93: ANALYSIS MODEL FOR STATIONARY HEAT FLOW, GROUNDWATER SEEPAGE, POTENTIAL FLOW, PLANE STRESS AND PLANE STRAIN PROBLEMS

DA SULON
19286422

DEPARTMENT OF CIVIL ENGINEERING

OCTOBER 2018

STUDY LEADER: DR. G.C. VAN ROOYEN

FAKULTEIT INGENIEURSWESE
FACULTY OF ENGINEERING

UNIVERSITEIT
STELLENBOSCH
UNIVERSITY

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / Plagiarism *Declaration*

1 Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2 Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
I agree that plagiarism is a punishable offence because it constitutes theft.

3 Ek verstaan ook dat direkte vertalings plagiaat is.
I also understand that direct translations are plagiarism.

4 Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.

5 Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *Student number*	**Handtekening / *Signature***
Voorletters en van / *Initials and surname*	**Datum / *Date***

Abstract

Stellenbosch University's Department of Civil Engineering offers a course in the finite element method to its students at undergraduate level with the aim of providing the engineering industry with graduates well equipped with skills in the finite element method. In the course, students are introduced to the application of the finite element method for the boundary value problem of stationary heat flow. In this study, an analysis model is developed which aims to assist in the development of students' understanding of the finite element method by linking the theory taught in the course to various types of physical problems. The problems investigated include groundwater seepage, potential flow and force-displacement problems. The force-displacement problem is introduced using the theory of elastic plane stress and plane strain. The study provides the data structure of the model including a description of how the analysis model is used given examples for each type of problem. The findings of this report include various improvements which can be made to the analysis model and topics of future investigation. This includes, the investigation of other types of boundary value problems and the development of a more contemporary approach to the graphical user interface.

Contents

List of Figures

List of Tables

1 Chapter 1: Introduction

1.1 Background to Study

In current engineering practice, finite element method procedures form an important part in the analysis and design stages of engineering projects (Bathe, 2014). The method can provide solutions to many physical problems such as force and displacement (structural analysis), heat transfer and mass transport, fluid flow in porous media and electrostatics to name a few (Logan, 2016). Engineers face the increasing demand to model complex systems using methods such as the finite element method. An advantage for engineers in this regard has been the persistent advancement in computer aided technologies. Currently, the development of user-friendly graphical interfaces has significantly reduced the difficulties associated with the application of the finite element method commercially. This has led to an increase in the need for graduate engineers to be familiar and competent in the finite element method. Throughout tertiary engineering institutions, the finite element method has been the subject of research at postgraduate level with very few students ever having opportunity and exposure to it at undergraduate level.

Stellenbosch University's Department of Civil Engineering offers a module to its undergraduate students known as Engineering Informatics 314 in which students are introduced to the finite element method. The aim of the module is to develop students' basic understanding of the finite element method as a powerful, general algorithm for the solution of a broad class of physical problems. The course introduces the finite element method using a simple physical problem, namely heat transfer in the stationary state, commonly called stationary heat flow.

As part of the educational tools of the module, students are provided with a Java program with functionality to execute the algorithm of the finite element method. The application was specifically developed with implementation limited to stationary heat flow problems. It was postulated that the application is somewhat under achieving its potential as an educational tool and may pose a constraint in the development and education of students in the module.

Delving deeper into the technicalities of the finite element method, it is clear that the work can be separated into 3 components. Firstly, the mathematical theory of the particular problem is of interest. For Engineering Informatics 314, this is the theory of the stationary heat flow problem. For students, the application of finite element method for stationary heat flow problems is rather abstract and may seem unimportant in the context of civil engineering. However, the theory is provided and structured in a way that can be easily linked to other physical problems of importance to civil engineers such as groundwater seepage, potential flow and structural analysis.

The second component of the finite element method is the development of an algorithm to solve the theoretical component first mentioned. The algorithm forms a key part of Engineering Informatics 314. While the module only focuses on the theory associated with the stationary heat flow problem, the algorithm to which students are exposed is much more general and can be used for a variety of physical problems. This fact raises significant educational difficulties with the application provided in Engineering Informatics 314 because the application does not show any real links between the algorithm and physical problems other than stationary heat flow.

The third component is computer implementation of the theory and algorithm of the finite element method. Implementation for Engineering Informatics 314 is developed using object modelling in Java and comprises the material taught in the module's predecessor. This implementation is provided to students in the form of a programming project, i.e. source code. An issue has been noted that the application contains no instructions, user manual or examples to aid students with its use. The issue is further amplified by the fact that the graphical user interface is incomplete and confusing. Moreover, the application provides no documentation or explanations of the classes and methods for implementing the finite element algorithm. The lack of educational resources provided with the application raises concerns over its effectiveness as an educational tool.

It is already well known that programming of finite element techniques is a complicated process (Nikishkov, 2010). Consequently, to address the issues identified above, an analysis model has been proposed which aims to highlight the algorithm's versatility by presenting and linking various problems using the finite element method. The analysis model is intended to help bridge the gap of understanding between stationary heat flow problems and, for example, more advanced structural analysis problems.

1.2 Aim and Scope of Study

The broad aim of this study is to develop an analysis model which can assist students' educational development of the finite element method for later use in structural analysis, hydraulics, geotechnique and pavement materials. Given this aim, increased enthusiasm and overall performance of students in Engineering Informatics 314 may follow. It was postulated that the aim could be achieved by providing students with an easily accessible analysis model and resources which highlight the versatility of finite element method which they are exposed to in Engineering informatics 314. A limitation in the development of the analysis model is to ensure that the program is developed using skills, tools and knowledge which students of

Engineering Informatics 314 are already familiar. The development of the analysis model will take place through the use of object modelling in Java since students in the module have already been exposed to this modelling in the module's predecessor.

It is obvious at first that an analysis model can be made to be more usable and understandable by providing a user with more resources to help with deciphering the source code and assist with the use of the analysis model. This study investigates improving the usability of the analysis model through the inclusion of documentation and implementation examples. The documentation investigated involves the use of Java documentation for the classes and packages in the program, good practical examples using the finite element method in civil engineering, as well as comprehensive development and documentation data structure of the analysis model. All the above components of the documentation are then to be evaluated on the basis of whether they are possible within the limitations and time constraints of this study.

In order to achieve the above primary aim, some more specific aims are identified. Firstly, a full review of the current programming project and documentation provided to students is to be conducted. The process will explore and review the original programming project. The review of the programming project may include the extraction of useful resources in order to provide a robust framework on which to expand the new analysis model. Furthermore, the review may include the removal of unnecessary and confusing code in order to make the source code more understandable to students – this may also include the removal of unnecessary classes and methods. Lastly, the review will evaluate the applicability and effectiveness of class names and methods. In essence, the review of the existing programming project will provide the basis of a data structure for the proposed analysis model.

The next specific aim of this study involves investigating the development of an integrated graphical user interface which is easily understandable and flows in a similar way to currently used finite element software. The scope of the graphical user interface includes investigating a clear, understandable and robust way to select and perform an analysis when using the model. At first inspection, this should include means and functionality which prevent the user from making errors when performing an analysis and ways to provide some assistance to the user if the process of using the program becomes unclear at any stage.

The next point of investigation will be to develop a way for the user to perform multiple analyses when running the program. Moreover, the study will investigate ways to provide the user with output of the results. The output looks at developing graphical representation of the model in a clear, obvious-to-use and easily understandable way.

Following the development of a graphical user interface, an investigation of literature is to be conducted to evaluate various modelling problems where the finite element method is applied in a similar way to which students are exposed in Engineering Informatics 314. It should be noted that the existing programming project provided to students focuses on implementation of the finite element method for two dimensional domians, and so this shall be a limitation of the literature researched for this study. The problems selected to be investigated include; stationary heat flow, groundwater seepage and potential flow.

It is already known at this stage that the stationary heat flow problem requires the use of one degree of freedom per node, namely the temperature value. Therefore, another specific aim of this study is to evaluate whether there is opportunity to develop the analysis model to provide functionality to execute finite element problems with more degrees of freedom per node. For this purpose, an investigation will be conducted into the use of finite element analysis for elastic plane stress and plane strain problems.

Once sufficient literature has been surveyed to gain enough knowledge on the use of the finite element method for the selected physical problems, implementation of the problems into the program will be examined. Implementation may include using the existing program infrastructure (which was limited to stationary heat flow) and adapting it to suit other types of problems. This will include opportunities and limitations of the program's implementation.

The development of an analysis model is an iterative process. It can be expected that parts of the analysis model may have to be redeveloped if these parts limit the further progress of this study. This iterative process poses significant time constraints on this study. To combat these constraints, a study plan has been developed to ensure the efficient development in this study. The study plan used is illustrated in Figure 1.

The study plan consists an investigation phase which includes compiling the investigated literature and knowledge required to develop the analysis model. Following this is the iterative process of developing the analysis model. This stage combines implementation of the investigated components into the analysis model. The analysis model is then tested for errors and bugs. If necessary, reimplementation or modifications to the implementation takes place. The output of the iterative process of development forms the results for this study as illustrated in Figure 1.

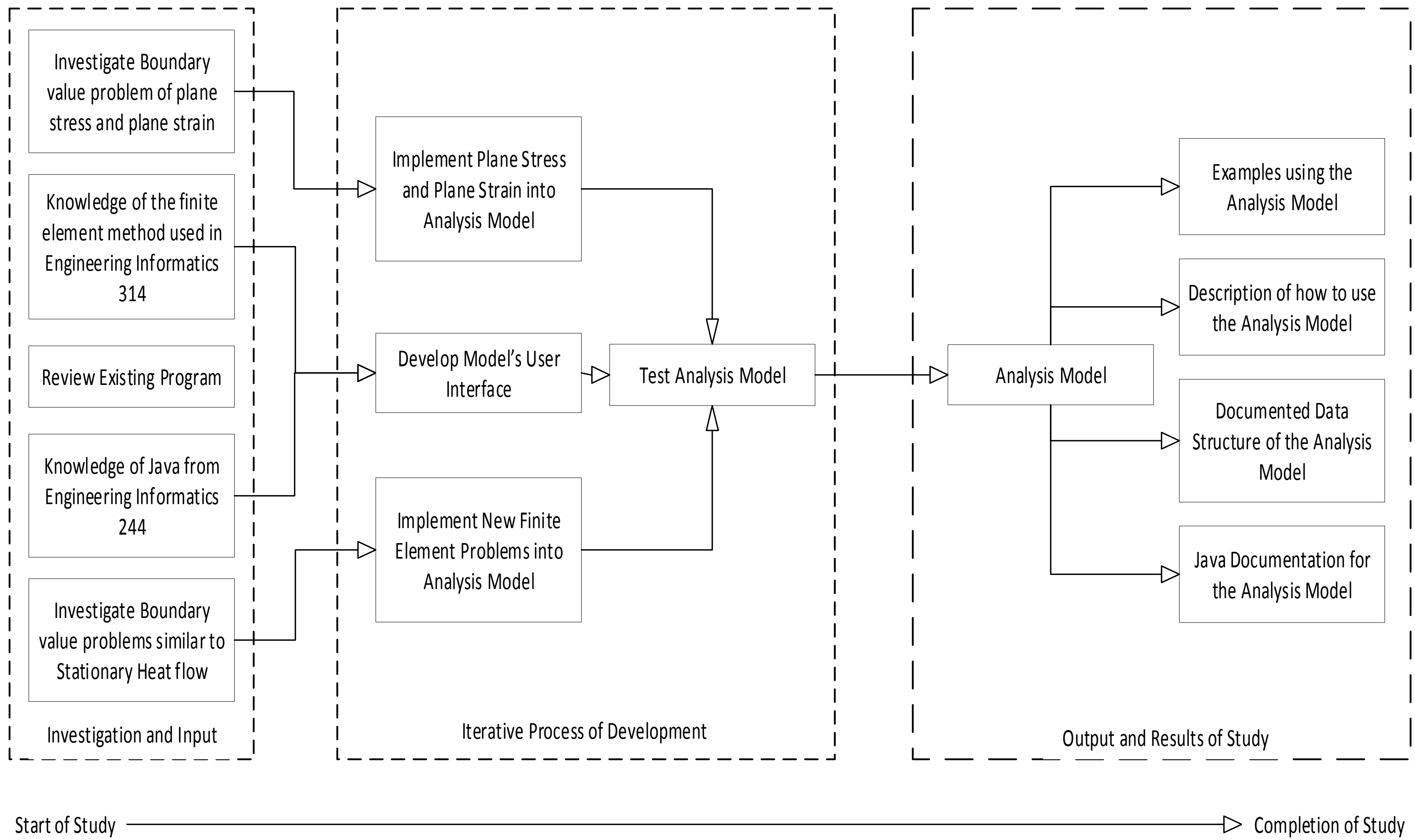


Figure 1: Study Plan

2 Chapter 2: Literature Review

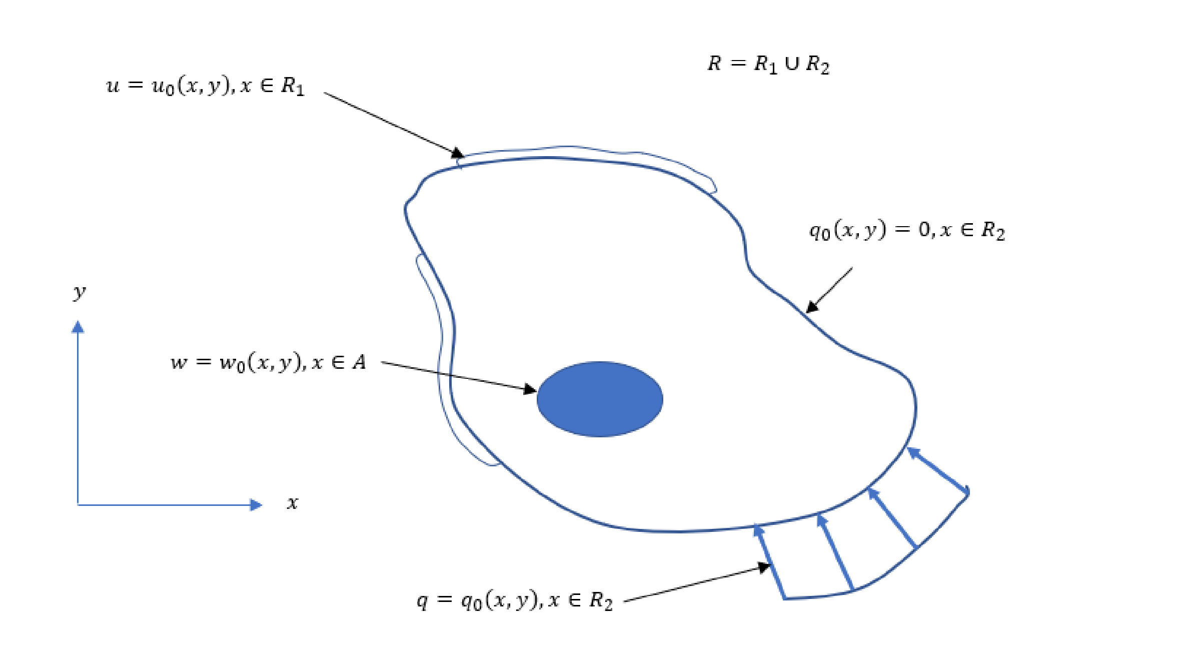
2.1 Introduction

The tasks required for this study are already well defined and do not require a significant quantity of literature. The initial focus of this chapter is to investigate the current literature provided to students of Engineering Informatics 314, more specifically the theory of stationary heat flow. Points addressed include the assumptions, geometry, material law and fundamental relationships to develop the boundary value problem of stationary heat flow. Following this, focus is shifted to how the aforementioned may be related to other modelling problems in civil engineering. In this step, literature is investigated to develop the boundary value problem in a similar manner for problems such as groundwater seepage and potential flow so that these problems can later be implemented into the proposed analysis model. Once investigation of various similar types of boundary value problems is completed, literature is explored on the boundary value problem for elastic plane stress and plane strain. These methods deviate slightly from the procedures of the previously mentioned problems. In this stage, the derivation of low order finite elements for elastic plane stress and plane strain is inspected for implementation into the proposed analysis model.

2.2 Engineering Informatics 314 Literature

2.2.1 Stationary Heat Flow Problem Definition

As described by Logan (2016) heat flow is inherently assumed to be conductive. This implies that when a body is exposed to some heat, there will be a transmittance of the heat through the body over some period of time (Bathe, 2014). The development of the problem of stationary heat flow according to van Rooyen (2017), is defined by first considering a body at rest. In this case, the heat flow inside the body is considered to be independent with time, that is, the condition of the body with respect to time is stationary or in other words the conditions within the body do not change over time. This is where the problem develops its title 'stationary' heat flow. Moreover, the problem definition accounts for defining heat flows within the body, i.e. if there is a source of heat within the body then it is possible to describe this in the problem. Furthermore, at all points on the body's surface, either the temperature or heat flux must be specified. The above definition by van Rooyen (2017) was verified with use of literature from Zienkiewicz and Taylor (2000), Bathe (2014) and Logan (2016). The stationary heat flow definition is shown graphically in Figure 2.



Where:

$q(x, y)$ is defined as the heat inflow per time and unit length on the boundary R

$w_0(x, y)$ is defined as heat inflow per time and unit area at a point inside the body A

$u_0(x, y)$ is defined as prescribed temperature at a point on the boundary R_1

$q_0(x, y)$ is defined as the prescribed heat inflow per time and unit length at a point on the boundary R_2

Figure 2: Stationary Heat Flow Problem Definition Diagram (van Rooyen, 2017)

2.2.2 The Strong Form of the Stationary Heat Flow Boundary Value Problem

In order to evaluate the stationary heat flow problem using the finite element method, it is essential to define some fundamental relationships between temperatures and heat flux (Bathe, 2014; Logan, 2016). Van Rooyen (2017) describes the fundamental relationships as the strong form of the boundary value problem. These relationships are as follows:

1. The *temperature gradient* within the body
2. A definition for *heat flux*
3. A way to relate heat flux on the boundary to the Cartesian coordinates of the body, described in the literature as the *heat state*
4. The *heat flux through an arbitrary plane*
5. The *heat balance equation*
6. The *material law* for heat flow

The definitions of the above listed are all provided in the literature of Engineering Informatics 314 and are not required to be expressed or examined further for this scope of this study. However, it is notable to this study that the meaning and importance of these relationships in establishing the finite element method for solving stationary heat flow problems is significant. In this study, it is essential to note that the assumptions used to develop the material law for the purposes of Engineering Informatics 314 assume that the material is homogeneous and isotropic. There is also the possibility for materials to be defined as heterogeneous and anisotropic. The derivation of the material law using heterogeneous and anisotropic assumptions can be found in the works of Zienkiewicz and Taylor (2000) as well as Bathe (2014).

2.2.3 Poisson Equation for Stationary Heat Flow

By combining the derived expressions for the heat balance equation and material law, the Poisson equation is derived (Zienkiewicz and Taylor, 2000a; Bathe, 2014; van Rooyen, 2017). In general, the Poisson equation can be used for a variety of physical problems as long as the problems are described in the same manner and form of differential equations as that of stationary heat flow (van Rooyen, 2017). The general form of the Poisson equation is given by equation 1.

$$k\nabla^2\varphi + Q = 0 \tag{1}$$

Where in the case of stationary heat flow:

k - refers to the material conductivity

∇ - refers to the gradient operator

φ - refers to the scalar temperature function

Q - refers to the internal heat generated or removed from the body

Van Rooyen (2017) defines solving the Poisson equation using the finite element method using the following four steps:

1. Establish the differential equations of the boundary value problem (strong form).
2. Set up the weighted residual integral form using the weighted residual method and perform partial integration of a particular term of the integral. This yields the integral form (weak form) of the problem.
3. Develop the concept of the finite element method.
4. Solve the problem using the algorithm of the finite element method.

While the literature for Engineering Informatics 314 describes each of the above process in great detail, there is no established link between the finite element method for stationary heat flow and other types of boundary value problems. To meet the aims of this study a similar procedure as the above mentioned will have to be founded and investigated for other types of boundary value problems.

2.2.4 Finite Element Method Equations

As described in the literature for Engineering Informatics 314, the boundary value problem for stationary heat flow can be solved through the discretization of the body in to finite elements. The process of discretization is known as the finite element method (Zienkiewicz and Taylor, 2000). In this process, vector {u}, {q} and {w} are derived. These vectors represent the nodal states for temperature, heat flux and internal sources of which both {q} and {w} have been converted to heat streams at the nodal points of the finite elements for the entire system. The finite element method for stationary heat flow can be expressed in the form of equation 2 (van Rooyen, 2017).

$$[K]\{u\} = \{q\} + \{w\} \tag{2}$$

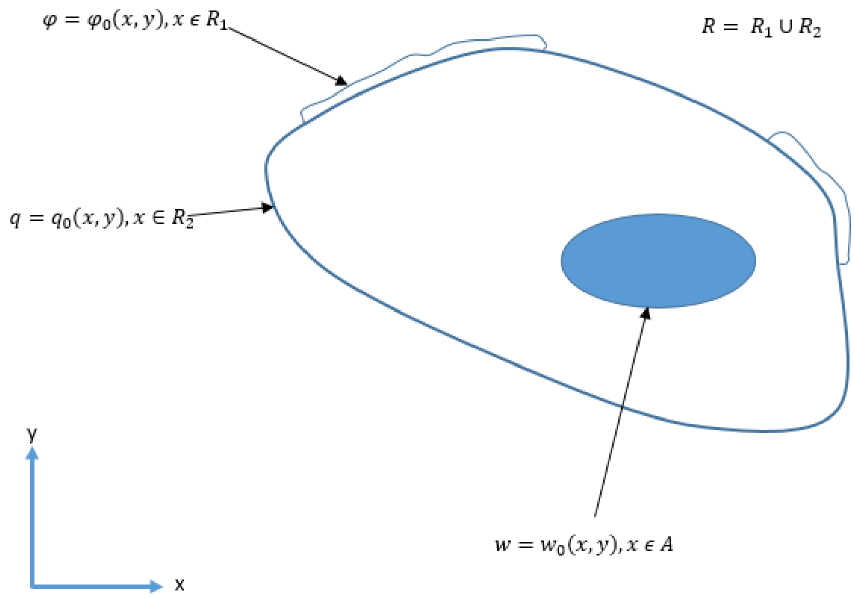
In equation 2, $[K]$ is determined as the union of all the element conductivity matrix contributions in the system coordinates, where an element conductivity matrix is calculated using equation 3. It is important to note that equation 3 is solved in Engineering Informatics 314 using Gaussian integration where necessary.

$$[k_e] = \int [S_x]c[S_x]^T \, dA \tag{3}$$

2.3 Groundwater Seepage

2.3.1 Problem Definition

Mariño and Luthin (1982) describe the process of groundwater seepage as soil physical problems that arise because of the physical flow of water through porous media. Similarly, the works of Harr (1963) highlight the importance of groundwater seepage in civil engineering. For this study, the equations describe a soil body at rest whose conditions with respect to time are constant. Furthermore, Dupuit theory for unconfined flow is considered (Harr, 1963; Wang and Anderson, 1982). Importantly, the Dupuit theory assumes that for small inclinations of the water table, the hydraulic gradient is equal to the water table (Mariño and Luthin, 1982). This means that it is possible to relate the flow of water to the total potential head at any given point in the body (Harr, 1963). Figure 3 shows a diagram of the layout of the groundwater seepage boundary value problem.



Where:

$q(x, y)$ is defined as the groundwater inflow per time and unit length on the boundary R
$w_0(x, y)$ is defined as groundwater inflow per time and unit area at a point inside the body A
$\varphi_0(x, y)$ is defined as prescribed total potential head at a point on the boundary R_1
$q_0(x, y)$ is defined as the prescribed groundwater inflow per time and unit length at a point on the boundary R_2

Figure 3: Groundwater Seepage Problem Definition Diagram

2.3.2 The Strong Form of the Boundary Value Problem

The strong form of the groundwater seepage problem begins by first deriving an expression for the gradient of the total potential head, {g}, at a point and is given by equation 4 (Wang and Anderson, 1982). This equation is required to describe the way in which the total potential head changes within the body. Note that this is analogous to the temperature gradient in stationary heat flow.

$$\{g\} = \nabla\varphi = \begin{Bmatrix} \dfrac{\partial\varphi}{\partial x} \\ \dfrac{\partial\varphi}{\partial y} \end{Bmatrix} \tag{4}$$

The next step according to Logan (2016) is to determine the conservation of mass for the system. Conservation of mass is derived on the assumption that the total flow within the system must be in balance (Harr, 1963). To derive the conservation of mass equation, it is necessary to determine an expression to relate velocities on the surface of the body to the velocities within the body (Wang and Anderson, 1982). This step is analogous to deriving an expression for heat state in the stationary heat flow problem. The velocity at a point in the direction of the Cartesian axes is given by equation 5.

$$\{v\} = \begin{Bmatrix} v_x \\ v_y \end{Bmatrix} \tag{5}$$

The conservation of mass for the body is derived by considering an infinitesimal element of the body (Harr, 1963). Figure 4 below, diagrammatically shows an infinitesimal element of the soil body in 2 dimensions.

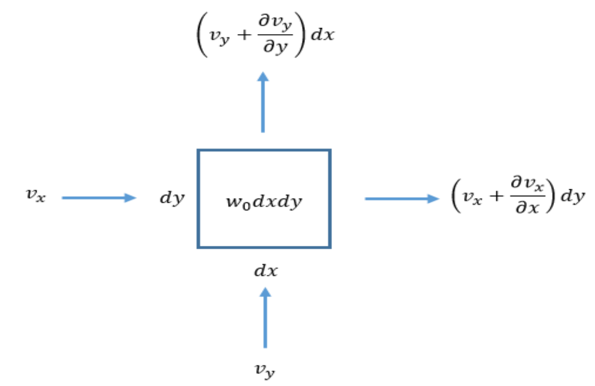


Figure 4: Infinitesimal element of soil

Considering flow of groundwater into the element and assuming that the conditions within the element with respect to time are stationary, equation 6 is developed (Harr, 1963; Wang and Anderson, 1982).

$$v_x dy + v_y dx + w_0 dxdy \tag{6}$$

Similarly, considering groundwater out of the body for stationary conditions equation 7 is established (Harr, 1963; Wang and Anderson, 1982).

$$\left(v_x + \frac{\partial v_x}{\partial x}\right) dy + \left(v_y + \frac{\partial v_y}{\partial y}\right) dx \tag{7}$$

Combining equations 6 and 7 yield equation 8.1 which results in equation 8.3 where v is the velocity function. Equation 8.3 can be likened to the heat balance equation in the stationary heat flow problem.

$$w_0 = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \tag{8.1}$$

$$w_0 = \begin{Bmatrix} \frac{\partial v_x}{\partial x} \\ \frac{\partial v_y}{\partial y} \end{Bmatrix} \tag{8.2}$$

$$w_0 = \nabla v \tag{8.3}$$

The definition for groundwater flux over an infinitesimal boundary is provided by equation 9 where the unit for q is in m/s and Q is the flow rate in m^2/s (Wang and Anderson, 1982).

$$q = \lim_{dR \to 0} \frac{dQ}{dR} \tag{9}$$

The groundwater flux through an arbitrary plane at a point is illustrated in Figure 5.

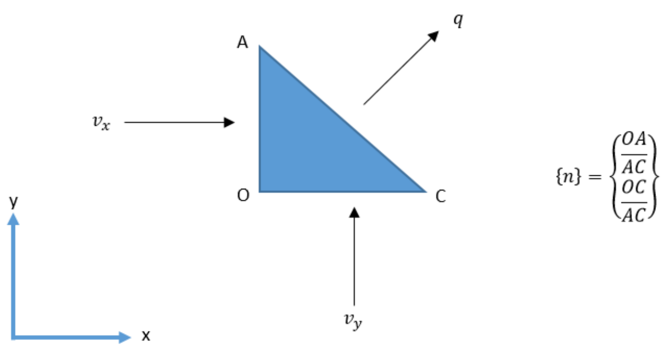


Figure 5: Groundwater flux through an arbitrary plane

From Figure 5, equation 10 is obtained. Note that equation 10 is analogous to heat flux through an arbitrary plane in stationary heat flow.

$$q = \{n\}^T\{v\} \tag{10}$$

Similar to stationary heat flow, the material law derived in this study for groundwater seepage is defined for a homogeneous and isotropic material. For literature on heterogeneous and anisotropic materials see Harr (1963), Mariño and Luthin (1982) as well as Logan (2016). The equation for material law is given by equation 11, where c is the material's permeability coefficient (Mariño and Luthin, 1982; Knappett and Craig, 2012). The derivation of material law is governed by Darcy's law (Harr, 1963; Mariño and Luthin, 1982; Knappett and Craig, 2012).

$$\{v\} = -c\{g\} \tag{11}$$

This concludes the necessary equation relationships required to establish the strong for of the boundary value problem for groundwater seepage considered in this study.

2.3.3 Poisson Equation for Groundwater Seepage

It can now be seen clearly that the strong form of the boundary value problem is formatted in the identical approach to that of stationary heat flow. Hence, it is now possible to derive the Poisson equation. To derive the Poisson equation, the conservation of mass expression and material law are combined (Wang and Anderson, 1982). The resultant Poisson equation is provided in equation 12.

$$c\left(\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial x^2}\right) + w_0 = 0 \tag{12}$$

It is important to note that the meanings of the expressions in the Poisson equation now differ to that of stationary heat flow. Table 1 below summarizes the meanings and units of the variables necessary for the Poisson equation above.

Table 1: Summary of Terms and Meanings for Groundwater Seepage

Variable	Meaning	Unit	Analogous to stationary heat flow
g	Potential head gradient at a point in the body	m/m	Temperature gradient
φ	Total potential head at a point on or in the body	m	Temperature
q	Groundwater inflow on the body	m/s	Heat flux
v	Velocity at a point within the body	m/s	Heat state
w_0	Internal groundwater source at a point in the body	m^3/s	Internal heat source
c	Material permeability coefficient	m/s	Material conductivity

2.3.4 Finite Element Method Equations

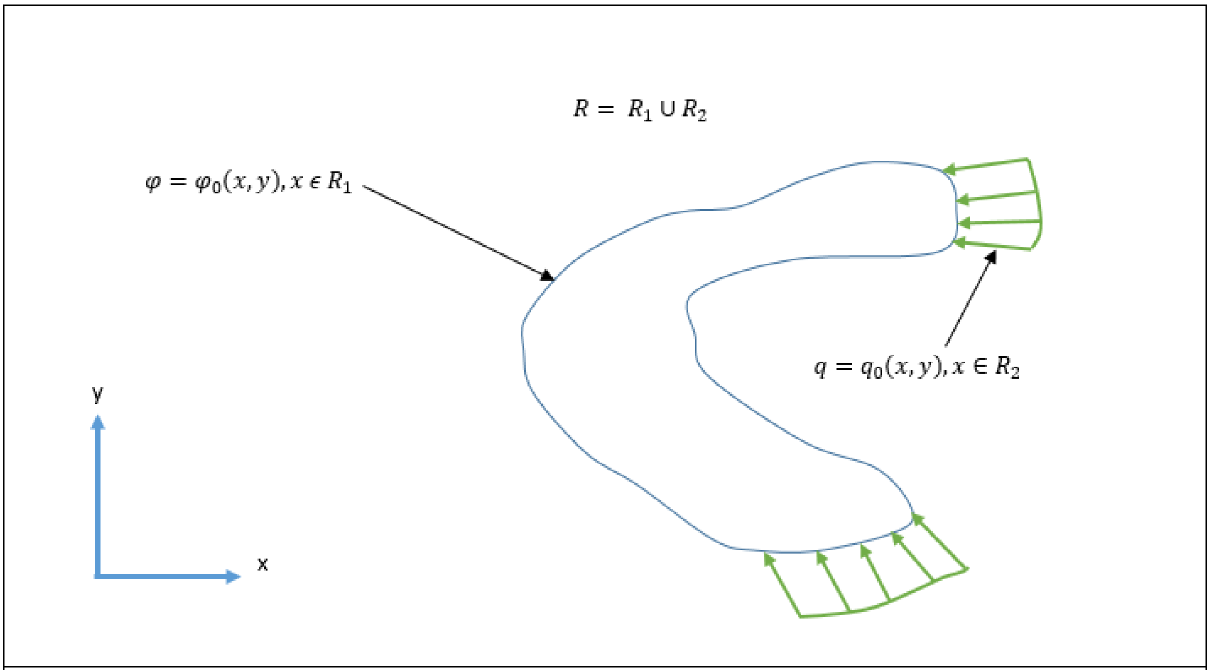
The finite element method solution for the system can be solved using the same equation as that which is used for stationary heat flow, namely equation 2 (Logan, 2016). However, it is important to note the changes in meanings of the variables within the equation.

2.4 Potential flow

2.4.1 Problem Definition

In literature by White (2010), the significance of potential flow in the context of fluid mechanics is emphasized. Importantly, White (2010) highlights that potential flows are not limited to the linear approximations and geometries of viscous solutions.

For the development of the boundary value problem for potential flow, it is assumed that the fluid is inviscid and irrotational, that is the viscous effects are neglected at very low speeds (Chadwick, 2013). These assumptions lead to the setup of the boundary value problem as depicted in Figure 6.



Where:

$q(x, y)$ is defined as the inflow of fluid per time and unit length on the boundary R
$w_0(x, y)$ is defined as inflow of fluid per time and unit area at a point inside the body A
$\varphi_0(x, y)$ is defined as prescribed velocity potential at a point on the boundary R_1
$q_0(x, y)$ is defined as the prescribed fluid inflow per time and unit length at a point on the boundary R_2

Figure 6: Potential Flow Problem Definition Diagram

2.4.2 The Strong Form of the Boundary Value Problem

The first step to developing the strong form of the boundary value problem is to define the velocity potential gradient described by equation 13 (White, 2010). This is necessary to describe the way in which the velocity potential changes within the body (Chadwick, 2013). This step is like determining the temperature gradient in stationary heat flow.

$$\{g\} = \nabla\varphi = \begin{Bmatrix} \frac{\partial\varphi}{\partial x} \\ \frac{\partial\varphi}{\partial y} \end{Bmatrix} \tag{13}$$

It is important to note the similarity of equations 13 to the equation for total potential head gradient for groundwater seepage (equation 4). Although these equations use the same notation, their meanings are different.

The next step is to define an expression for the velocity of flow at a point in terms of the Cartesian coordinates of the body of fluid. This is given by equation 14.

$$\{v\} = \begin{Bmatrix} v_x \\ v_y \end{Bmatrix} \tag{14}$$

Now consider the concept of continuity which is determined by using conservation of mass in the same way as groundwater seepage (Wang and Anderson, 1982). In this case however, an internal source of fluid is not considered (Chadwick, 2013). The result can be expressed as the equation 15.1 and 15.2 for incompressible flow in two dimensions.

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0 \tag{15.1}$$

$$\nabla v = 0 \tag{15.2}$$

Notably, if viscous effects are neglected, then there is no restriction of flow and so the equation to relate the velocities to the velocity potential at a point is given by equation 16 (White, 2010). This implies that there is no material law which governs the relationship between φ and v as there was for stationary heat flow.

$$\{v\} = \begin{Bmatrix} \frac{\partial\varphi}{\partial x} \\ \frac{\partial\varphi}{\partial y} \end{Bmatrix} \tag{16}$$

Following, the definition for flux over an infinitesimal boundary is provided by equation 17 where the unit for q is in m/s (Wang and Anderson, 1982).

$$q = \lim_{dR \to 0} \frac{dQ}{dR} \tag{17}$$

Note that from the above equation, the expression for flux on the boundary becomes analogous to groundwater seepage (Wang and Anderson, 1982).

$$q = \{n\}^T \{v\} \tag{18}$$

2.4.3 Poisson Equation for Potential Flow

The Poisson equation is quite simply derived by combining the expression for the velocity and the continuity equation. The Poisson equation is expressed by equation 19 (Chadwick, 2013).

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0 \tag{19}$$

It is important to note that now the Poisson equation contains no c term as before, this is because there is no restriction to flow as a result of viscous properties associated with the fluid (White, 2010). A summary of important variables and the meanings for potential flow problems is summarized in table 2.

Table 2: Summary of Variables and Important Meanings for Potential Flow Problems

Variable	Meaning	Unit	Analogous to stationary heat flow
g	Velocity potential gradient at a point in the body	m/s	Temperature gradient
φ	Velocity potential at a point on or in the body	m	Temperature
q	fluid inflow on the surface of the body	m/s	Heat flux
v	Velocity at a point within the body	m/s	Heat state

2.4.4 Finite Element Equations

The solution to the Poisson equation using the finite element method for the system can be solved using equation 20. Note the dissimilarity in the system equation of potential flow to that of stationary heat flow. These differences are due to derivations of internal sources of fluid flow which are not being considered for this study (White, 2010).

The finite element solution to the potential flow problem must contain at least 1 prescribed velocity potential at a node. This constraint forms part of the development of the boundary value problem for potential flow (White, 2010). It is obvious from equation 20 that if there is not at least 1 known velocity potential, then the problem is undefined.

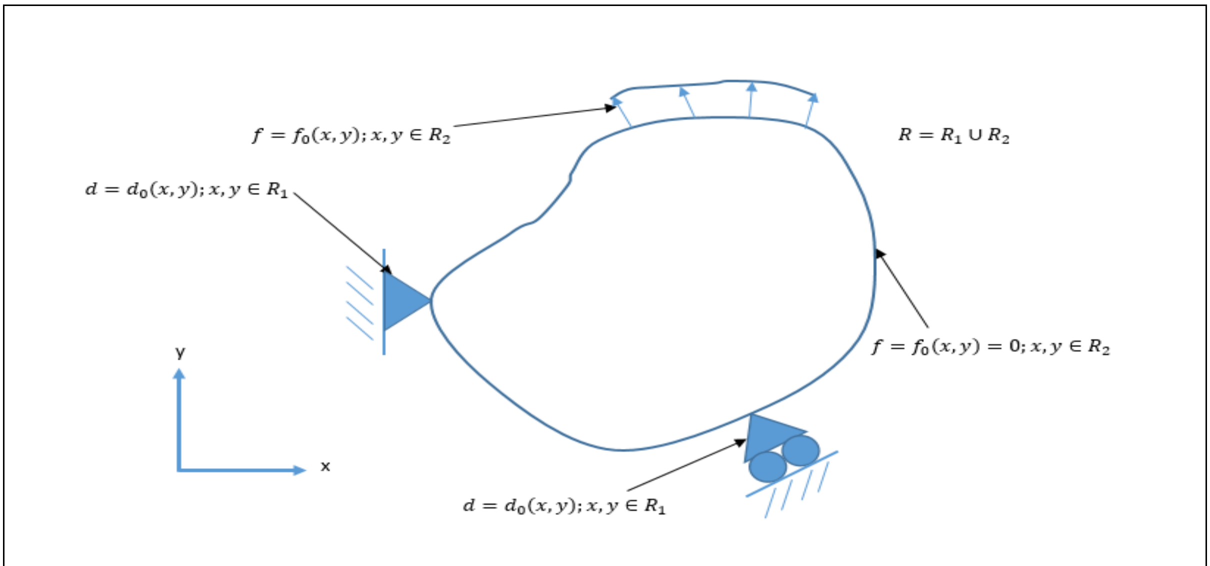
$$[K]\{\varphi\} = \{q\} \tag{20}$$

2.5 Plane Stress and Plane Strain

2.5.1 Problem Definition and Outline

This section considers the development of the boundary value problem for elastic plane stress and plane strain. Zienkiewicz and Taylor (2000) show that the application of the finite element method for solutions to elasticity boundary value problems is well known, and as such the derivation of the finite element method equations will not be discussed in detail in this study.

Hrennikoff (1968) notes that the equations which govern the formulation of the plane stress and plane strain boundary value problems are based on statics together with the elasticity and continuity constraints of the materials of which the body is considered. The boundary value problem considers planar elements with three or more nodes with displacements in the two dimensional plane (Logan, 2016). A graphical depiction of the typical two-dimensional elastic problem is provided in Figure 7.



Where:

$f(x,y)$ is defined as the force per unit length (or stress) on the boundary R

$d_0(x,y)$ is defined as the prescribed displacement at a point on the boundary R_1

$q_0(x,y)$ is defined as the prescribed force per unit length at a point on the boundary R_2

Figure 7: Two-Dimensional Elastic Planar Problem Definition

For the explanation in the study it is necessary to derive an expression to relate the displacements d to the Cartesian coordinate system as a vector. The expression is given by equation 21.

$$\{d\} = \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{21}$$

In order to solve the boundary value problem for plane stress of plane strain, it is necessary to derive a relationship for the change in displacement within the body (Logan, 2016). This is similar to the process of deriving the temperature gradient in stationary heat flow (Bathe, 2014). The concept of relating the change in displacement over the body is referred to the strain (Zienkiewicz and Taylor, 2000a; Bathe, 2014; Logan, 2016). In the state of plane stress and plane strain, the general conditions of normal and shear strains are considered to apply, that is, equations 22 can be derived for example (Zienkiewicz and Taylor, 2000a).

$$\varepsilon_x = \frac{\partial u}{\partial x} \qquad \varepsilon_y = \frac{\partial v}{\partial y} \qquad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{22}$$

Furthermore, it is necessary to describe the strains and stress at a point as a vector (Logan, 2016). This is given by equation 23.

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} \quad \{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{Bmatrix} \tag{23}$$

In three dimensions, the strains can be related to the stresses using equation 24 (Zienkiewicz and Taylor, 2000a). Equation 24 is governed by the material law for linear elastic behaviour and assumes the material to be homogeneous and isotropic (Bathe, 2014).

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{Bmatrix} = \begin{bmatrix} \hat{E}(1-v) & \hat{E}v & \hat{E}v & 0 & 0 & 0 \\ \hat{E}v & \hat{E}(1-v) & \hat{E}v & 0 & 0 & 0 \\ \hat{E}v & \hat{E}v & \hat{E}(1-v) & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} \tag{24}$$

$$\hat{E} = \frac{E}{(1+v)(1-2v)} \tag{25}$$

Logan (2016) shows that the relationship between the strains and stress for plane stress and plane strain cases can be expressed as equation 26.

$$\{\sigma\} = [D]\{\varepsilon\} \tag{26}$$

It is important to note that $[D]$ in equation 26 now represents a matrix in contrast to the material law which governed the previous boundary value problems. The occurrence of the material matrix $[D]$ is due to the strains being dependant on both x and y simultaneously. This contrasts with the stationary heat flow boundary value problem, where the temperature gradient in a direction is only dependant on either x or y.

For the plane stress case, it is assumed that σ_z, τ_{xz} and τ_{yz} are all zero (Logan, 2016). This leads to equation 27 from equation 24.

$$[D] = \frac{E}{1-v^2}\begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-v}{2} \end{bmatrix} \tag{27}$$

In contrast, plane strain assumes that ε_z, γ_{xz} and γ_{yz} are all zero (Logan, 2016). Thus equation 28 is developed from equation 24.

$$[D] = \frac{E}{(1+v)(1-2v)}\begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-2v}{2} \end{bmatrix} \tag{28}$$

Importantly, the plain strain [D] matrix can be represented in the same format as the plane stress matrix by adjusting the E and v terms. The adjusted terms can be calculated using equations 29.a and 29.b.

$$E = \frac{1}{1-v^2} \tag{29.a}$$

$$v = \frac{v}{1-v} \tag{29.b}$$

2.5.2 Finite Element System Equation

The general form of the finite element equation is given by equation 30.

$$\{F\} = [K]\{D\} \tag{30}$$

Here, $\{F\}$ is the system force vector resulted from the stresses on the boundary being converted to a nodal applied forces over the system (Bathe, 2014). The $\{D\}$ vector is the system displacement vector. The problem is solved by first solving the unknown forces at the prescribed displacements, followed by determining the unknown displacements of the system (Nikishkov, 2010).

2.5.3 Development of the Constant Stain Triangle (CST) Element

The derivation of the constant strain triangle is well documented in Logan (2016). However, for the purposes of this study it is only important to determine how the expressions to solve problems using CST elements are calculated. A diagram showing the layout of a CST element is provided in Figure 8.

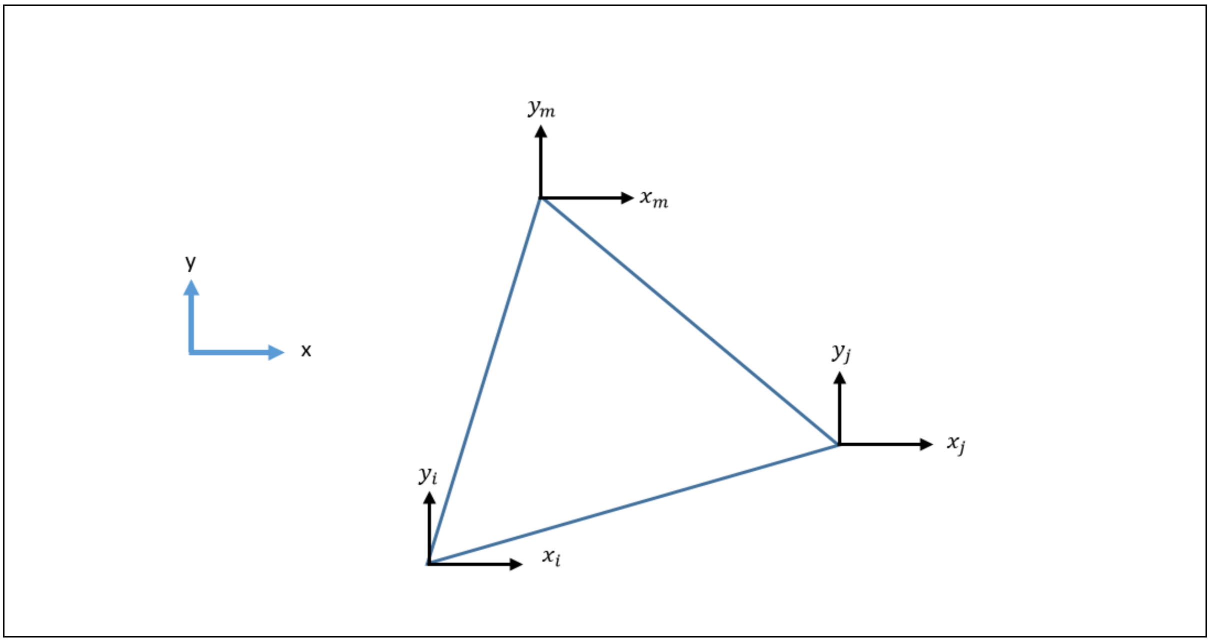


Figure 8: CST Element

The following expressions were extracted from Logan (2016) and verified through the works of Zienkiewicz and Taylor (2000) as well as Bathe (2014).

The expression which relates the strains to the displacements of the element is given by equation 31 where $\{d\}$ is the displacement vector for the CST element.

$$\{\varepsilon\} = [B]\{d\} \tag{31}$$

The expression for determining the stiffness matrix of a CST element is given by equation 32. It is evident that equation 32 is similar to the equation 3. However, the transpose now occurs on first [B] matrix in the expression. This occurrence is simply due to the use of different algebraic notation during the derivation of [B].

$$[k_e] = \int [B]^T [D][B] dA \tag{32}$$

In equation 31 and 32, [B] is expressed by equation set 33 and [D] is dependent on whether the problem is assumed to be plane stress or plane strain.

$$[B] = \begin{bmatrix} \beta_i & 0 & \beta_j & 0 & \beta_m & 0 \\ 0 & \gamma_i & 0 & \gamma_j & 0 & \gamma_m \\ \gamma_i & \beta_i & \gamma_j & \beta_j & \gamma_m & \beta_m \end{bmatrix} \qquad \begin{matrix} \beta_i = y_j - y_m & \gamma_i = x_m - x_j \\ \beta_j = y_m - y_i & \gamma_j = x_i - x_m \\ \beta_m = y_i - y_j & \gamma_m = x_j - x_i \end{matrix} \tag{33}$$

2.5.4 Isoparametric Formulation of the Plane Quadrilateral Element

The isoparametric formulation is a useful method to simplify computer implementation of plane stress and plane strain elements (Nikishkov, 2010). These methods however require the use of Gaussian integration (Logan, 2016). A graphical representation of an isoparametric quadrilateral element is shown in Figure 9.

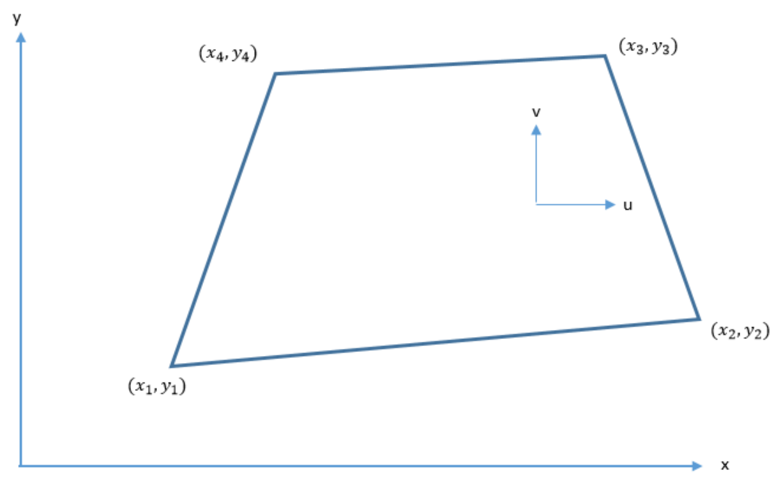


Figure 9: Isoparametric Quadrilateral Element

It follows similarly that the stresses and strains in the isoparametric quadrilateral element can be related to equations 26 and 31 respectively (Zienkiewicz and Taylor, 2000b; Bathe, 2014; Logan, 2016). However, $[B]$ is now developed using the isoparametric formulation for the element. The equation for $[B]$ is given by equation 34.1, where a, b, c and d are constants, $N_{i,s}$ and $N_{i,t}$ are the are the derivatives of the shape functions with respect to the s and t directions of the element and s and t are the coordinates of the integration points in the normalized coordinates (Logan, 2016).

$$[B] = \frac{1}{|J|}[[B_1] \quad [B_2] \quad [B_3] \quad [B_4]] \tag{34.1}$$

$$[B_i] = \begin{bmatrix} aN_{i,s} - bN_{i,t} & 0 \\ 0 & cN_{i,t} - dN_{i,s} \\ cN_{i,t} - dN_{i,s} & aN_{i,s} - bN_{i,t} \end{bmatrix} \tag{34.2}$$

$$a = \frac{1}{4}[y_1(s-1) + y_2(-s-1) + y_3(1+s) + y_4(1-s)] \tag{34.3}$$

$$b = \frac{1}{4}[y_1(t-1) + y_2(1-t) + y_3(1+t) + y_4(-1-t)] \tag{34.4}$$

$$c = \frac{1}{4}[x_1(t-1) + x_2(1-t) + x_3(1+t) + x_4(-1-t)] \tag{34.5}$$

$$d = \frac{1}{4}[x_1(s-1) + x_2(-s-1) + x_3(1+s) + x_4(1-s)] \tag{34.6}$$

$$[J] = \frac{1}{8}\{X_c\}^T \begin{bmatrix} 0 & 1-t & t-s & s-1 \\ t-1 & 0 & s+1 & -s-t \\ s-t & -s-1 & 0 & t+1 \\ 1-s & s+t & -t-1 & 0 \end{bmatrix} \{Y_c\} \tag{34.7}$$

$$\{X_c\}^T = \{x_1 \quad x_2 \quad x_3 \quad x_4\} \tag{34.8}$$

$$\{Y_c\} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} \tag{34.9}$$

3 Chapter 3: Analysis Model Structure

3.1 Introduction

In short, the proposed analysis model (or proposed program project) comprises a core framework consisting of a window, developed using Java Swing, which provides the user with all the required functionality to generate a finite element model from an input file. The user can subsequently perform an analysis on the finite element model and generate results and other graphical representations from the analysis. The core framework of the model is illustrated in Figure 10.

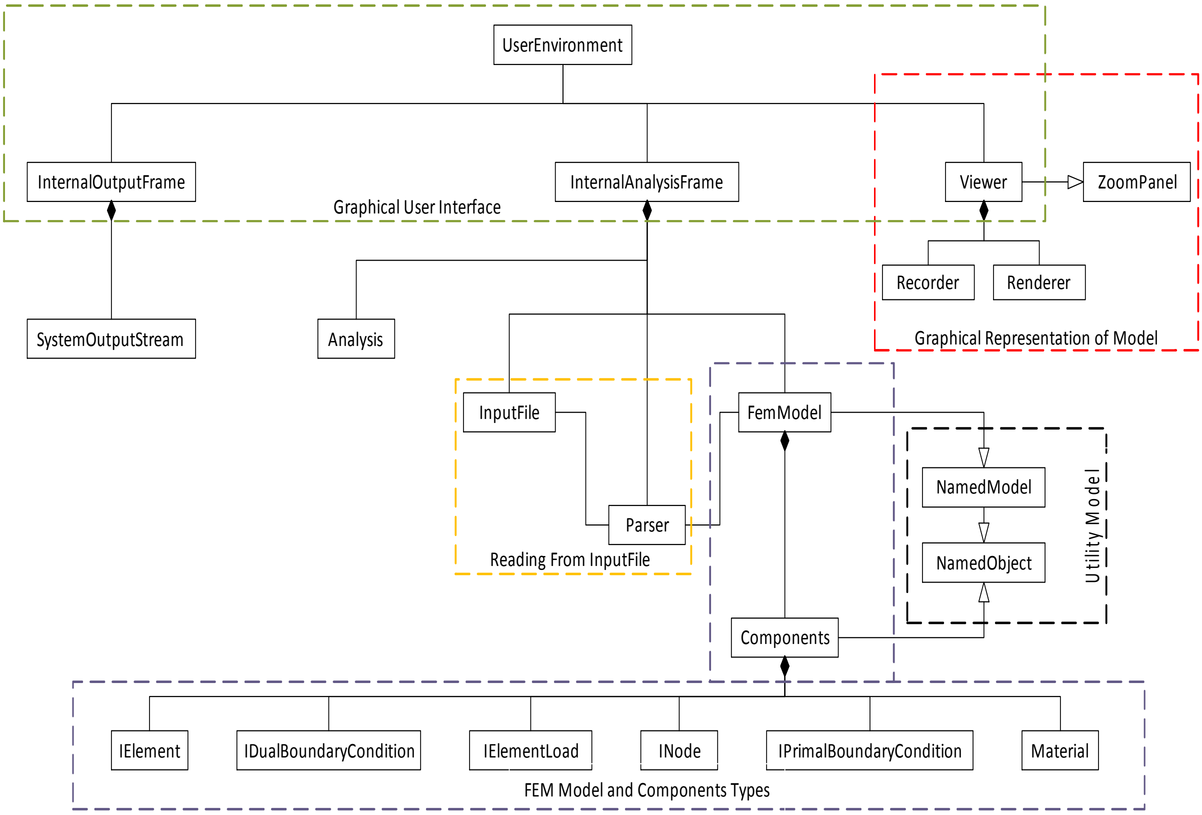


Figure 10: Core framework of the proposed Analysis Model

The core framework of the proposed model can be separated into sub structures as shown in the Figure. In the first section of this chapter, attention is drawn to the data structure for the *FemModel*. The *FemModel* is the model object on which an analysis is performed. The *FemModel* is constructed out of various component types which can be seen in Figure 10. Managing the components within the *FemModel* can be rather insurmountable and so an admirative structure has been developed to manage the set of components within the model. This administrative structure is called the Utility Model and is discussed herein.

As mentioned previously, components are generated through parsing an input file. These components include nodes, finite elements, material and boundary conditions, for example. The structure used for parsing input files is further explained herein. Following, the structure developed to generate a graphical representation of the *FemModel* is provided.

In the last section of this chapter, focus is shifted towards the overview and control flow of the program's user interface. This forms the basis of how to use the user interface. The program's user interface has been structured to allow the user to perform consecutive finite element analyses which do not necessarily have to be of the same type. Furthermore, the process illustrates ways in which to enable the user to successfully perform an analysis through prompts and messages on the screen.

3.2 FemModel and Components

A model in the context of this study is a construct of various sub sets of components when combined represent finite element objects. The component types required for generating a model can be seen in Figure 10. In general, the finite elements which make up the model, consist of nodes, materials and element loads which define the element geometry, material properties and the elements source vector respectively. Furthermore, the nodes which make up the finite element geometry may contain primal or dual boundary conditions.

All finite elements classes in the proposed model are sub classes of *AbstractElement*. Abstraction is used in this case to avoid repetitive code and methods. *AbstractElement* implements the interface *IElement* which contains all the necessary methods to generate the system equations from the finite element objects. The UML class for *AbstractElement* is provided in Figure 11. *AbstractElement* contains attributes for the number of degrees of freedom the element has as well as the number of degrees of freedom per node. It also contains the names of the nodes which define its geometry and the name of the material which defines its material properties. Furthermore, *AbstractElement* contains the shape functions of the element and integration scheme which is required for computing the [k] matrix. The class *AbstractElement* also contains general get and set methods for each of its attributes as well as a method to compute the [X] matrix and a method to compute the nodal primal vector [u] for the element. The methods for determining the element [k] matrix is specific to the type of element involved and thus cannot be determined in abstract element.

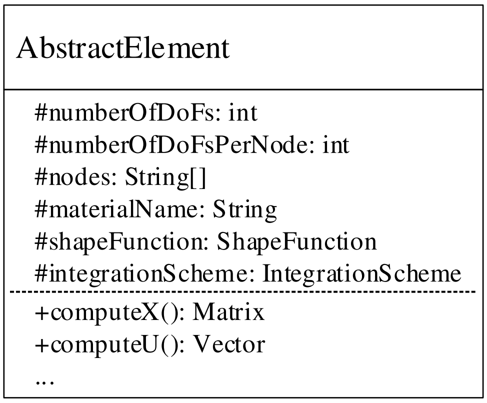


Figure 11: AbstractElement UML Class Diagram

The subclasses of *AbstractElement* currently implemented in the model include *Triangle*, *Quad4Iso*, *ConstantStrainTriangle* and *PlaneQuad4Iso*. *Triangle*, as the name implies, is a triangular element used for any type of analysis which only requires one degree of freedom per node. For elastic plane stress and plane strain analyses triangle elements can be implemented in the model using *ConstantStrainTriangle*. Similarly, *Quad4Iso*, as the name implies, is an isoparametric quadrilateral element. *Quad4Iso* can be used in any analysis which requires one degree of freedom per node. For elastic plane stress and plan strain analyses an isoparametric quadrilateral element has been develop named *PlaneQuad4Iso*.

Quad4Iso and *Triangle* elements contain functionality to add a constant element load across the entire element. This is done using *ElementLoad* objects, which require a name, name of the element on which the load occurs and the load intensity.

Abstraction and interfacing has further been used for the development of other components in the model such as for *Node*, *PrimalBoundaryCondition* and *DualZeroDBoundaryCondition*. This decision was made to ensure that any future extensions or additions avoid redeveloping the classes which are already implemented.

A class *FemModel* was developed to combine all these components required to perform a finite element analysis into a model structure. Adding and processing the above components into the *FemModel* follows the process illustrated in Figure 12.

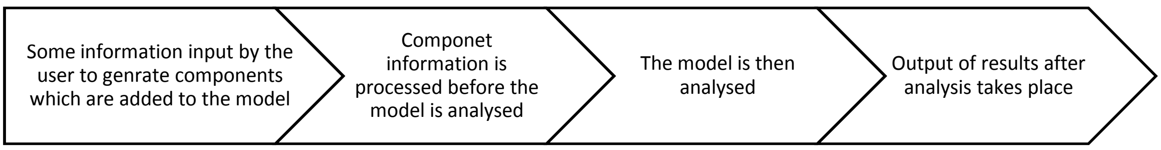


Figure 12: Structure of creating and analysing a model

Notably, *FemModel* contains methods which particularly address the processing before an analysis and methods for generating output after the analysis has been completed. The UML class diagram of *FemModel* is illustrated in Figure 13.

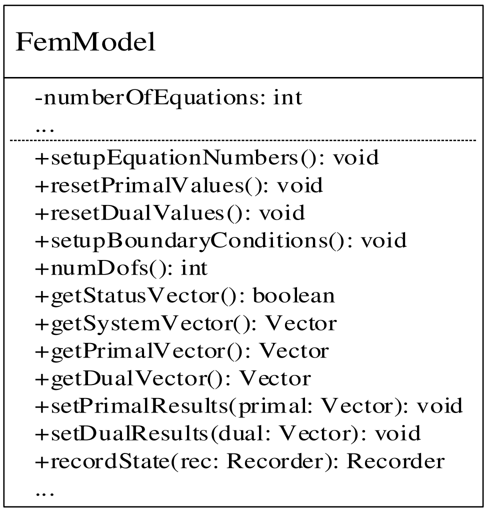


Figure 13: FemModel UML Class Diagram

As seen in Figure 13, the *FemModel* contains the attribute *numberOfEquations* which represents the total number of degrees of freedom for the model. The *FemModel* also contains get and set methods for developing the finite element system equations to perform an analysis. Furthermore, the *FemModel* also contains get and set methods for post analysis and generating the results of the analysis. In general, these get and set methods iterate through the applicable components of the model and extract and add information to the components where fit.

3.3 Utility Model

The utility model sub structure was developed to administrate all the components which are contained in the analysis model. The utility model comprises a *NamedModel* illustrated in Figure 14 with a registry of objects which are named and mapped to their particular names. This mapped registry contains objects of type INamedObject whose UML class is illustrated in Figure 15.

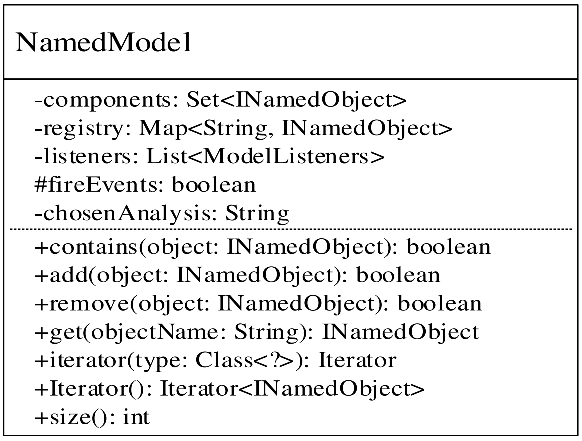


Figure 14: NamedModel UML Class Diagram

When components are required during the processing and analysing phases mentioned in Figure 12, the components are called from the registry by their name. This step is done by providing an *Iterator* for the particular component class of interest. A benefit of using this administrative technique is for example when *IElement* objects are instantiated, they are not provided with the *Node* objects of which they are comprised but only the node names of which they are comprised. This means that components in the model do not have to be created in order.

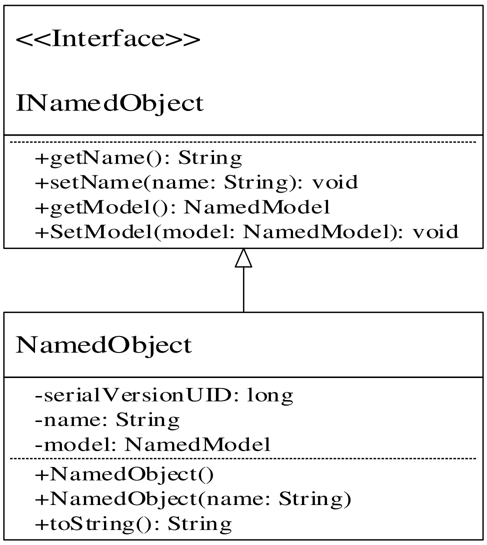


Figure 15: NamedObject UML Class Diagram

3.4 Reading from an Input File

Components are generated and added to the model using the method of parsing text read from an input file. An overview of the parser framework is illustrated in Figure 16.

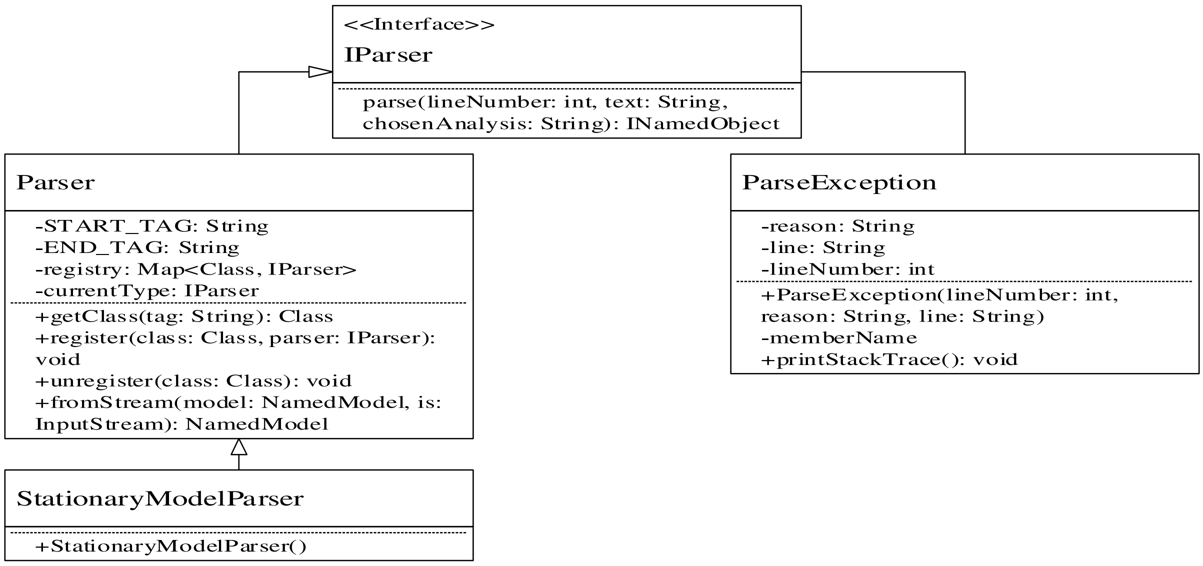


Figure 16: Parser Framework

Each component type to be parsed from an input file must have its own parser. This is because components do not have generic constructors. The parser generates objects only between start and end tag attributes. These attributes are specified in the class itself. To create a *Parser* which can read a new component, the *Parser* must implement *IParser* and be registered in the *StationaryModelParser*.

3.5 Graphical Representation of the FemModel

A graphical representation of the *FemModel* is displayed using a *Viewer* object. A *Viewer* object contains a *JMenuBar* object in which the user is given options to add or remove graphical objects from the *Viewer*. Furthermore, the name of the input file on which the analysis was performed is displayed within the viewer to ensure that the user has clarity on which analysis model is being represented.

Importantly, *FemModel* objects with selected analysis are stationary heat flow, potential flow or groundwater seepage are displayed in *Viewer* while elastic plane stress or plane strain analysis is displayed in an *ElasticViewer*. Elastic plane stress and plane strain problems require graphical output which differs to other types of finite element analyses. For example, the stresses and strains at points within an element or the displacement vectors at nodes may be required. Thus, a subclass of *Viewer* was established, namely *ElasticViewer*, in which the methods required to generate graphical representation are overridden. An illustration of the graphical representation framework is provide in Figure 17.

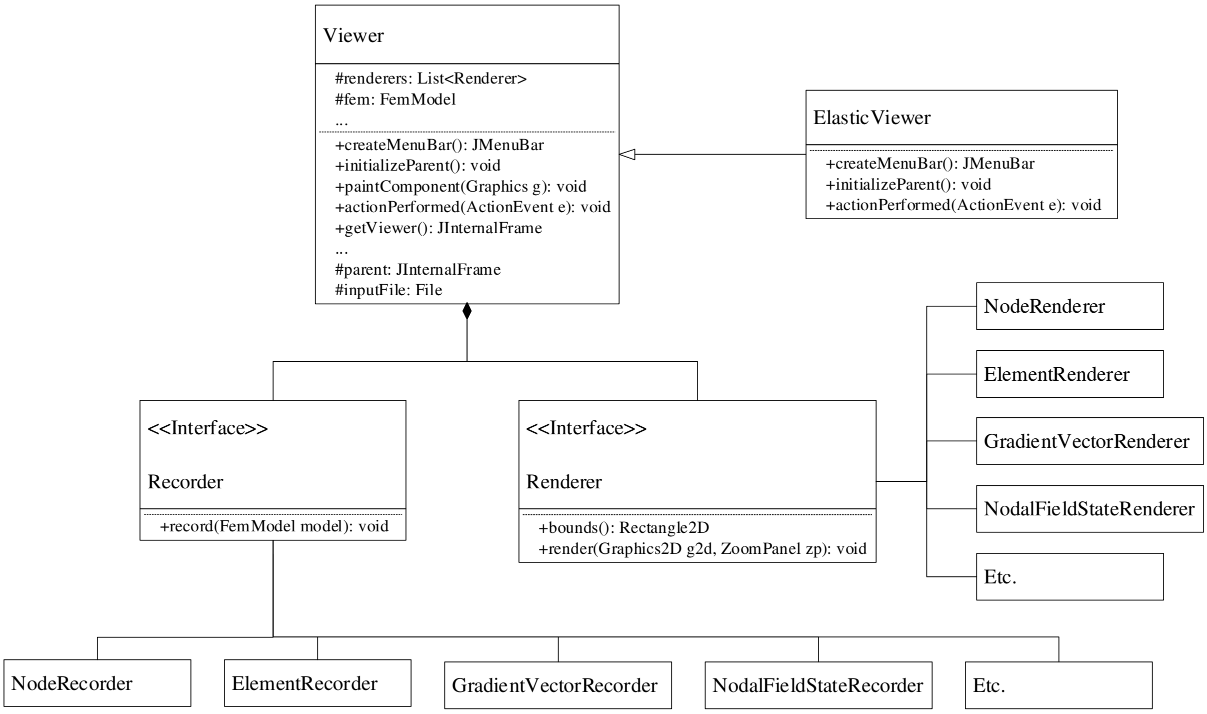


Figure 17: Graphical Representation of FemModel Framework

In order to render graphical objects into the *Viewer*, components which are required to be rendered must first be recorded from the *FemModel* using a *Recorder* object. *Recorder* objects contain an *Iterator* object which comprises all the components registered in the *FemModel* for the specific object *Class* to be rendered. The *Iterator* is then used to create new objects which are stored in the Recorder. These new objects can be used to later represent graphical objects in the Viewer. To display the new objects, a *Render* object iterates over the recorder and paints the components to the *Viewer* (or *ElasticViewer*).

3.6 Overview Graphical User Interface

3.6.1 Control Flow of Proposed Analysis Model

The design of the control flow for the user interface was developed to provide an understandable process in which the user is able to interact with the program. This process is shown in Figure 18, where the blue blocks indicate the main path in which an analysis can be executed by the user. The premise for the selected design is to provide a robust and organised structure to the graphical user interface which allows the user to purposefully make clear and consistent decisions throughout the program's use.

When the user starts the program, a frame with a menu appears on screen. From the menu, the user can choose to start a new analysis which results in an analysis frame appearing within the original frame. In the analysis frame, the user has the option to select which type of analysis they would like to perform. Once the analysis type has been selected, a dialog box appears on screen to ask the user to specify which input file is to be selected to perform the analysis. At this point, the process is protected by the fact that the user can only select input file extensions which are linked to the specific analysis type selected. For example, if the user has selected stationary heat flow analysis, the input dialog will only allow the user to select input files with the relevant extension ".shf". Once the input file is selected, a *FemModel* is created with the components parsed from the input file. Following this, an analysis is automatically performed on the *FemModel*.

When the analysis is complete, the analysis frame then indicates to the user that the analysis has been performed successfully. The user then has the option to review the results of the analysis by either opening the output of the analysis or generating graphical representation of the results of the analysis. Once the user is finished reviewing the results, it is possible to quit the program or close the analysis frame and open a new analysis. An essential portion of the control flow design is to account for tolerance in the user's decisions. It is possible that a user can make errors at some stage of the process. These errors can, to some extent, be anticipated and are accounted for in the design of the user interface. The orange blocks in Figure 17 depict the remedying processes for such errors. For example, the user may try to open the graphical output of an analysis before an analysis has been opened. To account for this, a dialog box opens indicating that a mistake has been made and an explanation is provided on how to rectify the mistake. In the implementation of the program, this was done using the Java Swing class JOptionPane. Although this may seem rudimentary to solve such a problem, it is a more efficient solution than developing new Java object classes to implement similar results for such a small program.

Importantly, an essential part of the design specified in the aims of this study is to provide users with resources to assist them with use of the program. These resources include java documentation, examples of analyses and a user manual. Incorporated into the design of the program is the functionality to open these mentioned documents at any stage in the process.

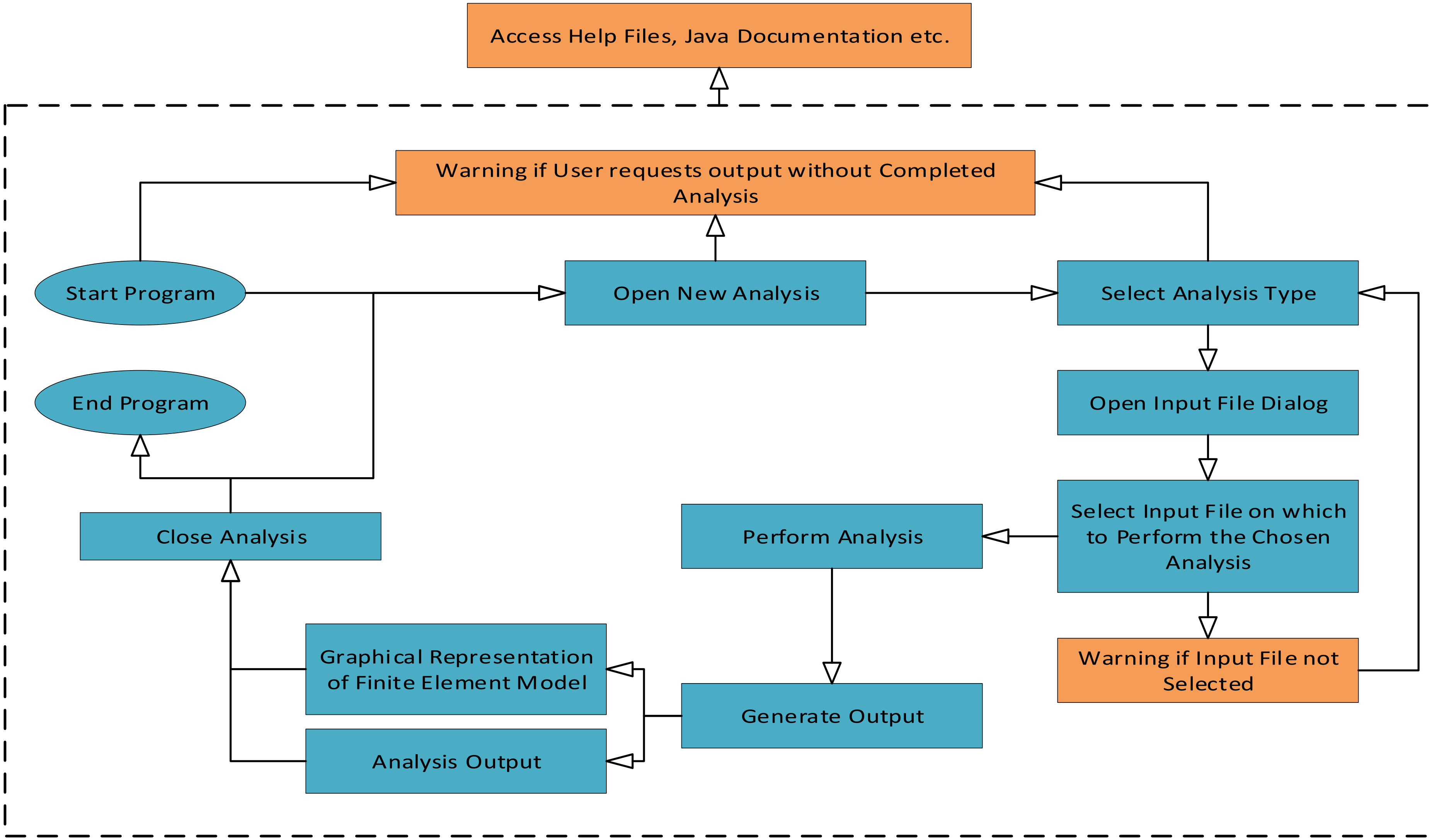


Figure 18: Control Flow Diagram of Graphical User Interface

3.6.2 Graphical User Interface Data Structure

In the proposed analysis model, the main window in which the user creates a finite element model and performs a finite element analysis is called the *UserEnvironment* whose parent class is *JFrame*. A UML class diagram of the *UserEnvironment* and its associated components is provided in Figure 19.

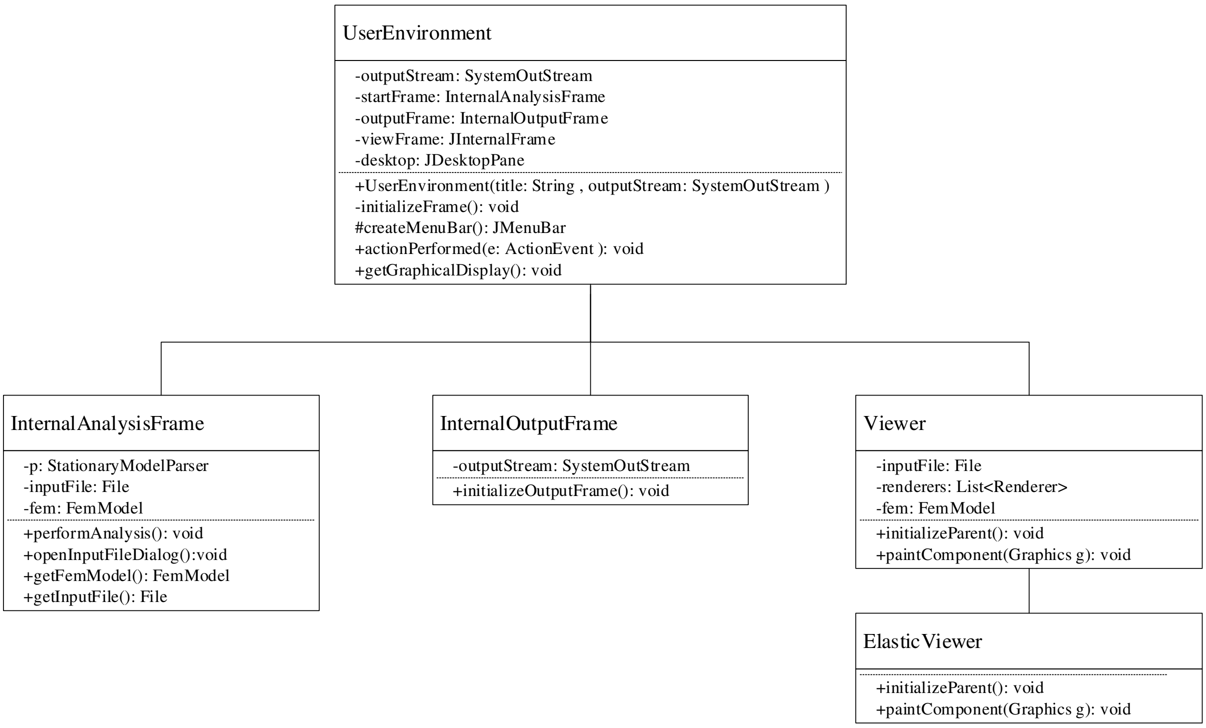


Figure 19: Graphical User Interface UML Diagram

The *UserEnvironment* object which the user creates when starting the program consists of a *Container* of class *JDesktopPane.* Within the *Container* sits internal frames which the user performs all the necessary works for selecting, creating and performing a finite element analysis. The reasons for this choice of *Container* is create the look and feel of a virtual desktop in which internal frames can be added moved and resized.

The *UserEnvironment* contains a *JMenuBar* in which the user can select functions they would like to perform the model. The *JMenuBar* contains a “Help” menu in which the user can access documents to assist with the use of the model. The documents which can be accessed from the “Help” menu include Java documentation for the classes of the analysis as well as this text. Java Documentation is tool which is developed in a predefined format allowing users to gain further understanding into the workings of the analysis model. The Java documents in turn offer additional explanations of classes and methods to the students of Engineering Informatics 314 and any other users of the analysis model. The java documentation provided for the analysis model is not complete for every package in the analysis model simply due to the time constraints of this study. The documents are only there to provide some essential information to the students of Engineering Informatics 314 and illustrate that the development of generic documentation can be done for projects such as this one. Importantly, the Java documentation is opened in the default internet browser of the computer on which the analysis model is installed. If the default browser is not set, the user will be prompted from the computer to select a browser in which to open the document. Other menus are also offered to the user in the *UserEnvironment* for example, the “Graphical Output” menu. This menu is where the user can find ways to display the output of an analysis. The *JMenuBar* also contains a “File” menu – in which the user can create a new analysis.

When a user creates a new analysis, an *InternalAnalysisFrame* is instantiated. The *InternalAnalysisFrame* extends *JInternalFrame* and is an attribute of the *UserEnvironment*. Within the *InternalAnalysisFrame*, the chosen analysis type is selected by the user. The reason for declaring the *InternalAnalysisFrame* as an attribute is to ensure that the chosen analysis can be linked between the *InternalAnalysisFrame* and other internal frames within the *UserEnvironment*. When the user selects an analysis type to perform, the method *openInputDialog()* is called. This method instantiates a JFileChooser object and restricts the extensions filter of the JFileChooser object to the chosen analysis' associated extension. The user can then select the file which contains the necessary objects which they wish to parse and add to the model. The *InternalAnalysisFrame* contains a *FemModel* object which is the model in which a *StationaryModelParser* adds the components parsed from the chosen input file. Once all the components have been parsed from the input file, an analysis is performed using the method *performAnalysis()*. When the analysis is completed, the internal analysis frame then displays the status of the analysis and the file on which the analysis has been performed.

The results of the analysis are displayed in an *InternalOutputFrame* which is a component of the *UserEnvironment*. The *InternalOutputFrame* contains a *SystemOutputStream* object which is responsible for redirecting all *System* output to a *JTextArea*. The results displayed in the *InternalOutputFrame* contain the system primal, dual and load vectors as well as the element and global matrices for the finite elements in the model. The graphical representation of the FEM Model is displayed in either a *Viewer* or *ElasticViewer* object depending on the chosen analysis.

4 Chapter 4: Examples using the Analysis Model

4.1 Introduction

The focus of this chapter is to present a method which must be followed to correctly determine solutions to various finite element problems using the analysis model. A worked example for each of the types of analysis is provided herein, namely stationary heat flow, groundwater seepage, potential flow and elastic plane stress. Emphasis is placed on properly applying boundary conditions including the correct use of flux on the boundary for each specific problem. The works herein are to be used in conjunction with the example input files associated with each problem. The specific input file required for each problem is specified in the section applicable.

4.2 Using the Analysis Model

When the executable for the analysis model is run, the UserEnvironment frame appears on computer's screen as illustrated in Figure 20.

Figure 20: Analysis Model UserEnvironment Frame

To perform an analysis, the user must select "New Analysis" from the "File" menu which opens a new *InternalAnalysisFrame* object in the *UserEnvironment*. The InternalAnalysisFrame prompts the user to select a type of analysis which they would like to perform. This is illustrated in Figure 21. The user can select from a list of analysis types in the *InternalAnalysisFrame*. A short description of the analysis type selected by the user is provided below.

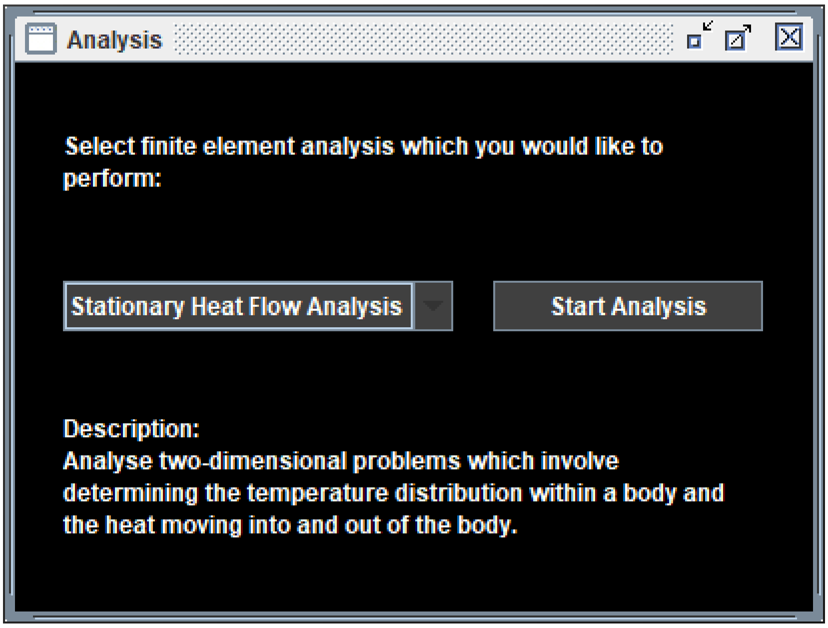
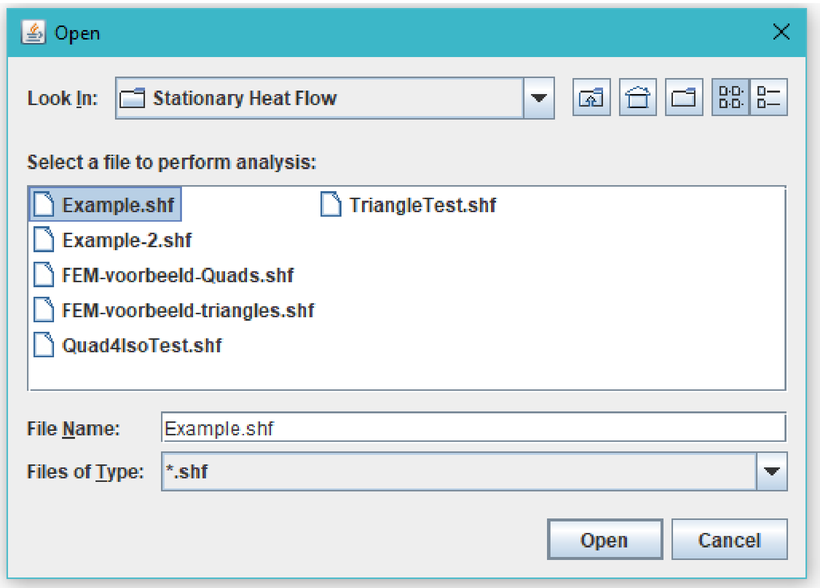


Figure 21: InternalAnalysisFrame Before Analysis is Performed

When the user clicks the “Start Analysis” button in the *InternalAnalysisFrame*, a *JFileChooser* appears on screen where the user is prompted to select the input file which they would like to analysis. A photograph of the *JFileChooser* is illustrated in Figure 22. Note the file extension of the *JFileChooser* is restricted to the type of analysis selected by the user. In the case of the Figure, “Stationary Heat Flow Analysis” was selected.



Extension Types:	
Problem	**Extension**
Stationary Heat Flow	.shf
Groundwater Seepage	.seep
Potential Flow	.ptflw
Elastic Plane Stress/Strain	.elas

Figure 22: Photograph of JFileChooser and associated extension types

Once the user has selected the appropriate file and clicks the “Open” button, an analysis is performed. The analysis may take several second for models with a high number of elements.

When the analysis is completed, the InternalAnalysisFrame reports the success of the analysis and the file name of the input file as illustrated in Figure 23.

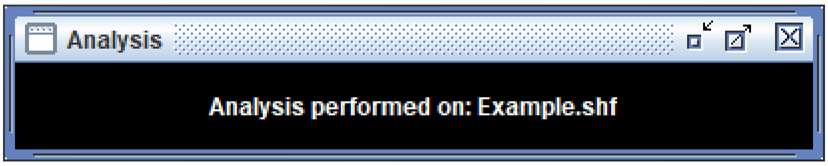


Figure 23: InternalAnalysisFrame After Analysis has been Performed

The user may then choose to select the results which they would like to display from the "Display" menu. The results of the analysis can be displayed by selecting "Show Output" from the "Display" menu. Similarly, a *Viewer* object which shows the graphical representation of the model can be displayed by selecting "Show Graphical Output" from the "Display" menu. The default view for the View shows the *Node* and *Element* components in the model as shown in Figure 24. The user has the option to add or remove components from the graphical represented model using the "Add" and "Remove" menus respectively. The *Viewer* object also shows the name of the input file on which the analysis was performed.

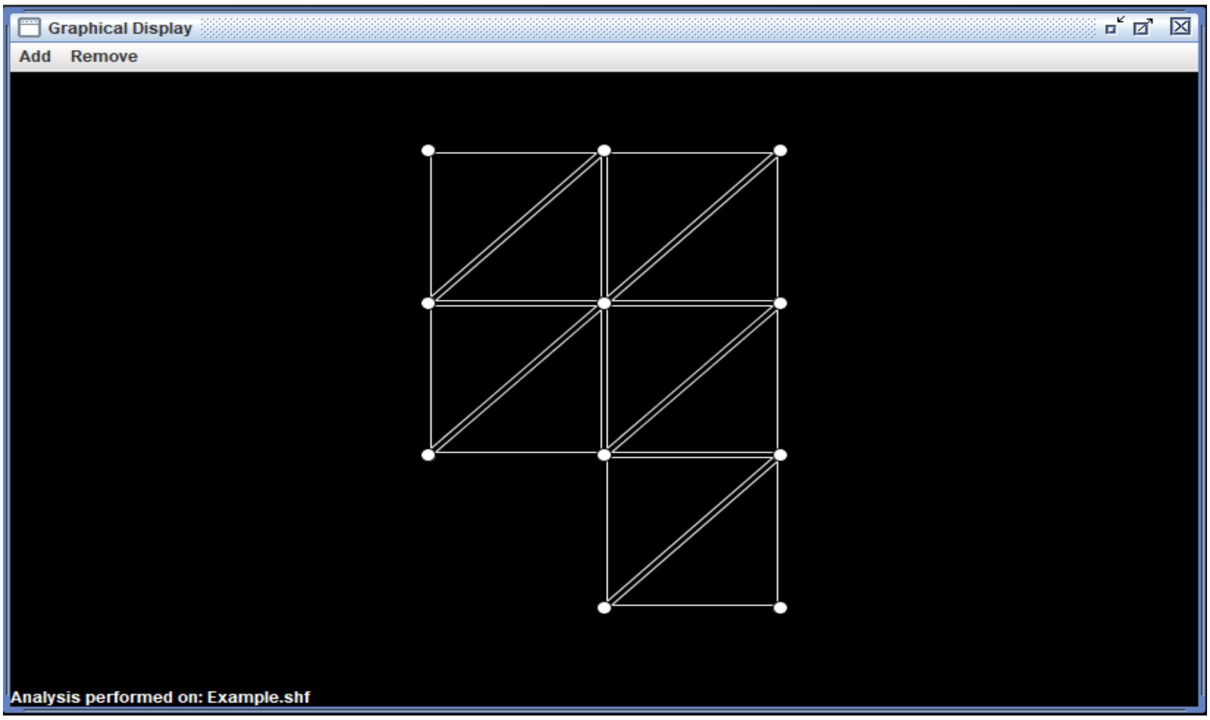


Figure 24: Viewer object or "Graphical Display"

4.3 Example Using Analysis Model for Groundwater Seepage Problem

4.3.1 Problem Statement

The formation of pore water pressure in the phenomenon of groundwater seepage forms a crucial parameter governing the stiffness and strength in permeable soils (Knappett and Craig, 2012). Therefore, it is critical that the pore water pressure is known when designing and constructing of structures such as foundations, retaining walls, dams and weirs for example. The case described here considers the modelling of a dam with geometry illustrated in Figure 25.

An Engineer has been asked to determine the uplift pressure under the base of the dam and indicate particular points where erosion may be occurring due to the seepage under the dam. A geotechnical survey confirmed the conditions in the permeable layer of soil. There was found to be no cross flows or sources present at or near by the dam. Furthermore, the permeability in the soil was found to be homogeneous and isotropic with permeability coefficient of 2.5×10^{-5}m/s.

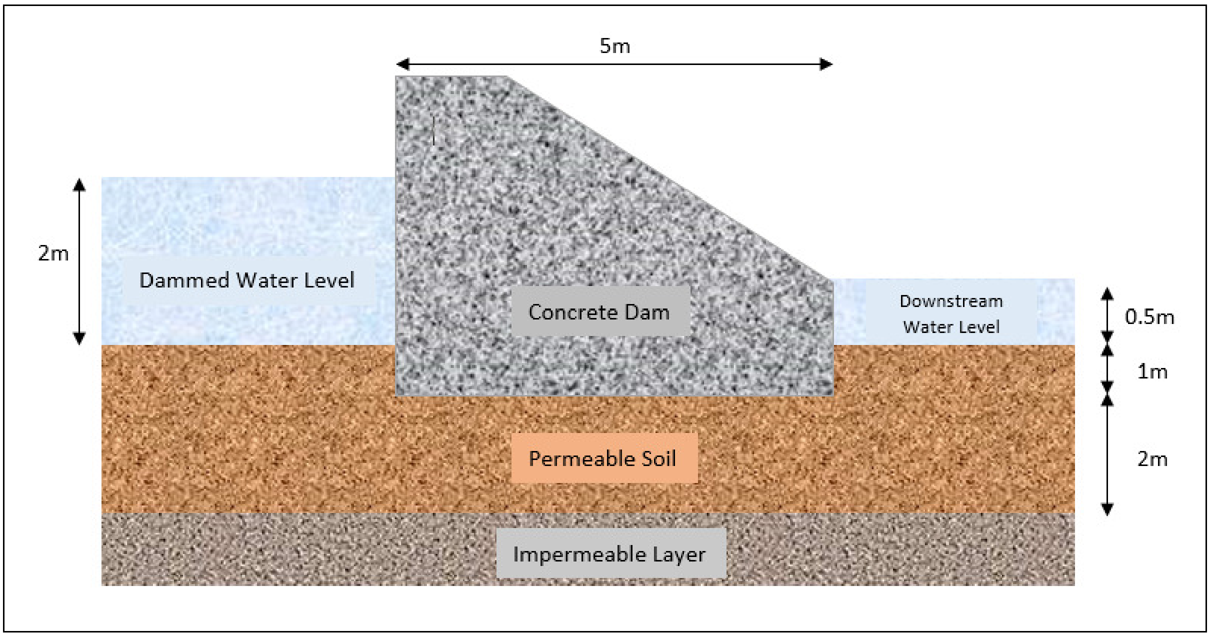


Figure 25: Geometry of Groundwater Seepage Case Example

4.3.2 Solution Outline

The problem described above is analysed in the analysis model using the input file "SeepageUnderDam.seep". The model developed considers a unit thickness of the dam structure and surrounding soil. The first step to the solution using the analysis model for the above problem is to determine a suitable mesh of elements for the permeable soil layer. This requires creating a set of *Node* objects which in union define the geometry of the finite elements making up the body of soil. Once the nodes have been established, the finite elements can be created - each with specified name, the names of the nodes specified in a counter clockwise direction and the name of the material. In the case of this example, a 1m x 1m Quad4Iso mesh was used.

Following, the boundary conditions can be specified at the nodes. Groundwater seepage problems require either the pore water pressure or flux to be specified at a point. Since there were no other flows or internal sources specified in the problem, the boundary conditions specified in the input file are only the points at which pore water pressure is known. These are created using *PrimalBoundaryCondition* objects, in which the name, nodes, and value of the boundary condition is specified. The water bodies upstream and downstream are considered to be stationary and so the value specified at the boundary is simply the water level at the nodes on the surface of the soil body upstream and downstream of the dam.

The last requirement to perform an analysis on the model is to specify a material which defines the finite element material properties. Materials in groundwater seepage analysis are created by developing *SeepageMaterial* objects through the input file as shown in "SeepageUnderDam.seep", in which the name of the material and the permeability coefficient is specified.

4.3.3 Results from the Analysis

The results of the analysis for determining the uplift pressure underneath the dam is displayed in Figure 26 using the "Graphical Display" of the analysis model. The results show the calculated pore water pressures at the nodes directly underneath the dam.

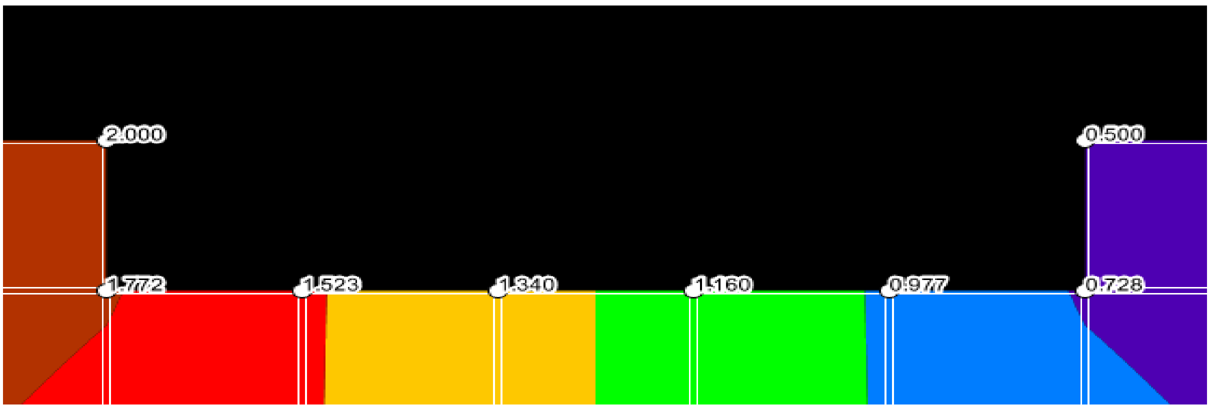


Figure 26: Pore Water Pressures underneath the Dam

From the above calculated pressures, the uplift pressure distribution can be determined simply by hand calculations and is provided in Figure 27.

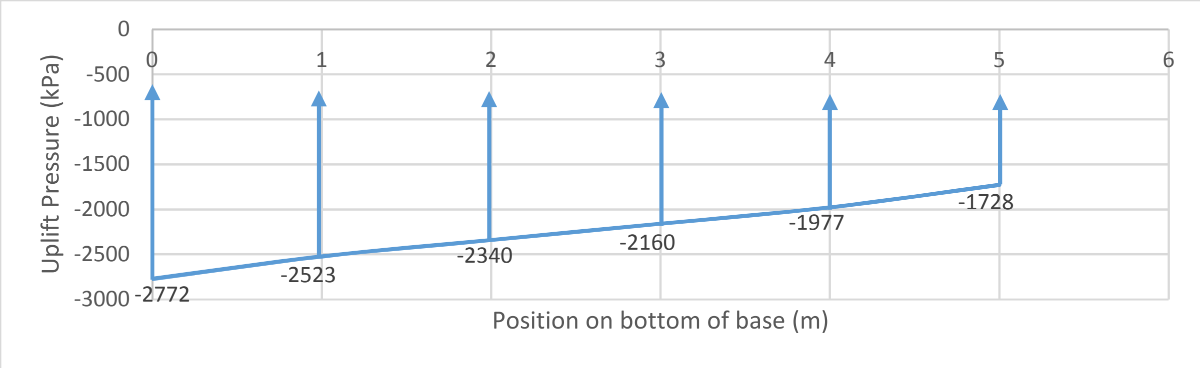


Figure 27: Uplift Pressure underneath Dam Base

The points that are most highly susceptible to erosion are the points at which the seeping water under the dam undergoes acceleration (Knappett and Craig, 2012). These points can be found by determining the points at which there is an increase downstream in the pore water pressure gradient - i.e. an increase in the flow velocities. This can be computed in the analysis model by adding "Gradient Vector" objects to the "Graphical Display" as shown in Figure 28.

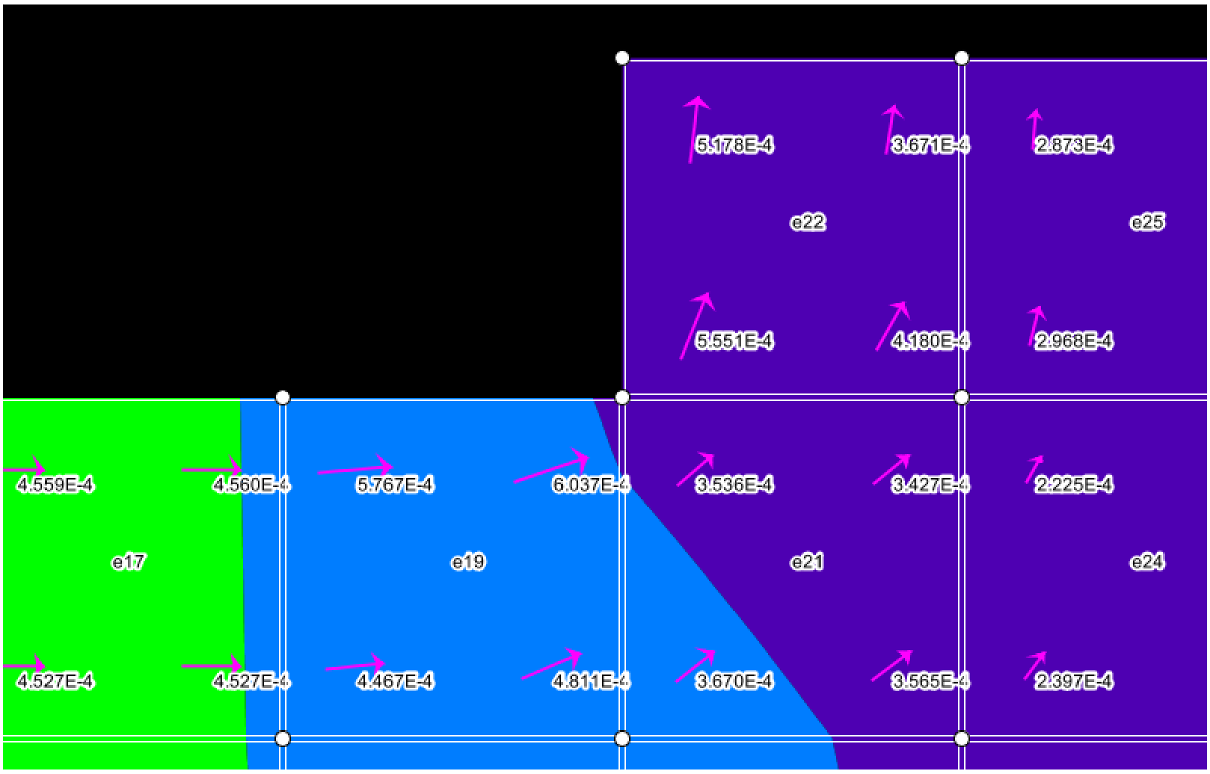


Figure 28: Gradient Vectors at Toe of Dam

Note the increase in the magnitude of the gradient vectors in elements "e19" and "e22" near the toe of the dam. These increases indicate particle acceleration and increased likelihood of erosion. However, it is important to note that the mesh used in the analysis is rather course and may be required to be refined for increased accuracy.

4.4 Example Using Analysis Model for Potential Flow

4.4.1 Problem Statement

The purpose of this section is to present an example of a potential flow analysis using the analysis model. The formation of the potential flow example discussed here considers the flow of an idealized fluid through a two-dimensional channel of unit thickness with geometry and boundary conditions as as shown in Figure 29. Important to note is that for conservation of mass in the body, the boundary conditions specified on the surface of the body must be in balance. That is, if flow enters the channel then the same amount of flow must exit the channel.

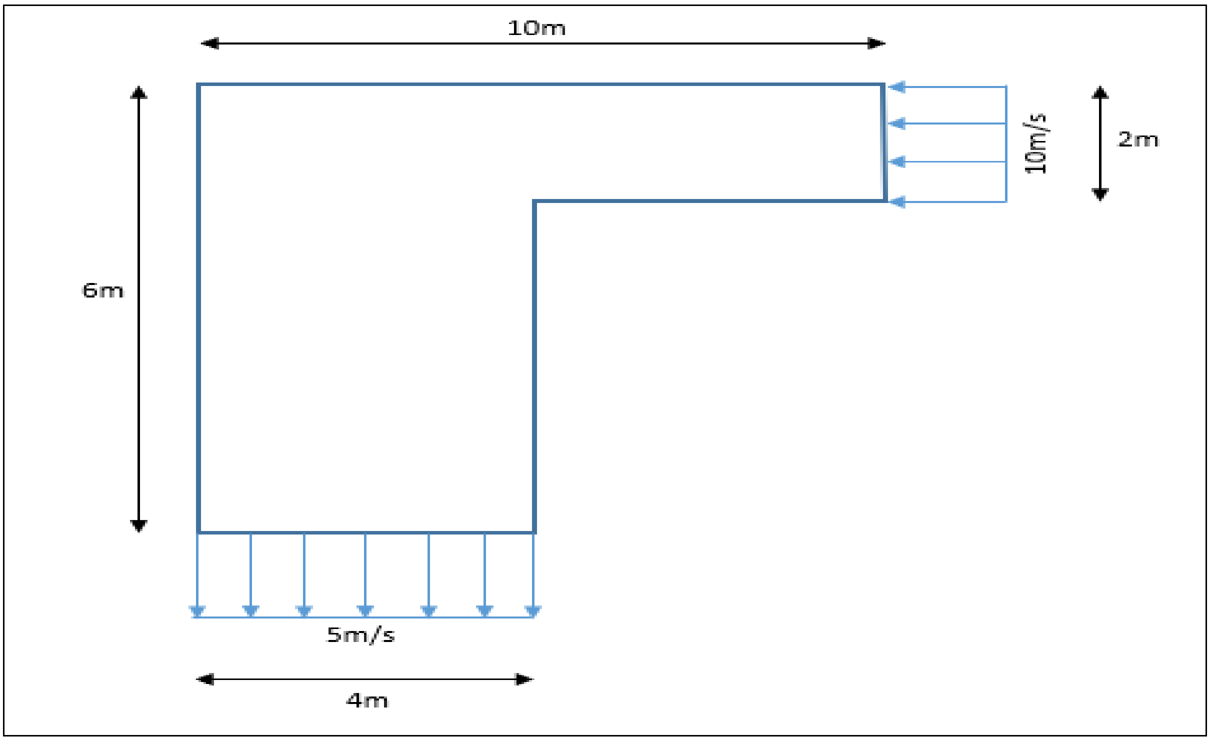


Figure 29: Diagram of Potential Flow Example

4.5 Solution Outline

A similar set up and approach is followed here as for groundwater seepage. The input file to be used in conjunction with this solution is named "PotentialFlowExample.ptflw". First the geometry of the body must be divided up into a number of finite element objects in the same manner as for groundwater seepage. First, *Node* objects are created to define the finite element geometry. Potential flow considers ideal fluid flow and therefore no material is required to define the material properties. Therefore, a finite element in the system only contains a name and its defining geometry. The geometry must be specified in a counter clockwise direction for each element. In the case of this example, a mesh of 1m x 1m *Quad4Iso* element objects where used as the finite elements for the system.

The boundary conditions in this example are clearly specified as flux in Figure 29. In order to evaluate the problem, the flux must first be converted to streams at nodes which the flux occurs. This is because the analysis model provides no implementation to define flux as a function on the boundary. The converted flux on the boundary is specified in the analysis model's input file as *DualZeroDBoundaryCondition* objects. These objects require a name, the name of the node at which the boundary condition occurs as we as a value of the stream intensity. Lastly, the analysis model requires at least one point on the boundary of the body to have a specified velocity potential. This is required to fully define the boundary value problem and to solve the finite element system equation. The velocity potential at this point can be considered as an "anchor" for the body somewhere in space, or otherwise considered as a datum reference for the system. The velocity potential boundary condition is specified using a *PrimalBoundaryCondition* object in the input file. For this object only the name, the name of the node at which it occurs as well as the velocity potential intensity are required.

4.5.1 Results of the Analysis

The results from the "Graphical Display" may provide two useful outcomes from the analysis. First the results of the subsequent velocity potentials at the nodes. It is important to note that the potentials at the nodes are determined relative to the anchor potential velocity mentioned previously. Therefore, the value of importance is in fact the difference between the results of the velocity potential at a node and the anchor velocity potential. Verifying this statement can easily be proved by changing the anchor velocity potential to a new value. The difference between the new results obtained at the nodes and the new anchor velocity potential will still be the same as determined in the first analysis. The results determined for the velocity potentials in the "Graphical Display" is given in Figure 30.

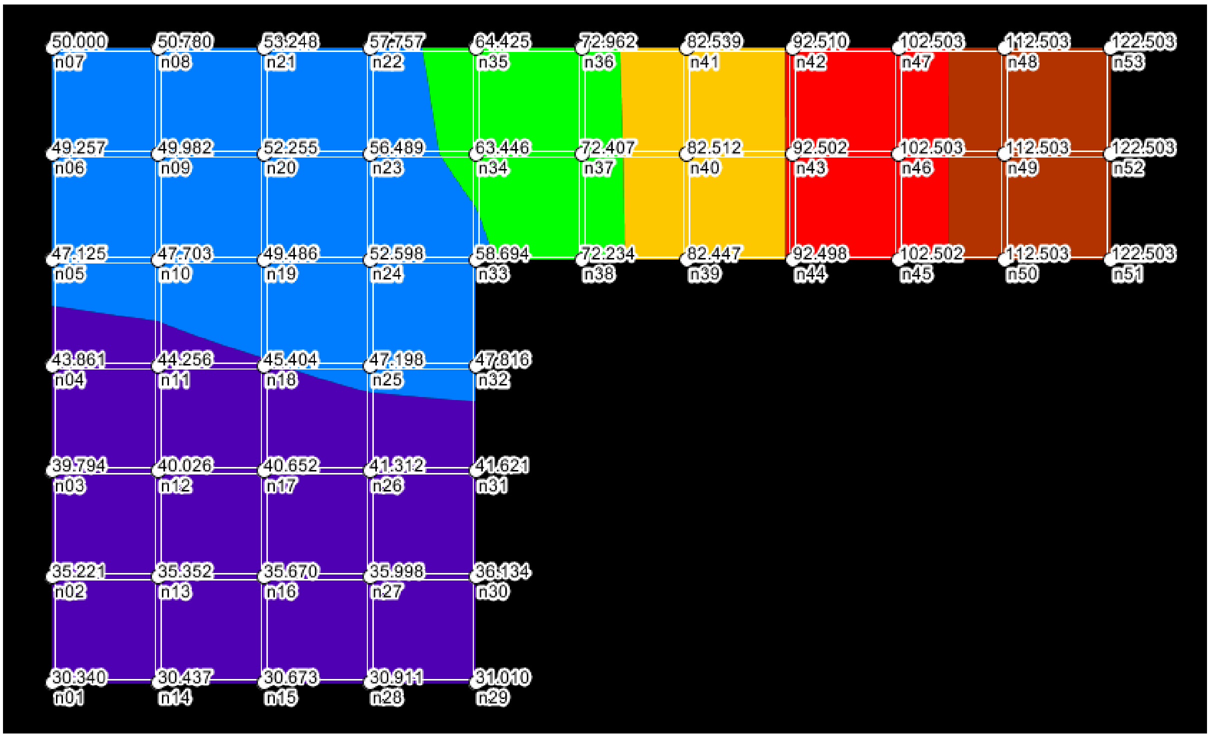


Figure 30: Results of the Velocity Potentials Determined for Potential Flow Example

The second useful result that may be useful to extract from the graphical display is velocity vectors at the integration points inside elements. From this problem, it can be expected that the that the velocities will slow down into the bend near node "n07" and follow the shortest path passing near node "n33". These results are confirmed by plotting "Gradient Vectors" in the "Graphical Display". The "Gradient Vectors" here, represent velocities at the integration points of the elements. A portion of the model emphasising the mentioned expected results is provided in Figure 31.

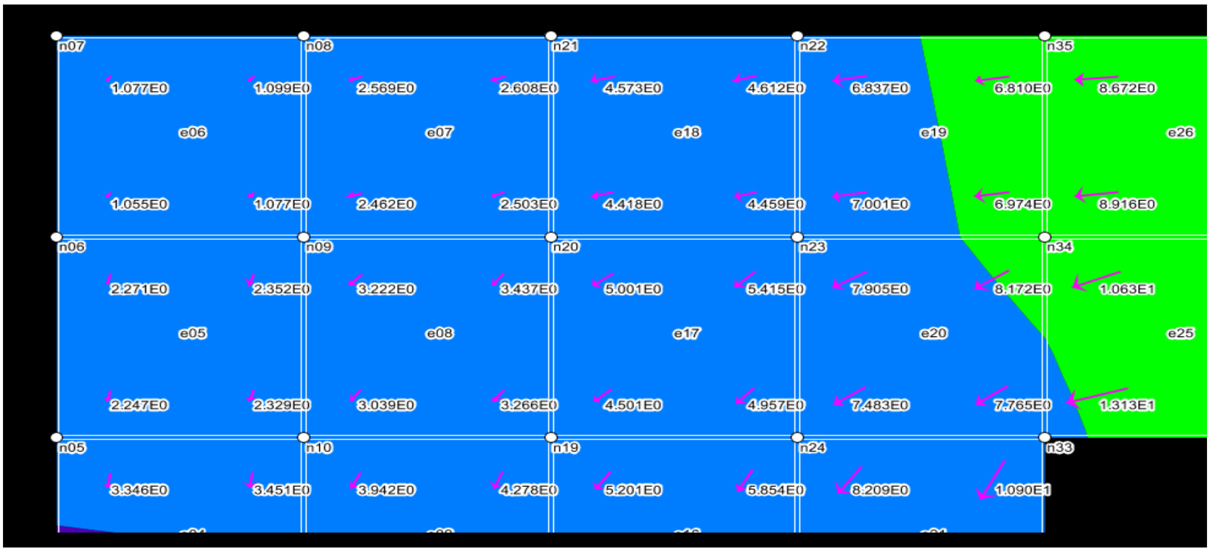


Figure 31: Velocities at the Integration Points for Potential Flow Example

4.6 Example Using Analysis Model for Stationary Heat Flow Problem

4.6.1 Problem Statement

The example discussed here considers a model for a bridge deck with geometry as shown in Figure 32. A group of Engineers who specialize in finite element analysis have been asked to investigate a reason which explains the cracking of concrete just under the surfacing layers at the positions illustrated in Figure 32.

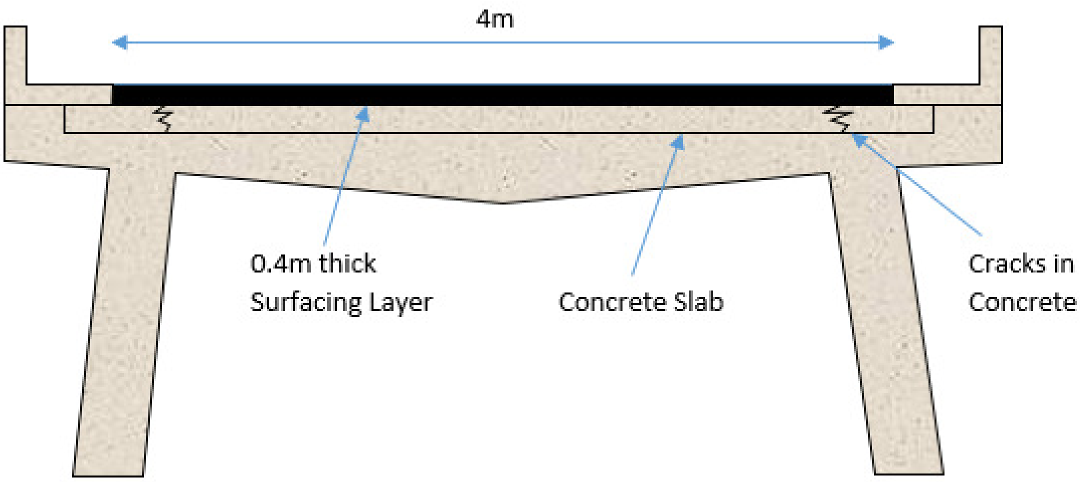


Figure 32: Layout of Bridge Deck

The team of engineers suspect that the cracks have developed due to the temperature deformation of the surfacing layer which sits on top of the concrete and have decided to perform a stationary heat flow analysis. Upon investigation, it was found that the heat conductivity of the surfacing layer was found to be 2200 W/m˚C. On extremely hot days the sun is expected to heat up the surface of the surfacing layer to 60˚C while the bottom of the layer is expected to remain at a constant temperature of 23˚C. A highly elastic rubber was placed between the surfacing layer and the side walk to avoid cracks. This rubber insulates the sides of the surfacing.

4.6.2 Solution Outline

The input file to be used in conjunction with this solution is named "StationaryHeatFlowExample.shf". The model used by the engineers considers a unit thickness and width of the 6m wide surfacing layer. The first step to the solution using the analysis model for the above problem is to determine a suitable mesh of elements to represent the surfacing layer. This requires specifying a set of *Node* objects in the input file which in union define the geometry of the finite elements making up the surfacing layer. Once the nodes have been established, the finite elements can be created - each with specified name, the names of the nodes specified in a counter clockwise direction and the name of the material. In the case of this example, a 0.1m x 1m Quad4Iso mesh was used. In stationary heat flow analysis, *StationaryHeatFlowMaterial* objects are used to define a material. *StationaryHeatFlowMaterial* objects contain a name and the material conductivity.

The boundary conditions in this example include temperatures and heat flux. It is important to note that stationary heat flow problems require either the temperature or flux to be specified at a point. Temperature boundary conditions are specified in the model using *PrimalBoundaryCondition* objects, in which the name, name of node, and value of the boundary condition is specified. The temperatures are defined for the problem on the top and bottom of the surfacing layer but not on the sides since the sides are insulated from flux (i.e. the flux is known). Flux is specified using *DualZeroDBoundaryCondition* objects. These objects require a name, name of the node at which it occurs and the intensity of flux. For insulated points, flux is specified with intensity of zero.

4.6.3 Results from the Analysis

The temperatures at the nodes from the analysis can be shown by selecting "Graphical Display" from the "Display" menu in the analysis model. Figure 33 shows a portion of the analysis model. The magenta vectors indicate the temperature gradients and the direction thereof.

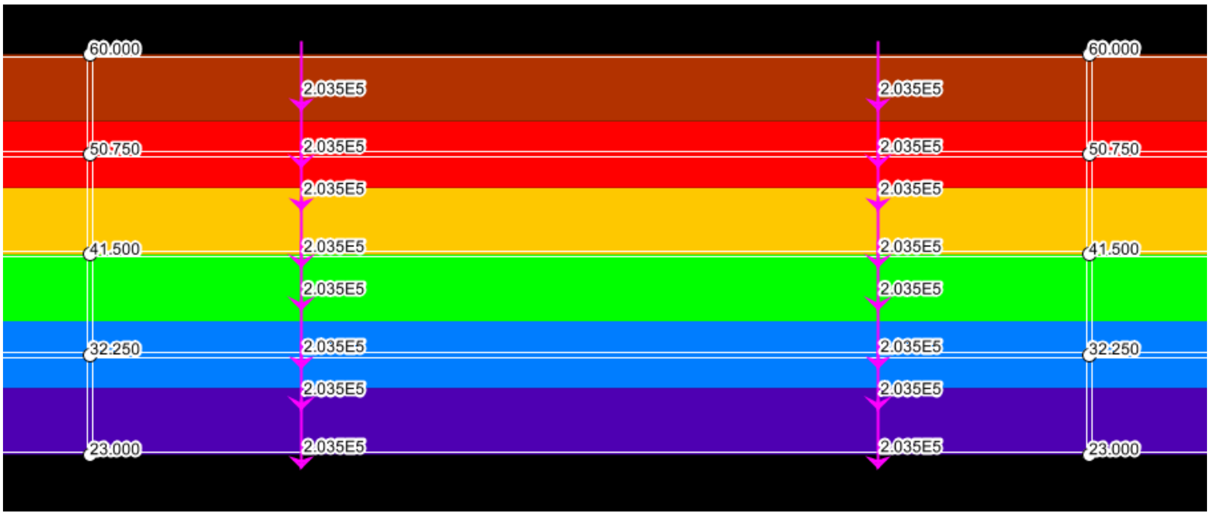


Figure 33: Section of the model for Stationary Heat Flow Example

Further investigation of this particular problem would involve using temperature differences in the above analysis in another finite element analysis which considers the thermal stresses which develop in the surfacing layer. The results of the thermal stress analysis will not be discussed in this study. However, it can be expected that there would be a significant difference in lateral displacement between the top and bottom of the surfacing layer. Figure 34 illustrates the deformed shape of the layer.

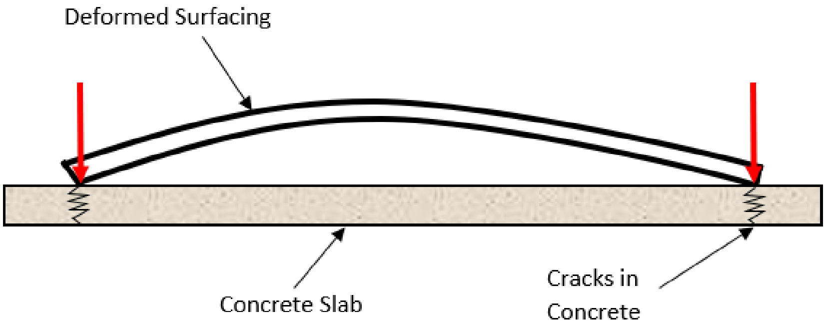


Figure 34: Deformed Shape of Surfacing Layer

Note from Figure 34 that the weight of the surfacing layer is no longer distributed along the concrete slab but now acts over a small area. This creates high levels of stress in the concrete and can be one of the explanations for the cracking in the concrete.

4.7 Example Using Elastic Plane Stress Model

4.7.1 Problem Statement

The case discussed her considers modelling of a cantilever slab which forms part of a restaurant in Stellenbosch. Figure 35 shows a diagrammatic illustration of the problem. The model considers a unit width of slab cast from concrete with Young's modulus and Poisson ratio provided in Figure 35.

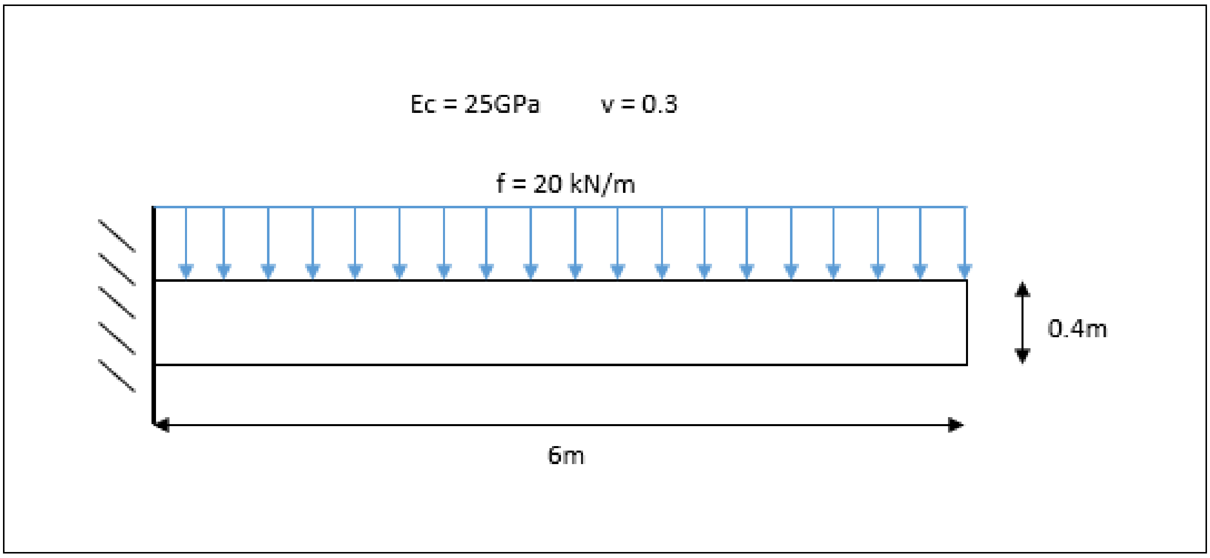


Figure 35: Diagram of Elastic Plane Stress Problem

4.7.2 Solution Outline

The input file to be used in conjunction with this solution is named “Cantilever.elas”. The first step in the solution using the analysis model is to develop a set of nodes which define the geometry of the cantilever section. This is done by creating *Node* objects which have a specified name, as well as the coordinates in the horizontal and vertical directions. These *Node* objects in union make up the Finite element geometry for the problem. Once this has taken place, an *ElasticMaterial* object can be specified in the input file to define the material properties of the cantilever section. *ElasticMaterial* objects contain a name, value of the Young’s modulus and lastly the Poisson ratio of the material. It is important to note that the default implementation of elastic plane analysis is the plane stress case. If the user wishes to perform a plain strain analysis, then the user must adjust the Young’s modulus and Poisson ratio according to literature discussed in this study (see equations 29.a and 29.b).

Now that the problem has defined the material properties and geometry, finite elements can be added to the model. Finite elements selected for the solution to this model are *ConstantStrainTriangle* objects. These objects require a name for the element, followed by the names of the nodes which must be defined in counter-clockwise direction and the material name.

Lastly, the boundary conditions for the model can be specified. The cantilever is considered to be fixed to the building and so it is necessary to define the prescribed displacements at the fixed connection. These displacements are specified using *PrimalBoundaryCondition* objects where the object is provided a name, the name of the node at which it occurs, followed by the degree of freedom in which the boundary condition is applied, and finally the value of the boundary condition. The degree of freedom of the boundary condition is provided as either the horizontal direction (indicated by “1” in the input file) or vertical direction (indicated by “2” in the input file). This is because plane stress and strain analysis have two degrees of freedom per node and as such the functionality has been provided to allow for either one or both degrees of freedom to be specified.

Following the displacement boundary conditions, the loading conditions can be specified in the input file. Loading on the structure is stated in the input file using *DualZeroDBoundaryCondition* objects. In order to evaluate the problem, the line load in the Figure must first be converted to point loads at the nodes which they occur. This is because the analysis model provides no implementation to define a loading as a function on the boundary. *DualZeroDBoundaryCondition* objects require a name, the name of the node at which the boundary condition occurs, a degree of freedom in which it is applied (specified in same way as for *PrimalBoundaryCondition*) as well as a value of the point load intensity.

4.7.3 Results from Analysis Model

The vertical displacements at the nodes from the results of the analysis can be shown using the "Graphical Display" from the "Display" menu. These results are illustrated in Figure 36 where the green vectors indicate the displacements and the direction. Note the maximum displacement at the end of the cantilever of 5.323 x 10^{-3} m.

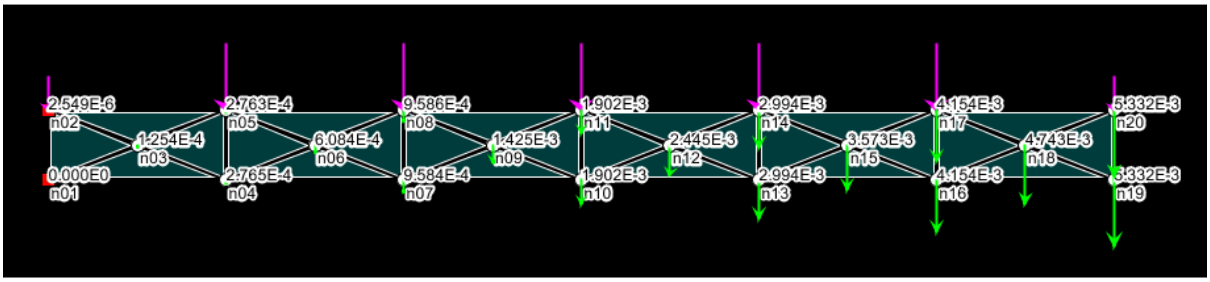


Figure 36: Vertical Displacements Determined from Analysis Model

As with any finite element analysis, the results must be evaluated and interpreted for reliability. The maximum displacement determined in the model can be compared to the maximum displacement using Euler-Bernoulli beam theory. The equation for maximum displacement using Euler-Bernoulli theory is given by equation 35.

$$\delta = \frac{wl^4}{8EI}$$

$$\delta = \frac{20000 * 6^4}{8 * 25 * 10^9 * \frac{1 * 0.4^3}{12}}$$

$$\delta = 24 * 10^{-3} m$$

It is quite clear that the results of the maximum vertical displacements differ quite considerably. However, these results can be explained by evaluating the mesh of the model. Consider a vertical section through node "n03" showing the horizontal strains illustrated in Figure 37.

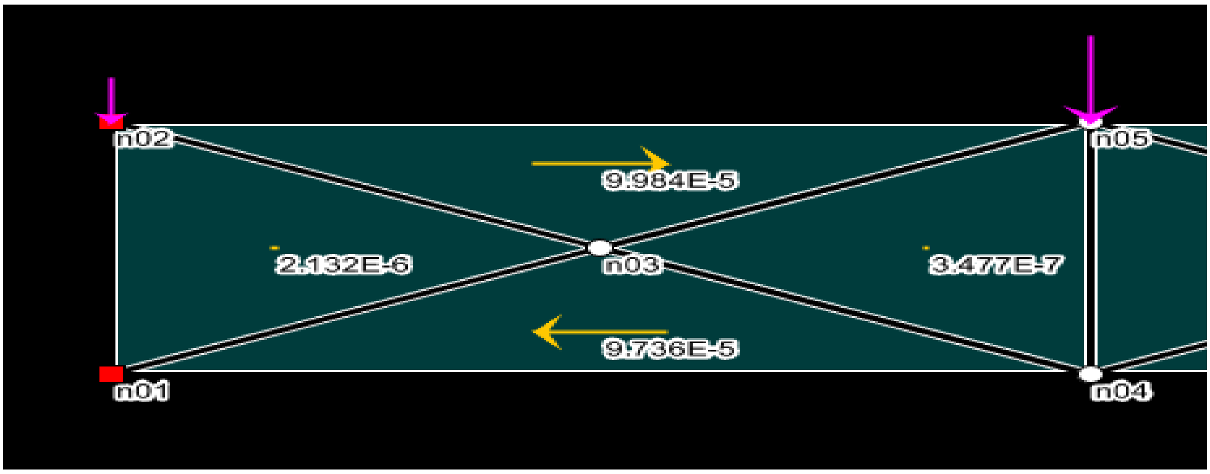


Figure 37: Snippet of Strains in Elements near Node "n03"

Figure 38 indicates that the strain distribution in the horizontal is uniform across the section considered. However, Euler-Bernoulli theory considers the strain to be linear across the section. A comparison of the *FemModel* and Euler-Bernoulli assumed strain distributions is shown in Figure 37.

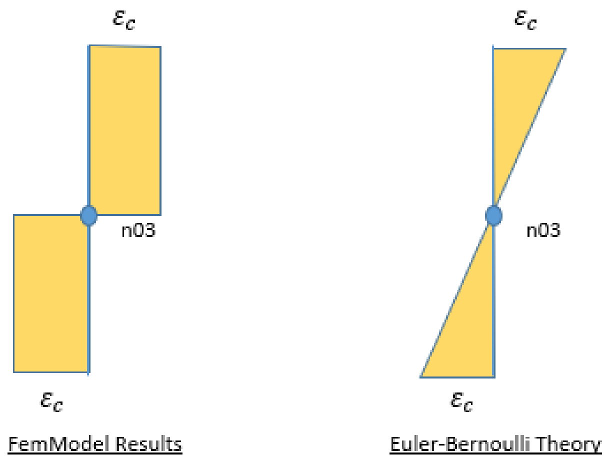


Figure 38: Comparison of Strain Distributions for Plane Stress Example

Figure 38 shows the results of the error due to the discretization of the model. Small errors such as this extending through the entire section model leads to highly unreliable results. A proposed solution to this problem would be to increase the number of rows of finite elements in the mesh to better approximate and represent the strains across the section. By adding more elements as shown in Figure 39, the strains may become more accurately represented by the model

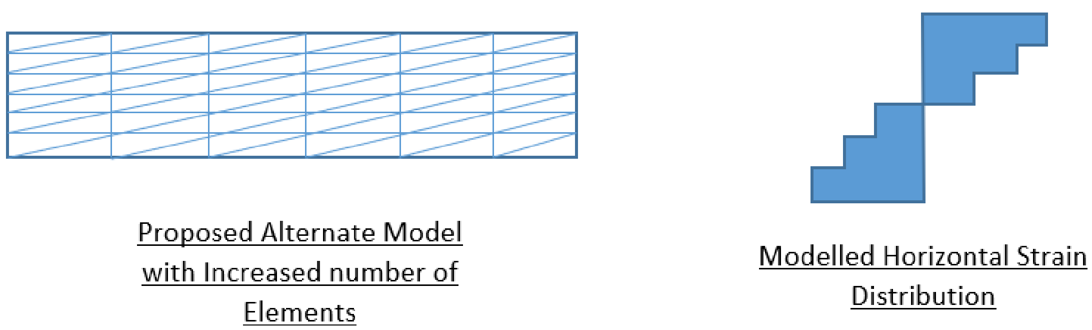


Figure 39: Proposed New FemModel of Cantilever

Another proposed solution would be to implement higher order elements in the mesh such as *PlaneQuad4Iso*. Using higher order elements such as *PlaneQuad4Iso* means that the strains in the model are no longer constant across an element and so the actual distribution is more accurately represented.

5 Chapter 5: Conclusions

5.1 Findings

5.1.1 Achievements

The results of this study confirmed that the primary aim was successfully achieved - which was to develop an analysis model to perform the investigated boundary value problems. This included the modification of the algorithm to implement finite element method problems which require multiple degrees of freedom per node. The study was also further able to successfully develop a comprehensive description of the data structure for the core framework of the analysis model. It is however important to note that the description of the core framework did not include explanations for all the classes and methods required to develop the implementation of the analysis model. This was due to the fact that the project had simply become too large to provide a complete comprehensive description for all classes given the time constraints of this study.

Similarly, the study was able to develop examples and explanations of how to use the analysis model. Furthermore, the study was able to partially develop Java documentation for the analysis model. The documentation is only completed for classes and packages that are particularly crucial to the understanding and use of the analysis model. Full documentation of the analysis model could not be developed due to the fact that the development would require too much time given the limitations of this study.

The study was not able to test or gauge if the additional documentation provided resulted in improvements to overall enthusiasm and development of students in Engineering Informatics 314. This is because the testing the improvements did not fall within the scope of the study. It is therefore with optimism that these documents of the can be used successfully by students of Engineering Informatics 314 to come.

5.1.2 Reflecting on the Graphical User Interface

The graphical user interface developed for the analysis model provides a very traditional way of creating a model and performing an analysis on the model. As mention in the core framework, the graphical user interface makes use of menus which the user selects from a list of options. However, a more contemporary approach may be to make use of buttons and control widgets within the analysis model window. Generally, buttons and widgets are advantageous because they are easily understandable by almost all people and provided a quick access solution compared to scrolling through a menu.

5.1.3 Model and User interaction

In any software development project is it vital to select an appropriate framework that fits the application and intended use of the software. In the case of this study, a particularly important constituent of the model's requirements was that the model should prevent intentional or unintentional errors from being made by the user. An approach which solves this dilemma is the model-view-controller (MVC) methodology. Figure 40 illustrates the pattern of approach to the development of the program.

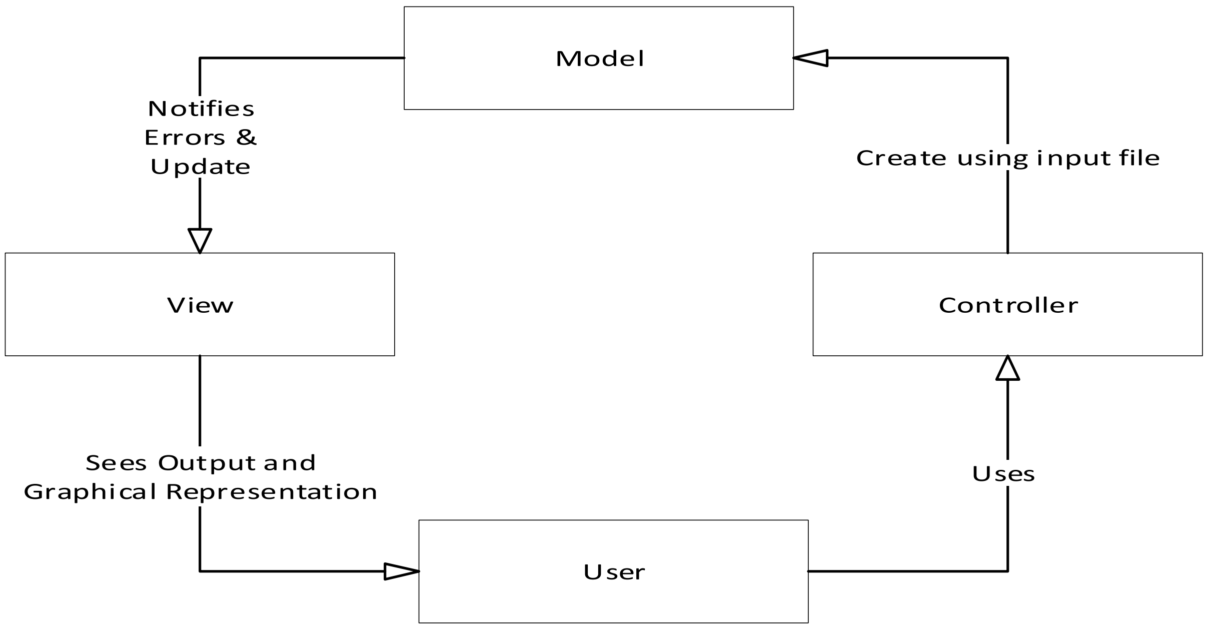


Figure 40: Model-View-Controller Approach

A significant advantage of using this approach is that the development of the *FemModel* data structure and graphical user interface are independent structures within the analysis model. This possesses significant benefits for future development and implementation in the analysis model. For example, Improvements to graphical user interface can occur concurrently to the improvements in the *FemModel* data structure or vice versa. Furthermore, the user cannot manipulate the finite element analysis through the graphical user interface. This protection of the analysis is important to ensure that the user does not intentionally or unintentionally make errors in the results. Moreover, future studies can consider implementing functionality whereby multiple *FemModel* objects can occur simultaneously in the analysis model. For the purpose of this study, it was presumed that implementing this functionality may seem confusing to first time users of the analysis model.

5.2 Limitations

There a serval notable limitations of the structure used for the developments and analysis of the *FemModel*. These limitations are highlighted in the sub sections of this section.

5.2.1 Input files are demanding to manage

First noted is significant limitations due to the method for constructing *Components* in the *FemModel*. As mentioned in the core framework, *Components* of *FemModel* are constructed through the use of parsing an input file. For larger models, this technique can become quite demanding for the user. It can quite easily be shown through the review of example input files that larger models require significant responsibility from the user. These responsibilities may include ensuring that input text has been specified and structured correctly. For example, a requirement of the text structure in the input file is that node names must be specified counter clockwise for finite elements in the model. If the user does not carefully review the input file, this error may go unnoticed.

5.2.2 Analysis Algorithm iterates through Nodes not Elements

Another significant limitation arises in the algorithm, namely *FirstOrderLinearAnalysis*, which performs the finite element analysis on the *FemModel*. The algorithm contains a method for determining the number of degrees of freedom in the system. This method contains an Iterator which iterates through all the *Node* objects the *FemModel's* registry. Therefore, if *Nodes* are specified in the model and go unused, the Iterator will still iterate these unused *Node* objects and specify their degrees of freedom in the system. The consequences of this that the analysis does not produce results. It is therefore imperative that the user correctly specifies the amount of node objects in the input file.

5.2.3 AbstractElement Objects do not know type Degrees of Freedom

As mentioned in the *FemModel* structure, *AbstractElement* objects contain variables which describe the number of degrees of freedom an element has as well as the number of degrees of freedom per node in the element. However, the *AbstractElement* objects do not know the type each degree of freedom is. This requires that the degrees of freedom at a node in the element must be specified in a predetermined order i.e. if it is a plane stress element, the first degree of freedom is always in the horizontal and second degree of freedom is always in the vertical. This means that if the user creates a model, it must be ensured that the model must only contain elements with the same number of degrees of freedom per node and same type of degrees of freedom at the each node.

5.2.4 Difficult to modify Components of FemModel

It was previously noted that the administration of the components of the FemModel provide plenty advantages, However, a limitation arises with the fact that the Components objects do not know anything about their sub components other than the name. This is a disadvantage if the user wishes to modify the model between the parsing of the input file and performing of analysis. The restriction can also be seen as a benefit since the current implementation of the analysis model specifically does not allow modifications to the model other than through parsing the input file.

5.3 Recommendations for Future Improvement

5.3.1 Extending the Documentation Reserve

The time constraints imposed on the study significantly limited the development of documentation for the analysis model. A decision was made at the start of the project to only develop documentation after the iterative development process for the analysis model was complete. This meant that the development of documentation was delayed since the source code had to be re-evaluated which took up some time. In hind sight, more Java documentation could have been produced if the development of the documentation coincided with the development of the classes and packages in the analysis model.

Further investigation may also include the development of additional examples using the Analysis model. This study only developed one example for each type of boundary value problem. It was therefore not possible to cover all aspects and implementations possible in the examples. This may raise learning concerns for the user. By developing more examples, it is predicted that any confusion or misinterpretations may be dealt with by reviewing further examples and explanations.

Therefore, it is proposed that in further research it may be useful to develop a larger documentation reserve whereby users have the opportunity to delve deeper into the workings of the analysis model and possible implement changes for the own benefits and uses.

5.3.2 Explore Extending Analysis Model for Other Types of Boundary Value Problems and Analyses

In this study the integration of the investigated boundary value problems into the analysis model was a success. This raises optimism for future expansion in the analysis model to allow for various other types of boundary value problems. Future research may explore adding additional types of boundary value problems, for example:

1. Electrostatics
2. Catchment area analysis of hydrological processes
3. River bed morphology and erosion processes
4. Sedimentation and scouring processes around coastal areas
5. Flow analysis in hydropower structures

It must be noted that the finite element algorithm used in this analysis model has been developed specifically for first order linear analysis of boundary value problems. Therefore, future research which wishes to uses the algorithm can only develop implementation of boundary value problems which make use of first order linear analysis assumptions. Therefore it is proposed that future research may also investigate the development of higher order or non-linear analyses which can be implemented into the analysis model.

5.3.3 Investigate Alternatives to the FemModel Structure

The structure used for the development of the *FemModel* includes many limitations as mentioned in this study. Future research can investigate different types of structures which can positively influence the user experience and the development of skills for proficient use and understanding of finite element method.

5.3.4 Add Higher Order Finite Elements to FemModel Structure

It is well known that improved interpolation using the finite element method can exist if there is a larger number of points at which the conditions is known. Furthermore, it is also well known that better interpolation can exist if there are more points through which interpolation can occur. It is therefore proposed that future implementation may investigate the implementation of higher order finite elements into the analysis model. Particularly useful elements may include the development of higher order isoparametric 6 node triangular elements or 8 node quadrilateral elements. These elements mentioned are particularly noted as they a specifically discussed in Engineering Informatics 314.

5.3.5 Introduce the concept of Degree of Freedom

As mentioned previously, the data structure does not define a specific object class which for degree of freedom. This means that there is currently no way to know what type of degree of freedom is used in each element. Therefore future investigations can explore implements the concept of a degree of freedom into the analysis model which may allow the user to add different types of element objects into the same model. In Elastic analyses for example, these new developments may allow the user to specify finite element objects with displacements in the same model with elements which contain displacements and rotations.

5.3.6 Provide Alternatives to Parsing Input File

The method of parsing an input file as it is currently used in the analysis model to develop components in this model is simple and elegant but can become quite tedious for larger models. It is therefore proposed that future research may explore new ways to generate objects for the analysis model. Most commercial finite element software makes use of computer aided drawing techniques to develop models. This technique however requires the developments of meshing algorithms. Future investigations may look at the development of a meshing algorithm which can be implemented into this analysis model

5.3.7 Develop a Contemporary Graphical User Interface

The development of the graphical user interface for the analysis model made use of Java Swing libraries. Swing applications still work on computers today but it is noted that the development of new swing infrastructure is scarce if not non-existent. Oracle, the developers of Java, have made it clear that they aim to replace the Swing with a more modern approach, namely JavaFX. Future research can explore the development of a graphical user interface using JavaFX and integrate the analysis model structure with the new interface.

5.4 Concluding Remarks

The analysis model developed clearly shows that it is possible to relate the finite element method to various type of physical problems in civil engineering. Further research may find more types of problems which may be suitable to the development of finite element method skills for students at undergraduate level. There is however plenty of room for improvement in the analysis model as illustrated in the findings of this report. With the development of new computer aided technologies, such as the development of the Java language, the finding of future research may greatly influence and improve on the structure of the analysis model. It is obvious that the resources developed herein can provide students of Engineering Informatics 314 with a much greater opportunity for understanding of the finite element method. Furthermore, the source code of the analysis model may provide students with useful insight into understanding the workings of commercial finite element software. It is therefore with great optimism that students at undergraduate level can greatly improve the skills and enthusiasm towards the finite element method through this work and future studies.

6 References

Bathe, K. J. (2014) *Finite Element Procedures*. Second. Watertown: K.J. Bathe.

Chadwick, A. (2013) *Hydraulics in Civil and Environmental Engineering*, *Journal of Hydraulic Engineering-asce - J HYDRAUL ENG-ASCE*. doi: 10.1061/(ASCE)0733-9429(2000)126:9(724).

Harr, M. E. (1963) *Groundwater and Seepage*, *Soil Science*. doi: 10.1097/00010694-196304000-00040.

Hrennikoff, A. (1968) 'The Finite Element Method in Application to Plane Stress'.

Knappett, J. A. and Craig, R. F. (2012) *Craig ' s Soil Mechanics*. Eighth. London: Sppon Press.

Logan, D. L. (2016) *A First Course in the Finite Element Method*. Fourth. Toronto: Thomson Canada Limited.

Mariño, M. A. and Luthin, J. N. (1982) *Seepage and Groundwater*, *Elsevier Scientific Publishing Company*. Amsterdam: Elsevier. doi: 10.1016/S0167-5648(08)70583-5.

Nikishkov, G. (2010) *Programming Finite Elements in Java*. London: Springer.

van Rooyen, G. (2017) 'Engineering Informatics 314 Object Modelling of Physical Problems'.

Wang, H. F. and Anderson, M. P. (1982) *Introduction to Groundwater Modeling*. San Diego: Academic Press.

White, F. (2010) 'Fluid Mechanics', *McGraw-Hill,New York*, p. 862. doi: 10.1111/j.1549-8719.2009.00016.x.Mechanobiology.

Zienkiewicz, O. C. and Taylor, R. L. (2000a) *The Finite Element Method*. Oxford: Butterworth-Heinemann.

Zienkiewicz, O. C. and Taylor, R. L. (2000b) *The Finite Element Method*. Oxford: Butterworth-Heinemann. doi: 10.1016/0022-460X(91)90505-E.