EAST WEST UNIVERSITY

Thesis Title

Implementation of Finite Element Method

Supervisor

Dr. Mohammed Moseeur Rahman

Assistant Professor

Department of Electronics and Communications Engineering

Submitted by :

Name	ID
Jobayer Sarkar	2014-2-55-004
Kh. Efta Khairul Alam Dhanna	2014-2-55-015
Mehedi Hasan Mithu	2014-2-55-020

Summer 2019

DECLARATION

This declaration is to clarify that all of the submitted contents of this thesis are original in its figure, excluding those, which have been admitted specifically in the references. All the simulation work involved is of our own. This thesis has been submitted as a partial fulfillment of the degree of Bachelor of Science in Electronics & Communications Engineering. We hereby declare that this work is of our own excluding for the references, documents and summaries those have been acknowledged.

Counter signed

………………………..

Dr. Mohammed MoseeurRahman
Assistant Professor and Chairperson
Department of Electronics & Communications
Engineering, East West University.
Dhaka, Bangladesh.

Signature of Students

………………………….

Mehedi Hasan Mithu
ID: 2014-2-55-020

……………………………

JobayerSarkar
ID: 2014-2-55-004

……………………………

Kh. Efta Khairul Alam Dhanna
ID: 2014-2-55-015

ACKNOWLEDGEMENT

Alhamdulillah. Thanks to Allah (SWT), to whom we belong. With His blessing we had the opportunity to complete this Final Year Thesis titled **Implementation of finite element method.**

At First, we would like to express our profound gratitude and deep regards to Dr. Mohammed Moseeur Rahman for his guidance, motivating thoughts and invariable support throughout the compilation of this thesis.

We also pleases to all our faculty members of ECE department for their guidance towards the completion of our graduation degree.

A special thanks to our parents whose encouragement and prayer were always with us.

Thanks you all for your grate support to us.

………………………….	……………………	…………………………
Mehedi Hasan Mithu	**Jobayer Sarkar**	**Kh. Efta Khairul Alam Dhanna**
Id: 2014-2-55-020	**Id: 2014-2-55-004**	**Id: 2014-2-55-015**

APPROVAL

This simulation based thesis titled **Implementation of finite element method** was prepared and submitted by**Mehedi Hasan Mithu (Id: 2014-2-55-020)** and **Jobayer Sarkar (Id: 2014-2-55-004)** and **Kh. Efta khairul Alam Dhanna(Id: 2014-2-55-015)** and has been found satisfactory in term of scope, quality and presentation as a partial fulfillment of the requirement for the degree of Bachelor of Science in Electronics & Communications Engineering at East West University.

Checked and Approved by

____

Supervisor

Dr. Mohammed Moseeur Rahman

Assistant Professor and Chairperson

Department of Electronics & Communications Engineering

East West University

ABSTRACT

The finite element method is a numerical technique used to perform finite element analysis of any given physical phenomenon. Finite element method is necessary to use mathematics to comprehensively understand and quantify any physical phenomena, such as structural or fluid behavior, thermal transport, wave propagation and the growth of biological cells etc. Here, in this thesis we showed the implementation of finite element method with the help of boundary value problem and the galerkin methods. Galerkin methods are a class of methods for converting a continuous operator problem to a discrete problem. In this thesis we used galerkin finite element method solution to solve 2D boundary valued problem using triangular elements. We developed a Matlab code to solve 2D boundary value problem. Then we got analytical and numerical solutions with proper figures. You will see those solutions and figures in our thesis paper.

INDEX

Contents **Page No**

- **Chapter 1 : Introduction --- 1- 14**
  - **1.1 Introduction ---2**
  - **1.2 History of finite element method ---------------------------------------2-3**
  - **1.3 Numerical methods ---3**
    - **1.3.1 Finite Difference Method --4-5**
    - **1.3.2 Finite Volume Method---5-7**
    - **1.3.3 Finite Element Method --7**
    - **1.3.4 Boundary Element Method --8**
      - **1.3.4.1 Advantages ---9**
      - **1.3.4.2 Disadvantages --10**
    - **1.3.5 Meshless Methods ---10-12**
  - **1.4 Concepts of elements and nodes --13-14**
    - **1.4.1 Nodes -- --13**
    - **1.4.2 Elements --14**
- **Chapter 2 : Finite Element Method (FEM) -------------------------------------15-28**
  - **2.1 Finite Element Method ---16**
  - **2.2 Analytical Solution --16**
  - **2.3 FEM--17**
  - **2.4 Common FEM Application --17**
  - **2.5 Types of Finite Elements ---18**
  - **2.6 Advantages of Finite element Method---------------------------------19**
  - **2.7 Principles of FEM --19-20**
  - **2.8 A General Procedure for Finite Element Analysis------------------20-21**
    - **2.8.1 Preprocessing --20**
    - **2.8.2 Solution ---21**
    - **2.8.3 Post Processing --21**
  - **2.9 Stiffness Matrix --21**
  - **2.10 Discretization of domain and generalization of an element ------22-28**
    - **2.10.1 Shape Functions ---22-26**
    - **2.10.2 Relationship of linear shape function with coordinates ----27-28**

- Chapter 3 : Boundary Value Problem And Galerkin Method ---------------------29-40
  - 3.1 Boundary Value Problem ---30-35
    - 3.1.1 Explanation ---31
    - 3.1.2 Boundary Conditions ---32
    - 3.1.3 Initial Value Problem --32
    - 3.1.4 Numerical Analysis of Boundary Value Problem -------------------33-35
    - 3.1.4.1 Shooting Method --33
    - 3.1.4.2 Example ---34-35

  - 3.2 Galerkin Method --36-40
    - 3.2.1 A one dimensional parabolic problem ------------------------------36
    - 3.2.2 WeakGalerkin Formulation ---37
    - 3.2.3 Semi- discretization ---37-39
    - 3.2.4 Multidimensional Problems --39-40

- Chapter 4 : Galerkin FEM Solution For 2D BVP Using Triangular Elements 41-53
  - 4.1 Governing Equation ---42-53

- Chapter 5 : Result and Simulation --54-57
  - 5.1: Figure --55-57
- Chapter 6 : Conclusion --58-60
- Appendix ---61-65
- References ---66-67

Chapter 1

Introduction

1.1 Introduction

The Finite Element Method (FEM) is a numerical technique to find approximate solutions of partial differential equations. It had been originated from the necessity of resolution complicated snap and structural analysis issues in Civil, Mechanical and part engineering etc. In a very structural simulation, FEM helps in manufacturing stiffness and strength visualizations. It also conjointly helps to minimize material weight and its price of the structures. FEM allows for elaborate visual image and indicates the distribution of stresses and strains within the body of a structure. Several of FE software are powerful yet complex tool - meant for skilled engineers with the coaching and education necessary to properly interpret the results.

Several fashionable FEM packages embody specific element like fluid, thermal, magnetism and structural operating environments. FEM permits entire styles to be made, refined and optimized before the design is manufactured. This powerful design tool has considerably improved both the quality of engineering designs and therefore the methodology of the design method in several industrial applications. The utilization of FEM has significantly diminished the time to take products from idea to the assembly line.

1.2 History of finite element method

The finite element analysis are often derived back to the work by Alexander Hrennikoff (1941)and Richard Courant(1942). Hrenikoff introduced the framework methodology, during which a plane elastic medium was drawn as collections of bars and beams.These pioneers share one essential characteristic: mesh discretization of a nonstop domain into a collection of discrete sub-domains, usually called elements.
Below some advancement in time- link about FEM is illustrated:

- In 1950s, solution of large number of simultaneous equations became possible because of the digitalcomputer.

- In 1960, Ray W. Clough first published a paper using term “Finite Element Method”.

- In 1965, First conference on “finite elements” was held
.

- In 1967, the first book on the “Finite Element Method” was published by Zienkiewicz and Chung.

- In the late 1960s and early 1970s, the FEM was applied to a wide variety of engineering problems.

- In the 1970s, most industrial FEM computer code packages (ABAQUS, NASTRAN, ANSYS, etc.) originated.Interactive FE programs on supercomputer result in riding of CAD systems.

- In the 1980s, algorithm on magnetic force applications, fluid flow and thermal analysis were developed with the utilization of FE program.

- Engineers will appraise ways that toregulate the vibrations and extend the use of flexible, deployablestructures in space using FE and alternative ways within the 1990s. Trends to resolve absolutely coupled resolution of fluid flows with structural interactions, bio-mechanics connected issues with a better level of accuracy were discovered during this decade.

With the event of finite element method, along with tremendous will increases in computing power and convenience, nowadays it is possible to understand structural behavior with levels of accuracy. This was in fact the beyond of imagination before the computer age.

1.3 Numerical methods

The formulation for structural analysis is usuallysupported the three elementary relations: equilibrium, constitutive and compatibility. There are two major approaches to the analysis: Analytical and Numerical. Analytical approach that ends up in closed-form solutions is effective just in case of simple geometry, boundary conditions, loadings and material properties. However, in reality, such straightforward cases might not arise. As a result, varied numerical methods are evolved for finding such issuesthat area unit complicated in nature. For numerical approach, the solutions are approximate once any of those relations are only approximately satisfied. The numerical methodology depends heavily on the process power of computers and is a lot of applicable to structures of absolute size and complexness. It’s common apply to use approximate solutions of differential equations because the basis for structural analysis. This can be sometimes done using numerical approximation techniques. Few numerical methods are commonly used to solve solid and fluid mechanics problems are given below.

- Finite Difference Method
- Finite Volume Method
- Finite Element Method
- Boundary Element Method
- Meshless Method

1.3.1 Finite difference Method:

Finite Differences are just algebraic schemes one can derive to approximate derivatives. The uses of Finite Differences are in any discipline where one might want to approximate derivatives. A common usage is for things like solving Differential Equations numerically, and approximating derivatives for root finding and numerical optimization schemes.

The mentioned areas of use span application in several domains, whether it's engineering, science, business, etc. They are a great tool in several areas, although they aren't essentially the only approach to approximating derivatives. They also aren't the only tools use to solve Differential Equations. But they have their important part to play historically and in many applications where their use makes sense.

The finite difference method is one of a family of methods for approximating the solution of partial differential equations such as heat transfer, stress/strain mechanics problems, fluid dynamics problems, electromagnetics problems, etc.

It is most easily derived using an orthonormal grid system so that, for example in space, the x, y, and potentially the z coordinate (in 3D) are decoupled.

Difference equations based on a difference approximation of the partial differential equations can then be established. This will result in a set of coupled linear equations usually represented as a matrix. Solution of this matrix equation yields an approximate solution to the partial differential equation.

One can build a simple demonstration by using the conduction heat transfer equation applied to a line. Break the line into n segments, assign a value of T to one end, and 0 to the other and solve the resulting difference equations across the assembly of line segments. This is a trivial example but demonstrates the process.

An important application of finite differences is in numerical analysis, especially in numerical differential equations, which aim at the numerical solution of ordinary and partial differential equations respectively. The idea is to replace the derivatives appearing in the differential equation by finite differences that approximate them. The resulting methods are called finite difference methods.

The common applications of the finite difference method are in computational science and engineering disciplines, such as thermal engineering, fluid mechanics, etc.

The application of finite difference method for engineering issues involves commutation the governing differential equations and therefore the boundary condition by appropriate algebraic equations.

1.3.2 Finite Volume Method:

The finite volume method is currently the most popular method in CFD. Generally the finite volume method is a special case of finite element.

Using Finite volume method, the solution domain is subdivided into a finite number of small control volumes by a grid. The gird defines to boundaries of the control volumes while the computational node lies at the center of the control volume. The advantage of FVM is that the integral conservation is satisfied exactly over the control volume.

The basis of the finite volume method is the integral conversation law. The essential idea is to divide the domain into many control volumes and approximate the integral conservation law on each of the control volumes. For example, as shown in figure 1.1,

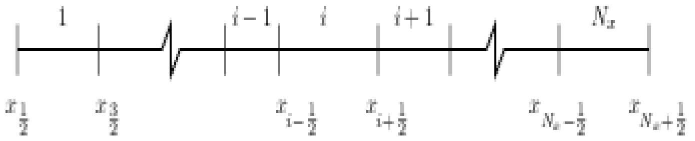


Figure 1.1: Mesh and notation for one-dimensional finite volume method.

The finite-volume method directly utilizes the conservation laws—the integral formulation of the Navier-Stokes/Euler equations. The finite-volume method discretizes the governing equations by first dividing the physical space into a number of arbitrary polyhedral control volumes. The accuracy of the spatial discretization depends on the particular scheme with which the fluxes are evaluated.

There are several possibilities of defining the shape and position of the control volume with respect to the grid. Two basic approaches can be distinguished:

Cell-centered scheme (Fig. 1.2a): Here the flow quantities are stored at the centroids of the grid cells. Thus, the control volumes are identical to the grid cells.

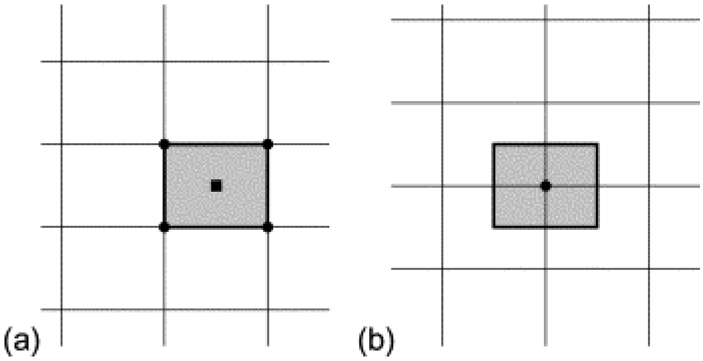


Figure 1.2: Control volume of cell-centered (a) and cell-vertex scheme (b).

Cell-vertex scheme (Fig. 1.2b): Here the flow variables are stored at the grid points. The control volume can then either be the union of all cells sharing the grid point, or some volume centered around the grid point. In the former case we speak of overlapping control volumes, in the second case of dual control volumes.

The important feature of finite volume schemes is their conservation properties. Since they are based on applying conservation principles over each small control volume, global conservation is also ensured.

- Initially we consider how they are applied on rectangular Cartesian grids. In later lectures we see how to adapt them to non-orthogonal and even unstructured grids.

- The method starts by dividing the flow domain into a number of small control volumes.

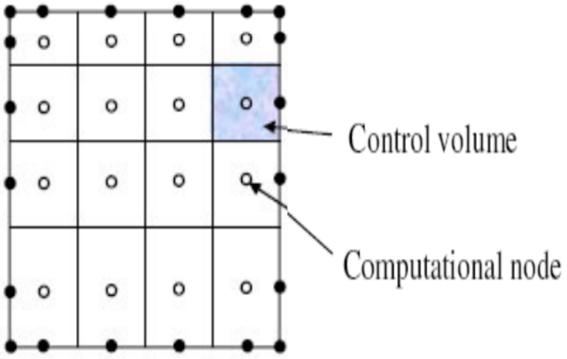


Figure 1.3: Discretization Approaches used in Computational Fluid Dynamics

- The grid points where variables are stored are typically defined as being at the center of each control volume.
- Extra boundary nodes are often added, as shown in the figure 1.3.
- The transport equation(s) are then integrated over each control volume.

1.3.3 Finite Element Method

The FEM method is very suitable for complex structure. In this method the simulation space can be divided with arbitrarily oriented, arbitrary shaped elements, commonly used triangle for 2D and tetrahedra for 3D. Here the governing equation is approximated over each element by some basis functions, which is a low-order polynomial. The solutions are made continuous at the boundaries of each element, and must be fit with in the enforced global boundary conditions. Its only disadvantage is the inherent complexity, as it requires some level of global knowledge of the simulation space. Although the used basis functions are local, as they are defined in each element, to enforce continuity at element boundaries a large sparse matrix needs to be solved, which enhances the computational time a lot.

For last couple of year discontinuous Galerkin methods leap forefront in the field of electromagnetic simulations. By relaxing the continuity between elements this method enforces strict locality rule. This idea of connecting the elements along their boundaries was borrowed from the finite volume method. Thus it provides explicitly localized and highly accurate algorithms. FEM is discussed in details later in Chapter 2.

1.3.4 Boundary Element Method

The boundary element method (BEM) is a technique for solving a range of engineering/physical problems.

It is most often used as an engineering design aid - similar to the more common finite element method - but the BEM has the distinction and advantage that only the surfaces of the domain need to be meshed.

Engineers who are familiar with finite elements very often ask why it is necessary to develop yet another computational technique. The answer is that finite elements have been proved to be inadequate or inefficient in many engineering applications and what is perhaps more important is in many cases cumbersome to use and hence difficult to implement in Computer Aided Engineering systems. Finite Element (FE) analysis is still a comparatively slow process due to the need to define and redefine meshes in the piece or domain under study.

Boundary elements (BE) have emerged as a powerful alternative to finite elements particularly in cases where better accuracy is required due to problems such as stress concentration or where the domain extends to infinity. The most important feature of boundary elements, however, is that different to the finite domain methods as, e.g., the finite difference method or the finite element method, the methodology of formulating boundary value problems as boundary integral equations describes problems only by equations withknown and unknown boundary states.Hence, it only requires discretization of the surface rather than the volume, i.e., the dimension of problems is reduced by one. Consequently, the necessary discretization effort is mostly much smaller and, moreover, meshes can easily be generated and design changes do not require a complete remeshing.

The BE method is especially advantageous in the case of problems with infinite orsemi-infinite domains, e.g., so-called exterior domain problems: there, although only thefinite surface of the infinite domain has to be discretized, the solution at any arbitrary point of the domain can be found after determining the unknown boundary data.

To be objective, the features of the BE method should be compared to its main rival, the FE method. Its advantages and disadvantages can be summarized as follows:

1.3.4.1 Advantages:

1. **Less data preparation time:** This is adirect result of the ’surface-only’ modeling. Thus, the analyst’s time required for data preparation and data checking for a given problem should be greatly reduced. Furthermore, subsequent changes in meshes are made easier.

2. **High resolution of stress:** Stresses are accurate because no further approximation is imposed on the solution at interior points, i.e., solution is exact and fully continuous inside the domain.

3. **Less computer time and storage:** For the same level of accuracy, the BE method uses a lesser number of nodes and elements (but a fully populated matrix), i.e., to achieve comparable accuracy in stress values, FE meshes would need more boundary divisions than the equivalent BE meshes.

4. **Less unwanted information:** In most engineering problems, the ’worst’ situation usually occur on the surface.

Thus, modeling an entire three-dimensional body with finite elements and calculating stress (or other states) at every nodal point is very inefficient because only a few of these values will be incorporated in the design analysis. Therefore, using boundary elements is a very effective use of computing resources, and, furthermore, since internal points in BE solutions are optional, the user can focus on a particular interior region rather than the whole interior.

1.3.4.2 Disadvantages

1. **Unfamiliar mathematics:** The mathematics used in BE formulations may seem unfamiliar to engineers (but not difficult to learn). However, many FE numerical proceduresare directly applicable to BE solutions (such as numerical integration, surface approximation, treatment of boundary conditions).

2. **In non-linear problems, the interior must be modeled:** Interior modeling is unavoidable in non-linear material problems. However, in many non-linear cases interior modeling can be restricted to selected areas such as the region around a crack tip.

3. **Fully populated and unsymmetrical solution matrix:** The solution matrix resulting from the BE formulation is unsymmetrical and fully populated with non-zero coefficients, whereas the FE solution matrices are usually much larger but sparsely populated. This means that the entire BE solution matrix must be saved in the computer core memory.

However, this is not a serious disadvantage because to obtain the same level of accuracy as the FE solution, the BE method needs only a relatively modest number of nodes and elements.

4. **Poor for thin structures (shell) three-dimensional analyses:** This is because of the large surface/volume ratio and the close proximity of nodal points on either side of the structure thickness. This causes inaccuracies in the numerical integrations.

1.3.5 Meshless Methods

Meshless methods belong to a class of techniques for solving boundary/initial value partial differential equations in which both geometry representation and numerical discretization are principally performed based on nodes or particles. In meshless methods, there is no inherent reliance on a particular mesh topology, meaning that no element connectivity is required. In practice, however, in many meshless methods, recourse must be taken to some kind of background meshes at least in one stage of the implementation.

Analysis of many practical processes in modern engineering requires modeling of problems with time-dependent geometry or boundary conditions. Pulsating flow of blood in heart, metal-forming processes, and stretching of a polymer filament are only a few examples. Conventional mesh-based methods such as the finite volume and finite element methods face serious difficulties when dealing with large element deformations and/or element entanglement. This is particularly true in the case of problems involving discontinuities and moving boundaries. Examples of such problems are complex free-surface flows, fluid–structure interactions, projectile impact, and material breakup during manufacturing processes. Standard mesh-based techniques handle such problems using adaptive remeshing techniques. This means that the computational mesh (grid) is remeshed either globally or locally to correctly represent the deformed geometry. This technique however suffers from two drawbacks. Firstly, generating a new mesh may not only be troublesome but also consume considerable time. This is particularly true when complex three-dimensional geometries are to be meshed using automatic mesh generators. Secondly, the mapping of the state variables from the old mesh to the new mesh can introduce significant numerical errors into the computations. To alleviate this problem, special numerical techniques have been devised in which the material and the mesh move independently.

In recent years, interest in meshless methods has grown rapidly because such methods can circumvent the abovementioned difficulties in a more convenient fashion. The main advantages of meshless methods can be summarized as follows:

- Problems with large deformations can be handled since the connectivity among nodes is generated during the computation and can change in time.

- There are no constraints imposed from the system geometry, and the system may evolve far from the initial conditions.

- The accuracy of the solution can be controlled easily, since nodes can be added in the regions where refinement is required.

- Complex geometries can be accurately represented by particles.

- The particles map onto the mass density of the fluid, leading automatically to higher resolution in the high-density regions.

- Fairly simple implementation procedure.

Abundance of meshless methods, as well as many variations in the terminology adopted in the literature, makes a thorough review unfeasible for the purpose of this manuscript. Instead, this entry attempts to give an overview of this class of methods by briefly describing some of their most important features.

Based on the physical principle, meshless (particle) methods can be classified as deterministic and probabilistic. Many of the meshless methods are based on probabilistic principles. The molecular dynamics, Monte Carlo methods, the Lagrangian probability density function (PDF) methods, and the Lattice Boltzmann (LBM) method are among these methods. Methods such as the smoothed particle hydrodynamics (SPH) and the vortex method initially developed as probabilistic methods, but nowadays they are most frequently used as deterministic.

The meshless methods can be also classified as strong and weak formulations of the associated partial differential equations. The strong formulation of a partial differential equation is usually obtained by a collocation technique. The smoothed particle hydrodynamics method, the vortex method, and the generalized finite difference (GFD) method are based on the strong form. On the other hand, weak forms are often based on Galerkin formulations. The diffuse element method (DEM), the element-free Galerkin method (EFGM), the reproducing kernel particle method (RKPM) (Kernel Function), the hp cloud method, the partition of unity method (PUM), and the meshless local Petrov-Galerkin method (MLPG) are all among this category. Some particle methods like the particle-in-cell method can be used in both strong and weak formulations.

1.4 Concepts of elements and nodes

1.4.1Nodes

A node is a coordinate location in space where the degrees of freedom (DOFs) are defined. The DOFs for this point represent the possible movement of this point due to the loading of the structure. The DOFs also represent which forces and moments are transferred from one element to the next. The results of a finite element analysis, (deflections and stresses), are usually given at the nodes.

In the real world, a point can move in 6 different directions, translation in X, Y, and Z, and rotation about X, Y, and Z. In FEA, a node may be limited in the calculated motions for a variety of reasons. For example, there is no need to calculate the out of plane translation on a 2-D element; it would not be a 2-D element if its nodes were allowed to move out of the plane.

The DOF of a node (which is based on the element type) also relates what types of forces and restraints are transmitted through the node to the element. A force (axial or shear) is equivalent to a translation DOF. A moment is equivalent to a rotational DOF. Thus, to transfer a moment about a certain axis, the node must have a rotational DOF about the axis. If a node does not have that rotational DOF, then applying a moment to the node will have no effect on the analysis. This fact may also place requirements on how two parts are connected together. Additional modeling may be required to insure that the connection between the parts does not produce a hinge.

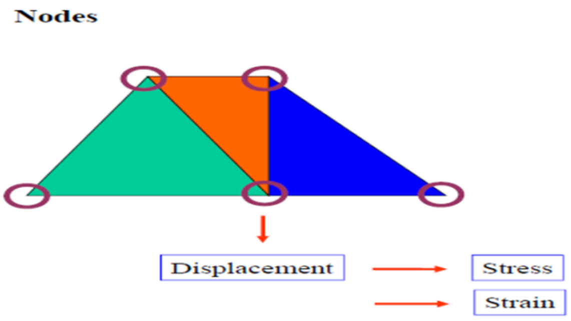


Figure 1.4: Nodes

1.4.2 Elements

An element is the basic building block of finite element analysis. There are several basic types of elements. Which type of element for finite elements analysis that is used depends on the type of object that is to be modeled for finite element analysis and the type of analysis that is going to be performed.

An element is a mathematical relation that defines how the degrees of freedom of a node relate to the next. These elements can be lines (trusses or beams), areas (2-D or 3-D plates and membranes) or solids (bricks or tetrahedrals). It also relates how the deflections create stresses.

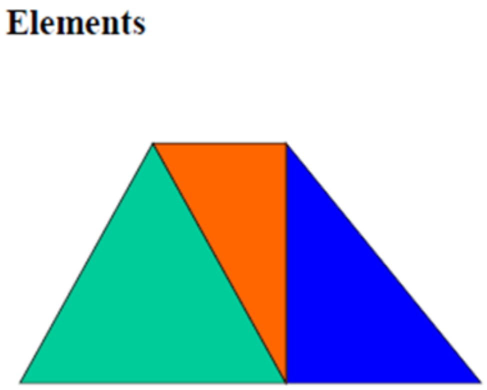


Figure1.5: Elements

Chapter 2

Finite Element Method (FEM)

2.1Finite Element Method (FEM)

The Finite Element Method (FEM) is one of the most potent and resourceful numerical method for solving problems of engineering and mathematical physics. In the year 1960 the term ‘Finite Elements’ was first introduced by Clough in defining a new technique for plane stress analysis.

As a numerical technique finite element method provides an approximate solution of a problem base on a set of governing equations, generally, in the algebraic, integral or differential forms through a discretization process of the domain in interest. Its main characteristicsistodiscretizethedomainoftheproblemintoasetofsmallersub-domainsor elements.

Two well-known features of finite element method may help in discriminating its dominance over other methods. Firstly, in this method any geometrically complex domain can be assumed as disintegration of simple sub-domains named finite-element, which can be of different shapes, for instance, triangular or rectangular. Secondly, a fundamental idea that any continuous function can be correspondent to by a linear combination of algebraic polynomials governs the derivation of the approximation functions over each finite-element.

The FEM always follows an orderly step-by-step process in providing solution of any problem.

2.2 Analytical Solution

• Stress analysis for trusses, beams, and other simple structures are carried out based on dramatic simplification and idealization:

- mass concentrated at the center of gravity
- beam simplified as a line segment (same cross-section)

• Design is based on the calculation results of the idealized structure & a large safety factor is given by experience.

2.3 FEM

• Design geometry is a lot more complex; and the accuracy requirement is a lot higher. We need

- To understand the physical behaviors of a complex object (strength, heat transfer capability, fluid flow, etc.)
- To predict the performance and behavior of the design; to calculate the safety margin; and to identify the weakness of the design accurately, and
- To identify the optimal design with confidence

2.4 Common FEM Applications

Below some applications of FEM analysis area are stated.

*****Mechanical/Aerospace/Civil/AutomotiveEngineering

*Structural/Stress Analysis

- Static/Dynamic
- Linear/Nonlinear

*Fluid Flow

*Heat Transfer

*Electromagnetic Fields

*Soil Mechanics

*Acoustics

*Biomechanics

2.5 Types of Finite Elements

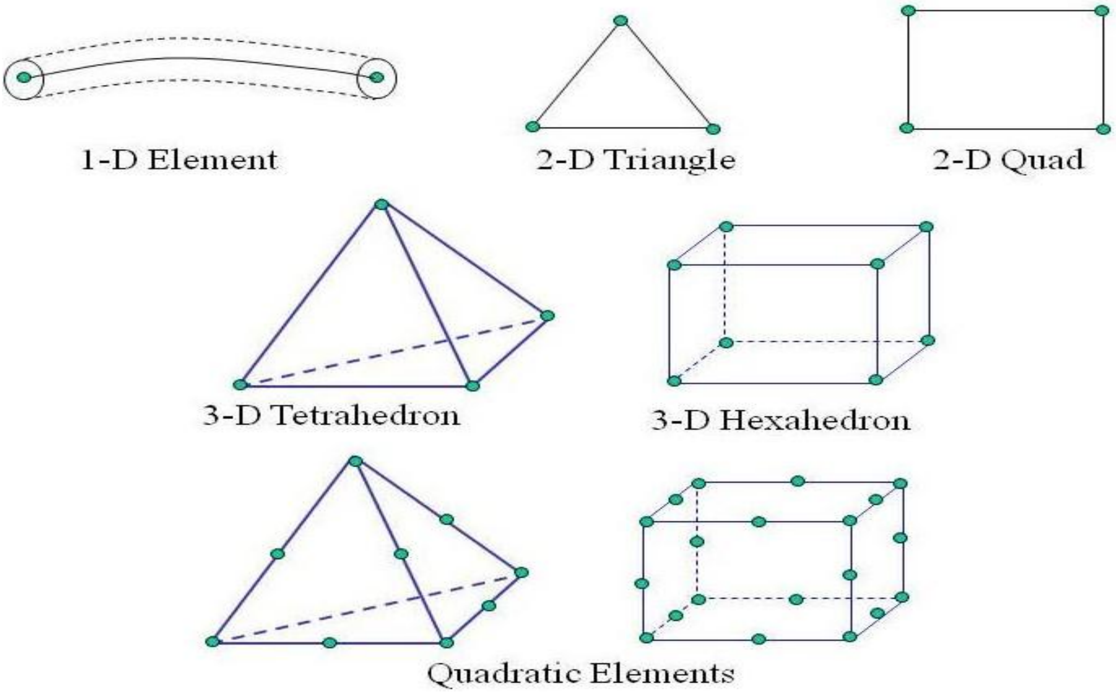


Fig. 2.1 Types of Finite Elements

2.6 Advantages of Finite element Method

1.Modeling of complex geometries and irregular shapes are easier as varieties of finite elements are available for discretization of domain.

2.Boundary conditions can be easily incorporated in FEM.

3.Different types of material properties can be easily accommodated in modeling from element to element or even within an element.

4.Higher order elements may be implemented.

5.FEM is simple, compact and result-oriented and hence widely popular among engineering community.

6.Availability of large number of computer software packages and literature makes FEM a versatile and powerful numerical method.

2.7 Principles of FEM

The finite element method (FEM), or finite element analysis (FEA), is a computational technique used to obtain approximate solutions of boundary value problems in engineering. Boundary value problems are also called field problems. The field is the domain of interest and most often represents a physical structure.

The field variables are the dependent variables of interest governed by the differential equation.

The boundary conditions are the specified values of the field variables (or related variables such as derivatives) on the boundaries of the field.

For simplicity, at this point, we assume a two-dimensional case with a single field variable φ(x, y) to be determined at every point P(x, y) such that a known governing equation (or equations) is satisfied exactly at every such point.

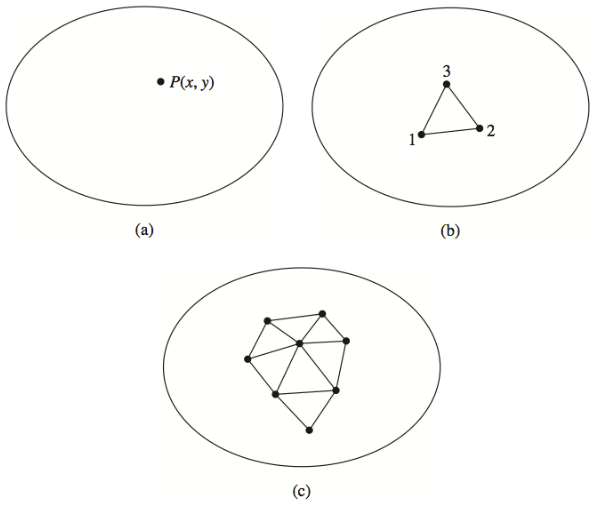


Fig: 2.2

(a) A general two-dimensional domain of field variable Ø(X, Y),
(b) A three-node finite element defined in the domain.
(c) Addition elements showing a partial finite element mesh of the domain.

A finite element is a differential element of size dx × dy.

- A node is a specific point in the finite element at which the value of the field variable is to be explicitly calculated.

2.8 A GENERAL PROCEDURE FORFINITE ELEMENT ANALYSIS

2.8.1 Preprocessing

– Define the geometric domain of the problem.

– Define the element type(s) to be used (Chapter 6).

– Define the material properties of the elements.

– Define the geometric properties of the elements (length, area, and the like).

– Define the element connectivity (mesh the model).

– Define the physical constraints (boundary conditions). Define the loadings.

2.8.2 Solution

– computes the unknown values of the primary field variable(s)

– computed values are then used by back substitution to compute additional, derived variables, such as reaction forces, element stresses, and heat flow.

2.8.3 Post processing

– Postprocessor software contains sophisticated routines used for sorting, printing, and plotting selected results from a finite element solution.

2.9 Stiffness Matrix

The primary characteristics of a finite element are embodied in the element stiffness matrix. For a structural finite element, the stiffness matrix contains the geometric and material behavior information that indicates the resistance of the element to deformation when subjected to loading. Such deformation may include axial, bending, shear, and torsional effects. For finite elements used in nonstructural analyses, such as fluid flow and heat transfer, the term stiffness matrix is also used, since the matrix represents the resistance of the element to change when subjected to external influences.

2.10Discretization of domain and generalization of an element:

The fundamental concept in FEM is to divide the domain of interest into a finite number of sub-domains or elements. Some elements in the domain may share nodal points and element boundaries. Thus the domain looks like a collection of elements as shown in Fig. 3.2. Element boundaries of an finite element are, generally straight lines. When the domain of interest has curved or irregular boundaries, approximation at the edge is made by using series of straight or flat segments or even by the use of isoparametric elements.

In implementation of FEM, after the domain of interest has been discretized by proper choice of elements, the unknown field variable's behavior over each element is approximated by continuous functions represented in terms of nodal values of the field variable and sometimes by its derivatives of certain order. The function defined over each element is named as shape function or interpolation function. The collection of the interpolation function for the domain of interest as a whole provide a piecewise approximation of the filed variable for that domain. In this thesis triangular elements are used to discretize two dimensional waveguide.

2.10.1Shape functions:

After discretization of the domain the unknown filed variable values at the vertices of the triangular shaped element is approximated by a set of polynomial approximation named the shape function. Because polynomials can be easily manipulated both algebraically and computationally, furthermore, any continuous function may be arbitrarily approximated closely by a suitable polynomial, so these are used as shape function. These shape functions are considered to be continuous within the element and across the element boundaries as well. It would not be possible to add separate contribution from each element to get the final solution without inter-element continuity. The physical significance is that the shape functions should possess continuity characteristics similar to that of the physical fields they approximated.

Although it is preferred that interpolation elements to be isotropic or geometrically invariant, however, shape functions may not be invariant. To ensure the solution isotropy, in each element the polynomial expression must be complete, without any preference for either x or y-directions. It means, if the function has an order q, the polynomial that approximating it should contain all possible terms xa1yb1 such that,

$0 \leq a1 + b1 \leq q$ ……… (2.4)

This type of polynomial should contain l = 1 2(q+1)(q+2) terms. The polynomial to be unique the number of terms in it should be equal to the total number of degree of freedom, i.e., the number of nodes associated with the element. For instance, a triangular element requires a polynomial consists of three terms. Equation 3.18 represents the relationship between the nodes in an element and the order of the shape function. The number of terms necessary for all possible polynomials up to the 3rd order is shown in Fig. 3.3 by Pascal triangle.

The triangle is one the most commonly used elements in two dimensional structure, although, rectangles and even quadrilaterals are also used. In this thesis triangle is used as element and first degree polynomials are used as shape functions, which consists of three coefficients, as stated earlier. These can be corresponded to the three nodal values of the triangular element's vertices.

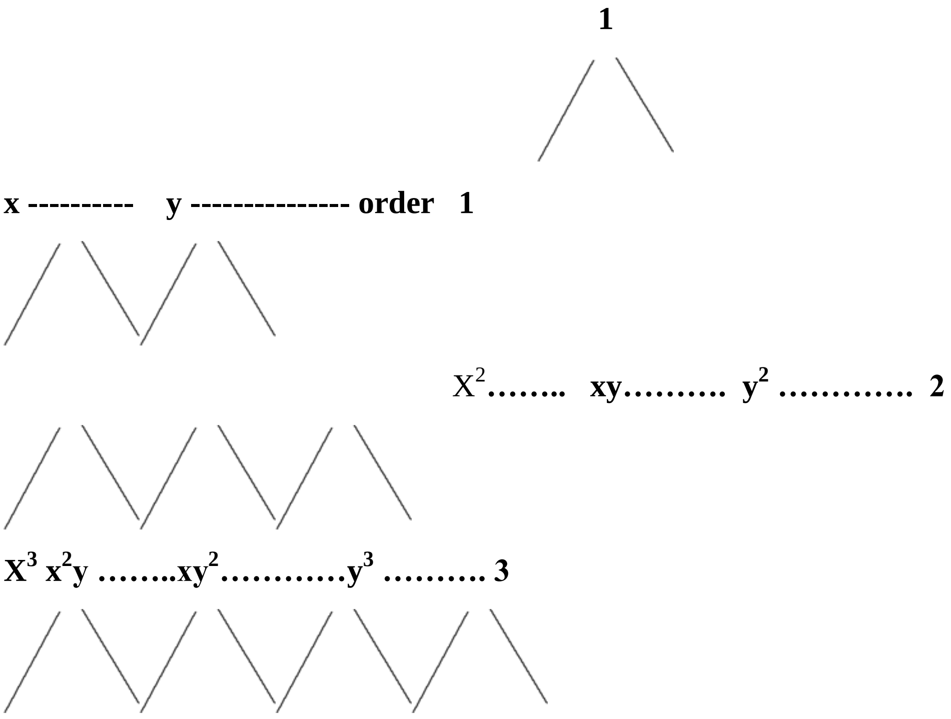

1
x ---------- y --------------- order 1
X^2…….. xy………. y^2 …………. 2
X^3 x^2y ……..xy^2…………y^3 ………. 3

Fig : 2.3 the Pascal triangle

Let us consider the continuous field function in the domain of interest is Φ(x,y), which can be replaced by set of discrete values (Φ_e, where e = 1, 2 and 3) as shown in Fig. 2.4. Such functions are considered as continuous across adjacent triangles. Field across the elements boundaries should be continuous for these functions to be permissible. A first degree polynomial of the type a + bx + cy can be used to represent the field.

The field in first order element can therefore be expressed as,

$$\Phi_e(x_i,y_i)= a^e + b^e x + c^e y \text{…...} \quad (2.5)$$

Here a^e , b^e and c^e are constants. At each of three vertices of the triangle we, then, have,

$$\Phi_e(x_i,y_i)=\Phi_i \qquad i= 1,2,3 \quad \text{……..} (2.6)$$

Then Φ_i, the nodal values can be expressed as,

$$\Phi_1 = \Phi_e(x_i,y_i)= a^e + b^e x_1 + c^e y_1 \ \text{………} (2.7)$$

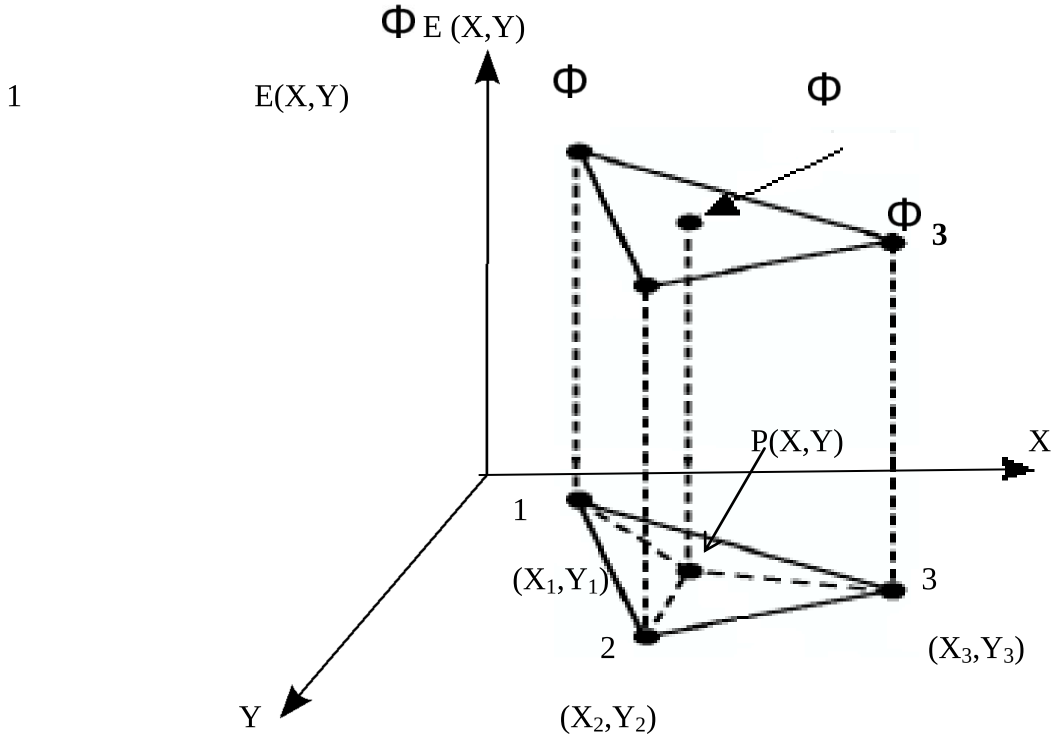


Fig:2.4A typical first order triangular element

$$\Phi_2 = \Phi_e(x_i,y_i)= a^e + b^e x_2 + c^e y_2 \ \text{……….} (2.8)$$

$$\Phi_3 = \Phi_e(x_i,y_i)= a^e + b^e x_3 + c^e y_3 \ \text{………..} \ (2.9)$$

In matrix Form,

$$\begin{bmatrix}\Phi 1\\ \Phi 2\\ \Phi 3\end{bmatrix}=\begin{bmatrix}1 & x1 & y1\\ 1 & x2 & y2\\ 1 & x3 & y3\end{bmatrix}\begin{bmatrix}a^e\\ b^e\\ c^e\end{bmatrix}\text{........ (2.10)}$$

In terms of Φ_i the constants a^e, b^e and c^e can be found as,

$$\begin{bmatrix}a^e\\ b^e\\ c^e\end{bmatrix}=\begin{bmatrix}1 & x1 & y1\\ 1 & x2 & y2\\ 1 & x3 & y3\end{bmatrix}^{-1}\begin{bmatrix}\Phi 1\\ \Phi 2\\ \Phi 3\end{bmatrix}\text{........... (2.11)}$$

In expanded form we can write from Eq. (2.11)

$$a^e = \frac{1}{2A_e}\left[\Phi_1(x2y3-x3y2)+\Phi_2(x3y1-x1y3)+\Phi_3(x1y2-x2y1)\right]\text{.......}\quad (2.12)$$

$$b^e = \frac{1}{2A_e}\left[\Phi_1(y2-y3)+\Phi_2(y3-y1)+\Phi_3(y1-y2)\right]\text{.........}\qquad (2.13)$$

$$c^e = \frac{1}{2A_e}\left[\Phi_1(x3-x2)+\Phi_2(x1-x3)+\Phi_3(x2-x1)\right]\qquad\text{.........}\qquad (2.14)$$

Here A_e is the area of the triangular element and can be presented as,

$$A_e = \frac{1}{2}\begin{vmatrix}1 & x1 & y1\\ 1 & x2 & y2\\ 1 & x3 & y3\end{vmatrix} = \frac{1}{2}\left[(x2y3-x3y2)+(x3y1-x1y3)+(x1y2-x2y1)\right]\text{....... (2.15)}$$

Substituting for a^e, b^e , c^e from Eq. (2.11) – (2.14) into Eq. (2.5) and rearranging will provide,

$$\Phi_e(x,y) = N_1(x,y)\cdot\Phi_1+N_2(x,y)\cdot\Phi_2+N_3(x,y)\cdot\Phi_3\quad\text{.......... (2.16)}$$

Or,

$$\Phi_e(x,y) = [N]\,[\Phi_e]\quad\text{.................. (2.17)}$$

where [N] is the shape function matrix and the column vector $[\Phi_e]$ is the vector corresponding to the element nodal field values.

Therefore, after discretizing the domain by using small triangular elements, the unknown field Φ_e in every element, can be written in terms of an interpolation of the field values ateach node, as given by Eq. (2.17). It can be shown that the element shape function can be written in the matrix notation form [Davies, 1989] as,

$$[N]^T = \begin{bmatrix} N1 \\ N2 \\ N3 \end{bmatrix} = \frac{1}{2A_e} \begin{bmatrix} x2y3 - x3y2 & y2 - y3 & x3 - x2 \\ x3y1 - x1y3 & y3 - y1 & x1 - x3 \\ x1y2 - x2y1 & y1 - y2 & x2 - x1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \ldots\ldots \quad (2.18)$$

Here T represents the transpose . This shape function matrix can also be expressed as,

$$[N]^T = \begin{bmatrix} N1 \\ N2 \\ N3 \end{bmatrix} = \frac{1}{2A_e} \begin{bmatrix} a_1^e & b_1^e x & c_1^e y \\ a_2^e & b_2^e x & c_2^e y \\ a_3^e & b_3^e x & c_3^e y \end{bmatrix} \ldots\ldots\ldots \quad (2.19)$$

The coefficients a_1^e , b_1^e and c_1^e can be calculated as,

a_1^e = x2y3 – x3y2 ………… (2.20)

b_1^e = y2 - y3 ………… (2.21)

c_1^e = x3 – x2 ………….. (2.22)

With cyclic exchange of 1→ 2→ 3 in Eq. (2.20)-(2.22), the other six coefficients can also be found.

The shape function N_i has useful property of taking the value 1 at the node I and 0 at all other nodes as,

$$N_i\,(x_j,y_j) = \delta_{ij} = \begin{cases} 1 \text{ for } i = j \\ 0 \text{ for } i \neq j \end{cases} \ldots\ldots(2.23)$$

2.10.2 Relationshipoflinearshapefunctionswithareacoordinates

Let us consider an arbitrary point P(x,y) inside the bottom triangle shown in Fig. 2.4. The area coordinates functions Li can be represented by utilizing the areas of the triangles as,

$$L_1 = \frac{\text{Area of the sub triangle P23}}{\text{Area of the full triangle 123}} \qquad \text{………} \ (2.24)$$

in similar way L_2 and L_3 can also be defined. The area coordinates functions Li has the following property,

$$\sum_{i=1}^{3} Li = 1 \qquad \text{……..}(2.25)$$

The perpendicular distance from P to side 23 is proportional to L1, which takes the value of1atnode-1and0intheothernodes. So,itisauniquefirstdegreeinterpolatingpolynomial for node-1. Similarly, L2 and L3 can also be defined. The interpolating polynomials or local area coordinates, Li, can be related to the global Cartesian coordinates by interpolation polynomial properties by,

$$x = x_1L_1 + x_2L_2 + x_3L_3 \qquad \text{…………..} \ (2.26)$$

$$y = y_1L_1 + y_2L_2 + y_3L_3 \qquad \text{………..} \ (2.27)$$

The Equation. (2.26) and (2.27) can be expressed in matrix form as,

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x1 & x2 & x3 \\ y1 & y2 & y3 \end{bmatrix} \begin{bmatrix} L1 \\ L2 \\ L3 \end{bmatrix} \qquad \text{……………} \ (2.28)$$

By using inverse transformation from Eq. (2.28), we have,

$$\begin{bmatrix} L1 \\ L2 \\ L3 \end{bmatrix} = \frac{1}{2A_e} \begin{bmatrix} a_1^e & b_1^e & c_1^e \\ a_2^e & b_2^e & c_2^e \\ a_3^e & b_3^e & c_3^e \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \qquad \text{…………} \ (2.29)$$

In terms of the normalized local area coordinates the operation like integration in global coordinatescanbeattained, whereitneedstoperformonlyonce. Mostofthosecanbefoundin

[Zienkiewicz, Book, 2005]. For instance, to obtain integration, following equation can be used

$$\iint_e L_1^d L_2^g L_3^f \, dxdy = 2\mathrm{A}_e \frac{d!g!f!}{(d+g+f+2)!} \qquad \text{………………}(2.30)$$

For the first order polynomial interpolation function, both area coordinates and shape functions are identical, i.e., Ni = Li. The area coordinates are useful in constructing higher order shape functions.

Chapter 3

Boundary value problemAndGalerkin method

3.1 Boundary Value Problem:

A boundary value problem is a problem, typically an ordinary differential equation or a partial differential equation, which has values assigned on the physical boundary of the domain in which the problem is specified. For example,

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - \nabla^2 u = f & \text{in } \Omega \\ u\,(0,\, t) = u_1 & \text{on } \partial\Omega \\ \dfrac{\partial u}{\partial t}\,(0,\, t) = u_2 & \text{on } \partial\Omega, \end{cases}$$

Where∂Ω denotes the boundary of Ω, is a boundary problem.

In mathematics, in the field of differential equations, a boundary value problem is a differential equation together with a set of additional constraints, called the boundary conditions. A solution to a boundary value problem is a solution to the differential equation which also satisfies the boundary conditions.

Boundary value problems arise in several 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. A large number of important boundary value problems are the Sturm–Lowville problems. The analysis of these problems involves the Eigen functions of a differential operator.

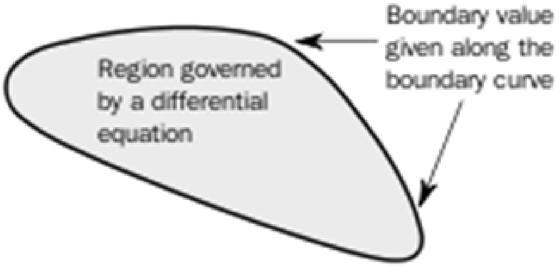

Boundary value given along the boundary curve
Region governed by a differential equation

Fig 3.1

3.1.1 Explanation

Boundary value problems are similar to initial value problems. A boundary value problem has conditions specified at the extremes ("boundaries") of the independent variable in the equation whereas an initial value problem has all of the conditions specified at the same value of the independent variable (and that value is at the lower boundary of the domain, thus the term "initial" value). A boundary value is a data value that corresponds to a minimum or maximum input, internal, or output value specified for a system or component.[1]

For example, if the independent variable is time over the domain [0,1], a boundary value problem would specify values for y(t) at both t=0 and t=1, whereas an initial value problem would specify a value of)} y(t) and y'(t) at time t=0.

Finding the temperature at all points of an iron bar with one end kept at absolute zero and the other end at the freezing point of water would be a boundary value problem. If the problem is dependent on both space and time, one could specify the value of the problem at a given point for all time or at a given time for all space. Concretely, an example of a boundary value (in one spatial dimension) is the problem

$y''(x)+y(x)=0$

to be solved for the unknown function y(x) with the boundary conditions

$y(0)=0,y(\pi/2)=2.$

Without the boundary conditions, the general solution to this equation is

$y(x)=A\sin(x)+B\cos(x).$

From the boundary condition y(0)=0 one obtains

0= A. 0+B.1

Which implies that B=0. From the boundary condition y(π/2)=2 one finds

2=A. 1

and so A=2. One sees that imposing boundary conditions allowed one to determine a unique solution, which in this case is

y(x)=2\sin(x)

3.1.2 Boundary Conditions:

There are three types of boundary conditions commonly encountered in the solution of partial differential equations:

1. Dirichlet boundary conditions specify the value of the function on a surface T=f(r,t).

2. Neumann boundary conditions specify the normal derivative of the function on a surface,

$\frac{\partial T}{\partial n}=\hat{n} \cdot \nabla T= f(r,t)$.

3. Robin boundary conditions. For an elliptic partial differential equation in a region Ω, Robin boundary conditions specify the sum of α u and the normal derivative of u=f at all points of the boundary of Ω, with α and f being prescribed.

3.1.3 Initial Value Problem

An initial value problem is a problem that has its conditions specified at some time t=t_0. Usually, the problem is an ordinary differential equation or a partial differential equation. For example,

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - \nabla^2 u = f & \text{in } \Omega \\ u = u_0 & t = t_0 \\ u = u_1 & \text{on } \partial\Omega, \end{cases}$$

Where $\partial\Omega$ denotes the boundary of Ω, is an initial value problem.

3.1.4 Numerical Analysis of Boundary Value Problem

3.1.4.1 Shooting Method

In numerical analysis, the shooting method is a method for solving a boundary value problem by reducing it to the system of an initial value problem. Roughly speaking, we 'shoot' out trajectories in different directions until we find a trajectory that has the desired boundary value. The following exposition may be clarified by this illustration of the shooting method.

For a boundary value problem of a second-order ordinary differential equation, the method is stated as follows. Let

y''(t)=f(t, y(t), y'(t)), y(t_o) = y_o, y(t_1) =y_1

be the boundary value problem. Let y(t; a) denote the solution of the initial value problem

y''(t)= f(t,y(t), y'(t)), y(t_o) = y_o, y'(t_o)= a

Define the function F(a) as the difference between y(t_1; a) and the specified boundary value y1.

F(a)= y(t_1;a) - y_1

If F has a root a then the solution y(t; a) of the corresponding initial value problem is also a solution of the boundary value problem. Conversely, if the boundary value problem has a solution y(t), then y(t) is also the unique solution y(t; a) of the initial value problem where a = y'(t_o), thus a is a root of F.

The usual methods for finding roots may be employed here, such as the bisection method or Newton's method.

3.1.4.2 Example

A boundary value problem is given as follows by Stoer and Burlisch[1] (Section 7.3.1).

$\omega''(t) = \frac{3}{2}\omega^2$, $\omega(0)=4$, $\omega(1)=1$

The initial value problem

$\omega''(t) = \frac{3}{2}\omega^2$, $\omega(0) = 4$, $\omega'(0) = s$

was solved for s = −1, −2, −3, ..., −100, and F(s) = ω(1;s) − 1 plotted in the first figure. Inspecting the plot of F, we see that there are roots near −8 and −36. Some trajectories of ω(t;s) are shown in the second figure.

Stoer and Burlisch[1] state that there are two solutions, which can be found by algebraic methods. These correspond to the initial conditions w′(0) = −8 and w′(0) = −35.9 (approximately).

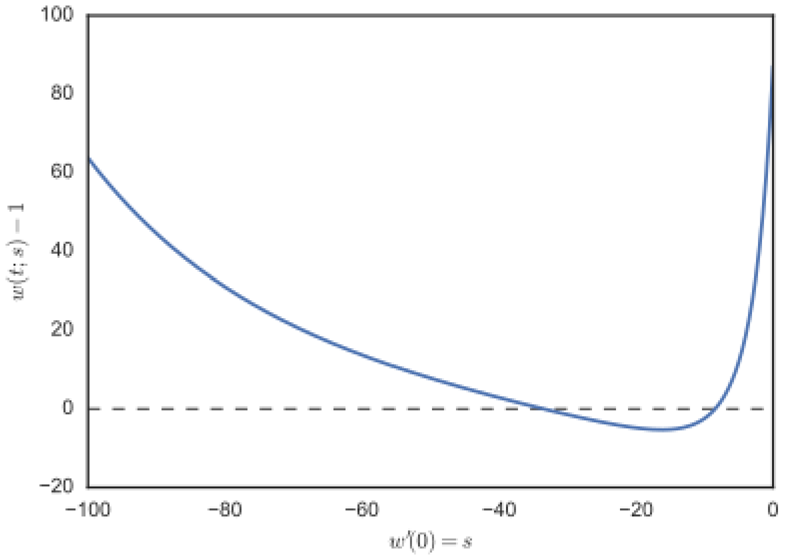


Fig 3.2

The function F(s) = w(1;s) − 1.

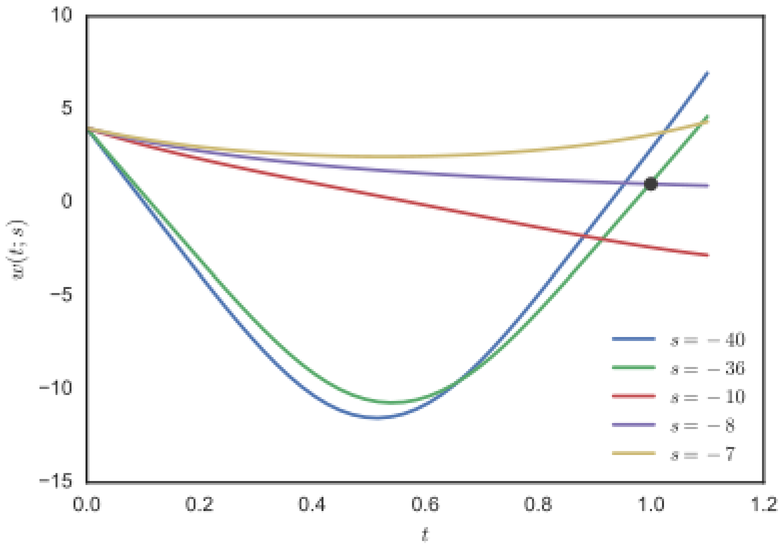


Fig: 3.3

Trajectories ω(t;s) for s = ω'(0) equal to −7, −8, −10, −36, and −40. The point (1,1) is marked with a circle.

3.2 Galerkin Method:

The Galerkin method is one of the most popular and powerful numerical techniques for solving transient partial differential equations of parabolic type. The Galerkin method can be viewed as a separation-of-variables technique combined with a weak finite element formulation to discretize the problem in space. This leads to a stiff system of ordinary differential equations that can be integrated by available off-the-shelf implicit ordinary differential equations (ode) solvers based on implicit time-stepping schemes such as backward differentiation formulas (BDF) or implicit Runge-Kutta (IRK) methods.

3.2.1 A one-dimensional parabolic problem

The basic idea of the Galerkin finite element method of lines will be demonstrated on the following one-dimensional linear parabolic partial differential problem

$$\begin{cases} u_t(x,t) - (a(x)u_x(x,t))_x + b(x)u(x,t) = f(x,t), x \in (0,1), 0 \leq t \leq T, \\ u(x,0) = g(x), x \in (0,1), \\ u(0,t) = u(1,t) = 0, 0 \leq t \leq T, \end{cases} \quad \ldots\ldots\ldots\ldots(1)$$

Where, a(x)>0 and b(x)≥0 on [0,1] .

3.2.2 Weak Galerkin formulation

A spatial weak formulation of the model problem is obtained by multiplying the parabolic equation (1) by a test function v in the SobolevspaceH_0^1, integrating over (0,1) and integrating by parts to show that u∈ H_0^1 satisfies

$$\begin{cases}(v, u_t(.,t)) + A(v, u(.,t)) = (v, f(.,t)), \forall\, v \in H_0^1, 0 \leq t \leq T, \\ u(x,0) = g(x), x \in (0,1), \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Where,

(v, u(.,t))= $\int_0^1 v(x)u(x,t)dx, A(v, u(.,t))$=$\int_0^1 [a(x)\frac{dv(x)}{dx}u_x(x,t) + b(x)v(x)u(x,t)]dx.$

3.2.3 Semi-discretization

The semi-discrete Galerkin finite formulation consists of subdividing the domain (0,1) into n elements I_i=(x_i, x_{i+1}), i=0,1,⋯,n and constructing a finite-dimensional subspace $S_p^N \subset H_0^1$. Here S_p^N consists of piecewise p-degree polynomial functions spanned by a set of basis functionsϕ_i, i=1,2,⋯,N . For instance, N=n−1, for piecewise linear finite elements.

The semi-discrete Galerkin method consists of finding

u_N(x,t) = $\sum_{i=1}^{N} c_i(t)\phi_i(x),$

with time-dependent coefficients c_i(t)∈R such that

$$\begin{cases}(v_N, (u\_N)_t(.,t)) + A(v_N, u_N(.,t)) = (v_N, f(.,t)), \forall\, v_N \in S_p^N, 0 \leq t \leq T, \\ (v_N, u_N(.,0) = (v_N, g), \forall v_N \in S_p^N. \end{cases}$$

Testing against $v_N = \phi_j$, j=1,2,⋯,N , leads to the following system of linear ordinary differential equations for c_i, i=1,⋯,N ,

$$\begin{cases} \sum_{i=1}^{N} c_i \, (\phi_j, \phi_i) + \sum_{i=1}^{N} c_i \, A(\phi_i, \phi_j) = (\phi_j, f(.,t)), j = 1,2,\cdots,N, t \geq 0, \\ \sum_{i=1}^{N} c_i \, (0)(\phi_j, \phi_i) = (\phi_j, g), j = 1,2,\cdots,N \end{cases} \qquad \ldots\ldots\ldots\ldots..(3)$$

Other optimal approximations of the initial condition include the elliptic projection of g(x)onto S_p^N defined by the equation

A(u_N(.,0),v_N)=A(g,v_N), $\forall v_N \in S_p^N$,

and the piecewise polynomial Lagrange interpolant of g defined by

u_N(x,0)=$\sum_{i=1}^{N} g(x_i)\phi_i(x),$

Where,ϕ_i are finite element basis functions of Lagrange type, i.e.,$\phi_i(x_j)$= δ_{ij} . Here δ_{ij}=1 if I =j and 0 otherwise.

Let c(t)= $[c_1(t),\ c_2(t),\cdots,c_N(t)]^t$; thus, the linear ordinary differential system (3) may be written in matrix form as

$$\begin{cases} Mc^{\cdot} + Kc(t) = f(t), 0 \leq t \leq T, \\ Mc(0) = g, \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..\ldots..(4)$$

Where, the N×N mass matrix M and stiffness matrix K defined by M_{ij}=M_{ji}=(ϕ_j,ϕ_i) and K_{ij}=K_{ji}=A(ϕ_j,ϕ_i) are symmetric positive definite. The vectors f, g$\in R^N$ are defined by f_j(t)=(ϕ_j,f(.,t)) and g_j=(ϕ_j,g) .

The initial-value problem (4) has a unique solution.

The mass matrix M may be replaced by a diagonal matrix using a low-order quadrature to approximate the L^2 inner product, which is easier to invert. This new method is called lumped masses finite element method.

3.2.4 Multidimensional problems

A Galerkin finite element method for multi-dimensional parabolic problems is derived in a similar manner, as illustrated on the following model problem:

$$\begin{cases} u_t(x,t) = \Delta u(x,t) + f(x,t), (x,t) \in \Omega \times [0,T], \\ u(x,t) = 0, (x,t) \in \partial\Omega \times [0,T], \\ u(x,0) = g(x),\ x \in \Omega, \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

Where, $\Omega \subset R^d$.

The domain Ω is partitioned into a regular mesh consisting of triangular or rectangular elements (in two dimensions) and tetrahedral or hexahedral elements in three dimensions. We apply the standard procedure to construct a finite-dimensional subspace S_p^N of H_0^1 spanned by a piecewise p-degree polynomial finite element basis ϕ_i(x), i=1,2,⋯,N, developed for elliptic problems; see (Axelson and Barker 2006, Baker et al. 1981).

The semi-discrete Galerkin finite element problem consists of finding

$$u_N(x,t) = \sum_{i=1}^{N} c_i(t)\phi_i(x),$$

such that

$$\begin{cases}(v_N, (u_N)t(.,t)) + A(v_N, u_N(.,t)) = (v_N, f(.,t)), \forall\, v_N \in S_p^N, t \geq 0, \\ (v_N, u_N(.,0) = (v_N, g), \forall\, v_N \in S_p^N,\end{cases}$$

Where,

(v,u(.,t))=∫Ωv(x)u(x,t)dx, A(v,u(.,t))=$\int \Omega\ [\nabla v(x)]^t \nabla u(x,t)dx$.

Testing against v_N=ϕ_j , j=1,2,⋯,N yields a system of differential equations of the form (4).

Again, an implicit time-stepping method should be used to integrate the ordinary differential system in time.

Chapter 4

Galerkin FEM Solution For 2D BVP Using Triangular Elements

4.1 Governing Equation:

$\frac{\partial}{\partial x}(\propto_x \frac{\delta u}{\delta x}) + \frac{\partial}{\partial y}(\propto_y \frac{\delta u}{\delta y})+\beta u = g$…….. (1)

and the domain is defined as,

$$x \in [0, a]$$

$$y \in [0, b]$$

Consider,

$$r_e = \frac{\partial}{\partial x}\left(\propto_x \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\propto_y \frac{\partial u}{\partial y}\right) + \beta u - g$$

The weighted residual integration along the given domain

$$\iint_{\Omega^e} \omega r_e dxdy = 0$$

=>$\iint_{\Omega^e} \omega \left[\frac{\partial}{\partial x}\left(\propto_x \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\propto_y \frac{\partial u}{\partial y}\right) + \beta u - g\right] dxdy = 0$………(2)

Now,

$$\frac{\partial}{\partial x}\left(\omega\alpha_x \frac{\delta u}{\delta x}\right) = \frac{\partial\omega}{\partial x}\left(\alpha_x \frac{\delta u}{\delta x}\right) + \omega\frac{\partial}{\partial x}\left(\alpha_x \frac{\delta u}{\delta x}\right)$$

$$=> \omega\frac{\partial}{\partial x}\left(\alpha_x \frac{\delta u}{\delta x}\right) = \frac{\partial}{\partial x}\left(\omega\alpha_x \frac{\delta u}{\delta x}\right) - \frac{\partial\omega}{\partial x}\left(\alpha_x \frac{\delta u}{\delta x}\right)$$

$$=>\omega\frac{\partial}{\partial x}\left(\alpha_x \frac{\delta u}{\delta x}\right) = \frac{\partial}{\partial x}\left(\omega\alpha_x \frac{\delta u}{\delta x}\right) - \alpha_x \frac{\delta\omega}{\delta x}\frac{\delta u}{\delta x} \text{……… (3)}$$

Similarly,

$$=> \omega\frac{\partial}{\partial y}\left(\alpha_y \frac{\delta u}{\delta y}\right) = \frac{\partial}{\partial y}\left(\omega\alpha_y \frac{\delta u}{\delta y}\right) - \alpha_y \frac{\delta\omega}{\delta y}\frac{\delta u}{\delta y} \quad \text{……….. (4)}$$

Using equation (3) and (4) in equation (2), we have,

$$\iint_{\Omega^e} \omega \left[\frac{\partial}{\partial x}\left(\omega \propto_x \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\omega \propto_y \frac{\partial u}{\partial y}\right)\right] dxdy - \iint_{\Omega^e} \omega[\alpha_x \frac{\delta\omega}{\delta x}\frac{\delta u}{\delta x} + \alpha_y \frac{\delta\omega}{\delta y}\frac{\delta u}{\delta y}]\, dxdy + \iint_{\Omega^e} \omega\beta u dxdy = \iint_{\Omega^e} \omega g dxdy \quad \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Again , using the Green's theorem, the first part of equation (5) can be rewritten as,

$$\iint_{\Omega^e} \omega \left[\frac{\partial}{\partial x}\left(\omega \propto_x \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\omega \propto_y \frac{\partial u}{\partial y}\right)\right] dxdy = \oint_{r^e} \omega(\alpha_x \frac{\partial u}{\partial x} n_x + \alpha_y \frac{\partial u}{\partial y} n_y)\ dl \quad \ldots\ldots\ldots\ldots (6)$$

$$[\hat{a}_n = \hat{a}_x n_x + \hat{a}_y n_y]$$

Based on equation (6) the equation (5) can be rewritten as ,

$$-\iint_{\Omega^e} \omega[\alpha_x \frac{\partial\omega}{\partial x}\frac{\partial u}{\partial x} + \alpha_y \frac{\partial\omega}{\partial y}\frac{\partial u}{\partial y}]\ \mathrm{dxdy} + \iint_{\Omega^e} \omega\beta u dxdy = \iint_{\Omega^e} \omega g dxdy - \oint_{r^e} \omega\ (\ \alpha_x \frac{\partial u}{\partial x} n_x + \alpha_y \frac{\partial u}{\partial y} n_y)\ \mathrm{dl} \quad \ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Now ,$u = \sum_{j=1}^{n} u_j^e \emptyset_j$. Let $\omega = \emptyset_i$ for $i = 1,2, \ldots \ldots, n,$ since, according to Galerkin's approach the weight function should belongs to the same shape function that is used to interpolate u.

From equation (7)

$$-\iint_{\Omega^e}[\ \alpha_x \frac{\partial \emptyset_i}{\partial x}\frac{\partial}{\partial x}(\sum_{j=1}^{n} u_j^e \emptyset_j) + \alpha_y \frac{\partial \emptyset_i}{\partial y}\frac{\partial}{\partial y}(\sum_{j=1}^{n} u_j^e \emptyset_j)]\ dxdy + \iint_{\Omega^e} \emptyset_i\, \beta\ (\sum_{j=1}^{n} u_j^e \emptyset_j) dxdy = \iint_{\Omega^e} \emptyset_i\, g dxdy - \oint_{r^e} \emptyset_i\ (\ \alpha_x \frac{\partial u}{\partial x} n_x + \alpha_y \frac{\partial u}{\partial y} n_y)\ \mathrm{dl}$$

For $i = 1,2, \ldots \ldots, n,$

$$-\iint_{\Omega^e}[\ \alpha_x \frac{\partial \emptyset_i}{\partial x}(\sum_{j=1}^{n} u_j^e \frac{\partial \emptyset_j}{\partial x}) + \alpha_y \frac{\partial \emptyset_i}{\partial y}(\sum_{j=1}^{n} u_j^e \frac{\partial \emptyset_j}{\partial y})]\ dxdy + \iint_{\Omega^e} \emptyset_i\, \beta\ (\sum_{j=1}^{n} u_j^e \emptyset_j)\ \mathrm{dxdy} = \iint_{\Omega^e} \emptyset_i\, g dxdy - \oint_{r^e} \emptyset_i\ (\ \alpha_x \frac{\partial u}{\partial x} n_x + \alpha_y \frac{\partial u}{\partial y} n_y)\mathrm{dl} \quad \ldots\ldots\ldots\ldots\ldots (8)$$

In matrix form ,

$$\begin{bmatrix} A_{11}^{e} & A_{12}^{e} & \cdots & A_{1n}^{e} \\ A_{21}^{e} & A_{22}^{e} & \cdots & A_{2n}^{e} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}^{e} & A_{n2}^{e} & \cdots & A_{nn}^{e} \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \\ \vdots \\ u_n^e \end{bmatrix} + \begin{bmatrix} H_{11}^{e} & H_{12}^{e} & \cdots & H_{1n}^{e} \\ H_{21}^{e} & H_{22}^{e} & \cdots & H_{2n}^{e} \\ \vdots & \vdots & \ddots & \vdots \\ H_{n1}^{e} & H_{n2}^{e} & \cdots & H_{nn}^{e} \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \\ \vdots \\ u_n^e \end{bmatrix} = \begin{bmatrix} b_1^e \\ b_2^e \\ \vdots \\ b_n^e \end{bmatrix} + \begin{bmatrix} d_1^e \\ d_2^e \\ \vdots \\ d_n^e \end{bmatrix} \ldots\ldots (9)$$

Here,

$$A_{ij}^{e} = - \iint_{\Omega^e} [\, \alpha_x \frac{\partial \emptyset_i}{\partial x}\frac{\partial \emptyset_j}{\partial x} + \alpha_y \frac{\partial \emptyset_i}{\partial y}\frac{\partial \emptyset_j}{\partial y}]\, dxdy \;\ldots\ldots\ldots\; (10)$$

$$H_{ij}^{e} = \int\int_{\Omega^e} \beta\phi_i\phi_j dxdy \qquad \ldots\ldots\ldots\ldots\; (11)$$

$$b_i^e = \int\int_{\Omega^e} \phi_i\, gdxdy \qquad \ldots\ldots\ldots\ldots..(12)$$

$$d_i^e = \oint_{r^e} \emptyset_i \,(\, \alpha_x \frac{\partial u}{\partial x} n_x + \alpha_y \frac{\partial u}{\partial y} n_y) \,\mathrm{dl} \quad \ldots\ldots\ldots (13)$$

Let us write the equation (9) in more compact form,

$$\begin{bmatrix} C_{11}^{e} & C_{12}^{e} & \cdots & C_{1n}^{e} \\ C_{21}^{e} & C_{22}^{e} & \cdots & C_{2n}^{e} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1}^{e} & C_{n2}^{e} & \cdots & C_{nn}^{e} \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \\ \vdots \\ u_n^e \end{bmatrix} = \begin{bmatrix} f_1^e \\ f_2^e \\ \vdots \\ f_n^e \end{bmatrix} \ldots\ldots\ldots\ldots (14)$$

Here ,

$C_{ij}^{e} = A_{ij}^{e} + H_{ij}^{e}$
$f_i^e = b_i^e + d_i^e$

Notice that the line integral in equation (13) will be zero for all interior edges, as it produces equal terms but of opposite sign while evaluating in sharing edge of two adjacent triangular elements . It is non – zero at **Ω** domain boundary Γ, where Γ = $\Gamma_1 \cup \Gamma_2$.

Here,

Γ_1= Boundary edges where Dirichlet condition to be imposed. The contribution of equation (13) being zero will be discarded for this case .

Γ_2= Boundary edges which is characterized by mixed boundary condition or boundary condition of 3^{rd} kind or Robin boundary condition,

$$(\alpha_x \frac{\partial u}{\partial x} n_x \quad + \quad \alpha_y \frac{\partial u}{\partial y} n_y \quad) + \lambda u = \gamma$$

So, the line integral in equation (13) becomes,

d_i^e= - $\int_{\Gamma_2} \phi_i \; (\gamma - \lambda u$) dl ………… (15)

The equation (15) only exixts for boundary elements of the domain **Ω**, which satisfy Robin boundary conditions .
Now, let us evaluate different entries in equation (9) . First, we will try to simplify the entries of matrix A^e .

We know, for triangular element the shape function in natural ξ - η
Co-ordinates are,
$\phi_1 = 1 - \xi - \eta$
$\phi_2 = \xi$
$\phi_3 = \eta$

Now, co-ordinates inside any element can be presented as ,

$$x = x_1^e + x_{21}^e \xi + x_{31}^e \; \eta \qquad \text{………… (16a)}$$

$$y = y_1^e + y\xi + y_{31}^e \; \eta \text{………….. (16b)}$$

Where x_{ij}^e = x_i^e - x_j^e, Again,

$$\frac{\partial \phi}{\partial \xi} = \frac{\partial \phi}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial \phi}{\partial y}\frac{\partial y}{\partial \xi}$$

$$\frac{\partial \phi}{\partial \eta} = \frac{\partial \phi}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial \phi}{\partial y}\frac{\partial y}{\partial \eta}$$

In matrix form

$$\begin{bmatrix} \dfrac{\partial \phi}{\partial \xi} \\ \dfrac{\partial \phi}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \phi}{\partial x} \\ \dfrac{\partial \phi}{\partial y} \end{bmatrix} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (17)$$

where $\begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}$ = Jacobian Matrix J.

using equation set (16) and equation (17) , we have,

$$\begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_{21} & y_{21} \\ x_{31} & y_{31} \end{bmatrix} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (18)$$

Equation (17) and (18) gives us the idea of co-ordinate transformation ,

$$\begin{bmatrix} \dfrac{\partial \phi}{\partial x} \\ \dfrac{\partial \phi}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \dfrac{\partial \phi}{\partial \xi} \\ \dfrac{\partial \phi}{\partial \eta} \end{bmatrix} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (19)$$

Where,

$$J^{-1} = \frac{1}{|J|} \begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix}$$

And | J | = $x_{21}y_{31}$ - $x_{31}y_{21}$ = 2 × Area of an element = 2× A_e. Now from equation (19)

$$\begin{bmatrix} \dfrac{\partial \phi}{\partial x} \\ \dfrac{\partial \phi}{\partial y} \end{bmatrix} = \frac{1}{2A_e} \begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \phi_1}{\partial \xi} \\ \dfrac{\partial \phi_1}{\partial \eta} \end{bmatrix}$$

$$= \frac{1}{2A_e}\begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

as$\emptyset_1 = 1 - \xi - \eta$

$$= \frac{1}{2A_e}\begin{bmatrix} y_{21} & -y_{31} \\ x_{31} & -x_{21} \end{bmatrix}$$

$$= \frac{1}{2A_e}\begin{bmatrix} y_{23} \\ x_{32} \end{bmatrix} \qquad \text{…………………..} \quad (20)$$

Hence,

$$\frac{\partial \emptyset_1}{\partial x} = \frac{y_{23}}{2A_e} \qquad \text{……………}(21a)$$

$$\frac{\partial \emptyset_1}{\partial y} = \frac{x_{32}}{2A_e} \qquad \text{…………….}(21b)$$

Similarly,

$$\frac{\partial \emptyset_2}{\partial x} = \frac{y_{31}}{2A_e}\text{…………….}(22a)$$

$$\frac{\partial \emptyset_2}{\partial y} = \frac{x_{13}}{2A_e}\text{…………….}(22b)$$

$$\frac{\partial \emptyset_3}{\partial x} = \frac{y_{12}}{2A_e}\text{…………….}(22c)$$

$$\frac{\partial \emptyset_3}{\partial y} = \frac{x_{21}}{2A_e}\text{……………}(22d)$$

Now the Jacobi transformation can helps us to transform a double integral, in equation (10), from the regular co- ordinate to the natural co-ordinate.

$$\iint_{\Omega^e} f(x,y)dxdy = \int_0^1\int_0^{-1} f[x(\xi,\eta),y(\xi,\eta)]\,|\;J\;|\,d\xi d\eta$$

[Ref: H.Anton, "Calculus", 7th edition, New York : Willy, 2002, p- 1075 to 1090]

So,

$$A_{11}^e = -\int_0^1\int_0^{1-\eta}\left[\alpha_x \frac{y_{23}}{2A_e}\frac{y_{23}}{2A_e} + \alpha_y \frac{x_{32}}{2A_e}\frac{x_{32}}{2A_e}\right]2A_e d\xi d\eta$$

$$= -\left[\alpha_x \frac{(y_{23})^2}{2A_e} + \alpha_y \frac{(x_{32})^2}{2A_e}\right] \int_0^1 \int_0^{1-\eta} d\xi d\eta$$

$$= -\left[\alpha_x \frac{(y_{23})^2}{4A_e} + \alpha_y \frac{(x_{32})^2}{4A_e}\right] \qquad \ldots\ldots\ldots\ldots\ldots.. (23)$$

Similarly, $A_{12}^e = A_{21}^e = -\left[\alpha_x \frac{y_{23}y_{31}}{4A_e} + \alpha_y \frac{x_{32}x_{13}}{4A_e}\right]$ ……….. (24)

And,

$$A_{13}^e = A_{31}^e = -\left[\alpha_x \frac{y_{23}y_{12}}{1A_e} + \alpha_y \frac{x_{32}x_{21}}{4A_e}\right] \ldots\ldots\ldots\ldots\ldots(25)$$

$$A_{22}^e = -\left[\alpha_x \frac{(y_{31})^2}{4A_e} + \alpha_y \frac{(x_{13})^2}{4A_e}\right] \ldots\ldots\ldots\ldots.. \quad (26)$$

$$A_{23}^e = A_{32}^e = -\left[\alpha_x \frac{y_{31}y_{12}}{1A_e} + \alpha_y \frac{x_{13}x_{21}}{4A_e}\right] \ldots\ldots\ldots\ldots\ldots\ldots (27)$$

$$A_{33}^e = -\left[\alpha_x \frac{(y_{12})^2}{4A_e} + \alpha_y \frac{(x_{21})^2}{4A_e}\right] \qquad \ldots\ldots\ldots\ldots.. \qquad (28)$$

It is noticeable that the matrix A^e is symmetric ,i.e., $A_{ij}^e = A_{ji}^e$
Again from equation (11)

$$H_{ij}^e = \int\int_{\Omega^e} \beta\phi_i\phi_j \, dxdy$$

Using Jacobian transformation,

$$H_{ij}^e = \int_0^1 \int_0^{1-\eta} \beta\phi_i\phi_j |J| d\xi d\eta$$

$$= \beta 2A_e \int_0^1 \int_0^{1-\eta} \phi_i\phi_j d\xi d\eta$$

Now, let us determine different entries of H^e matrix,

$$H_{11}^e = \beta 2A_e \int_0^1 \int_0^{1-\eta} (\phi_1)^2 d\xi d\eta$$

$$= \beta 2A_e \int_0^1 \int_0^{1-\eta} (1 - \xi - \eta)^2 d\xi d\eta$$

$$=\frac{\beta A_e}{6} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{ (29)}$$

Again , we know,

$$\int\int_{\Omega^e} (\phi_1)^k (\phi_2)^l (\phi_3)^m \, dxdy = \frac{k!l!m!}{(k+l+m+2)!} 2A_e$$

[Ref: The finite element method- R.L. Taylor, 4th edition, Vol. 1, Basic foundation of linear problems, New York: McGraw Hill, 1989]

Using this formula we can have the same result of equation (29),

$$H_{11}^e = \int\int_{\Omega^e} \beta(\phi_1)^2 \, dxdy$$
$$= \beta \frac{2!0!0!}{(2+0+0+2)!} 2A_e$$
$$= \frac{\beta A_e}{6}$$

Similarly the other entries of H^e,

$$H_{12}^e = H_{21}^e = H_{13}^e = H_{31}^e = H_{23}^e = H_{32}^e = \frac{\beta A_e}{12}$$

And,

$$H_{22}^e = H_{33}^e = \frac{\beta A_e}{6}$$

Again from equation (12)

$$b_i^e = \int\int_{\Omega^e} \phi_i \, g dxdy$$

Using the same formula,

$$b_1^e = g\int\int_{\Omega^e} \phi_1 \, dxdy = g\frac{1!0!0!}{(1+0+0+2)} 2A_e = \frac{gA_e}{3} = b_2^e = b_3^e$$

Now in equation (15) ,γ and λ are constant,

$$d_i^e = -\int_{\Gamma_2} \phi_i (\gamma - \lambda u) dl$$

Considering for a given Robin boundary condition,

$$d_i^e = -\int_{L_\Omega^e} \phi_i (\gamma - \lambda u) dl$$

L_Ω^e= Boundary edge that satisfy mixed boundary condition and coincides with Γ_2 .

$$=> d_i^e = -\int_{L_\Omega^e} \phi_i \gamma dl + \int_{L_\Omega^e} \phi_i \lambda \left(u_1^e + u_2^e \phi_2 + u_3^e \phi_3 \right) dl$$

Now, let us consider the case for two nodes 1 and 2, both are at the edge as shown in Fig. 1,

$$d_i^e = - \int_{1 \to 2} \phi_i \gamma dl + \int_{1 \to 2} \phi_i \lambda \left(u_1^e \phi_1 + u_2^e \phi_2 + u_3^e \phi_3 \right) dl \quad \ldots\ldots\ldots\ldots \quad (30)$$

At any point along edge 1→ 2 in x-y co-ordinate Fig. 1(a).

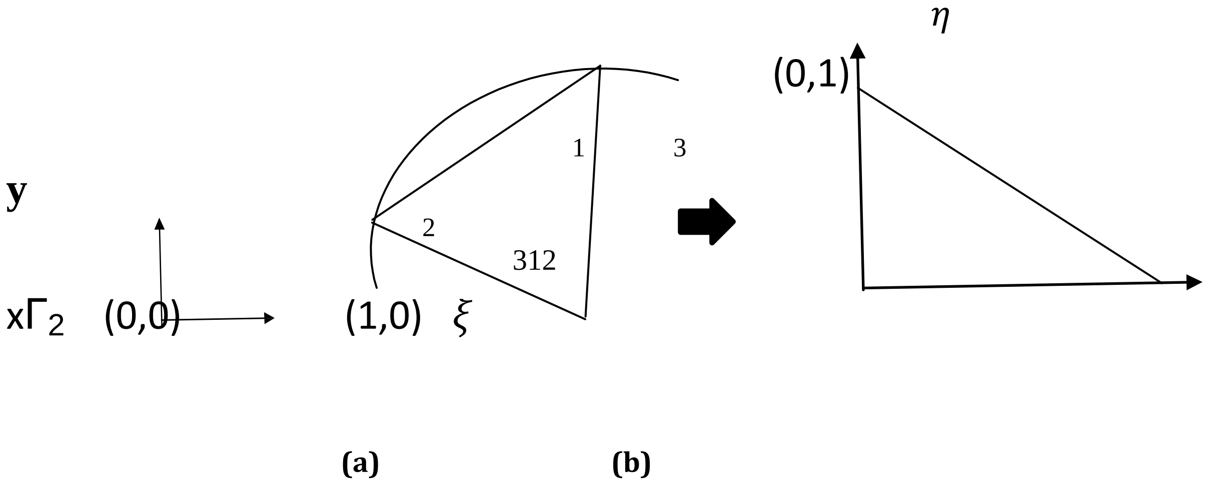


Figure 1: (a) Triangular element in x-y co-ordinate, (b) Triangular element in $\xi - \eta$ co-ordinate

$$x = x_1^e \phi_1(\xi, 0) + x_2^e \phi_2(\xi, 0)$$

$$= x_1^e (1 - \xi - 0) + x_2^e \xi$$

$$= x_1^e + x_{21} \xi$$

Similarly,

$$y = y_1^e + y_{21} \xi$$

Now, $$dl = \sqrt{(dx)^2 + (dy)^2}$$

$$= \sqrt{(x_{21})^2 + (y_{21})^2 d\xi} = l_{12} d\xi \quad \text{………………..} \ (31)$$

Using equation (31) in equation (30) to operate the integral in local node 1 of the triangle,

$$d_1^e = -\int_0^1 \phi_1(\xi,0)\gamma l_{12} d\xi + \int_0^1 \phi_1(\xi,0)\lambda\left[u_1^e \phi_1(\xi,0) + u_2^e \phi_2(\xi,0) + u_3^e \phi_3(\xi,0)\right] l_{12} d\xi$$

$$= -\int_0^1 (1-\xi-0)\gamma l_{12} d\xi + \int_0^1 (1-\xi-0)\lambda\left[u_1^e (1-\xi-0) + u_2^e \xi + u_3^e \times 0\right] l_{12} d\xi$$

$$= -\frac{\gamma l_{12}}{2} + \left[\frac{\lambda l_{12}}{3} u_1^e + \frac{\lambda l_{12}}{6} u_2^e\right] \quad \text{………………….} \ (32)$$

If we observe the effect of equation (32) in equation (14), we can see easily,

$$C_{11}^e = C_{11}^e - \frac{\lambda l_{12}}{3} \text{…………} \ (33a)$$

$$C_{12}^e = C_{12}^e - \frac{\lambda l_{12}}{6} \text{……………}(33b)$$

$$C_{13}^e = C_{13}^e - 0 \text{……………. } (33c)$$

In similar way, the entries for the local node 2,

$$d_2^e = -\int_0^1 \phi_2(\xi,0)\gamma l_{12} d\xi + \int_0^1 \phi_2(\xi,0)\lambda\left[u_1^e \phi_1(\xi,0) + u_2^e \phi_2(\xi,0) + u_3^e \phi_3(\xi,0)\right] l_{12} d\xi$$

$$= -\frac{\gamma l_{12}}{2} + \left[\frac{\gamma l_{12}}{6} u_1^e + \frac{\gamma l_{12}}{3} u_2^e\right]$$

The change in equation (14),

$$C_{21}^e = C_{21}^e - \frac{\lambda l_{12}}{6} \text{…………} \ (34a)$$

$$C_{22}^{e} = C_{22}^{e} - \frac{\lambda l_{12}}{3} \ldots\ldots\ldots\ldots \text{(34b)}$$

$$C_{23}^{e} \quad = C_{23}^{e} - 0 \ldots\ldots\ldots\ldots\ldots\ldots \text{(34c)}$$

Now for local node 3,

$$d_3^e = -\int_0^1 \phi_3(\xi,0)\gamma l_{12} d\xi + \int_0^1 \phi_3(\xi,0)\lambda\left[u_1^e\phi_1(\xi,0) + u_2^e\phi_2(\xi,0) + u_3^e\phi_3(\xi,0)\right] l_{12} d\xi$$

But it is a fact that $\phi_3(\xi,0)$= 0 along ξ – axis, as shown in Fig.1. So, for a boundary elemnent show in Fig.1, the vector d^e in equation (9) will be,

$$d^e = -\frac{\gamma l_{12}}{2}\begin{bmatrix}1\\1\\0\end{bmatrix} \ldots\ldots\ldots\ldots\ldots\ldots \text{(35)}$$

So far we have developed the necessary equations for the different entries in matrix representing a specific triangular element and work with $\phi_j(\xi,\eta)$ in $\xi - \eta$ co-ordinate system. Thus after determining u in term of natural co-ordinates we have to return in co-ordinate system x-y to express u as,$u(x,y) = \sum_{i=1}^{n} u_j^e \phi_j(x,y)$

Now we will see how to re-transform to the x-y co-ordinate from the natural co-ordinate system.As we see in equation (16), the x-y co-ordinates of a point inside a triangular element is given by,

$$x = x_1^e + x_{21}^e + x_{31}^e$$

$$y = y_1^e + y_{21}^e + y_{31}^e$$

Re-writing the above equation pair in matrix form,

$$\begin{bmatrix}x & -x_1^e\\ y & -y_1^e\end{bmatrix} = \begin{bmatrix}x_{21} & x_{31}\\ y_{21} & y_{31}\end{bmatrix}\begin{bmatrix}\xi\\ \eta\end{bmatrix}$$

$$=> \begin{bmatrix}\xi\\ \eta\end{bmatrix} = \frac{1}{x_{21}y_{31} - x_{31}y_{21}}\begin{bmatrix}y_{31} & -x_{31}\\ -y_{21} & x_{21}\end{bmatrix}\begin{bmatrix}x & -x_1^e\\ y & -y_1^e\end{bmatrix} \quad \ldots\ldots\ldots\ldots\text{(36)}$$

Just substituting the relations in equation (36) in shape function $\phi_j(\xi,\eta)$ we will get $\phi_j(x,y)$ in x-y co-ordinate system

Implementation:

Now we will apply the Galerkin approach on the same problem discussed in the chapter of " PDE 2D BVP". Let us recall the problem.

Problem :

Solve the Boundary value problem for the following square domain, where the domain is characterized by an Elliptic Laplace equation with domain supports $x \in [0,1]$and $y \in [0,1]$.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \ldots\ldots\ldots\ldots\ldots\ldots (37)$$

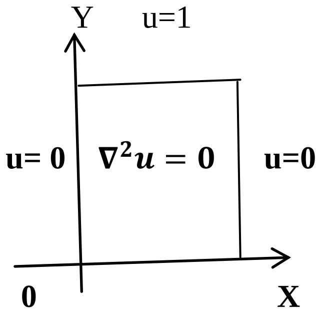


Fig : 2D BVP with Dirichlet boundary conditions

Analytical solution :

The analytical solution of the above problem is ;

$$u(x,y) = \frac{4}{\pi}\sum_{m=0}^{\infty} \frac{\sin(2m+1)\pi x}{2m+1} \frac{\sin\ h(2m+1)\pi y}{\sinh(2m+1)\pi} \ldots\ldots\ldots\ldots\ldots\ldots(38)$$

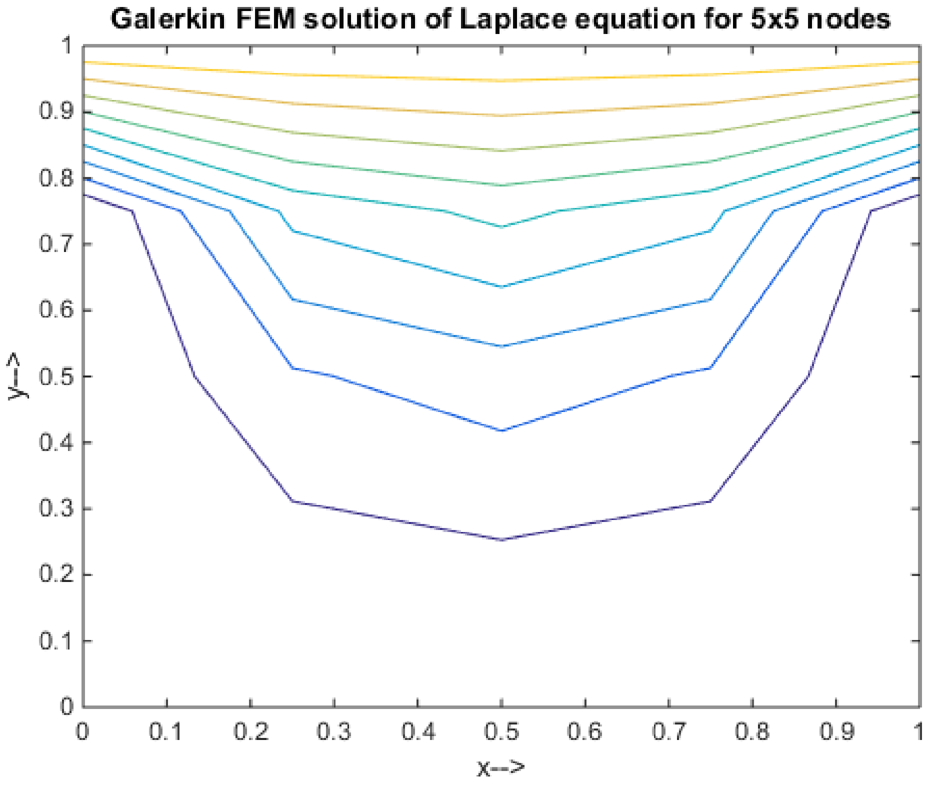
Numerical solution:

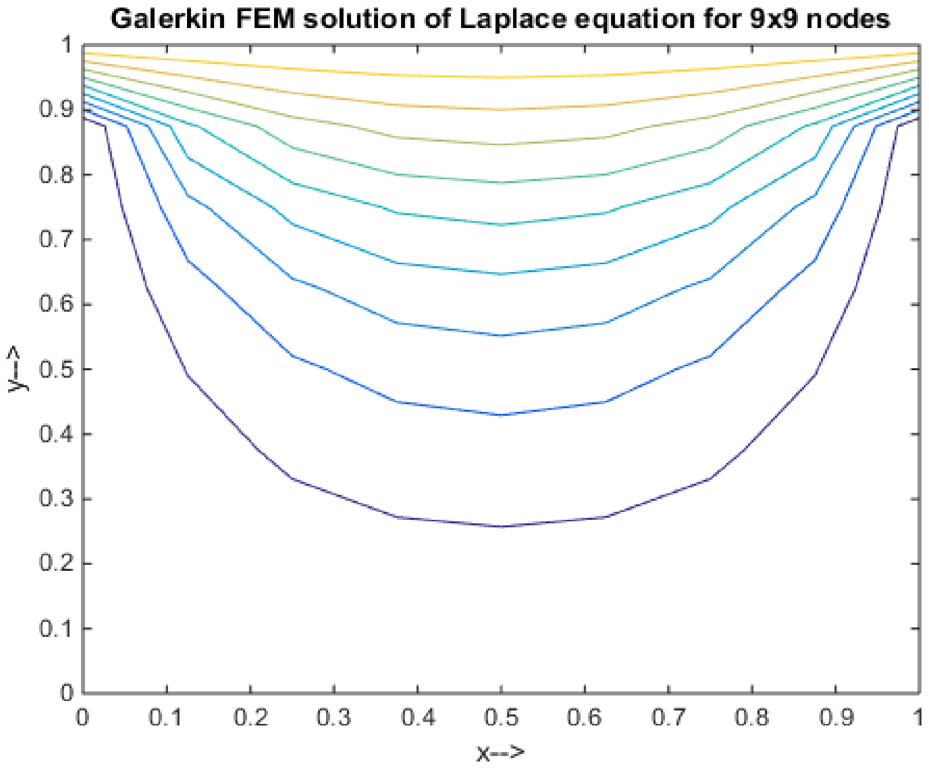
In equation (1) if we consider $\alpha_x = \alpha_y = 1$ and $\beta = g = 0$**,** then we will have equation (37). Now using equation (23) to (29) and with the help of Matlab we can have the following figures. Here we have used the triangular element in 2D FEM Galerkin approach for different domain discretization's.

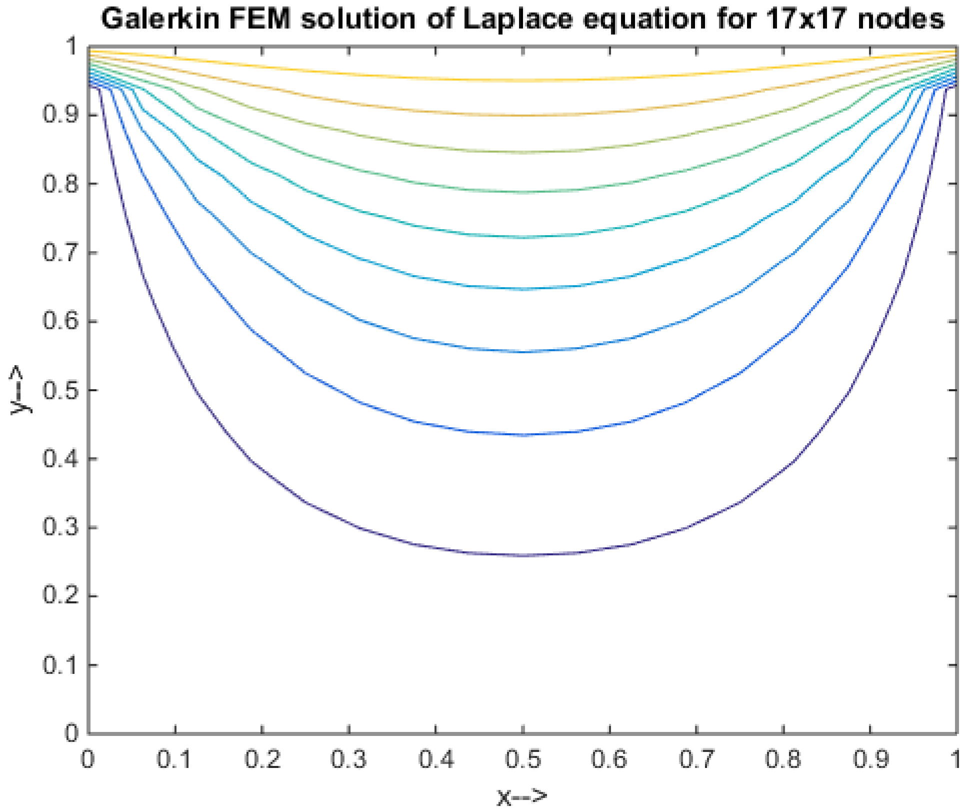
Chapter 5

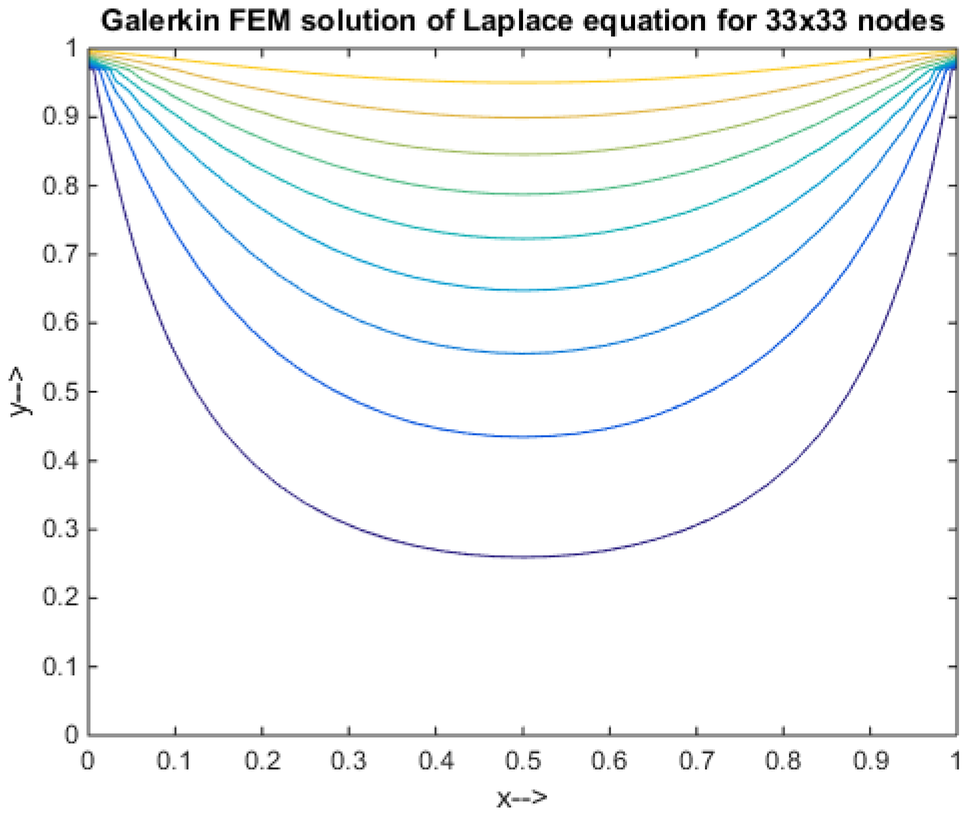
Result and Simulation

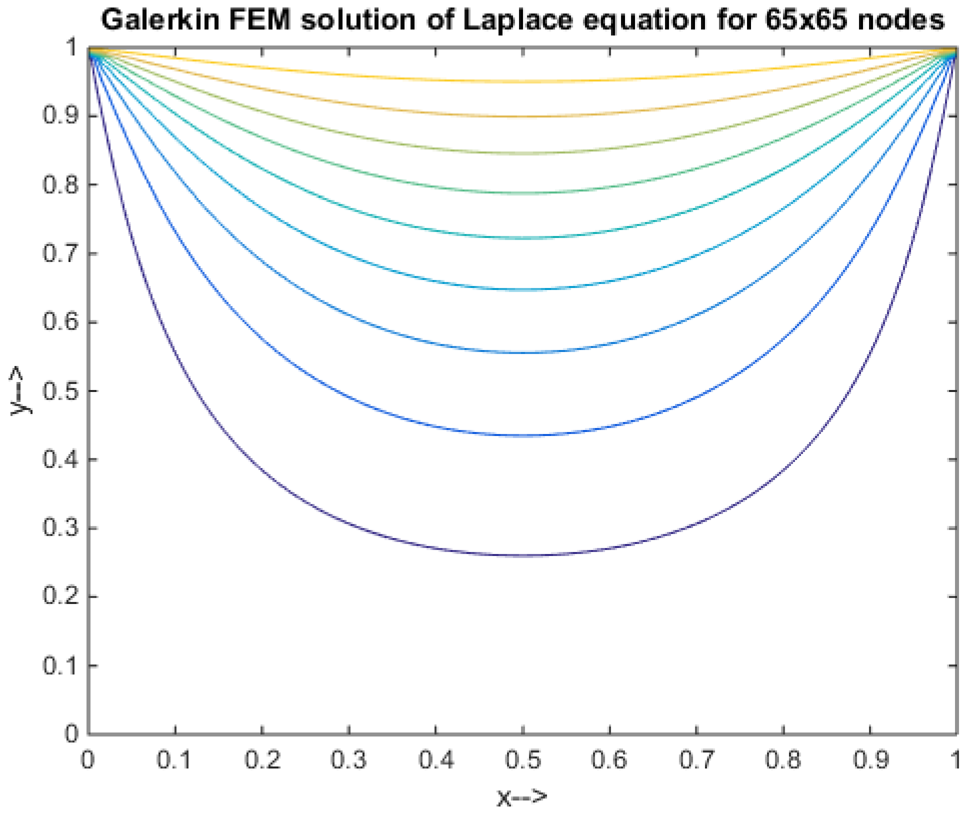
5.1 Figure


Galerkin FEM solution of Laplace equation for 5x5 nodes
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x-->
y-->


Galerkin FEM solution of Laplace equation for 9x9 nodes
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
y-->
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x-->


Galerkin FEM solution of Laplace equation for 17x17 nodes
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
y-->
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x-->


Galerkin FEM solution of Laplace equation for 33x33 nodes
y-->
x-->
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Galerkin FEM solution of Laplace equation for 65x65 nodes
y-->
x-->
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Chapter 6

Conclusion

We hope this thesis has covered the answer to the most important questions regarding what is the finite element method and also covered the answer about the Galerkin Method. We outline the many continuum fields and subjects in which FEM can be applied and showed how to solve the 2D boundary value problem by using finite element method.

At present days, engineers use computers and software in the design and manufacture of most products, processes and systems. Finite element method is one of the most important tools for an engineer or designer for design and analysis of products and processes.

In chapter 1, we have introduced the basic of Numerical analysis. You also knew the details of numerical methods and its types. We also gave the basic concepts of elements and nodes in the introduction part. From the introduction part you known about the history of finite element method.

In chapter 2, we discussed the details of finite element method. From this chapter you knew about the common FEM application, types of finite element, principle of finite element method, advantages of finite element method etc. Here, we also discussed about a general procedure for finite element analysis. You also knew the stiffness matrix and shape function from this chapter. So, in this chapter we gave you a proper concept about finite element method.

In chapter 3, boundary value problem and Galerkin Method were discussed in details. From this chapter, you knew about the boundary conditions, initial value problem etc. In Galerkin Method part, you knew about a one dimensional parabolic problem, weak Galerkin formulation, semi discretization and multidimensional problems. So, you had a proper concept of boundary value problem and Galerkin Method from this chapter.

In chapter 4, there was a Galerkin finite element solution. You knew about the governing equation and how its work from this chapter. We showed here the Galerkin FEM solution for 2D boundary value problem using triangular elements with the help of Matlab code. Using those codes we presented a numerical solution, of a space bounded by four arm where higher potential is applied to one arm and rests are grounded.

In chapter 5, we presented the result, of the 2D boundary value problem with figures.

So, in our thesis, we showed how to implement the finite element method and also showed the process of Galerkin Finite Method solution for 2D boundary value problem. Now you can solve and kind of 2D boundary value problem by following Galerkin Finite element method solution process.

Appendix

Matlab code

```
%%Galerkin FEM for 2D BVP Numerical Solution using triangular Elements
clc
clearall
clf
m=3;
N=2^m; %Number of Segment in each direction (even)
K=zeros((N+1)^2,(N+1)^2);
b=zeros((N+1)^2,1);
x_L=0;
x_R=1;  %Support in x direction
y_D=0;
y_U=1;  %support in y direction
h=(x_R-x_L)/N;  %step size or segment size
x=zeros(N+1,N+1);  %x co-ordinates
y=zeros(N+1,N+1);  %y co-ordinates
n=zeros(N+1,N+1);   %nodes number in serial sequence
P=1;  %increment factor
for i=1:(N+1)
for j=1:(N+1)
x(i,j)=h*(i-1);
y(i,j)=h*(j-1);
n(i,j)=P;
        P=P+1;
end
end
%Area of a single element, |J|=2*A_e
A_e=(1/2)*(((x(2,1)-x(1,1))*(y(2,2)-y(1,1)))-((y(2,1)-y(1,1))*(x(2,2)-x(1,1))));
%Elements co-ordinates and number
x_e=zeros(1,3);
y_e=zeros(1,3);
n_e=zeros(1,3);
%selecting element and assembling matrix
```

```
for i=1:N
for j=1:N
x_e(1,1)=x(i,j);
y_e(1,1)=y(i,j);
n_e(1,1)=n(i,j);
x_e(1,2)=x(i+1,j);
y_e(1,2)=y(i+1,j);
n_e(1,2)=n(i+1,j);
x_e(1,3)=x(i+1,j+1);
y_e(1,3)=y(i+1,j+1);
n_e(1,3)=n(i+1,j+1);

    [C,B] = matrix_assemble_G_2D(x_e,y_e,n_e,A_e);

    K(n_e(1,1),n_e(1,1))=K(n_e(1,1),n_e(1,1))+C(1,1);
    K(n_e(1,1),n_e(1,2))=K(n_e(1,1),n_e(1,2))+C(1,2);
    K(n_e(1,1),n_e(1,3))=K(n_e(1,1),n_e(1,3))+C(1,3);

    K(n_e(1,2),n_e(1,1))=K(n_e(1,2),n_e(1,1))+C(2,1);
    K(n_e(1,2),n_e(1,2))=K(n_e(1,2),n_e(1,2))+C(2,2);
    K(n_e(1,2),n_e(1,3))=K(n_e(1,2),n_e(1,3))+C(2,3);

    K(n_e(1,3),n_e(1,1))=K(n_e(1,3),n_e(1,1))+C(3,1);
    K(n_e(1,3),n_e(1,2))=K(n_e(1,3),n_e(1,2))+C(3,2);
    K(n_e(1,3),n_e(1,3))=K(n_e(1,3),n_e(1,3))+C(3,3);

    b(n_e(1,1),1)=b(n_e(1,1),1)+B(1,1);
    b(n_e(1,2),1)=b(n_e(1,2),1)+B(2,1);
    b(n_e(1,3),1)=b(n_e(1,3),1)+B(3,1);

x_e(1,1)=x(i+1,j+1);
y_e(1,1)=y(i+1,j+1);
n_e(1,1)=n(i+1,j+1);
x_e(1,2)=x(i,j+1);
y_e(1,2)=y(i,j+1);
n_e(1,2)=n(i,j+1);
x_e(1,3)=x(i,j);
y_e(1,3)=y(i,j);
n_e(1,3)=n(i,j);

    [C,B]=matrix_assemble_G_2D(x_e,y_e,n_e,A_e);

    K(n_e(1,1),n_e(1,1))=K(n_e(1,1),n_e(1,1))+C(1,1);
    K(n_e(1,1),n_e(1,2))=K(n_e(1,1),n_e(1,2))+C(1,2);
```

```
        K(n_e(1,1),n_e(1,3))=K(n_e(1,1),n_e(1,3))+C(1,3);

        K(n_e(1,2),n_e(1,1))=K(n_e(1,2),n_e(1,1))+C(2,1);
        K(n_e(1,2),n_e(1,2))=K(n_e(1,2),n_e(1,2))+C(2,2);
        K(n_e(1,2),n_e(1,3))=K(n_e(1,2),n_e(1,3))+C(2,3);

        K(n_e(1,3),n_e(1,1))=K(n_e(1,3),n_e(1,1))+C(3,1);
        K(n_e(1,3),n_e(1,2))=K(n_e(1,3),n_e(1,2))+C(3,2);
        K(n_e(1,3),n_e(1,3))=K(n_e(1,3),n_e(1,3))+C(3,3);

        b(n_e(1,1),1)=b(n_e(1,1),1)+B(1,1);
        b(n_e(1,2),1)=b(n_e(1,2),1)+B(2,1);
        b(n_e(1,3),1)=b(n_e(1,3),1)+B(3,1);
end
end

U=[zeros(1,N) 1];
for R=1:N
    U=[zeros(1,N) 1 U];
end
U=U';
b_subtract=K*U;
b=b-b_subtract;

%For (N+1)x(N+1) nodes
L=N+1;
K_1=[K(:,(L+2):(2*L-1))];
for i=2:(N-1)
    K_1=[K_1 K(:,(i*L+2):((i+1)*L-1))];
end

K_2=[K_1((L+2):(2*L-1),:)];
for i=2:(N-1)
    K_2=[K_2; K_1((i*L+2):((i+1)*L-1),:)];
end

b_1=[b((L+2):(2*L-1),1)];
for i=2:(N-1)
    b_1=[b_1; b((L+2):(2*L-1),1)];
end
u_vector=K_2\b_1;

%u_vector=K\b;
u_matrix=zeros(N+1,N+1);
```

```
M=2;
for Q=1:(N-1):((N-1)^2-(N-2))
u_matrix(M,2:N)=(u_vector(Q:(N-2+Q),1))';
   M=M+1;
end
%u_matrix(:,(N+1)) = ones((1:(N+1)),1);
u_matrix(:,(N+1)) = ones(N+1,1);
contour(x,y,u_matrix)
xlabel('x-->')
ylabel('y-->')
title('Galerkin FEM solution of Laplace equation for 65x65 nodes')
```

Function code

```
function [C,B] = matrix_assemble_G_2D(x_e,y_e,n_e,A_e)
Beta = 0;
g=0;

A=zeros(3,3);
H=zeros(3,3);
C=zeros(3,3);
B=zeros(3,1);

x1=x_e(1,1);
y1=y_e(1,1);
n1=n_e(1,1);
x2=x_e(1,2);
y2=y_e(1,2);
n2=n_e(1,2);
x3=x_e(1,3);
y3=y_e(1,3);
n3=n_e(1,3);

%Assembling matrix A
A(1,1)=-((y2-y3)^2+(x3-x2)^2)/(4*A_e);
A(2,2)=-((y3-y1)^2+(x1-x3)^2)/(4*A_e);
A(3,3)=-((y1-y2)^2+(x2-x1)^2)/(4*A_e);

A(1,2)=-((y2-y3)*(y3-y1)+(x3-x2)*(x1-x3))/(4*A_e);
A(2,1)=A(1,2);
```

```
A(1,3)=-((y2-y3)*(y1-y2)+(x3-x2)*(x2-x1))/(4*A_e);
A(3,1)=A(1,3);
A(2,3)=-((y3-y1)*(y1-y2)+(x1-x3)*(x2-x1))/(4*A_e);
A(3,2)=A(2,3);
%assembling matrix H and C
for i=1:3
for j=1:3
if i==j
H(i,j)=(Beta*A_e)/6;
else H(i,j)=(Beta*A_e)/12;
end
C(i,j)=A(i,j)+H(i,j);
end
end
%Assembling matrix B
B(1,1)=(g*A_e)/3;
B(2,1)=B(1,1);
B(3,1)=B(1,1);
```

References

1. etd.ohiolink.edu/!etd.send_file?accession=ohiou1182782321&disposition=inline
2. www.sciencedirect.com/science/article/pii/0377042796000179
3. www.sciencedirect.com/topics/engineering/finite-element-method
4. www.springer.com/gp/book/9781402091995
5. www.mathworks.com/discovery/finite-element-analysis.html
6. www.particleincell.com/2012/matlab-fem/
7. www.math.iitb.ac.in/~siva/ma41707/ode6.pdf?fbclid=IwAR13kymkATEF0rx7pRbPOIKmny3yVbsqz10n_yFsGUwLIBJnwPt1LYj8Fms
8. homepage.divms.uiowa.edu/~atkinson/m171.dir/sec_6-11.pdf?fbclid=IwAR3-FrLP8xI4v-iwWdupRfd9bWC8vJ0-SgI07350kj3afhX9mME7YLLc6Tc
9. www.stewartcalculus.com/data/CALCULUS%20Concepts%20and%20Contexts/upfiles/3c3-2ndOrderLinearEqns_Stu.pdf?fbclid=IwAR1lwrQqbajcyIDXPVwtFrWgHlF46WLXSOr7q5dXQSoajaoM0zfrlZQ_AOY
10. ieeexplore.ieee.org/abstract/document/1073669/metrics?fbclid=IwAR2IIfaq6f-Nz6ObNRLBNTBFTR10_bEcKzHekZdxIfbgrJL6LT9u4eINPWU#metrics
11. en.wikipedia.org/wiki/Finite_element_method
12. iopscience.iop.org/article/10.1088/1757-899X/231/1/012180/pdf?fbclid=IwAR1j3SCcmz8HsxLfoBrCp7S8t3-Ty181hhIzbyfnazk8xLNCXwfMJL6DxwU
13. en.wikipedia.org/wiki/Boundary_element_method
14. Shokrieh M M, Rezaei D. Analysis and optimization of a composite leaf spring [J]. Composite Structures, 2003, 60(3):317-325
15. Xiang lin. Finite element analysis about front-axle vehicle of steady and dynamic performances [D]. Nanning: Guangxi University, 2008.
16. link.springer.com/chapter/10.1007/978-3-642-33287-6_4
17. matematicas.unex.es/~coco/Modelizacion/FEMLarson_Bengzon.pdf

18. www.mathworks.com/matlabcentral/fileexchange/27826-fast-fem-assembly-nodal-elements
19. www.mathworks.com/help/pde/ug/assemblefematrices.html
20. www.researchgate.net/publication/220261613_Matlab_Implementation_of_the_Finite_Element_Method_in_Elasticity
21. en.wikipedia.org/wiki/Galerkin_method
22. mountainscholar.org/bitstream/handle/10217/189261/Wang_colostate_0053N_14624.pdf?sequence=1
23. D. Arnold, D. Boffi, and R. Falk. Quadrilateral H(div) finite elements. SIAM J. Numer. Anal., 42:2429–2451, 2005.
24. P. Bastian and B. Riviere. Superconvergence and h(div) projection for discontinuous galerkin methods. Int. J. Numer. Meth. Fluids, 42:1043–1057, 2003.
25. G. Lin, J. Liu, L. Mu, and X. Ye. Weak galerkin finite element methdos for darcy flow: Anistropy and heterogeneity. J. Comput. Phys., 276:422–437, 2014.
26. M. Wheeler and I. Yotov. A multipoint flux mixed finite element method. SIAM J. Numer. Anal., 44:2082–2106, 2006.
27. T. Wihler and B. Riviere. Discontinuous Galerkin methods for second-order elliptic pde with low-regularity solutions. J. Sci. Comput., 46:151–165, 2011.
28. M. Ainsworth and J. T. Oden (1997), A posteriori error estimation in finite element analysis, Comput. Meth. Appl. Mech. Engrg., 142, pp. 1–88.
29. www.scholarpedia.org/article/Boundary_value_problem
30. www.math.iitb.ac.in/~siva/ma41707/ode6.pdf
31. www.mathworks.com/help/matlab/boundary-value-problems.html