Boundary element techniques for some two-dimensional anisotropic heat conduction problems

Choo, Kin Kei

2005

Choo, K. K. (2005). Boundary element techniques for some two-dimensional anisotropic heat conduction problems. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/5636>

<https://doi.org/10.32657/10356/5636>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:14:26 SGT

Boundary Element Techniques for Some Two-Dimensional Anisotropic Heat Conduction Problems

Choo Kin Kei

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Master of Engineering

2005

Abstract

With extensive applications of composite materials in engineering, the analysis of heat conduction in such materials with numerical techniques has gained great importance. In this thesis, three different types of two-dimensional heat conduction problems are solved numerically using the Boundary Element Method (BEM).

For the first type, the problem is governed by a parabolic partial differential equation (PDE) with variable coefficients. Such a problem has applications for anisotropic inhomogeneous solids. A dual-reciprocity method (DRM) using the concept of fundamental solution is employed to convert the domain integral in the integral solution of the PDE to a discretised boundary integral equation (BIE) without the domain being discretised. The time domain is discretised into time elements and some polynomial time functions are used to approximate the time-dependent nodal variables such as temperature and normal heat flux.

The second type problem is governed by a third order PDE over a homogeneous anisotropic medium. The PDE is derived using a non-Fourier heat flux model which may account for thermal waves and/or microscopic effects. The DRM is employed to obtain the discretised BIE and a modified polynomial time function is used to approximate the time elements.

The BEM analysis of a steady-state temperature field in a thermally anisotropic bimaterial, which has homogeneously imperfect interface, is considered in the third type of problems. There is a temperature jump across the imperfect interface. To obtain a BIE without the interface being discretised, a modified Green's function (fundamental solution) which satisfies the required interface conditions is derived.

In all these BEM analyses, discontinuous linear elements are used to model the boundaries and the variables (temperature and normal flux) along the boundary elements. The systems of algebraic equations are set up to solve all the unknowns. For the purpose of analysing each proposed method, computer programmes are written to help to solve some numerical examples with known exact solutions. The BEM results are compared and plotted together with the exact ones. The numerical results obtained agree well with the exact solutions.

Acknowledgement

The author wishes to express his deepest appreciation and gratitude to his supervisor, Associate Professor Ang Whye Teong, for his guidance, encouragement and contributions during the course of his research and also for his confidence in me and the painstaking review of this thesis. His innovative ideas, knowledge and past experiences have also been a source of inspiration.

Gratitude is also extended to the Division of Engineering Mechanics, School of Mechanical and Aerospace Engineering, Nanyang Technological University for their financial support and co-operation.

Contents

Title Page	i
Abstract	ii
Acknowledgement.....	iii
Contents	iv
List of Tables.....	vi
List of Figures	vii
Abbreviations and Notations	viii
Chapter 1 Introduction to Boundary Element Method and Heat Conduction.....	1
1.1 Computational Techniques in Engineering	1
1.1.1 Boundary Element Method.....	3
1.1.2 BEM: Advantages and Disadvantages.....	4
1.2 Mathematical Preliminaries for the BEM.....	5
1.2.1 The Gradient Theorem.....	5
1.2.2 Gauss Divergence Theorem.....	6
1.2.3 Integration-by-parts Theorem.....	6
1.2.4 Green's First and Second Identities.....	7
1.2.5 2D Dirac-delta Function	7
1.3 Fundamental Theory of Heat Conduction in Solids	8
1.3.1 Law of Conservation of Heat Energy	8
1.3.2 Heat Conduction as (Initial-)Boundary Value Problems.....	11
1.4 Introduction of Work in Subsequent Chapters	13
Chapter 2 Unsteady-State Anisotropic Heat Conduction in Inhomogeneous	
Materials.....	18
2.1 PDE Transformation	18

2.2	Integro-Differential Equation.....	19
2.3	BEM with Discontinuous Linear Elements.....	20
2.4	Dual-Reciprocity Method.....	23
2.5	Time-Marching (TM) Scheme with Polynomial Approximation.....	24
2.6	Final System of Equation.....	26
2.7	Specific Test Problems.....	27
Chapter 3 Non-Classical Heat Conduction in Homogeneous Anisotropic Materials		36
3.1	Integro-Differential Equation	36
3.2	Dual-Reciprocity Boundary Element Method (DRBEM)	37
3.3	Time-Marching Scheme with Modified Polynomial Approximation.....	38
3.4	Final System of Equation.....	39
3.5	Specific Test Problems	40
Chapter 4 Steady-State Anisotropic Heat Conduction Across a Homogeneously		
Imperfect Interfaces		47
4.1	Boundary Integral Equation.....	47
4.2	Derivation of a Special Green's Function.....	50
4.3	BIE with DLEs Discretisation.....	52
4.4	Specific Test Problems.....	53
Chapter 5 Conclusions and Future Works.....		59
5.1	Summary and Conclusions	59
5.2	Recommendations for Future Work	60
References		R1
Appendix A		A1
A1	Analytical Formulae for Some Definite Integrals.....	A1
A2	Integration Parameters for Chapter 4.....	A3

A2.1	Explicit Formulae for $I_*^{(k)}(\xi)$ and $II_*^{(k)}(\xi)$	A3
A2.2	Explicit Formulae for $I_{\text{per}}^{(k)}(\xi)$ and $II_{\text{per}}^{(k)}(\xi)$	A4
A2.3	Analytical Formulae for $I_{\text{imp}}^{(k)}(\xi)$ and $II_{\text{imp}}^{(k)}(\xi)$	A5
A2.4	Analytical Formulae for $\Phi_{\text{imp}}^{\pm}$ and $\phi_{\text{imp}}^{\pm}$	A8
Appendix B		A9
B1	MATLAB Code for Chapter 2	A9
B2	MATLAB Code for Chapter 3	A18
B3	MATLAB Code for Chapter 4	A27

List of Tables

Table 2.1: Exact and computed numerical results at interior point $\xi=(1.5,1)$ obtained with

$M=20, K_{\Omega}=11, \Delta t=0.1$ (*Problem 2.1*)..... 31

Table 2.2: Corresponding results obtained with $M=41, K_{\Omega}=24, \Delta t=0.1$. (*Problem 2.1*)..... 31

Table 2.3: Exact $T(\xi, t)$ at four interior points. (*Problem 2.2*)..... 34

Table 2.4: Results computed with $K_{\tau}=4, \Delta t=0.1$ and $\zeta=0$. (*Problem 2.2*)..... 34

Table 3.1: Exact $T(\xi, t)$. (*Problem 3.1*)..... 42

Table 3.2: Results computed with $K_{\tau}=4, \Delta t=0.1, \zeta=0$. (*Problem 3.1*)..... 42

Table 3.3: Results computed at $\xi=(0.3,0.8)$ with different K_{τ} at $M=16, K_{\Omega}=9, \Delta t=0.08, \zeta=0$

(*Problem 3.2*)..... 44

Table 3.4: Results computed at $\xi=(0.3,0.8)$ with different K_{τ} at $M=32, K_{\Omega}=16, \Delta t=0.08, \zeta=0$

(*Problem 3.2*)..... 44

Table 4.1: Results at interior points (*Problem 4.1*)..... 55

Table 4.2: Results at selected points on boundary $x_1=0$ ($=\Gamma_2 \cup \Gamma_3$) (*Problem 4.1*)..... 55

Table 4.3: Results at interior points (*Problem 4.2*)..... 58

List of Figures

Figure 1.1: Finite difference grid	2
Figure 1.2: Finite element mesh.....	2
Figure 1.3: Boundary element mesh	3
Figure 1.4: 3D domain Ω bounded by surface Γ	5
Figure 1.5: Definition of λ and internal angle ϖ	8
Figure 1.6: 2D domain Ω bounded by curves Γ_1 and Γ_2	11
Figure 1.7: A bimaterial with a flat interface.....	16
Figure 2.1: Linear functions over rectilinear elements and the mapping in natural co- ordinate system.....	20
Figure 2.2: Case considered in <i>Problem 2.1</i> ($M=20, K_{\Omega}=11$).	28
Figure 2.3: Values at boundary nodes at $t=4.5$ with $M=20, K_{\Omega}=11, K_{\Gamma}=4, \Delta t=0.1, \zeta=0$. (<i>Problem 2.1</i>).....	32
Figure 2.4: Exact and computed values at boundary nodes at $t=2$ with $M=41, K_{\Omega}=24, K_{\Gamma}=4$, $\Delta t=0.1$ and $\zeta=0$. (<i>Problem 2.2</i>).....	35
Figure 3.1: Case considered in <i>Problem 3.1</i> ($M=16, K_{\Omega}=9$).	40
Figure 3.2: Plotting of exact and computed results at 3 interior points with $M=20, K_{\Omega}=25$, $K_{\Gamma}=5, \Delta t=0.06, \zeta=0$. (<i>Problem 3.3</i>)	46
Figure 3.3: Exact and computed values on boundary at $t=1.2$ with $M=20, K_{\Omega}=25, K_{\Gamma}=5$, $\Delta t=0.06, \zeta=0$. (<i>Problem 3.3</i>)	46
Figure 4.1: Case considered in <i>Problem 4.1</i> ($M=20$).	53
Figure 4.2: Exact and computed normal flux at boundary nodes on Γ_1 with $M=42$. (<i>Problem 4.1</i>).....	56
Figure 4.3: Case considered in <i>Problem 4.2</i> ($M=48$)	56
Figure 4.4: Exact and computed T and q on boundary nodes with $M=48$.(<i>Problem 4.2</i>)	58

Abbreviations and Notations

2D	Two-dimensional
3D	Three-dimensional
BC	Boundary condition
BEM	Boundary Element Method
BIE	Boundary integral equation
BVP	Boundary value problem
CLE	Continuous linear element
DLE	Discontinuous linear element
FDM	Finite difference method
FEM	Finite element method
DRBEM	Dual-reciprocity boundary element method
DRM	Dual-reciprocity method
IBVP	Initial-boundary value problem
LT	Laplace transform
PDE	Partial differential equation
TM	Time-Marching
C_p	Specific heat capacity of solids, [J/kg K]
Ei	Exponential integral
H	Heaviside unit-step function
K_r	Total number of nodal points on boundary
K_Ω	Total number of well-spaced out interior points
K_τ	Total number of time nodes in time element
L	Length of boundary elements
M	Total number of boundary elements
P	Coefficient of polynomial time function
Re	Real part of complex term
T	Temperature, [K]
$\dot{T}$	Time derivative of T
$\dot{V}$	Rate of internal energy generated per-unit volume inside solids, [W/m ³]
c_{ij}	Coefficient of thermal heat conductivity of solids, [W/m K]
$c_{ij}^\pm$	Coefficient of thermal heat conductivity of solids in bimaterial problem
$e, \exp$	Exponential
h	Heat flux vector [W/m ²]

---

i	$\sqrt{-1}$
k, l, p	Index of nodal points
$\hat{k}$	2 if k =odd, 1 if k =even
$\tilde{k}, m$	Index of boundary elements
l_1, l_2	Positions (0~0.5) of boundary nodal points in DLE mesh
$\mathbf{n}$	unit outward normal vector on boundary: $[n_1, n_2, n_3]$ for 3D, $[n_1, n_2]$ for 2D
q	Outward normal heat flux through the boundary, $[\text{W}/\text{m}^2]$
r, s	Index of time nodes
t	Time
Δt	Small time interval between time nodes
u	Intrinsic (natural) co-ordinate: $-1 \leq u \leq 1$
$\mathbf{x}$	Co-ordinate (x_1, x_2) for 2D case, (x_1, x_2, x_3) for 3D case
I, II	Integration parameters
Ω	2D/3D domain
$\Omega^\pm$	Domain in bimaterial problem
Γ	Boundary for Ω
$\Gamma^\pm$	Boundary for $\Omega^\pm$
Γ^0	Interface in bimaterial problem
Φ	Fundamental solution
$\Phi^\pm$	Modified Green's function for bimaterial problem
$\delta(\mathbf{x}-\xi)$	Dirac-delta function
δ_{pk}	Kronecker-delta
$\phi, \phi^\pm$	Normal derivative of Φ and $\Phi^\pm$
ζ	Ratio ($0 \leq \zeta < 1$) indicating position of t to be collocated
χ	DRM interpolation function
ρ	Mass density of solids, $[\text{kg}/\text{m}^3]$
ξ	Interior point (ξ_1, ξ_2) in 2D Ω , (ξ_1, ξ_2, ξ_3) in 3D Ω
τ_h	Small positive constant of phase lags of the heat flux components
τ_T	Small positive constant of phase lags of the temperature gradient
$\langle \mathbf{x} \rangle, \langle \mathbf{x}^{(k)} \rangle, \langle \xi \rangle$	In bimaterial problem: ' $\pm$ ' sign depending on the point inside the angle bracket.
$\overline{\langle \mathbf{x}^{(k)} \rangle}, \overline{\langle \xi \rangle}$	Sign opposite to $\langle \mathbf{x}^{(k)} \rangle$ and $\langle \xi \rangle$
$\langle \mathbf{x}^{(k)}, \xi \rangle$	In bimaterial problem: 1 if both $\mathbf{x}^{(k)}$ and ξ reside on the same material domain, otherwise -1.

Chapter 1 Introduction to Boundary Element Method and Heat Conduction

---

The study of heat transfer has gained great importance, as modern technology becomes more and more complex. In almost every phase of scientific and engineering work, processes involving the exchange of energy through heat flow are encountered. Mechanical and chemical engineering are particularly concerned with problems of heat transfer. Almost all pieces of apparatus used in these areas involve heat transfer by one means or another such as conduction, convection and radiation.

Many man-made materials that exhibit complexity in properties are widely used in various aspects of industrials. Hence the knowledge and analysis of heat conduction of such materials are particularly important nowadays. Some examples of such materials are laminated sheets, fibre-reinforced structures, heat shielding materials for space vehicles and functionally graded material (FGM) for electronic components.

In most of the cases, the geometries of the materials under consideration as well as the conditions imposed on the boundary complicate the required thermal analyses. Problems associated with inhomogeneous anisotropic materials are also inherently difficult to solve analytically. As such, one has to resort to using computational techniques for solving the mathematical problem involved in an approximate way.

Computational techniques such as Finite Difference Method (FDM) and Finite Element Method (FEM) have long been used to solve various heat conduction problems and other types of initial-boundary value problems (IBVPs). However, with the rapid development of the Boundary Element Method (BEM) and also due to certain advantages it has over FDM and FEM, solving heat conduction problems with BEM has received considerable attentions from researchers these days.

1.1 Computational Techniques in Engineering

Computational techniques for the numerical solution of engineering problems governed by PDEs can be grouped into three major classes: finite difference, finite element, and boundary element methods.

All the three methods involve the following procedures. (1) The replacement of the governing PDEs and relevant initial-boundary conditions with approximate algebraic

equations. (2) The use of a computational grid (usually called a mesh) to describe the geometry of the problem under consideration. The unknown function is determined at a finite set of points on this grid. (3) The approximate algebraic equations are solved to determine the unknown function on the grid.

A brief description of each of the methods is given below.

Figure 1.1: Finite difference grid

Figure 1.1 shows the computational grids used in the Finite Difference Method (FDM). The FDM is a simple approach which can be applied directly to the governing ordinary differential equation or PDEs. The derivatives of the unknown parameters are approximated by means of algebraic expressions in terms of the differences between the numerical values of the parameters at close points. The problem domain is usually represented in terms of a grid of nodes, and the difference equations result in a system of algebraic equations in terms of the values of the parameters at the specified nodes. The arrangement of the grids is chosen for the purpose of deriving suitable finite-difference formulae to approximate the derivatives in the differential equations. Treating irregular boundaries may prove to be cumbersome in the FDM. For more information on FDM, one may refer to Ames [3].

Figure 1.2: Finite element mesh

Figure 1.2 shows a typical mesh used in the Finite Element Method (FEM). The solution domain is divided into a number of smaller and simpler sub-domains known as finite elements. The relevant variational principle may then be applied over each element to obtain

an elemental equation for the unknown values at specified nodes. The equations from all the elements are then assembled together to obtain a matrix equation involving all the nodal values. The matrix equation can then be solved together with the given boundary conditions. The FEM is now a well-established and popular numerical tool among engineers. For details, one may refer to Chandrupatla and Belegunda [14].

1.1.1 Boundary Element Method

Figure 1.3: Boundary element mesh

Figure 1.3 shows a boundary element mesh that might be used for many problems of practical interest. In the BEM, the governing differential equation is transformed into an equivalent integral equation through the use of a suitable fundamental solution and a reciprocal theorem. In many problems, it is possible to obtain an integral equation which is taken over only the boundary of the solution domain. For such cases, in approximating the BIE in terms of linear algebraic equations, only nodes on the boundary are needed, as shown in Figure 1.3. However, for many other problems, such as those involving inhomogeneous solids, domain integrals may appear in the integral formulation. One popular method for treating the domain integrals is the dual-reciprocity method (DRM). The DRM requires the use of collocation nodes in the interior of the solution domain, but it does not require the entire solution domain to be meshed up into small elements as in the FEM. For further details on BEM, one may refer to an introductory textbook such as Hartmann [23] and París and Cañas [29].

The BEM has been successfully developed and applied to a great number of problems, such as potential theory and its applications in heat, fluid mechanics, electrical and magnetic theory, stress in elasticity problems, contact mechanics and acoustic analysis in mining and tunneling. However, thermal analysis with BEM in heat conduction in solids is the main objective of this report.

1.1.2 BEM: Advantages and Disadvantages

In the present report, the BEM will be developed and implemented on the computer for the numerical solution of several important classes of problems for heat conduction. Thus, it may be of some interest to discuss the advantages and disadvantages of the BEM in comparison with other domain based numerical techniques like the FEM.

The advantages of the BEM are as follows.

- (i) For many applications, the dimensionality of the problem is reduced by one. This results in a considerable reduction in the size of numerical calculation and computer time required for the analysis. This happens when the PDE can be transformed into an equivalent integral equation taken over only the boundary of the solution domain.
- (ii) No meshing of the solution domain into elements is needed, even if the integral formulation involves domain integrals. The domain integrals can be treated using DRM.
- (iii) The BEM is ideal for problems with infinite domains, such as problems of soil mechanics, fluid mechanics, and acoustics.
- (iv) Surface problems, such as those of elastic fracture mechanics, or elastic contact, are dealt with more efficiently and economically with the BEM.
- (v) The BEM offers a fully continuous solution inside the domain, and the problem parameters can be evaluated directly at any point there.

The BEM has, of course, its own drawbacks which have the effect of making it less popular among engineers, at least in the early stage of its development. The disadvantages can be summarised as follows.

- (i) The derivation of the integral equations may require a level of mathematics higher than that of other methods, but the procedure of the BEM itself is no different from that of the FEM. Without an appropriate fundamental solution, it is not possible to obtain integral equations which are taken over the boundary only.
- (ii) The BEM gives rise to fully populated matrices for the equations to be solved. Thus, it is not possible to employ elegant FEM solvers (for sparse matrices) such as the banded or frontal solver with the BEM.
- (iii) The integral formulation of non-linear problems and those problems involving inhomogeneous media may contain domain integrals. The main advantage of the dimensionality reduction mentioned above may be lost, even though in all cases it is possible to use various techniques to treat the domain integrals without having to discretise the entire solution domain into elements.

- (iv) A straightforward application of the method may not give accurate results for problems involving narrow strips or curved shell structures. However, this drawback can be overcome if suitable Green's functions can be derived for the narrow strips or curved shell structures.

For further details of the advantages and disadvantages mentioned above, one may refer to the textbook by El-Zafrany [20]. A historical account of the development of the BEM may also be found in the same book.

1.2 Mathematical Preliminaries for the BEM

Some mathematical preliminaries needed for the development of the BEM are introduced in this section. They are used for deriving integral formulations for the problems considered in subsequent chapters. Some of the results below can be found in standard engineering textbooks such as Kreyszig [26].

1.2.1 The Gradient Theorem

Figure 1.4: 3D domain Ω bounded by surface Γ

Consider a finite space 3D domain Ω which is defined in the Euclidean space using the Cartesian rectangular co-ordinates (x_1, x_2, x_3) and is bounded by a piecewise smooth orientable surface Γ , see Figure 1.4. If $f(x_1, x_2, x_3)$ is a continuous scalar function with at least second-order continuous partial derivatives, then

$$\begin{aligned} \int_{\Omega} \frac{\partial f}{\partial x_i} d\Omega &= \oint_{\Gamma} n_i f d\Gamma, \quad (i=1,2,3) \\ \text{or} \\ \int_{\Omega} \nabla f d\Omega &= \oint_{\Gamma} \mathbf{n} f d\Gamma \quad (\text{Gradient Theorem}) \end{aligned} \tag{1.1}$$

where $d\Omega = dx_1 dx_2 dx_3$, $\nabla = [\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}]$ denotes directional derivative (grad), and $\mathbf{n} = [n_1, n_2, n_3]$ is unit outward normal vector to the boundary Γ .

The Gradient Theorem relates the volume integration of the derivative to a surface integration of the function itself.

1.2.2 Gauss Divergence Theorem

Replacing f by ∇f in (1.1), we obtain the Gauss Divergence Theorem:

$$\begin{aligned} \int_{\Omega} \nabla^2 f d\Omega &= \oint_{\Gamma} \mathbf{n} \cdot \nabla f d\Gamma = \oint_{\Gamma} \frac{\partial f}{\partial n} d\Gamma \quad (\text{matrix notation}) \\ \int_{\Omega} f_{,ii} d\Omega &= \oint_{\Gamma} n_i f_{,i} d\Gamma \quad (\text{index notation}) \end{aligned} \quad (1.2)$$

where $\nabla^2 = \nabla \cdot \nabla = \text{div } \nabla$ is known as the Laplace operator, $\partial/\partial n = \nabla \cdot \mathbf{n} = n_i \partial/\partial x_i$ (sum over i) denotes normal derivative operator on surface Γ , $f_{,i} = \frac{\partial f}{\partial x_i}$ and $f_{,ii} = \frac{\partial^2 f}{\partial x_i \partial x_i} = \sum_{i=1}^3 \frac{\partial^2 f}{\partial x_i \partial x_i}$. Note that through out this report Einsteinian convention of summing over a repeated index holds for Latin subscripts i and j from 1 to 2 (for 2D case) or from 1 to 3 (for 3D case).

1.2.3 Integration-by-parts Theorem

Using $\Phi \partial f / \partial x_j$ ($j=1,2,3$) instead of f in (1.1), where $\Phi(x_1, x_2, x_3)$ is continuous function having at least second-order continuous partial derivatives, one finds that

$$\int_{\Omega} \left(\Phi \frac{\partial^2 f}{\partial x_i \partial x_j} + \frac{\partial \Phi}{\partial x_i} \frac{\partial f}{\partial x_j} \right) d\Omega = \oint_{\Gamma} n_i \Phi \frac{\partial f}{\partial x_j} d\Gamma \quad (1.3)$$

Interchanging f and Φ in (1.3) gives:

$$\int_{\Omega} \left(f \frac{\partial^2 \Phi}{\partial x_i \partial x_j} + \frac{\partial f}{\partial x_i} \frac{\partial \Phi}{\partial x_j} \right) d\Omega = \oint_{\Gamma} n_i f \frac{\partial \Phi}{\partial x_j} d\Gamma. \quad (1.4)$$

Subtracting (1.4) from (1.3) yields

$$\int_{\Omega} \left(\Phi \frac{\partial^2 f}{\partial x_i \partial x_j} - f \frac{\partial^2 \Phi}{\partial x_i \partial x_j} + \frac{\partial \Phi}{\partial x_i} \frac{\partial f}{\partial x_j} - \frac{\partial f}{\partial x_i} \frac{\partial \Phi}{\partial x_j} \right) d\Omega = \oint_{\Gamma} \left(n_i \Phi \frac{\partial f}{\partial x_j} - n_i f \frac{\partial \Phi}{\partial x_j} \right) d\Gamma \quad (1.5)$$

As one shall see later on in Chapter 2, equation (1.5) plays a key role in deriving integral formulations for the problems under consideration in the present thesis.

1.2.4 Green's First and Second Identities

Changing j to i and using $i=1,2,3$ in (1.3) and then adding the resulting equations, we deduce that

$$\int_{\Omega} (\Phi \nabla^2 f + \nabla \Phi \bullet \nabla f) d\Omega = \oint_{\Gamma} \Phi \frac{\partial f}{\partial n} d\Gamma$$

The above result is known as **Green's first identity**. In a similar manner, the **Green's second identity** can be deduced from (1.5) as:

$$\int_{\Omega} (\Phi \nabla^2 f - f \nabla^2 \Phi) d\Omega = \oint_{\Gamma} \left(\Phi \frac{\partial f}{\partial n} - f \frac{\partial \Phi}{\partial n} \right) d\Gamma, \quad (1.6)$$

which is also known as Green's reciprocity theorem. Equation (1.6) has extensive applications in analysis involving the Laplace equation. Note that all the above results for 3D domains are also valid for 2D domains.

1.2.5 2D Dirac-delta Function

In the analysis of many engineering problems, the concept of a source at a point is often used, e.g. concentrated loads in solid mechanics and concentrated sources of internal energy generation in thermal analysis. The Dirac-delta function was originally introduced as a suitable mathematical representation of this concept.

Consider a small circle with radius ε , centred at (ξ_1, ξ_2) inside a 2D domain Ω bounded by a simple closed curve Γ . For practical purpose, the 2D Dirac-delta function may be defined as follow:

$$\delta(x_1 - \xi_1, x_2 - \xi_2) = \lim_{\varepsilon \rightarrow 0^+} F_{\varepsilon}(x_1 - \xi_1, x_2 - \xi_2) \quad (1.7)$$

where $F_{\varepsilon}(x_1 - \xi_1, x_2 - \xi_2) = 1/(\pi \varepsilon^2)$ if $\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2} < \varepsilon$ otherwise $F_{\varepsilon}(x_1 - \xi_1, x_2 - \xi_2) = 0$. For a more rigorous treatment of the Dirac-delta function, one may refer to Hoskins [24].

The Dirac-delta function as defined in (1.7) above may be used to model a point source. If $f(x_1, x_2)$ is a finite continuous field function defined within the domain Ω , then

$$\int_{\Omega} \delta(x_1 - \xi_1, x_2 - \xi_2) f(x_1, x_2) d\Omega = \int_{\Omega} \delta(x_1 - \xi_1) \delta(x_2 - \xi_2) f(x_1, x_2) d\Omega = \lambda(\xi_1, \xi_2) f(\xi_1, \xi_2) \quad (1.8)$$

where $\lambda(\xi_1, \xi_2) = \omega(\xi_1, \xi_2)/(2\pi)$ with $\omega(\xi_1, \xi_2)$ being the internal angle at the point (ξ_1, ξ_2) , as depicted in Figure 1.5. $\lambda=1$ if $(\xi_1, \xi_2) \in \Omega$, $\lambda=1/2$ if (ξ_1, ξ_2) is on smooth part of Γ , $\lambda=0$ if (ξ_1, ξ_2) lies outside $\Omega \cup \Gamma$.

Figure 1.5: Definition of λ and internal angle ω .

1.3 Fundamental Theory of Heat Conduction in Solids

Conduction is one of the modes of heat transfer. It is the transfer of energy from the more energetic particles of the substance to the adjacent less energetic ones as a result of random interactions between the particles. Conduction can take place in solids, liquids, or gases. In solids, it is due to the combination of vibrations of the molecules in a lattice and the energy transport of free electrons.

The laws governing conduction in solids can be expressed in concise mathematical terms as described in this section. For further details on heat conduction, one may refer to Carslaw and Jaeger [13].

1.3.1 Law of Conservation of Heat Energy

Let $\mathbf{x}$ be any point (x_1, x_2, x_3) in Cartesian rectangular co-ordinate inside a body Ω bounded by surface Γ . We define

$C_p(\mathbf{x})$: specific heat capacity of the body, [J/kg K]

$\rho(\mathbf{x})$: mass density of solid, [kg/m³]

$c_{ij}(\mathbf{x})$: coefficient of thermal heat conductivity of the solid, [W/m K]

dm_s : infinitesimal mass at point $\mathbf{x}$ inside the solid, $dm_s = \rho(\mathbf{x})d\Omega$, [kg]

$T(\mathbf{x}, t)$: temperature at point $\mathbf{x}$ at time $t \geq 0$, [K]

$T_0(\mathbf{x}) = T(\mathbf{x}, 0)$: reference temperature or initial condition, [K]

$\dot{V}(\mathbf{x}, t)$: rate of internal energy generated per-unit volume, due to, for example, electrical resistance effects or exothermic chemical reactions, [W/m³]

Since the change of the amount of heat energy stored inside dm_s is proportional to $(T-T_o)$, we may write:

$$\text{'Total change of heat energy inside } dm_s \text{' } = C_p(T - T_o)dm_s.$$

Hence, it follows that

$$\text{'Total change of heat energy inside the body ' } = \int_{\Omega} \alpha(T - T_o)d\Omega,$$

where $\alpha = C_p\rho$. Differentiating with respect to time, we obtain:

$$\text{'Time rate change of heat energy inside the body' } = \frac{d}{dt} \int_{\Omega} \alpha(T - T_o)d\Omega = \int_{\Omega} \alpha \frac{\partial T}{\partial t} d\Omega$$

According to the law of conservation of heat energy, the time rate change of heat energy inside the body is equal to the sum of the heat energy coming into the body per-unit time through the surface Γ and the rate of energy generated by heat sources (if any) inside the region Ω , i.e.

$$\int_{\Omega} \alpha \frac{\partial T}{\partial t} d\Omega = - \oint_{\Gamma} \mathbf{h} \cdot \mathbf{n} d\Gamma + \int_{\Omega} \dot{V} d\Omega \quad (1.9)$$

where, according to the generalised classical Fourier law of heat conduction, $\mathbf{h}$ is the heat flux vector $[\text{W/m}^2]$ given by:

$$\mathbf{h}^T = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = - \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \partial T / \partial x_1 \\ \partial T / \partial x_2 \\ \partial T / \partial x_3 \end{bmatrix},$$

where c_{ij} are the coefficients for the thermal conductivity. Note that the above relation is valid for the most general anisotropic heat conduction. As discussed later, there are some physically motivated restrictions on these coefficients.

In index notation, the above relation can be written as

$$h_i = -c_{ij} \partial T / \partial x_j \text{ (sum over } j \text{)}. \quad (1.10)$$

Note that h_i is component of heat flux in x_i direction (i.e. rate of heat energy flow per-unit area in the direction of x_i). If q is the outwards normal heat flux through the boundary Γ (the rate of heat energy flow across the infinitesimal surface area in the direction of $\mathbf{n}$) then

$$q = \mathbf{h} \cdot \mathbf{n} = -n_i c_{ij} \frac{\partial T}{\partial x_j}, \text{ (sum over } i \text{ and } j \text{)}$$

Note that, if q is positive, then heat flows out from the body through Γ . Otherwise heat flows into the body.

Due to some consideration in physics, c_{ij} have the following properties:

- (1) $c_{ij} \geq 0$
- (2) Matrix $[c_{ij}]$ is symmetric, i.e. $c_{ij} = c_{ji}$. For isotropic heat conduction, $c_{ij} = 0$ if $i \neq j$ and $c_{11} = c_{22} = c_{33}$. For orthotropic heat conduction, $c_{ij} = 0$ if $i \neq j$ and c_{11} , c_{22} and c_{33} are possibly distinct positive coefficients.
- (3) $[c_{ij}]$ is a positive-definite matrix. For 2D cases, this results $c_{11}c_{22} - c_{12}^2 > 0$.
- (4) For homogeneous solids, $c_{ij} = \text{constant}$. For non-homogeneous solids, c_{ij} vary from point to point, i.e. $c_{ij} = c_{ij}(\mathbf{x})$.

One can apply Gauss's Theorem (1.2) to convert the surface integration in (1.9) to the corresponding volume integration, i.e.

$$\int_{\Omega} \alpha \frac{\partial T}{\partial t} d\Omega = - \int_{\Omega} \nabla \cdot \mathbf{h} d\Omega + \int_{\Omega} \dot{V} d\Omega$$

This gives rise to the energy equation:

$$-\nabla \cdot \mathbf{h} + \dot{V} = \alpha \frac{\partial T}{\partial t}. \quad (1.11)$$

If the Fourier law of heat conduction is valid, then by using (1.10), (1.11) becomes

$$\frac{\partial}{\partial x_i} (c_{ij} \frac{\partial T}{\partial x_j}) + \dot{V} = \alpha \frac{\partial T}{\partial t} \quad (\text{sum over } i \text{ and } j). \quad (1.12)$$

Several well-known governing PDEs could be obtained as special cases from (1.12). For example, for steady-state condition in homogeneous isotropic material, we find that T is independent of time, $c_{ij} = 0$ if $i \neq j$ and $c_{ij} = c$ if $i = j$ and hence we obtain:

$$\text{Poisson equation : } \nabla^2 T = \frac{\partial^2 T}{\partial x_i \partial x_i} = \frac{\dot{V}}{c}$$

Without internal energy generation, the Poisson reduces to:

$$\text{Laplace equation: } \nabla^2 T = 0 \quad (1.13)$$

For the steady-state heat conduction without internal energy generation in (1.12) in the most general anisotropic inhomogeneous solids, the governing equation is:

$$\text{Elliptic PDE: } \frac{\partial}{\partial x_i} (c_{ij} \frac{\partial T}{\partial x_j}) = 0. \quad (1.14)$$

For non-steady state heat conduction with $\dot{V} = 0$, (1.12) reads

$$\text{Parabolic PDE: } \frac{\partial}{\partial x_i} (c_{ij} \frac{\partial T}{\partial x_j}) = \alpha \frac{\partial T}{\partial t} \quad (1.15)$$

1.3.2 Heat Conduction as (Initial-)Boundary Value Problems

Figure 1.6: 2D domain Ω bounded by curves Γ_1 and Γ_2

Steady-state and non-steady state heat conduction problems can be posed as BVPs and initial-BVPs (IBVPs) respectively. To give a brief explanation on how steady-state conduction can be treated as BVPs, a Laplace equation will be used for the illustration. As for treating non-steady state conduction as IBVP, it will be explained in details in next two chapters, as it requires more complex mathematical descriptions.

Let us consider a homogeneous 2D domain Ω bounded by a single closed curve Γ , see Figure 1.6. The thermal field is governed by a 2D Laplace equation (1.13) subject to the boundary conditions (BCs):

$$\begin{aligned} \text{on } \Gamma_1: \quad T &= T_{\#} & (\text{Dirichlet condition}) \\ \text{on } \Gamma_2: \quad \frac{\partial T}{\partial n} &= \frac{\partial T}{\partial n}_{\#} & (\text{Neumann condition}) \end{aligned} \quad (1.16)$$

where $T_{\#}$ and $(\partial T / \partial n)_{\#}$ are the suitable prescribed functions. Γ_1 and Γ_2 are non-intersecting curves such that $\Gamma = \Gamma_1 \cup \Gamma_2$.

A BIE can be derived for (1.13) using the **Green's second identity** (1.6) (in 2D plane) together with an auxiliary function Φ . If the auxiliary function is carefully chosen as explained later, the domain integral which contains the potential function T of the actual problem can be eliminated.

Because of (1.13), (1.6) (with T replacing f) is reduced to

$$\int_{\Omega} T \nabla^2 \Phi \, d\Omega = \oint_{\Gamma} \left(T \frac{\partial \Phi}{\partial n} - \Phi \frac{\partial T}{\partial n} \right) d\Gamma \quad (1.17)$$

where $d\Omega=dx_1dx_2$.

If the auxiliary function Φ is chosen to satisfy

$$\nabla^2 \Phi = \delta(x_1 - \xi_1, x_2 - \xi_2) \quad (1.18)$$

where $\delta(x_1 - \xi_1, x_2 - \xi_2)$ is the 2D Dirac-delta function as defined in (1.7)-(1.8), then (1.17) reads

$$\lambda(\xi)T(\xi) = \oint_{\Gamma} \left[T(\mathbf{x}) \frac{\partial}{\partial n} \Phi(\mathbf{x}, \xi) - \Phi(\mathbf{x}, \xi) \frac{\partial}{\partial n} T(\mathbf{x}) \right] d\Gamma \quad (1.19)$$

where ξ is any point on $\Omega \cup \Gamma$; λ is as defined in (1.8). Note that through out this thesis $\mathbf{x}$ and ξ are (x_1, x_2) and (ξ_1, ξ_2) for 2D case, (x_1, x_2, x_3) and (ξ_1, ξ_2, ξ_3) for 3D case.

Equation (1.19) is known as the direct BIE of the 2D Laplace equation. Its significance or importance can be explained as follows. If both T and $\partial T / \partial n$ are known at all points on the exterior boundary Γ , then the right hand side of (1.19) is completely known and hence it can be used as a formula for calculating T at any point $\xi \in \Omega \cup \Gamma$. The problem is that according to (1.16) we know either T or $\partial T / \partial n$ (not both) on each and every point on the exterior boundary. Therefore, to overcome such problem, BEM is devised to help determine the unknown T and/or $\partial T / \partial n$ at the boundary points. Once they are all known, a formula for calculating $T(\xi)$ can be formed from (1.19). How the BEM can be used to solve BVPs and IBVPs of heat conduction will be fully illustrated in the subsequent three chapters when solving for problems that will be introduced in next section.

The auxiliary function Φ satisfying (1.18) given below is called the fundamental solution to the 2D Laplace equation:

$$\Phi(\mathbf{x}, \xi) = \frac{1}{2\pi} \ln(|\mathbf{x} - \xi|) \quad (1.20)$$

where $|\mathbf{x} - \xi|$ is the absolute distance between $\mathbf{x}$ and ξ . Note that

$$\frac{\partial}{\partial n} \Phi(\mathbf{x}, \xi) = \mathbf{n} \cdot \nabla \Phi = \frac{(x_1 - \xi_1)n_1(\mathbf{x}) - (x_2 - \xi_2)n_2(\mathbf{x})}{2\pi|\mathbf{x} - \xi|^2}, \quad (\forall \mathbf{x} \in \Gamma)$$

For 3D case, the corresponding fundamental solution is found to be $\Phi(\mathbf{x}, \xi) = 1/(4\pi|\mathbf{x} - \xi|)$. One may refer to [32] and [34] for a mathematical derivation of (1.20) and other fundamental solutions related to various differential operators.

1.4 Introduction of Work in Subsequent Chapters

Before going into the introduction of the subsequent work in each chapter, we shall first have a brief discussion on a common problem faced when using BEM in solving particular conduction cases, and how this can be overcome.

For conduction problem involving inhomogeneous media and/or non-steady state condition, deriving a BIE is not as easy and direct as we have seen in previous section. Since, in general, it is difficult to obtain a suitable analytical form to the fundamental solution of the PDE with spatially varying c_{ij} , or of the time-dependent PDE such as the one in (1.15). In the paper of Kassab and Divo [25], a generalised fundamental solution for heat conduction problem in inhomogeneous anisotropic media is derived. For most of the cases, however, it is difficult to determine analytically the explicit form of the fundamental solution and its derivative. In the paper of Paulino *et al* [31], a 3D Green's function for transient heat conduction in FGM is derived for only the case in which the thermal conductivity varies exponentially in one Cartesian co-ordinate.

If the fundamental solution for the homogeneous media (such as (1.20)) is used instead, the resulting integral formulation includes not only a boundary integral but also a domain integral. Traditionally the domain integral has been dealt with by discretising the domain into a series of internal cells and carrying out the domain integration of the inhomogeneous term over them. The need for the domain discretisation is an obvious drawback to an approach which is intended to be used in a boundary integral formulation. To treat the domain integral in more general situations the following methods have been frequently used: Dual-Reciprocity Method (DRM) [12], the Fourier Series Expansions [36] and the Multiple Reciprocity Method [28].

Due to its ease of implementation and strong flexibility of applying fundamental solution, the DRM technique stands out as the first choice in this report for solving the first two types of 2D heat conduction problems. The BEM together with the DRM, i.e. the DRBEM, is an approach proposed by Brebbia and Nardini [12] to solve some dynamic problems. It was extended later on to a wide range of different problems such as in stress analysis, heat transfer problems, potential problems and acoustics as described in the recent work by Partridge, Brebbia and Wrobel [30]. Some recent work in employing DRM for conduction problems are Tanaka *et al* [35], Kassab and Divo [25] and Ang *et al* [7] and etc.

The basic idea behind this approach is to employ a fundamental solution corresponding to a simpler equation to transform approximately the domain integrals resulting from

---

inhomogeneous terms in the PDE into boundary integrals. The procedure may involve a series expansion using either global approximating functions or local radial basis function and the application of reciprocity principles.

Now, we shall embark on a brief description of the work carried out in the present thesis.

Chapter 2

In a recent paper by Ang *et al* [7], the DRBEM is applied for solving a class of 2D BVPs governed by the elliptic PDE (1.14) involving non-homogeneous anisotropic media in which the coefficients c_{ij} take the form

$$c_{ij}(\mathbf{x}) = c_{ij}^{(o)} g(\mathbf{x}) \quad (1.21)$$

where g is a given positive function (grading) that can be partially differentiated at least twice with respect to x_i and $c_{ij}^{(o)}$ are non-negative constants satisfying $c_{ij}^{(o)} = c_{ji}^{(o)}$ and $c_{11}^{(o)} c_{22}^{(o)} - (c_{12}^{(o)})^2 > 0$.

The BVP governed by (1.14) has extensive applications in engineering problems involving both inhomogeneous isotropic and anisotropic media (such as FGM). In the paper by Ang *et al* [7], the grading function $g(\mathbf{x})$ can be general (so long as it is partially differentiable at least twice). In many other research papers, to simplify the mathematics involved, only exponential variation is assumed for the grading function (e.g. Paulino *et al* [31]).

In Chapter 2, the work published in [7] shall be extended to the case in which time-dependent state is considered, i.e. the parabolic PDE (1.15) is solved in place of (1.14). After using a substitution to re-write (1.15) in a suitable form, we employ the fundamental solution for the corresponding homogeneous anisotropic media, which takes the form of a simple logarithmic function, to derive an integro-differential formulation for the IBVP under consideration. With such a fundamental solution, the integro-differential formulation inevitably contains a domain integral over the 2D domain under consideration. To use the formulation for deriving a BEM for the numerical solution of the IBVP, we apply the DRM to convert the domain integral into a line integral approximately. The temperature and heat flux on the boundary are approximated using discontinuous linear elements (DLEs). To reduce the equations to a system of linear ordinary differential equations, each of the unknown functions giving the temperature at the nodal points at time t is approximated using a polynomial function in t over a small time interval. Once the linear ordinary differential equations are solved subject to a given initial condition, the temperature can be determined at any desired

point and time in the interior of the solution domain. Numerical results for two specific test problems are given.

Chapter 3

In the classical Fourier heat flux model, the heat flux is assumed to be instantaneously proportional to the temperature gradient of the thermal field. It leads to the physically undesirable conclusion that thermal waves travel at an infinite speed. Some researchers reported that thermal waves might possibly be observed in solids at normal room temperatures and they may be significant over a relatively long period of time. In addition, microscopic phenomena such as phonon scattering and phonon-electron interaction which the Fourier heat flux model fails to capture play an influential role in heat conduction in many micro-engineering applications, e.g. in dielectric films, insulators and semiconductors. Thus, heat flux models which can account for thermal waves and/or microscopic effects have become increasingly important in recent years (Ang [5]).

Dual-phase lag model proposed by Tzou [39] is one such heat flux model which has been successfully employed in the analyses of certain microscale thermal problems (e.g. Tzou [37], Tzou [38]). The thermal waves and microscopic effects are modelled by introducing small time delays required for the temperature gradient and the heat flux to be set up in the conduction process. Mathematically, the relation is expressed as follow:

$$h_i(\mathbf{x}, t + \tau_h) = -c_{ij} \frac{\partial}{\partial x_j} T(\mathbf{x}, t + \tau_T) \quad (1.22)$$

where c_{ij} considered here are positive constant thermal conductivity, τ_h and τ_T are small magnitude positive constants giving phase lags of the heat flux components and the temperature gradient respectively. Following Tzou [39], left and right hand sides of (1.22) can be expanded using Taylor-Maclaurin series. The second and higher order terms in τ_h and τ_T can be ignored to obtain the constitute equation relation:

$$h_i(\mathbf{x}, t) + \tau_h \frac{\partial}{\partial t} h_i(\mathbf{x}, t) = -c_{ij} \frac{\partial}{\partial x_j} \left(T(\mathbf{x}, t) + \tau_T \frac{\partial}{\partial t} T(\mathbf{x}, t) \right) \quad (1.23)$$

The use of (1.23) together with the heat equation (1.11) without internal energy generation gives rise to the governing PDE

$$c_{ij} \frac{\partial^2}{\partial x_i \partial x_j} \left(T + \tau_T \frac{\partial T}{\partial t} \right) = \alpha \left(\frac{\partial T}{\partial t} + \tau_h \frac{\partial^2 T}{\partial t^2} \right) \quad (1.24)$$

Note that, if $\tau_h = \tau_T = 0$, (1.22) and (1.24) reduce to (1.10) and (1.15) (with c_{ij} being constants) respectively, i.e. classical Fourier heat flux model is resumed.

There are other researchers who employ the FDM and FEM for the numerical solution of the PDE (1.24) for certain special cases. For example, Manzari and Manzari [27] in the case of $\tau_T = 0$; and Dai and Nassar [19] in the case that T depending only on x_1 and t . More recently, a DRBEM for solving numerically the PDE (1.24) is described by Ang [5]. In the paper, $c_{ij} = \delta_{ij}$ (i.e for isotropic case) and the Laplace transformation DRBEM (LT-DRBEM) is used. In Chapter 3, we will extend the work in Ang [5] to the case of anisotropic media. A different DRBEM approach based on a modified polynomial approximation to the time derivatives of the temperature will be proposed to solve numerically the PDE with suitable initial-boundary conditions. Numerical results of three different test problems are given.

Chapter 4

Figure 1.7: A bimaterial with a flat interface.

Another type of inhomogeneous solids which are of practical interest in engineering include those that are made up of dissimilar materials bonded together sharing common boundaries known as interfaces. An example is the bimaterial depicted in Figure 1.7. The regions Ω^+ and Ω^- may be occupied by materials with different properties. The dissimilar materials are bonded along the plane interface Γ^0 (which is part of the plane $x_2 = 0$).

In many studies, the interface between two different materials is assumed to be perfect, i.e. the materials are assumed to be so well bonded together that there is no gap at all between them along the interface, see e.g. Ang [4], Berger and Karageorghis [11], and Clements [17]. In reality, there is however no such thing as a perfect interface. At the microscopic level, all

surfaces are rough with highly irregular contours. Thus, microscopic gaps exist between materials along their interface, no matter how well they are bonded together.

For thermal problems, if an interface is perfect, the temperature and heat flux are continuous on the interface. However, for a microscopically imperfect interface, the temperature must be allowed to jump across the interface at all points. This leads to a non-classical interfacial condition.

The BEM can be applied to solve heat conduction for the bimaterial situation shown in Figure 1.7. For the perfect interface case, a suitable Green's function that can be used to obtain an integral equation which involves integration over only the exterior boundary $\Gamma^+ \cup \Gamma^-$ can be obtained (Berger and Karageorghis [11]). Such an approach does not require the interface to be discretised into elements and it leads to highly accurate computation of the temperature and heat flux at points near the interface. In a recent paper by Ang *et al* [9], for the problem of calculating the steady-state 2D temperature field in a thermally isotropic bimaterial which has a homogeneously imperfect interface, a special Green's function satisfying the appropriate interface condition had been successfully derived.

Therefore, it would be interesting to further extend the above mentioned Green's function approach to the imperfect interface case involving two anisotropic media. In Chapter 4, we seek to solve numerically the 2D heat conduction problem governed by the elliptic PDE (1.14) for two different material properties c_{ij}^+ for Ω^+ (i.e. $x_2 > 0$) and c_{ij}^- for Ω^- (i.e. $x_2 < 0$), where $c_{ij}^\pm$ are constants. The derivation of the required Green's function is presented. Two specific test problems are solved numerically using the BEM without discretising the imperfect interface Γ^0 .

Chapter 2 Non-steady Anisotropic Heat Conduction in Inhomogeneous Materials

This chapter is devoted to devising a DRBEM for numerically solving the non-steady 2D anisotropic inhomogeneous heat conduction governed by the parabolic PDE (1.15). The work presented here is also reported in Ang and Choo [10].

2.1 PDE Transformation

We are interested in solving the PDE (1.15)

$$\frac{\partial}{\partial x_i} (c_{ij} \frac{\partial T}{\partial x_j}) = \alpha \frac{\partial T}{\partial t}$$

in the region Ω subject to the initial-boundary conditions

$$\begin{aligned} \text{in } \Omega &: T(\mathbf{x}, 0) = T_0(\mathbf{x}) \\ \text{on } \Gamma_1 &: T(\mathbf{x}, t) = T_{\#}(\mathbf{x}, t) \\ \text{on } \Gamma_2 &: q(\mathbf{x}, t) = q_{\#}(\mathbf{x}, t) \end{aligned} \quad (2.1)$$

where $T_{\#}$ and $q_{\#}$ are suitably prescribed functions in such a way that initial condition $T_0(\mathbf{x})$ is compatible with the boundary conditions (BCs). The solution domain Ω is sketched in Figure 1.6.

If the coefficients c_{ij} take the form defined by (1.21), (1.15) can be re-written as

$$c_{ij}^{(o)} \frac{\partial^2 T}{\partial x_i \partial x_j} = \frac{1}{g} \left(\alpha \frac{\partial T}{\partial t} - c_{ij}^{(o)} \frac{\partial g}{\partial x_i} \frac{\partial T}{\partial x_j} \right). \quad (2.2)$$

The DRBEM together with a time-marching scheme for treating the time derivative will be employed to solve the PDE (2.2) subject to the initial-boundary condition defined in (2.1). As in Ang *et al* [7], in order to re-cast (2.2) in a form that allows the IBVP to be formulated in term of an integro-differential equation suitable for the development of DRBEM, the following substitution is used:

$$T(\mathbf{x}, t) = \frac{\varphi(\mathbf{x}, t)}{\sqrt{g(\mathbf{x})}}. \quad (2.3)$$

Thus (2.2) can be re-written as

$$c_{ij}^{(o)} \frac{\partial^2 \varphi}{\partial x_i \partial x_j} = W \frac{\varphi}{\sqrt{g}} + \frac{\alpha}{g} \frac{\partial \varphi}{\partial t} \quad (2.4)$$

where $W(\mathbf{x}) = c_{ij}^{(o)} \frac{\partial^2 \sqrt{g(\mathbf{x})}}{\partial x_i \partial x_j}$. Note that there is no partial derivative with respect to x_i in the right hand side of (2.4). The only derivative present there is the time derivative of φ .

2.2 Integro-Differential Equation

If $c_{ij}^{(o)}$ is multiplied to both sides of (1.5) with φ replacing f , and Einsteinian summation convention is assumed for the 2D case (i.e. the subscripts i and j running from 1 to 2), the following relation is obtained:

$$\int_{\Omega} \left(\Phi c_{ij}^{(o)} \frac{\partial^2 \varphi}{\partial x_i \partial x_j} - \varphi c_{ij}^{(o)} \frac{\partial^2 \Phi}{\partial x_i \partial x_j} \right) d\Omega = \oint_{\Gamma} \left(\Phi c_{ij}^{(o)} n_i \frac{\partial \varphi}{\partial x_j} - \varphi c_{ij}^{(o)} n_i \frac{\partial \Phi}{\partial x_j} \right) d\Gamma$$

If Φ (to be given shortly) is selected to satisfy

$$c_{ij}^{(o)} \frac{\partial^2 \Phi}{\partial x_i \partial x_j} = \delta(x_1 - \xi_1, x_2 - \xi_2),$$

then the above integral equation becomes:

$$\lambda(\xi) \varphi(\xi, t) = \int_{\Omega} \left(\Phi c_{ij}^{(o)} \frac{\partial^2 \varphi}{\partial x_i \partial x_j} \right) d\Omega - \oint_{\Gamma} \left(\Phi c_{ij}^{(o)} n_i \frac{\partial \varphi}{\partial x_j} - \varphi c_{ij}^{(o)} n_i \frac{\partial \Phi}{\partial x_j} \right) d\Gamma \quad (2.5)$$

The function Φ is the so-called fundamental solution to 2D anisotropic case (see Chang *et al* [15]) given by

$$\Phi(\mathbf{x}, \xi) = \frac{1}{2\pi \sqrt{c_{11}^{(o)} c_{22}^{(o)} - (c_{12}^{(o)})^2}} \text{Re}\{\ln(z - z_o)\}$$

$$\text{and hence } c_{ij}^{(o)} n_i \frac{\partial}{\partial x_j} \Phi(\mathbf{x}, \xi) = \frac{1}{2\pi \sqrt{c_{11}^{(o)} c_{22}^{(o)} - (c_{12}^{(o)})^2}} \text{Re} \left[\frac{(c_{11}^{(o)} + \kappa c_{12}^{(o)}) n_1 + (c_{21}^{(o)} + \kappa c_{22}^{(o)}) n_2}{z - z_o} \right]$$

where $z = x_1 + \kappa x_2$, $z_o = \xi_1 + \kappa \xi_2$ and $\kappa = [-c_{12}^{(o)} + i \sqrt{c_{11}^{(o)} c_{22}^{(o)} - (c_{12}^{(o)})^2}] / c_{22}^{(o)}$ ($i = \sqrt{-1}$). 'Re' indicates the real part of the complex term.

With the use of (2.3)-(2.4) and knowing that

$$c_{ij}^{(o)} n_i \frac{\partial \varphi}{\partial x_j} = c_{ij}^{(o)} n_i \frac{\partial}{\partial x_j} (\sqrt{g} T) = T G - \frac{q}{\sqrt{g}},$$

where $G = c_{ij}^{(o)} n_i \partial \sqrt{g} / \partial x_j$ and $q = -g c_{ij}^{(o)} n_i \partial T / \partial x_j$, we find that the integral equation (2.5) becomes:

$$\lambda(\xi)\sqrt{g(\xi)}T(\xi,t) = \int_{\Omega} \Phi \left(WT + \frac{\alpha}{\sqrt{g}} \frac{\partial T}{\partial t} \right) d\Omega + \oint_{\Gamma} \left(\frac{\Phi}{\sqrt{g}} q + (\phi\sqrt{g} - \Phi G) T \right) d\Gamma \quad (2.6)$$

where $\phi = c_{ij}^{(o)} n_i \partial \Phi / \partial x_j$.

As in other numerical techniques (e.g. FEM), BEM can be applied to deal with domains of arbitrary shapes. In BEM, only discretisation of boundary into piecewise smooth elements is required. We shall see how this can be done for our present problem in the following section.

2.3 BEM with Discontinuous Linear Elements

Figure 2.1: Linear functions over rectilinear elements and the mapping in natural co-ordinate system.

Let the boundary Γ be discretised into M straight-line boundary elements denoted by $\Gamma^{(m)}$ for $m=1,2,\dots,M$ (m denotes index of boundary element), in the anti-clockwise direction, i.e. we make the approximation:

$$\Gamma \approx \Gamma^{(1)} \cup \Gamma^{(2)} \cup \dots \cup \Gamma^{(M-1)} \cup \Gamma^{(M)}. \quad (2.7)$$

For a more accurate approximation of the function T and q along the elements, we use discontinuous linear elements (DLEs) in which $2M$ points are selected on Γ given by $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(K_I-1)}, \mathbf{x}^{(K_I)}$ where $K_I=2M$ is total number of boundary nodes. Every $\Gamma^{(m)}$ has two boundary nodes $\mathbf{x}^{(2m-1)}$ and $\mathbf{x}^{(2m)}$, each located at a fractional (l_1 and l_2) distance from the end of the element where l_1 and l_2 are between 0 and 0.5. Refer to Figure 2.1. When mapping each element m into intrinsic co-ordinate system u ($-1 < u < 1$), we may write the geometry representation of the element as

$$\mathbf{x}(u) = \tilde{\mathbf{x}}^{(m)} + \frac{L^{(m)}}{2} [-n_2^{(m)}, n_1^{(m)}] u, \quad (2.8)$$

where $\tilde{\mathbf{x}}^{(m)}$, $L^{(m)}$ and $[n_1^{(m)}, n_2^{(m)}]$ are the mid-point, length and outward unit normal vector of the m^{th} element respectively. With such a geometry representation, the infinitesimal length on boundary element is $d\Gamma = L^{(m)} du/2$. If f represents any function (e.g. GT , $T\sqrt{g}$ or $q/\sqrt{g}$) except Φ and ϕ along the elements, then f is assumed to be varying linearly along the elements, i.e.

$$f(u) = \frac{[1-2l_2-u, 1-2l_1+u]}{2(1-l_1-l_2)} \bullet [f^{(2m-1)}, f^{(2m)}], \quad (2.9)$$

where $f^{(2m-1)}$ and $f^{(2m)}$ are values at node $\mathbf{x}^{(2m-1)}$ and $\mathbf{x}^{(2m)}$ respectively.

Throughout this thesis, DLEs are used instead of another type of linear elements i.e. continuous linear elements (CLEs) in which boundary nodes are fixed at the adjacent of boundary elements (i.e. $l_1=l_2=0$). This is because the result accuracy of CLEs is normally much poorer than DLEs'. This may be due to the following reasons: (i) DLEs are more general, CLEs is just a special case of DLEs. In DLEs, the positions of boundary nodes (i.e. l_1 and l_2) are adjustable in order to obtain the optimum result, while CLEs lack of such advantage of flexibility. (ii) For the same number of M , the K_r in CLEs is equal to M , that is half of the one in DLEs. Hence the computed result is normally less accurate compared to DLEs. (iii) DLEs allow for discontinuity in boundary condition, especially the discontinuity of heat flux at sharp corner points on the boundary whereas CLEs fail to take this into consideration in the case of the Neumann condition. Furthermore, the formulations and hence the computer programming with CLEs are not as straight forward as in DLEs. In CLEs, we need to determine the individual internal angle ϖ (see Figure 1.5) at the boundary nodes, and to estimate the normal flux directions on the boundary nodes at sharp corner points.

For further details on other higher order approximations of the geometry and functions on boundary (e.g. parabolic elements) as well as the continuous elements, one may refer to París and Cañas [29].

With such DLEs using (2.8) and (2.9), (2.6) can now be approximated as

$$\begin{aligned} \lambda(\xi)\sqrt{g(\xi)}T(\xi,t) \approx \int_{\Omega} \Phi \left(WT + \frac{\alpha}{\sqrt{g}} \frac{\partial T}{\partial t} \right) d\Omega \\ + \sum_{k=1}^{K_r} \left[\left(\sqrt{g^{(k)}} I^{(k)}(\xi) - G^{(k)} II^{(k)}(\xi) \right) I^{(k)}(t) + \frac{1}{\sqrt{g^{(k)}}} II^{(k)}(\xi) q^{(k)}(t) \right] \end{aligned} \quad (2.10)$$

where k is index of nodal points. Here $^{(k)}$ indicates the term with such superscript takes the value corresponding to node $\mathbf{x}^{(k)}$. The integration parameters are defined as follows

$$I^{(k)}(\xi) = \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1-2l_{\hat{k}}-(-1)^{\hat{k}}u}{2(1-l_1-l_2)} \phi(\mathbf{x}(u), \xi) du = C^{(\bar{k})}(\xi) \int_{-1}^1 \frac{1-2l_{\hat{k}}-(-1)^{\hat{k}}u}{u^2 + A^{(\bar{k})}(\xi)u + B^{(\bar{k})}(\xi)} du$$

and

$$\begin{aligned} II^{(k)}(\xi) &= \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1-2l_{\hat{k}}-(-1)^{\hat{k}}u}{2(1-l_1-l_2)} \Phi(\mathbf{x}(u), \xi) du \\ &= 2(1-2l_{\hat{k}})D^{(\bar{k})} + E^{(\bar{k})} \int_{-1}^1 \left(1-2l_{\hat{k}}-(-1)^{\hat{k}}u\right) \ln(u^2 + A^{(\bar{k})}(\xi)u + B^{(\bar{k})}(\xi)) du \end{aligned}$$

where $\hat{k}$ and $\bar{k}$ are in such way that if $k=\text{odd}$, $\hat{k}=2$, $\bar{k}=(k+1)/2$ and, if $k=\text{even}$, $\hat{k}=1$, $\bar{k}=k/2$. In the above, $\bar{k}$ is the index of boundary elements replacing the index m used previously. Here $^{(\bar{k})}$ denotes the term carrying such superscript takes the value corresponding to element $\bar{k}$. The coefficients inside the parameters are given as

$$A^{(\bar{k})}(\xi) = \frac{4c_{22}^{(o)}}{L^{(\bar{k})}} \frac{(c_{11}^{(o)}n_1^{(\bar{k})} + c_{12}^{(o)}n_2^{(\bar{k})})(\tilde{x}_2^{(\bar{k})} - \xi_2) - (c_{12}^{(o)}n_1^{(\bar{k})} + c_{22}^{(o)}n_2^{(\bar{k})})(\tilde{x}_1^{(\bar{k})} - \xi_1)}{(c_{12}^{(o)}n_1^{(\bar{k})} + c_{22}^{(o)}n_2^{(\bar{k})})^2 + [c_{11}^{(o)}c_{22}^{(o)} - (c_{12}^{(o)})^2](n_1^{(\bar{k})})^2},$$

$$B^{(\bar{k})}(\xi) = \frac{4}{(L^{(\bar{k})})^2} \frac{[c_{22}^{(o)}(\tilde{x}_1^{(\bar{k})} - \xi_1) - c_{12}^{(o)}(\tilde{x}_2^{(\bar{k})} - \xi_2)]^2 + [c_{11}^{(o)}c_{22}^{(o)} - (c_{12}^{(o)})^2](\tilde{x}_2^{(\bar{k})} - \xi_2)^2}{(c_{12}^{(o)}n_1^{(\bar{k})} + c_{22}^{(o)}n_2^{(\bar{k})})^2 + [c_{11}^{(o)}c_{22}^{(o)} - (c_{12}^{(o)})^2](n_1^{(\bar{k})})^2} \geq 0,$$

$$C^{(\bar{k})}(\xi) = \frac{c_{22}^{(o)}\sqrt{c_{11}^{(o)}c_{22}^{(o)} - (c_{12}^{(o)})^2}}{2\pi(1-l_1-l_2)L^{(\bar{k})}} \frac{(\tilde{x}_1^{(\bar{k})} - \xi_1)n_1^{(\bar{k})} + (\tilde{x}_2^{(\bar{k})} - \xi_2)n_2^{(\bar{k})}}{(c_{12}^{(o)}n_1^{(\bar{k})} + c_{22}^{(o)}n_2^{(\bar{k})})^2 + [c_{11}^{(o)}c_{22}^{(o)} - (c_{12}^{(o)})^2](n_1^{(\bar{k})})^2},$$

$$D^{(\bar{k})} = E^{(\bar{k})} \ln \left[\left(\frac{L^{(\bar{k})}}{2c_{22}^{(o)}} \right)^2 \left((c_{12}^{(o)}n_1^{(\bar{k})} + c_{22}^{(o)}n_2^{(\bar{k})})^2 + [c_{11}^{(o)}c_{22}^{(o)} - (c_{12}^{(o)})^2](n_1^{(\bar{k})})^2 \right) \right],$$

$$E^{(\bar{k})} = \frac{L^{(\bar{k})}}{16\pi(1-l_1-l_2)\sqrt{c_{11}^{(o)}c_{22}^{(o)} - (c_{12}^{(o)})^2}} > 0.$$

It can be shown that $4B^{(\bar{k})} - (A^{(\bar{k})})^2 = [8\pi(1-l_1-l_2)C^{(\bar{k})}]^2 \geq 0$. Refer to Appendix A1 for the details on the integration parameters.

There is an undesirable feature in (2.10), i.e. the presence of the domain integral over Ω . In the traditional method, one has to discretise the whole solution domain Ω into small elements if (2.10) is used to find T . This destroys the advantage which the BEM has over other methods. However this can be overcome by using the DRM as described below.

2.4 Dual-Reciprocity Method

The dual-reciprocity method (DRM) requires collocating equations at points on Γ and the interior of Ω . For this purpose, we require another K_Ω well-spaced out points inside Ω denoted by $\mathbf{x}^{(K_\Gamma+1)}, \mathbf{x}^{(K_\Gamma+2)}, \dots, \mathbf{x}^{(K_\Gamma+K_\Omega)}$. To apply the DRM (which was originally proposed by Brebbia and Nardini [12]) to transform the domain integral in (2.10) into a line integral, we first make the approximation:

$$WT + \frac{\alpha}{\sqrt{g}} \frac{\partial T}{\partial t} \approx \sum_{l=1}^{K_\Gamma+K_\Omega} \mu^{(l)}(t) \chi^{(l)}(\mathbf{x}), \quad (2.11)$$

(for $l=1, 2, \dots, K_\Gamma+K_\Omega$)

where $\mu^{(l)}(t)$ are a set of initially unknown coefficients to be determined and the χ is an interpolating function which takes the form

$$\begin{aligned} \chi^{(l)}(\mathbf{x}) &= c_{ij}^{(o)} \frac{\partial^2}{\partial x_i \partial x_j} \theta^{(l)}(\mathbf{x}) \\ &= 1 + \left\{ \left[(x_1 - x_1^{(l)}) + \operatorname{Re}(\kappa)(x_2 - x_2^{(l)}) \right]^2 + \left[\operatorname{Im}(\kappa)(x_2 - x_2^{(l)}) \right]^2 \right\} \\ &\quad + \left\{ \left[(x_1 - x_1^{(l)}) + \operatorname{Re}(\kappa)(x_2 - x_2^{(l)}) \right]^2 + \left[\operatorname{Im}(\kappa)(x_2 - x_2^{(l)}) \right]^2 \right\}^{\frac{3}{2}} \end{aligned} \quad (2.12)$$

where

$$\begin{aligned} \theta^{(l)}(\mathbf{x}) &= \frac{c_{22}^{(o)}}{[c_{11}^{(o)} c_{22}^{(o)} - (c_{12}^{(o)})^2]} \left\{ \frac{1}{4} \left(\left[(x_1 - x_1^{(l)}) + \operatorname{Re}(\kappa)(x_2 - x_2^{(l)}) \right]^2 + \left[\operatorname{Im}(\kappa)(x_2 - x_2^{(l)}) \right]^2 \right) \right. \\ &\quad + \frac{1}{16} \left(\left[(x_1 - x_1^{(l)}) + \operatorname{Re}(\kappa)(x_2 - x_2^{(l)}) \right]^2 + \left[\operatorname{Im}(\kappa)(x_2 - x_2^{(l)}) \right]^2 \right)^2 \\ &\quad \left. + \frac{1}{25} \left(\left[(x_1 - x_1^{(l)}) + \operatorname{Re}(\kappa)(x_2 - x_2^{(l)}) \right]^2 + \left[\operatorname{Im}(\kappa)(x_2 - x_2^{(l)}) \right]^2 \right)^{\frac{5}{2}} \right\} \end{aligned}$$

Note that the function χ in (2.12) is used in Ang *et al* [7]. For the isotropic case, it reduces to those proposed by Zhang and Zhu [41]. This function can be chosen in other ways, for example, proposed by Golberg *et al* [22] (multiquadric interpolation), Chen *et al* [16] and Wu [40] (compactly supported radial basis functions).

If we let $\mathbf{x} = \mathbf{x}^{(k)}$ in (2.11) for $k=1, \dots, K_\Gamma+K_\Omega$, a system of linear ordinary differential equations in $\mu^{(l)}(t)$ can be set up and inverted to obtain

$$\mu^{(l)}(t) = \sum_{k=1}^{K_\Gamma+K_\Omega} \sigma_{lk} \left(W^{(k)} T^{(k)} + \frac{\alpha^{(k)}}{\sqrt{g^{(k)}}} \frac{dT^{(k)}}{dt} \right) \quad (2.13)$$

where $[\sigma_{lk}]$ is multiplicative inverse of $[\chi_{kl}]$ with $\chi_{kl} = \chi^{(l)}(\mathbf{x}^{(k)})$.

Approximation of (2.11) allows the transformation of domain integral in (2.10) into a line integral approximately when we apply (2.5) again with φ replaced by $\theta^{(l)}$, i.e.

$$\int_{\Omega} \Phi \left(WT + \frac{\alpha}{\sqrt{g}} \frac{\partial T}{\partial t} \right) d\Omega \approx \sum_{l=1}^{K_r+K_\Omega} \mu^{(l)}(t) \left\{ \lambda(\xi) \theta^{(l)}(\xi) + \oint_{\Gamma} (\Theta^{(l)} \Phi - \theta^{(l)} \phi) d\Gamma \right\} \quad (2.14)$$

where $\Theta^{(l)} = c_{ij}^{(o)} n_i \frac{\partial \theta^{(l)}}{\partial x_j}$.

Now, with equations (2.13) and (2.14), and applying (2.7), (2.10) becomes

$$\begin{aligned} \lambda(\xi) \sqrt{g(\xi)} T(\xi, t) \approx & \sum_{k=1}^{K_r+K_\Omega} \sum_{l=1}^{K_r+K_\Omega} \left(W^{(k)} T^{(k)}(t) + \frac{\alpha^{(k)}}{\sqrt{g^{(k)}}} \frac{dT^{(k)}(t)}{dt} \right) \Psi^{(l)}(\xi) \sigma_{lk} \\ & + \sum_{k=1}^{K_r} \left[\left(\sqrt{g^{(k)}} I^{(k)}(\xi) - G^{(k)} II^{(k)}(\xi) \right) I^{(k)}(t) + \frac{1}{\sqrt{g^{(k)}}} II^{(k)}(\xi) q^{(k)}(t) \right] \end{aligned} \quad (2.15)$$

where $\Psi^{(l)}(\xi) = \lambda(\xi) \theta^{(l)}(\xi) + \sum_{k=1}^{K_r} (\Theta^{(kl)} II^{(k)}(\xi) - \theta^{(kl)} I^{(k)}(\xi))$ with $\theta^{(kl)} = \theta^{(l)}(\mathbf{x}^{(k)})$ and

$\Theta^{(kl)} = \Theta^{(l)}(\mathbf{x}^{(k)})$.

2.5 Time-Marching (TM) Scheme with Polynomial Approximation

Note that time derivatives present in the discretised integro-differential equation (2.15). The approaches for treating the time-derivatives, as reported in the literature, can essentially be classified into two broad categories: (1) the transform space approach, and (2) the time domain approach.

In transform space approach, the time dependent derivative is eliminated to obtain a transform variable. However, once the differential system is solved in the transform space, reconstituting the solution in the time domain requires an inverse transform. Although this approach is simple and attractive, the accuracy depends upon an efficient and accurate numerical inverse transform. For diffusion problems, the Laplace transform (LT) seems to be the best choice. The first such formulation utilising the LT approach was proposed by Rizzo and Shippy [33] for the solution of heat conduction problems in solids. More recent work includes Ang [5] and Paulino [31].

An alternative to the transform space approach is to employ the time domain approach. With such an approach, a time-marching (or time-stepping) scheme associated with the BEM

solution at each time step is used. The solutions are found directly in the time domain. Examples of some early and more recent work using the time-stepping approach include those by Chang *et al* [15], Kassab and Divo [25], Ang *et al* [6], Ang [8] and other references therein.

In the present report, the time-marching scheme using polynomial function approximations is employed to deal with the time-dependent variables in the problem under consideration. The time domain is discretised into time elements (or time steps). Each element carries a number of time nodes. Each time node represents the state of the space domain at a time instant. Specifically, this is written as

$$T(\mathbf{x}, t) \approx \sum_{r=1}^{K_\tau} T(\mathbf{x}, t^{(r)}) P^{(r)}(t) \quad (2.16)$$

with the coefficient

$$P^{(r)}(t) = \prod_{a=1, a \neq r}^{K_\tau} \left(\frac{t - t^{(a)}}{t^{(r)} - t^{(a)}} \right), \quad (2.17)$$

where $t^{(r)}$ and K_τ denote time node and the total number of the time nodes for each time element, respectively. K_τ is a positive integer ≥ 2 , hence $(K_\tau - 1)\Delta t$ is the length of each time element and $K_\tau - 1$ is the order of the polynomial function in (2.16). $t^{(r)}$ (for $r=1, 2, \dots, K_\tau$) is defined as $t^{(r)} = t^{(1)} + (r-1)\Delta t$ with $t^{(1)}$ = 'initial time ≥ 0 '. Δt is a small positive time interval between the nodes. Note that in (2.17), $P^{(r)}(t^{(w)}) = 1$ if $w=r$ (for $w=1, 2, \dots, K_\tau$), otherwise $P^{(r)}(t^{(w)}) = 0$; and $t^{(1)} \leq t \leq t^{(K_\tau)}$. The corresponding approximation of $q(\mathbf{x}, t)$ can be obtained in the similar way as (2.16).

Differentiating (2.16) with respect to t , we obtain:

$$\frac{\partial}{\partial t} T(\mathbf{x}, t) \approx \sum_{r=1}^{K_\tau} T(\mathbf{x}, t^{(r)}) \dot{P}^{(r)}(t), \quad (2.18)$$

where

$$\dot{P}^{(r)}(t) = \frac{d}{dt} P^{(r)}(t) = \frac{\sum_{a=1, a \neq r}^{K_\tau} \prod_{b=1, b \neq r, b \neq a}^{K_\tau} (t - t^{(b)})}{\prod_{a=1, a \neq r}^{K_\tau} (t^{(r)} - t^{(a)})}.$$

With the substitution of (2.16) and (2.18) into (2.15), we finally arrive at the following simple form:

$$\lambda(\xi) \sqrt{g(\xi)} T(\xi, t) \approx \sum_{k=1}^{K_r + K_\Omega} \sum_{r=1}^{K_\tau} [T^{(k,r)} A_1^{(k,r)}(\xi, t) + q^{(k,r)} A_2^{(k,r)}(\xi, t)] \quad (2.19)$$

where $T^{(k,r)}=T^{(k)}(t^{(r)})$, $q^{(k,r)}=q^{(k)}(t^{(r)})$,

$$A_1^{(k,r)}(\xi,t)=\omega^{(k)}\left(\sqrt{g^{(k)}}I^{(k)}(\xi)-G^{(k)}II^{(k)}(\xi)\right)P^{(r)}(t) \\ +\left(W^{(k)}P^{(r)}(t)+\frac{\alpha^{(k)}}{\sqrt{g^{(k)}}}\dot{P}^{(r)}(t)\right)\sum_{l=1}^{K_I+K_\Omega}\sigma_{lk}\Psi^{(l)}(\xi),$$

$$A_2^{(k,r)}(\xi,t)=\omega^{(k)}\frac{1}{\sqrt{g^{(k)}}}II^{(k)}(\xi)P^{(r)}(t),$$

with $\omega^{(k)}=1$ if $k\leq K_I$, otherwise $\omega^{(k)}=0$. Therefore $A_2^{(k,r)}(\xi,t)=0$ for $k>K_I$. Note that the coefficients $A_n^{(k,r)}(\xi,t)$ ($n=1,2$) are known functions.

Equation (2.19) is ready to be used for setting up a system of algebraic equations for solving variables which are unknown at the boundary and internal nodes.

2.6 Final System of Equation

The right hand side of equation (2.19) contains $(2K_I+K_\Omega)K_\tau$ variables given by $2K_IK_\tau$ of $T^{(k,r)}$ and $q^{(k,r)}$ on Γ , and $K_\Omega K_\tau$ of $T^{(k,r)}$ inside Ω . According to initial-boundary conditions (2.1), $(2K_I+K_\Omega)$ of $T^{(k)}$ and $q^{(k)}$ are given at initial time $t^{(1)}$, and $K_I(K_\tau-1)$ of the variables on Γ at $t^{(r)}$ for $r\neq 1$ are known. Thus the other $(K_I+K_\Omega)(K_\tau-1)$ variables are unknowns. To solve the unknowns given by either $T^{(k,r)}$ or $q^{(k,r)}$ on Γ and $T^{(k,r)}$ inside Ω , we can collocate the point ξ at $\mathbf{x}^{(p)}$ (for $p=1,2,\dots,K_I+K_\Omega$) and let $t=t^{(s)}-\zeta\Delta t$ (for $s=2,3,\dots,K_\tau$) in (2.19). Note that ζ is a ratio indicating the position of t to be collocated, i.e. $0\leq\zeta<1$. As such, the discretised BIE (2.19) becomes

$$0=\sum_{k=1}^{K_I+K_\Omega}\sum_{r=1}^{1+K_\tau}\left[T^{(k,r)}\left(A_1^{(pk,sr)}-\delta_{pk}\lambda^{(p)}\sqrt{g^{(p)}}P^{(sr)}\right)+q^{(k,r)}A_2^{(pk,sr)}\right]$$

or

$$\sum_{k=1}^{K_I+K_\Omega}\sum_{r=1}^{K_\tau}\left[\gamma_1^{(k,r)}T^{(k,r)}\left(A_1^{(pk,sr)}-\delta_{pk}\lambda^{(p)}\sqrt{g^{(p)}}P^{(sr)}\right)+\gamma_2^{(k,r)}q^{(k,r)}A_2^{(pk,sr)}\right] \\ =-\sum_{k=1}^{K_I+K_\Omega}\sum_{r=2}^{K_\tau}\left[\gamma_2^{(k,r)}T^{(k,r)}\left(A_1^{(pk,sr)}-\delta_{pk}\lambda^{(p)}\sqrt{g^{(p)}}P^{(sr)}\right)+\gamma_1^{(k,r)}q^{(k,r)}A_2^{(pk,sr)}\right] \quad (2.20)$$

where $P^{(sr)}=P^{(r)}(t^{(s)}-\zeta\Delta t)$, $A_n^{(pk,sr)}=A_n^{(k,r)}(\mathbf{x}^{(p)},t^{(s)}-\zeta\Delta t)$ (for $n=1,2$), $\lambda^{(p)}=\lambda(\mathbf{x}^{(p)})$ and δ_{pk} is Kronecker-delta: $\delta_{pk}=1$ if $p=k$ otherwise $\delta_{pk}=0$. $\gamma_1^{(k,r)}=1$ if $r=1$ or $T^{(k)}(t)$ is known for $t\geq 0$,

otherwise $\gamma_1^{(k,r)} = 0$. $\gamma_2^{(k,r)} = 1$ if $r=1$ or $q^{(k)}(t)$ is known for $t \geq 0$, otherwise $\gamma_2^{(k,r)} = 0$. Note that the second summation on the right of (2.20) starts from $r=2$. It is necessary to clarify here that we can, of course, collocate t directly at the time node $t^{(s)}$ ($s=2,3,\dots,K_\tau$) in (2.19) i.e. $\zeta=0$. However, by introducing ζ , we will be able to observe how the later numerical results are affected by varying ζ , thus finding out possible optimum ζ .

From (2.20), a system of $(K_I + K_\Omega)(K_\tau - 1)$ algebraic equations involving $(K_I + K_\Omega)(K_\tau - 1)$ unknowns is generated:

$$\begin{matrix} \mathbf{b} = & \mathbf{a} & \times & \mathbf{U} \\ (K_I + K_\Omega)(K_\tau - 1) & (K_I + K_\Omega)(K_\tau - 1) & & (K_I + K_\Omega)(K_\tau - 1) \\ \times 1 & \times (K_I + K_\Omega)(K_\tau - 1) & & \times 1 \end{matrix} \quad (2.21)$$

where $\mathbf{b} = [b_{p_s}]$, b_{p_s} = 'left hand side of (2.20)',

$$\mathbf{a} = [a_{p_s k_r}], \quad a_{p_s k_r} = -\gamma_2^{(k,r)} \left(A_1^{(pk, sr)} - \delta_{pk} \lambda^{(p)} \sqrt{g^{(p)}} P^{(sr)} \right) - \gamma_1^{(k,r)} A_2^{(pk, sr)},$$

$$\mathbf{U} = [U_{k_r}], \quad U_{k_r} = 'k_r^{\text{th}} \text{ unknown}' = \gamma_2^{(k,r)} T^{(k,r)} + \gamma_1^{(k,r)} q^{(k,r)},$$

with $p_s = (p-1)(K_\tau - 1) + (s-1)$ and $k_r = (k-1)(K_\tau - 1) + (r-1)$ for p and k run from 1 to $K_I + K_\Omega$; s and r run from 2 to K_τ .

From initial condition, $T(\mathbf{x}, 0)$ and $q(\mathbf{x}, 0)$ are known. Thus, one may start at $t^{(1)} = 0$ to solve (2.21) for the unknowns either $T^{(k)}(t)$ or $q^{(k)}(t)$ at $t = (r-1)\Delta t$ for $r=2, \dots, K_\tau$. In this first solution step, $T^{(k)}(t^{(1)}) = T_0(\mathbf{x}^{(k)})$. Once the unknowns are determined, let $(K_\tau - 1)\Delta t$ be the initial time for the next step to solve for the unknowns at $t = (K_\tau + r - 2)\Delta t$ ($r=2, \dots, K_\tau$) and so on. Thus, one can find all $T^{(k,r)}$ and $q^{(k,r)}$ corresponding to each time step. Once all the unknowns are determined, $T(\xi, t)$ can be computed from (2.19) directly at any interior point in the solution domain. Note that, even though there exists ζ as mentioned earlier, it not necessary for us to determine $T^{(k)}(t)$ at $t = t^{(s)} - \zeta \Delta t$ at each solution step.

2.7 Specific Test Problems

We shall now apply DRBEM together with the polynomial time functions proposed above to solve some specific problems with known solutions. The numerical solutions are calculated using discretised equation (2.19).

Problem 2.1

Figure 2.2: Case considered in Problem 2.1 ($M=20$, $K_\Omega=11$).

Consider a 2D domain $\Omega: \{x_1 \geq 0, x_2 \geq 0, 1 \leq x_1^2 + x_2^2 \leq 4\}$, bounded by boundary Γ_1 , Γ_2 , Γ_3 and Γ_4 as shown in Figure 2.2. The PDE governing the heat conduction is given by

$$\begin{aligned} & \frac{\partial}{\partial x_1} \left\{ \left[\cos^2\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) + 3 \right] \left[2 \frac{\partial T}{\partial x_1} + \frac{\partial T}{\partial x_2} \right] \right\} + \frac{\partial}{\partial x_2} \left\{ \left[\cos^2\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) + 3 \right] \left[\frac{\partial T}{\partial x_1} + \frac{3}{2} \frac{\partial T}{\partial x_2} \right] \right\} \\ &= 3 \left[1 + \cos^2\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) \right] \frac{\partial T}{\partial t} \end{aligned} \quad (2.22)$$

which is to be solved subject to the initial-boundary conditions:

$$\text{In } \Omega: \quad T_0(\mathbf{x}) = x_1 - \frac{4}{3}x_2 + 3 + \sin\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right)$$

$$\text{on } \Gamma_1 \ (t > 0): \quad T(x_1, t) = x_1 + 3 + \sin\left(\frac{x_1}{2}\right) e^{-t}$$

$$\text{on } \Gamma_2 \ (t > 0): \quad q(\mathbf{x}, t) = \frac{1}{6} \left[3 + \cos^2\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) \right] \left[3x_2 - 2x_1 - (4x_1 + 3x_2) e^{-t} \cos\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) \right] \quad (2.23)$$

$$\text{on } \Gamma_3 \ (t > 0): \quad T(x_2, t) = -\frac{4}{3}x_2 + 3 + \sin\left(\frac{x_2}{3}\right) e^{-t}$$

$$\text{on } \Gamma_4 \ (t > 0): \quad q(\mathbf{x}, t) = \frac{1}{3} \left[3 + \cos^2\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) \right] \left[2x_1 - 3x_2 + (4x_1 + 3x_2) e^{-t} \cos\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) \right]$$

Note that for this particular IBVP, the material properties are given by

$$\frac{1}{2}c_{11} = c_{12} = \frac{2}{3}c_{22} = 3 + \cos^2\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right) \text{ and } \alpha(\mathbf{x}) = 3 + 3\cos^2\left(\frac{x_1+x_2}{2} + \frac{x_2}{3}\right).$$

Recall that $\alpha(\mathbf{x})$ is product of the mass density $\rho(\mathbf{x})$ and heat capacity $C_p(\mathbf{x})$ as defined in section 1.3

It is easy to verify that the IBVP defined by (2.22) and (2.23) has the exact solution

$$T(\mathbf{x}, t) = 3 + x_1 - \frac{4}{3}x_2 + \sin\left(\frac{x_1}{2} + \frac{x_2}{3}\right) e^{-t}.$$

To evaluate the effectiveness of the proposed method which uses polynomial time approximations, an interior point (1.5,1) is selected for computing at 15 different time level t with $K_\tau=2,3,4,5,7$ and 10 (i.e. 6 different orders of (2.16)). To evaluate the convergence, the numerical results are tabulated in Table 2.1 using $M=20$ (boundary elements) and $K_\Omega=11$ (interior collocation points) with $\Delta t=0.1$ initially. In Table 2.2, the values of M and K_Ω are then increased to 41 and 24 respectively. The norms of errors are also calculated for each column.

From Table 2.1 and Table 2.2, the computed results with $\zeta=0$ agree well with the exact one even though at higher t . The convergence of error is observed when the number of element M and K_Ω are increased. By comparing the norm of error, it is obvious that the cubic time element ($K_\tau=4$) delivers more accurate numerical values than the linear ($K_\tau=2$) and quadratic ($K_\tau=3$) ones. However, the results show little improvement as we increase K_τ further. The norm of error of the numerical results is defined as

$$\text{Norm of error} = \sqrt{\frac{1}{N} \sum_{n=1}^N (T_{\text{exact},n} - T_{\text{comput},n})^2}$$

where N is the total number of computed results. $N=15$ in Table 2.1 and Table 2.2.

It would be interesting to mention that, when $K_\tau=2$ and $\zeta=1/2$, the results obtained (in 3rd column) are exactly same as the one obtained by approximating $T^{(k)}(t)$ in (2.15) with the central difference formula, i.e.

$$T^{(k)}(t) \approx \frac{1}{2} \left[T^{(k)}\left(t + \frac{\Delta t}{2}\right) + T^{(k)}\left(t - \frac{\Delta t}{2}\right) \right] \text{ and } \frac{dT^{(k)}(t)}{dt} \approx \frac{1}{\Delta t} \left[T^{(k)}\left(t + \frac{\Delta t}{2}\right) - T^{(k)}\left(t - \frac{\Delta t}{2}\right) \right] \quad (2.24)$$

We may note that from 3rd column ($K_\tau=2$, $\zeta=1/2$) of Table 2.1 and Table 2.2, the errors ascend abruptly as t is large. This seems to suggest, unless Δt is correctly chosen, that central difference formula is only good for short-term response for the problem here.

Numerous different problems with different boundary conditions have been tested. In general, optimum results can be obtained with $\zeta=0$, $K_\tau \geq 4$ and $L_{\min}/10 < \Delta t < 0.2$, where $L_{\min}$ is the minimum element length. Error of the results may 'surge up' easily at large t when we used ζ far away from zero even though Δt is within the range mentioned above; while there is no

such problem when $\zeta=0$ unless we set Δt too small or too large. Larger Δt may delay the blowing up of error with non-zero ζ , but the results are normally poorer.

On the boundary Γ_2 , $q(\mathbf{x},t)$ is specified but $T(\mathbf{x},t)$ is unknown. On boundary Γ_3 , $T(\mathbf{x},t)$ is specified but $q(\mathbf{x},t)$ is unknown. In Figure 2.3, the numerical solution with $M=20$ and $K_{\mathcal{Q}}=11$ for T and q at each nodal point along Γ_2 and Γ_3 are plotted respectively at $t=4.5$. Figure 2.3 shows that the computed and exact values are in good agreement with each other. The error of $q^{(k)}(t)$ at node 25 and 30 near the sharp corners (0,2) and (0,1) could be further reduced by employing more and finer boundary elements near those points. More accurate $T^{(k)}(t)$ and $q^{(k)}(t)$ solved by (2.21) give better results at interior points.

It should be mentioned that the results of DLE presented here are obtained by setting $l_1=l_2=0.22$, i.e. the two collocation points $\mathbf{x}^{(2m-1)}$ and $\mathbf{x}^{(2m)}$ placed at 22% of the segment length from the end-points of each boundary element m . Numerous other test examples were investigated and it was found that the optimum values of l_1 and l_2 are problem-dependent, i.e. they are dependent on the type of BC and the geometry of the domain under consideration. In general, the optimum value is likely to occur in the interval 0.15~0.30. If l_n ($n=1,2$) is too close to the extreme values 0 or 1/2, i.e. two collocation points are too close to the end-point of each element or too close to each other, then the accuracy of the numerical solution is reduced.

Chapter 2 Non-steady Anisotropic Heat Conduction in Inhomogeneous Materials

Table 2.1: Exact and computed numerical results at interior point $\xi=(1.5,1)$ obtained with $M=20$, $K_\Omega=11$, $\Delta t=0.1$ (Problem 2.1)

t	Exact $T(\xi, t)$	Computed						
		$\zeta=1/2$, $K_{\bar{\tau}}=2$	$\zeta=0$					
			$K_{\bar{\tau}}=2$	$K_{\bar{\tau}}=3$	$K_{\bar{\tau}}=4$	$K_{\bar{\tau}}=5$	$K_{\bar{\tau}}=7$	$K_{\bar{\tau}}=10$
0.05	4.0071007	4.0092335	4.0069704	4.0078655	4.0079527	4.0079883	4.0080295	4.0080627
0.25	3.8547558	3.8574952	3.8619739	3.8568011	3.8566406	3.8566620	3.8566741	3.8566776
0.45	3.7300264	3.7330957	3.7409389	3.7325848	3.7324001	3.7323708	3.7323656	3.7323705
0.65	3.6279066	3.6312430	3.6405540	3.6306833	3.6304453	3.6304264	3.6304242	3.6304245
0.85	3.5442980	3.5480878	3.5574310	3.5471156	3.5468664	3.5468456	3.5468447	3.5468361
1.05	3.4758451	3.4806794	3.4886921	3.4786006	3.4783633	3.4783372	3.4783358	3.4783358
1.25	3.4198005	3.4270851	3.4319125	3.4224375	3.4222090	3.4221908	3.4221898	3.4221898
1.45	3.3739151	3.3868099	3.3850579	3.3764068	3.3761994	3.3761828	3.3761818	3.3761817
1.65	3.3363474	3.3618679	3.3464268	3.3386855	3.3385045	3.3384860	3.3384852	3.3384852
1.85	3.3055895	3.3592707	3.3146003	3.3077773	3.3076152	3.3076025	3.3076017	3.3076017
2.05	3.2804070	3.3966397	3.2883975	3.2824540	3.2823138	3.2823025	3.2823019	3.2823019
3.05	3.2085094	9.2961440	3.2127246	3.2100541	3.2099926	3.2099878	3.2099875	3.2099875
5.05	3.1723295	17259.07	3.1739727	3.1735501	3.1735407	3.1735400	3.1735399	3.1735399
7.05	3.1674330	48925184	3.1686641	3.1686041	3.1686028	3.1686026	3.1686026	3.1686026
9.05	3.1667704	1.387E+11	3.1679427	3.1679345	3.1679343	3.1679343	3.1679343	3.1679343
Norm of error =		3.582E+10	0.0089797	0.0021411	0.0019849	0.0019734	0.0019743	0.0019753

Table 2.2: Corresponding results obtained with $M=41$, $K_\Omega=24$, $\Delta t=0.1$ (Problem 2.1)

t	Exact $T(\xi, t)$	Computed						
		$\zeta=1/2$, $K_{\bar{\tau}}=2$	$\zeta=0$					
			$K_{\bar{\tau}}=2$	$K_{\bar{\tau}}=3$	$K_{\bar{\tau}}=4$	$K_{\bar{\tau}}=5$	$K_{\bar{\tau}}=7$	$K_{\bar{\tau}}=10$
0.05	4.0071007	4.0082046	4.0063060	4.0071022	4.0071448	4.0071532	4.0071626	4.0071705
0.25	3.8547558	3.8557965	3.8604598	3.8552010	3.8550755	3.8550700	3.8550712	3.8550719
0.45	3.7300264	3.7309710	3.7391348	3.7306745	3.7304843	3.7304543	3.7304522	3.7304531
0.65	3.6279066	3.6287581	3.6386268	3.6286544	3.6284110	3.6283912	3.6283897	3.6283895
0.85	3.5442980	3.5450650	3.5554818	3.5450813	3.5448273	3.5448058	3.5448047	3.5448031
1.05	3.4758451	3.4765369	3.4867817	3.4766251	3.4763840	3.4763571	3.4763557	3.4763557
1.25	3.4198005	3.4204253	3.4300755	3.4205549	3.4203233	3.4203046	3.4203036	3.4203036
1.45	3.3739151	3.3744790	3.3833117	3.3746321	3.3744224	3.3744055	3.3744043	3.3744043
1.65	3.3363474	3.3368537	3.3447778	3.3370220	3.3368394	3.3368206	3.3368198	3.3368197
1.85	3.3055895	3.3060379	3.3130478	3.3062211	3.3060579	3.3060450	3.3060442	3.3060442
2.05	3.2804070	3.2807921	3.2869367	3.2809975	3.2808565	3.2808451	3.2808445	3.2808445
3.05	3.2085094	3.2080487	3.2116030	3.2089484	3.2088871	3.2088823	3.2088820	3.2088820
5.05	3.1723295	3.0961413	3.1730858	3.1726684	3.1726591	3.1726584	3.1726583	3.1726583
7.05	3.1674330	-3.802985	3.1678156	3.1677565	3.1677552	3.1677551	3.1677551	3.1677551
9.05	3.1667704	-631.8694	3.1670998	3.1670917	3.1670915	3.1670915	3.1670915	3.1670915
Norm of error =		163.9755	0.0075248	0.0005888	0.0004314	0.0004172	0.0004165	0.0004166

Figure 2.3: Values at boundary nodes at $t=4.5$ with $M=20$, $K_{\bar{Q}}=11$, $K_{\bar{r}}=4$, $\Delta t=0.1$, $\zeta=0$. (Problem 2.1)

Problem 2.2

The same 2D domain in Figure 2.2 is again considered, but the governing PDE (1.15) is subject to Neumann condition i.e. only the heat flux is known on boundary:

$$\begin{aligned}
 \text{In } \Omega: T_0(\mathbf{x}) &= 1 + \frac{1}{2}x_1 - \frac{3}{10}x_2 + e^{-(x_1+2x_2)} + e^{-(x_1+x_2)} \\
 \text{On } \Gamma_1 (t \geq 0): q(x_1, t) &= -(5 + \frac{1}{10}e^{x_1} + \frac{1}{300}e^{-t}) \\
 \text{On } \Gamma_2 (t \geq 0): q(\mathbf{x}, t) &= \frac{1}{20}[30x_1 + 50x_2 + (x_2 - 2x_1)e^{(x_1+2x_2)} + (20x_1 + 30x_2)e^{(x_2-t)}] \\
 \text{On } \Gamma_3 (t \geq 0): q(x_2, t) &= -3 + \frac{1}{5}e^{2x_2} - 2e^{(x_2-t)} \\
 \text{On } \Gamma_4 (t \geq 0): q(\mathbf{x}, t) &= -\frac{1}{10}[30x_1 + 50x_2 + (x_2 - 2x_1)e^{(x_1+2x_2)} + (20x_1 + 30x_2)e^{(x_2-t)}]
 \end{aligned} \tag{2.25}$$

The thermal properties are given as $c_{11}=c_{12}=c_{22}/2=e^{(x_1+2x_2)}$ and $\alpha(\mathbf{x})=3e^{(x_1+2x_2)}$. With these properties, an exact temperature field satisfying the PDE and initial-boundary conditions (2.25) is found to be

$$T(\mathbf{x}, t) = 1 + \frac{1}{2}x_1 - \frac{3}{10}x_2 + e^{-(x_1+2x_2)} + e^{-(x_1+x_2+t)}$$

To assess the accuracy of the proposed method, four interior points are chosen for computing with $K_\tau=4$, $\Delta t=0.1$ and $\zeta=0$. Table 2.3 shows the exact value and Table 2.4 shows the corresponding computed results with $M=20$, $K_\Omega=11$ and $M=41$, $K_\Omega=24$. The average percentage errors are also computed with the equation

$$\text{Avg. \% error} = \frac{1}{N} \sum_{n=1}^N \left| 1 - \frac{T_{\text{comput},n}}{T_{\text{exact},n}} \right| \times 100\%,$$

where $N=80$ for present case under consideration.

By comparing Table 2.3 and Table 2.4, it is obvious that the BEM computed results at four interior points remain very good for the current case of Neumann boundary condition. The average percentage error is reduced significantly as number of boundary elements and internal nodal points are increased from $M=20$, $K_\Omega=11$ to $M=41$, $K_\Omega=24$. Note that, the computed results are slightly diverging away from the exact ones as t increases i.e. errors at larger t are relative higher than those at lower t ; however the gaps can be narrowed down by increasing the M and K_Ω . Figure 2.4 shows the plot of exact and computed results of T at the nodes around the boundary at $t=2$. Excellent agreement is again observed in the plot.

The DRBEM using DLEs together with polynomial time function for solving the above two test problems is coded with the help of MATLAB. The code is listed in Appendix B1

Chapter 2 Non-steady Anisotropic Heat Conduction in Inhomogeneous Materials

Table 2.3: Exact $T(\xi, t)$ at four interior points. (Problem 2.2)

t	$\xi^{(1)} = (1.4266, 0.4635)$	$\xi^{(2)} = (1.2135, 0.8817)$	$\xi^{(3)} = (0.8817, 1.2135)$	$\xi^{(4)} = (0.4635, 1.4266)$
0.03	1.815849	1.512620	1.232753	0.986651
0.23	1.789276	1.490974	1.211107	0.960078
0.43	1.767521	1.473253	1.193386	0.938323
0.63	1.749709	1.458744	1.178877	0.920511
0.83	1.735125	1.446865	1.166998	0.905927
1.03	1.723186	1.437139	1.157272	0.893988
1.23	1.713410	1.429176	1.149309	0.884212
1.43	1.705407	1.422657	1.142790	0.876209
1.63	1.698854	1.417319	1.137452	0.869656
1.83	1.693489	1.412949	1.133082	0.864291
2.03	1.689097	1.409371	1.129504	0.859899
2.53	1.681291	1.403013	1.123146	0.852093
3.03	1.676556	1.399156	1.119289	0.847358
3.53	1.673684	1.396817	1.116950	0.844486
4.03	1.671943	1.395398	1.115531	0.842745
4.53	1.670886	1.394537	1.114670	0.841688
5.03	1.670246	1.394015	1.114148	0.841048
6.03	1.669621	1.393507	1.113640	0.840423
7.03	1.669391	1.393320	1.113453	0.840194
8.03	1.669307	1.393251	1.113384	0.840109

Table 2.4: Results computed with $K_{\bar{r}}=4$, $\Delta t=0.1$ and $\zeta=0$. (Problem 2.2)

T	$M=20, K_{\bar{Q}}=11$				$M=41, K_{\bar{Q}}=24$			
	$\xi^{(1)}$	$\xi^{(2)}$	$\xi^{(3)}$	$\xi^{(4)}$	$\xi^{(1)}$	$\xi^{(2)}$	$\xi^{(3)}$	$\xi^{(4)}$
0.03	1.8165446	1.5130631	1.2324873	0.9863252	1.8159461	1.5127203	1.2328439	0.9867168
0.23	1.7879352	1.4900445	1.2096554	0.9582167	1.7887440	1.4906212	1.2108632	0.9598046
0.43	1.7641770	1.4710714	1.1909535	0.9352158	1.7663310	1.4724569	1.1928097	0.9377051
0.63	1.7443622	1.4553008	1.1754039	0.9159998	1.7478307	1.4574742	1.1779300	0.9194882
0.83	1.7276271	1.4420550	1.1624163	0.8999472	1.7325802	1.4451480	1.1657058	0.9045178
1.03	1.7136486	1.4309512	1.1515247	0.8864984	1.7199770	1.4349538	1.1556060	0.8921674
1.23	1.7018877	1.4215650	1.1423214	0.8751810	1.7095406	1.4264989	1.1472378	0.8819587
1.43	1.6919172	1.4135856	1.1345172	0.8656451	1.7009052	1.4195022	1.1403228	0.8735392
1.63	1.6834605	1.4067649	1.1278456	0.8575545	1.6937304	1.4136754	1.1345708	0.8665605
1.83	1.6762314	1.4008830	1.1220935	0.8506439	1.6877502	1.4088037	1.1297676	0.8607584
2.03	1.6700084	1.3957804	1.1171098	0.8447181	1.6827626	1.4047332	1.1257611	0.8559384
2.53	1.6577640	1.3855494	1.1071196	0.8330778	1.6735056	1.3971252	1.1182918	0.8470436
3.03	1.6487349	1.3777705	1.0995255	0.8245176	1.6673697	1.3920139	1.1132947	0.8412064
3.53	1.6416759	1.3714930	1.0933940	0.8178331	1.6631393	1.3884266	1.1098028	0.8372260
4.03	1.6358254	1.3661329	1.0881512	0.8122852	1.6600694	1.3857641	1.1072200	0.8343656
4.53	1.6307214	1.3613386	1.0834527	0.8074294	1.6577095	1.3836660	1.1051890	0.8321822
5.03	1.6260833	1.3568977	1.0790911	0.8029977	1.6557864	1.3819144	1.1034942	0.8304088
6.03	1.6175824	1.3486036	1.0709194	0.7948159	1.6526455	1.3789664	1.1006352	0.8275029
7.03	1.6096032	1.3407081	1.0631120	0.7870669	1.6499486	1.3763643	1.0980977	0.8249767
8.03	1.6018734	1.3330077	1.0554765	0.7795062	1.6474333	1.3739008	1.0956813	0.8225860
Avg. % error =	1.7356				0.52664			

Chapter 2 Non-steady Anisotropic Heat Conduction in Inhomogeneous Materials

Figure 2.4: Exact and computed values at boundary nodes at $t=2$ with $M=41$, $K_{\Omega}=24$, $K_{\Gamma}=4$, $\Delta t=0.1$ and $\zeta=0$.
(Problem 2.2)

Chapter 3 Micro-Scale Heat Conduction in Homogeneous Anisotropic Materials

In this chapter, the detailed steps of numerical solution to the 2D thermal problem governed by the 3rd order PDE (1.24) derived from a non-Fourier heat flux model is presented. Please refer back to section 1.4 for the introduction of this chapter.

3.1 Integro-Differential Equation

The PDE (1.24)

$$c_{ij} \frac{\partial^2}{\partial x_i \partial x_j} \left(T + \tau_T \frac{\partial T}{\partial t} \right) = \alpha \left(\frac{\partial T}{\partial t} + \tau_h \frac{\partial^2 T}{\partial t^2} \right)$$

is subject to the following initial-boundary conditions

$$\begin{aligned} \text{in } \Omega : \quad & T(\mathbf{x}, 0) = T_0(\mathbf{x}) \\ & : \quad \frac{\partial T}{\partial t} \Big|_{t=0} = \dot{T}(\mathbf{x}, 0) \\ \text{on } \Gamma_1 : \quad & T(\mathbf{x}, t) = T_{\#}(\mathbf{x}, t) \\ \text{on } \Gamma_2 : \quad & q(\mathbf{x}, t) = q_{\#}(\mathbf{x}, t) \end{aligned} \tag{3.1}$$

where $T_0(\mathbf{x})$ and $\dot{T}(\mathbf{x}, 0)$ are suitable given functions. The solution domain Ω is as sketched in Figure 1.6.

In Chapter 2, we have seen that how the governing PDE under consideration can be transformed initially into an integro-differential equation which still contains a domain integral. In this chapter, a similar process is applied to the PDE (1.24). By using reciprocal theorem (2.5) with the constant c_{ij} now replacing $c_{ij}^{(o)}$, we obtain the corresponding integro-differential equation,

$$\lambda(\xi) \left[T(\xi, t) + \tau_T \frac{d}{dt} T(\xi, t) \right] = \int_{\Omega} \alpha \Phi \left(\frac{\partial T}{\partial t} + \tau_h \frac{\partial^2 T}{\partial t^2} \right) d\Omega + \oint_{\Gamma} \left([q + \tau_T \frac{\partial q}{\partial t}] \Phi + [T + \tau_T \frac{\partial T}{\partial t}] \phi \right) d\Gamma \tag{3.2}$$

where Φ and q are respectively fundamental solution for anisotropic case and normal heat flux on Γ as defined in Chapter 2 with c_{ij} replacing $c_{ij}^{(o)}$.

In the following section, BEM together with DRM, i.e. DRBEM, using DLEs is employed to fully transform (3.2) into a discretised BIE.

3.2 Dual-Reciprocity Boundary Element Method (DRBEM)

As we have seen in Chapter 2, the DRBEM requires collocating equation at points on Γ and interior of Ω . For this purpose, the boundary Γ is firstly discretised into M straight-line boundary elements denoted by $\Gamma^{(m)}$ for $m=1,2,\dots,M$, in the anti-clockwise direction to obtain the approximation in (2.7). Subsequently, K_Γ points given by $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(K_\Gamma-1)}, \mathbf{x}^{(K_\Gamma)}$ on Γ , and another K_Ω well-spaced out points inside Ω denoted by $\mathbf{x}^{(K_\Gamma+1)}, \mathbf{x}^{(K_\Gamma+2)}, \dots, \mathbf{x}^{(K_\Gamma+K_\Omega)}$ are selected. As before, for the DLEs, each element $\Gamma^{(m)}$ has two boundary nodes and the functions are assumed to have a linear evolution along each boundary element.

For (3.2), we make the approximation

$$\alpha \left(\frac{\partial T}{\partial t} + \tau_h \frac{\partial^2 T}{\partial t^2} \right) \approx \sum_{l=1}^{K_\Gamma+K_\Omega} \mu^{(l)}(t) \chi^{(l)}(\mathbf{x}), \quad (3.3)$$

(for $l=1, 2, \dots, K_\Gamma+K_\Omega$)

and let $\mathbf{x}=\mathbf{x}^{(k)}$ in (3.3) for $k=1, \dots, K_\Gamma+K_\Omega$, a system of linear algebraic equations in $\mu^{(l)}(t)$ can be set up and inverted to obtain

$$\mu^{(l)}(t) = \sum_{k=1}^{K_\Gamma+K_\Omega} \sigma_{lk} \alpha \left(\frac{dT^{(k)}}{dt} + \tau_h \frac{d^2 T^{(k)}}{dt^2} \right) \quad (3.4)$$

Thus the domain integral in (3.2) can be transformed approximately into a line integral as follows:

$$\int_{\Omega} \alpha \Phi \left(\frac{\partial T}{\partial t} + \tau_h \frac{\partial^2 T}{\partial t^2} \right) d\Omega \approx \sum_{l=1}^{K_\Gamma+K_\Omega} \mu^{(l)}(t) \Psi^{(l)}(\xi) \quad (3.5)$$

Therefore, with (2.7) together with (3.4) and (3.5), (3.2) becomes

$$\begin{aligned} \lambda(\xi) [T(\xi, t) + \tau_T \frac{d}{dt} T(\xi, t)] &\approx \sum_{k=1}^{K_\Gamma+K_\Omega} \sum_{l=1}^{K_\Gamma+K_\Omega} \alpha \left(\frac{dT^{(k)}}{dt} + \tau_h \frac{d^2 T^{(k)}}{dt^2} \right) \sigma_{lk} \Psi^{(l)}(\xi) \\ &+ \sum_{k=1}^{K_\Gamma} \left([T^{(k)} + \tau_T \frac{dT^{(k)}}{dt}] I^{(k)}(\xi) + [q^{(k)} + \tau_T \frac{dq^{(k)}}{dt}] II^{(k)}(\xi) \right) \end{aligned} \quad (3.6)$$

The terms $\chi^{(l)}(\mathbf{x})$, σ_{lk} , $\Psi^{(l)}(\xi)$, $I^{(k)}$ and $II^{(k)}$ in (3.3)–(3.6) are same as defined in Chapter 2 with constants c_{ij} replacing $c_{ij}^{(o)}$.

3.3 Time-Marching Scheme with Modified Polynomial Approximation

To apply the polynomial time functions in the time-marching scheme to approximate the time dependent variables in (3.6) more accurately, we modify (2.16) and (2.17) to cater for the initial conditions in (3.1). After some mathematical manipulations, which will be explained later, with $\partial T/\partial t|_{t=0}$ taken into consideration, we obtain

$$T(\mathbf{x}, t) \approx \frac{\dot{T}(\mathbf{x}, t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} P^{(1+K_\tau)}(t) + \sum_{r=1}^{K_\tau} T(\mathbf{x}, t^{(r)}) \left[P^{(r)}(t) - \frac{\dot{P}^{(r)}(t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} P^{(1+K_\tau)}(t) \right], \quad (3.7)$$

where

$$P^{(r)}(t) = \prod_{a=1, a \neq r}^{1+K_\tau} \left(\frac{t-t^{(a)}}{t^{(r)}-t^{(a)}} \right) \quad (r=1, 2, \dots, 1+K_\tau).$$

and the upper dot represents time derivative. Note that the upper limit of the product in $P^{(r)}(t)$ is now $1+K_\tau$. The term K_τ , which is the total number (an integer ≥ 2) of time nodes in each time element, is now also the order of the polynomial function. Differentiating (3.7) once and twice, we can obtain the corresponding approximations for $\partial T/\partial t$ and $\partial^2 T/\partial t^2$ as

$$\dot{P}^{(r)}(t) = \frac{d}{dt} P^{(r)}(t) = \frac{\sum_{a=1, a \neq r}^{1+K_\tau} \prod_{b=1, b \neq r, b \neq a}^{1+K_\tau} (t-t^{(b)})}{\prod_{a=1, a \neq r}^{1+K_\tau} (t^{(r)}-t^{(a)})} \quad \text{and} \quad \ddot{P}^{(r)}(t) = \frac{d^2}{dt^2} P^{(r)}(t) = \frac{\sum_{a=1, a \neq r}^{1+K_\tau} \sum_{b=1, b \neq r, b \neq a}^{1+K_\tau} \prod_{c=1, c \neq r, c \neq a, c \neq b}^{1+K_\tau} (t-t^{(c)})}{\prod_{a=1, a \neq r}^{1+K_\tau} (t^{(r)}-t^{(a)})}.$$

The corresponding approximation of $q(\mathbf{x}, t)$ can be written in a similar way as (3.7).

Equation (3.7) has taken into account the initial condition of $\partial T/\partial t$. To do this, an extra time node is added to the time element which already has K_τ nodes. This extra node may be placed inside or outside the time element, but the value at this instant time does not have to be determined. For convenience, in the present chapter, we place the node outside the time element. The node is denoted by $t^{(1+K_\tau)}$ i.e. $t^{(K_\tau)} + \Delta t$. Equation (2.16) can then be modified to give

$$T(\mathbf{x}, t) \approx T(\mathbf{x}, t^{(1+K_\tau)}) P^{(1+K_\tau)}(t) + \sum_{r=1}^{K_\tau} T(\mathbf{x}, t^{(r)}) P^{(r)}(t) \quad (3.8)$$

where $P^{(r)}(t)$ is the one defined above in this section. It follows that we could derive $T(\mathbf{x}, t^{(1+K_\tau)})$ in term of $\dot{T}(\mathbf{x}, t^{(1)})$ from the first time derivative of (3.8) as

$$T(\mathbf{x}, t^{(1+K_\tau)}) = \frac{1}{\dot{P}^{(1+K_\tau)}(t^{(1)})} \left[\dot{T}(\mathbf{x}, t^{(1)}) - \sum_{r=1}^{K_\tau} T(\mathbf{x}, t^{(r)}) \dot{P}^{(r)}(t^{(1)}) \right].$$

This is then substituted back into (3.8) to obtain (3.7).

By substituting (3.7) and the corresponding time derivatives into (3.6) and rearranging it, one finally obtain the discretised BIE in the form

$$\lambda(\xi) \sum_{w=1}^{K_\tau} T(\xi, t^{(w)}) P_T^{(w)}(t) \approx A_0(\xi, t) + \sum_{k=1}^{K_\Gamma+K_\Omega} \sum_{r=1}^{K_\tau} [T^{(k,r)} A_1^{(k,r)}(\xi, t) + q^{(k,r)} A_2^{(k,r)}(\xi, t)] \quad (3.9)$$

where

$$\begin{aligned} P_T^{(w)}(t) &= P^{(w)}(t) + \tau_T \dot{P}^{(w)}(t) - \frac{\dot{P}^{(w)}(t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} [P^{(1+K_\tau)}(t) + \tau_T \dot{P}^{(1+K_\tau)}(t)], \\ A_0(\xi, t) &= -\lambda(\xi) \frac{\dot{T}(\xi, t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} [P^{(1+K_\tau)}(t) + \tau_T \dot{P}^{(1+K_\tau)}(t)] \\ &\quad + \sum_{k=1}^{K_\Gamma+K_\Omega} \left\{ \alpha \frac{\dot{T}(\mathbf{x}^{(k)}, t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} [\dot{P}^{(1+K_\tau)}(t) + \tau_h \ddot{P}^{(1+K_\tau)}(t)] \sum_{l=1}^{K_\Gamma+K_\Omega} \sigma_{lk} \Psi^{(l)}(\xi) \right. \\ &\quad \left. + \omega^{(k)} \left(\frac{\dot{T}(\mathbf{x}^{(k)}, t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} I^{(k)}(\xi) + \frac{\dot{q}(\mathbf{x}^{(k)}, t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} II^{(k)}(\xi) \right) [P^{(1+K_\tau)}(t) + \tau_T \dot{P}^{(1+K_\tau)}(t)] \right\} \\ A_1^{(k,r)}(\xi, t) &= \omega^{(k)} P_T^{(r)}(t) I^{(k)}(\xi) + \alpha \left(\dot{P}^{(r)}(t) + \tau_h \ddot{P}^{(r)}(t) - \frac{\dot{P}^{(r)}(t^{(1)})}{\dot{P}^{(1+K_\tau)}(t^{(1)})} [\dot{P}^{(1+K_\tau)}(t) + \tau_h \ddot{P}^{(1+K_\tau)}(t)] \right) \sum_{l=1}^{K_\Gamma+K_\Omega} \sigma_{lk} \Psi^{(l)}(\xi) \end{aligned}$$

and $A_2^{(k,r)}(\xi, t) = \omega^{(k)} P_T^{(r)}(t) II^{(k)}(\xi)$ with $T^{(k,r)}$ and $q^{(k,r)}$ as defined in Chapter 2.

3.4 Final System of Equations

To solve for the unknowns given by either $T^{(k,r)}$ or $q^{(k,r)}$ on Γ and $T^{(k,r)}$ inside Ω for each solution step, we can collocate ξ at $\mathbf{x}^{(p)}$ for $p=1, 2, \dots, K_\Gamma+K_\Omega$ and let $t=t^{(s)}-\zeta\Delta t$ for $s=2, 3, \dots, K_\tau$ to set up a system of linear algebraic equations:

$$\begin{aligned} A_0^{(p,s)} + \sum_{k=1}^{K_\Gamma+K_\Omega} \sum_{r=1}^{1+K_\tau} [\gamma_1^{(k,r)} T^{(k,r)} (A_1^{(pk, sr)} - \delta_{pk} \lambda^{(p)} P_T^{(sr)}) + \gamma_2^{(k,r)} q^{(k,r)} A_2^{(pk, sr)}] \\ \approx - \sum_{k=1}^{K_\Gamma+K_\Omega} \sum_{r=2}^{1+K_\tau} [\gamma_2^{(k,r)} T^{(k,r)} (A_1^{(pk, sr)} - \delta_{pk} \lambda^{(p)} P_T^{(sr)}) + \gamma_1^{(k,r)} q^{(k,r)} A_2^{(pk, sr)}] \end{aligned} \quad (3.10)$$

where $A_0^{(p,s)} = A_0(\mathbf{x}^{(p)}, t^{(s)} - \zeta\Delta t)$, $P_T^{(sr)} = P_T^{(r)}(t^{(s)} - \zeta\Delta t)$, and $\gamma_n^{(k,r)}$ ($n=1, 2$) are same as defined in Chapter 2. In the present case, the terms in (2.21) are as follows: b_{p_s} = 'left hand side of the (3.10)', $a_{p_s k_r} = -\gamma_2^{(k,r)} (A_1^{(pk, sr)} - \delta_{pk} \lambda^{(p)} P_T^{(sr)}) - \gamma_1^{(k,r)} A_2^{(pk, sr)}$, and U_{k_r} is same as defined in (2.21).

Once all the above mentioned unknowns are determined, we may proceed to find $T(\xi, t^{(w)})$ in (3.9) for $w=2,3,\dots,K_\tau$, because $T(\xi, t)$ for $\xi \in \Omega \cup \Gamma$ at any $t > 0$ can not be computed immediately from (3.9). Subsequently, $T(\xi, t)$ can be found directly from (3.7). To find $T(\xi, t^{(w)})$, we collocate (3.9) at $t=t^{(v)}-\zeta\Delta t$ for $v=2,3,\dots,K_\tau$ to set up another $K_\tau-1$ linear algebraic system and invert to obtain

$$\lambda(\xi)T(\xi, t^{(w)}) \approx \sum_{v=2}^{K_\tau} X_{wv} \left(A_0(\xi, t^{(v)} - \zeta\Delta t) - \lambda(\xi)T(\xi, t^{(1)})P_T^{(1)}(t^{(v)} - \zeta\Delta t) + \sum_{k=1}^{K_\Gamma+K_\Omega} \sum_{r=1}^{K_\tau} [T^{(k,r)} A_1^{(k, vr)}(\xi) + q^{(k,r)} A_2^{(k, vr)}(\xi)] \right) \quad (3.11)$$

where $A_n^{(k, vr)}(\xi) = A_n^{(k, r)}(\xi, t^{(v)} - \zeta\Delta t)$ for $n=1,2$; $[X_{wv}]$ is multiplicative inverse of $[Y_{wv}]$ with $Y_{wv} = P_T^{(w)}(t^{(v)} - \zeta\Delta t)$. In this section, one may see that we have set up two systems of algebraic equations before $T(\xi, t)$ can be computed, the first one is for solving either T or q at all the space and time nodes, while the other is for solving $T(\xi)$ at the time nodes.

3.5 Specific Test Problems

To test the proposed DRBEM with the modified polynomial time functions, three different numerical examples are presented in this section. Equations (3.10) and (3.11) are initially employed for solving unknowns at all space and time nodal points. The numerical results are then computed using (3.7).

Problem 3.1

Figure 3.1: Case considered in Problem 3.1 ($M=16$, $K_\Omega=9$).

Lest us consider a square domain $\Omega = \{x: 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}$ as shown in Figure 3.1. The governing PDE (1.24) is subject to the following initial-boundary conditions

$$\text{in } \Omega: T_0(\mathbf{x}) = 1 + \frac{1}{2}x_1^2 - \frac{2}{3}x_2^2 + \sin\left(\frac{4}{3}x_2 + \frac{\pi}{4}\right) \exp\left(x_1 - \frac{2}{3}x_2\right)$$

$$\text{in } \Omega: \dot{T}(\mathbf{x}, 0) = -\sin\left(\frac{4}{3}x_2 + \frac{\pi}{4}\right) \exp\left(x_1 - \frac{2}{3}x_2\right)$$

$$\text{on } \Gamma_1(t \geq 0): T(x_1, t) = 1 + \frac{1}{2}x_1^2 + \frac{1}{\sqrt{2}} \exp(x_1 - t)$$

$$\text{on } \Gamma_2(t \geq 0): q(x_2, t) = -2 + \frac{4}{3}x_2 - \frac{4}{3} \left[\sin\left(\frac{4}{3}x_2 + \frac{\pi}{4}\right) + \cos\left(\frac{4}{3}x_2 + \frac{\pi}{4}\right) \right] \exp\left(1 - \frac{2}{3}x_2 - t\right)$$

$$\text{on } \Gamma_3(t \geq 0): q(x_1, t) = 2 - x_1 - 2 \cos\left(\frac{4}{3} + \frac{\pi}{4}\right) \exp\left(x_1 - \frac{2}{3} - t\right)$$

$$\text{on } \Gamma_4(t \geq 0): T(x_2, t) = 1 - \frac{2}{3}x_2^2 + \sin\left(\frac{4}{3}x_2 + \frac{\pi}{4}\right) \exp\left(-\frac{2}{3}x_2 - t\right)$$

The corresponding thermal properties are given by $c_{11}=2$, $c_{12}=1$, $c_{22}=3/2$, $\alpha=1$, $\tau_T=1/2$, $\tau_h=1/3$.

It is easy to check that the IBVP defined above has the exact solution

$$T(\mathbf{x}, t) = 1 + \frac{1}{2}x_1^2 - \frac{2}{3}x_2^2 + \sin\left(\frac{4}{3}x_2 + \frac{\pi}{4}\right) \exp\left(x_1 - \frac{2}{3}x_2 - t\right).$$

To evaluate the accuracy and convergent ability of the proposed method, one interior point (0.5, 0.6), two smooth boundary points (1, 0.5) and (0.6, 1) and one point at corner (1, 1) are selected for computing temperature at 18 different incremental t from 0.05 to 8. The exact values are tabulated in Table 3.1. For the BEM results, the boundary is firstly discretised into $M=16$ DLEs and $K_\Omega=9$ well-spaced out internal collocation points are selected. Finer elements and more internal collocation points are then used, i.e. with $M=32$, $K_\Omega=16$. The numerical results obtained using $K_\tau=4$, $\Delta t=0.1$, $\zeta=0$ and the overall norm of errors are shown in Table 3.2.

By comparing Table 3.1 and Table 3.2, the exact and the computed results are in excellent agreement even at the points on boundary and at corner. In Table 3.2, there is a significant improvement in overall norm of error as M and K_Ω are increased, i.e. convergence is observed. Note that the $\lambda(\xi)$ in (3.11) is given by 0.5 and 0.25 for boundary and corner points respectively.

Table 3.1: Exact $T(\xi, t)$. (Problem 3.1)

t	$\xi^{(1)}=(0.5,0.6)$	$\xi^{(2)}=(1,0.5)$	$\xi^{(3)}=(0.6,1)$	$\xi^{(4)}=(1,1)$
0.05	1.9361590	3.1730311	1.2729384	1.9665307
0.25	1.7456162	2.8395505	1.1352454	1.7611168
0.45	1.5896130	2.5665196	1.0225119	1.5929382
0.65	1.4618883	2.3429808	0.9302135	1.4552452
0.85	1.3573162	2.1599628	0.8546460	1.3425117
1.05	1.2716998	2.0101203	0.7927765	1.2502134
1.25	1.2016030	1.8874396	0.7421221	1.1746458
1.45	1.1442126	1.7869971	0.7006497	1.1127764
1.65	1.0972253	1.7047618	0.6666951	1.0621220
1.85	1.0587554	1.6374332	0.6388953	1.0206496
2.05	1.0272589	1.5823093	0.6161348	0.9866949
2.5	0.9757083	1.4920874	0.5788825	0.9311210
3	0.9400174	1.4296225	0.5530910	0.8926446
3.5	0.9183697	1.3917356	0.5374476	0.8693074
4	0.9052398	1.3687561	0.5279594	0.8551527
5	0.8924458	1.3463646	0.5187141	0.8413603
6	0.8877392	1.3381272	0.5153129	0.8362863
8	0.8853707	1.3339820	0.5136014	0.8337330

Table 3.2: Results computed with $K_\tau=4$, $\Delta t=0.1$, $\zeta=0$. (Problem 3.1)

t	$M=16, K_\Omega=9$				$M=32, K_\Omega=16$			
	$\xi^{(1)}$	$\xi^{(2)}$	$\xi^{(3)}$	$\xi^{(4)}$	$\xi^{(1)}$	$\xi^{(2)}$	$\xi^{(3)}$	$\xi^{(4)}$
0.05	1.9361590	3.1732392	1.2730178	1.9664953	1.9361586	3.1730829	1.2729358	1.9665191
0.25	1.7456147	2.8414605	1.1359666	1.7608158	1.7456131	2.8400310	1.1352220	1.7610194
0.45	1.5896097	2.5694669	1.0236112	1.5925640	1.5896073	2.5672591	1.0224754	1.5928113
0.65	1.4618882	2.3465488	0.9315301	1.4549110	1.4618831	2.3438760	0.9301714	1.4551243
0.85	1.3573212	2.1638710	0.8560731	1.3422814	1.3573125	2.1609426	0.8546025	1.3424166
1.05	1.2717101	2.0141941	0.7942485	1.2501151	1.2716973	2.0111398	0.7927335	1.2501524
1.25	1.2016196	1.8915751	0.7436013	1.1746890	1.2016027	1.8884737	0.7420814	1.1746232
1.45	1.1442354	1.7911332	0.7021150	1.1129572	1.1442146	1.7880304	0.7006121	1.1127916
1.65	1.0972531	1.7088639	0.6681348	1.0624292	1.0972288	1.7057849	0.6666600	1.0621716
1.85	1.0587883	1.6414860	0.6403057	1.0210710	1.0587609	1.6384431	0.6388633	1.0207311
2.05	1.0272964	1.5863066	0.6175154	0.9872165	1.0272662	1.5833046	0.6161057	0.9868045
2.5	0.9757534	1.4959646	0.5802021	0.9318188	0.9757183	1.4930508	0.5788583	0.9312797
3	0.9400682	1.4333952	0.5543599	0.8934738	0.9400293	1.4305583	0.5530706	0.8928399
3.5	0.9184242	1.3954367	0.5386826	0.8702203	0.9183830	1.3926526	0.5374297	0.8695262
4	0.9052968	1.3724109	0.5291728	0.8561180	0.9052541	1.3696610	0.5279434	0.8553864
5	0.8925056	1.3499725	0.5199056	0.8423778	0.8924613	1.3472576	0.5186999	0.8416089
6	0.8877995	1.3417166	0.5164958	0.8373227	0.8877547	1.3390151	0.5152990	0.8365399
8	0.8854319	1.3375628	0.5147803	0.8347797	0.8853868	1.3348680	0.5135881	0.8339898
Norm of error =				1.94060E-03	4.57350E-04			

Problem 3.2

It would be interesting to see how the response of the proposed BEM is when the boundary is subjected to a fluctuating boundary value that is different from the usual exponential time evolution as seen in *Problem 3.1*. One such form of boundary value is a time sinusoidal function. For this purpose, let us consider again the 2D square domain Figure 3.1 with the governing PDE (1.24) subject to the mixed initial-boundary conditions:

$$\begin{aligned}
 \text{in } \Omega: T_0(\mathbf{x}) &= 2 - \frac{1}{2}x_1x_2 + \frac{1}{6}x_2^2 - \frac{1}{2}\sin(0.7448864x_2) \exp\left(\frac{1}{2}x_1 + 0.4529425x_2\right) \\
 \text{in } \Omega: \dot{T}(\mathbf{x}, 0) &= \frac{1}{2}[\sin(0.7448864x_2) - f_q \cos(0.7448864x_2)] \exp\left(\frac{1}{2}x_1 + 0.4529425x_2\right) \\
 \text{on } \Gamma_1(t \geq 0): q(x_1, t) &= -\frac{3}{2}x_1 - [0.9294138\sin(3t) + 1.1173296\cos(3t)] \exp(x_1/2 - t) \\
 \text{on } \Gamma_2(t \geq 0): T(x_2, t) &= 2 - \frac{1}{2}x_2 + \frac{1}{6}x_2^2 - \frac{1}{2}\sin(0.7448864x_2 + 3t) \exp\left(\frac{1}{2} + 0.4529425x_2 - t\right) \\
 \text{on } \Gamma_3(t \geq 0): T(x_1, t) &= \frac{13}{6} - \frac{1}{2}x_1 - \frac{1}{2}\sin(0.7448864 + 3t) \exp\left(\frac{1}{2}x_1 + 0.4529425 - t\right) \\
 \text{on } \Gamma_4(t \geq 0): q(x_2, t) &= \frac{1}{6}x_2 - [0.4764713\sin(0.7448864x_2 + 3t) \\
 &\quad + 0.3724432\cos(0.7448864x_2 + 3t)] \exp(0.4529425x_2 - t)
 \end{aligned}$$

One can easily see that the frequency of the sine wave is $f_q=3$. The material properties are given by $c_{11}=1$, $c_{12}=1$, $c_{22}=3$, $\alpha=1$, $\tau_T=1/3$, $\tau_h=1/4$. The exact solution to this problem is

$$T(\mathbf{x}, t) = 2 - \frac{1}{2}x_1x_2 + \frac{1}{6}x_2^2 - \frac{1}{2}\sin(0.7448864x_2 + 3t) \exp\left(\frac{1}{2}x_1 + 0.4529425x_2 - t\right).$$

To assess the results obtained with different order of time element K_τ , one interior point (0.3, 0.8) is selected for computing the solution at 16 different incremental t using $\zeta=0$ and $\Delta t=0.08$. The boundary is discretised with DLEs using $M=16$, $K_\Omega=9$ initially and then with finer elements using $M=32$, $K_\Omega=16$. The numerical results obtained are tabulated in Table 3.3 and Table 3.4. The norm of error is computed at bottom of each column.

Both tables show that the computed results agree well with the exact ones. By comparing the norm of errors column by column between both tables, we see that the errors have improved significantly as M and K_Ω are increased, especially for $K_\tau \geq 4$. The improvement is not so obvious for quadratic time element i.e. $K_\tau=2$. From each table, it is clear that the time element with $K_\tau=4$ delivers more accurate numerical values than the cubic ($K_\tau=3$) and quadratic ($K_\tau=2$) ones. However there is little improvement as K_τ is increased further ≥ 7 . Several different problems with various BCs and boundary geometries have been tested. It was found that the optimum K_τ and Δt are problem dependent. However, in general, good

Chapter 3 Micro-Scale Heat Conduction in Homogeneous Anisotropic Materials

results can be obtained with $\zeta=0$, $K_{\tau} \geq 3$, $L_{\min}/5 < \Delta t < K_{\tau}/10$ for $f_q=0$, and $L_{\min}/5 < \Delta t < \pi/(4|f_q|)$ for $|f_q| \neq 0$.

Table 3.3: Results computed at $\xi=(0.3,0.8)$ with different K_{τ} at $M=16$, $K_{\Omega}=9$, $\Delta t=0.08$, $\zeta=0$ (Problem 3.2)

t	Exact $T(\xi, t)$	Computed $T(\xi, t)$					
		$K_{\tau}=2$	$K_{\tau}=3$	$K_{\tau}=4$	$K_{\tau}=5$	$K_{\tau}=7$	$K_{\tau}=10$
0.03	1.4736685	1.4736645	1.4737100	1.4736733	1.4736673	1.4736681	1.4736683
0.28	1.3616132	1.3609265	1.3615423	1.3615622	1.3615322	1.3615341	1.3615342
0.53	1.5854551	1.5823146	1.5851951	1.5853400	1.5852861	1.5852845	1.5852844
0.78	1.9085280	1.9039359	1.9080564	1.9083897	1.9083431	1.9083442	1.9083440
1.03	2.1409608	2.1368240	2.1405759	2.1408553	2.1408310	2.1408356	2.1408352
1.28	2.2099060	2.2075986	2.2097449	2.2098658	2.2098582	2.2098640	2.2098636
1.53	2.1475039	2.1471774	2.1475618	2.1475036	2.1475162	2.1475256	2.1475250
1.78	2.0345681	2.0354766	2.0346891	2.0346055	2.0346021	2.0346134	2.0346130
2.03	1.9437063	1.9449446	1.9438945	1.9437233	1.9437285	1.9437413	1.9437407
2.28	1.9086519	1.9094188	1.9087509	1.9086704	1.9086532	1.9086608	1.9086605
2.5	1.9201530	1.9204036	1.9202071	1.9201378	1.9201331	1.9201413	1.9201409
3	1.9937430	1.9933100	1.9937457	1.9937138	1.9937097	1.9937202	1.9937197
3.5	2.0117429	2.0116302	2.0117902	2.0117259	2.0117280	2.0117375	2.0117370
4	1.9862152	1.9863633	1.9862690	1.9862056	1.9862051	1.9862154	1.9862149
4.5	1.9774029	1.9774388	1.9774295	1.9773923	1.9773907	1.9773972	1.9773967
5	1.9860378	1.9859551	1.9860510	1.9860281	1.9860215	1.9860295	1.9860291
norm of error =		1.891E-03	1.842E-04	5.672E-05	7.604E-05	7.549E-05	7.557E-05

Table 3.4: Results computed at $\xi=(0.3,0.8)$ with different K_{τ} at $M=32$, $K_{\Omega}=16$, $\Delta t=0.08$, $\zeta=0$ (Problem 3.2)

t	Exact $T(\xi, t)$	Computed $T(\xi, t)$					
		$K_{\tau}=2$	$K_{\tau}=3$	$K_{\tau}=4$	$K_{\tau}=5$	$K_{\tau}=7$	$K_{\tau}=10$
0.03	1.4736685	1.4736645	1.4737098	1.4736729	1.4736667	1.4736671	1.4736670
0.28	1.3616132	1.3609859	1.3616037	1.3616252	1.3615945	1.3615963	1.3615962
0.53	1.5854551	1.5824880	1.5853612	1.5855044	1.5854502	1.5854486	1.5854486
0.78	1.9085280	1.9041481	1.9082528	1.9085833	1.9085366	1.9085377	1.9085377
1.03	2.1409608	2.1369863	2.1407228	2.1409985	2.1409742	2.1409789	2.1409785
1.28	2.2099060	2.2076676	2.2098057	2.2099235	2.2099158	2.2099216	2.2099213
1.53	2.1475039	2.1471650	2.1475498	2.1474893	2.1475019	2.1475112	2.1475106
1.78	2.0345681	2.0354257	2.0346442	2.0345588	2.0345553	2.0345666	2.0345662
2.03	1.9437063	1.9448970	1.9438545	1.9436820	1.9436872	1.9437000	1.9436994
2.28	1.9086519	1.9093990	1.9087368	1.9086546	1.9086373	1.9086449	1.9086447
2.5	1.9201530	1.9204089	1.9202158	1.9201448	1.9201401	1.9201483	1.9201479
3	1.9937430	1.9933351	1.9937710	1.9937371	1.9937330	1.9937436	1.9937431
3.5	2.0117429	2.0116366	2.0117980	2.0117318	2.0117339	2.0117434	2.0117429
4	1.9862152	1.9863609	1.9862693	1.9862039	1.9862035	1.9862137	1.9862133
4.5	1.9774029	1.9774428	1.9774357	1.9773964	1.9773948	1.9774013	1.9774008
5	1.9860378	1.9859630	1.9860606	1.9860354	1.9860289	1.9860369	1.9860365
norm of error =		1.808E-03	1.125E-04	2.327E-05	1.145E-05	8.539E-06	8.509E-06

Problem 3.3

In this section, we seek to solve another problem numerically in which the square domain Figure 3.1 is subject to Neumann boundary condition:

in Ω :

$$T_0(\mathbf{x}) = 6 + x_1^2 - \frac{1}{2}x_2^2 - \cos\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right)\exp\left(\frac{3\sqrt{39}}{10}x_2\right)$$

$$\dot{T}(\mathbf{x}, 0) = \left[6\sin\left(\frac{2\sqrt{39}}{5}x_2 + \frac{\pi}{3}\right) + \frac{3}{4}\cos\left(\frac{2\sqrt{39}}{5}x_2 + \frac{\pi}{3}\right)\right]\exp\left(\frac{3\sqrt{39}}{10}x_2\right)$$

on boundary for $t \geq 0$:

$$\Gamma_1: q(x_1, t) = 2x_1 + \frac{\sqrt{39}}{20} \left[4\sin\left(6t + \frac{\pi}{3}\right) - 3\cos\left(6t + \frac{\pi}{3}\right)\right]\exp\left(-\frac{3}{4}t\right)$$

$$\Gamma_2: q(x_2, t) = -2 + \frac{1}{2}x_2 - \frac{\sqrt{39}}{20} \left[4\sin\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right) + 3\cos\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right)\right]\exp\left(\frac{3\sqrt{39}}{10}x_2 - \frac{3}{4}t\right)$$

$$\Gamma_3: q(x_1, t) = 2 - x_1 + \frac{\sqrt{39}}{5} \left[3\cos\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right) - 4\sin\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right)\right]\exp\left(\frac{3\sqrt{39}}{10}x_2 - \frac{3}{4}t\right)$$

$$\Gamma_4: q(x_2, t) = -\frac{1}{2}x_2 + \frac{\sqrt{39}}{20} \left[4\sin\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right) - 3\cos\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right)\right]\exp\left(\frac{3\sqrt{39}}{10}x_2 - \frac{3}{4}t\right)$$

The heat flux on the boundary is highly fluctuating with frequency $f_q=6$. The exact solution which satisfies the PDE (1.24) and the above initial-boundary conditions is

$$T(\mathbf{x}, t) = 6 + x_1^2 - \frac{1}{2}x_2^2 - \cos\left(\frac{2\sqrt{39}}{5}x_2 + 6t + \frac{\pi}{3}\right)\exp\left(\frac{3\sqrt{39}}{10}x_2 - \frac{3}{4}t\right)$$

The thermal parameters are given by $c_{11}=1$, $c_{12}=1/2$, $c_{22}=2$, $\alpha=2$, $\tau_T=2/5$ and $\tau_h=2/3$.

For the numerical results, three interior points (0.1, 0.3), (0.5, 0.3) and (0.9, 0.3) are selected for computing the solution at time from $t=0$ to 5 using $M=20$, $K_{\mathcal{L}}=25$, $K_{\mathcal{E}}=5$, $\Delta t=0.06$ and $\zeta=0$. From the results plotted in Figure 3.2, there is no doubt that the exact and numerical results are in good agreement.

As temperature is not specified on the boundary, the temperature at each boundary node is considered unknown at $t>0$ and they have to be computed using (3.10). The computed values of T at boundary nodes at $t=1.2$ are plotted in Figure 3.3. They match very well with the exact ones.

The corresponding computer code to solve the numerical problems in this chapter is shown in Appendix B2.

Chapter 3 Micro-Scale Heat Conduction in Homogeneous Anisotropic Materials

Figure 3.2: Plotting of exact and computed results at 3 interior points with $M=20$, $K_{\Omega}=25$, $K_{\tau}=5$, $\Delta t=0.06$, $\zeta=0$.
(Problem 3.3)

Figure 3.3: Exact and computed values on boundary at $t=1.2$ with $M=20$, $K_{\Omega}=25$, $K_{\tau}=5$, $\Delta t=0.06$, $\zeta=0$.
(Problem 3.3)

Chapter 4 Steady-State Anisotropic Heat Conduction Across a Homogeneously Imperfect Interfaces

As mentioned in section 1.4, in this chapter, we seek to devise a BEM using modified Green's function to solve numerically the steady-state heat conduction problem across imperfect interfaces of homogeneously anisotropic bimaterial without discretising the interface. The present work is an extension of the one by Ang *et al* [9].

4.1 Boundary Integral Equation

With reference to Figure 1.7, two different homogeneous anisotropic materials are bonded together at imperfect interface Γ^0 (i.e. from $x_1=a_1$ to a_2 at $x_2=0$, where $a_2>a_1$). The thermal field is governed by the following elliptic PDEs

$$c_{ij}^{\pm} \frac{\partial^2 T}{\partial x_i \partial x_j} = 0 \quad \text{in } \Omega^{\pm} \cup \Gamma^{\pm} \cup \Gamma^0 \quad (4.1)$$

$$\Rightarrow c_{ij}^{+} \frac{\partial^2 T}{\partial x_i \partial x_j} = 0 \text{ for } x_2 > 0 \text{ and } c_{ij}^{-} \frac{\partial^2 T}{\partial x_i \partial x_j} = 0 \text{ for } x_2 < 0.$$

By employing the reciprocal theorem (2.5) with $\varphi(\xi, t)$ and $c_{ij}^{(o)}$ replaced by $T(\xi)$ and the constants $c_{ij}^{\pm}$ respectively, two BIEs can be derived as follows

$$\lambda^{+}(\xi)T(\xi) = \int_{\Gamma^{+}} \left(T c_{ij}^{+} n_i \frac{\partial \Phi}{\partial x_j} - \Phi c_{ij}^{+} n_i \frac{\partial T}{\partial x_j} \right) d\Gamma^{+} + \int_{a_1}^{a_2} \left(\Phi c_{2j}^{+} \frac{\partial T}{\partial x_j} - T c_{2j}^{+} \frac{\partial \Phi}{\partial x_j} \right)_{x_2=0^{+}} dx_1 \quad (4.2)$$

and

$$\lambda^{-}(\xi)T(\xi) = \int_{\Gamma^{-}} \left(T c_{ij}^{-} n_i \frac{\partial \Phi}{\partial x_j} - \Phi c_{ij}^{-} n_i \frac{\partial T}{\partial x_j} \right) d\Gamma^{-} + \int_{a_1}^{a_2} \left(T c_{2j}^{-} \frac{\partial \Phi}{\partial x_j} - \Phi c_{2j}^{-} \frac{\partial T}{\partial x_j} \right)_{x_2=0^{-}} dx_1 \quad (4.3)$$

where $\lambda^{+}(\xi)=0$ if $\xi \notin \Omega^{+} \cup \Gamma^{+}$, $\lambda^{+}(\xi)=1$ if $\xi \in \Omega^{+}$, $0 < \lambda^{+}(\xi) < 1$ if $\xi \in \Gamma^{+}$, and $\lambda^{-}(\xi)=0$ if $\xi \notin \Omega^{-} \cup \Gamma^{-}$, $\lambda^{-}(\xi)=1$ if $\xi \in \Omega^{-}$, $0 < \lambda^{-}(\xi) < 1$ if $\xi \in \Gamma^{-}$. The fundamental solution Φ is same as in Chapter 2 ($c_{ij}^{\pm}$ replacing $c_{ij}^{(o)}$). However, from (4.2) and (4.3), it can be seen that there involves line integration along interface Γ^0 , this implies that Γ^0 has to be discretised in the later development of BEM if the both equations are to be used.

To develop a BEM without discretising the Γ^0 , firstly we consider to modify Φ for each BIE above, says $\Phi \rightarrow \Phi^+$ in (4.2) and $\Phi \rightarrow \Phi^-$ in (4.3), in such a way that (4.2) and (4.3) can be combined to form a single integral equation and eventually the line integral along Γ^0 can be eliminated. Note that such $\Phi^\pm$ have to satisfy $c_{ij}^\pm (\partial^2 \Phi^\pm / \partial x_i \partial x_j) = \delta(x_1 - \xi_1, x_2 - \xi_2)$. In doing so, firstly we let

$$c_{2j}^+ \frac{\partial T}{\partial x_j} \Big|_{x_2=0^+} = c_{2j}^- \frac{\partial T}{\partial x_j} \Big|_{x_2=0^-} \quad \text{and} \quad c_{2j}^+ \frac{\partial \Phi^+}{\partial x_j} \Big|_{x_2=0^+} = c_{2j}^- \frac{\partial \Phi^-}{\partial x_j} \Big|_{x_2=0^-} \quad (4.4)$$

which means normal derivatives of the functions are continuous at interface. It follows that we can combine (4.2) and (4.3) to yield:

$$\begin{aligned} \lambda(\xi)T(\xi) = & \oint_{\Gamma} \left(T c_{ij}^{\langle x \rangle} n_i \frac{\partial \Phi^{\langle x \rangle}}{\partial x_j} - \Phi^{\langle x \rangle} c_{ij}^{\langle x \rangle} n_i \frac{\partial T}{\partial x_j} \right) d\Gamma \\ & + \int_{a_1}^{a_2} \left(c_{2j}^+ \frac{\partial T}{\partial x_j} \Big|_{x_2=0^+} (\Phi^+|_{x_2=0^+} - \Phi^-|_{x_2=0^-}) - c_{2j}^+ \frac{\partial \Phi^+}{\partial x_j} \Big|_{x_2=0^+} (T|_{x_2=0^+} - T|_{x_2=0^-}) \right) dx_1 \end{aligned} \quad (4.5)$$

where $\lambda(\xi)$ is same as defined in Figure 1.5 with $\Gamma = \Gamma^+ \cup \Gamma^-$ and $\Omega = \Omega^+ \cup \Omega^-$. The notation $\langle x \rangle$ represents '+' or '-' sign if the point x belongs to Γ^+ or Γ^- respectively, i.e. $c_{ij}^{\langle x \rangle} = c_{ij}^\pm$ and $\Phi^{\langle x \rangle} = \Phi^\pm$ when $x \in \Gamma^\pm$. The notation $\langle x \rangle$ is introduced in this chapter in order to simplify the equation writing and the subsequent computer coding as well.

Perfect Interface

To obtain the BIE which just involves $\Gamma^+ \cup \Gamma^-$, Berger and Karageorghis [11] has successfully derived a modified Green's function (fundamental solution) $\Phi^\pm$ for the perfect interface. In the work of [11], the functions are considered continuous at interface:

$$T|_{x_2=0^+} = T|_{x_2=0^-} \quad \text{and} \quad \Phi^+|_{x_2=0^+} = \Phi^-|_{x_2=0^-}. \quad (4.6)$$

With this interface condition, the line integration along Γ^0 in (4.5) completely disappears.

In [11], the modified Green's functions satisfying (4.4) and (4.6) are found to be

$$\Phi^\pm = \Phi_*^\pm + \Phi_{\text{per}}^\pm \quad \text{for } \pm x_2 > 0$$

where

$$\Phi_*^\pm = \frac{1}{2\pi c^\pm} \text{Re}[\ln(z^\pm - z_0)] \quad (4.7)$$

and

$$\Phi_{\text{per}}^\pm = \frac{c^\pm - c^\mp}{2\pi c^\pm (c^+ + c^-)} \text{Re}[H(\pm \xi_2) \ln(z^\pm - \bar{z}_0) + H(\mp \xi_2) \ln(z^\pm - z_0)] \quad (4.8)$$

The notation $H(\pm\xi_2)$ is the usual Heaviside unit-step function: $H(\pm\xi_2)=1$ if $\pm\xi_2>0$, otherwise $H(\pm\xi_2)=0$; $z^\pm=x_1+\kappa^\pm x_2$, $z_0=\xi_1+\kappa^{\langle\xi\rangle}\xi_2$ where $\langle\xi\rangle=\pm$ sign' when $\xi\in\Omega^\pm\cup\Gamma^\pm$; $\kappa^\pm=[-c_{12}^\pm+ic_{22}^\pm]/c_{22}^\pm$ and $c^\pm=\sqrt{c_{11}^\pm c_{22}^\pm-(c_{12}^\pm)^2}>0$. The over bar seen in (4.8) denotes the complex conjugate of the complex number. Note that, the notations used to define $\Phi^\pm$ above are different from the ones in [11].

One can easily see that, for the bimaterial having same material properties, i.e. $c_{ij}^+=c_{ij}^-$, the perfect interface bimaterial case is reduced to the single material case. Thus, the function $\Phi^\pm$ is reduced to the one as defined in Chapter 2, i.e. $\Phi_{\text{per}}^\pm=0$ and $\Phi^\pm=\Phi_*^\pm=\Phi$.

Imperfect Interface

As mentioned early in section 1.4, in reality, there are microscopic gaps between materials along their interface. For solving such problem, Ang *et al* [9], in considering 2D steady-state temperature field in a thermally isotropic bimaterial with a homogeneously imperfect interface, has successfully proposed a special Green's function that satisfies the appropriate interface condition. Following this work, we now extend the method to our present anisotropic case.

According to [9], the Green's function $\Phi^\pm$ has to be further modified to

$$\Phi^\pm = \Phi_*^\pm + \Phi_{\text{per}}^\pm + \Phi_{\text{imp}}^\pm \text{ for } \pm x_2 > 0$$

where $\Phi_{\text{imp}}^\pm$ is an additional arbitrary term corresponding to the imperfect interface condition and it must satisfy $c_{ij}^\pm(\partial^2\Phi_{\text{imp}}^\pm/\partial x_i\partial x_j)=0$. The T and $\Phi^\pm$ must be allowed to jump across the interface at all points. In the present anisotropic case, T and $\Phi^\pm$ satisfy

$$c_{2j}^+ \frac{\partial T}{\partial x_j} \Big|_{x_2=0^+} = \beta(T|_{x_2=0^+} - T|_{x_2=0^-}) \text{ and } c_{2j}^+ \frac{\partial \Phi^+}{\partial x_j} \Big|_{x_2=0^+} = \beta(\Phi^+|_{x_2=0^+} - \Phi^-|_{x_2=0^-}) \quad (4.9)$$

for $a_1 \leq x_1 \leq a_2$. The term β is assumed to be a positive constant, i.e. the microscopic imperfection or damage is uniformly distributed over the interface. Equation (4.9) implies that the normal heat flux at interface is proportional to the temperature difference between the two close points $(x_1, 0^+)$ and $(x_1, 0^-)$. Note that if we let $\beta \rightarrow +\infty$, the jumps of the T and $\Phi^\pm$ in (4.9) vanish: $(T|_{x_2=0^+} - T|_{x_2=0^-}) \rightarrow 0$ and $(\Phi^+|_{x_2=0^+} - \Phi^-|_{x_2=0^-}) \rightarrow 0$. That is, the perfect interface condition (4.6) is recovered and hence $\Phi_{\text{imp}}^\pm=0$ accordingly.

With (4.9), one can easily see that (4.5) is subsequently reduced to the following simple BIE

$$\lambda(\xi)T(\xi) = \oint_{\Gamma} (T\phi^{(x)} + \Phi^{(x)} q) d\Gamma \quad (4.10)$$

where $\phi^{(x)} = c_{ij}^{(x)} n_i (\partial \Phi^{(x)} / \partial x_j)$ and the normal heat flux $q = -c_{ij}^{(x)} n_i (\partial T / \partial x_j)$.

From previous discussion, we know that (4.6) is actually $(\Phi_*^+ + \Phi_{\text{per}}^+)|_{x_2=0^+} = (\Phi_*^- + \Phi_{\text{per}}^-)|_{x_2=0^-}$, thus (4.4) and (4.9) involving $\Phi^\pm$ can be simplified to the following imperfect interface conditions respectively

$$(c_{2j}^+ \frac{\partial}{\partial x_j} \Phi_{\text{imp}}^+) \Big|_{x_2=0^+} = (c_{2j}^- \frac{\partial}{\partial x_j} \Phi_{\text{imp}}^-) \Big|_{x_2=0^-} \quad (4.11)$$

and

$$(c_{2j}^+ \frac{\partial}{\partial x_j} \Phi^+) \Big|_{x_2=0^+} = \beta (\Phi_{\text{imp}}^+|_{x_2=0^+} - \Phi_{\text{imp}}^-|_{x_2=0^-}) \quad (4.12)$$

The task to determine $\Phi_{\text{imp}}^\pm$ is presented in the following section.

4.2 Derivation of a Special Green's Function

To determine $\Phi_{\text{imp}}^\pm$ which satisfy (4.11) and (4.12) for $-\infty < x_1 < +\infty$, we may take the following Fourier's representations

$$\Phi_{\text{imp}}^\pm(\mathbf{x}, \xi) = \frac{1}{2\pi} \text{Re} \left(\int_0^\infty Q^\pm(v, \xi) e^{\pm i v z^\pm} dv \right) \quad (4.13)$$

where $Q^\pm(v, \xi)$ are possibly complex function of v which is yet to be determined. It is easy to check that $\Phi_{\text{imp}}^\pm \rightarrow 0$ as $x_2 \rightarrow \pm\infty$ and $\Phi_{\text{imp}}^\pm$ satisfy $c_{ij}^\pm (\partial^2 \Phi_{\text{imp}}^\pm / \partial x_i \partial x_j) = 0$.

By substituting (4.13) into (4.11), we obtain

$$\left[c_{2j}^+ \frac{\partial}{\partial x_j} \int_0^\infty (Q^+ e^{i v z^+} + \overline{Q^+} e^{-i v z^+}) dv \right] \Big|_{x_2=0^+} = \left[c_{2j}^- \frac{\partial}{\partial x_j} \int_0^\infty (Q^- e^{-i v z^-} + \overline{Q^-} e^{i v z^-}) dv \right] \Big|_{x_2=0^-}$$

which can be further simplified to

$$-c^+ \int_0^\infty v (Q^+ e^{i v x_1} + \overline{Q^+} e^{-i v x_1}) dv = c^- \int_0^\infty v (Q^- e^{-i v x_1} + \overline{Q^-} e^{i v x_1}) dv$$

Multiplying above by $\exp(-i w x_1)$ on both sides, where w is a real positive constant, and integrating over x_1 from $-\infty$ to ∞ , we obtain:

Chapter 4 Steady-State Anisotropic Heat Conduction Across a Homogeneously Imperfect Interfaces

$$-c^+ \int_0^\infty v \left(Q^+ \int_{-\infty}^\infty e^{i(v-w)x_1} dx_1 + \overline{Q^+} \int_{-\infty}^\infty e^{-i(v+w)x_1} dx_1 \right) dv = c^- \int_0^\infty v \left(Q^- \int_{-\infty}^\infty e^{-i(v+w)x_1} dx_1 + \overline{Q^-} \int_{-\infty}^\infty e^{i(v-w)x_1} dx_1 \right) dv \quad (4.14)$$

To further simplify (4.14), we can use the following result

$$\int_{-\infty}^\infty e^{i(v-w)x_1} dx_1 = 2\pi \delta(v-w), \quad (4.15)$$

where $\delta(v-w)$ is one dimensional Dirac-delta function in such a way that

$\int_0^\infty f(v) \delta(v-w) dv = f(w)$ if $v=w$, otherwise it equals to zero. Using (4.15), we can see that the

first and second term respectively on the right and left of (4.14) disappear, since it is impossible for $v=-w$, as both v and w are real positives. It follows that (4.14) is reduced to

$$-c^+ Q^+(w, \xi) = c^- \overline{Q^-}(w, \xi). \quad (4.16)$$

By substituting (4.13) into (4.12), we obtain:

$$\begin{aligned} & \left[c_{2j}^+ \frac{\partial}{\partial x_j} (\Phi_*^+ + \Phi_{\text{per}}^+) + c_{2j}^+ \frac{\partial}{\partial x_j} \int_0^\infty \frac{1}{4\pi} (Q^+ e^{ivz^+} + \overline{Q^+} e^{-ivz^+}) dv \right]_{x_2=0^+} \\ &= \frac{\beta}{4\pi} \int_0^\infty \left([Q^+ e^{ivz^+} + \overline{Q^+} e^{-ivz^+}]_{x_2=0^+} - [Q^- e^{-ivz^-} + \overline{Q^-} e^{ivz^-}]_{x_2=0^-} \right) dv \end{aligned}$$

By substituting (4.7), (4.8) and (4.16), the above equation becomes

$$\frac{i}{2\pi(c^+ + c^-)} [H(\xi_2)c^- + H(-\xi_2)c^+] \left(\frac{1}{x_1 - z_0} - \frac{1}{x_1 - \bar{z}_0} \right) = \frac{1}{4\pi} \int_0^\infty \left[\left(c^+ v + \beta \left(1 + \frac{c^+}{c^-} \right) \right) (Q^+ e^{ivx_1} + \overline{Q^+} e^{-ivx_1}) \right] dv$$

Again, we can multiply above equation by $\exp(-iwx_1)$ and integrate over x_1 from $-\infty$ to ∞ .

Then, by using the result (4.15), we get

$$\frac{1}{2} \left(\beta \left(1 + \frac{c^+}{c^-} \right) + c^+ w \right) Q^+(w, \xi) = \frac{i}{2\pi(c^+ + c^-)} [H(\xi_2)c^- + H(-\xi_2)c^+] \int_{-\infty}^\infty \left(\frac{1}{x_1 - z_0} - \frac{1}{x_1 - \bar{z}_0} \right) e^{-iwx_1} dx_1$$

At this stage, after applying the following result from [21]

$$\int_{-\infty}^\infty \frac{\exp(-itx)}{(\zeta + i\eta - x)} dx = H(-\eta) 2\pi \exp(-it[\zeta + i\eta]),$$

we obtain

$$\frac{1}{2} \left(\beta \left(1 + \frac{c^+}{c^-} \right) + c^+ w \right) Q^+ = \frac{1}{c^+ + c^-} [H(\xi_2)c^- + H(-\xi_2)c^+] \left[H \left(\frac{-c^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) e^{-iwx_0} - H \left(\frac{c^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) e^{-iwx_0} \right].$$

From the definition of unit-step function, the above equation reduces to

$$\left(\beta(1 + \frac{c^+}{c^-}) + c^+ w \right) Q^+ = \frac{-2}{c^+ + c^-} [H(\xi_2) c^- e^{-i w \bar{z}_0} - H(-\xi_2) c^+ e^{-i w z_0}]$$

With the above result, (4.16) gives,

$$\left(\beta(1 + \frac{c^-}{c^+}) + c^- w \right) Q^- = \frac{2}{c^+ + c^-} [H(\xi_2) c^- e^{i w z_0} - H(-\xi_2) c^+ e^{-i w \bar{z}_0}]$$

Finally, $Q^\pm$ are determined and can be written in the following compact form

$$Q^\pm(w, \xi) = \frac{-2}{(c^+ + c^-) c^\pm} \frac{H(\pm \xi_2) c^\mp e^{\mp i w \bar{z}_0} - H(\mp \xi_2) c^\pm e^{\mp i w z_0}}{\beta(1/c^+ + 1/c^-) + w} \quad (4.17)$$

As one can see that, as $\beta \rightarrow \infty$, $Q^\pm \rightarrow 0$, thus $\Phi_{\text{imp}}^\pm \rightarrow 0$. This is consistent with our earlier discussion that the perfect interface case is recovered when $\beta \rightarrow \infty$. Note that, for $\xi_2 \neq 0$, the function $Q^\pm$ decay exponentially to zero as w increases. Therefore $\Phi_{\text{imp}}^\pm$ as given by (4.13) and its normal derivative $\phi_{\text{imp}}^\pm = c_{ij}^\pm n_i (\partial^2 \Phi_{\text{imp}}^\pm / \partial x_j)$ can be easily computed numerically using a suitable integration formula in Reference [1]. The analytical forms of $\Phi_{\text{imp}}^\pm$ and $\phi_{\text{imp}}^\pm$ are shown in Appendix A2.4 just for reference. In fact, the numerical values of $\Phi_{\text{imp}}^\pm$ and $\phi_{\text{imp}}^\pm$ are not required to be determined in the later BEM calculation since the boundary is going to be discretised with DLEs instead of constant elements. With constant elements, functions are considered constants over each boundary element.

4.3 BIE with DLEs Discretisation

As the required Green's function has been determined, the discretised BIE can be readily formed from (4.10) by approximating the exterior boundary Γ with M number DLEs:

$$\lambda(\xi) T(\xi) \approx \sum_{m=1}^M \int_{\Gamma^{(m)}} [T \phi^{(x)} + q \Phi^{(x)}] d\Gamma$$

As there are $2M$ nodes on Γ given by $\mathbf{x}^{(k)}$ (for $k=1, 2, \dots, 2M$), and the element's geometry and functions are respectively approximated by (2.8) and (2.9), thus

$$\lambda(\xi) T(\xi) \approx \sum_{k=1}^{2M} [T^{(k)} I^{(k)}(\xi) + q^{(k)} II^{(k)}(\xi)] \quad (4.18)$$

with the integration parameters

$$I^{(k)}(\xi) = \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1 - 2l_{\hat{k}} - (-1)^{\hat{k}} u}{2(1 - l_1 - l_2)} \phi^{(\mathbf{x}^{(k)})}(\mathbf{x}(u), \xi) du = I_*^{(k)}(\xi) + I_{\text{per}}^{(k)}(\xi) + I_{\text{imp}}^{(k)}(\xi)$$

$$H^{(k)}(\xi) = \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1-2I_{\hat{k}} - (-1)^{\hat{k}} u}{2(1-l_1-l_2)} \Phi^{\langle \mathbf{x}^{(k)} \rangle}(\mathbf{x}(u), \xi) du = H_*^{(k)}(\xi) + H_{\text{per}}^{(k)}(\xi) + H_{\text{imp}}^{(k)}(\xi)$$

Notation $\langle \mathbf{x}^{(k)} \rangle$ represents ' $\pm$ ' sign when $\mathbf{x}^{(k)} \in \Gamma^\pm$.

As usual, to solve for unknowns either $T^{(k)}$ or $q^{(k)}$ on Γ , we can collocate ξ at $\mathbf{x}^{(p)}$ (for $p=1,2,\dots,2M$) to generate a $2M \times 2M$ algebraic system. Once all unknowns are determined, T at any point $\xi \in \Omega \cup \Gamma$ can be calculated from (4.18) immediately.

To implement the computer code for calculating the numerical results, it is always a good practice to find the explicit solution to each integral in order to speed up the computing process. The analytical solutions to the integration parameters $I_*^{(k)}$, $H_*^{(k)}$, $I_{\text{per}}^{(k)}$, $H_{\text{per}}^{(k)}$, $I_{\text{imp}}^{(k)}$ and $H_{\text{imp}}^{(k)}$ are shown in the Appendix A2.1-A2.3.

4.4 Specific Test Problems

To test the BEM using the proposed Green's function without discretising the interface, let us consider the following two numerical examples.

Problem 4.1

Figure 4.1: Case considered in Problem 4.1 ($M=20$).

With reference to a semicircle in Figure 4.1, we have Ω^+ being the region: $\{x_1 \geq 0, x_2 > 0, x_1^2 + x_2^2 \leq 1\}$ with $c_{11}^+ = 2$, $c_{12}^+ = 1$ and $c_{22}^+ = 1$; and Ω^- being

$\{x_1 \geq 0, x_2 < 0, x_1^2 + x_2^2 \leq 1\}$ with $c_{11}^- = 5/2$, $c_{12}^- = 1$ and $c_{22}^- = 2$. On the interface between Ω^+ and Ω^- , i.e. $0 \leq x_1 \leq 1, x_2 = 0$, the condition (4.9) is imposed with $\beta = 1/2$.

A solution, which satisfies (4.1) and the interface condition (4.4) and (4.9) with the above stated $c_{ij}^\pm$ and β , is given by

$$\begin{aligned} T(\mathbf{x}) &= 2\exp\left(\frac{x_2}{4}\right) \left[\cos\left(\frac{x_1 - x_2}{4}\right) - \sin\left(\frac{x_1 - x_2}{4}\right) \right] \text{ for } \mathbf{x} \in \Omega^+ \\ T(\mathbf{x}) &= \exp\left(\frac{x_2}{4}\right) \left[\cos\left(\frac{x_1}{4} - \frac{x_2}{8}\right) - \sin\left(\frac{x_1}{4} - \frac{x_2}{8}\right) \right] \text{ for } \mathbf{x} \in \Omega^- \end{aligned} \quad (4.19)$$

To derive a test problem, (4.19) is used to generate boundary values T on curve $\Gamma_1 \cup \Gamma_4: \{x_1 \geq 0, x_1^2 + x_2^2 = 1\}$ and normal heat flux q on $\Gamma_2 \cup \Gamma_3: \{x_1 = 0, -1 \leq x_2 \leq 1\}$:

$$\begin{aligned} \text{On } \Gamma_1 (t \geq 0): T(\mathbf{x}) &= 2\exp\left(\frac{x_2}{4}\right) \left[\cos\left(\frac{x_1 - x_2}{4}\right) - \sin\left(\frac{x_1 - x_2}{4}\right) \right] \\ \text{On } \Gamma_2 (t \geq 0): q(x_2) &= \cos\left(\frac{x_2}{4}\right) \exp\left(\frac{x_2}{4}\right) \\ \text{On } \Gamma_3 (t \geq 0): q(x_2) &= \frac{1}{4} \left[\sin\left(\frac{x_2}{8}\right) + 3\cos\left(\frac{x_2}{8}\right) \right] \exp\left(\frac{x_2}{4}\right) \\ \text{On } \Gamma_4 (t \geq 0): T(\mathbf{x}) &= \exp\left(\frac{x_2}{4}\right) \left[\cos\left(\frac{x_1}{4} - \frac{x_2}{8}\right) - \sin\left(\frac{x_1}{4} - \frac{x_2}{8}\right) \right] \end{aligned}$$

To implement the BEM, the boundary is discretised into M DLEs. In order to ease computation and also avoid ambiguity, the elements are required to be either on Ω^+ and Ω^- but not partly on both regions. The ends of the elements which are close to the interface are allowed to be on the interface.

To evaluate the accuracy of the method, 9 interior test points including 3 points close to the interface are selected. The exact and numerical results are tabulated in Table 4.1. Since T is not specified on $\Gamma_2 \cup \Gamma_3$, another 9 points on these boundaries are also chosen for computing and the results are shown in Table 4.2. From both tables, the numerical results computed are in good agreement with the exact ones even at points close to the interface (last 3 rows in Table 4.1) and the one at (0, -0.01) in Table 4.2. The results percentage errors are improved significantly as M is increased from $M=20$ to $M=42$ that is convergence is observed.

The flux q is not specified on $\Gamma_1 \cup \Gamma_4$. To assess the accuracy of the computed q along the boundary, the exact and the computed $q^{(k)}$ (with $M=42$) at boundary nodes along $\Gamma_1: \{x_1 \geq 0, x_2 > 0, x_1^2 + x_2^2 = 1\}$, are plotted for comparison. Figure 4.2 shows that they are in excellent agreement.

Note that, due to the following considerations, points placed exactly on the interface $x_2=0$ are not selected for computing in the test problems. As mentioned in section 1.4, in reality, microscopic gaps exist between materials along their interface at the microscopic level. Thus, no temperature can be observed at the points where gaps exist. Secondly, the points at $x_2=0$ neither belong to Ω^+ nor Ω^- , hence the material properties are undetermined. Therefore, it does not make physical sense to apply BIE (4.18) to compute any value at the points on such temperature-discontinuous interface. This is unlike the case of perfect interface problem in which the temperature is continuous on the interface.

However, from the computational point of view, there is no problem in computing the temperature at such points on the imperfect interface case provided that the material properties of the points are properly assigned.

Table 4.1: Results at interior points (*Problem 4.1*)

ξ	Exact $T(\xi)$	$M=20$		$M=42$	
		Computed	% error	computed	% error
(0.12941,-0.48296)	0.8003975	0.8003641	4.1700E-03	0.8003916	7.3602E-04
(0.35355,-0.35355)	0.7863600	0.7863390	2.6703E-03	0.7863552	6.1347E-04
(0.48296,-0.12941)	0.8269604	0.8269379	2.7191E-03	0.8269551	6.3071E-04
(0.48296,0.12941)	1.8753470	1.8753427	2.3051E-04	1.8753437	1.7588E-04
(0.35355,0.35355)	2.1848246	2.1848272	1.2036E-04	2.1848225	9.3923E-05
(0.12941,0.48296)	2.4470585	2.4470845	1.0618E-03	2.4470591	2.1518E-05
(0.25,0.01)	1.8811701	1.8811543	8.4289E-04	1.8811665	1.9149E-04
(0.5,-0.01)	0.8639636	0.8639393	2.8115E-03	0.8639579	6.5350E-04
(0.75,0.01)	1.6019794	1.6020619	5.1503E-03	1.6019765	1.8178E-04

Table 4.2: Results at selected points on boundary $x_1=0$ ($=\Gamma_2 \cup \Gamma_3$) (*Problem 4.1*)

ξ	Exact $T(\xi)$	$M=20$		$M=42$	
		computed	% error	computed	% error
(0,0.8)	2.879423	2.879435	4.2166E-04	2.879426	1.0240E-04
(0,0.6)	2.644821	2.644811	3.6680E-04	2.644825	1.6256E-04
(0,0.4)	2.419965	2.419956	3.7620E-04	2.419968	1.2509E-04
(0,0.2)	2.204998	2.205001	1.4800E-04	2.204997	2.4737E-05
(0,-0.01)	0.996255	0.996115	1.4141E-02	0.996227	2.8926E-03
(0,-0.2)	0.927154	0.927115	4.1827E-03	0.927146	8.8352E-04
(0,-0.4)	0.858484	0.858453	3.5067E-03	0.858477	7.1203E-04
(0,-0.6)	0.793796	0.793774	2.7195E-03	0.793791	5.5633E-04
(0,-0.8)	0.732904	0.732896	1.0122E-03	0.732902	2.9552E-04

Figure 4.2: Exact and computed normal flux at boundary nodes on Γ_1 with $M=42$. (Problem 4.1)

Problem 4.2

Figure 4.3: Case considered in Problem 4.2 ($M=48$)

Let us consider another numerical example with a hollow circle depicted in Figure 4.3 in which domains Ω^+ and Ω^- are respectively bounded by boundaries ' $x_2=0^+ \cup \Gamma_1 \cup \Gamma_2$ ' and ' $x_2=0^- \cup \Gamma_3 \cup \Gamma_4$ ' where $\Gamma_1: \{x_2 > 0, x_1^2 + x_2^2 = 1.5^2\}$, $\Gamma_2: \{x_2 > 0, x_1^2 + x_2^2 = 0.5^2\}$, $\Gamma_3: \{x_2 < 0, x_1^2 + x_2^2 = 0.5^2\}$,

Chapter 4 Steady-State Anisotropic Heat Conduction Across a Homogeneously Imperfect Interfaces

$\Gamma_4: \{x_2 < 0, x_1^2 + x_2^2 = 1.5^2\}$. The thermal properties are $c_{11}^+ = 5/12$, $c_{12}^+ = 1/6$, $c_{22}^+ = 1/3$ for Ω^+ and $c_{11}^- = 1/2$, $c_{12}^- = 1/2$ and $c_{22}^- = 1$ for Ω^- . Condition (4.9) with $\beta = 1/3$ is applied at interface between Ω^+ and Ω^- .

Let the thermal field be given by

$$\begin{aligned} T(\mathbf{x}) &= \frac{1}{2} \exp\left(\frac{x_2}{3}\right) \cos\left(\frac{x_1}{3} - \frac{x_2}{6}\right) \text{ for } \mathbf{x} \in \Omega^+ \\ T(\mathbf{x}) &= \frac{1}{3} \exp\left(\frac{x_2}{6}\right) \cos\left(\frac{x_1}{3} - \frac{x_2}{6}\right) \text{ for } \mathbf{x} \in \Omega^- \end{aligned} \quad (4.20)$$

which satisfy (4.1), (4.4) and (4.9). To construct the problem, (4.20) is used to generate boundary values q on curve $\Gamma_1 \cup \Gamma_2$ and T on $\Gamma_3 \cup \Gamma_4$:

$$\begin{aligned} \text{on } \Gamma_1: q(\mathbf{x}) &= \frac{1}{54} [2x_1 \sin\left(\frac{x_1}{3} - \frac{x_2}{6}\right) - (x_1 + 2x_2) \cos\left(\frac{x_1}{3} - \frac{x_2}{6}\right)] \exp\left(\frac{x_2}{3}\right) \\ \text{on } \Gamma_2: q(\mathbf{x}) &= \frac{-1}{18} [2x_1 \sin\left(\frac{x_1}{3} - \frac{x_2}{6}\right) - (x_1 + 2x_2) \cos\left(\frac{x_1}{3} - \frac{x_2}{6}\right)] \exp\left(\frac{x_2}{3}\right) . \\ \text{on } \Gamma_3 \cup \Gamma_4: T(\mathbf{x}) &= \frac{1}{3} \exp\left(\frac{x_2}{6}\right) \cos\left(\frac{x_1}{3} - \frac{x_2}{6}\right) \end{aligned}$$

For the implementation of BEM, all the boundaries are discretised into M DLEs; 8 internal test points including 4 points close to interface are selected for computing and the results are tabulated in Table 4.3. From the table, one can see that the numerical results are again in good agreement with each other. The percentage errors reduce significantly as M is doubled from 48 to 96.

Since T and q are not specified on $\Gamma_1 \cup \Gamma_2$ and $\Gamma_3 \cup \Gamma_4$ respectively, we shall plot the corresponding exact and BEM values ($M=48$) at the boundary nodes in Figure 4.4 for comparisons. From Figure 4.4, one can see that the BEM results match very well with the analytical ones. Note that the nodes on each boundary are counted in this way: on Γ_1 , starts from point (1.5,0) in anti-clockwise direction; on Γ_2 , starts from point (-0.5,0) in clockwise direction; on Γ_3 , starts from point (0.5,0) in clockwise direction; and on Γ_4 , starts from point (-1.5,0) in anti-clockwise direction.

The corresponding computer code to solve the numerical problems in this chapter is found in Appendix B3.

Table 4.3: Results at interior points (*Problem 4.2*)

ξ	Exact $T(\xi)$	$M=48$		$M=96$	
		Computed	% error	computed	% error
$(-0.707,-0.707)$	0.2942210	0.2942242	1.0704E-03	0.2942217	2.3625E-04
$(0.707,-0.707)$	0.2779509	0.2779504	1.8164E-04	0.2779509	3.5129E-05
$(0.707,0.707)$	0.6285086	0.6284970	1.8486E-03	0.6285059	4.3297E-04
$(-0.707,0.707)$	0.5937529	0.5937401	2.1465E-03	0.5937501	4.5757E-04
$(-1.3,0.01)$	0.4549490	0.4549381	2.3930E-03	0.4549469	4.6008E-04
$(-0.8,-0.01)$	0.3211618	0.3211624	2.0456E-04	0.3211619	4.8945E-05
$(0.8,0.01)$	0.4841574	0.4841439	2.7910E-03	0.4841542	6.5554E-04
$(1.3,-0.01)$	0.3017866	0.3017826	1.3245E-03	0.3017858	2.8187E-04

Figure 4.4: Exact and computed T and q on boundary nodes with $M=48$. (*Problem 4.2*)

Chapter 5 Conclusions and Future Work

5.1 Summary and Conclusions

In Chapter 2, the task of numerically solving the heat conduction IBVPs in the non-homogeneous anisotropic media governed by the 2D parabolic PDE (1.15) is considered. The thermal coefficients take the form (1.21). With a suitable substitution, the PDE is initially recast into a form that favours the development of DRBEM for transforming the domain integral completely into a discretised BIE. By approximating variables inside the BIE using some polynomial functions in time, the equation can then be written in terms of the temperature and heat flux at selected nodes in space and time.

In Chapter 3, the DRBEM for solving homogeneous anisotropic non-Fourier heat flux model governed by PDE (1.24) has been successfully proposed. The variables in the BIE are also treated effectively by some modified polynomial time functions.

In Chapter 4, a special Green's function has been successfully derived and applied to the 2D steady-state heat conduction problem across imperfect interfaces of an anisotropic bimaterial without having to discretise the interface.

In all the chapters above, every BEM point is collocated to set up a system of algebraic equations to solve for unknowns. Temperature at any point and time (Chapter 2 and 3) inside the domain can then be computed, once the linear algebraic equations are solved.

To verify the proposed methods, numerous problems with different domains and initial-boundary conditions including mixed and Neumann (Chapter 2 and 3) conditions have been solved. The BEM results show good accuracy and convergence as the number of boundary elements (M) and internal collocation points (K_{Ω} for Chapter 2 and 3) are increased.

From the numerical results obtained in Chapter 2 and 3, we can conclude that the polynomial time function used in TM approach can deal effectively with the non-steady state IBVPs even for the problems with highly fluctuating boundary values. The optimum number of the time nodes (K_t) is found to be at least 3 or 4, and it is best to collocate the time exactly at each time node (i.e. $\xi=0$) when solving for unknowns.

Compared with the central difference formula (2.24), the system using polynomial approximation remains stable after many numbers of solution steps, i.e. errors do not appear to 'blow-up' easily. Another advantage of the polynomial approximation over (2.24) is that such approximation can be further applied to higher order time-dependent PDEs.

The major advantage of the TM scheme in time domain approach over the transform space approach (such as LT) is that the solution is obtained directly in the physical domain, no inversion of integral transform is required. However, this approach can be numerically inefficient. Finding the solution at a particular time is possible only if all the unknowns at preceding time levels are determined. Thus, this causes more computational time and computer memory being used.

5.2 Recommendations for Future Work

Having been successful in employing the BEM for solving heat conduction problems, it would be necessary and interesting to seek further extension of the BEM to some more challenging cases as part of future work.

In the present report, the problems considered are just limited to 2D cases. Therefore, it would be quite challenging when the problems considered in Chapter 2 and 3 are posed as 3D problems. The discretisation of the boundary and integration involved would be more complicated than the 2D one.

Apart from that, the problem considered in Chapter 2 may also be extended to the case in which the thermal coefficients are also temperature-dependent taking the form $c_{ij}(\mathbf{x}) = c_{ij}^{(o)} g_1(\mathbf{x}) g_2(T)$, where g_2 is a known function of temperature. In this case the properties of material not just vary from space to space, but are also affected by temperature. This leads to a non-linear PDE to solve. For the steady-state case, the non-linear PDE can be easily transformed to a linear one by the so-called Kirchoff transformation [18].

The problem of non-classical Fourier's heat flux model governed by PDE (1.24) considered in Chapter 3 may be extended to inhomogeneous case where the coefficient taking the form (1.21). In such an extension, the problem presently considered in Chapter 2 (where all time lags of temperature and heat flux are zeros) will become the special case of the above mentioned.

As in the case in Chapter 4, it would be quite interesting to extend it to the non-steady state condition. In such case, the DRM together with the special Green's function derived in Chapter 4 will be employed to derive the required BIE.

References

- [1] Abramowitz M, Stegun IA, *Handbook of Mathematical Functions*, Dover Publications, New York 1972.
- [2] Alan Jeffrey, *Handbook of Mathematical Formulas and Integrals*, Academic Press 1995.
- [3] Ames WF, *Numerical Methods for Partial Differential Equations*, Academic Press, New York, 1977.
- [4] Ang WT, A crack in an anisotropic layered material under the action of impact loading, *ASME Journal of Applied Mechanics*, 1988; Vol. 55: 122-125.
- [5] Ang WT, A Laplace transform dual-reciprocity boundary element method for a class of two-dimensional microscale thermal problems, *Engineering Computations*, 2002; Vol.19: 467-478.
- [6] Ang WT, Ang KC, Dehghan M, The determination of control parameter in a two-dimensional diffusion equation using a dual-reciprocity boundary element method, *International Journal of Computer Mathematics*, 2003; Vol. 80: 65-74.
- [7] Ang WT, Clements DL, Nader Vahdati., A dual-reciprocity boundary element method for a class of elliptic boundary value problems for nonhomogeneous anisotropic media, *Engineering Analysis with Boundary Elements*, 2003; Vol. 27: 49-55.
- [8] Ang WT, The two-dimensional reaction-diffusion Brusselator system: a dual-reciprocity boundary element solution, *Engineering Analysis with Boundary Elements*, 2003; Vol. 27: 897-903.
- [9] Ang WT, Choo KK and Fan H, A Green's function for steady-state two-dimensional isotropic heat conduction across a homogeneously imperfect interface, *Communications In Numerical Methods In Engineering*, 2004; Vol .20: 391 –399.
- [10] Ang WT, Choo KK, A dual-reciprocity boundary element method for anisotropic heat conduction in inhomogeneous solids, *Proceedings of the 1st International Conference "From Scientific Computing to Computational Engineering"*, University of Patras, Athens, Greece 8-10 September, 2004.
- [11] Berger JR, Karageorghis A., The method of fundamental solutions for heat conduction in layered materials, *International Journal for Numerical Methods in Engineering* 1999; 45:1681-1694.
- [12] Brebbia CA, Nardini D, A new approach to free vibration analysis using boundary elements, in *Boundary Element Methods in Engineering*, Brebbia CA (Ed.), Springer-Verlag, Berlin, 1982.

References

- [13] Carslaw HS, Jaeger JC, *Conduction of Heat in Solids*, Oxford University Press, London, 1959.
  - [14] Chandrupatla TR, Belegunda AD, *Introduction to Finite Elements in Engineering* (2nd Edition), Prentice Hall, USA, 1997.
  - [15] Chang YP, Kang CS, Chen DJ, The use of fundamental Green's functions for the solution of heat conduction in anisotropic media,, *International Journal of Heat and Mass Transfer*, 1973; Vol. 16: 1905-1918.
  - [16] Chen CS, Brebbia CA, Power H, Dual reciprocity method using compactly supported radial basis function, *Communications in Numerical Methods in Engineering*, 1999; Vol. 15: 137-150.
  - [17] Clements DL, A crack in anisotropic layered materials, *Rozprawy Inzynierskie*, 1979; Vol.27: 171-180.
  - [18] Clements DL, Budhi WS, A boundary element method for the solution of a class of steady-state problems for anisotropic media *Journal of Heat Transfer-Transactions of the ASME*, May. 1999; Vol.121 (2): 462-465.
  - [19] Dai W, Nassar R, A finite difference scheme for solving the heat transport equation at the microscale, *Numerical Method for Partial Differential Equations*, 1999; Vol 15: 697-708.
  - [20] El-Zafrany A, *Techniques of The Boundary Element Method*, Ellis Horwood Limited, 1993.
  - [21] Erdelyi A, Magnus M, Oberhettinger F, Tricomi F., *Tables of Integral Transforms*, Volume1, MacGraw-Hill, 1954.
  - [22] Golberg MA, Chen CS, Karur S, Improved multiquadric approximation for partial differential equations, *Boundary Element Communication*, 1994; Vol 5: 57-61.
  - [23] Hartmann F, *Introduction to Boundary Elements: Theory and Applications*, Springer-verlag, Berlin, 1989.
  - [24] Hoskins RF, *Delta Functions: Introduction to Generalised Functions*, Horwood Series in Mathematics & Applications, 1999.
  - [25] Kassab AJ, Divo E, A generalised boundary integral equation for isotropic heat conduction equation with spatially varying thermal conductivity, *Engineering Analysis with Boundary Elements*, 1996; Vol 18: 273-286.
  - [26] Kreyszig E, *Advanced Engineering Mathematics* (8th Edition), John Willey & Sons, 1999.
  - [27] Manzari MT, Manzari MT, A mixed approach to finite-element analysis of hyperbolic heat conduction problems, *International Journal for Numerical Methods in Heat and Fluid Flow*, 1998; Vol 8: 83-97.
-

References

-
- [28] Nowak AJ, The multiple reciprocity method of solving transient heat conduction problem, *Boundary element XI, vol.2, Computational Mechanics Publication, Southampton and Springer-Verlag*, Berlin and New York, 1989.
  - [29] París F, Cañas J, *Boundary Element Method: Fundamentals and Applications*, Oxford University Press, Oxford, 1997.
  - [30] Partridge PW, Brebbia CA, Wrobel LC, *The Dual Reciprocity Boundary Element Method, Computational Mechanics Publications-Elsevier Applied Science*, 1992.
  - [31] Paulino GH, Gray LJ, Sutradhar A, Transient heat conduction in homogeneous and non-homogeneous materials by the Laplace transform Galerkin boundary element method, *Engineering Analysis with Boundary Elements*, 2002; Vol. 26: 119-132.
  - [32] Prem K. Kythe, *Fundamental Solutions for Differential Operators and Applications*, Birkhauser, Boston 1996.
  - [33] Rizzo F, Shippy DJA, Method of solution for certain problems of transient heat conduction. *AIAA Journal*, 1970; Vol. 8: 2004–2009.
  - [34] Roach G.F, *Green's Function*, 2nd edition, Cambridge University Press, 1982.
  - [35] Tanaka M, Matsumoto T, Suda Y, A dual-reciprocity boundary element method applied to the steady-state heat conduction problem of functionally gradient materials. Presented at *BETEQ 2001-The 2nd International Conference on Boundary Element Techniques*, Rutgers University, USA, 2001.
  - [36] Tang W, Brebbia CA, Telles JCF, A generalised approach to transfer the domain integrals onto boundary ones for potential problem in BEM, *Proc. BEM IX*, 1987, pp.99-116.
  - [37] Tzou DY, Experimental support for the lagging behavior in heat propagation, *AIAAJ Thermophysic Heat Transfer*, 1995; Vol. 9(4): 686-693.
  - [38] Tzou DY, A unified field approach for heat conduction from micro- to macro-scales, *ASME Journal of Heat Transfer*, 1995; Vol. 117: 8-16.
  - [39] Tzou DY, *Macro-to Microscale Heat Transfer, The Lagging Behaviour*, Taylor and Francis, Bristol, 1997.
  - [40] Wu Z, Multivariate compactly supported positive definite radial functions, *Advanced in Computational Mathematics*, 1995; Vol. 4:283-292.
  - [41] Zhang Y, Zhu S, On the choice of interpolation functions used in the dual-reciprocity boundary-element method, *Engineering Analysis with Boundary Elements*, 1994; Vol.13:387-396.
-

Appendix A

A1. Analytical Formulae for Some Definite Integrals

Notes:

The subscript '4q' in $\tan^{-1}(Y,X)_{4q}$ represents 4-quadrant inverse tangent applied to $\tan^{-1}(Y/X)$. If $Y \geq 0$ then $0 \leq \tan^{-1}(Y,X)_{4q} \leq \pi$, else $-\pi < \tan^{-1}(Y,X)_{4q} \leq 0$.

$$\int_{-1}^1 \ln(u^2 + Au + B) du$$

$$= \begin{cases} \text{For } 4B-A^2 > 0 \\ \left[\left(u + \frac{A}{2} \right) \ln(u^2 + Au + B) - 2u + \sqrt{4B-A^2} \tan^{-1} \left(\frac{2u+A}{\sqrt{4B-A^2}} \right) \right]_{-1}^1 \\ = \left(1 + \frac{A}{2} \right) \ln(1+A+B) + \left(1 - \frac{A}{2} \right) \ln(1-A+B) - 4 + \sqrt{4B-A^2} \tan^{-1}(\sqrt{4B-A^2}, B-1)_{4q} \\ \text{For } 4B-A^2 = 0 \\ \left[\left(u + \frac{A}{2} \right) \ln \left(u + \frac{A}{2} \right)^2 - 2u \right]_{-1}^1 \\ = \frac{(2+A)}{2} \ln(2+A)^2 + \frac{(2-A)}{2} \ln(2-A)^2 - 4(\ln 2 + 1) = 4 \ln 2 - 4 \text{ if } A = \pm 2 \end{cases}$$

$$\int_{-1}^1 u \ln(u^2 + Au + B) du$$

$$= \begin{cases} \text{For } 4B-A^2 > 0 \\ \left[\frac{1}{4} (2u^2 - A^2 + 2B) \ln(u^2 + Au + B) + \frac{u}{2} (A-u) - \frac{A}{2} \sqrt{4B-A^2} \tan^{-1} \left(\frac{2u+A}{\sqrt{4B-A^2}} \right) \right]_{-1}^1 \\ = \frac{1}{4} (2-A^2 + 2B) \ln \left(\frac{1+A+B}{1-A+B} \right) + A - \frac{A}{2} \sqrt{4B-A^2} \tan^{-1}(\sqrt{4B-A^2}, B-1)_{4q} \\ \text{For } 4B-A^2 = 0 \\ \left[\frac{1}{2} \left(u^2 - \frac{A^2}{4} \right) \ln \left(u + \frac{A}{2} \right)^2 - u(u-A) \right]_{-1}^1 = \frac{1}{8} (4-A^2) \ln \left(\frac{2+A}{2-A} \right)^2 + A = \pm 2 \text{ if } A = \pm 2 \end{cases}$$

Appendix A

$$\int_{-1}^1 \frac{du}{u^2 + Au + B} = \begin{cases} \text{For } 4B - A^2 > 0 \\ \left[\frac{2}{\sqrt{4B - A^2}} \tan^{-1} \left(\frac{2u + A}{\sqrt{4B - A^2}} \right) \right]_{-1}^1 = \frac{2}{\sqrt{4B - A^2}} \tan^{-1}(\sqrt{4B - A^2}, B - 1)_{4q} \\ \text{For } 4B - A^2 = 0 \\ \left[\frac{-2}{2u + A} \right]_{-1}^1 = \frac{8}{A^2 - 4} \end{cases}$$

$$\int_{-1}^1 \frac{u}{u^2 + Au + B} du = \begin{cases} \text{For } 4B - A^2 > 0 \\ \frac{1}{2} \left[\ln(u^2 + Au + B) - \frac{2A}{\sqrt{4B - A^2}} \tan^{-1} \left(\frac{2u + A}{\sqrt{4B - A^2}} \right) \right]_{-1}^1 \\ = \frac{1}{2} \ln \left(\frac{1 + A + B}{1 - A + B} \right) - \frac{A}{\sqrt{4B - A^2}} \tan^{-1}(\sqrt{4B - A^2}, B - 1)_{4q} \\ \text{For } 4B - A^2 = 0 \\ \left[\ln \left(u + \frac{A}{2} \right) + \frac{A}{3u + A} \right]_{-1}^1 = \ln \left(\frac{A + 2}{A - 2} \right) - \frac{4A}{A^2 - 4} \end{cases}$$

Exponential Integral Ei(z)

$$\text{Ei}(z) = \int_z^\infty \frac{e^{-w}}{w} dw = -\gamma - \ln z - \sum_{n=1}^{\infty} \frac{(-1)^n z^n}{n \cdot n!}, \quad (|\text{Arg}(z)| < \pi, \text{ i.e. } z \neq 0 \cup \text{'negative real'})$$

$$\text{where } \gamma = \lim_{s \rightarrow \infty} \left[\sum_{n=1}^s \frac{1}{n} - \ln(s) \right] = \lim_{s \rightarrow \infty} \left[1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{s} - \ln(s) \right] = 0.5772 \, 15664 \, 90153 \dots \quad \text{which is}$$

known in the literature as Euler's constant.

$$\int_0^\infty \frac{e^{-zw}}{a + wb} dw = \frac{1}{b} e^{az/b} \text{Ei} \left(\frac{az}{b} \right), \quad (\text{Re}(z) \geq 0) \quad (\text{A-1})$$

$$\int_0^\infty \frac{e^{-z_1 w} - e^{-z_2 w}}{w} dw = \ln \left(\frac{z_2}{z_1} \right), \quad (\text{Re}(z_1) \geq 0 \text{ and } \text{Re}(z_2) \geq 0) \quad (\text{A-2})$$

$$\int \frac{e^{-zw}}{w^2} dw = -\frac{e^{-zw}}{w} - z \int \frac{e^{-zw}}{w} dw, \quad (\text{Re}(z) \geq 0) \quad (\text{A-3})$$

$$\lim_{w \rightarrow 0} \left(\frac{e^{z_1 w} - e^{z_2 w}}{w} \right) = z_1 - z_2. \quad (\text{A-4})$$

The above formulas and more other related definite, indefinite integrals and limits could be found in [1] and [2]. Numerical values of Ei( ) could be found from [1].

A2. Integration Parameters for Chapter 4

A2.1 Explicit Formulae for $I_*^{(k)}(\xi)$ and $II_*^{(k)}(\xi)$

$$I_*^{(k)}(\xi) = \frac{L^{(\bar{k})}}{2} \int_{I^{(\bar{k})}} \frac{1-2l_{\hat{k}}-(-1)^{\hat{k}}u}{2(1-l_1-l_2)} \phi_*^{\langle x^{(k)} \rangle}(x(u), \xi) du = C_*^{(k)}(\xi) \int_{-1}^1 \frac{1-2l_{\hat{k}}-(-1)^{\hat{k}}u}{u^2 + A_*^{(k)}(\xi)u + B_*^{(k)}(\xi)} du$$

$$\begin{aligned} II_*^{(k)}(\xi) &= \frac{L^{(\bar{k})}}{2} \int_{I^{(\bar{k})}} \frac{1-2l_{\hat{k}}-(-1)^{\hat{k}}u}{2(1-l_1-l_2)} \Phi_*^{\langle x^{(k)} \rangle}(x(u), \xi) du \\ &= 2(1-2l_{\hat{k}})D_*^{(k)} + E_*^{(k)} \int_{-1}^1 (1-2l_{\hat{k}}-(-1)^{\hat{k}}u) \ln(u^2 + A_*^{(k)}(\xi)u + B_*^{(k)}(\xi)) du \end{aligned}$$

where the coefficients

$$A_*^{(k)}(\xi) = \frac{4c_{22}^{\langle x^{(k)} \rangle}}{L^{(\bar{k})}} \frac{\left(\frac{c_{22}^{\langle x^{(k)} \rangle}}{c_{22}^{\langle \xi \rangle}} \tilde{x}_2^{(\bar{k})} - \frac{c_{22}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) c^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} - \left(\tilde{x}_1^{(\bar{k})} - \xi_1 - \frac{c_{12}^{\langle x^{(k)} \rangle}}{c_{22}^{\langle x^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \frac{c_{12}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) (c_{12}^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle x^{(k)} \rangle} n_2^{(\bar{k})})}{(c_{12}^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle x^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle x^{(k)} \rangle} n_1^{(\bar{k})})^2}$$

$$B_*^{(k)}(\xi) = \left(\frac{2c_{22}^{\langle x^{(k)} \rangle}}{L^{(\bar{k})}} \right)^2 \frac{\left(\frac{c_{22}^{\langle x^{(k)} \rangle}}{c_{22}^{\langle \xi \rangle}} \tilde{x}_2^{(\bar{k})} - \frac{c_{22}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right)^2 + \left(\tilde{x}_1^{(\bar{k})} - \xi_1 - \frac{c_{12}^{\langle x^{(k)} \rangle}}{c_{22}^{\langle x^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \frac{c_{12}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right)^2}{(c_{12}^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle x^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle x^{(k)} \rangle} n_1^{(\bar{k})})^2} \geq 0,$$

$$\begin{aligned} C_*^{(k)}(\xi) &= \frac{c_{22}^{\langle x^{(k)} \rangle}}{2\pi L^{(\bar{k})}(1-l_1-l_2)} \\ &\times \frac{\left(\frac{c_{22}^{\langle x^{(k)} \rangle}}{c_{22}^{\langle \xi \rangle}} \tilde{x}_2^{(\bar{k})} - \frac{c_{22}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) (c_{12}^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle x^{(k)} \rangle} n_2^{(\bar{k})}) + \left(\tilde{x}_1^{(\bar{k})} - \xi_1 - \frac{c_{12}^{\langle x^{(k)} \rangle}}{c_{22}^{\langle x^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \frac{c_{12}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) c^{\langle x^{(k)} \rangle} n_1^{(\bar{k})}}{(c_{12}^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle x^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle x^{(k)} \rangle} n_1^{(\bar{k})})^2} \end{aligned}$$

$$D_*^{(k)} = E_*^{(k)} \ln \left(\left(\frac{L^{(\bar{k})}}{2c_{22}^{\langle x^{(k)} \rangle}} \right)^2 \left[(c_{12}^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle x^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle x^{(k)} \rangle} n_1^{(\bar{k})})^2 \right] \right),$$

$$E_*^{(k)} = \frac{L^{(\bar{k})}}{16\pi(1-l_1-l_2)c^{\langle x^{(k)} \rangle}} > 0,$$

$$\text{It can be shown that } 4B_*^{(k)} - (A_*^{(k)})^2 = [8\pi(1-l_1-l_2)C_*^{(k)}]^2 \geq 0.$$

The analytical solutions to the above integration parameters $I_*^{(k)}$, $II_*^{(k)}$ and the ones in following section A2.2 can be known from Appendix A1.

Appendix A

A2.2 Explicit Formulae for $I_{\text{per}}^{(k)}(\xi)$ and $II_{\text{per}}^{(k)}(\xi)$

$$I_{\text{per}}^{(k)}(\xi) = \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1-2l_{\bar{k}}-(-1)^{\hat{k}}u}{2(1-l_1-l_2)} \phi_{\text{per}}^{\langle \mathbf{x}^{(k)} \rangle}(\mathbf{x}(u), \xi) du = C_{\text{per}}^{(k)}(\xi) \int_{-1}^1 \frac{1-2l_{\bar{k}}-(-1)^{\hat{k}}u}{u^2 + A_{\text{per}}^{(k)}(\xi)u + B_{\text{per}}^{(k)}(\xi)} du$$

$$\begin{aligned} II_{\text{per}}^{(k)}(\xi) &= \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1-2l_{\bar{k}}-(-1)^{\hat{k}}u}{2(1-l_1-l_2)} \Phi_{\text{per}}^{\langle \mathbf{x}^{(k)} \rangle}(\mathbf{x}(u), \xi) du \\ &= 2(1-2l_{\bar{k}})D_{\text{per}}^{(k)} + E_{\text{per}}^{(k)} \int_{-1}^1 (1-2l_{\bar{k}}-(-1)^{\hat{k}}u) \ln(u^2 + A_{\text{per}}^{(k)}(\xi)u + B_{\text{per}}^{(k)}(\xi)) du \end{aligned}$$

where the coefficients

$$\begin{aligned} A_{\text{per}}^{(k)}(\xi) &= \frac{4c_{22}^{\langle \mathbf{x}^{(k)} \rangle}}{L^{(\bar{k})}} \frac{[\frac{c^{\langle \mathbf{x}^{(k)} \rangle}}{c_{22}^{\langle \mathbf{x}^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \langle \mathbf{x}^{(k)} \rangle, \xi \frac{c^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2] c^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} - [\tilde{x}_1^{(\bar{k})} - \xi_1 - \frac{c_{12}^{\langle \mathbf{x}^{(k)} \rangle}}{c_{22}^{\langle \mathbf{x}^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \frac{c_{12}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2] [c_{12}^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle \mathbf{x}^{(k)} \rangle} n_2^{(\bar{k})}]}{(c_{12}^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle \mathbf{x}^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})})^2} \\ B_{\text{per}}^{(k)}(\xi) &= \left(\frac{2c_{22}^{\langle \mathbf{x}^{(k)} \rangle}}{L^{(\bar{k})}} \right)^2 \frac{\left(\frac{c^{\langle \mathbf{x}^{(k)} \rangle}}{c_{22}^{\langle \mathbf{x}^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \langle \mathbf{x}^{(k)} \rangle, \xi \frac{c^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right)^2 + \left(\tilde{x}_1^{(\bar{k})} - \xi_1 - \frac{c_{12}^{\langle \mathbf{x}^{(k)} \rangle}}{c_{22}^{\langle \mathbf{x}^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \frac{c_{12}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right)^2}{(c_{12}^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle \mathbf{x}^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})})^2} \geq 0, \\ C_{\text{per}}^{(k)}(\xi) &= \frac{c_{22}^{\langle \mathbf{x}^{(k)} \rangle} (c^{\langle \mathbf{x}^{(k)} \rangle} - \overline{c^{\langle \mathbf{x}^{(k)} \rangle}})}{2\pi L^{(\bar{k})} (1-l_1-l_2) (c^+ + c^-)} \\ &\quad \times \frac{\left(\frac{c^{\langle \mathbf{x}^{(k)} \rangle}}{c_{22}^{\langle \mathbf{x}^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \langle \mathbf{x}^{(k)} \rangle, \xi \frac{c^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) (c_{12}^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle \mathbf{x}^{(k)} \rangle} n_2^{(\bar{k})}) + \left(\tilde{x}_1^{(\bar{k})} - \xi_1 - \frac{c_{12}^{\langle \mathbf{x}^{(k)} \rangle}}{c_{22}^{\langle \mathbf{x}^{(k)} \rangle}} \tilde{x}_2^{(\bar{k})} + \frac{c_{12}^{\langle \xi \rangle}}{c_{22}^{\langle \xi \rangle}} \xi_2 \right) c^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})}}{(c_{12}^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle \mathbf{x}^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})})^2} \\ D_{\text{per}}^{(k)} &= E_{\text{per}}^{(k)} \ln \left(\left(\frac{L^{(\bar{k})}}{2c_{22}^{\langle \mathbf{x}^{(k)} \rangle}} \right)^2 \left[(c_{12}^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} + c_{22}^{\langle \mathbf{x}^{(k)} \rangle} n_2^{(\bar{k})})^2 + (c^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})})^2 \right] \right), \\ E_{\text{per}}^{(k)} &= \frac{L^{(\bar{k})} (c^{\langle \mathbf{x}^{(k)} \rangle} - \overline{c^{\langle \mathbf{x}^{(k)} \rangle}})}{16\pi c^{\langle \mathbf{x}^{(k)} \rangle} (1-l_1-l_2) (c^+ + c^-)}, \end{aligned}$$

with $\langle \mathbf{x}^{(k)}, \xi \rangle = \langle \mathbf{x}^{(k)} \rangle \langle \xi \rangle$: if both $\mathbf{x}^{(k)}$ and ξ reside on the same material domain, i.e. either $\Omega^+ \cup \Gamma^+$ or $\Omega^- \cup \Gamma^-$ then $\langle \mathbf{x}^{(k)}, \xi \rangle = 1$, otherwise $\langle \mathbf{x}^{(k)}, \xi \rangle = -1$. $\overline{\langle \mathbf{x}^{(k)} \rangle} = \mp \text{sign}$ if $\mathbf{x}^{(k)} \in \Omega^\pm \cup \Gamma^\pm$. It can be shown that

$$4B_{\text{per}}^{(k)} - (A_{\text{per}}^{(k)})^2 = \left[\frac{8\pi(1-l_1-l_2)(c^+ + c^-)}{c^{\langle \mathbf{x}^{(k)} \rangle} - \overline{c^{\langle \mathbf{x}^{(k)} \rangle}}} C_{\text{per}}^{(k)} \right]^2 \geq 0.$$

Appendix A

A2.3 Analytical Formulae for $I_{\text{imp}}^{(k)}(\xi)$ and $II_{\text{imp}}^{(k)}(\xi)$

Formulae for $I_{\text{imp}}^{(k)}(\xi)$

$$\begin{aligned} I_{\text{imp}}^{(k)}(\xi) &= \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1-2l_{\hat{k}}-(-1)^{\hat{k}}u}{2(1-l_1-l_2)} \phi_{\text{imp}}^{\langle \mathbf{x}^{(k)} \rangle}(\mathbf{x}(u), \xi) du \\ &= \frac{\langle \mathbf{x}^{(k)} \rangle L^{(\bar{k})}}{8\pi(1-l_1-l_2)} \text{Re} \int_0^\infty \left\{ \frac{iw(c_{1j}^{\langle \mathbf{x}^{(k)} \rangle} n_j^{(\bar{k})} + \kappa^{\langle \mathbf{x}^{(k)} \rangle} c_{2j}^{\langle \mathbf{x}^{(k)} \rangle} n_j^{(\bar{k})}) Q^{\langle \mathbf{x}^{(k)} \rangle}}{-[wL^{(\bar{k})}(\kappa^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})})/2]^2} \right. \\ &\quad \times \left[\langle \mathbf{x}^{(k)} \rangle \frac{1}{2} iwL^{(\bar{k})} [\kappa^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}] [1-2l_{\hat{k}}-(-1)^{\hat{k}}u] + (-1)^{\hat{k}} \right) \\ &\quad \times \exp \left(\langle \mathbf{x}^{(k)} \rangle iw \left[\frac{L^{(\bar{k})}}{2} (\kappa^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}) u + \tilde{x}_1^{(\bar{k})} + \kappa^{\langle \mathbf{x}^{(k)} \rangle} \tilde{x}_2^{(\bar{k})} \right] \right) \Bigg]_{-1}^1 \Bigg\} dw \end{aligned}$$

From above, the superscript ' $\pm$ ' sign of $Q^\pm$ depends on $\mathbf{x}^{(k)}$, so we can rewrite (4.17) in the following form

$$Q^{\langle \mathbf{x}^{(k)} \rangle} = \frac{-\langle \mathbf{x}^{(k)} \rangle, \xi \rangle 2c^{\langle \xi \rangle}}{(c^+ + c^-)c^{\langle \mathbf{x}^{(k)} \rangle}} \frac{\exp(\langle \mathbf{x}^{(k)} \rangle iw \{ \xi_1 - [c_{12}^{\langle \xi \rangle} + \langle \mathbf{x}^{(k)} \rangle, \xi \rangle ic^{\langle \xi \rangle}] \xi_2 / c_{22}^{\langle \xi \rangle} \})}{\beta(1/c^+ + 1/c^-) + w} \quad (\text{A-5})$$

where $\overline{\langle \xi \rangle} = \mp$ sign' if $\xi \in \Omega^\pm \cup \Gamma^\pm$.

By substituting (A-5) and rearranging it, we yield

$$I_{\text{imp}}^{(k)}(\xi) = \text{Re} \left\{ D_{\text{imp}}^{(k)} \int_0^\infty \frac{[wA_{\text{imp}(1)}^{(k)} + (-1)^{\hat{k}}] \exp(wB_{\text{imp}(1)}^{(k)}) - [wA_{\text{imp}(2)}^{(k)} + (-1)^{\hat{k}}] \exp(wB_{\text{imp}(2)}^{(k)})}{w[C_{\text{imp}}^{(k)} + wc^{\langle \mathbf{x}^{(k)} \rangle}]} dw \right\} \quad (\text{A-6})$$

where the coefficients

$$A_{\text{imp}(j)}^{(k)} = \langle \mathbf{x}^{(k)} \rangle \frac{1}{2} iL^{(\bar{k})} [\kappa^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}] [1-2l_{\hat{k}} + (-1)^{j+\hat{k}}], \quad (j=1, 2)$$

$$B_{\text{imp}(j)}^{(k)} = \langle \mathbf{x}^{(k)} \rangle i \left(\tilde{x}_1^{(\bar{k})} + \kappa^{\langle \mathbf{x}^{(k)} \rangle} \tilde{x}_2^{(\bar{k})} - \xi_1 + [c_{12}^{\langle \xi \rangle} + i \langle \mathbf{x}^{(k)} \rangle, \xi \rangle c^{\langle \xi \rangle}] \frac{\xi_2}{c_{22}^{\langle \xi \rangle}} - (-1)^j \frac{L^{(\bar{k})}}{2} [\kappa^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}] \right)$$

$$C_{\text{imp}}^{(k)} = \beta c^{\langle \mathbf{x}^{(k)} \rangle} (1/c^+ + 1/c^-) > 0,$$

$$D_{\text{imp}}^{(k)} = \frac{\langle \xi \rangle i c^{\langle \xi \rangle} [c_{1j}^{\langle \mathbf{x}^{(k)} \rangle} n_j^{(\bar{k})} + \kappa^{\langle \mathbf{x}^{(k)} \rangle} c_{2j}^{\langle \mathbf{x}^{(k)} \rangle} n_j^{(\bar{k})}]}{\pi L^{(\bar{k})} (1-l_1-l_2) (c^+ + c^-) [\kappa^{\langle \mathbf{x}^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}]^2}.$$

Note that $\text{Re}(B_{\text{imp}(j)}^{(k)}) < 0$.

By expanding (A-6), we obtain

Appendix A

$$I_{\text{imp}}^{(k)}(\xi) = \text{Re} \left\{ D_{\text{imp}}^{(k)} \left[A_{\text{imp}(1)}^{(k)} - (-1)^{\hat{k}} \frac{c^{\langle x^{(k)} \rangle}}{C_{\text{imp}}^{(k)}} \int_0^\infty \frac{\exp(w B_{\text{imp}(1)}^{(k)})}{C_{\text{imp}}^{(k)} + w c^{\langle x^{(k)} \rangle}} dw - \left[A_{\text{imp}(2)}^{(k)} - (-1)^{\hat{k}} \frac{c^{\langle x^{(k)} \rangle}}{C_{\text{imp}}^{(k)}} \int_0^\infty \frac{\exp(w B_{\text{imp}(2)}^{(k)})}{C_{\text{imp}}^{(k)} + w c^{\langle x^{(k)} \rangle}} dw \right. \right. \right. \\ \left. \left. \left. + (-1)^{\hat{k}} \int_0^\infty \frac{\exp(w B_{\text{imp}(1)}^{(k)}) - \exp(w B_{\text{imp}(2)}^{(k)})}{w} dw \right] \right\}$$

Using the results (A-1) and (A-2), we can obtain the required analytical solution from above equation

$$I_{\text{imp}}^{(k)}(\xi) = \text{Re} \left\{ D_{\text{imp}}^{(k)} \left[\frac{1}{c^{\langle x^{(k)} \rangle}} A_{\text{imp}(1)}^{(k)} - \frac{(-1)^{\hat{k}}}{C_{\text{imp}}^{(k)}} \exp \left(\frac{-B_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) \text{Ei} \left(\frac{-B_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) \right. \right. \\ \left. \left. - \left[\frac{1}{c^{\langle x^{(k)} \rangle}} A_{\text{imp}(2)}^{(k)} - \frac{(-1)^{\hat{k}}}{C_{\text{imp}}^{(k)}} \exp \left(\frac{-B_{\text{imp}(2)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) \text{Ei} \left(\frac{-B_{\text{imp}(2)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) + \frac{(-1)^{\hat{k}}}{C_{\text{imp}}^{(k)}} \ln \left(\frac{B_{\text{imp}(2)}^{(k)}}{B_{\text{imp}(1)}^{(k)}} \right) \right] \right\}$$

Formulae for $I_{\text{imp}}^{(k)}(\xi)$

For the second integration parameter $I_{\text{imp}}^{(k)}(\xi)$, we use the same procedure as before, then

$$I_{\text{imp}}^{(k)}(\xi) = \frac{L^{(\bar{k})}}{2} \int_{\Gamma^{(\bar{k})}} \frac{1 - 2l_{\hat{k}} - (-1)^{\hat{k}} u}{2(1 - l_1 - l_2)} \Phi_{\text{imp}}^{\langle x^{(k)} \rangle}(x(u), \xi) du \\ = \frac{L^{(\bar{k})}}{8\pi(1 - l_1 - l_2)} \\ \times \text{Re} \int_0^\infty \left\{ Q^{\langle x^{(k)} \rangle} \int_{-1}^1 (1 - 2l_{\hat{k}} - (-1)^{\hat{k}} u) \exp \left(\langle x^{(k)} \rangle i w \left[\frac{L^{(\bar{k})}}{2} (\kappa^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}) u + \tilde{x}_1^{(\bar{k})} + \kappa^{\langle x^{(k)} \rangle} \tilde{x}_2^{(\bar{k})} \right] \right) du \right\} dw \\ = \frac{L^{(\bar{k})}}{8\pi(1 - l_1 - l_2)} \text{Re} \int_0^\infty \left\{ \frac{Q^{\langle x^{(k)} \rangle}}{-[w L^{(\bar{k})} (\kappa^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}) / 2]^2} \right. \\ \times \left[\langle x^{(k)} \rangle \frac{1}{2} i w L^{(\bar{k})} [\kappa^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}] [1 - 2l_{\hat{k}} - (-1)^{\hat{k}} u] + (-1)^{\hat{k}} \right) \\ \left. \times \exp \left(\langle x^{(k)} \rangle i w \left[\frac{L^{(\bar{k})}}{2} (\kappa^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}) u + \tilde{x}_1^{(\bar{k})} + \kappa^{\langle x^{(k)} \rangle} \tilde{x}_2^{(\bar{k})} \right] \right) \right]_{-1}^1 \right\} dw$$

Substituting (A-5) into the above equation and rearranging it, we obtain

$$I_{\text{imp}}^{(k)}(\xi) = \text{Re} \left\{ E_{\text{imp}}^{(k)} \int_0^\infty \frac{[w A_{\text{imp}(1)}^{(k)} + (-1)^{\hat{k}}] \exp(w B_{\text{imp}(1)}^{(k)}) - [w A_{\text{imp}(2)}^{(k)} + (-1)^{\hat{k}}] \exp(w B_{\text{imp}(2)}^{(k)})}{w^2 [C_{\text{imp}}^{(k)} + w c^{\langle x^{(k)} \rangle}]} dw \right\}$$

$$\text{where } E_{\text{imp}}^{(k)} = \frac{\langle x^{(k)}, \xi \rangle c^{\langle \xi \rangle}}{\pi L^{(\bar{k})} (1 - l_1 - l_2) (c^+ + c^-) [\kappa^{\langle x^{(k)} \rangle} n_1^{(\bar{k})} - n_2^{(\bar{k})}]^2}.$$

Appendix A

Expanding the fractional terms in above equation and using (A-3), we can obtain

$$\begin{aligned}
 H_{\text{imp}}^{(k)}(\xi) = & \text{Re} \left\{ E_{\text{imp}}^{(k)} \left[\frac{(-1)^{\hat{k}}}{C_{\text{imp}}^{(k)}} \left[\frac{\exp(w B_{\text{imp}(2)}^{(k)}) - \exp(w B_{\text{imp}(1)}^{(k)})}{w} \right]_0^{\infty} \right. \right. \\
 & + \frac{[A_{\text{imp}(1)}^{(k)} + (-1)^{\hat{k}} B_{\text{imp}(1)}^{(k)}] - [A_{\text{imp}(2)}^{(k)} + (-1)^{\hat{k}} B_{\text{imp}(2)}^{(k)}]}{C_{\text{imp}}^{(k)}} \int_0^{\infty} \frac{\exp(w B_{\text{imp}(2)}^{(k)})}{w} dw \\
 & + \frac{A_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)} - (-1)^{\hat{k}} [c^{\langle x^{(k)} \rangle} - C_{\text{imp}}^{(k)} B_{\text{imp}(1)}^{(k)}]}{(C_{\text{imp}}^{(k)})^2} \int_0^{\infty} \frac{\exp(w B_{\text{imp}(1)}^{(k)}) - \exp(w B_{\text{imp}(2)}^{(k)})}{w} dw \\
 & - \frac{c^{\langle x^{(k)} \rangle}}{(C_{\text{imp}}^{(k)})^2} [A_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)} - (-1)^{\hat{k}} c^{\langle x^{(k)} \rangle}] \int_0^{\infty} \frac{\exp(w B_{\text{imp}(1)}^{(k)})}{C_{\text{imp}}^{(k)} + w c^{\langle x^{(k)} \rangle}} dw \\
 & \left. \left. + \frac{c^{\langle x^{(k)} \rangle}}{(C_{\text{imp}}^{(k)})^2} [A_{\text{imp}(2)}^{(k)} C_{\text{imp}}^{(k)} - (-1)^{\hat{k}} c^{\langle x^{(k)} \rangle}] \int_0^{\infty} \frac{\exp(w B_{\text{imp}(2)}^{(k)})}{C_{\text{imp}}^{(k)} + w c^{\langle x^{(k)} \rangle}} dw \right] \right\}
 \end{aligned}$$

It is easy to check that $A_{\text{imp}(1)}^{(k)} + (-1)^{\hat{k}} B_{\text{imp}(1)}^{(k)} = A_{\text{imp}(2)}^{(k)} + (-1)^{\hat{k}} B_{\text{imp}(2)}^{(k)}$. Thus the second term on the right vanishes. By using the results of the limit (A-4) and the exponential integral (A-1)-(A-2) from Appendix A1, the following analytical solution is obtained:

$$\begin{aligned}
 H_{\text{imp}}^{(k)}(\xi) = & \text{Re} \left\{ \frac{E_{\text{imp}}^{(k)}}{(C_{\text{imp}}^{(k)})^2} \left[(-1)^{\hat{k}} C_{\text{imp}}^{(k)} [B_{\text{imp}(1)}^{(k)} - B_{\text{imp}(2)}^{(k)}] + (A_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)} - (-1)^{\hat{k}} [c^{\langle x^{(k)} \rangle} - C_{\text{imp}}^{(k)} B_{\text{imp}(1)}^{(k)}]) \ln \left(\frac{B_{\text{imp}(2)}^{(k)}}{B_{\text{imp}(1)}^{(k)}} \right) \right. \right. \\
 & - [A_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)} - (-1)^{\hat{k}} c^{\langle x^{(k)} \rangle}] \exp \left(\frac{-B_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) \text{Ei} \left(\frac{-B_{\text{imp}(1)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) \\
 & \left. \left. + [A_{\text{imp}(2)}^{(k)} C_{\text{imp}}^{(k)} - (-1)^{\hat{k}} c^{\langle x^{(k)} \rangle}] \exp \left(\frac{-B_{\text{imp}(2)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) \text{Ei} \left(\frac{-B_{\text{imp}(2)}^{(k)} C_{\text{imp}}^{(k)}}{c^{\langle x^{(k)} \rangle}} \right) \right] \right\}
 \end{aligned}$$

A2.4 Analytical Formulae for $\Phi_{\text{imp}}^{\pm}$ and $\phi_{\text{imp}}^{\pm}$

Formulae for $\Phi_{\text{imp}}^{\pm}(\mathbf{x}, \xi)$

$$\begin{aligned}\Phi_{\text{imp}}^{\pm}(\mathbf{x}, \xi) &= \frac{1}{2\pi} \text{Re} \left(\int_0^{\infty} Q^{\pm} e^{\pm i w z^{\pm}} dw \right) \\ &= \frac{-1}{\pi(c^{+} + c^{-})c^{\pm}} \text{Re} \left(\int_0^{\infty} \frac{H(\pm \xi_2) c^{\mp} e^{\pm i w(z^{\pm} - \bar{z}_0)} - H(\mp \xi_2) c^{\pm} e^{\pm i w(z^{\pm} - z_0)}}{\beta(1/c^{+} + 1/c^{-}) + w} dw \right) \\ &= \frac{-1}{\pi(c^{+} + c^{-})c^{\pm}} \text{Re} \left[H(\pm \xi_2) c^{\mp} \exp\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - \bar{z}_0)\right) \text{Ei}\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - \bar{z}_0)\right) \right. \\ &\quad \left. - H(\mp \xi_2) c^{\pm} \exp\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - z_0)\right) \text{Ei}\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - z_0)\right) \right]\end{aligned}$$

The final solution at the last line is obtained using result (A-1) applied to the 2nd line.

Formulae for $\phi_{\text{imp}}^{\pm}(\mathbf{x}, \xi)$

$$\begin{aligned}\phi_{\text{imp}}^{\pm}(\mathbf{x}, \xi) &= c_{ij}^{\pm} n_j \frac{\partial}{\partial x_j} \Phi_{\text{imp}}^{\pm}(\mathbf{x}, \xi) = \pm \frac{1}{2\pi} \text{Re} \int_0^{\infty} i w (c_{1j}^{\pm} n_j + \kappa^{\pm} c_{2j}^{\pm} n_j) Q^{\pm} e^{\pm i w z^{\pm}} dw \\ &= \frac{\mp 1}{\pi c^{\pm} (c^{+} + c^{-})} \text{Re} \left\{ i (c_{1j}^{\pm} n_j + \kappa^{\pm} c_{2j}^{\pm} n_j) \right. \\ &\quad \times \left[H(\pm \xi_2) c^{\mp} \int_0^{\infty} \left(\exp(\pm i w (z^{\pm} - \bar{z}_0)) - \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right) \frac{\exp(\pm i w (z^{\pm} - \bar{z}_0))}{\beta(1/c^{+} + 1/c^{-}) + w} \right) dw \right. \\ &\quad \left. \left. - H(\mp \xi_2) c^{\pm} \int_0^{\infty} \left(\exp(\pm i w (z^{\pm} - z_0)) - \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right) \frac{\exp(\pm i w (z^{\pm} - z_0))}{\beta(1/c^{+} + 1/c^{-}) + w} \right) dw \right] \right\} \\ &= \frac{1}{\pi c^{\pm} (c^{+} + c^{-})} \text{Re} \left\{ (c_{1j}^{\pm} n_j + \kappa^{\pm} c_{2j}^{\pm} n_j) \right. \\ &\quad \times \left[H(\pm \xi_2) c^{\mp} \left(\frac{1}{z^{\pm} - \bar{z}_0} \pm i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right) \exp\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - \bar{z}_0)\right) \text{Ei}\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - \bar{z}_0)\right) \right) \right. \\ &\quad \left. \left. - H(\mp \xi_2) c^{\pm} \left(\frac{1}{z^{\pm} - z_0} \pm i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right) \exp\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - z_0)\right) \text{Ei}\left(\mp i \beta \left(\frac{1}{c^{+}} + \frac{1}{c^{-}}\right)(z^{\pm} - z_0)\right) \right) \right] \right\}\end{aligned}$$

The 2nd line is obtained by expanding the fraction terms after substituting $Q^{\pm}$ in the 1st line.

The final form at the last line requires the result (A-1). Note that Einsteinian convention is applied to the repeated subscript j in $(c_{1j}^{\pm} n_j + \kappa^{\pm} c_{2j}^{\pm} n_j)$ above.

Appendix B

Appendix B

B1 MATLAB Code for Chapter 2

```

starting=clock;
format long g
syms x y t
%%
('Input/Output control');
Bmodel='1';
    %%Boundary model:'1' for circular segment,'2' for rectangular;
    Br1=1;Br2=2;ag11=pi/2;xOff=0;yOff=0;'circular seg.';%%give boundary geometry (Br1<Br2,0<ag113);
    LC1=1;LC2=1;corX=0;corY=0;'Rectangular';%%length & position of 1st corner;
    ml_m2=3;TcA_PtLcA=2;%%give boundary elements no. on C1 or C2 and no. of collo. arc or no. pt. on last collo. arc;
    C1='0';C2='0';C3='0';C4='0';%%'1' if T known on C(i),else '0' if q known;
    ll=0.22;12=0.22;    %%0<(l1,l2)<0.5,best accuracy:0.20<(l1,l2)<0.23
    testProblem='1';    %%give type of test problem.'1'-6';
    c011=1;c012=1;c022=2;tc=1;%%for sol.1~4:c011*c022>c012*2,(c011,c012,c022,tc)>0;
    'for sol.1~4:tc(>0)=time const.;tc=very big>1000 to turn to steady state prob.';
    recWellSpaceOutInpt='y';%%'y'('yes') to recover internal well-space-out points,else 'n'('no');
    TrA=1;PtLrA=1;    %%give integer no. of well-space out internal recovered arcs,
    recSBPT='n';d=0.28;    %%'y' to recover all pt. on smooth boundary/-0.5<q<0.5,distance from midpoint in %L(be) on element-be;
    recICPT='n';recAPT='n';recSCPT='n';%%'y' to recover all pt. on adjacent or sharp corner, else 'n';
    recRanPt='n';    %%'y' to recover randomly given pt.,
    xi_=[1.5];eta_=[1];lambdaRan=[1];
    %%give random points & the corresponding lambdas;
    ETRA='n';    %%ETRA='y' for entire temporal range approx. else 'n'.
    dt=0.1;TN=4;%%dt(>0)=small time-interval;TN(integer>=2)=total no. of temporal nodes in polynomial approx. of func.;
    RCT=0;    %%RCT(0<=ratio<1)=position of collocated time from upper temporal node;
    timeR=[0.03:0.2:2.03,2.03:0.5:5.03,6.03:1:11.03];%%timeR(>=0)=time at which T to be recovered;
    'RCT=1 OK! if (C1,C2,C3,C4)=(0,0,0,0)';'Optimum:RCT=0, TN>3, L/10<dt<0.18';
    'Central differential scheme when TN=2 & RCT=1/2,';
    minExBVer='y';    %%'y' to turn error message if exact BV<0;
    onSound='y';    %%'y' to turn sound on,else 'n';
    %%check:
    figBmodel='n';    %%'y' to show fig. of boundary model with colloc. & recovered pt.;
    plotValue='n';BVanyNumOcca=4;%%'y' to plot any few occasions of T(k) & q(k) on boundary;
    plotTvsTime='y';    %%'y' to plot T vs time at selected computed pt.
    copyResults='n';    %%'y' for tabulate results;
    %%
('Fool proof');
ml_m2=max(1,round(abs(ml_m2)));TcA_PtLcA=max(1,round(abs(TcA_PtLcA)));TrA=max(1,round(abs(TrA)));PtLrA=max(1,round(abs(PtLrA)));
l1=min(0.5,max(0,l1));l2=min(0.5,max(0,l2));d=min(0.5,max(-0.5,d));tc=max(0.001,tc);
TN=max(2,round(abs(TN)));dt=abs(dt);RCT=min(1,max(0,RCT));timeR=unique(abs(timeR));
if ETRA=='y' TSS=1;dt=max(timeR)/(TN-1);else TSS=max(1,ceil(double(vpa(max(timeR)/((TN-1)*dt),5))));end
('1.select test problems')
switch testProblem
case '1','q1=-q2;either [(p1,p2)<0 & (p1+q1,p2+q2)>0] or [(p1,p2)>0 & (p1+q1,p2+q2)<0]';
p1=-1;p2=-1;q1=1;q2=2;a1=1/2;a2=-a1*(q1*c011+q2*c012)/(q1*c012+q2*c022);a3=1;c=1;

```

Appendix B

```

g=exp(q1*x+q2*y);alfa=-exp(q1*x+q2*y)*((q1*c011+q2*c012)*p1+(q1*c012+q2*c022)*p2+(2*c012*p1*p2+c011*pl^2+c022*p2^2))/tc;alfa_=-alfa;
T=c*exp((-q1*x-q2*y)+(a1*x+a2*y+a3)+exp(p1*x+p2*y)*exp(-tc*t));
case '2', 'm=-even;q1=~0;either (x,y,q1,q2)>0 or (x,y,q1,q2)<0';
m=-1;q1=1;q2=1/2;q3=1/2;a1=0;a2=-a1*(q1*c011+q2*c012)/(q1*c012+q2*c022);a3=1;c=2;
g=(q1*x+q2*y+q3)*m;alfa=-m/tc*(q1*x+q2*y+q3)^(m-2)*(c011*q1^2+2*c012*q1*q2+c022*q2^2);alfa_=-alfa;
T=c/(1-m)*(q1*x+q2*y+q3)^(1-m)+(a1*x+a2*y+a3)+(q1*x+q2*y+q3)*exp(-tc*t);
case '3', 'y=c012*(x+q2/(q1*x))/c011 not be in domain,(q1,q2)>0,q2/q1 as big as possible';
a1=3;a2=-1;a3=1;q1=1;q2=3;p1=0.2;p2=0.2;p3=0.5;p4=6;
g=(q1*x^2-c011*q1*x*y+c012+q2)*~2; alfa=g*(c011*p1^2+2*c012*p1*p2+c022*p2^2)/tc; alfa_=-alfa;
T=l(a1*x+a2*y+a3)+p4*cos(p1*x+p2*y+p3)*exp(-tc*t))/sqrt(g);
case '4', 'q3>2*q4>0;either omeg=1 or omeg=0';
omeg>0;q1=0.5;q2=1/3;q3=3;q4=1;a1=a2=-a1*(q1*c011+q2*c012)/(q1*c012+q2*c022);a3=3;b1=1;b2=1;
g=q4*(cos(q1*x+q2*y))^2+q3;alfa=[omeg*(q3+q4-3*q4*(sin(q1*x+q2*y))^2)+(1-
omeg)*(3*q4*(cos(q1*x+q2*y))^2+q3)]*(c011*q1^2+2*c012*q1*q2+c022*q2^2)/tc;alfa_=-alfa;
T=(a1*x+a2*y+a3)+[omeg*b1*cos(q1*x+q2*y)+(1-omeg)*b2*sin(q1*x+q2*y)]*exp(-tc*t);
case '5', '2D-unsteady state homogeneous isotropic problem;alfa=const.>=1';
'alfa>b1^2/tc>0 & alfa>b2^2/tc>0;(a1,a2,b1,b2)=real';
a1=a2=2;b1=1;b2=1/2;alfa=1.5;
'fixed';c011=1;c022=1;c012=0;g=1;alfa=max([0.1,alfa,b1^2/tc,b2^2/tc]);alfa_=-num2str(alfa);
T=x*y+[a1*sin(b1*x+sqrt(alfa*tc-b1^2))*y+a2*cos(b2*x+sqrt(alfa*tc-b2^2))*y]*exp(-tc*t);
otherwise, '2D-steady state inhomogeneous isotropic problem';
'c011=c022,c012=0;a3=real const.;alfa>0;x=-2 or y=-2 not be in domain';
c011=1;c012=0;c022=c011;a3=0;
g=(x+2)^4*(y+2)^4; alfa=0.1; alfa_=-num2str(alfa);
T=((x+2)*cos(x+2)-sin(x+2))*((y+3)*exp(-y-2))/((x+2)^3/(y+2)^3+3+a3;
end

('2.Build-up boundary model')
switch Bmodel
case '1', LC1=Br2-Br1;LC2=ag13*Br2;str='Cir. seg.';
otherwise Br1=1E8;Br2=Br1+LC1;ag13=LC2/Br1;ag11=-ag13/2;xOff=-Br1+corX;yOff=LC2/2+corY;str='Rect. ';
end
if LC1<=LC2 m1=m1_m2;m2=round(LC2/LC1*m1_m2);m3=m1_m2;m4=ceil(ag13*Br1/LC1*m1_m2);else
m1=round(LC1/LC2*m1_m2);m2=m1_m2;m3=m1;m4=ceil(ag13*Br1/LC2*m1_m2);end
M=m1+m2+m3+m4;K1=2*M;
('Element end pt. ');
for be=1:M
if be<=m1, Xe(be)=xOff+(Br1+(be-1)*(Br2-Br1)/m1)*cos(ag11);Ye(be)=yOff+(Br1+(be-1)*(Br2-Br1)/m1)*sin(ag11);
elseif be>=m1+1 & be<=m1+m2,Xe(be)=xOff+Br2*cos(ag11+(be-m1-1)*ag13/m2);Ye(be)=yOff+Br2*sin(ag11+(be-m1-1)*ag13/m2);
elseif be>=m1+m2+1 & be<=m1+m2+m3,
Xe(be)=xOff+(Br2-(Br2-Br1)*(be-m1-m2-1)/m3)*cos(ag11+ag13);Ye(be)=yOff+(Br2-(Br2-Br1)*(be-m1-m2-1)/m3)*sin(ag11+ag13);
else
Xe(be)=xOff+Br1*cos(ag11+(M+1-be)*ag13/m4);Ye(be)=yOff+Br1*sin(ag11+(M+1-be)*ag13/m4);
end
end
Xe(M+1)=Xe(1);Ye(M+1)=Ye(1);
('Element L,midpt.,boundary collo. pt.,n & B.C');
for be=1:M
if be==M l=1;else l=be+1;end;
L(be)=sqrt((Xe(1)-Xe(be))^2+(Ye(1)-Ye(be))^2);
midX(be)=(Xe(be)+Xe(1))/2;
cosAlpha(be)=(Xe(1)-Xe(be))/L(be); sinAlpha(be)=(Ye(1)-Ye(be))/L(be);
n1(be)=sinAlpha(be);
node=1,-L(be)*(1-2*l)/2;1,L(be)*(1-2*l)/2)*[midX(be),midY(be);cosAlpha(be),sinAlpha(be)];
X(2*be-1)=node(1,1);
Y(2*be-1)=node(1,2);
X(2*be)=node(2,1);
Y(2*be)=node(2,2);
lambdaC(2*be)=1/2;
lambdaC(2*be)=1/2;

```

Appendix B

```

end
('Internal adjacent angle');
for be=1:M
    if be==1 j=M;else j=be-1;end;
    iAJAg1=real(acos(dot(n(j,:),n(be,:)))));
    if sum(cross(n(j,:),n(be,:)))>=0 lambdaA(be)=(pi-iAJAg1)/(2*pi);
    else lambdaA(be)=(pi+iAJAg1)/(2*pi);end
end
('Give well-spaced out inetrnal collocation pt. ');
if LC1<=LC2 TcA=TcA_PtLcA;PtLcA=round(LC2/LC1*TcA_PtLcA);else PtLcA=TcA_PtLcA;TcA=round(LC1/LC2*TcA_PtLcA);end
K2=0;
for cA=1:TcA
    cr(cA)=Br1+(Br2-Br1)*cA/(TcA+1);
    ticPt(cA)=round((1+(TcA+1)*(PtLcA-1)*cr(cA))/(Br1+Br2*TcA));
    for icPtA=1:ticPt(cA)
        K2=K2+1;lambdaC(K1+K2)=1;
        X(K1+K2)=xOff+cr(cA)*cos(ag1+ag13*icPtA/(ticPt(cA)+1));
        Y(K1+K2)=yOff+cr(cA)*sin(ag1+ag13*icPtA/(ticPt(cA)+1));
    end
end
('3.Initial-BC, exact values at collo.pt., RBF, Integration const., temporal & differentiated temporal matrix')
('alfaC(k),gC(k),G(k),w(k) at collo. pt. ');
for k=1:K1+K2
    x=X(k);y=Y(k);
    switch testProblem
        case '1',dSqrtgdx=1/2*exp(q1*x+q2*y)^(1/2)*q1;dSqrtgdy=1/2*exp(q1*x+q2*y)^(1/2)*q2;
            w(k)=1/4*c011*exp(q1*x+q2*y)^(1/2)*q1^2+1/2*c012*exp(q1*x+q2*y)^(1/2)*q1*q2+1/4*c022*exp(q1*x+q2*y)^(1/2)*q2^2;
        case '2',dSqrtgdx=1/2*(q1*x+q2*y+q3)^m^(1/2)*m*q1/(q1*x+q2*y+q3);
            dSqrtgdy=1/2*((q1*x+q2*y+q3)^m)^(1/2)*m^2/(q1*x+q2*y+q3);
            w(k)=1/4*((q1*x+q2*y+q3)^m)^(1/2)*m*(m-2)*(c011*q1^2+2*c012*q1*q2+c022*q2^2)/(q1*x+q2*y+q3)^2;
        case '3',dSqrtgdx=2*q1*x-c011*q1*y/c012; dSqrtgdy=-c011*q1*x/c012; w(k)=0;
        case '4',dSqrtgdx=-1/(q4*cos(q1*x+q2*y)^2+q3)^(1/2)*q4*cos(q1*x+q2*y)*sin(q1*x+q2*y)*q1;
            dSqrtgdy=-1/(q4*cos(q1*x+q2*y)^2+q3)^(1/2)*q4*cos(q1*x+q2*y)*sin(q1*x+q2*y)*q2;
            w(k)=-q4*(-q3*c011*q1^2+2*cos(q1*x+q2*y)^2*q3*c011*q1^2-2*q3*c012*q1*q2+4*cos(q1*x+q2*y)^4*q4*c012*q1*q2-
            q3*c022*q2^2+2*cos(q1*x+q2*y)^2*q3*c022*q2^2+cos(q1*x+q2*y)^4*q4*c011*q1^2+2*cos(q1*x+q2*y)^4*q4*c012*q1*q2+cos(q1*x+q2*y)^4*q4*c012*q1*q2)/(q4*cos(q1*x
            +q2*y)^2+q3)^(3/2);
        case '5',dSqrtgdx=0;dSqrtgdy=0;w(k)=0;
            otherwise,dSqrtgdx=2/((x+2)^4*(y+2)^4)^(1/2)*(x+2)^3*(y+2)^4;
            dSqrtgdy=2/((x+2)^4*(y+2)^4)^(1/2)*(x+2)^4*(y+2)^3;
            w(k)=2*c011*(x^2*y^2+4*x*y^2+4*y^2+4*x^2*y+16*x*y+16*y^2+4*x^2*y+16*x*y+16*y^2+4*x^2*y+16*x*y+16*y^2)/(x+2)^4*(y+2)^4)^(1/2);
    end
    alfaC(k)=subs(alfa);
    gC(k)=subs(g);if k<=K1 be=round(k/2);G(k)=nl(be)*(c011*dSqrtgdx+c012*dSqrtgdy)+n2(be)*(c012*dSqrtgdx+c022*dSqrtgdy);else G(k)=0;end
end
('exact T(k) & q(k) at all collo. pt. at different times>=0; Initial-BC');
for k=1:K1+K2 x=X(k);y=Y(k);
    for tIdx=1:(TN-1)*TSS+1 t=dt*(tIdxC-1);
        switch testProblem
            case '1',dTdx=-c*q1*exp(-q1*x-q2*y)+al+p1*exp(p1*x+p2*y-tc*t); dTdy=-c*q2*exp(-q1*x-q2*y)+a2+p2*exp(p1*x+p2*y-tc*t);
            case '2',dTdx=c*(q1*x+q2*y+q3)^(-m)*q1*al+q1*exp(-tc*t); dTdy=c*(q1*x+q2*y+q3)^(-m)*q2+a2+q2*exp(-tc*t);
            case '3',dTdx=(al-p4*sin(p1*x+p2*y+p3)*p1*exp(-tc*t))/(q1*x^2-c011*q1*x*y/c012+q2)^(3/2); dTdy=(a2-p4*sin(p1*x+p2*y+p3)*p1*exp(-tc*t))/(q1*x^2-c011*q1*x*y/c012+q2)^(3/2);
            case '4',dTdx=(al-p4*sin(p1*x+p2*y+p3)*p1*exp(-tc*t))/(q1*x^2-c011*q1*x*y/c012+q2)^(3/2); dTdy=(a2-p4*sin(p1*x+p2*y+p3)*p1*exp(-tc*t))/(q1*x^2-c011*q1*x*y/c012+q2)^(3/2);
            case '5',dTdx=(al-p4*sin(p1*x+p2*y+p3)*p1*exp(-tc*t))/(q1*x^2-c011*q1*x*y/c012+q2)^(3/2); dTdy=(a2-p4*sin(p1*x+p2*y+p3)*p1*exp(-tc*t))/(q1*x^2-c011*q1*x*y/c012+q2)^(3/2);
        end
    end
end

```

Appendix B

```

case '4', dTdx=a1+(-omeg*b1*sin(q1*x+q2*y)*q1+(1-omeg)*b2*cos(q1*x+q2*y)*q1)*exp(-tc*t);
dTdy=a2+(-omeg*b1*sin(q1*x+q2*y)*q2+(1-omeg)*b2*cos(q1*x+q2*y)*q2)*exp(-tc*t);
case '5', dTdx=y+(a1*cos(b1*x+(alfa*tc-bl^2)^(1/2))*y)*b1-a2*sin(b2*x+(alfa*tc-b2^2)^(1/2))*y*(alfa*tc-b2^2)^(1/2))*exp(-tc*t);
dTdy=x+(a1*cos(b1*x+(alfa*tc-bl^2)^(1/2))*y)*(alfa*tc-bl^2)^(1/2)-a2*sin(b2*x+(alfa*tc-b2^2)^(1/2))*y*(alfa*tc-b2^2)^(1/2))*exp(-tc*t);
otherwise, dTdx=-1/(x+2)^2*sin(x+2)*(y+3)*exp(-y-2)/(y+2)^3-3*((x+2)*cos(x+2)-sin(x+2))*(y+3)*exp(-y-2)/(x+2)^3/(y+2)^3-3*((x+2)*cos(x+2)-
sin(x+2))*(y+3)*exp(-y-2)/(x+2)^3/(y+2)^4;
end
exTc(tIdxC,k)=subs(T);
if k<=K1 be=round(k/2);ex_g(tIdxC,k)=-gc(k)*[n1(be)*(c011*dTdx+c012*dTdy)+n2(be)*(c012*dTdx+c022*dTdy)] ;else ex_g(tIdxC,k)=0;end
end
if
k<=2*m1
elseif (k>=2*m1+1 & k<=2*(m1+m2))
elseif (k>=2*(m1+m2)+1 & k<=2*(m1+m2+m3))
elseif (k>=2*(m1+m2+m3)+1 & k<=K1)
else
end
end
end

if minExBVer== 'y' & min(min(exTc))<0
soundsc(wavread('c:\PROGRA-1\NETMEE-1\RINGIN.WAV')) ;error(['negative value on boundary, min. T= ',num2str(min(min(exTc)))]);
else '-',end

for r=1:TN if r==1 gamal(r,:)=ones(1,K1+K2);else gamal(r,:)=gama;end; end
gama2=gama1/gama2(2:TN,:)=ones(TN-1,K1+K2)-gama2(2:TN,:);clear gama
('Integration constants ID1..2, IID1..2:titaC,RBF');
for p=1:K1+K2
x=X(p);y=Y(p);
for be=1:M+X2
xi=X(p);eta=Y(p);
if be<=M
A=4*[(c011*sinAlpha(be)-c012*cosAlpha(be))* (midY(be)-eta)+(c022*cosAlpha(be)-c012*sinAlpha(be))* (midX(be)-xi)]/L(be)/(c022*cosAlpha(be)^2-
2*c012*cosAlpha(be)*sinAlpha(be)+c011*sinAlpha(be)^2);
B=4*[c022*(midX(be)-xi)^2-2*c012*(midX(be)-xi)*(midY(be)-eta)+c011*(midY(be)-eta)^2]/L(be)^2/[c022*cosAlpha(be)^2-
2*c012*cosAlpha(be)*sinAlpha(be)+c011*sinAlpha(be)^2];
D=sqrt(c011*c022-c012^2)*[(midX(be)-xi)^2*(midY(be)-eta)^2]/(L(be)^2*pi);
E1=L(be)*log(L(be)^2*(c022*cosAlpha(be)+c011*sinAlpha(be)^2)/(L(be)^2*pi);
E2=L(be)/(8*pi)*sqrt(c011*c022-c012^2);
if 4*B-A^2<1E-12
ID1(p,2*be-1)=0;ID2(p,2*be)=0;
if abs(A-2)<1e-12
IID1(p,2*be-1)=[(1-2*12)*E1+E2*((1-2*12)*log(4)+4*12-3)]/(1-11-12);
IID2(p,2*be)=[(1-2*11)*E1+E2*((1-2*11)*log(4)+4*11-1)]/(1-11-12);
elseif abs(A+2)<1e-12
IID1(p,2*be-1)=[(1-2*12)*E1+E2*((1-2*12)*log(4)+4*11-1)]/(1-11-12);
IID2(p,2*be)=[(1-2*11)*E1+E2*((1-2*11)*log(4)+4*11-3)]/(1-11-12);
else
IID1(p,2*be-1)=real((1-2*12)*E1+E2/8*((2-8*12+A)*(2+A)*log(2+A)+(6-8*12+A)*(2-A)*log(2-A)-16*(1-2*12)*(log(2)+1)-4*A)]/(1-11-12);
IID2(p,2*be)=real((1-2*11)*E1+E2/8*((2-8*11-A)*(2-A)*log(2-A)+(6-8*11-A)*(2+A)*log(2+A)-16*(1-2*11)*(log(2)+1)+4*A)]/(1-11-12);
end
end
ID1(p,2*be-1)=real((D/2)/(1-11-12))*[log((1-A+B)/(1+A+B))/2+(2*(1-2*12)+A)*atan2(sqrt(4*B-A^2),B-1)/sqrt(4*B-A^2)]];
ID2(p,2*be)=real((D/2)/(1-11-12))*[log((1-A+B)/(1+A+B))/2+(2*(1-2*11)-A)*atan2(sqrt(4*B-A^2),B-1)/sqrt(4*B-A^2)]];
IID1(p,2*be-1)=real((1-2*12)*E1+E2/8*((1-A+B)^2*(2+A)*log(1+A+B)-(A^2+2*A-2*B-6+4*12*(2-A))*log(1-A+B)-4*(4-8*12+A)+2*(2-
4*12+A)*sqrt(4*B-A^2))*atan2(sqrt(4*B-A^2),B-1)]/(1-11-12);

```

Appendix B

```

A) *sqrt(4*B-A^2)*atan2(sqrt(4*B-A^2), B-1))/(1-11-12);
    IID2(p, 2*be)=real((1-2*11)*E1+E2/8*((A^2-2*A-2*B+2-4*11*(2-A))*log(1-A+B)-(A^2-2*A-2*B-6+4*11*(2+A))*log((1+A+B)-4*(4-8*11-A)+2*(2-4*11-
end
else
    ID1(p, M+be)=0; ID2(p, M+be)=0; IID1(p, M+be)=0; IID2(p, M+be)=0;
end
end
for k=1:K1+K2
    titac(p, k)=c022*(1/4*(x-X(k)-c012/c022*(y-Y(k)))^2+1/4*(c011*c022-c012^2)/c022^2*(y-Y(k))-c012/c022*(y-Y(k)))^2+(c011*c022*(y-Y(k)))^2+(c011*c022-
c012^2)/c022^2*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022-
DtitacDX=c022*(1/2*x-1/2*X(k)-1/2*c012/c022*(y-Y(k))+1/8*((x-X(k)-c012/c022*(y-Y(k)))^2+(c011*c022-c012^2)/c022^2*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022-
2*c012/c022*(y-Y(k))+1/10*((x-X(k)-c012/c022*(y-Y(k)))^2+(c011*c022-c012^2)/c022^2*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022-
c012^2);
    DtitacDY=c022*(-1/2*(x-X(k)-c012/c022*(y-Y(k)))^2+1/2*(c011*c022-c012^2)/c022^2*(y-Y(k))+1/8*((x-X(k)-c012/c022*(y-Y(k)))^2+(c011*c022*(y-Y(k))-c012/c022-
c012^2)/c022^2*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-
Y(k)))^2+(c011*c022-c012^2)/c022^2*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022*(y-Y(k))-c012/c022-
c012^2);
    RBF(p, k)=1+(x-X(k)-c012/c022*(y-Y(k)))^2+(c011*c022-c012^2)/c022^2*(y-Y(k))+1/8*((x-X(k)-c012/c022*(y-Y(k)))^2+(c011*c022*(y-Y(k))-c012/c022*(y-
Y(k)))^2)^(3/2);
    if p<=K1 be=round(p/2); TITA(p, k)=n1(be)*(c011*DtitacDX+c012*DtitacDY)+n2(be)*(c012*DtitacDX+c022*DtitacDY); else TITA(p, k)=0; end
end
end
ID=ID1+ID2; IID=IID1+IID2; clear ID1 ID2 IID1 IID2
sigma=inv(RBF); Psi=diag(lambdaC)*titac+IID*TITA-ID*titac;
('Temporal & differentiated temporal matrix');
for s=2:TN t_c=(s-1-RCT)*dt;
    for r=1:TN t_m1=1; DTM1=0;
        for r1=1:TN t_r1=(r1-1)*dt; DTM2=1;
            if r1==r
                t_m1=(t_c-t_r1)*TM1;
                for r2=1:TN t_r2=(r2-1)*dt;
                    if r2==r r DTM2=(t_c-t_r2)*DTM2; else '-'; end
                end
                DTM1=DTM1+DTM2;
            else '-'; end
        end
    end
    TM(s-1, r)=TM1/((-1)^(TN-r))*factorial(r-1)*factorial(TN-r)*dt^(TN-1);
    DTM(s-1, r)=DTM1/((-1)^(TN-r))*factorial(r-1)*factorial(TN-r)*dt^(TN-1)];
end
end
('coefficients');
Phi1=ID*diag(gc.^0.5)-IID*diag(G)+Psi*sigma*diag(w);
Phi12=Psi*sigma*diag(alfac.*gc.^(-0.5));
Phi21=IID*diag(gc.^(-0.5));
for p=1:K1+K2
    Phi0((p-1)*(TN-1)+1:p*(TN-1), (p-1)*TN+1:p*TN)=lambdaC(p)*gc(p)^(0.5)*TM;
    for k=1:K1+K2
        Phi1((p-1)*(TN-1)+1:p*(TN-1), (k-1)*TN+1:k*TN)=Phi11(p, k)*TM+Phi12(p, k)*DTM;
        Phi2((p-1)*(TN-1)+1:p*(TN-1), (k-1)*TN+1:k*TN)=Phi21(p, k)*TM;
    end
end
end
('4.assemble overall sys. of eqn. & solve unknown for t>0')
a=-(Phi1-Phi0)*diag(gama2(:))+Phi2*diag((gama1(:)));
for k=1:K1+K2 a(:, (k-1)*(TN-1)+1)=[]; end

```


Appendix B

```

approxTc=exTc(1,:);approx_q=ex_q(1,:);
for SS=1:TSS
    approxTc((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=exTc((TN-1)*(SS-1)+2:SS*(TN-1)+1,:).*gama1(2:TN,:);
    approx_q((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=ex_q((TN-1)*(SS-1)+2:SS*(TN-1)+1,:).*gama2(2:TN,:);
    b=(Phi1-Phi0)*reshape(approxTc((TN-1)*(SS-1)+1:SS*(TN-1)+1,:),[(K1+K2)*TN 1])+Phi2*reshape(approx_q((TN-1)*(SS-1)+1:SS*(TN-1)+1,:),[(K1+K2)*TN 1]);
    unknown=reshape(inv(a)*b,[TN-1 K1+K2]);
    approxTc((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=approxTc((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)+unknown.*gama2(2:TN,:);
    approx_q((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=approx_q((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)+unknown.*gama1(2:TN,:);
end

clear ID IID Psi Phi11 Phi12 Phi21 Phi0 Phi1 Phi2 TM DTM a b
('5.Determine pt. to be computed')
('Give well-spaced out inetrnal points to be computed');
rPt=0;TirPt=0;
if recWellSpaceOutInPt=='y' & TrA>0 & PTLrA>0
    ('Give well-spaced out inetrnal points');
    for rA=1:TrA
        radA(rA)=Br1+(Br2-Br1)*rA/(TrA+1);
        tiPt(rA)=round(1+(TrA+1)*(PtLrA-1)*radA(rA)/(Br1+Br2*TrA));
        for iPt=1:tiPt(rA)
            rPt=rPt+1;TirPt=TirPt+1;lambdAR(rPt)=1;
            xi(rPt)=xOff+radA(rA)*cos(agll+agll3*iPt/(tiPt(rA)+1)); eta(rPt)=yOff+radA(rA)*sin(agll+agll3*iPt/(tiPt(rA)+1));
        end
    end
else TirPt=0;end
if recSBPt=='y' ('Give well-spaced out smooth boundary pt. ');
    TSBPt=M;element=0;
    for be=1:M
        rPt=rPt+1;
        element(rPt)=be;
        lambdAR(rPt)=1/2;
        xi(rPt)=midX(be)+d*L(be)*cosAlpha(be);
        eta(rPt)=midY(be)+d*L(be)*sinAlpha(be);
    end
else TSBPt=0;end
if recICPt=='y' ('Include internal collocation pt. to be computed');TiCpt=K2;
    for iCpt=1:TiCpt
        rPt=rPt+1;lambdAR(rPt)=1;xi(rPt)=X(K1+iCpt);eta(rPt)=Y(K1+iCpt);
    end
else TiCpt=0;end
if recAPT=='y' ('Give well-spaced out adjacent pt. ');
    skip=0;
    for be=1:M
        if be==1|be==m1+1|be==m1+m2+1|be==m1+m2+m3+1 skip=skip+1;
        else rPt=rPt+1;element(rPt)=be;lambdAR(rPt)=lambdAA(be);xi(rPt)=Xe(be);eta(rPt)=Ye(be);
        end
    end
end
TaPt=M-skip;
TaPt=0;end
if recSCPt=='y' ('Give pt. at sharp corners');TSCPt=4;
    for SCPt=1:TSCPt
        if SCPt==1, k=1;elseif SCPt==2,k=m1+1;elseif SCPt==3,k=m1+m2+1;else k=m1+m2+m3+1;end
        rPt=rPt+1;element(rPt)=k;lambdAR(rPt)=lambdAA(k);xi(rPt)=Xe(k);eta(rPt)=Ye(k);
    end
else TSCPt=0;end

```

Appendix B

```

if recRanPt=='y' ('Include specified random pt.'):TranPt=length(xi_);
for ranPt=1:TranPt rPt=rPt+1;xi(rPt)=xi_(ranPt);eta(rPt)=eta_(ranPt);lambdar(rPt)=lambdarR(ranPt);end
else TranPt=0;end
TrPt=TirPt+TiCpt+TSBpt+TaPt+TSCpt+TranPt;

('6.Compute approximate T at selected pt.')
if TrPt>0
    ('Integration const. ID1...2 & IID1...2');
    for rPt=1:TrPt
        for be=1:M+K2
            if be<=M
                A=4*(c011*sinAlpha(be)-c012*cosAlpha(be))*(midY(be)-eta(rPt))+(c022*cosAlpha(be)-c012*sinAlpha(be))*(midX(be)-
                xi(rPt))/L(be)/(c022*cosAlpha(be)^2-2*c012*cosAlpha(be)*sinAlpha(be)+c011*sinAlpha(be)^2);
                B=4*(c022*(midX(be)-xi(rPt))^2-2*c012*(midX(be)-xi(rPt))*(midY(be)-eta(rPt))+c011*(midY(be)-eta(rPt))^2)/L(be)^2/[c022*cosAlpha(be)^2-
                2*c012*cosAlpha(be)*sinAlpha(be)+c011*sinAlpha(be)^2];
                D=sqrt(c011*c022-c012^2)*[(midX(be)-xi(rPt))*n1(be)+(midY(be)-eta(rPt))*n2(be)]/(c022*cosAlpha(be)^2-
                2*c012*cosAlpha(be)*sinAlpha(be)+c011*sinAlpha(be)^2)/L(be)*pi);
                E1=L(be)*log(L(be)^2*(c022*cosAlpha(be)^2-2*c012*cosAlpha(be)+c011*sinAlpha(be)^2)/(4*c022))/(8*pi*sqrt(c011*c022-c012^2));
                E2=L(be)/(8*pi*sqrt(c011*c022-c012^2));
                if 4*B-A^2<1E-12
                    ID1(rPt,2*be-1)=0;ID2(rPt,2*be)=0;
                    if abs(A-2)<1e-12
                        IID1(rPt,2*be-1)=[(1-2*12)*E1+E2*((1-2*12)*log(4)+4*12-3)]/(1-11-12);
                        IID2(rPt,2*be)=[(1-2*11)*E1+E2*((1-2*11)*log(4)+4*11-1)]/(1-11-12);
                    elseif abs(A+2)<1e-12
                        IID1(rPt,2*be-1)=[(1-2*12)*E1+E2*((1-2*12)*log(4)+4*11-1)]/(1-11-12);
                        IID2(rPt,2*be)=[(1-2*11)*E1+E2*((1-2*11)*log(4)+4*11-3)]/(1-11-12);
                    else IID1(rPt,2*be-1)=real((1-2*12)*E1+E2/8*[(2-8*12+A)*(2+A)*log(2+A)+(6-8*12+A)*(2-A)*log(2+A)-16*(1-2*12)*(log(2)+1)-4*A])/((1-11-12);
                        IID2(rPt,2*be)=real((1-2*11)*E1+E2/8*[(2-8*11-A)*(2-A)*log(2-A)+(6-8*11-A)*(2+A)*log(2+A)-16*(1-2*11)*(log(2)+1)+4*A])/((1-11-12);
                    end
                else
                    ID1(rPt,2*be-1)=real((D/2)/(1-11-12))*[log((1-A+B)/(1+A+B))/2+(2*(1-2*11)+A)*atan2(sqrt(4*B-A^2),B-1)/sqrt(4*B-A^2)]];
                    ID2(rPt,2*be)=real((D/2)/(1-11-12))*[log((1+A+B)/(1-A+B))/2+(2*(1-2*11)-A)*atan2(sqrt(4*B-A^2),B-1)/sqrt(4*B-A^2)]];
                    IID1(rPt,2*be-1)=real((1-2*12)*E1+E2/8*[(A^2+2*A-2*B+2-4*12*(2+A))*log(1+A+B)-(A^2+2*A-2*B+6+4*12*(2-A))*log(1-A+B)-4*(4-
                    8*12+A)+2*(2-4*12+A)*sqrt(4*B-A^2)*atan2(sqrt(4*B-A^2),B-1)]/(1-11-12);
                    IID2(rPt,2*be)=real((1-2*11)*E1+E2/8*[(A^2-2*A-2*B+2-4*11*(2-A))*log(1-A+B)-(A^2-2*A-2*B+6+4*11*(2+A))*log(1+A+B)-4*(4-8*11-A)+2*(2-
                    4*11-A)*sqrt(4*B-A^2)*atan2(sqrt(4*B-A^2),B-1)]/(1-11-12);
                end
            else
                ID1(rPt,M+be)=0;ID2(rPt,M+be)=0;IID1(rPt,M+be)=0;IID2(rPt,M+be)=0;
            end
        end
    end
    ('gR,titaR & Psi');
    x=xi(rPt);y=eta(rPt);gR(rPt)=subs(g);
    for k=1:K1+K2
        titaR(rPt,k)=c022*(1/4*(x-X(k)-c012/c022*(y-Y(k)))^2+1/4*(c011*c022-c012^2)/c022^2*(y-Y(k))-c012/c022*(y-
        Y(k)))^2+(c011*c022-c012^2)/c022^2*(y-Y(k))^2+1/25*((x-X(k)-c012/c022*(y-Y(k)))^2+(c011*c022-c012^2)/c022^2*(y-Y(k))^2)/(c011*c022-c012^2);
    end
end
ID=ID1+ID2;IID=IID1+IID2;Psi=diag(lambdar)*titaR+IID*TITA-ID*titaC;
clear ID1 ID2 IID1 ID2 titaC titaR TITA
('Temporal & differentiated temporal matrix');
for tIdxR=1:length(timer) SS=max(1,ceil(double(vpa(timer(tIdxR))/((TN-1)*dt),5)))));
for r=1:TN TM1=1;DTM1=0;
    for r1=1:TN t_r1=((SS-1)*(TN-1)+r1-1)*dt;DTM2=1;
        if r1~=r

```


Appendix B

```

TM1=(timer(tIdxR)-t_r1)*TM1;
for r2=1:TN t_r2=((SS-1)*(TN-1)+r2-1)*dt;
    if r2~=r1 DTM2=(timer(tIdxR)-t_r2)*DTM2;else '-';end
end
DTM1=DTM1+DTM2;
else '-';end
end
end
TM(tIdxR,r)=TM1/((-1)^(TN-r)*factorial(r-1)*factorial(TN-r)*dt^(TN-1));
DTM(tIdxR,r)=DTM1/((-1)^(TN-r)*factorial(r-1)*factorial(TN-r)*dt^(TN-1));
end
end
('Exact & approx. T at pt. to be computed at different timeR');
Phi1=ID*diag(gc.^0.5)-IID*diag(G)+Psi*sigma*diag(w);
Phi12=Psi*sigma*diag(alfaC.*gc.^(-0.5)); Phi21=IID*diag(gc.^(-0.5));
for tIdxR=1:length(timeR) t=timeR(tIdxR);SS=max(1,ceil(double(vpa(timer(tIdxR)/(TN-1)*dt),5))));
    for rpt=1:TrPt x=xi(rPt);y=eta(rPt);
        for k=1:K1+K2
            Phil(rPt,(k-1)*TN+1:k*TN)=Phil1(rPt,k)*TM(tIdxR,:)+Phil2(rPt,k)*DTM(tIdxR,:);
            Phi2(rPt,(k-1)*TN+1:k*TN)=Phi21(rPt,k)*TM(tIdxR,:);
        end
        exTr(rPt,tIdxR)=subs(T);
    end
    approxTr(:,tIdxR)=[Phil*reshape(approxTc((TN-1)*(SS-1)+1,:),[(K1+K2)*TN 1])+Phi2*reshape(approx_q((TN-1)*(SS-1)+1:SS*(TN-1)+1,:),[(K1+K2)*TN 1])/((lambdaR.*gr.^(1/2))].');
end
errTr=exTr-approxTr;
percentErrTr=abs(exTr.^(-1).*errTr)*100;

('7.Show results of PDRDLBEM')
if Br1==0 lastL=0;else lastL=L(m1+m2+m3+1);end
[istr,'m1,m2,m3,m4,M,K1,K2,element length at C1,C2,C3,C4,B.C)=(',' num2str([m1,m2,m3,m4,M,K1,K2,L(1),L(m1+1),L(m1+m2+1),lastL]),',
',C1,C2,C3,C4,')'],
['(dt,TN,RCT,TSS)=(',' num2str([dt,TN,RCT,TSS]),')']
['Prob.','char(testProblem),'T(x,y,t)=' char(T),' g(x,y)=' char(g),' alfa(x,y)=' char(alfa_),'
(c011,c012,c022)=(',' num2str([c011,c012,c022]),')']
if TirPt>0 ('Internal well spaced-out pt.')
    'Exact T',
    char([' xi > ',' num2str(xi(1:TirPt))'],[ ' t /eta > ',' num2str(eta(1:TirPt))']),
    [timeR.','exTr(1:TirPt,:).'],
    'computed T', approxTr(1:TirPt,:).',
    'Error', errTr(1:TirPt,:).',
    '% Error', percentErrTr(1:TirPt,:).',
else '-';end
if recSBpt=='y' ('Smooth boundary pt.')
    'Exact T',
    char([' element> ',' num2str(element(1+TirPt:TirPt+TSBpt))'],[ ' xi> ',' num2str(xi(1+TirPt:TirPt+TSBpt))'],[ ' t/eta> ',' num2str(eta(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt))']),
    [timeR.','exTr(1+TirPt:TirPt+TSBpt,:).'],
    'computed T', approxTr(1+TirPt:TirPt+TSBpt,:).',
    'Error', errTr(1+TirPt:TirPt+TSBpt,:).',
    '% Error', percentErrTr(1+TirPt:TirPt+TSBpt,:).',
else '-';end
if recICpt=='y' ('Internal collocation pt.')
    'Exact T',
    char([' xi> ',' num2str(xi(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt))'],[ ' t/eta> ',' num2str(eta(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt))']),

```

Appendix B

```

[timeR.',exTr(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt,:).'],
'computed T', approxTr(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt,:).',
'Error', errTr(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt,:).',
'% Error', percentErrTr(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt,:).',
else '-';end
if recApt=='y'
'Exact T', ('Adjacent pt.')
char(['', element> ' num2str(element(1+TirPt+TSBpt+TiCpt:TirPt+TSBpt+TiCpt+TaPt)),[' xi>
' num2str(x(1+TirPt+TSBpt+TiCpt:TirPt+TSBpt+TiCpt+TaPt)),['
[timeR.',exTr(1+TirPt+TSBpt:TirPt+TSBpt+TiCpt+TaPt+TSCpt)),['
'computed T', approxTr(1+TirPt+TSBpt+TiCpt+TaPt+TSCpt,:).'],
'Error', errTr(1+TirPt+TSBpt+TiCpt+TaPt+TSCpt+TSCpt,:).',
'% Error', percentErrTr(1+TirPt+TSBpt+TiCpt+TaPt+TSCpt+TSCpt,:).',
else '-';end
if recSCpt=='y' ('Sharp corner pt.')
'Exact T',
char(['', element> ' num2str(element(1+TirPt+TiCpt+TSBpt+TaPt:TirPt+TiCpt+TSBpt+TaPt+TSCpt)),[' xi>
' num2str(x(1+TirPt+TiCpt+TSBpt+TaPt:TirPt+TiCpt+TSBpt+TaPt+TSCpt)),['
[timeR.',exTr(1+TirPt+TiCpt+TSBpt+TaPt:TirPt+TiCpt+TSBpt+TaPt+TSCpt)),['
'computed T', approxTr(1+TirPt+TiCpt+TSBpt+TaPt+TSCpt+TSCpt,:).'],
'Error', errTr(1+TirPt+TiCpt+TSBpt+TaPt:TirPt+TiCpt+TSBpt+TaPt+TSCpt,:).',
'% Error', percentErrTr(1+TirPt+TiCpt+TSBpt+TaPt:TirPt+TiCpt+TSBpt+TaPt+TSCpt,:).',
else '-';end
if recRanPt=='y' ('Randomly given pt.')
'Exact T',
char(['', xi > ' num2str(x(1+TirPt+TiCpt+TSBpt+TaPt+TSCpt:TrPt)),['
[timeR.',exTr(1+TirPt+TiCpt+TSBpt+TaPt+TSCpt:TrPt)),['
'computed T', approxTr(1+TirPt+TiCpt+TSBpt+TaPt+TSCpt:TrPt,:).'],
'Error', errTr(1+TirPt+TiCpt+TSBpt+TaPt+TSCpt:TrPt,:).',
'% Error', percentErrTr(1+TirPt+TiCpt+TSBpt+TaPt+TSCpt:TrPt,:).',
else '-';end
if
copyResults=='y', 't', timeR.', 'xi', xi.', 'eta', eta.', exTr=:), approxTr=:), errTr=:errTr.', errTr(:),
percentErrTr=:percentErrTr.', '%err.', percentErrTr(:), else '-';end
['Tol. %err.=%', num2str(sum(sum(percentErrTr))), '% avg %err.=', num2str(sum(sum(percentErrTr)))/(TrPt*length(timeR))), '% norm of
err.=', num2str(sqrt(num2str(sum(errTr.^2))/(TrPt*length(timeR))))), '% avg. abs. Err.=', num2str(sum(abs(errTr)))/(TrPt*length(timeR)))]],
else xi=-1e16;eta=-1e16; 'No results shown',end
('Checkig');
if figBmodel=='y' figure;plot(Xe,Ye,'k-',X,Y,'b.',xi,eta,'rx');title('Boundary model');xlabel('x');ylabel('y');axis equal;axis([min(Xe)-abs(max(Xe)-
min(Xe))/10 max(Xe)+abs(max(Xe)-min(Xe))/10 min(Ye)-abs(max(Ye)-min(Ye))/10]');legend('boundary elements','DLE
collocation pt.','Points under tested');
text(midX(round(m1/2)),midY(round(m1/2)),'\Gamma_1');text(midX(m1+round(m2/2)),midY(m1+round(m2/2)),'\Gamma_2');
text(midX(m1+m2+round(m3/2)),midY(m1+m2+round(m3/2)),'\Gamma_3');text(midX(m1+m2+m3+round(m4/2)),midY(m1+m2+m3+round(m4/2)),'\Gamma_4');
else '-';end
if plotBvalue=='y' & BVanyNumOcca>=1
if BVanyNumOcca<(TN-1)*TSS
for i=1:BVanyNumOcca occa(i)=ceil(i*(TN-1)*TSS/BVanyNumOcca);end;
else occa=2:1:(TN-1)*TSS;end
for tidxC=occa
figure;
subplot(2,1,1);plot(1:K1,exTc(tIdxC,1:K1),'b.',1:K1,double(approxTc(tIdxC,1:K1)), 'rx');title(['T(k) on the boundary at time=',num2str((tIdxC-
1)*dt)]);xlabel('node(k)');ylabel('T(k)');legend('Exact','Approximated')

```

Appendix B

```

subplot(2,1,2);plot(1:K1,ex_q(tIdxC,1:K1),'b',1:K1,double(approx_q(tIdxC,1:K1)), 'rx');title(['q(k) on the boundary at time=',num2str((tIdxC-
1)*dt)]) ;xlabel('node(k)');ylabel('q(k)');
text(1,min(double(approx_q(tIdxC,1:K1))), ['PPRDLBEM:',str,'m1,m2,m3,m4'],'; K2=',num2str(K2),'
;C1,C2,C3,C4=',C1,C2,C3,C4], 'VerticalAlignment','bottom');
end
else '-';end
if plotTvsTime=='y' & TrPt>0 & BVanyNumOcca>=1
figure;plot(timer,exTr,'b-',timeR,approxTr,'r-');legend('Exact','Approximated');
title('Approx. T vs time');xlabel('time');ylabel('T');
else '-';end

char(['min. B.E length=',num2str(min(L))',' Exact min. boundary T=',num2str(min(min(exTc)))]),
ending=clock;elpT=etime(ending,starting);
char(['starting=',num2str(starting)], ['Elapsed Time=',num2str(floor(elpT/3600))], 'hr
',num2str(floor(rem(elpT,3600)/60)), 'min ',num2str(rem(elpT,60)), 'sec'],
if onSound=='y' soundsc(wavread('C:\PROGRA~1\ONLINE~1\COMPUS~1\Support\WCINACT.WAV')));else '-';end
clear

%
%('Input/Output control');
Bmodel='2'; %Boundary model:'1' for circular segment,'2' for rectangular;
Br1=1;Br2=2;agll=0;agll3=pi/2;xOff=0;yOff=0;'circular seg.'; %give boundary geometry(Br1<Br2,0<agll3);
LC1=1;LC2=1;corY=0;'Rectangular'; %length & position of 1st corner;
m1,m2=5;TcA_PtLcA=5; %give boundary elements no. on C1 or C2 and no. of collo. arc or no. pt. on last collo. arc;
BC=['0','0','0','0']; %boundary cond.: '1' if T known on C(i), else '0' if q known;
l=[0.22,0.22]; %0<l<0.5,best accuracy:0.20<l<0.23
testProblem='1'; %select test prob.'1~5';
c1l=1;c1l2=1/2;c22=2;tc=3/4;%(c1l~c22)for prob.1~2;tc=time const.:-'prob.1':tc=any real(0/+ preferred),'prob.2 & 4':tc>0;tc=inf>1000 to turn to
steady state prob.;
tauT=2/5;tauh=2/3;alfa=2;%0<(tauT,tauh)<1,alfa>0;
freq=6; %real frequency for prob.1
recWellSpaceOutInPt='n', %'y'('yes') to recover internal well-space-out points,else 'n'('no');
TrA=1;PtLrA=2; %give integer no. of well-space out internal recovered arcs,
recSB_Apt='n';rBe=0.28;% 'y' to recover all pt. on smooth boundary(-0.5<rBe<0.5)/at adjacent(rBe=-0.5);rBe=distance from midpoint in %L(be) on element-
be;
recIOPt='n'; %'y' to recover all internal collocation pt., else 'n';n
recRanPt='y'; %'y' to recover randomly given pt.,
xiRan=[.1,.5,.9];etaRan=[.3,.3,.3];lambdaRan=[1,1,1];
%give random points & the corresponding lambdas;
ETRA='n'; %ETRA='y' for entire temporal range approx. else 'n'.
TN=5;dt=.06;RCT=0;%dt(>0)=small time interval btw. temporal nodes;RCT(0<=ratio<1)=position of collocated time from upper temporal
node;TN(integer>=2)=total no. of temporal nodes in polynomial approx. of func.;
timeR=[0.05:0.1:3,3:0.2:5];%timeR(>=0)=time at which T to be recovered;
'optimum RCT=0, TN=4;for:freq=0:min(L)/5<dt<TN/10,freq not zero:min(L)/5<dt<p1/(4*freq)';
minExBVer='y'; %'y' to turn error message if exact BV<0;
onSound='y'; %'y' to turn sound on,else 'n';
%check:
figBmodel='n'; %'y' to show fig. of boundary model with colloc. & recovered pt.;
plotBvalue='n',BVanyNumOcca=12;% 'y' to plot any few occasions of T(k) & q(k) on boundary;

```

Appendix B

```

plotTvsTime='n';      %%'y' to plot T vs time at selected computed pt.
copyResults='n';      %%'y' for tabulate results;
%
('Fool proof');
m1_m2=max(1,round(mi_m2));TcA_PtLcA=max(1,round(TcA_PtLcA));Tra=max(1,round(PtLra));
l=min(0.5,max(0,abs(1)));rBe=min(0.499,max(-0.5,rBe));
TN=max(2,round(abs(TN)));RCT=min(1,max(0,RCT));dt=unique(max(0.001,dt));timeR=unique(max(0,timeR));
c11=max(0.001,abs(c11));c12=abs(c12);c22=max(0.001,abs(c22));
if c11*c22<c12^2 error('error on: c11*c22<c12^2');else end
if testProblem='1' tc=max(0.001,tc);else '-' end
if ETFA='y' TSS=1;dt=max(timer)/(TN-1);else TSS=max(1,ceil(double(vpa(max(timer)/(TN-1)*dt),5)))) end

('1.Build-up boundary model')
switch Bmodel
case '1', LC1=Br2-Br1;LC2=ag13*Br2;str='Cir. seg.';
otherwise Br1=1E8;Br2=Br1+LC1;ag13=LC2/Br1;ag11=-ag13/2;xOff=-Br1+corX;yOff=LC2/2+corY;str='Rect.';
end
if LC1<=LC2 m_(1)=m1_m2;m_(2)=round(LC2/LC1*m1_m2);m_(3)=m1_m2;m_(4)=ceil(ag13*Br1/LC1*m1_m2);else
m_(1)=round(LC1/LC2*m1_m2);m_(2)=m1_m2;m_(3)=m1;m_(4)=ceil(ag13*Br1/LC2*m1_m2);end
M=sum(m_);K1=2*M;
('Element end pt. ');
for be=1:M
if be<=m_(1), Xe(be)=xOff+(Br1+(be-1)*(Br2-Br1)/m_(1))*cos(ag11);Ye(be)=yOff+(Br1+(be-1)*(Br2-Br1)/m_(1))*sin(ag11);
elseif be>=1+m_(1) & be<=sum(m_(1:2)),Xe(be)=xOff+Br2*cos(ag11+(be-m_(1)-1)*ag13/m_(2));Ye(be)=yOff+Br2*sin(ag11+(be-m_(1)-1)*ag13/m_(2));
elseif be>=1+sum(m_(1:2)) & be<=sum(m_(1:3)),
Xe(be)=xOff+(Br2-(Br2-Br1)*(be-m_(1)-m_(2)-1)/m_(3))*cos(ag11+ag13);Ye(be)=yOff+(Br2-(Br2-Br1)*(be-m_(1)-m_(2)-1)/m_(3))*sin(ag11+ag13);
else
Xe(be)=xOff+Br1*cos(ag11+(M+1-be)*ag13/m_(4));Ye(be)=yOff+Br1*sin(ag11+(M+1-be)*ag13/m_(4));
end
end
Xe(M+1)=Xe(1);Ye(M+1)=Ye(1);
('Element L,midpt.,boundary collo. pt.,n & B.C');
for be=1:M
if be==M Fbe=1;else Fbe=be+1;end;
L(be)=sqrt((Xe(Fbe)-Xe(be))^2+(Ye(Fbe)-Ye(be))^2);
midX(be)=(Xe(be)+Xe(Fbe))/2;
cosAlpha(be)=(Xe(Fbe)-Xe(be))/L(be); sinAlpha(be)=(Ye(Fbe)-Ye(be))/L(be);
n1(be)=sinAlpha(be); n2(be)=-cosAlpha(be); n(be,:)=[n1(be),n2(be),0];
node=[1,-L(be)*(1-2*(1)/2);1,L(be)*(1-2*(1)/2)]*midX(be);cosAlpha(be),sinAlpha(be)];
X(2*be-1)=node(1,1); Y(2*be-1)=node(1,2);
X(2*be)=node(2,1); Y(2*be)=node(2,2);
end
i2=0;for i1=1:4 if m_(i1)~=0 i2=i2+1;m(i2)=m_(i1);L_(i2)=L(sum(m(1:i2)));else end;end
tB=i2;
('Internal adjacent angle');
for be=1:M
if be=1 j=M;else j=be-1;end;
iaJAg1=real(acos(dot(n(j,:),n(be,:)))));
if sum(cross(n(j,:),n(be,:)))>=0 lambdaA(be)=(pi-iaJAg1)/(2*pi);
else lambdaA(be)=(pi+iaJAg1)/(2*pi);end
end
('Give well-spaced out internal collocation pt. ');
if LC1<=LC2 TcA=TcA_PtLcA;PtLcA=round(LC2/LC1*TcA_PtLcA);else PtLcA=TcA_PtLcA;TcA=round(LC1/LC2*TcA_PtLcA);end
K2=0;
for cA=1:TcA
cr(cA)=Br1+(Br2-Br1)*cA/(TcA+1); ticPt(cA)=round(1+(TcA+1)*(PtLcA-1)*cr(cA)/(Br1+Br2*TcA));

```

Appendix B

```

for icPtA=1:tiCpt(cA)
    K2=K2+1; lambdaC(K1+K2)=1;
    X(K1+K2)=xOff+cr(cA)*cos(agll+agll3*icPtA/(ticPt(cA)+1));
    Y(K1+K2)=yOff+cr(cA)*sin(agll+agll3*icPtA/(ticPt(cA)+1));
end
end

('2.Determine pt. to be computed')
('Give well-spaced out inetrnl points to be computed');
rPt=0; TirPt=0;
if recWellSpaceOutInPt=='y' & TrA>0 & PTLrA>0
    ('Give well-spaced out inetrnl points');
    for rA=1:TrA
        radA(rA)=Br1+(Br2-Br1)*rA/(TrA+1);
        tiPt(rA)=round(1+(TrA+1)*(PTLrA-1)*radA(rA)/(Br1+Br2*TrA));
        for ipt=1:tiPt(rA)
            rPt=rPt+1; TirPt=TirPt+1; lambdaR(rPt)=1;
            xiR(rPt)=xOff+radA(rA)*cos(agll+agll3*ipt/(tiPt(rA)+1)); etaR(rPt)=yOff+radA(rA)*sin(agll+agll3*ipt/(tiPt(rA)+1));
        end
    end
else TirPt=0; end
('Give well-spaced out pt. on smooth boundary or at adjacent');
if recSB_Apt=='y'
    TSB_Apt=M;
    for be=1:M
        rPt=rPt+1; element(rPt)=be;
        if rBe==0.5 lambdaR(rPt)=1/2; else lambdaR(rPt)=lambdaA(be); end
        xiR(rPt)=midX(be)-rBe*L(be)*n2(be); etaR(rPt)=midY(be)+rBe*L(be)*n1(be);
    end
else TSB_Apt=0; end
if recICPt=='y' ('Include internal collocation pt. to be computed'); TiCpt=K2;
for iCpt=1:TiCpt
    rPt=rPt+1; lambdaR(rPt)=1; xiR(rPt)=X(K1+iCpt); etaR(rPt)=Y(K1+iCpt);
end
else TiCpt=0; end
if recRanPt=='y' ('Include specified random pt. '); TranPt=length(xiRan);
for ranPt=1:TranPt rPt=rPt+1; xiR(rPt)=xiRan(ranPt); etaR(rPt)=etaRan(ranPt); lambdaR(rPt)=lambdaRan(ranPt); end
else TranPt=0; end
TrPt=TirPt+TiCpt+TSB_Apt+TranPt;

('3.Initial-BC, exact values at collo. pt., RBF, Integration const., temporal & differentiated temporal matrix')
('exact T(k) & q(k) at all collo. pt. at different times>=0; Initial-BC');
for k=1:K1+K2 x=X(k); y=Y(k);
    for tn=1:(TN-1)*TSS+1 t=dt*(tn-1);
        switch testProblem
            case '1', Non-classical 2D-unsteady Homogeneous Anisotropic';
                (x,y)>0; ra0>>0, (a1,a2,b1)=real; |b0|<<1; freq<2*pi/4/dt)= not too big; tc=any real time const.(>=0 preferred)';
                a0=6; ra1=1; ra2=0; b0=1; b1=-pi/6; k0=0;
                a3=- (a1*c11+a2*c12)/c22; if tc*tauT==1 tc=tc+1/2; else '-' ; end
                kl=real([-k0*c12+sqrt((c12^2-c11*c22)*k0^2+alfa*c22*(-tc+i*freq)*(1+tauh*(-tc+i*freq))/(1+taut*(-tc+i*freq)))]/c22);
                k2=imag([-k0*c12+sqrt((c12^2-c11*c22)*k0^2+alfa*c22*(-tc+i*freq)*(1+tauh*(-tc+i*freq))/(1+taut*(-tc+i*freq)))]/c22);
                T=a0+a1*x^2+a2*y+a3*y^2+b0*sin(k2*y+freq*t+b1)*exp(k0*x+kl*y-tc*t);
                dTdx=2*a1*x+a2*y+b0*sin(k2*y+freq*t+b1)*k0*exp(k0*x+kl*y-tc*t);
                dTdy=a2*x+2*a3*y+b0*cos(k2*y+freq*t+b1)*k2*exp(k0*x+kl*y-tc*t)+b0*sin(k2*y+freq*t+b1)*k1*exp(k0*x+kl*y-tc*t);
                dTdt=b0*[cos(k2*y+b1)*freq-sin(k2*y+b1)*tc]*exp(k0*x+kl*y);

```

Appendix B

```

d2Tdxdt0=b0*k0*(freq*cos(k2*y+b1)-tc*sin(k2*y+b1))*exp(k0*x+k1*y);
d2Tdydt0=b0*[ (freq*k1-k2*tc)*cos(k2*y+b1)-(freq*k2+k1*tc)*sin(k2*y+b1)]*exp(k0*x+k1*y);
case '2', Non-classical 2D-unsteady Homogeneous Anisotropic;
(a0,b0)>>>0;(a1,a2,b1,b4)=real,[c0]<1;tc>0;
a0=1;a1=0;a2=1;b0=0;b1=1;b4=0;c0=1;
a3=- (a1*c11+a2*c12)/c22;b2=-c12/c22*b1;b3=sqrt(c11*c22-c12^2)/c22*b1;tc=1/tauh;
T=a0*a1*x^2+a2*x*y+a3*y^2+[b0*c0*exp(b1*x+b2*y)*sin(b3*y+b4)]*exp(-tc*t);
dTdx=2*a1*x+a2*y+c0*b1*sin(b3*y+b4)*exp(b1*x+b2*y-tc*t);
dTdy=a2*x+2*a3*y+c0*b2*sin(b3*y+b4)*exp(b1*x+b2*y-tc*t)+c0*cos(b3*y+b4)*b3*exp(b1*x+b2*y-tc*t);
d2Tdxdt0=-c0*b1*tc*sin(b3*y+b4)*sin(b3*y+b4);
d2Tdydt0=-tc*c0*(b2*sin(b3*y+b4)+b3*cos(b3*y+b4))*exp(b1*x+b2*y);
case '3', Non-classical 2D-unsteady Homogeneous Isoatropic;(x,y)>0;
('fixed:');alfa=1;c11=1;c22=1;c12=0;taut=1/2;tauh=1/4;alfa=1;
T=(1+x*y+(1+x)*sin(sqrt(3/2)*y)*exp(-t);
dTdx=y+sin(sqrt(3/2)*y)*exp(-t);dTdy=x+sqrt(3/2)*(1+x)*cos(sqrt(3/2)*y)*exp(-t);
dTdt0=- (1+x)*sin(sqrt(3/2)*y);d2Tdxdt0=-sin(sqrt(3/2)*y);d2Tdydt0=-sqrt(3/2)*(1+x)*cos(sqrt(3/2)*y);
case '4', Classical 2D-unsteady Homogeneous Isotropic';
'alfa>(b1^2/tc)>0 & alfa>(b2^2/tc)>0;tc>0;a0>0,(a1,a2,b1,b2)=real';
a0=1;a1=1;a2=2;b1=1;b2=1/2;alfa=1.5;
'fixed:':alfa=max(10.1,alfa,b1^2/tc,b2^2/tc);c11=1;c22=1;c12=0;taut=0;tauh=0;
T=a0*x*y+[a1*sin(b1*x+sqrt(alfa*tc-b1^2)*y)+a2*cos(b2*x+sqrt(alfa*tc-b2^2)*y)]*exp(-tc*t);
dTdx=y+(a1*cos(b1*x*(alfa*tc-b1^2)^(1/2)*y)+b1-a2*sin(b2*x+(alfa*tc-b2^2)^(1/2)*y)*b2)*exp(-tc*t);
dTdy=x+(a1*cos(b1*x*(alfa*tc-b1^2)^(1/2)*y)*(alfa*tc-b1^2)^(1/2)-a2*sin(b2*x+(alfa*tc-b2^2)^(1/2)*y)*(alfa*tc-b2^2)^(1/2))*exp(-tc*t);
d2Tdxdt0=-tc*(a1*sin(b1*x+sqrt(alfa*tc-b1^2)*y)+a2*cos(b2*x+sqrt(alfa*tc-b2^2)*y))-al*b1*cos(b1*x+sqrt(alfa*tc-b1^2)*y);
d2Tdydt0=tc*(a2*sin(b2*x+sqrt(alfa*tc-b2^2)*y)*sqrt(alfa*tc-b2^2)-al*cos(b1*x+sqrt(alfa*tc-b1^2)*y)*sqrt(alfa*tc-b1^2));
otherwise,'Classical 2D-steady Homogeneous Anisotropic,a0>0';
a0=4;
c11=1;c12=2;c22=5;alfa=0;taut=0;tauh=0;('fixed');
T=a0+5*x^2-y^2:dTdx=10*x:dTdy=-2*y:dTdt0=0;d2Tdxdt0=0;d2Tdydt0=0;
end
if k<=K1 be=round(k/2);ex_gnC(tn,k)=-[n1(be)*(c11*dTdx+c12*dTdy)+n2(be)*(c12*dTdx+c22*dTdy)];else ex_gnC(tn,k)=0:end
end
d2Tdt0cSS(1,k)=dTdt0;
if k<=K1 be=round(k/2);dqdt0cSS(1,k)=-[n1(be)*(c11*d2Tdxdt0+c12*d2Tdydt0)+n2(be)*(c12*d2Tdxdt0+c22*d2Tdydt0)];else dqdt0cSS(1,k)=0:end
end
if minExBVerr=='y'
if testProblem=='1' & tc<0 '-';
elseif min(min(exTnC))<0 soundsc(wavread('C:\PROGRA~1\NETMEE~1\RINGIN.WAV')));error(['negative value on boundary, min.
T=',num2str(min(min(exTnC)))]);
else '-';end
else '-';end
('B.C');
for k=1:K1+K2
if k<=2*m(1)
elseif k>=1+2*m(1) & k<=2*sum(m(1:2))
elseif k>=1+2*sum(m(1:2)) & k<=2*sum(m(1:3)) if BC(3)=='1' gama(k)=1;else gama(k)=0;end
elseif k>=1+2*sum(m(1:3)) & k<=K1
elseif k>=1+2*sum(m(1:3)) & k<=K1
else
end
end
for r=1:TN if r==1 gamal(r,:)=ones(1,K1+K2);else gamal(r,:)=gama;end; end
gama2=gama1;gama2(2:TN,:)=ones(TN-1,K1+K2)-gama2(2:TN,:);clear gama

```


Appendix B

```

DTM1=DTM1+DTM2;DDTM1=DDTM1+DDTM2;
    else '-';end
end
TMC(s-1,r)=TM1/((-1)^(1+TN-r))*factorial(r-1)*factorial(1+TN-r)*dt^TN];
DTMC(s-1,r)=DTM1/((-1)^(1+TN-r))*factorial(r-1)*factorial(1+TN-r)*dt^TN];
DDTMC(s-1,r)=DDTM1/((-1)^(1+TN-r))*factorial(r-1)*factorial(1+TN-r)*dt^TN];
end
end
('Temporal & diff. temporal matrix at temporal nodes');
for s=1:TN t_c=(s-1)*dt;%at temporal node
    for r=1:1+TN TM1=1;DTM1=0;
        for r1=1:1+TN t_r1=(r1-1)*dt;DTM2=1;
            if r1~=r
                TM1=(t_c-t_r1)*TM1;
                for r2=1:1+TN t_r2=(r2-1)*dt;
                    if r2~=r & r2~=r1
                        DTM2=(t_c-t_r2)*DTM2;
                    else '-';end
                end
            end
            DTM1=DTM1+DTM2;
        else '-';end
    end
    TMn(s,r)=TM1/((-1)^(1+TN-r))*factorial(r-1)*factorial(1+TN-r)*dt^TN];
    DTMn(s,r)=DTM1/((-1)^(1+TN-r))*factorial(r-1)*factorial(1+TN-r)*dt^TN];
end
end
('coefficients');
alpPsiSig=alfa*Psi*sigma;
xDTM=DTMC(:,1:TN)-DTMC(:,1+TN)*DTMn(1,1:TN)/DTMn(1,1+TN)+tauh*[DDTMC(:,1:TN)-DDTMC(:,1+TN)*DTMn(1,1+TN)]];
xTM=TMc(:,1:TN)-TMC(:,1+TN)*DTMn(1,1:TN)/DTMn(1,1+TN)+taut*[DTMC(:,1:TN)-DTMC(:,1+TN)*DTMn(1,1+TN)]];
for p=1:K1+K2
    Phi00((p-1)*(TN-1)+1:p*(TN-1),(p-1)*TN+1:p*TN)=lambdaC(p)*xTM;
    for k=1:K1+K2
        Phi1((p-1)*(TN-1)+1:p*(TN-1),(k-1)*TN+1:k*TN)=alpPsiSig(p,k)*xDTM+I(p,k)*xTM;
        Phi2((p-1)*(TN-1)+1:p*(TN-1),(k-1)*TN+1:k*TN)=II(p,k)*xTM;
    end
end
('4.assemble overall sys. of eqn. & solve unknown for t>0')
a=-[(Phi1-Phi00)*diag(gama2(:))+Phi2*diag(gama1(:))];for k=1:K1+K2 a(:,(k-1)*(TN-1)+1)=[];end
inv_a=inv(a);
appTnC(1,:)=exTnC(1,:);app_qnC(1,:)=ex_qnC(1,:);
for SS=1:TSS
    if SS~=1
        dtdt0cSS(SS,:)=DTMn(TN,1+TN)*dtdt0cSS(SS-1,:)/DTMn(TN,1+TN)+[DTMn(TN,1+TN)-DTMn(TN,1:TN)]*appTnC(1,1+TN)]*appTnC(1,1+(SS-1)*(TN-1):1+(SS-1)*TN);
        dtdt0cSS(SS,:)=DTMn(TN,1+TN)*dtdt0cSS(SS-1,:)/DTMn(TN,1+TN)+[DTMn(TN,1:TN)-DTMn(TN,1+TN)]*app_qnC(1,1+(SS-1)*(TN-1):1+(SS-1)*TN);
    else;end
    Phi0=[(TMc(:,1+TN)+taut*DTMC(:,1+TN))*(I*dtdt0cSS(SS,:)+II*dqdt0cSS(SS,:))'-
        (lambdaC.*dtdt0cSS(SS,:))'.']+(DTMC(:,1+TN)+tauh*DDTMC(:,1+TN))*(alpPsiSig*dtdt0cSS(SS,:))'.']./DTMn(1,1+TN);
    appTnC((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=exTnC((TN-1)*(SS-1)+2:SS*(TN-1)+1,:).*gama1(2:TN,:);
    app_qnC((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=ex_qnC((TN-1)*(SS-1)+2:SS*(TN-1)+1,:).*gama2(2:TN,:);
    z1=appTnC((TN-1)*(SS-1)+1:SS*(TN-1)+1,:);z2=app_qnC((TN-1)*(SS-1)+1:SS*(TN-1)+1,:);
    b=Phi0(:)+(Phi1-Phi00)*z1(:)+Phi2*z2(:);
    unknown=reshape(inv_a*b,[TN-1 K1+K2]);
end

```


Appendix B

```

appTnC((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=appfnc((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)+unknown.*gamma2(2:TN,:);
app_qnC((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)=app_qnC((TN-1)*(SS-1)+2:SS*(TN-1)+1,:)+unknown.*gamma1(2:TN,:);
end

clear I II Psi alPsiSig Phi00 Phi0 Phi1 Phi2 a b unknown
syms x y t
('5,Compute approx.T at selected pt. & time')
if TrPt>0
    ('Exact T at selected time,initial T & dT/dt of pt. to be computed');
    switch testProblem
    case '1','Non-classical 2D-unsteady Homogeneous Anisotropic';
        T=a0*al*x^2+a2*x*y+a3*y^2+b0*sin(k2*y+freq*t-b1)*exp(k0*x+kl*y-tc*t);
        dTdt0=b0*[cos(k2*y+b1)*freq-sin(k2*y+b1)*tc]*exp(k0*x+kl*y);
    case '2','Non-classical 2D-unsteady Homogeneous Anisotropic';
        T=a0*al*x^2+a2*x*y+a3*y^2+b0+c0*exp(b1*x+b2*y)*sin(b3*y+b4)]*exp(-tc*t);
        dTdt0=-tc*(b0+c0*exp(b1*x+b2*y)*sin(b3*y+b4));
    case '3','Non-classical 2D-unsteady Homogeneous Isoatropic;(x,y)>0';
        T=1+x*y+(1+x)*sin(sqrt(3/2)*y)*exp(-t);
        dTdt0=-(1+x)*sin(sqrt(3/2)*y);d2Tdxdt0=-sin(sqrt(3/2)*y);d2Tdydt0=-sqrt(3/2)*(1+x)*cos(sqrt(3/2)*y);
    case '4','Classical 2D-unsteady Homogeneous Isotropic';
        T=a0*x*y+al*sin(b1*x+sqrt(alfa*tc-b1^2)*y)+a2*cos(b2*x+sqrt(alfa*tc-b2^2)*y)]*exp(-tc*t);
        dTdt0=-tc*(al*sin(b1*x+sqrt(alfa*tc-b1^2)*y)+a2*cos(b2*x+sqrt(alfa*tc-b2^2)*y));
    otherwise,'Classical 2D-steady Homogeneous Anisotropic,a0>>0';
        T=a0+5*x^2-y^2;dTdt0=0;d2Tdxdt0=0;d2Tdydt0=0;
    end
    for rPt=1:TrPt x=xiR(rPt);y=etaR(rPt); t=0;T0rSS(1,rPt)=subs(T);dTdt0rSS(1,rPt)=subs(dTdt0);
    for tr=1:length(timer) t=timer(tr);exTr(tr,rPt)=subs(T);
    end
end
('Integration const. I & II');
for rPt=1:TrPt xi=xiR(rPt);eta=etaR(rPt);
    for k=1:K1+K2,be=round(k/2); if rem(k,2)=1 eo=2;else eo=1;end
    if k<=K1
        A=4*[c11*n1(be)+c12*n2(be)]*(midY(be)-eta)*(c22*n2(be)+c12*n1(be))*(midX(be)-
        xi)]/L(be)/(c22*n2(be)^2+2*c12*n2(be)*n1(be)+c11*n1(be)^2);
        B=4*[c22*(midX(be)-xi)^2-2*c12*(midX(be)-xi)*(midY(be)-eta)+c11*(midY(be)-
        eta)^2]/L(be)^2/[c22*n2(be)^2+2*c12*n2(be)*n1(be)+c11*n1(be)^2];
        C=sqrt(c11*c22-c12^2)/[2*pi*L(be)*(1-l(1)-l(2))]*[(midX(be)-xi)*n1(be)+(midY(be)-
        eta)*n2(be)]/(c22*n2(be)^2+2*c12*n2(be)*n1(be)+c11*n1(be)^2);
        D=L(be)/[16*pi*sqrt(c11*c22-c12^2)*(1-l(1)-l(2))]^2*log(L(be)^2*(c22*n2(be)^2+2*c12*n2(be)*n1(be)+c11*n1(be)^2)/(4*c22));
        E=L(be)/[16*pi*sqrt(c11*c22-c12^2)*(1-l(1)-l(2))]^2*log(L(be)^2*(c22*n2(be)^2+2*c12*n2(be)*n1(be)+c11*n1(be)^2)/(4*c22));
        if 4*B-A^2<1E-12
            I(rPt,k)=0;
            if abs(A^2-4)<1e-12
                II(rPt,k)=2*(1-2*l(eo))*D+2*E*[2*(1-2*l(eo))*log(2)+4*l(eo)-(2+(-1)^eo*A/2)];
                II(rPt,k)=(2*D*(1-2*l(eo))+E/8*[(2-8*l(eo)+(-1)^eo*A)*(2+(-1)^eo*A)*log((2+(-1)^eo*A)^2)+(6-8*l(eo)+(-1)^eo*A)*(2-(-
                1)^eo*A)*log((2-(-1)^eo*A)^2)-32*(1-2*l(eo))*log(2)+1)-8*(-1)^eo*A];
            end
        else
            I(rPt,k)=C*[1/2*log((1-(-1)^eo*A+B)/(1+(-1)^eo*A+B)))+(2*(1-2*l(eo))+(-1)^eo*A)*atan2(sqrt(4*B-A^2),B-1)/sqrt(4*B-A^2)];
            II(rPt,k)=2*(1-2*l(eo))*D+E/4*[A^2+2*(-1)^eo*A-2*B+2-4*l(eo)*(2+(-1)^eo*A))*log(1+(-1)^eo*A)-A^2+2*(-1)^eo*A-2*B-
            6+4*l(eo)*(2-(-1)^eo*A))*log(1-(-1)^eo*A+B)-4*(4-8*l(eo)+(-1)^eo*A)+2*(2-4*l(eo)+(-1)^eo*A)*sqrt(4*B-A^2)*atan2(sqrt(4*B-A^2),B-1)];
        end
    else
        I(rPt,k)=0;II(rPt,k)=0;
    end
end

```

A25

Appendix B

```

char('Error=',num2str(errTr(:,1:TirPt))),
char('% Error=',num2str(percentErrTr(:,1:TirPt))),
else '-',end
if recSB_Apt== 'y' ('Smooth boundary pt./at adjacent')
char('Exact T=',['element>',num2str(element(1+TirPt:TirPt+TSB_Apt))],['xi>
',num2str(xiR(1+TirPt:TirPt+TSB_Apt))],['t/eta>',num2str(etaR(1+TirPt:TirPt+TSB_Apt))],num2str([timer.',exTr(:,1+TirPt:TirPt+TSB_Apt)])),
char('computed T=',num2str(appTr(:,1+TirPt:TirPt+TSB_Apt))),
char('Error=',num2str(errTr(:,1+TirPt:TirPt+TSB_Apt))),
char('% Error=',num2str(percentErrTr(:,1+TirPt:TirPt+TSB_Apt))),
else '-',end
if recICPt== 'y' ('Internal collocation pt.')
char('Exact T',[ 'xi>
',num2str(xiR(1+TirPt+TSB_Apt:TirPt+TSB_Apt+TiCpt))],['t/eta>',num2str(etaR(1+TirPt+TSB_Apt:TirPt+TSB_Apt+TiCpt))],num2str([timer.',exTr(:,1+TirPt+TSB_
Apt:TirPt+TSB_Apt+TiCpt)])),
char('computed T=',num2str(appTr(:,1+TirPt+TSB_Apt:TirPt+TSB_Apt+TiCpt))),
char('Error=',num2str(errTr(:,1+TirPt+TSB_Apt:TirPt+TSB_Apt+TiCpt))),
char('% Error=',num2str(percentErrTr(:,1+TirPt+TSB_Apt:TirPt+TSB_Apt+TiCpt))),
else '-',end
if recRanPt== 'y' ('Randomly given pt.')
char('Exact T',[ 'xi >',num2str(xiR(1+TirPt+TSB_Apt+TiCpt:TrPt))],['t /eta
>',num2str(etaR(1+TirPt+TSB_Apt+TiCpt:TrPt))],num2str([timer.',exTr(:,1+TirPt+TSB_Apt+TiCpt:TrPt)])),
char('computed T=',num2str(appTr(:,1+TirPt+TSB_Apt+TiCpt:TrPt))),
char('Error=',num2str(errTr(:,1+TirPt+TSB_Apt+TiCpt:TrPt))),
char('% Error=',num2str(percentErrTr(:,1+TirPt+TSB_Apt+TiCpt:TrPt))),
else '-',end
if copyResults== 'y', 't', timer.', 'xi', 'xi.', 'eta', 'eta.', 'ex.T', exTr(:), 'approx.T', appTr(:), 'err.', errTr(:), '%err.', percentErrTr(:), else '-';end
['Tol. %err.=%',num2str(sum(sum(percentErrTr))),']; avg.%err.=%', num2str(sum(sum(percentErrTr)))/(TrPt*length(timeR))),']; norm of
err.=',num2str(sqrt(sum(sum(errTr.^2))/(TrPt*length(timeR))))),']; avg. abs. Err.=', num2str(sum(sum(abs(errTr)))/(TrPt*length(timeR))))),
else xi=-1e16;eta=-1e16; No results shown',end

('Checkig');
if figBmodel== 'y' noB={ '\Gamma_1', '\Gamma_2', '\Gamma_3', '\Gamma_4' };
figure;plot(Xe,Ye,'k-',X,Y,'b.',xiR,etaR,'rx');title('Boundary model');xlabel('x');ylabel('y');axis equal;axis([min(Xe)-abs(max(Xe)-min(Xe))/10
max(Xe)+abs(max(Xe)-min(Xe))/10 min(Ye)-abs(max(Ye)-min(Ye))/10 max(Ye)+abs(max(Ye)-min(Ye))/10]);legend('boundary elements','DLE collocation
pt.','Points under tested');
for il=1:tB i2=sum(m(1:i1-1))+round(m(i1)/2);text(midX(i2),midY(i2),noB(il));end
else '-',end
if plotTvsTime== 'y' & TrPt>0 & BVanyNumOcca>=1
figure;plot(timer,extr,'b-',timer,appTr,'rx');title('Approx. T vs time');xlabel('time');ylabel('T');legend('Exact','Approximated');
text(min(timer),min(min(min(exTr)),min(min(appTr))),['PDRDLBEM:Prob.',char(testProblem),',',str,'m=(',num2str(m),'), (M,K1,K2,dt,TN,RCT)=( ',num2str([
M,K1,K2,dt,TN,RCT]),', B.C= ',BC(1:tB)], 'VerticalAlignment','bottom');
else '-',end
if plotBValue== 'y' & BVanyNumOcca>=1
if BVanyNumOcca<(TN-1)*TSS
for i=1:BVanyNumOcca occa(i)=ceil(i*(TN-1)*TSS/BVanyNumOcca);end;
else occa=2:1:(TN-1)*TSS;end
for tn=occa
figure;
subplot(2,1,1);plot(1:K1,exTnC(tn,1:K1),'b.',1:K1,double(appTnC(tn,1:K1)),'rx');title(['T(k) on the boundary at time=',num2str((tn-
1)*dt)])>ylabel('node(k)');ylabel('T(k)');legend('Exact','Approximated')
subplot(2,1,2);plot(1:K1,ex_gnC(tn,1:K1),'b.',1:K1,double(app_gnC(tn,1:K1)),'rx');title(['q(k) on the boundary at time=',num2str((tn-
1)*dt)])>ylabel('node(k)');ylabel('q(k)');
text(1,min(min(app_gnC(tn,:)),min(ex_gnC(tn,:))),min(ex_gnC(tn,:)),['PDRDLBEM:Prob.',char(testProblem),',',str,'m=(',num2str(m),'), (M,K1,K2,dt,TN,RCT)=( ',num2str([M,K
1,K2,dt,TN,RCT]),', B.C= ',BC(1:tB)], 'VerticalAlignment','bottom');

```

Appendix B

```

end
else '-';end
char(['min. B.E length=',num2str(min(L)),',', Exact min. boundary T=',num2str(min(min(exTnC)))],
ending=clock;elpT=etime(ending,starting);
char(['starting=',num2str(starting)],['Elapsed Time=',num2str(floor(elpT/3600)),'hr
',num2str(floor(rem(elpT,3600)/60)),'min ',num2str(rem(elpT,60)),'sec'],
if onSound=='y' soundsc(wavread('C:\PROGRAM~1\ONLINE~1\COMPUS~1\Support\WCINACT.WAV'));else '-';end
clear

```

B3 MATLAB Code for Chapter 4

```

starting=clock;
format long
syms x y u w
%
('Input/Output control');
Bmodel='1';
Br1=0.5;Br2=1.5;ag11=pi;ag12=-pi;xOff=0;'circular seg. ';%give radius & angles for circular seg.(Br1<Br2,0<ag11<=pi,-pi<=ag12<0);
lefB=0;rigB=1;lowB=-0.5;topB=.5;'Rectangular';%give left/right/low/top boundary positions(rigB>lefB,topB>0,lowB<0)
min_m=4;%give no. boundary elements no. to the shortest boundary;
BC=['0','0','1','1','0','0'];%boundary cond. of 1st few no. of boundaries(start from right of y>0):'1' if T known on C(i),else '0' if q known;
l=[0.22,0.22];%Collo. pt. position:- 0<(l(1),l(2))<0.5,best accuracy:0.20<(l(1),l(2))<0.23
wellSpOutInPt='y';% 'y' to recover internal well-space out points;
TrLn=1;PthrLn=4;%TrLn=Integer no. of total well-space out internal line,PthrLn=Integer no. of total pt. at last internal line
recSB_APT='n';rBe=-0.28;% 'y' to recover all pt. on smooth boundary(-0.5<rBe<0.5)/at adjacent(rBe=-0.5);rBe=distance from midpoint in %L(be) on
element-be;
recRanPt='y';% 'y' to recover randomly given pt.,
xiRan=[-1.3,-.8,.8,1.3];etaRan=[.01,-.01,.01,-.01];%give random points
lambdaRan=[1,1,1,1];%the corresponding lambdas
testProb='1';%give type of test problem.1~4;
gama=1/3;%gama=real>0; perfect case if gama=inf;
c11=[5/12,1/2];c12=[1/6,1/2];c22=[1/3,1];'c11*c22>c12^2';%cij(1~2)=thermal conductivities of 1st & 2nd materials>0;
onSound='y';% 'y' to turn sound on,else 'n';
minExValueChk='y';% 'y' to check the min. value of exact T at colloc. and selected pt.;
%check:
figBmodel='n';% 'y' to show boundary model with colloc. & recovered pt.;
plotBvalue='n';% 'y' to plot approx. & exact value on boundary;
copyResults='y';% 'y' for tabulate results;
%
('Fool proof');
Br1=abs(Br1);Br2=max(abs(Br2),Br1+0.1);ag11=min(pi,max(pi/10,abs(ag11)));ag12=max(-pi,min(-pi/10,-abs(ag12)));
rigB=max(abs(rigB),lefB+0.1);lowB=-max(abs(lowB),0.1);topB=max(abs(topB),0.1);
min_m=max(1,round(min_m));l=min(0.5,max(0,abs(l)));rBe=min(0.499,max(-0.5,rBe));
TrLn=round(abs(TrLn));PthrLn=round(abs(PthrLn));
gama=max(0.001,abs(gama));c11=max(0.001,abs(c11));c12=abs(c12);c22=max(0.001,abs(c22));
('1.select test problems')
sCij(1)=sqrt(c11(1)*c22(1)-c12(1)^2);sCij(2)=sqrt(c11(2)*c22(2)-c12(2)^2);
switch testProb,
case '1','Imperfect,uni/bimaterial,iso/anisotropic';

```

Appendix B

```

al=0;a2=1;a3=0;a4=0;'al~a4=real';
T(1)=al+scij(2)*exp(gama*y*(scij(2)-scij(1))/c22(1)/scij(2)+a4)*[a2*cos(gama*(scij(1)-scij(2))*(x-
y*c12(1)/c22(1))/scij(1)/scij(2))+a3*sin(gama*(scij(1)-scij(2))*(x-y*c12(1)/c22(1))/scij(1)/scij(2))];
T(2)=al+scij(1)*exp(gama*y*(scij(1)-scij(2))/c22(1)/a4)*[a2*cos(gama*(scij(1)-scij(2))*(x-
y*c12(2)/c22(2))/scij(1)/scij(2))+a3*sin(gama*(scij(1)-scij(2))*(x-y*c12(2)/c22(2))/scij(1)/scij(2))];
case '2','Imperfect,bimaterial,anisotropic';
'fixed':gama=1/2;c11(1)=1/2;c12(1)=0;c22(1)=1/3;c12(2)=0;c22(2)=1/3;scij(1)=1/2;scij(2)=1/3;
T(1)=[3*cos(y)+2*sin(y)]*exp(-x);
T(2)=[cos(y)+3*sin(y)]*exp(-x);
otherwise, 'Perfect,uni/bimaterial,iso/anisotropic';
al=1;a2=5;a3(1)=2;'all real';
'fixed':gama=inf;a3(2)=[2*a2*(c12(1)-c12(2))+c22(1)*a3(1)]/c22(2);a4(1)=-[(a2*c11(1)+a3(1)*c12(1))/c22(1)];a4(2)=-
[(a2*c11(2)+a3(2)*c12(2))/c22(2)];
T(1)=al+a2*x^2+a3(1)*x*y+a4(1)*y^2;
T(2)=al+a2*x^2+a3(2)*x*y+a4(2)*y^2;
end
CsCij(1)=CsCij(2);CsCij(2)=scij(1);tau(1)=[-c12(1)+i*scij(1)]/c22(1);tau(2)=[-c12(2)+i*scij(2)]/c22(2);
if c11(1)*c22(1)<c12(1)^2 | c11(2)*c22(2)<c12(2)^2 error('error at:c11*c22<c12^2');else;end
if recRanPt=='y' if length(xiRan)==length(etaRan) & length(lambdaRan)==length(etaRan);else error('error at:dimensions of xiRan,etaRan & lambdaRan not
match');end;else;end
dtdx(1)=diff(T(1),x);dtdy(1)=diff(T(1),y);dtdx(2)=diff(T(2),x);dtdy(2)=diff(T(2),y);

('2.Build-up boundary model & colloc. pt.')
switch Bmodel
case '1', LC(1)=agl1*Br2;LC(2)=Br2-Br1;LC(3)=agl1*Br1;LC(4)=abs(agl2*Br1);LC(5)=LC(2);LC(6)=abs(agl2*Br2);str='Cir. seg.';
otherwise LC(1)=topB;LC(2)=rigB-lefB;LC(3)=LC(1);LC(4)=abs(lowB);LC(5)=LC(2);LC(6)=LC(4);str='Rect.';
end
if Bmodel=='1' & (agl1-agl2)==2*pi LC(2)=0;LC(5)=0;else;end
minLC=LC(1);
for il=2:6 if LC(il)>0 & LC(il)<minLC;minLC=LC(il);else;end;end
for il=1:6 m_(il)=round(min_m*LC(il)/minLC);end
if Bmodel=='1' & (agl1-agl2)==2*pi m_(2)=1;m_(5)=1;else;end
('Element end pt. ');
for be=1:sum(m_)
if Bmodel=='1'
if be<=m_(1), Xe(be)=xOff+Br2*cos((be-1)*agl1/m_(1));Ye(be)=Br2*sin((be-1)*agl1/m_(1));
elseif be>m_(1) & be<=sum(m_(1:2)),Xe(be)=xOff+[Br2-(be-m_(1)-1)*cos(agl1)/m_(2)]*cos(agl1);Ye(be)=[Br2-(be-m_(1)-1)*(Br2-
Br1)/m_(2)]*sin(agl1);
elseif be>sum(m_(1:2)) & be<=sum(m_(1:3)),
Xe(be)=xOff+Br1*cos((be-sum(m_(1:2))-1)*agl1/m_(3));Ye(be)=Br1*sin(agl1-(be-sum(m_(1:2))-1)*agl1/m_(3));
elseif be>sum(m_(1:3)) & be<=sum(m_(1:4)),
Xe(be)=xOff+Br1*cos((be-sum(m_(1:3))-1)*agl2/m_(4));Ye(be)=Br1*sin((be-sum(m_(1:3))-1)*agl2/m_(4));
be>sum(m_(1:4)) & be<=sum(m_(1:5)),
Xe(be)=xOff+[Br1+(be-sum(m_(1:4))-1)*(Br2-Br1)/m_(5)]*cos(agl2);Ye(be)=[Br1+(be-sum(m_(1:4))-1)*(Br2-Br1)/m_(5)]*sin(agl2);
Xe(be)=xOff+Br2*cos((sum(m_)-be+1)*agl2/m_(6));
Ye(be)=Br2*sin((sum(m_)-be+1)*agl2/m_(6));
end
else
if be<=m_(1), Xe(be)=rigB;Ye(be)=(be-1)*topB/m_(1);
if
be>m_(1) & be<=sum(m_(1:2)),Xe(be)=rigB-(be-1-m_(1))*(rigB-lefB)/m_(2);Ye(be)=topB;
elseif
be>sum(m_(1:2)) & be<=sum(m_(1:3)),
Xe(be)=lefB;Ye(be)=topB-(be-1-sum(m_(1:2)))*topB/m_(3);
elseif
be>sum(m_(1:3)) & be<=sum(m_(1:4)),
Xe(be)=lefB;Ye(be)=(be-1-sum(m_(1:3)))*lowB/m_(4);

```

Appendix B

```

elseif be>sum(m_(1:4)) & be<=sum(m_(1:5)),
    Xe(be)=lefB+(be-1-sum(m_(1:4)))*(rigB-lefB)/m_(5);Ye(be)=lowB;
    Xe(be)=rigB;Ye(be)=(sum(m_)-be+1)*lowB/m_(6);
else
end
end
Xe(sum(m_)+1)=Xe(1);Ye(sum(m_)+1)=Ye(1);
('Element L,midpt.,n,boundary collo. pt.,nodal pt. mtl. property');
be=0;mm(1:6)=0;
for be=1:sum(m_)
    if Bmodel=='1' & agll-agl2==2*pi & [(be>m_(1) & be<=sum(m_(1:2))) | (be>sum(m_(1:4)) & be<=sum(m_(1:5)))];
    else
        if be==sum(m_) Nxbe_='1';else Nxbe_=-be_+1;end;be=be+1;
        L(be)=sqrt((Xe(Nxbe_)-Xe(be_-))^(2)+(Ye(Nxbe_)-Ye(be_-))^2);
        midX(be)=(Xe(be_)+Xe(Nxbe_))/2; midY(be)=(Ye(be_)+Ye(Nxbe_))/2;
        n1(be)=(Ye(Nxbe_)-Ye(be_-))/L(be); n2(be)=-Xe(Nxbe_)-Xe(be_-))/L(be); n(be,:)=n1(be),n2(be),0];
        node=[1,-L(be)*(1-2*(1))/2;1,L(be)*(1-2*(1))/2]*[midX(be),midY(be);-n2(be),n1(be)];%matrix
        X(2*be-1)=node(1,1); Y(2*be-1)=node(1,2);
        X(2*be)=node(2,1); Y(2*be)=node(2,2);
    end
end
i2=0;
for i1=1:6
    if m_(i1)==0 | [Bmodel=='1' & agll-agl2==2*pi & (i1==2 | i1==5)];
    else i2=i2+1;m(i2)=m_(i1);L_(i2)=L(sum(m(1:i2)));
    end
end
tB=i2:M=sum(m);mC(1:2*sum(m(1:i2/2)))=1;mC(1+2*sum(m(1:i2/2)):2*M)=2;
('Internal adjacent angle');
for be=1:M
    if be==1 j_=M;
    elseif Bmodel=='1' & (agll-agl2)==2*pi & Br1~=0 & be==m(1)+1 j_=sum(m(1:3));
    elseif Bmodel=='1' & (agll-agl2)==2*pi & Br1~=0 & be==sum(m(1:3))+1 j_=m(1);
    else j_=-be-1;end;
    iAjAg1=real(acos(dot(n(j_,:),n(be,:)))));
    if sum(cross(n(j_,:),n(be,:)))>0 lambdaA(be)=(pi-iAjAg1)/(2*pi);
    else lambdaA(be)=(pi+iAjAg1)/(2*pi);end
end
('3.Determine pt. to be recovered')
rPt=0;
('Give well-spaced out inetrnl points');
TirPt=0;
if wellSpoutInPt=='y' & TrLn>0 & PtlrLn>0
    for rLn=1:TrLn
        if Bmodel=='1'
            r(rLn)=Br1+(Br2-Br1)*rLn/(TrLn+1);
            tiPt(rLn)=round((1+(TrLn+1)*(PtlrLn-1)*r(rLn))/(Br1+Br2*TrLn));
            for ipt=1:tiPt(rLn)
                rPt=rPt+1;TirPt=TirPt+1;lambdaR(rPt)=1;
                xlr(rPt)=xOff+r(rLn)*cos(agl2+(ipt-1/2)*(agll-agl2)/tiPt(rLn)); etaR(rPt)=r(rLn)*sin(agl2+(ipt-1/2)*(agll-agl2)/tiPt(rLn));
            end
        else
            for ipt=1:PtlrLn

```

Appendix B

```

rPt=rPt+1;TrPt=TrPt+1;lambdaR(rPt)=1;
xiR(rPt)=lefB+rLn*(rigB-lefB)/(TrLn+1);etaR(rPt)=lowB+iPt*(topB-lowB)/(PtLrLn+1);
end
end
else ;end
('Give well-spaced out pt. on smooth boundary or at adjacent');
if recSB_Apt=='y'
    TSB_Apt=M;
    for be=1:M
        rPt=rPt+1;element(rPt)=be;
        if rBe==0.5 lambdaR(rPt)=1/2;else lambdaR(rPt)=lambdaA(be);end
        xiR(rPt)=midX(be)-rBe*L(be)*n2(be);etaR(rPt)=midY(be)+rBe*L(be)*n1(be);
    end
else TSB_Apt=0;end
('Include specified random pt. ');
if recRanPt=='y' TranPt=length(xiRan);
for ranPt=1:TranPt rPt=rPt+1;xiR(rPt)=xiRan(ranPt);etaR(rPt)=etaRan(ranPt);lambdaR(rPt)=lambdaRan(ranPt);end
else TranPt=0;end
TrPt=TrPt+TSB_Apt+TranPt;
('determine mtl. property of pt. to be computed');
for rPt=1:TrPt
    if Bmodel=='1'
        posAg=angle(xiR(rPt)+i*etaR(rPt));
        if posAg<0 posAg=2*pi+posAg;else;end
        if posAg<agl1 mR(rPt)=1;else mR(rPt)=2;end
    else if etaR(rPt)>0 mR(rPt)=1;else mR(rPt)=2;end
    end
end
('4.Boundary cond.,Exact B. value & exact value at selected pt. ');
for k=1:2*M
    if k<=2*m(1)
        if BC(1)=='1' bc(k)=1;else bc(k)=0;end
    elseif k>2*m(1) & k<=2*(sum(m(1:2)))
        if BC(2)=='1' bc(k)=1;else bc(k)=0;end
    elseif k>2*(sum(m(1:2))) & k<=2*(sum(m(1:3)))
        if BC(3)=='1' bc(k)=1;else bc(k)=0;end
    elseif k>2*(sum(m(1:3))) & k<=2*(sum(m(1:4)))
        if BC(4)=='1' bc(k)=1;else bc(k)=0;end
    elseif k>2*(sum(m(1:4))) & k<=2*(sum(m(1:5)))
        if BC(5)=='1' bc(k)=1;else bc(k)=0;end
    else
        if BC(6)=='1' bc(k)=1;else bc(k)=0;end
    end
    be=round(k/2);x=X(k);y=Y(k);
    exTc(k)=subs(T(mC(k)));ex_gc(k)=-
    subs((c11(mC(k))*dTdx(mC(k))+c12(mC(k))*dTdy(mC(k))*n1(be)+(c12(mC(k))*dTdx(mC(k))+c22(mC(k))*dTdy(mC(k))*n2(be));
end
for rPt=1:TrPt x=xiR(rPt);y=etaR(rPt);exTr(rPt)=subs(T(mR(rPt)));end
if minExValueChk=='y' & min([exTc,exTr])<0
    soundsc(wavread('C:\PROGRA~1\NETWEE~1\RINGIN.WAV'));error(['negative value on boundary or at selected pt., min. T=',num2str(min([exTc,exTr]))]);
else '-';end
('5.Integration parameters I & II ;Assemble sys. of equation [a(p,k)]& [b(p)]')
for il=1:3
    for p=1:2*M; xi=X(p);eta=Y(p);
    for k=1:2*M;be=round(k/2);if rem(k,2)==1 eo=2;else eo=1;end
    if il=1 sw1=-1;sw2=1;else if mC(k)=mC(p) sw1=-1;end;sw2=[sCiJ(mC(k))-CsCiJ(mC(k))]/[sCiJ(1)+sCiJ(2)];end
    if il~=3

```


Appendix B

```

app_qc=bc.*unknown+(ones(1,2*M)-bc).*ex_qc;

clear I II p k
('7.Recover approximated Temp.')
```

```

if TrPt>0
    for il=1:3
        for rPt=1:TrPt;
            xi=xiR(rPt);eta=etaR(rPt);
            for k=1:2*M;bebound(k/2); if rem(k,2)==1 eo=2;else eo=1;end
                if il==1 sw1=-1;sw2=1;else if mc(k)=mR(rPt) sw1=1;else
                    if il~=3
                        A=4*c22(mc(k))*[sCij(mc(k))*midY(be)/c22(mc(k))*sw1*sCij(mR(rPt))*eta/c22(mR(rPt))*sCij(mc(k))*n1(be)-(midX(be)-xi-
                            ij(mc(k))*n1(be))^2];
                        B=[2*c22(mc(k))/L(be)]^2*[sCij(mc(k))*midY(be)/c22(mc(k))+sw1*sCij(mR(rPt))*eta/c22(mR(rPt))*^2+(midX(be)-xi-
                            c12(mc(k))*n1(be))^2];
                        C=sw2*c22(mc(k))/[2*pi*L(be)*[1-1(1)-
                            1(2)]]*[c12(mc(k))*n1(be)+c22(mR(rPt))*n2(be)]/[c12(mc(k))*n1(be)+c22(mR(rPt))*^2+(midX(be)-xi-
                            c12(mc(k))*n1(be))^2];
                        D=sw2*L(be)/[16*pi*sCij(mc(k))*[1-1(1)-
                            1(2)]]*log((L(be)/2/c22(mc(k))^2*[c12(mc(k))*n1(be)+c22(mc(k))*n2(be)]/[c12(mc(k))*n1(be)+c22(mR(rPt))*n2(be))^2+
                            E=sw2*L(be)/[16*pi*sCij(mc(k))*[1-1(1)-1(2)]]);
                    end
                end
            end
        end
    end
end
if 4*B-A^2<1E-12
    I(rPt,k,il)=0;
    if abs(A^2-4)<1e-12
        II(rPt,k,il)=2*(1-2*I(eo))*D+2*E*[2*(1-2*I(eo))*log(2)+4*I(eo)-(2+(-1)^eo*A)/2];
    else
        II(rPt,k,il)=(2*D*(1-2*I(eo))+E/8*[2-8*I(eo)+(-1)^eo*A])*(2+(-1)^eo*A)*log((2+(-1)^eo*A)^2+(6-8*I(eo)+(-1)^eo*A)*(2-(-
            1)^eo*A)*log((2-(-1)^eo*A)^2)-32*(1-2*I(eo))*[log(2)+1]-8*(-1)^eo*A)];
    end
end
I(rPt,k,il)=C*[1/2*log((1-(-1)^eo*A+B))/(1+(-1)^eo*A+B))+2*(1-2*I(eo))+(-1)^eo*A]*atan2(sqrt(4*B-A^2),B-1)/sqrt(4*B-A^2)];
II(rPt,k,il)=2*(1-2*I(eo))*D+E/4*[A^2+2*(-1)^eo*A-2*B+2-4*I(eo)*(2+(-1)^eo*A)]*log(1+(-1)^eo*A+B)-(A^2+2*(-1)^eo*A-2*B-
    6+4*I(eo)*(2-(-1)^eo*A))*log(1-(-1)^eo*A+B)-4*(4-8*I(eo)+(-1)^eo*A)+2*(2-4*I(eo)+(-1)^eo*A)*sqrt(4*B-A^2)*atan2(sqrt(4*B-A^2),B-1)];
end
end
if gama~=inf
    for i2=1:2
        A(i2)=(-1)^(mc(k)+1)*i*L(be)/2*(tau(mc(k))*n1(be)-n2(be))*[1-2*I(eo)+(-1)^(i2+eo)];
        B(i2)=(-1)^(mc(k)+1)*i*[midX(be)-
            B(i2)]/c22(mR(rPt))*eta/c22(mR(rPt))+i*sw1*sCij(mR(rPt))*eta/c22(mR(rPt))-(-1)^i2*L(be)/2*(tau(mc(k))*n1(be)-n2(be))];
        xi+tau(mc(k))*midY(be)+c12(mR(rPt))*eta/c22(mR(rPt))*n1(be)+c22(mc(k))*n1(be)+c22(mR(rPt))*n2(be))/pi/L(be))/(1-1(1)-
            1(2))/[sCij(1)+sCij(2))/[tau(mc(k))*n1(be)-n2(be)]^2;
        D=(-
            C=gama*(1+sCij(mc(k))/CsCij(mc(k)));
            D=(-
            1)^(mc(k)+1)*sw1*i*CsCij(mR(rPt))*[c11(mc(k))*n1(be)+c12(mc(k))*n2(be)+tau(mc(k))*n1(be)+c22(mc(k))*n2(be))/pi/L(be))/(1-1(1)-
            1(2))/[sCij(1)+sCij(2))/[tau(mc(k))*n1(be)-n2(be)]^2;
            E=sw1*CsCij(mR(rPt))/pi/L(be)/(1-1(1)-1(2))/[sCij(1)+sCij(2))/[tau(mc(k))*n1(be)-n2(be)]^2;
            I(rPt,k,3)=real(D*[A(1)/sCij(mc(k))-(-1)^eo/C]*exp(-B(1)*C/sCij(mc(k)))+(-1)^eo*log(B(2)/B(1))/C];
            II(rPt,k,3)=real(E*[A(2)/C-(-1)^eo*sCij(mc(k))/C^2]*exp(-B(2)*C/sCij(mc(k)))-expint(-B(2)*C/sCij(mc(k)))-(A(1)/C-(-
            1)^eo*sCij(mc(k))/C^2)*exp(-B(1)*C/sCij(mc(k)))*expint(-B(1)*C/sCij(mc(k)))+(-1)^eo*(B(1)-B(2))/C+[A(1)/C-(-1)^eo*(sCij(mc(k))/C^2-
            B(1)/C)]*log(B(2)/B(1))]);
        else
            I(rPt,k,3)=0;II(rPt,k,3)=0;
        end
    end
end
end
end
end

```

Appendix B

```

end
end
I=sum(I,3);II=sum(II,3);
appTr=inv(diag(lambdaR))*real(I*appTc.'+II*app_gc. ');
Err=exTr.'-appTr;
percentErr=abs(real(Err./exTr. '))*100;

('8.Show results')
['m=(' ,num2str(m), ),M=' ,num2str(M), ,BC=(' ,num2str(BC(1:tB)), ),element length at each boundary=(',num2str(L_),' )'],
char('Exact sol.:',['gamma=' ,num2str(gama)],['mtl. 1:T(x,y)=' ,char(T(1)), , (c11,c12,c22)=(',num2str([c11(1),c12(1),c22(1)]),['mtl.
2:T(x,y)=' ,char(T(2)), , (c11,c12,c22)=(',num2str([c11(2),c12(2),c22(2)]),[' )'])
if TirPt>0
char('Internal well-space out pt.',['xi',' , 'eta',' , 'exact T',' , 'approx. T',' , 'err.',' ,
', '% err.'])
char(num2str([xiR(1:TirPt), ,etaR(1:TirPt), ,exTr(1:TirPt), ,appTr(1:TirPt),percentErr(1:TirPt)]))
else '-';end
if recSB_APT== 'y'
char('Pt. on smooth boundary/at adjacent ',['at element',' , 'xi',' , 'eta',' , 'exact T',' , 'approx. T',' , 'err.',' ,
T',' , 'err.',' , ', '% err.'])
char('Randomly given pt.',['xi',' , 'eta',' , 'exact T',' , 'approx. T',' , 'err.',' , ', '%
err.'])
char(num2str([xiR(1+TirPt+TSB_APT:TrPt), ,etaR(1+TirPt+TSB_APT:TrPt), ,exTr(1+TirPt+TSB_APT:TrPt), ,appTr(1+TirPt+TSB_APT:
TrPt),percentErr(1+TirPt+TSB_APT:TrPt)]))
else '-';end
if copyResults='y','xi','xiR','eta','etaR','ex.T',exTr(:), 'approx.T',appTr(:), '%err.',Err(:), '%err.',percentErr(:),else '-';end
char(['Tot. % err.=%',num2str(sum(percentErr)),'; norm of err.=',num2str(sqrt(sum(Err.^2)/TrPt)),'; avg. abs.
err.=',num2str(sum(abs(Err))/TrPt)],[' Exact min. boundary T=',num2str(min(min(exTc))),'; min.& max.L=(',num2str([min(L),max(L)]),[' )']),
else xiR=NaN;etaR=NaN; 'No results shown',end

('Checking')
if figBmodel=='y' 'show boundary model';noB={'\Gamma_1','\Gamma_2','\Gamma_3','\Gamma_4','\Gamma_5','\Gamma_6'};
if Bmodel=='1' intEnd=xOff+Brl;else intEnd=leftB;end
figure,plot([Xe,intEnd],[Ye,0], 'k-',X,Y,'b',xiR,etaR,'rx');title('Boundary model');xlabel('x'),ylabel('y');axis equal;legend('boundary
elements','DLE collocation pt.','Pt. under test');
for i1=1:tB i2=sum(m(1:i1-1))+round(m(i1)/2);text(m(i1)/2,miY(i2),noB(i1));end
else '-';end
if plotBvalue=='y' 'plot boundary values'
figure; subplot(2,1,1);plot(1:2*M,exTc(1:2*M),'b', 1:2*M,appTc(1:2*M),'rx');title('T^{(k)} around the
boundary');xlabel('node^{(k)}'),ylabel('T^{(k)}');legend('Exact','Approximated')
subplot(2,1,2);plot(1:2*M,ex_gc(1:2*M),'b', 1:2*M,app_gc(1:2*M),'rx');title('q^{(k)} around the
boundary');xlabel('node^{(k)}'),ylabel('q^{(k)}');
text(1,min(ex_gc),['DLEEM for bi-mtl.','str','m=',num2str(m),'; M=' ,num2str(M), ,BC=' ,BC(1:tB)],'VerticalAlignment','bottom');
else '-';end
ending=clock;elpT=etime(ending,starting);
char(['starting=',num2str(starting)],['ending=',num2str(ending)],['Elapsed Time=',num2str(floor(elpT/3600)),',hr
,num2str(floor(rem(elpT,3600)/60)),',min ,num2str(rem(elpT,60)),',sec']),
if onSound== 'y' soundsc(wavread('C:\PROGRA~1\ONLINE~1\COMPUS~1\Support\WCINACT.WAV')));else '-';end
clear

```