THERMAL CONDUCTION EQUATIONS FOR A MEDIUM WITH AN INCLUSION USING GALERKIN METHOD

by

THIAGARAJAN SIVALINGAM

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2009

This thesis is dedicated to my parents and friends for

their continuous support and encouragement

ACKNOWLEDGMENTS

To begin with I would like to extend my gratitude to my advisor Prof. Seiichi Nomura for giving me an opportunity to work in this interesting topic. His strong source of motivation, guidance and belief kept me moving on with the projects. I would also like to thank my committee members Professor Bo Ping Wang and Professor Haji-Sheikh for their valuable time. Finally I thank my parents, all my friends and roommates for their support and encouragement throughout my Masters program.

October 16, 2009

ABSTRACT

THERMAL CONDUCTION EQUATIONS FOR A MEDIUM WITH AN INCLUSION USING GALERKIN METHOD

Thiagarajan Sivalingam, MS

The University of Texas at Arlington, 2009

Supervising Professor: Seiichi Nomura

The Galerkin method is used to semi-analytically solve the heat conduction equation in non-homogeneous materials. The problem under deliberation is a square plate with a circular inclusion having different thermal conductivities.

A generalized procedure that involves the Galerkin method and formulation of the final solution in terms of the procured base functions is adopted. The Galerkin method basically involves expressing the given boundary value problem in terms of a standard mathematical relation, generating a set of continuous base functions, formulating the matrix equation, and determining the solution. For the non-homogeneous material, a set of base functions for the plate and inclusion are determined separately, through which the solution is formulated for the entire domain. The Galerkin

method involves tedious and time-consuming computations, which is facilitated with the aid of a computer algebra system, *Mathematica*.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

CHAPTER 1

INTRODUCTION

Many problems in engineering and science are described as a boundary value problem in which the physical/mechanical field is expressed as a partial differential equation along with a boundary condition. Solving a boundary value problem has a long history and numerous analytical and numerical methods have been developed. A boundary value problem is a partial differential equation together with a set of additional restraints, called the boundary conditions. A solution to a boundary value problem is a solution to the differential equation which also satisfies the boundary conditions.

Partial differential equations describe phenomena that change continuously. They arise in models throughout mathematics, science, and engineering. By itself, a system of partial differential equations has many solutions [1]. Commonly a solution of interest is determined by specifying the values of all its components at a single point $x = a$. This is an initial value problem (IVP). However, in many applications a solution is determined in a more complicated way. A boundary value problem (BVP) specifies values or equations for solution components at more than one x. Unlike IVPs, a boundary value problem may not have a solution, or may have a finite number, or may have infinitely many [1]. Because of this, programs for solving BVPs

require users to provide a guess for the solution desired. Often there are parameters that have to be determined so that the BVP has a solution. Again there might be more than one possibility, so programs require a guess for the parameters desired. Singularities in coefficients and problems posed on infinite intervals are not unusual.

Boundary value problems arise in many branches of physics as any physical differential equation will have them. Problems involving the wave equation, such as the determination of normal modes, are often stated as boundary value problems [2]. A large class of important boundary value problems is the Sturm-Liouville problems. The analysis of these problems involves the eigenfunctions of a differential operator.

A boundary value problem must be well posed. This means that given the input to the problem, there exists a unique solution which depends continuously on the input. Much theoretical work in the field of partial differential equations is devoted to proving that boundary value problems arising from scientific and engineering applications are in fact well-posed.

For a long time now, differential equations, particularly partial differential equations (PDEs), have played a vital role in describing many of the constitutive laws in physics, engineering and various other scientific fields [2]. One of the most important classes of partial differential equations is the diffusion equation. This equation has its roots dating back to the nineteenth century, when the French physicist and

mathematician, Joseph Fourier, introduced the basic equation (Fourier's law of heat conduction) that relates heat flux with temperature in a homogeneous body. The diffusion equation, as it is generally referred to, is a parabolic PDE that describes the conduction of heat through a material and it can assume several forms depending on the nature of the material and its geometry.

1.1 The Sturm-Liouville system

Many problems in engineering/physics are described by second-order partial differential equations known as the Sturm-Liouville (S-L system) system along with boundary conditions. They are classified into

1) Parabolic (diffusion) equations
2) Elliptic (Laplace) equations
3) Hyperbolic (wave) equations

Commonly used numerical techniques for S-L systems include the method of weighted residuals, finite element method, finite difference method, etc. Typical analytical methods include separation of variables, the eigenfunction expansion method, the Green's function method, etc. [2].

1.2 Analytical methods

1.2.1 Separation of variables

The diffusion equation, as it is generally referred to, is a parabolic partial differential equation that describes the conduction of heat through a material and it can assume several forms depending on the nature of the material and its geometry. For some of the simplified differential equations, analytical (exact) solutions are readily available [3]. In conductive heat transfer, one of the well-known and commonly used analytical techniques to solve the diffusion equation is *the classical separation of variables method* that can be applied directly to linear homogenous boundary value problems (BVPs) with homogenous boundary conditions, but this method has limited applicability with respect to the fact that it cannot solve BVPs with non-homogenous conditions.

1.2.2 Green's function method

Another analytical method that is becoming increasingly popular in solving transient and steady-state heat conduction problems is the Green's function solution method. This method can be applied to homogenous and non-homogenous materials.

1.2.3 Fourier series method

The Fourier series method is another widely used analytical technique that allows the solution to be expressed in terms of an infinite sum of sines and cosines

(trigonometric series) for a problem in the Cartesian coordinate system or in terms of special functions such as the Bessel functions for a circular geometry [3]. The Fourier series can also be extended to solve problems in the spherical coordinate system, by employing Legendre polynomials. However, this method is not easily expanded for non-homogeneous materials or problems of complex geometry.

1.3 Numerical methods

1.3.1 Finite element method

The finite element method (FEM) (sometimes referred to as finite element analysis) is a numerical technique for finding approximate solutions of partial differential equations (PDE) as well as of integral equations [3]. The solution approach is based either on eliminating the differential equation completely (steady state problems), or rendering the PDE into an approximating system of ordinary differential equations, which are then numerically integrated using standard techniques such as Euler's method, Runge-Kutta, etc. The finite element method is a good choice for solving partial differential equations over complex domains, when the domain changes (as during a solid state reaction with a moving boundary), when the desired precision varies over the entire domain, or when the solution lacks smoothness.

FEM allows detailed visualization of bent or twisted structures, and indicates the distribution of stresses and displacements. FEM software provides a wide range of simulation options for controlling the complexity of both modeling and analysis of a

system. The desired level of accuracy required and associated computational time requirements can be managed simultaneously for most engineering applications. FEM allows entire designs to be constructed, refined, and optimized before the design is manufactured [3]. The advantages of FEM include increased accuracy, enhanced design and better insight into critical design parameters, virtual prototyping, fewer hardware prototypes, a faster and less expensive design cycle, increased productivity, and increased revenue.

1.3.2 Finite difference method

The *finite difference method* (FDM) is an alternative way of approximating solutions of PDEs. The differences between FEM and FDM are, the finite difference method is an approximation to the differential equation; the finite element method is an approximation to its solution. The most attractive feature of the FEM is its ability to handle complicated geometries (and boundaries) with relative ease [4]. While FDM in its basic form is restricted to handle rectangular shapes and simple alterations thereof, the handling of geometries in FEM is theoretically straightforward. The most attractive feature of finite differences is that it can be very easy to implement. There are several ways one could consider the FDM a special case of the FEM approach. One might choose basis functions as either piecewise constant functions or Dirac delta functions [5]. In both approaches, the approximations are defined on the entire domain, but need not be continuous. Alternatively, one might define the function on a discrete domain,

with the result that the continuous differential operator no longer makes sense, however this approach is not FEM. There are reasons to consider the mathematical foundation of the finite element approximation more sound, for instance, because the quality of the approximation between grid points is poor in FDM.

The quality of a FEM approximation is often higher than in the corresponding FDM approach, but this is extremely problem dependent and several examples to the contrary can be provided [5]. Generally, FEM is the method of choice in all types of analysis in structural mechanics (i.e. solving for deformation and stresses in solid bodies or dynamics of structures) while computational fluid dynamics (CFD) tends to use FDM or other methods like finite volume method (FVM). CFD problems usually require discretization of the problem into a large number of cells/grid points (millions and more), therefore cost of the solution favors simpler, lower order approximation within each cell. This is especially true for 'external flow' problems, like air flow around the car or airplane, or weather simulation in a large area.

One method, which falls in a category intermediate to numerical and analytical techniques, is the Generalized Fourier Series Expansion Method, also referred to as the *Eigenfunction Expansion Method* (EEM) [6]. This method involves determining a set of continuous base functions that satisfy the given conditions, expressing the given BVP (governing equation) in terms of the Sturm-Liouville (S-L) problem to obtain the eigenvalues and the corresponding set of orthonormal eigenfunctions, and finally, representing the solution as a series expansion of the computed eigenfunctions. There

has been far-reaching use of EEM in various engineering fields. For instance, this method was used to construct the numerical Green's function to solve for the deflection of irregular-shaped classical plates. EEM, in conjunction with the Method of Fundamental Solutions (MFS) and the Method of Particular Solutions (MPS) was used to fabricate a numerical scheme to solve a non-homogenous diffusion equation. Remarkably, EEM has also been used in the realm of quantum physics to solve the quantum-wire problem. In order to validate the effectiveness of EEM, certain problems (diffusion equations in homogeneous materials) are considered for which the Fourier series solution (FSS) has already been established. It will be seen that the results obtained from EEM are in good agreement with the formerly corroborated results.

1.3.3 Weighted residual method

More often than not, it becomes absolutely imperative to adopt numerical (approximation) methods, when an analytical solution does not exist. One such numerical technique that was introduced in 1915 by a Russian mathematician, Boris Galerkin, is the Galerkin method, which is a subclass of the Method of Weighted Residuals (MWR) [6]. This method acts as an effective tool in transforming differential equations into a problem in linear algebra, hence converting the original problem into a finite-dimensional linear system and making the solution process more facile. Compared to other weighted residual methods, the Galerkin method ascertains the convergence of the solution for an adequate number of terms and has been considerably used to solve problems in heat transfer and fluid flow.

In this thesis, a semi-analytical approach is developed to address some of the short-comings of purely numerical methods. Using symbolically capable software, it is possible to find the solution for the second-order partial differential equation.

The following paragraphs briefly outline the forthcoming chapters that will be discussed in this manuscript.

Chapter 2 deals with the process involved in solving partial differential equations, with emphasis on the Galerkin method. The procedure involved in procuring the solution (using the Galerkin method) is discussed in detail. Also, certain expressions are derived that would aid in representing the final solution.

Chapter 3 contains related problems considered in this thesis (a square plate with a circular or elliptical inclusion) that is solved using the Galerkin method, and compared with its corresponding Fourier series solution (exact solution). This would help in validating the efficacy of the Galerkin method.

Chapter 4 involves solutions to partial differential equations in non-homogeneous materials. The 3-D plot and the contour plot for different orders of polynomial are shown, and the final solution is outlined and discussed. Chapter 4, also includes the conclusions, certain discussions and recommendations that came about during the course of this thesis.

Finally, the Appendix, which contains the *Mathematica* codes used in formulating and solving the problems considered in this paper, and the references, are included.

CHAPTER 2

GALERKIN'S METHOD

2.1 The method of weighted residuals

In this Chapter, a brief explanation of the method employed in this thesis is presented. To solve the equation $L\,u = C$, we use the method of weighted residuals. Prior to the development of the finite element method, there existed an approximation technique for solving boundary value problems, called the method of weighted residuals (MWR).

Our objective is to solve a general linear equation of the form

$$L\,u = c \tag{2.1}$$

where L is a linear operator, u is the unknown function and c is a given function. The homogeneous boundary condition of the Dirichlet type is assumed.

An approximate solution to the above equation is sought by a linear combination of N base functions in the linear space as

$$\tilde{u} = \sum_{i=1}^{N} u_i e_i \tag{2.2}$$

where u_i is an unknown coefficient and $e_i$is a trial function that satisfies the homogeneous boundary condition.We wish to approximate u by a function $\tilde{u}$, which is a linear combination of trial functions chosen from a linearly independent set.

Now, when substituted into the differential operator L, the result of the operations is not, in general, c. Hence an error or residual E is defined as

$$E(x) = R(x) .$$

$$= L\big(\tilde{u}(x)\big) - c \qquad (2.3)$$

The notion in the MWR is to force the residual to be zero in some averaging sense over the domain. That is,

$$\int R(x)W_i(x)\ dx = 0 \qquad i = 1,2 \ldots . n \qquad (2.4)$$

where, $W_i(x)$ is the weight function.

The residual (error),R, is the difference between the approximate solution above and the exact solution and is defined as

$$R \equiv L\tilde{u} - c$$

$$= L\sum_{i=1}^{N} u_i e_i - c$$

$$= \sum_{i=1}^{N} u_i Le_i - c \tag{2.5}$$

Note that, R is an element in a function space and is a function of the position, i.e.

$$R(x) \quad = \sum_{i=1}^{N} u_i Le_i\,(x) - c \tag{2.6}$$

where the number of weight functions W_i is exactly equal to the number of unknown constants u_i in $\tilde{u}$. The result is a set of *n* algebraic equations for the unknown constants u_i.

There are (at least) three MWR sub-methods, according to the choices of the weight functions W_is.

2.2 Types of MWR

These three types of MWR are as follows.

1. Collocation method
2. Least square method
3. Galerkin method

2.2.1 Collocation method

In this method, the weight functions are taken from a family of the Dirac delta functions in the domain [4]. That is,

$$W_i(x) \equiv \delta(x - x_i) \tag{2.7}$$

The Dirac delta function has a property that

$$\delta(x - x_i) = \begin{cases} \infty, & x \to x_i \\ 0, & otherwise \end{cases} \tag{2.8}$$

Hence the integration of the weighted residual statement results in the forcing of the residual to be zero at specific points in the domain.

Hence the integration of equation (2.6) with $W_i(x) = \delta(x - x_i)$ results in $R(x_i) = 0$.

Choose u_i so that the residual (error) vanishes at N selected points, i.e.

$$R(x_i) = 0, \quad i = 1,2, \ldots., N$$

Although this method gives the exact values at the selected points, there is no guarantee that the approximation behaves accurately between the selected points.

2.2.2 Least square method

The principle of the least square method is to minimize the norm of the residual. If the norm of the squared residuals is minimized, the rationale behind the name can be seen. In other words, a minimum of the following equation [4].

$$S = \int R(x)R(x)dx = \int R^2(x)dx \tag{2.9}$$

In order to achieve the minimum of this scalar function, the derivatives of S with respect to all the parameters must be zero. That is,

$$\frac{\partial S}{\partial u_i} = 2\int R(x)\frac{\partial R}{\partial u_i}\,dx = 0 \tag{2.10}$$

Comparing with equation (2.6), the weight functions are seen to be

$$W_i = \frac{\partial R}{\partial u_i} \tag{2.11}$$

Therefore the weight functions for the least square method are just the derivatives of the residuals with respect to the unknown constants.

This method is expected to give an overall well-behaved approximation.

2.2.3 Galerkin method

This method was named after B. G. Galerkin, who developed it in the year1915. Earlier, this method was used by I.G Bubnov in 1913. During that time, this method was referred to as the Bubnov-Galerkin method [5].

This method may be viewed as a modification of the Least Squares Method. Rather than using the derivative of the residual with respect to the unknown u_i, the derivative of the approximating function is used. That is, if the function is approximated as in equations (2.6), then the weight functions are chosen as

$$W_i = \frac{\partial \tilde{u}}{\partial u_i} \tag{2.12}$$

Note that these are then identical to the original base functions appearing in equation (2.6)

$$W_i = \frac{\partial \tilde{u}}{\partial u_i} = e_i(x) \tag{2.13}$$

u_i is chosen so that the residual (error) R, is orthogonal to N base functions(e_i), i.e.

$$(R, e_i) = 0 \quad i = 1,2, \dots \dots \dots, N \tag{2.14}$$

The idea of the Galerkin method is that if e_i's span the entire linear space, a vector that is perpendicular to all the base vectors must be a zero vector.

This method acts as an effective tool in transforming differential equations into an algebraic equation, hence converting the original problem into a finite-dimensional linear system and making the solution process more facile.

Equation (2.14) leads to the following set of algebraic equations

$$\sum_{i=1}^{N} (L\, e_j, e_i) u_i = (e_i, c) \qquad (i = 1,2, \dots \dots, N) \tag{2.15}$$

If the set $\{e_1, e_{2,} \dots, e_N\}$ is complete and the residual R is orthogonal to $\{e_1, e_2 \dots, e\}$ then $r \to \theta$ as N$\to\infty$. Of course, we settle for some finite N, but for an arbitrarily small ε we can have the norm $\|R\| < \varepsilon$ by taking N sufficiently large.

2.3 Step by step procedure

In this section, a more detailed procedure is presented using the following Poisson equation as an example

$$L\, u = c \tag{2.16}$$

where, L is the Laplacian operator and c is the constant. Extension to other types of partial differential equations is straightforward.

The differential operator can be identified by comparing the given governing equation with equation above. Also, by comparing the given BVP with the S-L system, the related inner product can be defined.

The next step is to determine a set of continuous base functions that satisfy the given boundary conditions. In the case of non-homogeneous materials, the base functions also need to satisfy continuity conditions across the phase. In order to find these base functions, a trial function needs to be defined. As an illustration, a simple trial function (for a 2-D problem in the Cartesian coordinate system) can be expressed as

$$f(x) = \sum a_{ij} x^i y^j \tag{2.17}$$

Equation (2.17) is then subjected to the conditions prescribed in the given BVP, so as to obtain a set of simultaneous equations for the unknown coefficients (a_{ij}). The solution to these simultaneous equations yields a set of expressions for the unknown coefficients that are substituted back into the above equation [7]. Finally, by extracting the terms (polynomials) associated with the unknown coefficients, an array of continuous base functions (b_{i}) is generated.

This will yield an equation of the form

$$\sum_{i=1}^{N}\left(u_i\ L\ e_j, e_i\right) = (e_i, c) \qquad (2.18)$$

The above equation (2.18) can also be written as,

$$\sum_{i=1}^{N}\left(L\ e_j, e_i\right)u_i = (e_i, c) \qquad (i = 1,2, \ldots \ldots, N) \qquad (2.19)$$

In order to solve the above set of equations, equation (2.19) can be written in matrix form as

$$A.p = c$$

where,

$$A = \begin{pmatrix} (L\ e_1 e_1) & (L\ e_1 e_2) & \ldots & (L\ e_1 e_N) \\ (L\ e_2 e_1) & (L\ e_2 e_2) & \ldots & (L\ e_2 e_N) \\ \ldots & \ldots & \ldots & \ldots \\ (L\ e_N e_1) & (L\ e_N e_2) & \ldots & (L\ e_N e_N) \end{pmatrix}$$

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_N \end{pmatrix}$$

and

$$p = \begin{pmatrix} (e_1, c) \\ (e_2, c) \\ \dots \\ (e_N, c) \end{pmatrix}$$

The above set of equations can be solved for u_i as follows.

$$u = A^{-1}.p$$

CHAPTER 3

FORMULATION OF SOLUTION

In CHAPTER 2, the method of weighted residue and then the different types of solving boundary value problems in that method were described in detail. In this Chapter, the Galerkin method will be discussed elaborately, followed by the step by step procedure involved in solving the boundary value problem for the considered non-homogeneous system (a square plate with an elliptical inclusion). The main aim is to find out semi-analytical solution to the problem, as there is no accurate method to solve such problems.

As is well known, every ellipse has a major axis and a minor axis, and the ratio between the major axis and the minor axis is called as *aspect ratio* of the ellipse. In this research, a square plate with an elliptical inclusion (non-homogeneous condition) is considered. In the first case, the major axis and the minor axis are set to be the same (a circle).

This means that the aspect ratio of the ellipse is one, or in other words, a square plate with a circular inclusion is considered. For the second case, the major and minor axes are set to be different, with $a > b$.

3.1 Trial functions

For the purpose of finding the analytical solution, we must find the trial functions for the given boundary value problem. Trial functions are functions which are formed in such a way that the boundary conditions and the continuity condition across the phases are satisfied.

To make the understanding of the finding the trial functions and then the analytical solution for the problem considered, a simple geometry is taken into consideration. A square plate is considered without any inclusion. This problem is explained and solved before beginning with the actual solution for the problem considered. After finding the solution for the square plate problem, the solution for a square plate with a circular inclusion is solved, followed by the solution for a square plate with elliptical inclusion.

3.2 Construction of trial functions

Basically, there are two types by which trial functions can be constructed. These two methods are as follows.

1. Dirichlet type
2. Neumann type

3.2.1 Dirichlet type

Dirichlet type boundary conditions are boundary conditions in which the values of the unknown function are given on the boundary. In the Dirichlet type, the trial function is constructed in the following manner

$$f_i = h(x,y)\sum c_{ij}x^i y^j$$

3.2.2 Neumann type

Neumann type boundary conditions are boundary conditions which give the normal derivative on a surface.

In the Neumann type, the trial function is constructed in the following manner

$$f_i = \sum c_{ij}x^i y^j$$

3.3 Examples of some geometry

First, a simple square plate with the co-ordinate points (-1,-1), (-1, 1), (1, 1) and (1,-1) is considered.

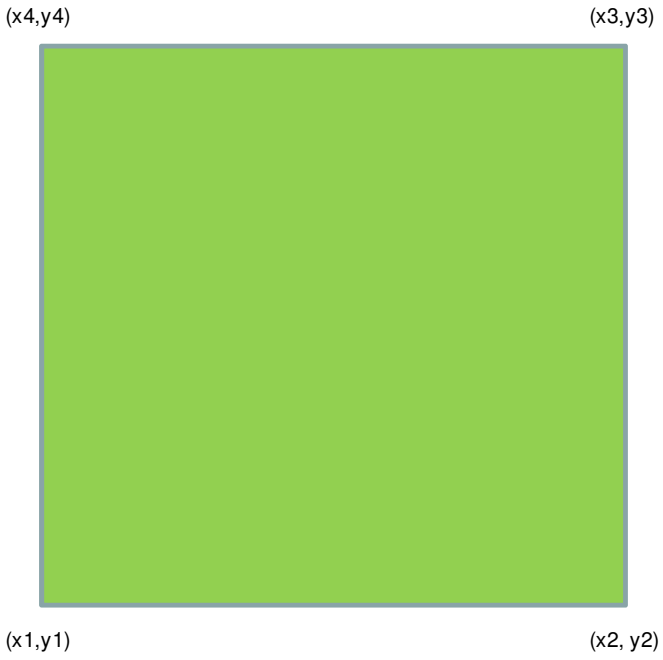

Figure 3.1 A simple square plate

Hence, a trial function which satisfies these boundary conditions must be taken into account in order to find the approximate solution for the problem.

Let us consider a function that vanished on the boundary as

$$(x^2 - 1)(y^2 - 1) \tag{3.1}$$

Now the base functions are obtained by multiplying the above function (3.1) with a complete set of polynomials (in x and y) as

$$f = (x^2 - 1)(y^2 - 1)x^i y^{j-i} \tag{3.2}$$

where $i = 0,1,2,3, \ldots \ldots, j$

In equation (3.2), j represents the order of the polynomial. As an illustration, the first few base functions will be,

$$b(x, y) = (x^2 - 1)(y^2 - 1) \tag{3.3}$$

$$b_1(x, y) = (x^2 - 1)(y^2 - 1)y^2$$

$$b_2(x, y) = (x^2 - 1)(y^2 - 1)xy$$

$$b_3(x, y) = (x^2 - 1)(y^2 - 1)x^2$$

$$b_4(x, y) = (x^2 - 1)(y^2 - 1)xy^2$$

$$b_5(x, y) = (x^2 - 1)(y^2 - 1)x^2y$$

$$b_6(x, y) = (x^2 - 1)(y^2 - 1)y^3$$

Secondly, a square plate with a circular inclusion is considered. The co-ordinate points of the square plate are again (-1,-1), (-1, 1), (1, 1) and (1,-1).

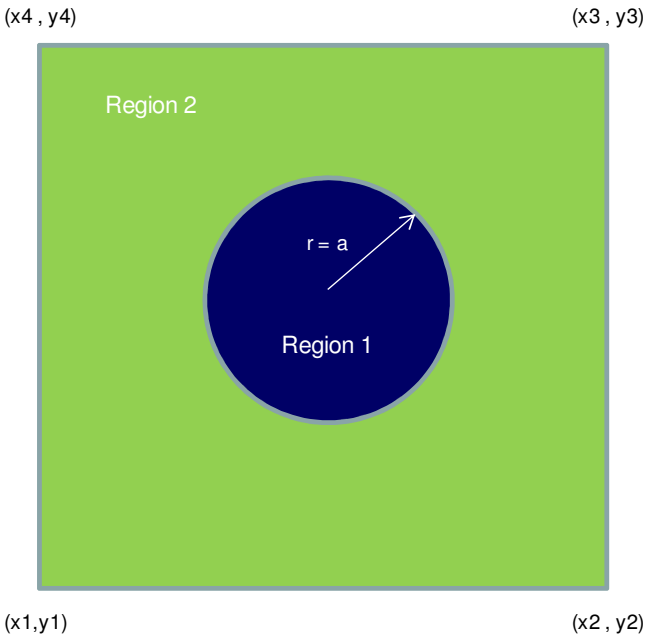

Figure 3.2 A Square plate with circular inclusion

Finally, a square plate with an elliptical inclusion is considered with the same coordinate points.

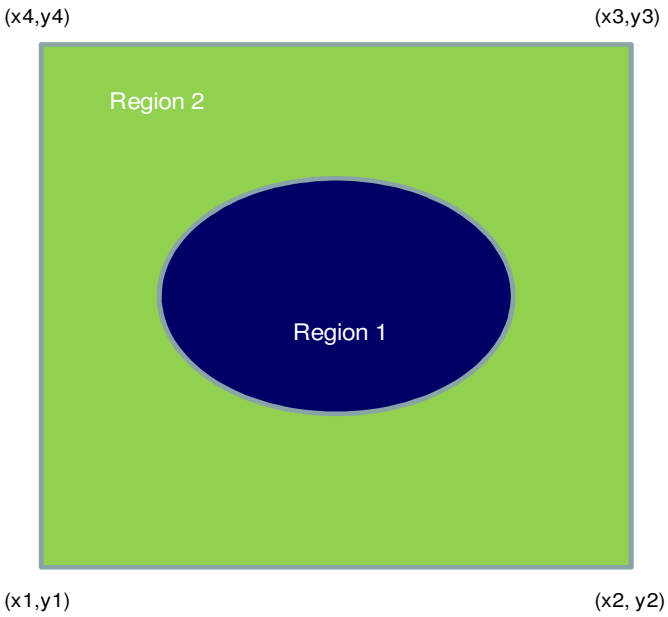

Figure 3.3 A Square plate with elliptical inclusion

For the square plate with a circular inclusion, a trial function which satisfies these boundary conditions must be taken into account in order to find the analytical solution for the problem.

Shown below, is the trial function corresponding to region 1

$$f_1(x,y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + \ldots\ldots$$

In order to define the trial function for region 2, firstly, a function is chosen such that it satisfies all the homogeneous Dirichlet type boundary conditions on the square plate as

$$(x^2 - 1)(y^2 - 1)$$

Employing the above function, the trial function for region 2 is defined as

$$f_2(x,y) = (x^2 - 1)(y^2 - 1)(a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + \cdots)$$

As discussed earlier, manipulation of the trial functions ($f_1(x,y)\ and\ f_2(x,y)$) results in 2 sets of continuous base functions for regions 1 and 2 –*i.e.*, the coefficients of $b_i(x,y)_1$and $b_i(x,y)_2$, are determined. For integrations that involve over the circular region, the base functions in x and y can be converted into base functions in r and θ – *i.e.* $b_i(r,\theta)_1$and $b_i(r,\theta)_2$, so as to suit the requirements of the given boundary value problem.

$$x = r\cos(\theta)\,, \qquad y = r\sin(\theta)$$

The matrices A_{ij} and B_{ij} are defined in the following manner:

For the circular inclusion:

$$(A_{ij})_{CI} = k_1 \int_0^a \int_0^{2\pi} L_{r\theta}\,[b_i(r,\theta)_1]\ (b_j(r,\theta)_1) r\,dr\,d\theta$$

$$(B_{ij})_{CI} = \int_0^a \int_0^{2\pi} (b_i(r,\theta)_1)\ (b_j(r,\theta)_1) r\,dr\,d\theta$$

For the entire square plate:

$$(A_{ij})_{SP} = k_2 \int_{-1}^{1} \int_{-1}^{1} L_{xy}\,[b_i(x,y)_2]\ (b_j(x,y)_2)\,dx\,dy$$

$$(B_{ij})_{SP} = \int_{-1}^{1} \int_{-1}^{1} (b_i(x,y)_2)\ (b_j(x,y)_2)\,dx\,dy$$

For the circular portion extracted from the square plate:

$$(A_{ij})_{EX} = k_2 \int_0^a \int_0^{2\pi} L_{r\theta}\,[b_i(r,\theta)_2]\ (b_j(r,\theta)_2) r\,dr\,d\theta$$

$$(B_{ij})_{EX} = \int_0^a \int_0^{2\pi} (b_i(r,\theta)_2)\ (b_j(r,\theta)_2) r\ dr\ d\theta$$

where

$$L_{r\theta} = -(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2})$$

$$L_{xy} = -(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$$

$L_{r\theta}$ is an equivalent form of L_{xy}, in the polar coordinate system.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results obtained

The detailed procedure explained in the previous chapters is employed to obtain the final solution. A square plate with circular inclusion is considered for this process. The solution is obtained with the software, *Mathematica.* From the plots obtained, it can be seen that the solution gradually converges as the order of the polynomial increases. In other words, the accuracy of the solution increases for higher order of polynomials. In the following section, the plots obtained for different order of polynomials are illustrated.

4.2 Approximate solutions

4.2.1 Order: 4

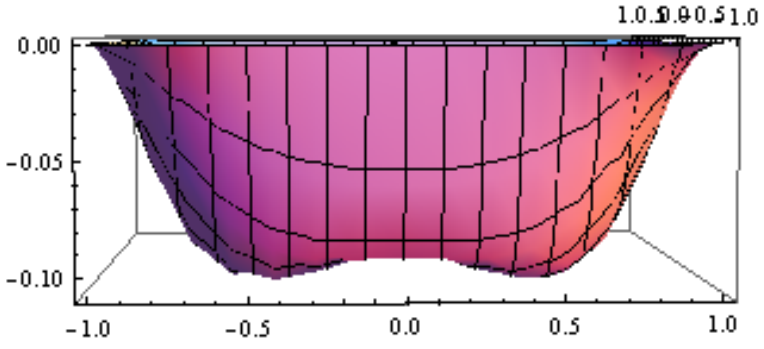

Figure 4.1 3D plot for fourth order polynomial

4.2.2 Order: 5

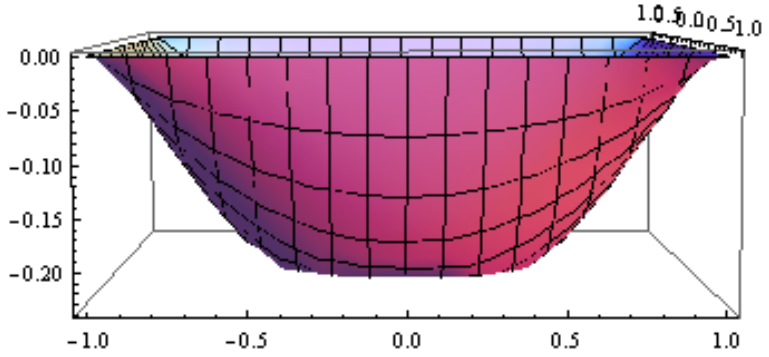

Figure 4.2 3D plot for fifth order polynomial

4.2.3 Order: 6

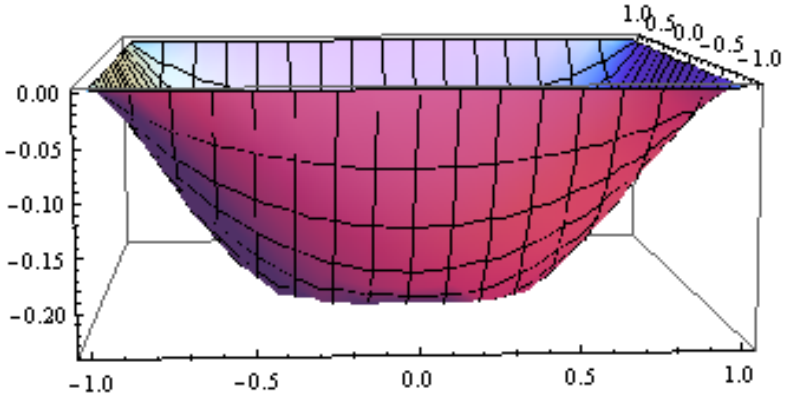

Figure 4.3 3D plot for sixth order polynomial

4.2.4 Order: 7

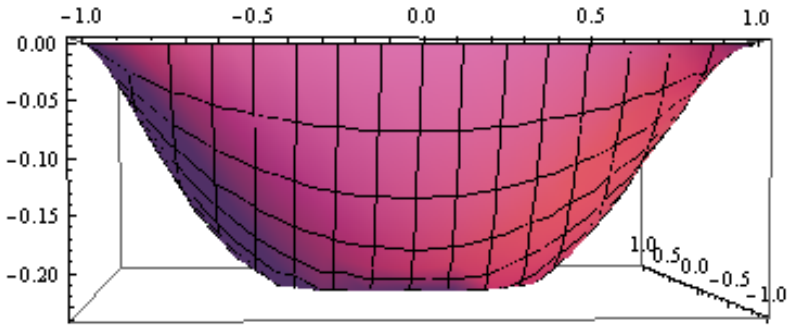

Figure 4.4 3D plot for seventh order polynomial

4.3 Conclusion

The Galerkin method was used to solve differential equations in homogenous and non-homogeneous materials. However, the primary focus was to solve differential equations in non-homogeneous materials (for which closed form solutions do not exist). *Mathematica*, a computer algebra system, was extensively employed to achieve the desired results.

In this proposition, the approach adopted to solve diffusion equations can be summarized as follows:

- The given BVP was expressed in the form of a standard mathematical relation:

$$Lu = c$$

- The corresponding base functions for the given order of polynomial were determined using the Galerkin method

- After finding the A_{ij} matrix and the c vector, the final solution, p was found out using

$$p = A^{-1}c$$

For differential equations in homogeneous materials, for which the Fourier series solution (FSS) was available, the solutions obtained from the Galerkin method are in good agreement with the FSS solutions, validating the efficiency of the proposed method. The solution obtained from the Galerkin method can be further refined by considering more number of terms/base functions. This would result in a more enhanced approximation of the trigonometric and Bessel functions as well as the final solution.

Since the Galerkin method was effective in solving differential equations in non-homogeneous materials, the possibility of employing this method in fields such as electronic packing and composites cannot be ruled out. Also, considering the versatility of this method in solving PDEs, the procedure involved in the Galerkin method could be extended to solve 3-D problems.

The conclusion and discussion can be summarized as follows.

1. The trial functions that satisfy any given conditions were derived.

2. The Poisson equations for both homogeneous and inhomogeneous materials were solved.

3. The method can be used to different types of differential equations and different types of geometries.

4. Complete solutions for elliptic inclusions should be tried in the future.

APPENDIX A

SQUARE PLATE WITH CIRCULAR INCLUSION

```
o=4;
l={3,6,10,15,21,28,36,45};

Table[Table[x^iy^(n-i),{i,0,n}],{n,0,o}];
base1=Flatten[%];

poly1=Sum[a[i] base1[[i]],{i,1,l[[o]]}];
poly2=(poly1/.a→b) (x^2-1) (y^2-1);

polyheat1=k1 (D[poly1,x]*Cos [θ]+D[poly1,y]*Sin [θ])//Simplify;
polyheat2=k2 (D[poly2,x]*Cos [θ]+D[poly2,y]*Sin [θ])//Simplify;
```

(*FUNCTION FOR FINDING THE EQUATIONS TO BE SOLVED*)

```
decomp[f_]:=CoefficientList[TrigReduce[f/.{x→r Cos[θ],y→r
Sin[θ]}]/.{Cos[i_.*θ]→cosine^i,Sin[i_.*θ]→sine^i},{cosine,sine}]//Flat
ten
```

(*FINDING THE EQUATIONS AND COMBINING THEM INTO A SINGLE TABLE*)

```
eq1=decomp[poly1-poly2]/.r→a;
eq2=decomp[polyheat1-polyheat2]/.r→a;
eq3=Flatten[{eq1,eq2}];
eq4=Map[#==0&,eq3];
```

(*TABLE FOR THE UNKNOWNS*)

```
variable=Flatten[{Table[a[i],{i,l[[o]]}],Table[b[i],{i,l[[o]]}]}];
```

(*SOLVING FOR THE UNKNOWNS*)

```
solution=Solve[eq4,variable];

polysub1=poly1/.solution;
polysub2=poly2/.solution;

table1=Flatten[{Table[Coefficient[polysub1,
a[i]],{i,1,l[[o]]}],Table[Coefficient[polysub1, b[i]],{i,1,l[[o]]}]}];

table2=Flatten[{Table[Coefficient[polysub2,
a[i]],{i,1,l[[o]]}],Table[Coefficient[polysub2, b[i]],{i,1,l[[o]]}]}];
```

(*FINDING THE BASE FUNCTIONS*)

```
base=DeleteCases[Transpose[{table1,table2}],{0,0}];
```

(*USING THE 'Flatten' COMMAND TO MAKE IT EASIER TO USE THE LIST ELEMENTS*)

```
e=Flatten[base];
```

(*FINDING THE LENGTH WHICH IS USED TO AUTOMATE THE CODE*)

```
g=Length[e]
```

(*FORMING A LIST WITH THE INNER REGION TERMS ALONE, SO THAT IT CAN BE USED EASILY IN INTEGRATION*)

```
in=Table[e[[b]],{b,1,g,2}];
```

(*FORMING A LIST WITH THE OUTER REGION TERMS ALONE, SO THAT IT CAN BE USED EASILY IN INTEGRATION*)

```
out=Table[e[[c]],{c,2,g,2}];
```

(*FORMING A SEPERATE LIST FOR THE SECOND PART OF THE FORMULA WHICH WILL BE USED FOR INTEGRATING ALONG THE SQUARE REGION)*)

```
kmat=Table[-k2((D[out[[i]],x]*D[out[[j]],x])+(D[out[[i]],y]
*D[out[[j]],y])),{i,1,(g/2)},{j,1,(g/2)}]//Simplify;
```

(*FORMING A SEPERATE LIST FOR THE SECOND PART OF THE FORMULA WHICH WILL BE USED FOR INTEGRATING ALONG THE CIRCULAR REGION)*)

```
lmat1=Table[-k1 ((D[in[[i]],x]*D[in[[j]],x])+(D[in[[i]],y]
*D[in[[j]],y]))+k2 ((D[out[[i]],x]*D[out[[j]],x])+(D[out[[i]],y]
*D[out[[j]],y])),{i,1,(g/2)},{j,1,(g/2)}]//Simplify;
```

(*MULTIPLYING WITH 'r' BECAUSE WE ARE INTEGRATING THE INNER REGION IN POLAR COORDINATE SYSTEM*)

```
lmat=lmat1*r/.{x→r Cos[θ],y→r Sin[θ]};
```

(*WRITING A FUNCTION WHICH PERFORMS INNER PRODUCT*)

```
int[k_,l_]:=Integrate[k,{x,-1,1},{y,-1,1}]+Integrate[l,{r,0,a},{θ,0,2
π}]
```

(*FINDING THE ELEMENTS Aij OF THE SOLUTION MATRIX*)

```
val=Table[int[kmat[[i,j]],lmat[[i,j]]],{i,1,(g/2)},{j,1,(g/2)}];//Simp
lify
```

(*SUBSTITUTING THE VALUES FOR 'k1','k2' and 'a', WE GET THE Aij MATRIX, WHICH IS SYMMETRICAL*)

```
Aij=val/.{k1→1,k2→10,a→0.1};
```

(*INNER BASE FUNCTION MINUS OUTER BASE FUNTION*)

```
bmat2=Table[r*(-k1 e[[i]]+k2 e[[i+1]]),{i,1,g,2}]/.{x→r Cos[θ],y→r
Sin[θ]};
```

(*OUTER BASE FUNCTION*)

```
bmat1=-k2 out;
```

(*FINDING Bij MATRIX*)

```
Bmat=Table[int[bmat1[[i]],bmat2[[i]]] ,{i,1,(g/2)}];
Bmat/.{k1→1,k2→10,a→0.1};
```

(*SOLVING FOR THE UNKNOWNS - TEMPERATURE VALUES*)

```
Umat=Inverse[Aij].Bmat/.{k1→1,k2→10,a→0.1}
SolIn=Sum[Umat [[i]]*in[[i]],{i,1,(g/2)}]/.{k1→1 , k2→10,
a→0.1}//Simplify;
SolOut=Sum[Umat [[i]]*out[[i]],{i,1,(g/2)}]/.{k1→1 , k2→10,
a→0.1}//Simplify;
```

(*PLOTTING THE GRAPH*)

```
p[x_,y_]:=If[( x^2+y^2)< (0.1)^2, SolIn, SolOut]
Plot3D[p[x,y],{x,-1,1},{y,-1,1}]
ContourPlot[p[x,y],{x,-1,1},{y,-1,1}]
```

APPENDIX B

SQUARE PLATE WITH AN ELLIPTIC INCLUSION

(*DEFINING THE ORDER OF THE POLYNOMIAL*)

```
o=4;
Table[Table[x^i y^(n-i),{i,0,n}],{n,0,o}];
base1=Flatten[%];
```

(*FINDING POLY1 [OUTER REGION] AND POLY2 [INNER REGION]*)

```
poly1=Sum[a[i] base1[[i]],{i,1,Length[base1]}];
poly2=(poly1/.a□b) (x^2-1) (y^2-1);
```

(*MODIFYING THE ABOVE EQUTIONS WITH THE THERMAL CNODUCTIVITY VALUES, K1 AND K2, TO GET POLYHEAT1 AND POLYHEAT2*)

```
polyheat1=k1 (D[poly1,x]*(x/a^2)+D[poly1,y]*(y/b^2))//Simplify;
polyheat2=k2 (D[poly2,x]*(x/a^2)+D[poly2,y]*(y/b^2))//Simplify;
```

(*FUNCTION FOR FINDING THE EQUATION TO BE SOLVED*)

```
decomp[f_]:=CoefficientList[TrigReduce[f/.{x□a Cos[□],y□b
Sin[□]}]/.{Cos[i_.*□]□cosine^i,Sin[i_.*□]□sine^i},{cosine,sine}]//Flat
ten
```

(*EMPLOYING (POLY1-POLY2) & (POLYHEAT1-POLYHEAT2) IN THE ABOVE FUNCTION*)

```
eq1=decomp[poly1-poly2];
eq2=decomp[polyheat1-polyheat2];
```

(*FORMING THE EQUATIONS TO BE SOLVED*)

```
eq3=Flatten[{eq1,eq2}];
eq4=Map[#□0&,eq3];
```

(*FORMING A LIST OF ALL THE VARIABLES TO BE SOLVED FOR*)

```
variable=Flatten[{Table[a[i],{i,Length[base1]}],Table[b[i],{i,Length[b
ase1]}]}];
```

(*SOLVING THE ABOVE EQUATION (EQ4)FOR ALL THE UNKNOWN VARIABLES*)

```
solution=Solve[eq4,variable]//Simplify;
```

(*SUBSTITUTE THE FOUND VALUES BACK INTO POLY1 AND POLY2*)

```
polysub1=poly1/.solution;
polysub2=poly2/.solution;
```

(*USING THE 'COEFFICIENT' FUNCTION TO FIND THE BASE FUNCTIONS*)

```
table1=Flatten[{Table[Coefficient[polysub1,
a[i]],{i,1,Length[base1]}],Table[Coefficient[polysub1,
b[i]],{i,1,Length[base1]}]}];
table2=Flatten[{Table[Coefficient[polysub2,
a[i]],{i,1,Length[base1]}],Table[Coefficient[polysub2,
b[i]],{i,1,Length[base1]}]}];
base=DeleteCases[Transpose[{table1,table2}],{0,0}];
e=Flatten[base]//Expand;
g=Length[e];
```

(*FORMING A LIST WITH THE INNER REGION TERMS ALONE, SO THAT IT CAN BE USED EASILY IN INTEGRATION*)

```
InnerBase=Table[e[[b]],{b,1,g,2}];
```

(*FORMING A LIST WITH THE OUTER REGION TERMS ALONE, SO THAT IT CAN BE USED EASILY IN INTEGRATION*)

```
OuterBase=Table[e[[c]],{c,2,g,2}];
```

(*FORMING A SEPERATE LIST FOR THE SECOND PART OF THE FORMULA WHICH WILL BE USED FOR INTEGRATING ALONG THE SQUARE REGION)*)

```
kmat=Table[-k2
((D[OuterBase[[i]],x]*D[OuterBase[[j]],x])+(D[OuterBase[[i]],y]*D[Oute
rBase[[j]],y])),{i,1,(g/2)},{j,1,(g/2)}]//Simplify;
```

(*USING THE 'GAMMA' FUNCTION TO INTEGRATE ALONG THE ELLIPTICAL REGION*)

```
InnerPoly=Flatten[Table[(InnerBase[[i]] InnerBase[[j]]-OuterBase[[i]]
OuterBase[[j]]),{i,1,2},{j,1,2}]];
CoeffList1=Table[CoefficientList[InnerPoly[[i]],{x,y}],{i,1,g}];
CoeffList=Table[If[CoeffList1[[k]]□{},Table[i*0+j*0,{i,9},{j,9}],Coeff
List1[[k]]],{k,1,Length[CoeffList1]}];
Dim1=Table[Dimensions[CoeffList[[j]]],{j,1,(g)}];
Dim=Table[If[Dim1[[i]]□{0},{2,3},Dim1[[i]]],{i,1,(g)}]
```

```
G[l1_,l2_]:=If[EvenQ[l1]&& EvenQ[l2], ((a^l1*b^l2)/4)*(Gamma[l1/2]
Gamma[l2/2])/(Gamma[(l1/2)+(l2/2)+1]),0]
```

REFERENCES

[1] C.A.J. Fletcher (1984). Computational Galerkin Method. Springer-Verlag Inc.

[2] Ruel V. Churchill, & James Ward Brown (1987). Fourier Series And Boundary Value Problems,Fourth Edition. McGraw-Hill Book Company.

[3] George Cain, & Gunter H. Meyer (2006). Separation of Variables for Partial Differential Equations, An Eigenfunction Approach.Chapman & Hall/CRC.

[4] R.E. Diaz-Contreras, & S. Nomura (2005). Numerical Green's Function Approach to Finite-sized Plate Analysis. Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, Arlington, TX.

[5] Stephen Wolfram (1996). The Mathematica Book, Third Edition. Wolfram Media, & Cambridge University Press.

[6] Erwin Kreyszig (2001). Advanced Engineering Mathematics. John Wiley & Sons.

[7] G. F. Roach (1982). Green's Functions. Cambridge University Press.

[8] Y.C. Fung (1977). A First Course In Continuum Mechanics. Prentice Hall.

BIOGRAPHICAL INFORMATION

Thiagarajan Sivalingam did his Masters in Mechanical Engineering at the University of Texas, Arlington. He earned his Bachelor degree in Anna University, India. His research interest is focused on the thermal conduction equations in a medium with different types of inclusions. His research works uses the powerful computation software *Mathematica.* His future plan is to gain some industrial experience related to his research interest and excel in this field.