary and internal points are specified with initial values.

8.6 Numerical results

In this section, we shall examine some test examples to assess the performance of the RIBIDE/RIBIE formulations. We apply the RIBIDE/RIBIE for homogenous, non-homogeneous and variable coefficient diffusion equations on a square domain, for which an exact analytical solution, u_{exact} , is available. Computer programs were developed by using Matlab. Moreover, $\phi(R) = R^3$ is adopted in the test examples. The total number of nodes is 81 (32 on the boundary plus 49 in the interior). Also, the top and bottom sides of the plates for all tests examples have prescribed temperature u (Dirichlet boundary conditions), while the left and right sides are imposed with heat flux t (Neumann boundary conditions). Also, the relative error and Root Mean Square (RMS) error also calculated as given by Eqs.(2.35) and (6.28) to check the convergence of the proposed methods. These errors have been calculated for $J= 25, 81, 289$ and 1089 in all test examples and a time interval $0 \leq t \leq 1$ will be used in all test examples for the diffusion equation.

8.6.1 Numerical results for homogeneous diffusion equation with constant $a(x)$

Test 1

Consider a square domain, $\Omega = [1, 2]^2$. The value of $D(x, t)$ is set as constant, $D(x, t) = \frac{1}{2}$ and $f(x, t) = 0$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = x_1^2 + x_2^2.$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$\begin{aligned} u(x_1, 1, t) &= x_1^2 + 1 + 8t, & u(x_1, 2, t) &= x_1^2 + 4 + 8t, \\ \frac{\partial u(1, x_2, t)}{\partial n} &= -2, & \frac{\partial u(2, x_2, t)}{\partial n} &= 4. \end{aligned}$$

The exact solution of this problem is $u(x_1, x_2, t) = x_1^2 + x_2^2 + 8t$. Figs. 8.2 and 8.3 plot the relative and RMS errors for RIBIDE and RIBIE, respectively. It can be clearly seen that the RIBIE produces better results than the RIBIDE.

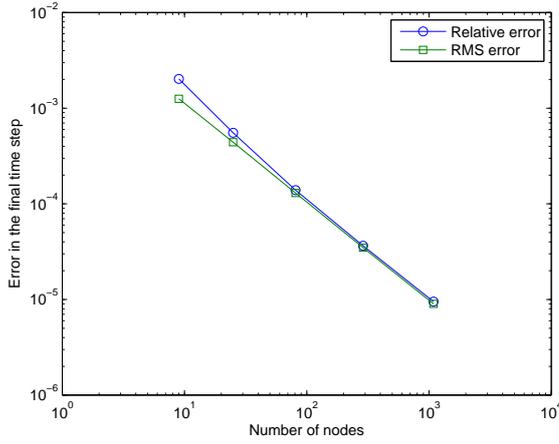


Figure 8.2: Relative and RMS errors for RIBIE method for test 1 with $\Delta t = 0.1$

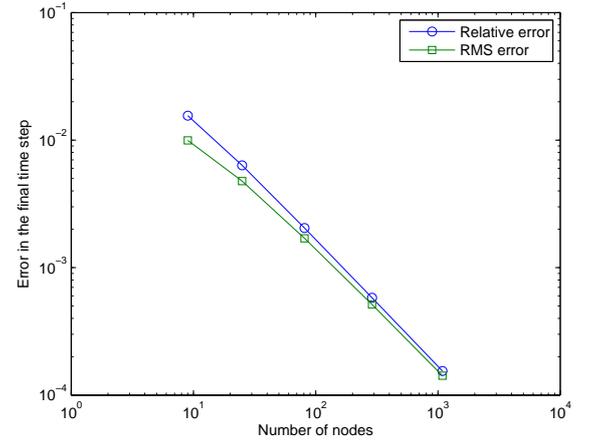


Figure 8.3: Relative and RMS errors for RIBIDE method for test 1 with $\Delta t = 0.1$

Test 2

Consider a square domain, $\Omega = [0, 1]^2$. The value of $D(x, t)$ is set as constant, $D(x, t) = 1$ and $f(x, t) = 0$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = 1 + \cos\left(\frac{\pi}{4}x_1\right) \sin\left(\frac{\pi}{4}x_2\right).$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$\begin{aligned} u(x_1, 0, t) &= 1, & u(x_1, 1, t) &= 1 + e^{-\frac{\pi^2}{8}t} \cos\left(\frac{\pi}{4}x_1\right) \sin\left(\frac{\pi}{4}\right), \\ \frac{\partial u(0, x_2, t)}{\partial n} &= 0, & \frac{\partial u(1, x_2, t)}{\partial n} &= -\frac{\pi}{4} e^{-\frac{\pi^2}{8}t} \sin\left(\frac{\pi}{4}\right) \sin\left(\frac{\pi}{4}x_2\right). \end{aligned}$$

The exact solution of this problem is $u(x_1, x_2, t) = 1 + e^{-\frac{\pi^2}{8}t} \cos\left(\frac{\pi}{4}x_1\right) \sin\left(\frac{\pi}{4}x_2\right)$.

Figs. 8.4-8.6 show the variation of $u(x)$ along the line $x_2 = 0.5$ for the three internal points ($x_1 = 0.25, 0.5$ and 0.75) using RIBIDE and RIBIE with $\Delta t = 0.01$. The total number of nodes is fixed to 81 (32 on the boundary plus 49 in the interior).

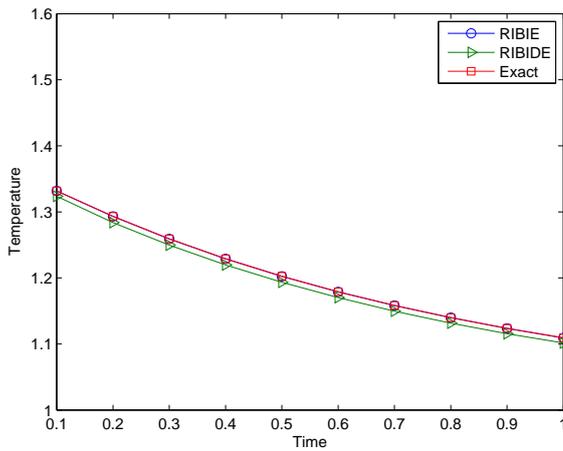


Figure 8.4: Temperature distribution along the line $x_2 = 0.5$ with $x_1 = 0.25$

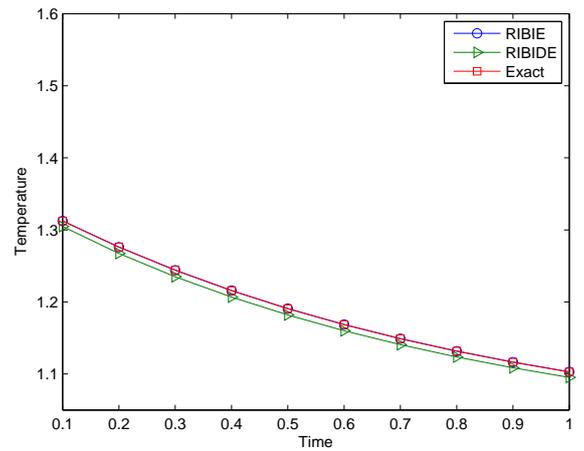


Figure 8.5: Temperature distribution along the line $x_2 = 0.5$ with $x_1 = 0.5$

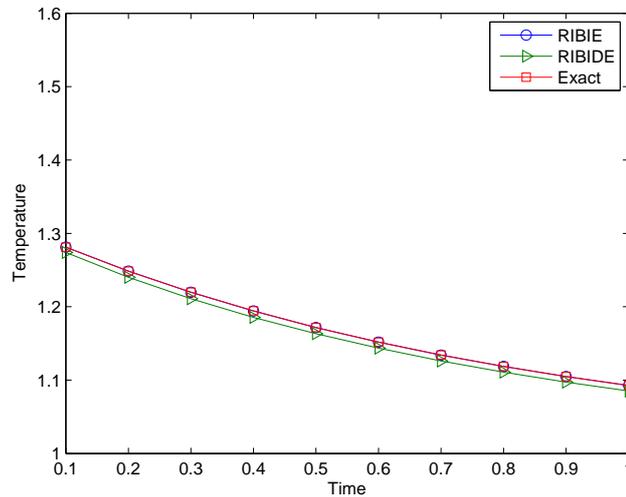


Figure 8.6: Temperature distribution along the line $x_2 = 0.5$ with $x_1 = 0.75$

Test 3

Consider a square domain, $\Omega = [0, 1]^2$. The value of $D(x, t)$ is set as constant, $D(x, t) = 1$ and $f(x, t) = 0$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = (1 - x_2)e^{x_1}.$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$u(x_1, 0, t) = e^{(x_1+t)}, \quad u(x_1, 1, t) = 0,$$

$$\frac{\partial u(0, x_2, t)}{\partial n} = -(1 - x_2)e^t, \quad \frac{\partial u(1, x_2, t)}{\partial n} = (1 - x_2)e^{(t+1)}.$$

The exact solution of this problem is $u(x_1, x_2, t) = (1 - x_2)e^{(x_1+t)}$. Tables 8.1-8.6 list the computed values of $u(x)$ along the middle line of the plate $x_2 = 0.5$ with three internal points ($x_1 = 0.25, 0.5$ and 0.75) using RIBIDE and RIBIE with different times steps. The total number of nodes is fixed to 81 (32 on the boundary plus 49 in the interior). In order to study convergence, the time steps adopted are $\Delta t = 0.1$, $\Delta t = 0.01$ and $\Delta t = 0.001$.

Table 8.1: Computed temperatures using RIBIE along line $x_2 = 0.5$ with $x_1 = 0.25$

time	RIBIE ($\Delta t = 0.1$)	RIBIE ($\Delta t = 0.01$)	RIBIE ($\Delta t = 0.001$)	Exact
0.1	0.7076	0.7097	0.7099	0.7095
0.2	0.7818	0.7845	0.7847	0.7842
0.3	0.8657	0.8672	0.8674	0.8666
0.4	0.9562	0.9584	0.9586	0.9578
0.5	1.0572	1.0592	1.0594	1.0585
0.6	1.1682	1.1706	1.1709	1.1698
0.7	1.2911	1.2937	1.2940	1.2929
0.8	1.4269	1.4298	1.4301	1.4288
0.9	1.5770	1.5802	1.5805	1.5791
1	1.7429	1.7464	1.7467	1.7452

Table 8.2: Computed temperatures using RIBIDE along line $x_2 = 0.5$ with $x_1 = 0.25$

time	RIBIDE ($\Delta t = 0.1$)	RIBIDE ($\Delta t = 0.01$)	RIBIDE ($\Delta t = 0.001$)	Exact
0.1	0.7268	0.7156	0.7140	0.7095
0.2	0.8276	0.7949	0.7920	0.7842
0.3	0.9172	0.8801	0.8764	0.8666
0.4	1.0163	0.9733	0.9690	0.9578
0.5	1.1245	1.0759	1.0711	1.0585
0.6	1.2417	1.1891	1.1838	1.1698
0.7	1.3738	1.3142	1.3083	1.2929
0.8	1.5169	1.4524	1.4459	1.4288
0.9	1.6778	1.6052	1.5980	1.5791
1	1.8530	1.7740	1.7661	1.7452

Table 8.3: Computed temperatures using RIBIE along line $x_2 = 0.5$ with $x_1 = 0.5$

time	RIBIE ($\Delta t = 0.1$)	RIBIE ($\Delta t = 0.01$)	RIBIE ($\Delta t = 0.001$)	Exact
0.1	0.9128	0.9117	0.9115	0.9111
0.2	1.0113	1.0079	1.0076	1.0069
0.3	1.1180	1.1140	1.1136	1.1128
0.4	1.2356	1.2312	1.2308	1.2298
0.5	1.3658	1.3607	1.3603	1.3591
0.6	1.5093	1.5039	1.5033	1.5021
0.7	1.6682	1.6620	1.6614	1.6601
0.8	1.8435	1.8368	1.8362	1.8346
0.9	2.0375	2.0300	2.0293	2.0276
1	2.2517	2.2435	1.7661	2.2408

Table 8.4: Computed temperatures using RIBIDE along line $x_2 = 0.5$ with $x_1 = 0.5$

time	RIBIDE ($\Delta t = 0.1$)	RIBIDE ($\Delta t = 0.01$)	RIBIDE ($\Delta t = 0.001$)	Exact
0.1	0.9362	0.9191	0.9168	0.9111
0.2	1.0650	1.0202	1.0163	1.0069
0.3	1.1765	1.1292	1.1244	1.1128
0.4	1.3043	1.2486	1.2431	1.2298
0.5	1.4423	1.3801	1.3740	1.3591
0.6	1.5929	1.5254	1.5185	1.5021
0.7	1.7623	1.6858	1.6783	1.6601
0.8	1.9457	1.8631	1.8548	1.8346
0.9	2.1522	2.0591	2.0498	2.0276
1	2.3769	2.2756	2.2654	2.2408

Table 8.5: Computed temperatures using RIBIE along line $x_2 = 0.5$ with $x_1 = 0.75$

time	RIBIE ($\Delta t = 0.1$)	RIBIE ($\Delta t = 0.01$)	RIBIE ($\Delta t = 0.001$)	Exact
0.1	1.1781	1.1712	1.1704	1.1698
0.2	1.3072	1.2948	1.2938	1.2929
0.3	1.4426	1.4311	1.4299	1.4288
0.4	1.5962	1.5817	1.5803	1.5791
0.5	1.7631	1.7481	1.7465	1.7452
0.6	1.9492	1.9319	1.9302	1.9287
0.7	2.1538	2.1351	2.1332	2.1316
0.8	2.3805	2.3597	2.3576	2.3557
0.9	2.6308	2.6078	2.6055	2.6035
1	2.9074	2.8821	2.8796	2.8773

Table 8.6: Computed temperatures using RIBIDE along line $x_2 = 0.5$ with $x_1 = 0.75$

time	RIBIDE ($\Delta t = 0.1$)	RIBIDE ($\Delta t = 0.01$)	RIBIDE ($\Delta t = 0.001$)	Exact
0.1	1.2057	1.1798	1.1765	1.1698
0.2	1.3690	1.3088	1.3036	1.2929
0.3	1.5082	1.4482	1.4420	1.4288
0.4	1.6736	1.6011	1.5940	1.5791
0.5	1.8489	1.7697	1.7691	1.7452
0.6	2.0430	1.9560	1.9472	1.9287
0.7	2.2595	2.1617	2.1520	2.1316
0.8	2.4951	2.3891	2.3784	2.3557
0.9	2.7597	2.6403	2.6285	2.6035
1	3.0478	2.9180	2.9049	2.8773

8.6.2 Numerical results for non-homogeneous diffusion equation with constant $a(x)$

Test 4

Consider a square domain, $\Omega = [1, 2]^2$. The value of $D(x, t)$ is set as constant, $D(x, t) = 1$ and $f(x, t) = -2$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = x_1 + x_2.$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$u(x_1, 1, t) = x_1 + 1 + 2t, \quad u(x_1, 2, t) = x_1 + 2 + 2t,$$

$$\frac{\partial u(1, x_2, t)}{\partial n} = -1, \quad \frac{\partial u(2, x_2, t)}{\partial n} = 1.$$

The exact solution of this problem is $u(x_1, x_2, t) = x_1 + x_2 + 2t$. Figs. 8.7 and 8.8 plot the relative and RMS errors for RIBIDE and RIBIE, respectively. It can be clearly seen that the RIBIE produces better results than the RIBIDE.

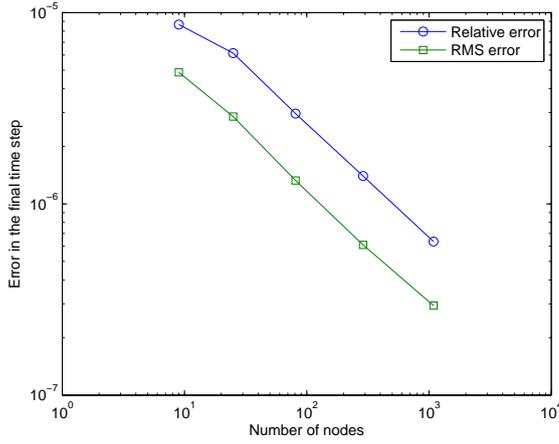


Figure 8.7: Relative and RMS errors for RIBIE method for test 4 with $\Delta t = 0.1$

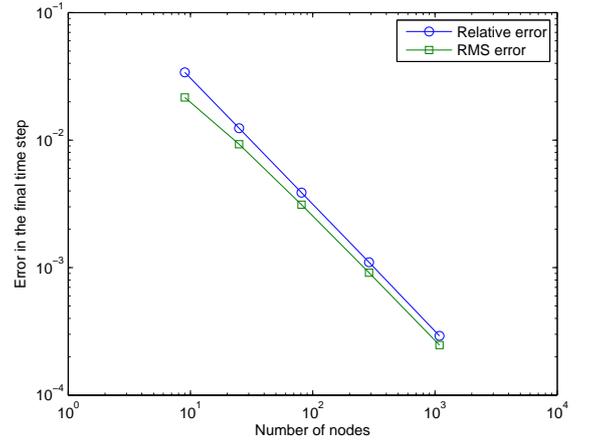


Figure 8.8: Relative and RMS errors for RIBIDE method for test 4 with $\Delta t = 0.1$

Test 5

Consider a square domain, $\Omega = [1, 2]^2$. The value of $D(x, t)$ is set as constant, $D(x, t) = \frac{1}{6}$ and $f(x, t) = 6(x_1 + x_2) - 2t$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = x_1^3 + x_2^3.$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$u(x_1, 1, t) = x_1^3 + 1 + 6t^2, \quad u(x_1, 2, t) = x_1^3 + 8 + 6t^2,$$

$$\frac{\partial u(1, x_2, t)}{\partial n} = -3, \quad \frac{\partial u(2, x_2, t)}{\partial n} = 12.$$

The exact solution of this problem is $u(x_1, x_2, t) = x_1^3 + x_2^3 + 6t^2$. Figs. 8.9-8.12 plot the relative and RMS errors for RIBIDE and RIBIE, respectively, with increased number of nodes and time steps of $\Delta t = 0.1$ and $\Delta t = 0.01$.

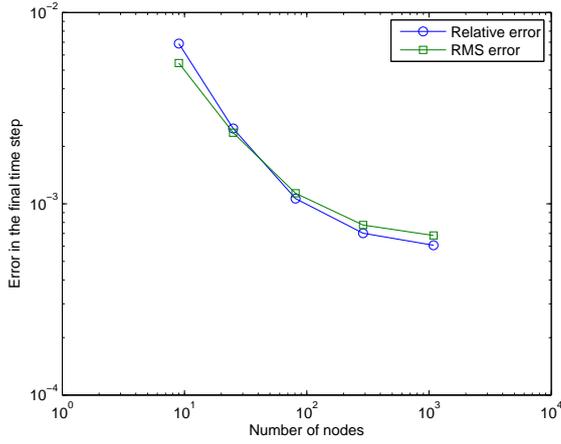


Figure 8.9: Relative and RMS errors for RIBIE method for test 5 with $\Delta t = 0.1$

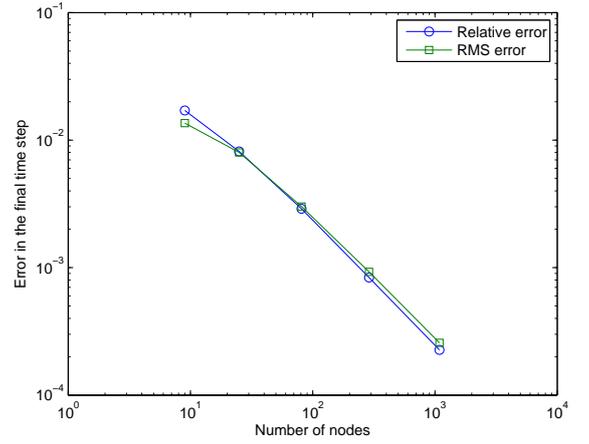


Figure 8.10: Relative and RMS errors for RIBIDE method for test 5 with $\Delta t = 0.1$

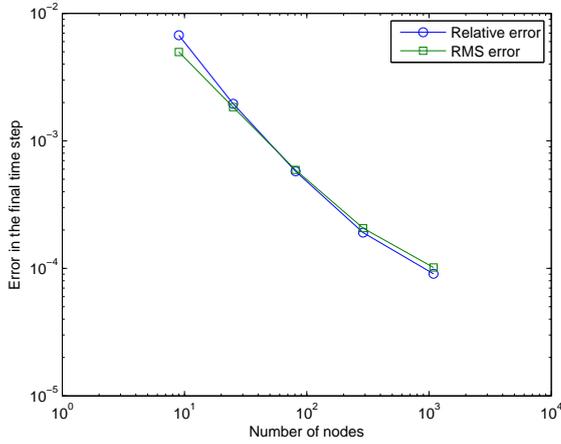


Figure 8.11: Relative and RMS errors for RIBIE method for test 5 with $\Delta t = 0.01$

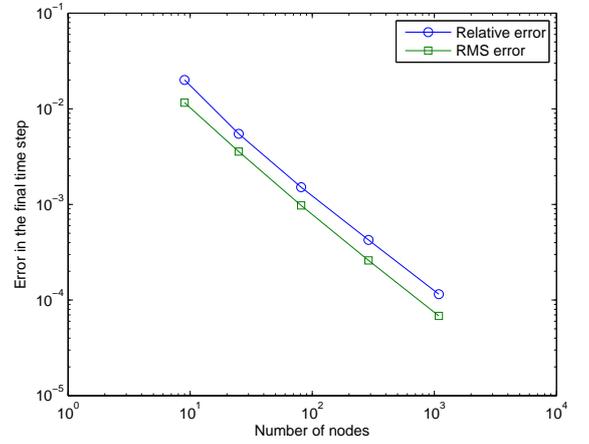


Figure 8.12: Relative and RMS errors for RIBIDE method for test 5 with $\Delta t = 0.01$

8.6.3 Numerical results for non-homogeneous diffusion equation with variable $a(x)$

Test 6

Consider a square domain, $\Omega = [1, 2]^2$. The value of $D(x, t)$ is set as constant, $D(x, t) = 1$, $f(x, t) = 2(x_1 + x_2) + \frac{1}{3} \sin(t)$ and $a(x) = x_1^2 + x_2^2$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = x_1 + x_2 + \frac{1}{3}.$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$u(x_1, 1, t) = x_1 + 1 + \frac{1}{3} \cos(t), \quad u(x_1, 2, t) = x_1 + 2 + \frac{1}{3} \cos(t),$$

$$\frac{\partial u(1, x_2, t)}{\partial n} = -(1 + x_2^2), \quad \frac{\partial u(2, x_2, t)}{\partial n} = 4 + x_2^2.$$

The exact solution of this problem is $u(x_1, x_2, t) = x_1 + x_2 + \frac{1}{3} \cos(t)$. Figs. 8.13 and 8.14 plot the relative and RMS errors for RIBIDE and RIBIE, respectively.

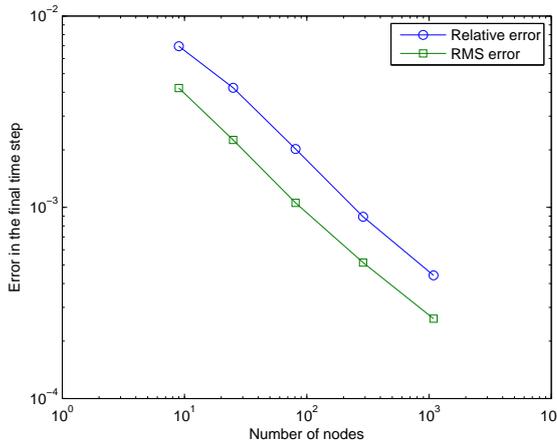


Figure 8.13: Relative and RMS errors for RIBIE method for test 6 with $\Delta t = 0.1$

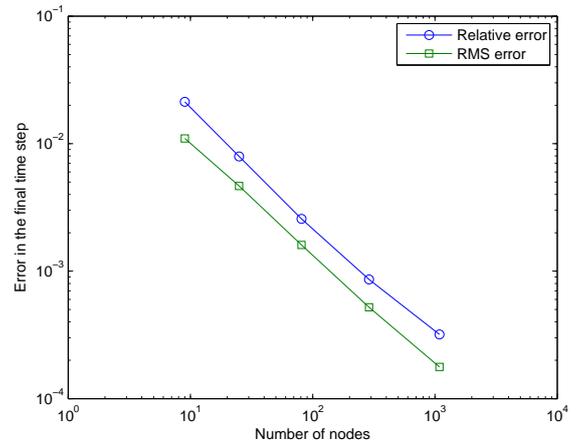


Figure 8.14: Relative and RMS errors for RIBIDE method for test 6 with $\Delta t = 0.1$

Test 7

Consider a square domain, $\Omega = [1, 2]^2$. The value of $D(x, t)$ is set as constant, $D(x, t) = 1$, $f(x, t) = 6(x_1 + x_2) - 4$ and $a(x) = x_1 + x_2$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = x_1^2 + x_2^2.$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$u(x_1, 1, t) = x_1^2 + 1 + 4t, \quad u(x_1, 2, t) = x_1^2 + 4 + 4t,$$

$$\frac{\partial u(1, x_2, t)}{\partial n} = -2(1 + x_2), \quad \frac{\partial u(2, x_2, t)}{\partial n} = 4(2 + x_2).$$

The exact solution of this problem is $u(x_1, x_2, t) = x_1^2 + x_2^2 + 4t$. Figs. 8.15 and 8.16 plot the relative and RMS errors for each time step using RIBIE and RIBIDE, respectively, while Figs. 8.17 and 8.18 plot the relative and RMS errors for the final time step using

RIBIE and RIBIDE, respectively. It can be clearly seen that by increasing the number of nodes the accuracy increases and the relative and RMS errors reduce.

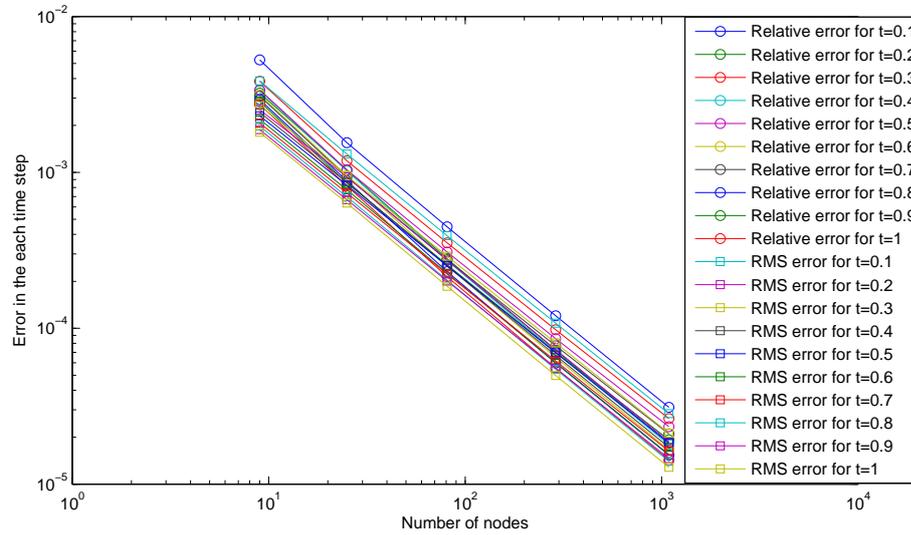


Figure 8.15: Relative and RMS errors for each time step using RIBIE for test 7

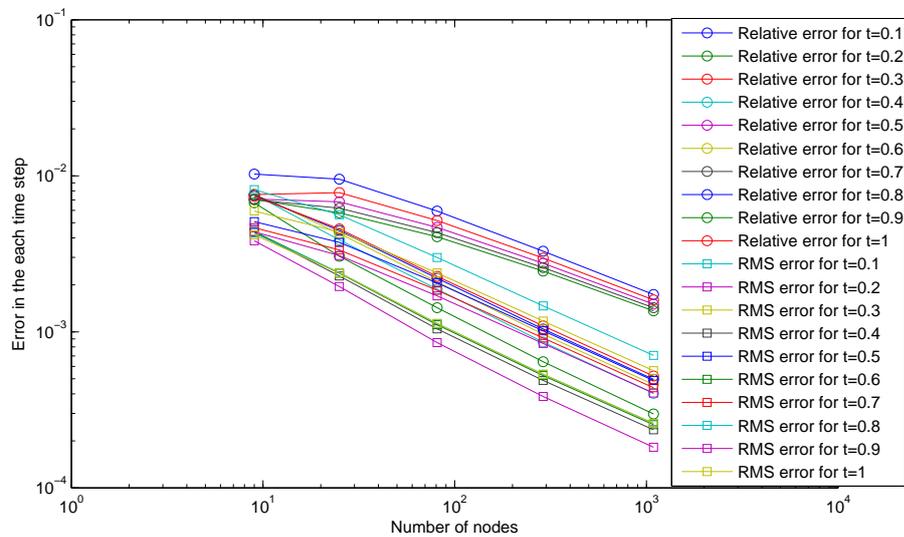


Figure 8.16: Relative and RMS errors for each time step using RIBIDE for test 7

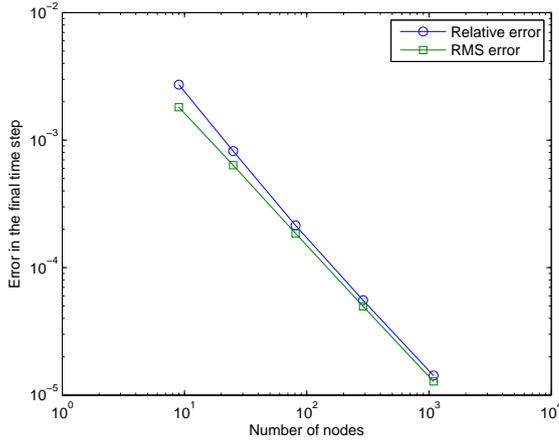


Figure 8.17: Relative and RMS errors for final time step using RIBIE for test 7

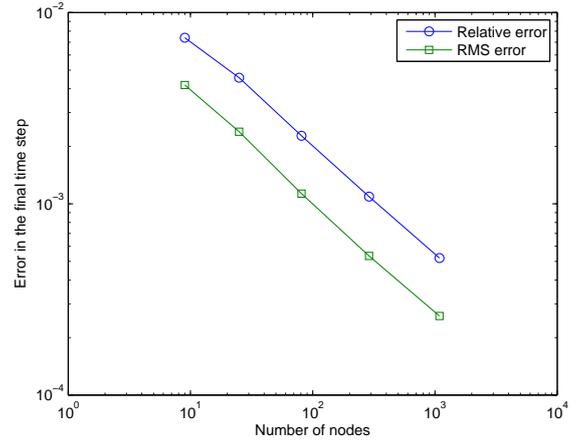


Figure 8.18: Relative and RMS errors for final time step using RIBIDE for test 7

Test 8

Consider a square domain, $\Omega = [1, 2]^2$. The value of $D(x, t)$ is set as variable, $D(x, t) = x_1 + x_2$, $f(x, t) = 5(x_1 + x_2)$ and $a(x) = x_1 + x_2$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = x_1^2 + x_2^2.$$

The following time dependent mixed boundary conditions are applied on the square domain:

$$\begin{aligned} u(x_1, 1, t) &= x_1^2 + 1 + t, & u(x_1, 2, t) &= x_1^2 + 4 + t, \\ \frac{\partial u(1, x_2, t)}{\partial n} &= -2(1 + x_2), & \frac{\partial u(2, x_2, t)}{\partial n} &= 4(2 + x_2). \end{aligned}$$

The exact solution of this problem is $u(x_1, x_2, t) = x_1^2 + x_2^2 + t$. In this test, both the material parameter $a(x)$ and $D(x, t)$ are variable. Figs. 8.19 and 8.20 plot the relative and RMS errors for each time step using RIBIE and RIBIDE, respectively, while Figs. 8.21 and 8.22 plot the relative and RMS errors for the final time step using RIBIE and RIBIDE, respectively. It can be clearly seen by increasing the number of nodes the accuracy increases and the relative and RMS errors reduce.

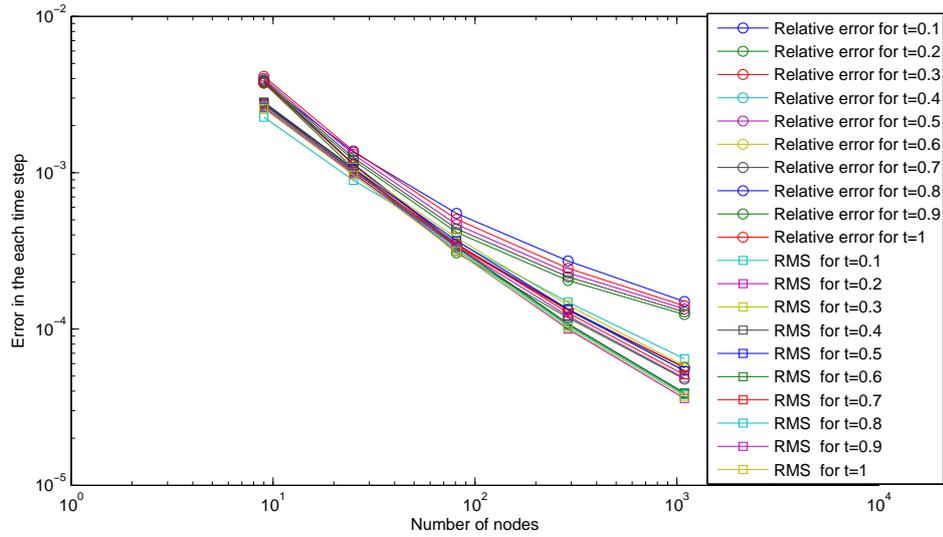


Figure 8.19: Relative and RMS errors for each time step using RIBIE for test 8

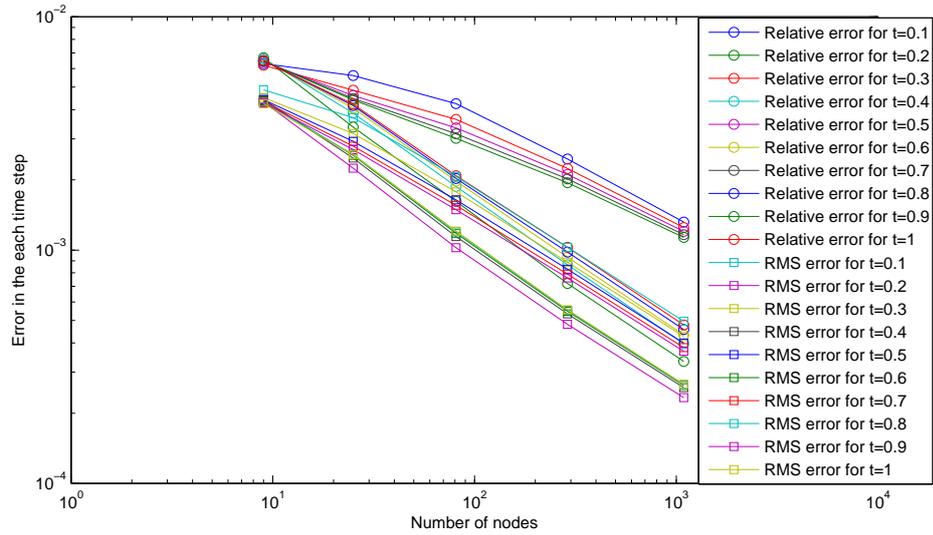


Figure 8.20: Relative and RMS errors for each time step using RIBIDE for test 8

Test 9

Consider a square domain, $\Omega = [1, 2]^2$. The value of $D(x, t)$ is set as variable, $D(x, t) = 1 + t$, $f(x, t) = 9(x_1^2 + x_2^2) + 12(x_1x_2) - 1 - t$ and $a(x) = x_1 + x_2$. Initially, at $t = 0$, the scalar field function $u(x_1, x_2, t)$ has a value of

$$u(x_1, x_2, 0) = x_1^3 + x_2^3.$$

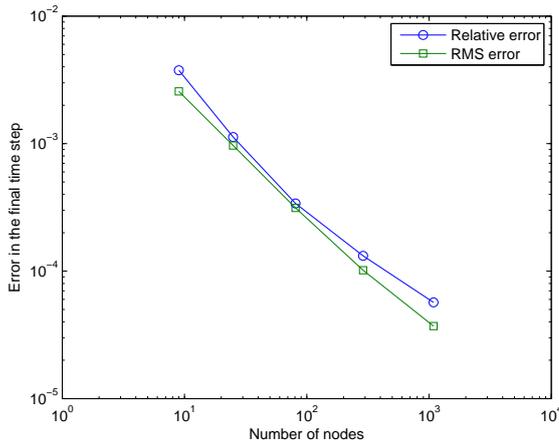


Figure 8.21: Relative and RMS errors for final time step using RIBIE for test 8

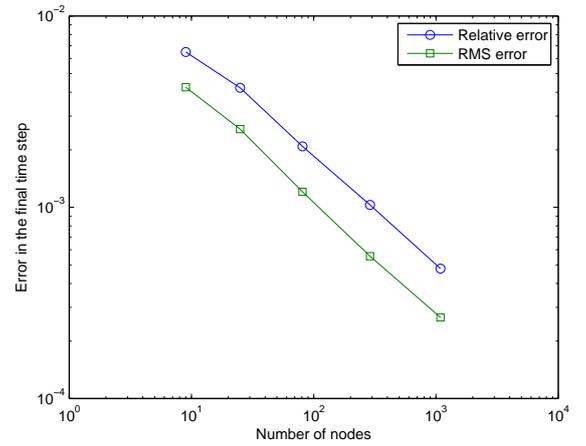


Figure 8.22: Relative and RMS errors for final time step using RIBIDE for test 8

The following time dependent mixed boundary conditions are applied on the square domain:

$$u(x_1, 1, t) = x_1^3 + 1 + t, \quad u(x_1, 2, t) = x_1^3 + 8 + t,$$

$$\frac{\partial u(1, x_2, t)}{\partial n} = -3(1 + x_2), \quad \frac{\partial u(2, x_2, t)}{\partial n} = 12(2 + x_2).$$

The exact solution of this problem is $u(x_1, x_2, t) = x_1^3 + x_2^3 + t$. Figs. 8.23 and 8.24 plot the relative and RMS errors for each time step using RIBIE and RIBIDE, respectively, while Figs. 8.25 and 8.26 plot the relative and RMS errors for the final time step using RIBIE and RIBIDE, respectively. It can be clearly seen that good accuracy and convergence are obtained with mesh refinement.

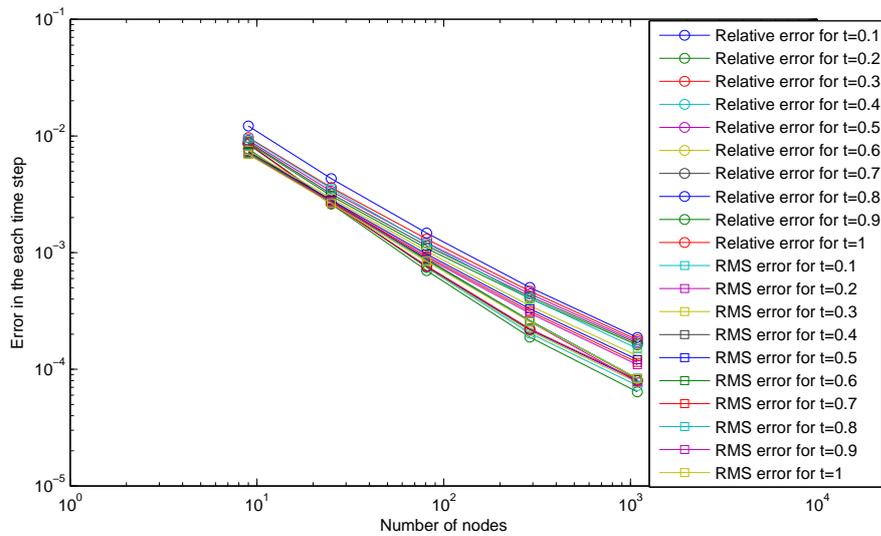


Figure 8.23: Relative and RMS errors for each time step using RIBIE for test 9

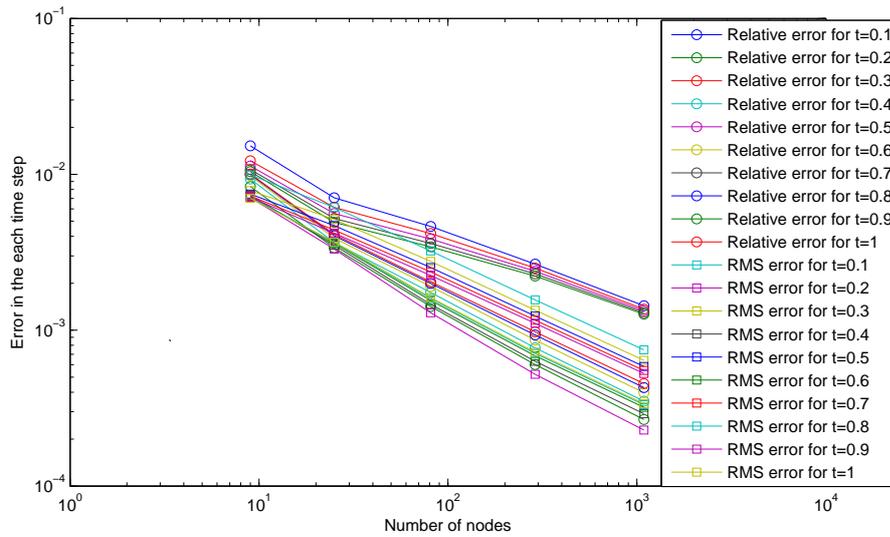


Figure 8.24: Relative and RMS errors for each time step using RIBIDE for test 9

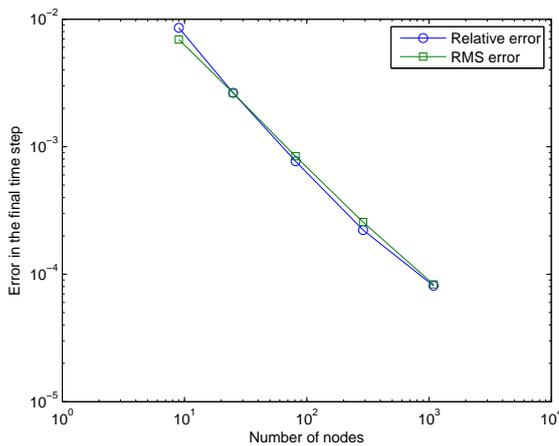


Figure 8.25: Relative and RMS errors for final time step using RIBIE for test 9

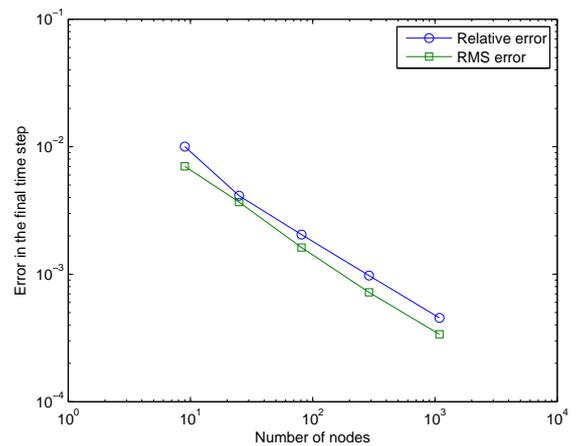


Figure 8.26: Relative and RMS errors for final time step using RIBIDE for test 9

8.7 Concluding remarks

In this chapter, the BDIE and BDIDE formulations are derived for the two-dimensional diffusion equation with variable coefficients. The radial integration method is used to transform the domain integrals appearing in both the BDIE and BDIDE formulations. The resulting RIBIE/RIBIDE formulations are implemented for the numerical solution of three possible cases, homogeneous, non-homogeneous and variable coefficient diffusion equations. From the numerical results presented in this chapter, we can conclude the following general remarks:

- By fixing the time step and increasing the number of nodes, we achieved satisfactory results and convergence was demonstrated (in tests 1 and 4). In general, the RIBIE produces better results than the RIBIDE;
- In test 2 both the number of nodes and the time step are fixed. The computed values of $u(x)$ along the middle line of the plate $x_2 = 0.5$ at three internal points ($x_1 = 0.25, 0.5$ and 0.75) using RIBIDE and RIBIE are presented. It is observed that the RIBIE produces better results than the RIBIDE;
- In test 3 we fixed the number of nodes and reduced the initial time step of $\Delta t = 0.1$ to $\Delta t = 0.01$ and $\Delta t = 0.001$. The values of $u(x)$ were computed along the middle line of the plate $x_2 = 0.5$ at three internal points ($x_1 = 0.25, 0.5$ and 0.75). It can be seen that both the RIBIE and RIBIDE methods are able to generate accurate solutions in good agreement with the exact solution when the time steps are reduced;
- In test 5 both the time step is reduced and the number of nodes is increased. It can be clearly seen that the results are convergent;
- Tests 6 and 7 deal with problems with variable coefficients, with increasing degree of complexity of the variation of the material parameter coefficients. Therefore, a new domain integral appears due to the remainder. Both RIBIE and RIBIDE provide satisfactory results. Moreover, the relative and RMS errors for each time step using both the RIBIE and RIBIDE are presented for test 7. It can be clearly seen the convergence at each time step.
- The results presented in this chapter for the diffusion equation appear to be less accurate than for steady-state heat conduction and Helmholtz equations in previous chapters. One possible reason is the new domain integral appearing due to the time derivative term. This domain integral is converted to the boundary using the RIM, but still requires more numerical integrations. Another reason is that the time marching scheme for solving the time-dependent system of equations is a first-order finite difference technique, probably using high order algorithms can improve the accuracy of the results.

Chapter 9

Conclusions and Future Work

The main aim of this thesis was to derive the radial integration boundary integral and integro-differential equations formulations and then implement the boundary element method to provide the numerical solution for PDEs with variable coefficients. This aim was achieved by studying steady state heat conduction, Helmholtz equation and transient heat conduction problems (diffusion equation) in an isotropic inhomogeneous medium. In this chapter we shall review the main results presented in the thesis and make suggestions for future work.

9.1 Conclusions

The behaviour of many modern industrial materials, for instance functionally graded materials, can be mathematically modeled by PDEs with variable coefficients. The solution of PDEs with variable coefficients is therefore important in many practical engineering problems.

The application of the BEM to PDEs with variable coefficients is hampered by the need to find appropriate fundamental solutions. It is difficult or impossible to derive an analytic expression for the fundamental solution for general PDEs with variable coefficients, except for some special cases. The ability of finding the fundamental solution for this type of problem has been restricted to only very specific cases of variable coefficients. Even for such simple cases the mathematical procedures are very complicated. As a result, if the fundamental solution cannot be found for PDEs with general types of variable coefficients, domain integrals will remain in the BEM formulation. If a domain discretisation

is required, the dimensionality reduction advantage of the BEM is lost.

Several techniques have been proposed which allow to convert domain integrals to equivalent boundary integrals. The radial integration method (RIM) appears to be the most promising method for avoiding domain integration for general PDEs with variable coefficients. However, it has been observed that the RIM available in the literature is restricted to star-shaped geometries due to the way the radial integral is calculated through the domain. Modifications have been introduced to the RIM in its application to the BDIE and BDIDE formulations, particularly the fact that the radial integral is calculated by using a transformation proposed by Fata [53] that was introduced to treat the domain integral using an extension of the fundamental theorem of calculus to higher dimension, and the divergence theorem.

The conclusions obtained from this thesis are as follows:

- 1) Both BDIE and BDIDE methods presented in chapters 3 and 4 for two-dimensional second-order linear elliptic equations, for heat conduction and wave propagation with variable coefficients, provided satisfactory results. To calculate the boundary integrals we used a standard Gaussian quadrature rule. For the domain integrals, we tested many existing formulations (see appendix B) and implemented a Gaussian quadrature rule with Duffy transformation, which provides the best results in comparison to the others methods. However, both BDIE and BDIDE formulations, also when applied to the diffusion equation, generate domain integrals in the corresponding integral equation. This feature makes the BEM less attractive as a domain discretisation is then required.
- 2) The RIM is used to convert the domain integrals appearing in both BDIE and BDIDE formulations to equivalent boundary integrals. For domain integrals consisting of known functions the transformation is straightforward, while for domain integrals that include unknown variables the transformation is accomplished with the use of augmented RBFs, similar to the DRM. Modifications have been introduced to the RIM developed by Gao [45] in its application to the BDIE and BDIDE formulations, particularly the fact that the radial integral is calculated by using a transformation proposed by Fata [53] which produces a pure boundary-only formulation and relaxes the “star-shaped” requirement of the RIM as the straight path from the source point to any field point will always exist.
- 3) As both u and t along the boundary are calculated in the BDIE and RIBIE methods, we implemented mixed boundary elements with linear u and constant t to avoid the

discontinuities of t at corner points. In this case, collocation was tested at the mid and end points of each boundary element. It was shown that end-node collocation generally provides higher accuracy than mid-node collocation. However, using end-node collocation leads to over-determined systems of equations which need be solved by using a least square technique. Different Matlab solvers have been discussed in chapter eight, with the advantages and disadvantages of each one.

4) The only boundary variable in the BDIDE and RIBIDE method is u along Neumann boundaries, thus there is no need for collocation along Dirichlet boundaries. Thus, the problem caused by the discontinuity of the normal derivative at corner points is avoided. Second, the system of linear equations is smaller than the one for the RIBIE. This feature will save memory and computational time when we apply the RIBIDE to practical problems. Finally, the assembly of matrix A and vector b is much easier than in the RIBIE. In chapter eight, we followed the standard BEM to assemble the system of equations for the RIBIE (which is also valid for BDIE). We noticed that this implementation can make the RIBIE faster even than the RIBIDE, although it needed more memory storage in comparison to the RIBIDE.

5) The BDIE, BDIDE, RIBIE and RIBIDE formulations have been extended to study the diffusion equation with variable coefficients. The implementations of RIBIE and RIBIDE have been applied for the three cases, homogeneous, non-homogeneous and non-homogeneous with variable coefficients. It has been observed that both methods are able to generate satisfactory results. Also, an implicit time marching solution scheme was developed for solving the time-dependent system of equations. Several tests were carried out to select the best values for the parameter θ for RIBIE and RIBIDE, and we concluded that the best accuracy was achieved with the Crank-Nicolson scheme used with $\theta_u = \frac{1}{2}$, and implicit for $\theta_q = 1$ and $\theta_b = 1$.

9.2 Future work

We will now suggest some ideas on how this thesis might be modified and extended, including some ideas for future work, which can be summarized in the following points:

1) It has been noticed in the application of the RIM in chapters 5-8 for domain integrals consisting of known functions that the transformation is straightforward and more accurate

than the DRM as there is an exact transformation to the boundary. However, for domain integrals that include unknown variables, the transformation is accomplished with the use of augmented RBFs, and it appears that the numerical integrations in the MATLAB code are time consuming. One possible recent solution is to use a fourth-order spline RBF to approximate the unknown function variation and then integrate the resulting integral analytically [52]. Although the use of analytical expressions can considerably improve the computational efficiency of the techniques, this work is restricted only to a fourth-order spline RBF. Our plan to improve the efficiency of the code is by writing parts of the code (the numerical integration of radial integrals) in C/C++ and link it into MATLAB using a MEX file (a MEX file (also written as MEX-file) provides an interface between MATLAB and subroutines written in C, C++ or Fortran).

Another possible plan is to mix DRM and RIM, by applying RIM for known integrand functions (the advantage being that the RIM does not resort to particular solutions as in the DRM and exact transformations are available) and use the DRM to convert the domain integrals that include unknown functions to the boundary without the need for further numerical integration. This is based on using the available matrices from the boundary integrals, as discussed in great detail in the book of Partridge et al. [2]. Using this procedure, it may be possible to simplify and speed up the calculation of the integrals.

- 2) It would be useful and interesting to develop a theoretical estimate of the convergence rate of the RIM for all problems in the present thesis.
- 3) Theoretically, it is straightforward to extend the RIBIE and RIBIDE methods to three-dimensional problems of heat conduction, non-homogeneous Helmholtz and diffusion equations in an isotropic non-homogeneous medium using the fundamental solution for the three-dimensional Laplace equation.
- 4) The work in the present thesis could also be extended to anisotropic and orthotropic media, which is of great importance in engineering practice.

References

- [1] Brebbia C.A., Telles J.C.F., Wrobel L.C. Boundary element techniques, Berlin: Springer; 1984.
- [2] Partridge P.W., Brebbia C.A., Wrobel L.C. The dual reciprocity boundary element method. Computational Mechanics Publications: Southampton; 1992.
- [3] Brebbia C.A., Dominguez J. Boundary elements: an introductory course, second edition, McGraw-Hill, New York; 1992.
- [4] Paris F., Canas J. Boundary element method fundamentals and applications, New York: Oxford University Press; 1997.
- [5] Wrobel L.C. The boundary element method, vol.1, Chichester: Wiley; 2002.
- [6] Ang W.T. A beginner's course in boundary element methods, Boca Raton, USA: Universal Publishers; 2007.
- [7] Divo E.A., Kassab A.J. Boundary element methods for heat conduction: With applications in non-homogeneous media, WIT Press: Southampton; 2003.
- [8] Mikhailov S.E. Localized boundary-domain integral formulations for problems with variable coefficients. Engineering Analysis with Boundary Elements, 26:681-690; 2002.
- [9] Ang K.-C. Introducing the boundary element method with MATLAB, International Journal of Mathematical Education in Science and Technology, 39, 505-519; 2008.
- [10] Gockenbach M.S. Understanding and implementing the finite element method, Michigan Technological University, The Society for Industrial and Applied Mathematics, USA; 2006.

-
- [11] Sladek V., Sladek J., Zhang Ch. Domain element local integral equation method for potential problems in anisotropic and functionally graded materials. *Computational Mechanics*, 37:78-85; 2005.
- [12] Brunton I. Solving variable coefficient partial differential equations using the boundary element method, PhD thesis, University of Auckland, New Zealand; 1996.
- [13] Clements, D.L. A boundary integral equation method for the numerical solution of a second order elliptic equation with variable coefficients, *Journal of the Australian Mathematical Society*, 22 (Series B) 218-228; 1980.
- [14] Shaw R.P. Green's functions for heterogeneous media potential problems, *Engineering Analysis with Boundary Elements*, 13:219-221; 1994.
- [15] Shaw R.P, Gipson G.S. A BIE formulation of a linearly layered potential problem, *Engineering Analysis with Boundary Elements*, 16:1-3; 1995.
- [16] Ang W.T., Kusuma J., Clements, D.L. A boundary element method for a second order elliptic partial differential equation with variable coefficients, *Engineering Analysis with Boundary Elements*, 18:1-3; 1996.
- [17] Pomp A. Levi functions for linear elliptic system with variable coefficients including shell equations, *Computational Mechanics*, 22:93-99; 1998.
- [18] Venturini W.S. Further developments of boundary element formulation for zoned domain problems. In C.A Brebbia, J.J. Connor (eds), *Advances in Boundary Elements, Computational Mechanics Publications*, Southampton, and Springer-Verlag, Berlin, pp.251-267; 1989.
- [19] Mikhailov S.E., Nakhova I.S. Mesh-based numerical implementation of the localized boundary-domain integral equation method to a variable-coefficient Neumann problem, *Journal of Engineering Mathematics*, 51:251-259; 2005.
- [20] Skerget L., Rek Z. Boundary-domain integral method using a velocity-vorticity formulation, *Engineering Analysis with Boundary Elements*, 15:359-370; 1995.
- [21] Skerget L., Hribersek M., Kuhn G. Computational fluid dynamics by boundary-domain integral method, *International Journal for Numerical Methods in Engineering*, 46:1291-1311; 1999.

- [22] Jecl R., Skerget L., Petresin E. Boundary-domain integral method for transport phenomena in porous media, *International Journal for Numerical Methods in Fluids*, 35:39-54; 2001.
- [23] Popov V., Power H. The DRM-MD integral equation method: an efficient approach for the numerical solution of domain-dominant problems, *International Journal for Numerical Methods in Engineering*, 44:327-353; 1999.
- [24] Florez W., Power H., Chejne F. Multi-domain dual reciprocity BEM approach for the Navier-Stokes system of equations, *Communications in Numerical Methods in Engineering*, 16:671-681; 2000.
- [25] Peratta A., Popov V. A new scheme for numerical modelling of flow and transport processes in 3D fractured porous media, *Advances in Water Resources*, 29:42-61; 2006.
- [26] Katsikadelis J.T. The analog equation method-A powerful BEM-based solution technique for solving linear and nonlinear engineering problems, In: Brebbia C.A. (ed.), *Boundary Element Method XVI*, Computational Mechanics Publications, Southampton; pp. 167-182; 1994.
- [27] Katsikadelis J.T., Nerantzaki M.S. The boundary element method for nonlinear problems, *Engineering Analysis with Boundary Elements*, 23:365-373; 1999.
- [28] AL-Jawary M.A., Wrobel L.C., Maischak M. Numerical solution of a Dirichlet problem with variable coefficients by the boundary-domain integro-differential equation method. In: Lesnic D., editor. *Eighth UK Conference on Boundary Integral Methods*, University of Leeds Press, pp. 25-32; 2011.
- [29] AL-Jawary M.A., Wrobel L.C., Maischak M. Numerical solution of a Neumann problem with variable coefficients by the boundary-domain integral equation method. In: Lesnic D., editor. *Eighth UK Conference on Boundary Integral Methods*, University of Leeds Press, pp. 33-40; 2011.
- [30] AL-Jawary M.A., Wrobel L.C. Numerical solution of a mixed problem with variable coefficients by the boundary-domain integral and integro-differential equation methods, *The International Association for Boundary Element Methods*, (IABEM 2011), Brescia, Italy, University of Brescia Press, pp.19-26; 2011.

-
- [31] AL-Jawary M.A., Wrobel L.C. Numerical solution of two-dimensional mixed problems with variable coefficients by the boundary-domain integral and integro-differential equation methods, *Engineering Analysis with Boundary Elements*, 35:1279-1287; 2011.
- [32] Steinbach O. Numerical approximation methods for elliptic boundary value problems: finite and boundary elements, Springer, Berlin; 2008.
- [33] Mikhailov S.E. Finite-dimensional perturbations of linear operators and some applications to boundary integral equations, *Engineering Analysis with Boundary Elements*, 23:805-813; 1999.
- [34] Lesnic D. The boundary element method for solving the Laplace equation in two-dimensions with oblique derivative boundary conditions, *Communications in Numerical Methods in Engineering*, 23:1071-1080; 2007.
- [35] Shouxin W., Xiping L., Tianguo P., Zhongsheng Z., Suh Z. The BEM for solving the nonhomogeneous Helmholtz equation with variable coefficients, *Applied Mathematics and Mechanics*, 17:85-89; 1996.
- [36] Rangogni R. The solution of the nonhomogeneous Helmholtz equation by means of the boundary element method, *Applied Mathematical Modelling*, 8:442-444; 1984.
- [37] Blyth M.G., Pozrikidis C. A comparative study of the boundary and finite element methods for the Helmholtz equation in two dimensions, *Engineering Analysis with Boundary Elements*, 31:35-49; 2007.
- [38] Clements, D.L., Larsson, A. A boundary-element method for the solution of a class of time-dependent problems for inhomogeneous media, *Communications in Numerical Methods in Engineering*, 9:111-119; 1993.
- [39] Marin L., Elliott L., Heggs P.J., Ingham D.B., Lesnic D., Wen X. Dual reciprocity boundary element method solution of the Cauchy problem for Helmholtz-type equations with variable coefficients, *Journal of Sound and Vibration*, 297:89-105; 2006.
- [40] Marin L., Elliott L., Heggs P.J., Ingham D.B., Lesnic D., Wen X. Parameter identification in Helmholtz-type equations with a variable coefficient using a regularized DRBEM, *Inverse Problems in Science and Engineering*, 14:837-858; 2006.

-
- [41] Grigoriev M.M. and Dargush G.F. A fast multi-level boundary element method for the Helmholtz equation, *Computer Methods in Applied Mechanics and Engineering*, 193:165-203; 2004.
- [42] Power H. On the existence of Kassab and Divo's generalized boundary integral equation formulation for isotropic heterogeneous steady state heat conduction problems. *Engineering Analysis with Boundary Elements*, 20:341-345; 1997.
- [43] Bonnet M., Guiggiani M. Comments about the paper entitled "A generalized boundary integral equation for isotropic heat conduction with spatially varying thermal conductivity" by A. J. Kassab and E. Divo, *Engineering Analysis with Boundary Elements*, 22:235-240; 1998.
- [44] Nowak A.J., Brebbia C.A. The multiple-reciprocity method. A new approach for transforming B.E.M. domain integrals to the boundary, *Engineering Analysis with Boundary Elements*, 6:164-168; 1989.
- [45] Gao X.W. The radial integration method for evaluation of domain integrals with boundary-only discretization. *Engineering Analysis with Boundary Elements*, 26:905-916; 2002.
- [46] Nardini D, Brebbia C.A. A new approach for free vibration analysis using boundary elements, In *Boundary Element Methods in Engineering*, Brebbia C.A. (ed.). Springer: Berlin, 312-326; 1982.
- [47] Gao X.W. A boundary element method without internal cells for two-dimensional and three-dimensional elastoplastic problems. *Journal of Applied Mechanics (ASME)*, 69:154-160; 2002.
- [48] Gao X.W. Boundary element analysis in thermoelasticity with and without internal cells, *International Journal for Numerical Methods in Engineering*, 57:975-990; 2003.
- [49] Gao X.W. A meshless BEM for isotropic heat conduction problems with heat generation and spatially varying conductivity, *International Journal for Numerical Methods in Engineering*, 66:1411-1431; 2006.
- [50] Albuquerque E.L., Sollero P., Paiva W.P. The radial integration method applied to dynamic problems of anisotropic plates, *Communications in Numerical Methods in Engineering*, 23:805-818; 2007.

-
- [51] Gao X.W., Zhang C.H., Guo L. Boundary-only element solutions of 2D and 3D nonlinear and nonhomogeneous elastic problems, *Engineering Analysis with Boundary Elements*, 31:974-982; 2007.
- [52] Yang K, Gao X.W., Liu Y.F. Using analytical expressions in radial integration BEM for variable coefficient heat conduction problems, *Engineering Analysis with Boundary Elements*, 35:1085-1089; 2011.
- [53] Fata S.N. Treatment of domain integrals in boundary element methods, *Applied Numerical Mathematics*, 62:720-735; 2012.
- [54] Bridges T.R., Wrobel L.C. A dual reciprocity formulation for elasticity problems with body forces using augmented thin plate splines, *Communications in Numerical Methods in Engineering*, 12:209-220; 1996.
- [55] Karur S.R., Ramachandran P.A. Augmented thin plate spline approximation in DRM, *Boundary Elements Communications*, 6:55-58; 1995.
- [56] AL-Jawary M.A., Wrobel L.C. Radial integration boundary integral and integro-differential equation methods for two-dimensional heat conduction problems with variable coefficients, *Engineering Analysis with Boundary Elements*, 36:685-695; 2012.
- [57] AL-Jawary M.A., Wrobel L.C. Numerical solution of the two-dimensional Helmholtz equation with variable coefficients by the radial integration boundary integral and integro-differential equation methods, *International Journal of Computer Mathematics* (In press); 2012.
- [58] Yang K., Gao X.W. Radial integration BEM for transient heat conduction problems, *Engineering Analysis with Boundary Elements*, 34:557-563; 2010.
- [59] Wrobel L.C., Brebbia C.A., Nardini D. The dual reciprocity boundary element formulation for transient heat conduction, *Proc. VI International Conference on Finite Elements in Water Resources*, Springer-Verlag, New York, pp. 801-812; 1986.
- [60] Wrobel L.C., Brebbia C.A. The dual reciprocity boundary element formulation for nonlinear diffusion problems, *Computer Methods in Applied Mechanics and Engineering*, 65:147-164; 1987.

-
- [61] Hematiyan M.R., Karami G. A meshless boundary element method formulation for transient heat conduction problems with heat sources, *Scientia Iranica*, 15:348-359; 2008.
- [62] Mikhailov S.E. Analysis of united boundary-domain integro-differential and integral equations for a mixed BVP with variable coefficient, *Mathematical Methods in the Applied Sciences*, 29:715-739; 2006.
- [63] Chkadua O., Mikhailov S.E., Natroshvili D. Analysis of direct boundary-domain integral equations for a mixed BVP with variable coefficient, I: Equivalence and invertibility, *Journal of Integral Equations and Applications*, 21:499-543; 2009.
- [64] Hoitinga W. Direct minimization of equation residuals in least squares hp-finite element methods: a direct and iterative solution method, Internal Report, Delft University of Technology, The Netherlands; 2004.
- [65] Hansen P.B. Householder reduction of linear equations, *ACM Computing Surveys*, Association for Computing Machinery, 24:185-194; 1992.
- [66] Hoitinga W., de Groot R., Kwakkel M., Gerritsma M. Direct minimization of the least-squares spectral element functional - part I: direct solver, *Journal of Computational Physics*, 227:2411-2429; 2008.
- [67] Björck A. Least squares methods. Elsevier Science Publishers B.V., Holland; 1990.
- [68] Björck A. Numerical methods for least squares problems, SIAM, Philadelphia; 1996.
- [69] <http://www.mathworks.co.uk/help/toolbox/stats/regress.html>.
- [70] Chatterjee S., Hadi A.S. Influential observations, high leverage points, and outliers in linear regression, *Statistical Science*, 1:379-416; 1986.
- [71] Divo E.A., Kassab A.J. Transient non-linear heat conduction solution by a dual reciprocity boundary element method with an effective posteriori error estimator, *Computers, Materials, and Continua*, 2:277-288; 2005.
- [72] Application of DRM-Trefftz and DRM-MFS to transient heat conduction analysis, *Recent Patents on Space Technology*, 2:41-50; 2010.
- [73] Maischak M. Lecture notes in computational mathematics 2008/2009, Brunel University; UK.

- [74] Stroud A.H., Secrest D. Gaussian quadrature formulas, Prentice-Hall, Englewood Cliffs, NJ; 1966.
- [75] Stroud A. Approximate calculation of multiple integrals, Prentice-Hall, Englewood Cliffs, NJ; 1971.
- [76] Cowper G.R. Gauss quadrature formulas for triangle, International Journal for Numerical Methods in Engineering, 7:405-408; 1973.
- [77] Lyness J.N., Jespersen D. Moderate degree symmetric quadrature rules for the triangle, IMA Journal of Applied Mathematics, 15:19-32; 1975.
- [78] Akin J.E. Class notes: My huge quadrature database can be seen, f95 1/28/09, Rice University Department of Mechanical Engineering and Materials Science, Texas, USA. Available online: <http://www.clear.rice.edu/mech517/>.
- [79] Rathod H.T., Nagaraja K.V., Venkatesudu B., Ramesh N.L. Short Communication: Gauss Legendre quadrature over a triangle, Indian Institute of Science, 84:183-188; 2004.
- [80] Rathod H.T., Nagaraja K.V., Venkatesudu B. On the application of two symmetric Gauss Legendre quadrature rules for composite numerical integration over a triangle surface, Applied Mathematics and Computation, 190:21-39; 2007.
- [81] Atkinson K.E. The numerical solution of integral equations of the second kind, Cambridge University Press, Cambridge; 1997.
- [82] Duffy M.G. Quadrature over a pyramid or cube of integrands with a singularity at a vertex, SIAM Journal on Numerical Analysis, 19:1260-1262; 1982.
- [83] Mousavi S.E., Sukumar N. Generalized Duffy transformation for integrating vertex singularities, Computational Mechanics, 45:127-140; 2010.

Appendix A

Matlab codes for the three examples in chapter 2

Matlab programs to implement the BEM for the Laplace equation for the three examples in chapter 2 with given boundary conditions are given in this appendix. The main core of the code (in the processing stage) is the same for all three tests. However, for the pre-processing to create a uniform mesh of boundary coordinates for each side, values and type of boundary condition and post-processing to calculate the interior nodes and plot the figures (exact, approximate, difference between them and relative error), the programs are not the same but depend on the problem.

A.1 Main program for example 1

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

(Main program to calculate the approximate, exact solutions, the difference between them and relative error)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
1 clear;
2 clc;
3 N=[];E=[];
4 for i=1:4
5 r=2*i;
```

```

6  [s1,z1,d1,w1]=majeed4(r);
7  E(i)=(max(max(abs(w1)))/(max(max(abs(z1))))) ;
8  N(i)=4*r;
9  end
10 for j = 1:99
11 y(j) = 0.01*j;
12 for i = 1:99
13 x(i) = 0.01*i;
14 end
15 end
16 subplot(2,2,1); surface (x, y, z1,'EdgeColor','none');grid on
17 title('Exact solution')
18 subplot(2,2,2); surface (x, y, s1,'EdgeColor','none');grid on
19 title('Approximate solution')
20 subplot(2,2,3);surface (x, y, d1,'EdgeColor','none');grid on
21 title('Exact-Approximate')
22 subplot(2,2,4);loglog(N,E,'-o')
23 title('Relative error')

```

%%%

First stage: create a uniform mesh of boundary coordinates for each side , values and type of boundary condition

%%%

```

1  function [s1,z1,d1,w1]=majeed4(r)
2  for i=0:r-1
3  xb(i+1)=i/r;
4  yb(i+1)=0;
5  bt(i+1)=1;
6  bv(i+1)=0;
7  end
8  for i=0:r-1
9  xb(i+1+r)=1;
10 yb(i+1+r)=i/r;
11 bt(i+1+r)=0;
12 my=(i+0.5)/r;
13 mx=1;

```



```

1  for m = 1:n
2  b(m) = 0;
3  for k = 1:n
4  if(k == m)
5  G = 0.0;
6  F = lm(k)/(2.0*pi)*(log(lm(k)/2.0) - 1.0);
7  del = 1.0;
8  else
9  [F, G] =findfg (xm(m), ym(m), xb(k), yb(k), nx(k), ny(k), lm(k));
10 del = 0.0;
11 end
12 if (bt (k) == 0)
13 A (m, k) = -F;
14 b(m) = b(m) + bv(k)*(- G + 0.5*del);
15 else
16 A(m, k) = G -0.5*del;
17 b(m) = b(m) + bv(k)*F;
18 end
19 end
20 z=A\b';
21 end
22 for m = 1:n
23 u (m) = (1 - bt (m))*bv (m) + bt (m)*z(m);
24 q(m) = (1 - bt (m))*z(m) + bt(m)*bv (m);
25 end

```

%%%

Third stage: Find values at required points

%%%

```

1  for j = 1:99
2  y(j) = 0.01*j;
3  for i = 1:99
4  x(i) = 0.01*i;
5  s1(j,i) = 0;
6  for k = 1:n

```

```

7 [F, G] =findfg (x (i), y (j), xb (k), yb (k), nx (k), ny (k), lm(k));
8 s1(j,i) = s1(j,i) + u(k)*G - q(k)*F;
9 z1(j,i)=0;
10 y1=sinh(pi*x(i));
11 y2=cos(pi*y(j));
12 y3=sinh(pi);
13 z1(j,i)=z1(j,i)+(y1.*y2)/y3;
14 end
15 end
16 end
17 d1=z1-s1;w1=s1-z1;
18 end

```

%%%%%%%%%% End of main program.%%%%%%%%%%

When $k \neq m$, the integrals can be evaluated by using numerical methods such as Gauss quadrature which can be achieved by calling a function `quadl` and the code is:

```

1 function [F,G] =findfg(xi, eta, xk, yk, nkx, nky, lk)
2 F = (lk/(4.0*pi))*quadl(@(t) intf (t, xi, eta, xk, yk, nkx, nky, ...
   lk),0, 1, 1e-8);
3 G = (lk/(2.0*pi))*quadl(@(t) intg (t, xi, eta, xk, yk, nkx, nky, ...
   lk),0, 1, 1e-8);

```

where `intf` and `intg` are defined respectively as:

```

1 function y = intf (t, xi, eta, xk, yk, nkx, nky, lk)
2 y = log ((xk - t*lk*nky - xi).^2 + (yk + t*lk*nkx - eta).^2);
3 end

```

```

1 function y = intg(t, xi, eta, xk, yk, nkx, nky, lk)
2 y = (nkx*(xk - t*lk*nky - xi) + nky*(yk + t*lk*nkx - eta))./((xk ...
   - t*lk*nky - xi).^2 + (yk + t*lk*nkx-eta).^2);
3 end

```



```

1 function [s1,z1,d1,w1]=majeed10(r)
2 for i=1:r
3   xb(i)=(i-1)/r;
4   yb(i)=0;
5   bt(i)=1;
6   bv(i)=-1;
7 end
8 for i=1:2*r
9   x1=(pi)/(4*r);
10  xb(i+r)=cos((i-1)*x1);
11  yb(i+r)=sin((i-1)*x1);
12  bt(i+r)=0;
13  x1m=cos(((i+0.5)-1)*x1);
14  y1m=sin(((i+0.5)-1)*x1);
15  bv(i+r)=x1m+y1m;
16 end
17 for i=1:r
18  xb(i+3*r)= 0;
19  yb(i+3*r)=1-((i-1)/r);
20  bt(i+3*r)=0;
21  bv(i+3*r)=1-(((i+0.5)-1)/r);
22  xb(4*r+1)=xb(1);
23  yb(4*r+1)=yb(1);
24 end
25 n = 4*r;

```

%%%

Find midpoints, lengths of elements and their unit normal vectors

%%%

```

1 for i = 1:n
2   xm(i) = 0.5*(xb(i) + xb(i + 1));
3   ym(i) = 0.5*(yb(i) + yb(i + 1));
4   lm(i) = sqrt((xb(i + 1) - xb(i))^2 + (yb(i + 1) - yb(i))^2);
5   nx(i) = (yb(i + 1) - yb(i))/lm(i);
6   ny(i) = (xb(i) - xb(i + 1))/lm(i);
7 end

```

%%%

Second stage : find approximations for unknown boundary values by: constructing matrix A and vector b then solving the system "Ax=b" for x

%%%

```

1  for m = 1:n
2  b(m) = 0;
3  for k = 1:n
4  if(k == m)
5  G = 0.0;
6  F = lm(k)/(2.0*pi)*(log(lm(k)/2.0) - 1.0);
7  del = 1.0;
8  else
9  [F, G] =findfg (xm(m), ym(m), xb(k), yb(k), nx(k), ny(k), lm(k));
10 del = 0.0;
11 end
12 if (bt (k) == 0)
13 A (m, k) = -F;
14 b(m) = b(m) + bv(k)*(- G + 0.5*del);
15 else
16 A(m, k) = G -0.5*del;
17 b(m) = b(m) + bv(k)*F;
18 end
19 end
20 z=A\b';
21 end
22 for m = 1:n
23 u (m) = (1 - bt (m))*bv (m) + bt (m)*z(m);
24 q(m) = (1 - bt (m))*z(m) + bt(m)*bv (m);
25 end

```

%%%

Third stage: Find values at required points

%%%

```

1  for j = 1:99

```

```

2  y(j) = 0.01*j;
3  for i = 1:99
4  x(i) = 0.01*i;
5  y1=x(i);
6  y2=y(j);
7  s1(j,i) = 0;
8  for k = 1:n
9  if (y1.^2+y2.^2<1)
10 [F, G] = findfg (x (i), y (j), xb (k), yb (k), nx (k), ny ...
      (k), lm(k));
11 s1(j,i) = s1(j,i) + u(k)*G - q(k)*F;
12 else s1(j,i)=0;
13 end
14 z1(j,i)=0;
15 if (y1.^2+y2.^2<1)
16 z1(j,i)=z1(j,i)+(y1+y2);
17 else (z1(j,i)==0);
18 end
19 end
20 end
21 end
22 d1=z1-s1;w1=s1-z1;
23 end

```

%%%%%%%%%% End of main program.%%%%%%%%%%

A.3 Main program for example 3

%%%%%%%%%%

(Main program to calculate the approximate, exact solutions, the difference between them and relative error)

%%%%%%%%%%

```

1  clear;
2  clc;
3  N=[];E=[];

```

```

4 for i=3:6
5 r=2^i;
6 [s1,z1,d1,w1]=majeed11(r);
7 E(i)=(max(max(abs(w1))))/(max(max(abs(z1)))) ;
8 N(i)=8*r;
9 end
10 for j = 1:199
11 y(j) = 0.01*j;
12 for i = 1:199
13 x(i) = 0.01*i;
14 end
15 end
16 subplot(2,2,1); surface (x, y, z1,'EdgeColor','none')
17 title('Exact solution')
18 subplot(2,2,2); surface (x, y, s1,'EdgeColor','none')
19 title('Approximate solution')
20 subplot(2,2,3);surface (x, y, d1,'EdgeColor','none')
21 title('Exact-Approximate')
22 subplot(2,2,4);loglog(N,E,'-o')
23 title('Relative error')

```

%%%

First stage: create a uniform mesh of boundary coordinates for each side , values and type of boundary condition

%%%

```

1 function [s1,z1,d1,w1]=majeed11(r)
2 for i=1:r
3 xb(i)=1+((i-1)/r);
4 yb(i)=0;
5 bt(i)=1;
6 bv(i)= 0;
7 end
8 for i=1:4*r
9 x1=(pi)/(8*r);
10 xb(i+r)=2*cos((i-1)*x1);
11 yb(i+r)=2*sin((i-1)*x1);

```

```

12 bt(i+r)=0;
13 mx=2*cos(((i+0.5)-1)*x1);
14 my=2*sin(((i+0.5)-1)*x1);
15 bv(i+r)=3*cos(4*atan(my/mx));
16 end
17 for i=1:r
18 xb(i+5*r)= 0;
19 yb(i+5*r)=2-((i-1)/(r));
20 bt(i+5*r)=1;
21 bv(i+5*r)=0;
22 end
23 for i=1:2*r
24 x2=(pi)/(4*r);
25 xb(i+6*r)=sin((i-1)*x2);
26 yb(i+6*r)=cos((i-1)*x2);
27 bt(i+6*r)=0;
28 mx=sin(((i+0.5)-1)*x2);
29 my=cos(((i+0.5)-1)*x2);
30 bv((i+6*r))=cos(4*atan((my)/(mx)));
31 xb(8*r+1)=xb(1);
32 yb(8*r+1)=yb(1);
33 bt(8*r+1)=bt(1);
34 bv(8*r+1)=bv(1);
35 end
36 n = 8*r;

```

%%%

Find midpoints, lengths of elements and their unit normal vectors

%%%

```

1 for i = 1:n
2 xm(i) = 0.5*(xb(i) + xb(i + 1));
3 ym(i) = 0.5*(yb(i) + yb(i + 1));
4 lm(i) = sqrt((xb(i + 1) - xb(i))^2 + (yb(i + 1) - yb(i))^2);
5 nx(i) = (yb(i + 1) - yb(i))/lm(i);
6 ny(i) = (xb(i) - xb(i + 1))/lm(i);
7 end

```

%%%

Second stage : find approximations for unknown boundary values by: constructing matrix A and vector b then solving the system "Ax=b" for x

%%%

```

1  for m = 1:n
2  b(m) = 0;
3  for k = 1:n
4  if(k == m)
5  G = 0.0;
6  F = lm(k)/(2.0*pi)*(log(lm(k)/2.0) - 1.0);
7  del = 1.0;
8  else
9  [F, G] =findfg (xm(m), ym(m), xb(k), yb(k), nx(k), ny(k), lm(k));
10 del = 0.0;
11 end
12 if (bt (k) == 0)
13 A (m, k) = -F;
14 b(m) = b(m) + bv(k)*(- G + 0.5*del);
15 else
16 A(m, k) = G -0.5*del;
17 b(m) = b(m) + bv(k)*F;
18 end
19 end
20 z=A\b';
21 end
22 for m = 1:n
23 u (m) = (1 - bt (m))*bv (m) + bt (m)*z(m);
24 q(m) = (1 - bt (m))*z(m) + bt(m)*bv (m);
25 end

```

%%%

Third stage: Find values at required points

%%%

```

1  for j = 1:199

```

```

2 y(j) = 0.01*j;
3 for i = 1:199
4 x(i) = 0.01*i;
5 y1=x(i);
6 y2=y(j);
7 s1(j,i) = 0;
8 for k = 1:n
9     if ((y1.^2+y2.^2)>1)&&((y1.^2+y2.^2)<4)
10 [F, G] =findfg (x (i), y (j), xb (k), yb (k), nx (k), ny (k), lm(k));
11 s1(j,i) = s1(j,i) + u(k)*G - q(k)*F;
12 else s1(j,i)=0;
13 end
14 z1(j,i)=0;
15 if ((y1.^2+y2.^2)>1)&&((y1.^2+y2.^2)<4)
16 e1=(16/85*((y1.^2+y2.^2)^2-(1/(y1.^2+y2.^2)^2))...
17     -(16/255)*((y1.^2+y2.^2)^2/16)...
18     -(16)/(y1.^2+y2.^2)^2).*cos(4*atan(y2/y1));
19 z1(j,i)=z1(j,i)+(e1);
20 else (z1(j,i)==0);
21 end
22 end
23 end
24 end
25 d1=z1-s1;w1=s1-z1;
26 end

```

%%%%%%%%%% End of main program.%%%%%%%%%%

Appendix B

Numerical implementation for chapters 3 and 4

B.1 Computation of matrix A

In this section, we are going to explain how the elements of matrix A can be computed for some boundary and domain integrals that appear in chapters 3 and 4. Let us begin by introducing the reference triangular element $T_r = \{(t_1, t_2) \mid 0 \leq t_1, t_2 \leq 1, t_1 + t_2 \leq 1\}$ and defining a transformation

$$F : \begin{cases} T_r \Rightarrow T \\ t \Rightarrow x \end{cases}$$

such that $T \subset \Omega \subset \mathbb{R}^2$ and T is any triangle in our domain Ω . Also, if u is a function defined on $\Omega \subset \mathbb{R}^2$, we have $u(x) = \tilde{u}(F^{-1}(x)) = \tilde{u} \circ F^{-1}(x)$, where \tilde{u} is a function defined on T_r . So, we have with $x = F(t) \implies dx = |\frac{\partial F}{\partial t}| dt$. Since an affine transformation makes it possible to transform a reference triangle T_r to any triangle T , we have just to consider numerical integration on T_r .

B.1.1 Computation of sub-matrix Q'

$$Q'_{ij} = \int_{\partial\Omega \cap \bar{\omega}_j} P(x, x^i) T u(x) d\Gamma(x), \quad (\text{B.1})$$

which can be written as

$$Q'_{ij} = \int_{\partial\Omega \cap \bar{\omega}_j} P(x, x^i) a(x) \nabla_x u(x) n(x) d\Gamma(x),$$

since $[Tu](x) := a(x) \frac{\partial u(x)}{\partial n(x)}$, where $n(x)$ is the external normal unit vector to the boundary $\partial\Omega$ of the domain Ω .

Now our goal is to calculate the $\nabla_x u(x)$, by using slightly the same method in [73], which has been implemented for the finite element method, we get

$$Q'_{ij} = \int_{\partial\Omega \cap \bar{\omega}_j} P(x, x^i) a(x) \nabla_x \tilde{u}(F^{-1}(x)) n(x) d\Gamma(x). \quad (\text{B.2})$$

We have shown in section (B.1), that

$$u(x) = \tilde{u}(F^{-1}(x)) = \tilde{u} \circ F^{-1}(x)$$

Then

$$u(x) = \tilde{u}(t(x)), \text{ since } x = F(t).$$

Therefore,

$$u(x_1, x_2) = \tilde{u}(t_1(x_1, x_2), t_2(x_1, x_2)).$$

Taking the partial derivative with respect to x_1 for both sides, we get

$$\frac{\partial u(x_1, x_2)}{\partial x_1} = \frac{\partial \tilde{u}(t_1(x_1, x_2), t_2(x_1, x_2))}{\partial x_1} = \frac{\partial t_1}{\partial x_1} \cdot \frac{\partial \tilde{u}}{\partial t_1} + \frac{\partial t_2}{\partial x_1} \cdot \frac{\partial \tilde{u}}{\partial t_2}.$$

Similarly, taking the partial derivative with respect to x_2 for both sides, we get

$$\frac{\partial u(x_1, x_2)}{\partial x_2} = \frac{\partial \tilde{u}(t_1(x_1, x_2), t_2(x_1, x_2))}{\partial x_2} = \frac{\partial t_1}{\partial x_2} \cdot \frac{\partial \tilde{u}}{\partial t_1} + \frac{\partial t_2}{\partial x_2} \cdot \frac{\partial \tilde{u}}{\partial t_2}.$$

Therefore,

$$\frac{\partial u(x)}{\partial x_k} = \frac{\partial t_1}{\partial x_k} \cdot \frac{\partial \tilde{u}}{\partial t_1} + \frac{\partial t_2}{\partial x_k} \cdot \frac{\partial \tilde{u}}{\partial t_2} = \sum_{k=1}^2 \sum_{s=1}^2 \frac{\partial t_s}{\partial x_k} \frac{\partial \tilde{u}}{\partial t_s}.$$

Substituting the value of $\frac{\partial u(x)}{\partial x_k}$ in Eq.(B.2), we have

$$Q'_{ij} = \int_{\partial\Omega \cap \bar{\omega}_j} \left\{ P(x, x^i) a(x) n(x) \sum_{k=1}^2 \sum_{s=1}^2 \frac{\partial t_s}{\partial x_k} \frac{\partial \tilde{u}}{\partial t_s} \right\} d\Gamma(x),$$

$$= \int_{T_r} \left\{ P(x, x^i) a(x) n(x) \sum_{k=1}^2 \sum_{s=1}^2 \left(\frac{\partial t_s}{\partial x_k} \frac{\partial \tilde{u}}{\partial t_s} \right) \right\} \left| \frac{\partial F}{\partial t} \right| dt,$$

Due to $t = F^{-1} \circ F(t)$ we have

$$I = \nabla_t t = \nabla_t F^{-1} \circ F(t) = [(\nabla_x F^{-1}) \circ F(t)] \nabla_t F(t)$$

that means

$$I = \left(\frac{\partial t_s}{\partial t_j} \right)_{s,j} = \left(\frac{\partial t_s(x(t))}{\partial t_j} \right)_{s,j} = \left(\sum_{k=1}^2 \frac{\partial t_s}{\partial x_k} \frac{\partial x_k}{\partial t_j} \right)_{s,j} = \left(\frac{\partial t_s}{\partial x_k} \right)_{s,k} \left(\frac{\partial x_k}{\partial t_j} \right)_{k,j}$$

leading to

$$\left(\frac{\partial t_s}{\partial x_k} \right)_{s,k} = \left[\left(\frac{\partial x_k}{\partial t_j} \right)_{k,j} \right]^{-1}.$$

Therefore, we obtain in two dimensions

$$\begin{pmatrix} \frac{\partial t_1}{\partial x_1} & \frac{\partial t_1}{\partial x_2} \\ \frac{\partial t_2}{\partial x_1} & \frac{\partial t_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_1}{\partial t_1} & \frac{\partial x_1}{\partial t_2} \\ \frac{\partial x_2}{\partial t_1} & \frac{\partial x_2}{\partial t_2} \end{pmatrix}^{-1} = \frac{1}{\left| \frac{\partial F}{\partial t} \right|} \begin{pmatrix} \frac{\partial x_2}{\partial t_2} & -\frac{\partial x_1}{\partial t_2} \\ -\frac{\partial x_2}{\partial t_1} & \frac{\partial x_1}{\partial t_1} \end{pmatrix}$$

with $\left| \frac{\partial F}{\partial t} \right| = \frac{\partial x_1}{\partial t_1} \frac{\partial x_2}{\partial t_2} - \frac{\partial x_1}{\partial t_2} \frac{\partial x_2}{\partial t_1}$.

Therefore,

$$Q'_{ij} = \int_{T_r} \left\{ P(x, x^i) a(x) n(x) \left[\left[\left(\frac{\partial x_2}{\partial t_2}, -\frac{\partial x_2}{\partial t_1} \right) \nabla_t \tilde{u} \right] + \left[\left(-\frac{\partial x_1}{\partial t_2}, -\frac{\partial x_1}{\partial t_1} \right) \nabla_t \tilde{u} \right] \right] \right\} \frac{1}{\left| \frac{\partial F}{\partial t} \right|} \left| \frac{\partial F}{\partial t} \right| dt. \quad (\text{B.3})$$

Finally,

$$Q'_{ij} = \int_{T_r} \left\{ P(x, x^i) a(x) n(x) \left[\left[\left(\frac{\partial x_2}{\partial t_2}, -\frac{\partial x_2}{\partial t_1} \right) \partial_{t_1} \tilde{u} \right] + \left[\left(-\frac{\partial x_1}{\partial t_2}, -\frac{\partial x_1}{\partial t_1} \right) \partial_{t_2} \tilde{u} \right] \right] \right\} dt. \quad (\text{B.4})$$

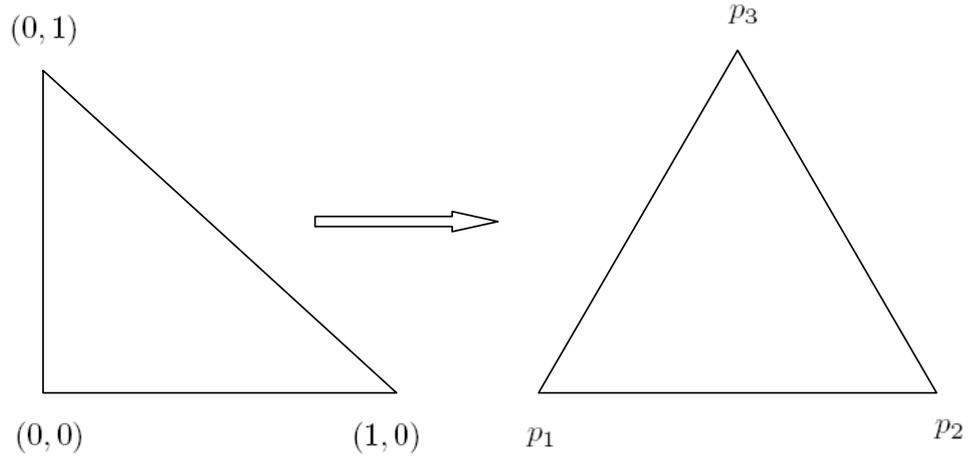


Figure B.1: Mapping reference triangle to arbitrary triangle

B.2 Mapping the reference triangle to an arbitrary triangle

Given the reference triangle T_r with vertices $(0,0)$, $(1,0)$, $(0,1)$ and a triangle T of the mesh in Ω of vertices p_1 , p_2 , p_3 , the transformation $F : T_r \rightarrow T$ is given by [10, 73]

$$F(t) = a + a_1 t_1 + a_2 t_2$$

and its coefficients depend on T . Using F , we can write that

$$p_1 = a, \quad p_2 = a + a_1, \quad p_3 = a + a_2,$$

and this gives

$$a = p_1, \quad a_1 = p_2 - p_1, \quad a_2 = p_3 - p_1.$$

Therefore, we get the Jacobian matrix

$$\frac{\partial F}{\partial t} = \begin{pmatrix} \frac{\partial F_1}{\partial t_1} & \frac{\partial F_1}{\partial t_2} \\ \frac{\partial F_2}{\partial t_1} & \frac{\partial F_2}{\partial t_2} \end{pmatrix} = \begin{pmatrix} a_1(1) & a_2(1) \\ a_1(2) & a_2(2) \end{pmatrix}. \quad (\text{B.5})$$

For the determinant we get,

$$\left| \frac{\partial F}{\partial t} \right| = a_1(1)a_2(2) - a_1(2)a_2(1).$$

B.2.1 Computation of sub-matrix K'

$$K'_{ij} = \int_{\omega_j} \Phi_j(x) R(x, x^i) d\Omega(x), \tag{B.6}$$

which can be written using the reference triangle element,

$$K'_{ij} = \int_{T_r} \Phi_j(F(t)) R(F(t), x^i) \left| \frac{\partial F}{\partial t} \right| dt, \tag{B.7}$$

where, $F(t) = a + a_1 t_1 + a_2 t_2$ and $\left| \frac{\partial F}{\partial t} \right| = a_1(1)a_2(2) - a_1(2)a_2(1)$.

B.3 Numerical integration of boundary integrals

The standard BEM for two-dimensional problems requires the numerical integration of shape functions, and the product of shape functions with the fundamental solution or its derivative. The Gauss Legendre quadrature formula is widely used see for example [74, 75].

B.3.1 Mapping reference interval to arbitrary interval

Suppose the reference interval $I_r = [-1, 1]$, and arbitrary interval $I = [p_1, p_2]$, the transformation $\tilde{D} : I_r \rightarrow I$ is given by

$$\tilde{D}(t) = a + a_1 t_1,$$

where $a = \frac{p_1+p_2}{2}$, and $a_1 = \frac{p_2-p_1}{2}$. Therefore we get the Jacobian, $\left| \frac{d\tilde{D}}{dt} \right| = |a_1|$.

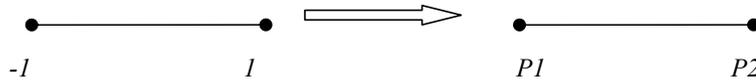


Figure B.2: Mapping reference interval to arbitrary interval

Therefore, to find the value of one boundary integral given in chapter 3 by the Gauss Legendre quadrature formula:

$$\int_{\partial\Omega} \bar{u}(x) T_x P(x, x^i) d\Gamma(x) = \sum_{s=1}^N \int_{\Gamma_s} \sum_{m=1}^K \bar{u}(\tilde{D}(t)) T_x P(\tilde{D}(t), x^i) |a_1| w(m) dt,$$

where N is the number of boundary elements, K is the number of Gauss points and w are the Gauss weights. In our work, we used an eight points Gaussian quadrature rule to get sufficiently accurate results.

B.4 Numerical integration of domain integrals

Applying the BEM for problems with internal sources requires a domain discretisation into a number of internal cells (triangular elements), and then numerical integration over such triangles. Since an affine transformation makes it possible to transform the reference triangle T_r to any triangle T , we have just to consider numerical integration on T_r as discussed in subsection B.2. The integral of an arbitrary function f , over the reference triangle T_r is given by

$$I = \iint_{T_r} f(t_1, t_2) dt_1 dt_2 = \int_0^1 dt_1 \int_0^{1-t_1} f(t_1, t_2) dt_2 = \int_0^1 dt_2 \int_0^{1-t_2} f(t_1, t_2) dt_1. \quad (\text{B.8})$$

Therefore, in order to find the value of the integral in Eq.(B.8), there are many methods, as discussed in the next sub-sections.

B.4.1 Symmetric Gauss quadrature formula for unit right triangle

The quadrature formula for numerical integration of the integral in Eq.(B.8), can be written in the standard form

$$I = \sum_{m=1}^K w(m) f(t_1(m), t_2(m)),$$

where $w(m)$ are the weights associated with specific points $(t_1(m), t_2(m))$ and K is the number of points. In [76], the author presented symmetric Gauss quadrature formulas up to 13 integration points using Cartesian co-ordinates originating at the centroid. Also, [77] derived quadrature rules for a triangle with vertices at $(-1, 0)$, $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ and $(\frac{1}{2}, -\frac{\sqrt{3}}{2})$ and produced weights and points for quadrature rules up to 28 integration points for this equilateral triangle. For a recent review, see [78], an on-line database containing symmetric Gauss quadrature formulas for a unit right triangle up to 13 integration points;

these formulas are especially useful for numerical integration of domain integrals for the finite element method and our boundary-domain integral and integro-differential equation (BDIE and BDIDE) methods. The three points, see table B.1, we have been tested in our work and provided good results. However, for BDIE or BDIDE, the domain integrals with a singularity either of log type or $1/r$ type, need special treatment to overcome such problems.

B.4.2 Gauss Legendre quadrature formula for unit right triangle

In this method, the reference triangle, unit right triangle T_r with vertices $(0,0)$, $(1,0)$, $(0,1)$, in (t_1, t_2) space, map to a standard square S in (q_1, q_2) space : $\{(q_1, q_2) | 0 \leq q_1, q_2 \leq 1\}$, see Figure B.3.

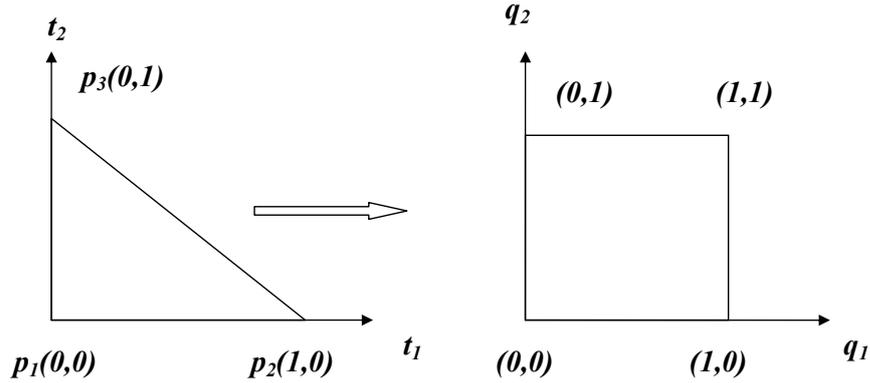


Figure B.3: Mapping unit right triangle to unit square

This overcomes the difficulties associated with the derivation of new weight and integration points and yields accurate and reliable results, see [79, 80] . To do so, let us introduce the following transformation:

$$t_1 = q_1, t_2 = (1 - q_1) * q_2$$

Then, the determinant of the Jacobian and the differential area are:

$$\left| \frac{\partial(t_1, t_2)}{\partial(q_1, q_2)} \right| = \left| \begin{pmatrix} 1 & 0 \\ -q_2 & (1 - q_1) \end{pmatrix} \right| = (1 - q_1). \tag{B.9}$$

Number of points	t_1	t_2	w
1	1/3	1/3	1/2
3	1/6	1/6	1/6
	2/3	1/6	1/6
	1/6	2/3	1/6
4	1/3	1/3	-27/96
	0.2	0.2	25/96
	0.2	0.6	25/96
	0.6	0.2	25/96
6	0.81684757	0.091576214	0.054975872
	0.091576214	0.091576214	0.054975872
	0.091576214	0.81684757	0.054975872
	0.10810302	0.44594849	0.111690795
	0.44594849	0.44594849	0.111690795
	0.44594849	0.10810302	0.111690795
7	0.10128651	0.10128651	0.06296959
	0.47014206	0.05971587	0.06619708
	0.79742699	0.10128651	0.06296959
	0.47014206	0.47014206	0.06619708
	0.10128651	0.79742699	0.06296959
	0.05971587	0.47014206	0.06619708
	1/3	1/3	0.11250000
12	0.87382197	0.063089014	0.025422453
	0.063089014	0.063089014	0.025422453
	0.063089014	0.87382197	0.025422453
	0.50142651	0.24928675	0.058393138
	0.24928675	0.24928675	0.058393138
	0.24928675	0.50142651	0.058393138
	0.63650250	0.31035245	0.041425538
	0.31035245	0.053145050	0.041425538
	0.053145050	0.63650250	0.041425538
	0.63650250	0.053145050	0.041425538
	0.31035245	0.63650250	0.041425538
	0.053145050	0.31035245	0.041425538
13	1/3	1/3	- 0.074785022
	0.47930807	0.26034597	0.087807629
	0.26034597	0.26034597	0.087807629
	0.26034597	0.47930807	0.087807629
	0.86973979	0.065130103	0.026673618
	0.065130103	0.065130103	0.026673618
	0.065130103	0.86973979	0.026673618
	0.63844419	0.31286550	0.038556880
	0.31286550	0.048690315	0.038556880
	0.048690315	0.63844419	0.038556880
	0.63844419	0.048690315	0.038556880
	0.31286550	0.63844419	0.038556880
	0.048690315	0.31286550	0.038556880

Table B.1: Symmetric Gauss quadrature formula for unit right triangle

Then, Eq.(B.8), can be written

$$I = \int_0^1 dt_1 \int_0^{1-t_1} f(t_1, t_2) dt_2 = \int_0^1 \int_0^1 f(q_1, (1 - q_1) * q_2)(1 - q_1) dq_1 dq_2 \quad (\text{B.10})$$

In order to solve the integral in Eq.(B.10) using the product of standard Gauss Legendre formulas for x and y directions, there are two possible ways:

First, the integral in Eq.(B.10), can be transformed further into an integral over a standard square R in (r_1, r_2) space : $\{(r_1, r_2) | -1 \leq r_1, r_2 \leq 1\}$, by introducing the following transformation:

$$q_1 = \frac{(1 + r_1)}{2}, \quad q_2 = \frac{(1 + r_2)}{2}.$$

Then, the determinant of the Jacobian and the differential area are:

$$\left| \frac{\partial(q_1, q_2)}{\partial(r_1, r_2)} \right| = \left| \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \right| = \frac{1}{4}. \quad (\text{B.11})$$

Now, using the new transformation, the integral in Eq.(B.10) can be written

$$I = \int_0^1 dt_1 \int_0^{1-t_1} f(t_1, t_2) dt_2 = \int_{-1}^1 \int_{-1}^1 f\left(\frac{1+r_1}{2}, \frac{(1-r_1)(1+r_2)}{4}\right) \left(\frac{1-r_1}{8}\right) dr_1 dr_2. \quad (\text{B.12})$$

Then, the integral in Eq.(B.12) can be written using a quadrature formula, see for example [75],

$$I = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1-r_1(i)}{8}\right) w(i)w(j) f(t_1(r_1(i), r_2(j)), t_2(r_1(i), r_2(j))),$$

where $r_1(i)$, $r_2(i)$ are Gaussian points in the r_1 , r_2 directions, respectively, and $w(i)$ and $w(j)$ are the corresponding weights.

Second, solving the integral in Eq.(B.10) without mapping it to a new square R in (r_1, r_2) space : $\{(r_1, r_2) | -1 \leq r_1, r_2 \leq 1\}$, can be achieved just by shifting the original and standard Gaussian points from $[-1, 1]$ to $[0, 1]$ by using the transformation:

$$s = \frac{(t+1)}{2}; \quad s \in [0, 1]; \quad t \in [-1, 1].$$

Then, the integral in Eq.(B.10) can be written using the quadrature formula

$$I = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1 - q_1(i)}{4} \right) w(i)w(j)f(t_1(q_1(i), q_2(j)), t_2(q_1(i), q_2(j))),$$

where $q_1(i), q_2(i)$ are Gaussian points in the q_1, q_2 directions on $[0, 1]$, respectively.

B.4.3 Other useful quadrature formulas

We found three additional formulas that are useful in applications to treat the regular and weakly singular domain integrals.

First, a 7-points formula from [81] page 171. The integral in Eq.(B.8) can be calculated using the values of t_1, t_2 and w in table B.2:

$$I = \int_0^1 dt_1 \int_0^{1-t_1} f(t_1, t_2)dt_2 = \sum_{m=1}^7 w(m)f(t_1(m), t_2(m)). \tag{B.13}$$

Number of points	t_1	t_2	w
7	1/3	1/3	9/80
	$(6 - \sqrt{15})/21$	$(6 - \sqrt{15})/21$	$(155 - \sqrt{15})/2400$
	$(6 - \sqrt{15})/21$	$(9 + 2*\sqrt{15})/21$	$(155 - \sqrt{15})/2400$
	$(9 + 2*\sqrt{15})/21$	$(6 - \sqrt{15})/21$	$(155 - \sqrt{15})/2400$
	$(6 + \sqrt{15})/21$	$(6 + \sqrt{15})/21$	$(155 + \sqrt{15})/2400$
	$(6 + \sqrt{15})/21$	$(9 - 2*\sqrt{15})/21$	$(155 + \sqrt{15})/2400$
	$(9 - 2*\sqrt{15})/21$	$(6 + \sqrt{15})/21$	$(155 + \sqrt{15})/2400$

Table B.2: 7-points quadrature formula

Second, a 16-points formula from [6] pages 208-209. The integral in Eq.(B.10) can be calculated using the values of q_1, q_2 and w in table B.3:

$$I = \int_0^1 dt_1 \int_0^{1-t_1} f(t_1, t_2)dt_2 = \int_0^1 \int_0^1 f(q_1, (1 - q_1) * q_2)(1 - q_1)dq_1dq_2 = \sum_{m=1}^{16} w(m)f(q_1(m), (1 - q_1(m)) * q_2(m))(1 - q_1(m)). \tag{B.14}$$

Third, a 16-points formulas from [75] page 314. The integral in Eq.(B.8) can be calculated

Number of points	t_1	t_2	w
16	$1/4+1/(4*\sqrt{3})$	$1/4+1/(4*\sqrt{3})$	1/16
	$1/4+1/(4*\sqrt{3})$	$1/4-1/(4*\sqrt{3})$	1/16
	$1/4-1/(4*\sqrt{3})$	$1/4+1/(4*\sqrt{3})$	1/16
	$1/4-1/(4*\sqrt{3})$	$1/4-1/(4*\sqrt{3})$	1/16
	$3/4+1/(4*\sqrt{3})$	$1/4+1/(4*\sqrt{3})$	1/16
	$3/4+1/(4*\sqrt{3})$	$1/4-1/(4*\sqrt{3})$	1/16
	$3/4-1/(4*\sqrt{3})$	$1/4+1/(4*\sqrt{3})$	1/16
	$3/4-1/(4*\sqrt{3})$	$1/4-1/(4*\sqrt{3})$	1/16
	$3/4+1/(4*\sqrt{3})$	$3/4+1/(4*\sqrt{3})$	1/16
	$3/4+1/(4*\sqrt{3})$	$3/4-1/(4*\sqrt{3})$	1/16
	$3/4-1/(4*\sqrt{3})$	$3/4+1/(4*\sqrt{3})$	1/16
	$3/4-1/(4*\sqrt{3})$	$3/4-1/(4*\sqrt{3})$	1/16
	$1/4+1/(4*\sqrt{3})$	$3/4+1/(4*\sqrt{3})$	1/16
	$1/4+1/(4*\sqrt{3})$	$3/4-1/(4*\sqrt{3})$	1/16
	$1/4-1/(4*\sqrt{3})$	$3/4+1/(4*\sqrt{3})$	1/16
	$1/4-1/(4*\sqrt{3})$	$3/4-1/(4*\sqrt{3})$	1/16

Table B.3: 16-points quadrature formula

using the values of t_1 , t_2 and w in table B.4:

$$I = \int_0^1 dt_1 \int_0^{1-t_1} f(t_1, t_2) dt_2 = \sum_{m=1}^{16} w(m) f(t_1(m), t_2(m)). \quad (\text{B.15})$$

B.5 Treatment of weak singularity for domain integrals using Duffy transformation

Multidimensional integrals of singular functions can be troublesome to evaluate numerically. This is particularly clear when the singularity occurs at corner points. In order to solve the weakly singular integrands, variable transformation method has been used to treat such type of singularity, see [5]. The main idea in this method is to map the reference triangle element to a square element, then the singularity is removed through the introduction of the Jacobian. The important transformation which is widely used, the so-called Duffy transformation, see [82], from a triangle in 2D and a pyramid in 3D to a square and cube respectively is: $(q_1, q_2, q_3) \rightarrow (t_1, t_2, t_3) : t_1 = q_1, t_2 = t_1 q_2 = q_1 q_2, t_3 = t_1 q_3 = q_1 q_3$, which eliminates singularities of the type $1/r$, see Figure B.3. The new kernel after the mapping over a square in two-dimensions is smooth enough and can be integrated using

Number of points	t_1	t_2	w
16	0.0571041961	0.0694318422*(1-0.0571041961)	0.023568368192143
	0.2768430136	0.0694318422*(1-0.2768430136)	0.035388067902662
	0.5835904324	0.0694318422*(1-0.5835904324)	0.022584049284999
	0.8602401357	0.0694318422*(1-0.8602401357)	0.005423225902803
	0.0571041961	0.3300094782*(1-0.0571041961)	0.044185088507857
	0.2768430136	0.3300094782*(1-0.2768430136)	0.066344216097338
	0.5835904324	0.3300094782*(1-0.5835904324)	0.042339724515001
	0.8602401357	0.3300094782*(1-0.8602401357)	0.010167259547197
	0.0571041961	0.6699905218*(1-0.0571041961)	0.044185088507857
	0.2768430136	0.6699905218*(1-0.2768430136)	0.066344216097338
	0.5835904324	0.6699905218*(1-0.5835904324)	0.042339724515001
	0.8602401357	0.6699905218*(1-0.8602401357)	0.010167259547197
	0.0571041961	0.9305681558*(1-0.0571041961)	0.023568368192143
	0.2768430136	0.9305681558*(1-0.2768430136)	0.035388067902662
	0.5835904324	0.9305681558*(1-0.5835904324)	0.022584049284999
	0.8602401357	0.9305681558*(1-0.8602401357)	0.005423225902803

Table B.4: 16-points Stroud quadrature formula

Gauss Legendre quadrature, as discussed in subsection B.4.2.

In our implementation the collocation points live always on the vertices of the triangles. However, when the singularity falls inside an element (this case does not appear in the current thesis), then the element can be divided into triangles in \mathbb{R}^2 and pyramids in \mathbb{R}^3 with the singularity lying at a vertex of the subdivisions and the transformation can be applied to each subdomain separately.

Even though the Duffy transformation works very well for a $1/r$ singularity, it is not as efficient for $1/r^\alpha$ when $\alpha \neq 1$; to treat such type of singularities, see [83].

Fortunately, the boundary-domain integral and integro-differential equation (BDIE and BDIDE) methods only have weak singular domain integrals with either log type or $1/r$ type. The two domain integrals come from either the remainder $R(x, y)$ with $1/r$ singularity or from the right-hand side $f(x)$ multiplied by the parametrix $P(x, y)$, which has a log singularity coming from the parametrix, and both are weakly singular.

$$\int_{\omega_j} \Phi_j(x) R(x, x^i) d\Omega(x), \tag{B.16}$$

$$\int_{\Omega} f(x) P(x, x^i) d\Omega(x). \tag{B.17}$$

For numerical implementation, the simple way to calculate such domain integrals with weak singularity is to use the quadrature formulas in subsection (B.4). In fact, these formulas are working fine and produce good results. However, in order to achieve better accuracy, we have implemented in our Matlab code Gaussian quadrature rule for two dimensions with Duffy transformation. The collocation points are at vertices of the triangles, so when we do the integration over each element the singularity is either in p_1 , p_2 or p_3 in Figure B.3.

Suppose that p_2 is a singular point; in this case, the simple transformation is given in subsection B.4.2:

$$t_1 = q_1, t_2 = (1 - q_1) * q_2.$$

Then, the determinant of the Jacobian is $(1 - q_1)$. Also, we have derived other transformations for p_1 , and p_3 by:

$$t_1 = (1 - q_1) * q_2, t_2 = (1 - q_1) * (1 - q_2),$$

with the determinant of the Jacobian being $(1 - q_1)$ when p_1 is the singular point. Finally, when p_3 is singular the transformation will be:

$$t_1 = (1 - q_1) * q_2, t_2 = q_1,$$

with the same determinant of the Jacobian equal to $(1 - q_1)$. It will be useful to mention that when we do the integration over a regular element, we can just use the same procedure discussed in subsection B.4.2.

Appendix C

Matlab codes for calculating of the radial integral in chapter 5

Matlab programs for the analytical calculation of the radial integral in Eqs.(5.9) and (5.13) appearing in chapter 5, and general numerical codes for using the RIM to numerically calculate the general boundary integral in Eqs.(5.9) and (5.13) are implemented in this appendix.

%%%

Analytic calculation of the radial integral in Eq.(5.9)

%%%

```
1 function result=Radialintegral1
2 clear;
3 clc;
4 syms x1 x2 y1 y2 r1 r2 r
5 val=f([y1+r1*r,y2+r2*r]);
6 result=int(val*r,r,0,r);
7 result=subs(result,r1,(x1-y1)/r);
8 result=subs(result,r2,(x2-y2)/r);
9 result=subs(result,x1,'x(1)');
10 result=subs(result,x2,'x(2)');
11 result=subs(result,y1,'y(1)');
12 result=subs(result,y2,'y(2)');
13 result=simplify(result);
```

```

14 end
15 function val=f(x)
16 %val=2;
17 %val=x(1)+x(2);
18 %val=x(1)^2+x(2)^2;
19 %val=x(1)^3+x(2)^3;
20 %val=exp(x(1)+x(2));
21 val=exp(x(1)+x(2))+cos(x(1)+x(2))+sin(x(1)+x(2))+log(x(1)+x(2));
22 end

```

%%%

Analytic calculation of the radial integral in Eq.(5.13)

%%%

```

1 function result=Radialintegral2
2 clear;
3 clc;
4 syms x1 x2 y1 y2 r1 r2 ra r t
5 val=f([y1+r1*r*t,y2+r2*r*t]);
6 result=int(val*r*r*t,t,0,1);
7 result=subs(result,r1,(x1-y1)/r);
8 result=subs(result,r2,(x2-y2)/r);
9 result=subs(result,x1,'x(1)');
10 result=subs(result,x2,'x(2)');
11 result=subs(result,y1,'y(1)');
12 result=subs(result,y2,'y(2)');
13 result=simplify(result);
14 end
15 function val=f(x)
16 %val=2;
17 %val=x(1)+x(2);
18 %val=x(1)^2+x(2)^2;
19 %val=x(1)^3+x(2)^3;
20 %val=exp(x(1)+x(2));
21 val=exp(x(1)+x(2))+cos(x(1)+x(2))+sin(x(1)+x(2))+log(x(1)+x(2));
22 end

```

%%%

The main Matlab code for numerical calculation of the boundary integral in Eq.(5.10) with corresponding numerical calculation of the radial integral in Eqs.(5.9) and (5.13)

%%%

```

1 clear;
2 clc;
3 for i=1:2
4   [coord nodes]=ex2mesh(2^i);
5   neigh=setneigh(nodes);
6   bp3=m3(coord,nodes,neigh);
7 end

```

%%%

This function generates the coordinates of vertices and the number of corners for each triangle, we will have n triangles per edge. The algorithm uses the special structure of a square $[1, 2] \times [1, 2]$

%%%

```

1 function [coord nodes]=ex2mesh(n)
2 coord=[]; nodes=[];
3 nnodes=(n+1)^2;
4 coord=zeros(nnodes,2);
5 nodes=zeros(n^2*2,3);
6 for i=0:n
7   xi=i/n;
8   txi=xi+1;
9   for j=0:n
10    xj=j/n;
11    txj=xj+1;
12    coord(i*(n+1)+j+1,1)=txi;
13    coord(i*(n+1)+j+1,2)=txj;
14  end
15 end
16 for i=0:n-1
17  for j=0:n-1

```

```

18     n1=i*(n+1)+j+1;
19     n2=(i+1)*(n+1)+j+1;
20     n3=(i+1)*(n+1)+j+2;
21     n4=i*(n+1)+j+2;
22     nodes((i*n+j)*2+1,1:3)=[n1; n2; n3];
23     nodes((i*n+j)*2+2,1:3)=[n1; n4; n3];
24     end
25 end

```

%%%

This function to check the boundary edges for each element, if neigh=0 means the edge is on the boundary (it has no neighbor)

%%%

```

1 function neigh=setneigh(nodes)
2 neigh=zeros(size(nodes));
3 [n1 n2]=size(nodes);
4 nnodes=max(max(nodes)); % largest node number, max(nodes) gives ...
   max for columns
5 noel=zeros(nnodes,1); % number of elements per node
6 for i=1:n1
7     for k=1:n2
8         noel(nodes(i,k))=noel(nodes(i,k))+1;
9     end
10 end
11 maxnoel=max(noel);
12 noel=zeros(nnodes,1); elem=zeros(nnodes,maxnoel);
13 for i=1:n1
14     for k=1:n2
15         node=nodes(i,k);
16         noel(node)=noel(node)+1;
17         elem(node,noel(node))=i;
18     end
19 end
20 % now compare only elements which are attached to the same node
21 for node=1:nnodes
22     for p=1:noel(node)

```

```

23  i=elem(node,p);
24  x(1:n2)=nodes(i,:); x(n2+1)=x(1);
25  for q=p+1:noel(node)
26  j=elem(node,q);
27  y(1:n2)=nodes(j,:); y(n2+1)=y(1);
28  for k=1:n2
29  if neigh(i,k)>0
30  continue % neighbour already found, try next edge, skip rest
31  end
32  for l=1:n2
33  if neigh(j,l)>0
34  continue % neighbour already found, try next edge, skip rest
35  end
36  if (x(k)==y(l) && x(k+1)==y(l+1) ) || (x(k)==y(l+1) && x(k+1)==y(l))
37  neigh(i,k)=j;
38  neigh(j,l)=i;
39  break % exit inner loop, test next edge for k
40  end
41  end
42  end
43  end
44  end
45  end

```

%%%

Find the end boundary nodes, and their unit normal vectors

%%%

```

1  function [nx,p1,p2]=normalvector(l,k,nodes,coord)
2  n(1:3)=nodes(l,1:3);
3  n(4)=n(1);
4  p1=coord(n(k),:);
5  p2=coord(n(k+1),:);
6  t=p2-p1;
7  nx(2)=-t(1);
8  nx(1)=t(2);
9  nx=nx/norm(nx);

```

```

10 c=(coord(n(1),:)+coord(n(2),:)+coord(n(3),:))/3;
11 m=1/2*(p1+p2);
12 if((c-m)*nx'>0)
13 nx=-nx;
14 end
15 end

```

%%%

The numerical calculation of the boundary integral in Eq.(5.10) with corresponding numerical calculation of the radial integral in Eqs.(5.9) and (5.13)

%%%

```

1 function bp3=m3(coord,nodes,neigh)
2 [b1 b2]=size(coord);nnodes=max(max(nodes));bp3=zeros(nnodes,1);
3 for i=1:b1
4     bp3(i)=intt(coord(i,:),coord,nodes,neigh);
5 end
6 end
7
8 function val=intt(y,coord,nodes,neigh)
9 val=0;[n1 n2]=size(nodes);
10 for i=1:n1
11     for k=1:3
12         if neigh(i,k)==0
13             [nx,p1,p2]=normalvector(i,k,nodes,coord);
14             val=val+intuelem2(y,p1,p2,nx);
15         end
16     end
17 end
18 end
19
20 function b=intuelem2(y,p1,p2,nx)
21 a=(p1+p2)/2; a1=(p2-p1)/2;
22 b=0;
23 t=[-0.861136311594000    -0.339981043585000    0.339981043585000    ...
24     0.861136311594000];
25 w=[0.347854845137000    0.652145154863000    0.652145154863000    ...

```

```
    0.347854845137000];
25 for q=1:4
26 m=a+a1*t(q);
27 b=b+rf(m,y,nx)*norm(a1)*w(q);
28 end
29 end
30 function val=rf(x,y,nx)
31 r=norm(x-y);
32 integrand=@(t) t.*funh(repmat(y',1,size(t,2))+(x-y)'*t/r); % ...
    integrand of Eq.(5.9)
33 % integrand=@(t) r.^2.*t.*funh(repmat(y',1,size(t,2))+(x-y)'*t); ...
    % integrand of Eq.(5.13)
34 I=quad(integrand,0,r); % numerical integration of Eq.(5.9)
35 % I=quad(integrand,0,1); % numerical integration of Eq.(5.13)
36 val=I*((x-y)/r^2)*nx';
37 end
38
39 function val=funh(x)
40 %val=2;
41 %val=x(1,:)+x(2,:);
42 %val=x(1,:).^2+x(2,:).^2;
43 %val=x(1,:).^3+x(2,:).^3;
44 %val=exp(x(1,:)+x(2,:));
45 val=exp(x(1)+x(2))+cos(x(1)+x(2))+sin(x(1)+x(2))+log(x(1)+x(2));
46 end
```

Appendix D

Numerical results for RIBIE for heat conduction

In this appendix, we applied the new implementations of the RIBIE method discussed in chapter 8, to three steady-state heat conduction problems on a square domain recalled them from chapter 3 (Poisson and variable coefficients). Also, the new implementation of the BIE for the Laplace equation, leads to the same results as in chapters 3 (for Laplace equation) and 6 (for Poisson and variable coefficients).

D.1 Laplace's equation with mixed boundary conditions

A square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = 1$, $f(x) = 0$ for $x \in \bar{\Omega}$ with mixed boundary conditions:

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 3 + x_1, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = n_1(x) + n_2(x), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

D.2 Poisson's equation with mixed boundary conditions

Assume a square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = 1$, $f(x) = 4$ for $x \in \bar{\Omega}$, with boundary conditions,

$$\bar{u}(x) = 4 + x_1^2, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 9 + x_1^2, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = 2(x_1 n_1(x) + x_2 n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1^2 + x_2^2$, $x \in \bar{\Omega}$.

D.3 Variable coefficients

Square domain $\bar{\Omega} = \{(x_1, x_2) : 2 \leq x_1, x_2 \leq 3\}$, $a(x) = 2(x_1 + x_2)$, $f(x) = 4$ for $x \in \bar{\Omega}$, with boundary conditions:

$$\bar{u}(x) = 2 + x_1, \text{ for } x_2 = 2; \quad 2 \leq x_1 \leq 3,$$

$$\bar{u}(x) = 3 + x_1, \text{ for } x_2 = 3; \quad 2 \leq x_1 \leq 3,$$

$$\bar{t}(x) = 2(x_1 + x_2)(n_1(x) + n_2(x)), \text{ for } x_1 = 2 \text{ or } x_1 = 3; \quad 2 \leq x_2 \leq 3.$$

The exact solution for this problem is $u_{exact}(x) = x_1 + x_2$, $x \in \bar{\Omega}$.

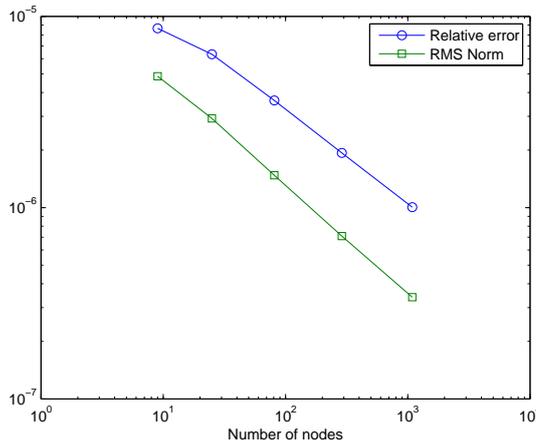


Figure D.1: Relative and RMS errors for BIE method for Laplace's equation

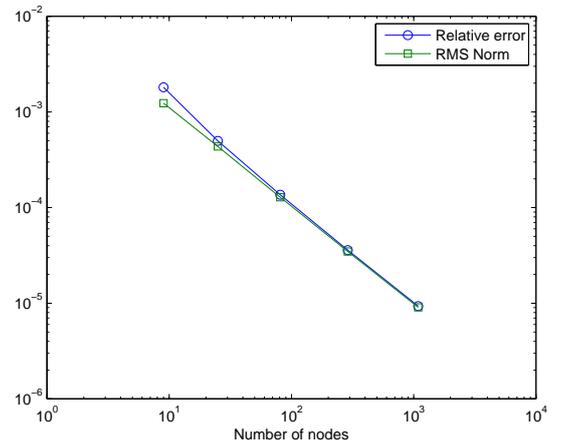


Figure D.2: Relative and RMS errors for RIBIE method for Poisson's equation

It can be clearly seen from Figs.D.1, D.2 and D.3 that the new implementation of the BIE and the RIBIE for Laplace, Poisson and variable coefficients equations provides satisfactory results. As the exact solution for Laplace and variable coefficients is linear, there are no interpolation errors in both cases. However, there is interpolation error for the Poisson

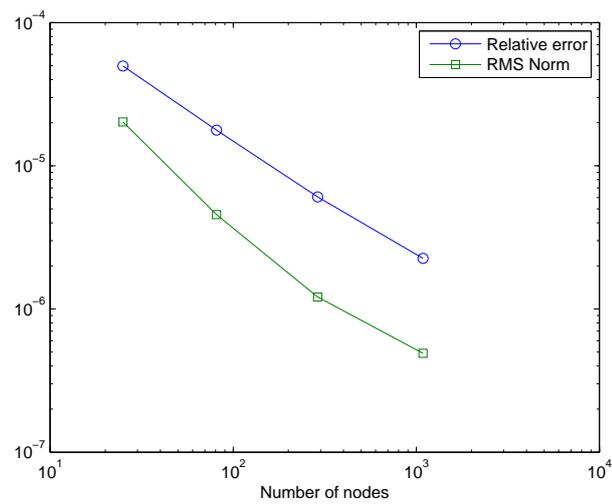


Figure D.3: Relative and RMS errors for RIBIE method for variable coefficients

equation, as the exact solution is quadratic, and lower accuracy is achieved in comparison to the other cases. Moreover, it can be seen the convergence of the solution by increasing the number of nodes, for all tests.