ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

SOFTWARE DEVELOPMENT FOR ANALYSIS OF PRIMARY CONSOLIDATION PROBLEMS AND ASSESSMENT OF ONE-DIMENSIONAL CONSOLIDATION THEORY

ANTENEH MASRESHA

June 2020

Addis Ababa

SOFTWARE DEVELOPMENT FOR ANALYSIS OF PRIMARY CONSOLIDATION PROBLEMS AND ASSESSMENT OF ONE-DIMENSIONAL CONSOLIDATION THEORY

BY

ANTENEH MASRESHA ZERIHUN

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirement for the Degree of Master of Science in Geotechnical Engineering

(ERA Sponsor)

Dr.-Ing. Henok Fikre
Thesis Advisor

Addis Ababa University
Addis Ababa

June 2020

Software Development for Analysis of Primary Consolidation Problems and Assessment of One-Dimensional Consolidation Theory

BY ANTENEH MASRESHA

A thesis submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirement for the Degree of Master of Science in Geotechnical Engineering

Approved by Board of Examiners:

1.	Dr-Ing Henok Fikre Advisor	[signature] Signature	27/10/2020 Date
2.	Dr-Ing Tensay Gebremedhin Internal Examiner	[signature] Signature	03/11/2020 Date
3.	Dr. Samuel Tadesse External Examiner	[signature] Signature	03/11/2020 Date
4.	Chairman	Mebruk Mohammed (Dr.-Ing.) Dean, School of Civil & Environmental Engineering Signature	Date

DECLARATION

I hereby declare that the research work titled "Software Development for Analysis of Primary Consolidation Problems and Assessment of One-Dimensional Consolidation Theory" is my work. Where material has been used from other sources, it has been duly acknowledged and due references have been provided on all supporting literature and resources.

Name: Anteneh Masresha

Signature: _______________

Place: Addis Ababa University

Date of Submission: June 2020

ACKNOWLEDGMENT

My deepest gratitude goes to my advisor Dr. Henok Fikre for his professional, genuine guidance, and valuable advice to accomplish this thesis on time.

I would like to extend my special appreciation to my Employer, Ethiopian Roads Authority, for financial support on my graduate study. I also seize this opportunity to express my profound heartfelt thanks to those people who have collaborated with me, especially my family and friends.

ABSTRACT

Terzaghi gave the theory for determination of the rate of consolidation of a saturated soil mass subjected to a static, steady load with some assumptions and the theory gives an estimate of the total time and the time rate of settlement of a structure with simple calculations. However, as Terzaghi's one-dimensional consolidation theory assumes the drainage of water occurs only in the vertical direction, it leads to an overestimation of the decay times of the settlement process and makes it questionable for two- and three-dimensional problems.

In this research, a Finite Difference Method solution is used to develop a computer program for analysis of two-dimensional consolidation problems easily. The exact solution of a consolidation problem is used to validate the developed computer program and found to be satisfactory and reasonably in good agreement. Then, a comparison between results from one-dimensional analysis and two-dimensional analysis is made using five different models.

From the comparison, it is shown that the errors from using Terzaghi's one–dimensional consolidation analysis for two-dimensional consolidation problems are significant, which gives error up to 60.0 % and 52.6 % when calculating the 50 % and 90 % consolidation times respectively for width to depth ratio of two.

Finally, this research thesis recommends conditions where a one-dimensional analysis could be implemented without a significant error.

Key Words*:2D Consolidation, Terzaghi's Theory, Finite Difference Method (FDM), Alternating Directional Implicit (ADI)*

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background

A study of soil consolidation due to additional pressure is extremely useful for predicting the magnitude and time of the settlement of the structure. With some assumptions, Terzaghi provided a theory for the evaluation of the rate of consolidation of fully saturated soil mass under a static load. (K.R.Arora, 2004) His theory is well known in the field of soil mechanics and has great importance for settlement analysis of many types of soils under structures. (Biot, 1941)

Even though, Terzaghi's theory gives an estimation of the rate of settlement with simpler computations for structures built on soil, some of the assumptions made by Terzaghi are not fully fulfilled in actual field problems. From those assumptions, the one stated as *"Drainage of water occurs only in the vertical direction"* is not based on an actual condition of the soil masses and expected to result in a larger error. (K.R.Arora, 2004) This assumption ignores the horizontal or radial drainage of water and limits the consolidation theory to only one-dimensional, hence leads to over-estimating the time for the settlement process. (Francesco, 2011)

Hence, using a one-dimensional consolidation analysis for field problems that have additional drainage conditions in addition to vertical drainage may lead to the aforementioned large error. (K.R.Arora, 2004) Examples of consolidation problems that do not belong to a one-dimensional problem include strip footing, dam, embankment, and axisymmetric loadings as in Circular tank loads. These consolidation problems are for cases that cannot be precisely approximated with the one-dimensional consolidation analysis. (Schmertmann, 1970) There have been lots of researches done to search the exact solutions of consolidation problems and these researches further highlighted this hypothesis. (Francesco, 2011)

Therefore, the question is that; if the errors are arising from using one-dimensional consolidation analysis while the problems are two – dimensional consolidation problems, why do not we use an appropriate analysis? The research tries to solve these issues by developing a computer program to extend the analysis to two-dimensional consolidation problems. Comparisons between the one-dimensional consolidation analysis and two-dimensional consolidation analysis are made for several cases and from these comparisons, this research thesis shows error from the one-dimensional analysis is significant for two-dimensional problems. Finally, the cases where two-dimensional analysis becomes mandatory is identified.

The computer program is then evaluated using previously studied exact solutions for consolidation problems.

In the end, user guideline is prepared for the developed software.

1.2 Objective of the Study

1.2.1 General Objectives

The general objectives of this research are:

- To investigate and show a one-dimensional consolidation analysis significantly overestimates the duration of primary consolidation.
- To develop a computer program for analysis of one-dimensional and two-dimensional consolidation problems.

1.2.2 Specific Objectives

- To explore Finite Difference solutions for one- and two-dimensional consolidation problems from the governing differential equation and use the appropriate method that suits for a computer program.
- To design a user-friendly Graphical User Interface (GUI) and develop an algorithm for the selected finite-difference solution.
- To make comparisons between the results obtained from two-dimensional analysis and one-dimensional analysis.
- To provide user guidelines for the developed software.

1.3 Statement of the Problem

Consolidation problems are being analyzed with a well-known Terzaghi's one-dimensional consolidation theory. However, the assumptions that are the basis for the theory leads to a large error because the differential equation for the one-dimensional consolidation overestimates the decay times by limiting the drainage options of the phenomenon. The error becomes significantly large for two- and three-dimensional problems like in embankment fill, earth fill dam and strip footings. Therefore, since the one-dimensional consolidation assumptions are not the best representative of the reality, the analysis for abovementioned problems has to be incorporated and shall be extended to two- and three-dimensional fields.

The main challenge in analyzing two- and three- dimensional consolidation problem is the complexity and difficulty of the equations and solution methods. Furthermore, the shortage in accessible computer tools is also the main factor to use the one-dimensional analysis with its limitation.

Therefore, this research is initiated observing the possible problems stated above and a passion to give a solution to overcome those problems.

1.4 Method

To show the significance of the error from a one-dimensional consolidation analysis a comparison is made by analyzing a two-Dimensional consolidation problem using Terzaghi's one-dimensional consolidation theory and using the newly developed software which is based on Finite Element Method is undertaken.

Microsoft Visual Studio is used to write, compile, and test the codes. The first task performed in solving the governing equation using a numerical solution is to approximate the second-order derivatives by a finite difference method.

Among several methods of finite difference method, the Alternating Directional Implicit (ADI) Method which is an implicit solution, convergent, and unconditionally stable method is used in this research.

To validate the developed software, a comparison between Terzaghi's exact solution and the software output for the average degree of consolidation, U versus the dimensionless time factor, Tv is done and close results were found from the Finite Difference Solution of the software.

Comparison between the results obtained from the software for two-dimensional consolidation analysis and one-dimensional consolidation analysis is performed to show the need for the two-dimensional analysis. Several tests were run to compare a one-dimensional analysis with two-dimensional analysis by varying geometry of the soil layer and soil parameters. From the test results, the effect of these variable parameters on the significance of the errors on one-dimensional analysis is observed.

Finally, the best scenario that a one-dimensional analysis could be used without significant error is recommended.

1.5 Content of the Thesis

This thesis has five chapters. The first chapter gives the basic introduction to the research, describes the structure of the thesis, and outlines the scope and objectives of the research. The basic principles of primary consolidation are given in the second chapter along with a detailed review of previous researches done to derive the solutions of consolidation equations.

The methodology and development of an algorithm for the computer program are presented in the third chapter, where detailed discussions are made regarding the development of the Software.

Chapter four considers the output of the software for various cases with different soil geometries and coefficients of consolidation to do detailed comparisons and discussions between results obtained from a one- and two-dimensional consolidation analysis approach. Finally, chapter five closes this research thesis by making conclusions and recommendations.

1.6 Applications and Limitations

The study is limited to a primary consolidation analysis of the soil. Strain in the horizontal axis is not considered in this research and settlement of the soil due to consolidation is assumed to take place only in the vertical direction as stated in Terzaghi's one-dimensional consolidation theory. However, in 2D consolidation which is a plane strain problem, strain in the horizontal axis may occur.

Furthermore, this work does not include analysis of multi-layered soil. It has been presumed that the existence of multi-layered soil does not affect the comparison result between one-dimensional and two-dimensional analysis. Hence, a homogeneous soil layer is considered throughout the research.

2 LITERATURE REVIEW

2.1 General

One of the main problems in the field of geotechnical engineering is the settlement of a foundation. A settlement of foundation occurs when the underneath soil moves downward under the weight of the structure. A significant amount of this downward movement of the foundation soil can make a structure above it unserviceable. (Budhu, 2011)

Among different causes of a settlement of a built structure, compression of soils is the common one. The compression of soils occurs due to compression of solid particles, compression of voids and expulsion of air and water in the voids. The compression of solid and water in the void is extremely small and can be neglected. Hence, the compression of soil occurs mainly due to the expulsion of water in voids and compression and expulsion of air voids. Therefore, if a fully saturated soil is considered, the compression of soil occurs mainly due to the expulsion of pore water. (K.R.Arora, 2004) This expulsion of pore water is called Consolidation and it is one of the causes for the reduction of volume of saturated soils under pressure. The total reduction of volume of soil usually consists of three parts; immediate/elastic compression, primary consolidation, and secondary compression. (Budhu, 2011)

i. Immediate or elastic Compression is the soil's compression without any change in the water content and it is caused by the elastic deformation the soil. if the elastic modulus of the soil layer is known, this part of compression is calculated using equations derived from the elastic theory.

ii. Primary Consolidation, which is the focus of this research, is caused by the expulsion of water from the voids and, consequently, the load from the excess pore water pressure is transferred to the soil particles. (Budhu, 2011) Primary consolidation takes a significant portion of the compression of the soil and takes a long time especially in fine-grained soils.

    In this thesis, the term consolidation refers to the primary consolidation as defined here.

iii. Secondary Compression is caused by the plastic adjustment of soil fabrics and it is a continuation of the reduction in the volume of soil after the excess hydrostatic pressure developed by the applied pressure is fully dissipated. It takes place after the primary consolidation is complete and usually occurs at a much slower rate. (R.Butterfield, 2005)

Since primary consolidation takes a significant amount from the total reduction of volume of soil mass, a study of consolidation characteristics is extremely useful to precisely predict the time and magnitude of the settlement of the structure. (K.R.Arora, 2004)

The main parameters that affect the rate of consolidation for a homogeneous soil are the permeability (hydraulic conductivity) of the soil, the thickness of the consolidating soil layer, and the boundary drainage conditions.

i. Permeability (Hydraulic conductivity) is a measure of the soil's property to transmit water under a hydraulic gradient. An increase in permeability of the soil, keeping other factors constant, results in a faster rate of consolidation since it leads to quick dissipation of pore pressures from the soil.

ii. Thickness of soil is the full depth of the consolidating soil layer. An increase in layer thickness makes the consolidation process occur at a lower rate because a larger layer thickness has a lesser hydraulic head gradient during the expulsion of pore water. Besides, a soil with larger layer thickness also has more volume of water that needs to be expelled.

iii. Boundary Conditions is the amount of drainage boundary for water to be expelled. A soil with increased drainage faces will have a faster rate of consolidation settlement since the expulsion of water will proceed at a faster rate. (Budhu, 2011)

From these factors, by assuming the problem as one-dimensional consolidation in the analysis we are neglecting the horizontal drainage faces and this increases the time required to drain the initial excess pore water. In research conducted by Di Francesco, it is concluded that a one-dimensional consolidation analysis overestimates the decay times of the phenomenon and recommended to be extended to two- and three-dimensional fields. (Francesco, 2011)

Furthermore, Arora in his Soil mechanics book pointed out that Terzaghi's assumption regarding the drainage path in one-dimensional consolidation is a cause of the large error. (K.R.Arora, 2004)

To overcome this, several studies have been done by different researchers, among these studies, an equation in polar coordinates is developed by Carillo for three-dimensional consolidation. The equation is developed using the principle of superposition by considering an aggregated result from the vertical and radial consolidation. (Carrillo, 1942)

Extending this equation a two- and three-dimensional consolidation in Cartesian coordinates is derived by *Tewatia* which is practicable when drainage of water is allowed in horizontal and vertical directions. (Tewatia S. K., 2013)

There are also researches in the development of numerical and analytical solutions for solving two- and three-dimensional consolidation problems. Haase, Exner, and Reichel together developed two-dimensional numerical modeling of consolidation to solve two-dimensional consolidation problems. (Haase, Mario, & Uwe Reichel) In their model the three-dimensional approach is reduced to two-dimensional radial axisymmetry during the algorithm (numerical operation). The model has been validated by comparing with field experiments and analytical solutions. (Haase, Mario, & Uwe Reichel)

These all researches give a good insight for the development of software for 2D and 3D consolidation analysis to overcome errors due to an assumption made by Terzaghi's one-dimensional consolidation theory and extend the analysis to two- and three-dimensional solution.

2.2 One Dimensional Consolidation Theory

2.2.1 Consolidation Equation

It is possible to derive the theory of one-dimensional consolidation for the time rate of settlement using an element of the soil sample of thickness dz and cross-sectional area $dA = dx \cdot dy$ as shown in Figure 2-1.

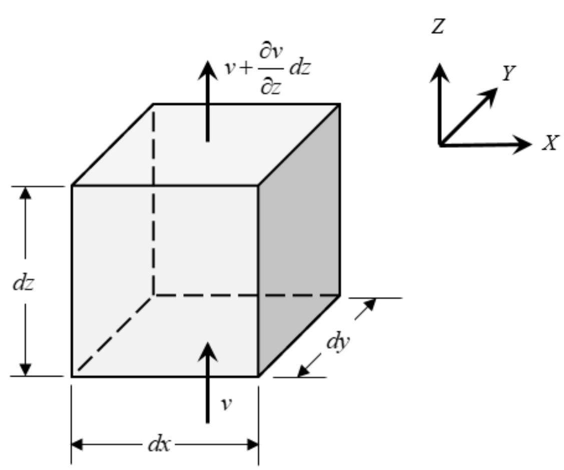


Figure 2-1: One-dimensional flow through a soil element (Budhu, 2011)

The following points are assumed in the derivation of the equation:

a) The soil is fully saturated and homogeneous.
b) Darcy's law is valid i.e. discharge is directly proportional to the hydraulic gradient and the permeability (hydraulic conductivity).
c) The soil particles and the water in the voids are incompressible, i.e. the consolidation takes place due to a flow of water.
d) Flow occurs only in a vertical axis.
e) The strains are negligibly small.

The following observations are also to be used in the equation:

i. The change in volume of soil, ΔV, is equal to the change in volume of pore water dissipated, ΔV_w, which is the change in the volume of the voids, ΔV_v for fully saturated soils. Since the soil is laterally constrained, the area of the soil is constant and, thus, the change in volume is directly proportional to the change in height.

ii. In addition, since the total applied stress remains constant, any drop in the initial excess pore water pressure needs to be balanced by an equal rise in effective stress. Therefore, the change in vertical effective stress, at any depth, is equal to the change in excess pore water pressure at that depth. That is, $\partial\sigma_z^{'} = \partial u$.

The flow rate is the product of the velocity and the cross-sectional area normal to its direction. For our soil element in Figure 2-1, the inflow of water is $v\ dA$ and the outflow over the elemental thickness dz is $[v + (\partial v/\partial z)dz]dA$. The change in flow is then $(\partial v/\partial z)\ dz\ dA$. The rate of change in the volume of water expelled, which is equal to the rate of change of volume of the soil, must equal the change in flow. That is,

$$\frac{\partial V}{\partial t} = \frac{\partial v}{\partial z} dzdA \tag{2-1}$$

But, the volumetric strain $\varepsilon_p = \partial V/V = \partial e/(1+e_0)$, where e_0 is the initial void ratio, and therefore

$$\partial V = \frac{\partial e}{1+e_o} dzdA = m_v \partial u dzdA \tag{2-2}$$

Substituting Equation (2-2) into Equation (2-1) and simplifying, we obtain

$$\frac{\partial v}{\partial z} = \frac{\partial u}{\partial t} m_v \qquad (2\text{-}3)$$

The one-dimensional flow of water from Darcy's law is

$$v = k_z i = k_z \frac{\partial h}{\partial z} \qquad (2\text{-}4)$$

Where k_z is the hydraulic conductivity in the vertical direction.

Partial differentiation of equation (2-4) with respect to *z* gives

$$\frac{\partial v}{\partial z} = k_z \frac{\partial^2 h}{\partial z^2} \qquad (2\text{-}5)$$

But, the pore water pressure at any time *t* is

$$u = h\gamma_w \qquad (2\text{-}6)$$

Where *h* is the equivalent height of water in the piezometer.

Partial differentiation of the above equation (2-6) with respect to *z* gives

$$\frac{\partial^2 h}{\partial z^2} = \frac{1}{\gamma_w} \frac{\partial^2 u}{\partial z^2} \qquad (2\text{-}7)$$

By substitution of Equation (2-7) into Equation (2-5), we get

$$\frac{\partial v}{\partial z} = \frac{k_z}{\gamma_w} \frac{\partial^2 u}{\partial z^2} \qquad (2\text{-}8)$$

Equating Equation (2-3) and Equation (2-8), we obtain

$$\frac{\partial u}{\partial t} = \frac{k_z}{m_v \gamma_w} \frac{\partial^2 u}{\partial z^2} \qquad (2\text{-}9)$$

We can replace $\frac{k_z}{m_v \gamma_w}$ by a coefficient C_v called the coefficient of consolidation.

The units for C_v are $length^2 / time$, for example, $cm^2 / \min$. Rewriting Equation (2-9) by substituting C_v, we get the general equation for one-dimensional consolidation as

$$\frac{\partial u}{\partial t} = C_v \frac{\partial^2 u}{\partial z^2} \tag{2-10}$$

This equation describes the spatial variation of excess pore water pressure (Δu) with time (t) and depth (z). (Budhu, 2011)

2.2.2 Solution of Governing Consolidation Equation Using Fourier series

The solution of any differential equation requires a knowledge of the boundary conditions. By specification of the initial distribution of excess pore water pressures at the boundaries, we can obtain solutions for the spatial variation of excess pore water pressure with time and depth. Various distributions of pore water pressures within a soil layer are possible. One of these is a uniform distribution of initial excess pore water pressure with depth. This may occur in a thin layer of fine-grained soils. The other is a triangular distribution that may occur in a thick layer of fine-grained soils.

The boundary conditions for double drainage condition with uniform distribution of initial excess pore water pressure are:

- When $t = 0, \Delta u = \Delta u_0 = \Delta \sigma_z$
- At the top boundary, $z = 0, \Delta u = 0$.
- At the bottom boundary, $z = 2H_{dr} \Delta u = 0,$ where H_{dr} is the length of the drainage path.

A solution for the governing consolidation equation, Equation (2-10), which satisfies these boundary conditions, is obtained using the Fourier series,

$$\Delta u(z,t) = \sum_{m=0}^{\infty} \frac{2\Delta u_o}{M} \sin\left(\frac{Mz}{H_{dr}}\right) \exp(-M^2 T_v) \tag{2-11}$$

Where $M = (\pi / 2)(2m+1)$ and m is a positive integer with values from 0 to ∞ and

$$T_v = \frac{C_v t}{H_{dr}^2} \tag{2-12}$$

Where T_v is known as the time factor; it is a dimensionless term.

A plot of Equation (2-11) gives the variation of excess pore water pressure with depth at different time. At time $t = 0$ $(T_v = 0)$, the initial excess pore water pressure, Δu_o, is equal to the applied vertical stress throughout the solid layer. As soon as drainage occurs, the initial excess pore water pressure will immediately fall to zero at the permeable boundaries. The maximum excess pore water pressure occurs at the center of the soil layer because the drainage path there is the longest

At time $t > 0$, the total applied vertical stress increment $\Delta\sigma_z$ at a depth z is equal to the sum of the vertical effective stress increment $\Delta\sigma_z$ and the excess pore water pressure Δu_z. After considerable time $(t \to \infty)$, the excess pore water pressure decrease to zero and vertical effective stress increment becomes equal to the vertical total stress increment.

The degree of consolidation or consolidation ratio, U_z, can be expressed mathematically as

$$U_z = 1 - \frac{\Delta u_z}{\Delta u_o} = 1 - \sum_{m=0}^{\infty} \frac{2}{M} \sin\left(\frac{Mz}{H_{dr}}\right) \exp(-M^2 T_v) \qquad (2\text{-}13)$$

At the beginning of the consolidation, the degree of consolidation is equal to zero at each point in the soil mass, $\Delta u_z = \Delta u_o$ but gradually rises to one when the initial excess pore water pressure dissipates.

For geotechnical problems, the average degree of consolidation, U, of the full soil layer at a particular time is more concerning than the consolidation at a particular depth.

Mathematically, the average degree of consolidation can be expressed from the solution of the one-dimensional consolidation equation (2-13) as

$$U = 1 - \sum_{m=0}^{\infty} \frac{2}{M^2} \exp(-M^2 T_v) \qquad (2\text{-}14)$$

Using a curve fitting of Tv vs U curve, a suitable set of equations for double drainage can be found as follows:

$$T_v = \frac{\pi}{4}\left(\frac{U}{100}\right)^2 \qquad \text{for } U \geq 60\% \qquad (2\text{-}15)$$

And $$T_v = 1.781 - 0.933\log(100 - U) \qquad \text{for } U \geq 60\% \qquad (2\text{-}16)$$

2.2.3 Finite Difference Solution of the Governing Consolidation Equation

Numerical methods (finite difference, finite element, and boundary element) provide approximate solutions to differential and integral equations for boundary conditions in which analytical solutions are not possible. Here the finite difference method is used to find a solution to the consolidation equation because it involves only the expansion of the differential equation using Taylor's theorem and can easily be adapted for computer applications. (Budhu, 2011)

Using Taylor's theorem,

$$\frac{\partial u}{\partial t} = \frac{1}{\Delta t}(u_{i,j+1} - u_{i,j}) \tag{2-17}$$

and

$$\frac{\partial^2 u}{\partial z^2} = \frac{1}{(\Delta z)^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) \tag{2-18}$$

Where (i, j) denotes a nodal position at the intersection of row i and column j. Columns represent time divisions and rows represent soil depth divisions. The assumption implicit in Equation (2-17) is that the excess pore water pressure between two adjacent nodes changes linearly with time. This assumption is reasonable if the distance between the two nodes is small. Substituting Equations (2-17) and (2-18) in the governing consolidation equation (2-10) and rearranging, we get

$$u_{i,j+1} = u_{i,j} + \frac{C_v \Delta t}{(\Delta z)^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) \tag{2-19}$$

Equation (2-19) is valid for nodes that are not boundary nodes. For example, at an impermeable boundary, no flow across it can occur and, consequently, $\partial u / \partial z = 0$, for which the finite difference equation is

$$\frac{\partial u}{\partial z} = 0 = \frac{1}{2\Delta z}(u_{i-1.j} - u_{i+1.j}) = 0 \tag{2-20}$$

And the governing consolidation equation becomes

$$u_{i,j+1} = u_{i,j} + \frac{C_v \Delta t}{(\Delta z)^2}(2u_{i-1,j} - 2u_{i,j}) \tag{2-21}$$

One has to establish the initial excess pore water pressure at the boundaries to determine how the pore water pressure is distributed within a soil at a given time. Then, estimate the variation of the initial excess pore water pressure within the soil. One may, for example, assume a linear distribution of initial excess pore water pressure with depth if the soil layer is thin or has a triangular distribution if a soil layer is thick.

2.3 Two and Three Dimensional Consolidation Theory

2.3.1 Equation of 2D and 3D consolidation in Cartesian Coordinates

Terzaghi's one-dimensional consolidation theory can be extended to a two-dimensional and three-dimensional consolidation equation for some cases, such as in sand drains and long embankments, where there is a significant horizontal or radial drainage, in addition to the vertical drainage. The equation for three-dimensional consolidation is derived considering the following assumptions.

i. The soil mass is homogenous
ii. The soil is fully saturated
iii. The soil particles as well as the water in the voids are incompressible, i.e. the consolidation takes place due to reduction in voids caused by flow water.
iv. Darcy's law, in generalized form, is applicable to anisotropic soils.
v. Pressure increment is applied instantaneously to develop an initial excess pore water pressure u_i.

Figure 2-2 shows a parallelepiped of soil mass with sides dx, dy, dx.

The volume of water entering the parallelepiped per unit time (Q_i) is obtained from the products of the relative velocities and area. Thus

$$Q_i = \left(v_x - \frac{\partial v_x}{2} . \frac{dx}{2} \right) dydz + \left(v_y - \frac{\partial v_y}{\partial z} . \frac{dy}{2} \right) dxdz + \left(v_z + \frac{\partial v_z}{\partial z} . \frac{dz}{2} \right) \times dxdy \qquad \text{(2-22)}$$

Likewise, the volume of water going out per unit time (Q_0) is given by

$$Q_0 = \left(v_x + \frac{\partial v_x}{2} . \frac{dx}{2} \right) dydz + \left(v_y + \frac{\partial v_y}{\partial z} . \frac{dy}{2} \right) dxdz + \left(v_z + \frac{\partial v_z}{\partial z} . \frac{dz}{2} \right) \times dxdy \qquad \text{(2-23)}$$

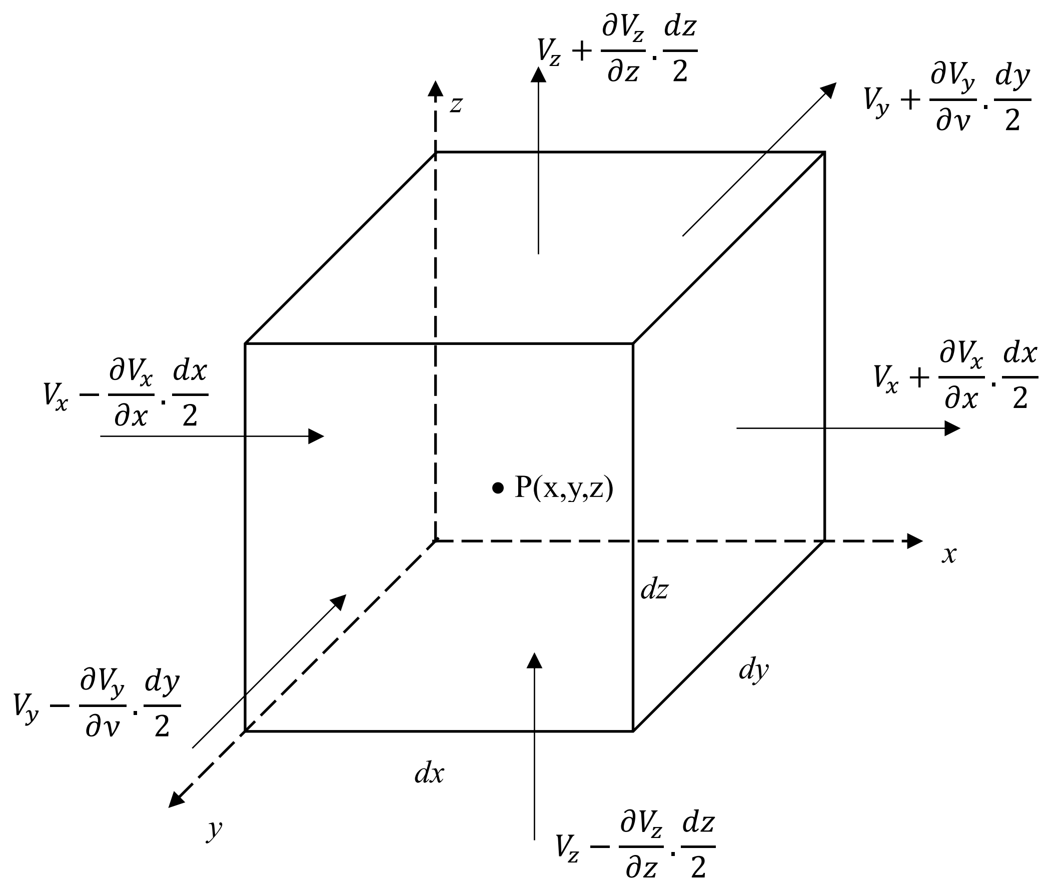


Figure 2-2: A parallelepiped of soil mass with sides dx, dy, dx

Therefore, the volume of water squeezed out of the parallelepiped per unit time is given by

$$dQ = Q_0 - Q_i$$

Or

$$dQ = \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right) dxdydz \tag{2-24}$$

The volume of the parallelepiped V is equal to $dxdydz,$ it is also equal to $V_s(1+e)$, where V_s is the volume of the solid e is the void ratio.

Thus

$$
\begin{aligned}
&V_S = \frac{V}{1+e} = \frac{dxdydz}{1+e} \\
&V = V_S(1+e) \\
&\frac{\partial V}{\partial t} = \frac{\partial}{\partial t}\left[V_S(1+e)\right] = V_S\,\frac{\partial e}{\partial t} \\
&\frac{\partial V}{\partial t} = \frac{dxdydz}{1+e}.\frac{\partial e}{\partial t}
\end{aligned}
\tag{2-25}
$$

The volume of water squeezed out per unit time is equal to the change in volume of parallelepiped per unit time. From Eqs. (2-24) and (2-25),

$$\frac{dxdydz}{1+e}.\frac{\partial e}{\partial t} = \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right)dxdydz$$

Or

$$\frac{\partial e}{\partial t} = (1+e)\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right) \tag{2-26}$$

If $\bar{u}$ is the excess pore water pressure, the velocities in x, y, and z directions are obtained from Darcy's law as:

$$
\begin{aligned}
&v_x - k_x i_x - k_x\,\frac{\partial h}{\partial x} = k_x\,\frac{1}{\gamma w}.\frac{\partial \bar{u}}{\partial x} \\
&v_y - k_y i_y - k_x\,\frac{\partial h}{\partial y} = k_y\,\frac{1}{\gamma w}.\frac{\partial \bar{u}}{\partial y} \\
&v_z - k_z i_z - k_z\,\frac{\partial h}{\partial z} = k_z\,\frac{1}{\gamma w}.\frac{\partial \bar{u}}{\partial z}
\end{aligned}
\tag{2-27}
$$

Substituting the above velocities in Eq. (2-26).

$$\frac{\partial e}{\partial t} - \frac{1+e}{\gamma_w}\left(k_x\,\frac{\partial^2 \bar{u}}{\partial x^2} + k_y\,\frac{\partial^2 \bar{u}}{\partial y^2} + k_z\,\frac{\partial^2 \bar{u}}{\partial z^2}\right) \tag{2-28}$$

As soon as the pressure increment $(\Delta\sigma)$ is applied, the pore water pressure develops. Initially, the load is entirely taken by pore water, but as time passes, water is squeezed out. The excess pore water pressure gradually decreases and the effective stress increases, as in the one-dimensional consolidation, Thus, $$\Delta\sigma = \Delta\bar{\sigma} + \Delta\bar{u} \qquad (2\text{-}29)$$

Where $\Delta\bar{\sigma} =$ effective stress, and $\Delta\bar{u} =$ pore water pressure.

As any increase in effective stress $(\bar{\sigma})$ is equal to a decrease in excess hydrostatic pressure $\bar{u}$,

$$\Delta\sigma = -\Delta\bar{u}$$

Therefore, $$\frac{\partial e}{\partial\bar{\sigma}} = -\frac{\partial e}{\partial\bar{u}} \qquad (2\text{-}30)$$

But $\frac{\partial e}{\partial\bar{\sigma}}$ = coefficient of compressibility, a_v

Therefore, $\frac{\partial e}{\partial\bar{\sigma}} = -a_v$

From the rule of partial differentiation,

$$\frac{\partial e}{\partial t} = \frac{\partial e}{\partial\bar{u}}.\frac{\partial\bar{u}}{\partial t} = a_v.\frac{\partial\bar{u}}{\partial t} \qquad (2\text{-}31)$$

From Eqs. (2-28) and (2-31),

$$a_v\frac{\partial\bar{u}}{\partial t} = \frac{1+e}{\gamma_w}\left(k_x\frac{\partial^2\bar{u}}{\partial x^2} + k_y\frac{\partial^2\bar{u}}{\partial y^2} + k_z\frac{\partial^2\bar{u}}{\partial z^2}\right)$$

Or $$\frac{\partial\bar{u}}{\partial t} = \frac{1+e}{a_v}.\frac{1}{\gamma_w}\left(k_x\frac{\partial^2\bar{u}}{\partial x^2} + k_y\frac{\partial^2\bar{u}}{\partial y^2} + k_z\frac{\partial^2\bar{u}}{\partial z^2}\right) \qquad (2\text{-}32)$$

Equation (2-32) can be written in terms of the coefficient of volume change m_v,

$$m_v = \frac{a_v}{1+e}$$

Thus $$\frac{\partial\bar{u}}{\partial t} = \frac{1}{m_v\gamma_w}\left(k_x\frac{\partial^2\bar{u}}{\partial x^2} + k_y\frac{\partial^2\bar{u}}{\partial y^2} + k_z\frac{\partial^2\bar{u}}{\partial z^2}\right) \qquad (2\text{-}33)$$

The equation can be written in terms of coefficients of consolidation C_{hx}, C_{hy} and C_v in the three directions using the relation $C_v = \frac{k}{m_v\gamma_w}$

Therefore, $$\frac{\partial \bar{u}}{\partial t} = C_{vx}\frac{\partial^2 \bar{u}}{\partial x^2} + C_{vy}\frac{\partial^2 \bar{u}}{\partial y^2} + C_{vz}\frac{\partial^2 \bar{u}}{\partial z^2} \tag{2-34}$$

Eq. (2-34) is the general equation for three-dimensional consolidation

Likewise, the general equation for two-dimensional consolidation with drainage in two axes can be expressed as:

$$\frac{\partial \bar{u}}{\partial t} = C_{vy}\frac{\partial^2 \bar{u}}{\partial y^2} + C_{vz}\frac{\partial^2 \bar{u}}{\partial z^2} \tag{2-35}$$

2.3.2 Solution of general 2D and 3D consolidation equation in Cartesian coordinates

Using a principle of superposition, the solution of a general two-dimensional and three-dimensional consolidation equation can be derived from the solution of one-dimensional consolidation. The solution of Equation (2-34) can be computed by allowing the drainage from the soil in X and Y directions in addition to the vertical axis Z from a parallelepiped sample shown in Figure 2-2 with settlement allowed in the vertical direction only.

The principle of superposition is a method that can be applied for all linear systems to compute the net response of the system due to two or more factors. It states that the net response of a system for two or more factors is the sum of the individual responses due to each factor as if acting on the system separately. (Widipedia, n.d.) Therefore, using the principle of superposition, the rate of settlement for multi-dimensional consolidation can be derived easily with simple arithmetic of individual rate of settlements of each axis. An example stated by S.K. Tawatia is a good description of the superposition principle. It states that, Assume a tank with two drainage pipes mounted on it, with discharge due to pipe one, Q_1, and discharge due to pipe two, Q_2, then the total discharge due to pipe-1 and pipe-2, Q is equal to Q_1+Q_2. At any moment, the rate of dropping of water level of the tank is directly proportional to the discharges from the pipes. Suppose, at any instance, δ is the water level in the tank, $\left(\frac{d\delta}{dt}\right)_1$ is the rate of dropping of water level due to pipe-1 and $\left(\frac{d\delta}{dt}\right)_2$ is the rate of lowering of the water level due to pipe-2, Then the resulting rate of settlement due to both pipes is

$$\left(\frac{d\delta}{dt}\right)_{1+2} = \left(\frac{d\delta}{dt}\right)_{1} + \left(\frac{d\delta}{dt}\right)_{2} \tag{2-36}$$

The same analogy works for consolidation since the rate of settlement of a soil mass is proportional to the discharge rate of water from the soil. The discharge rate of water from the soil in turn is the summation of discharge rates due to vertical and horizontal drainages. Hence, the resultant rate of settlement in two- and three-dimensional consolidation will be the summation of individual rates of settlements due to horizontal and vertical drainages. (Tewatia S. K., 2015)

Figure 2-3 shows settlement versus the rate of settlement data of clay soil under vertical pressure from 100–200 kPa taken from research work by Tewatia, on Equation of three-dimensional consolidation in cartesian coordinate system.

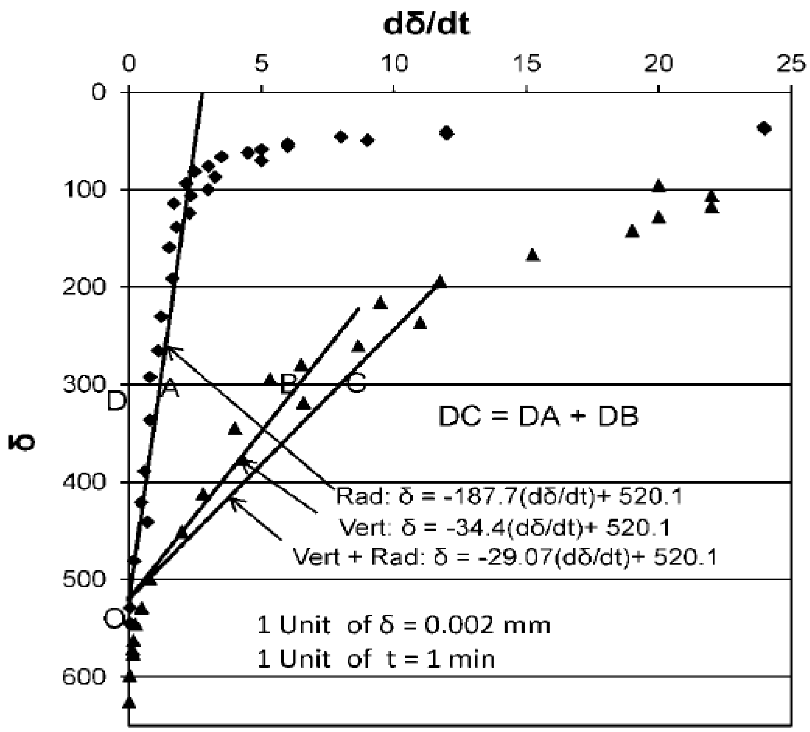


Figure 2-3: Settlement versus rate of settlement plot fora clay soil (Tewatia S. K., 2013)

In Figure 2-3, curve OC is a combined vertical and radial consolidation that is drawn by summing the horizontal compnent (rate of settlements) of curves OA and OB. (Tewatia S. K., 2013)

Principle of superposition states that

$$DC = DA + DB \tag{2-37}$$

Or

$$\left(\frac{d\delta}{dt}\right)_{rv} = \left(\frac{d\delta}{dt}\right)_{r} + \left(\frac{d\delta}{dt}\right)_{v} \tag{2-38}$$

Let m_r be the slope of straight line OA, m_v be the slope of line OB and m_{rv} is the slope of line OC; then $DC = OD/m_{rv}$, $DA = OD/m_r$ and $DB = OD/m_v$. Substituting these in equation (2-37) gives:

$$\frac{1}{m_{rv}} = \frac{1}{m_r} + \frac{1}{m_v} \tag{2-39}$$

Equation of the straight line OA that connect ponts O and A, i.e. for radial consolidation

$$\delta = m_r \frac{d\delta}{dt} + \delta_{100} \tag{2-40}$$

which is the equation in the form of $y = mx + b$, where $y = \delta,\ x = d\delta/dt,\ m = m_r$ and $b = \delta_{100}$. In a similar way, equations for OB and OC respectively are:

$$\delta = m_r \frac{d\delta}{dt} + \delta_{100} \tag{2-41}$$

$$\delta = m_r \frac{d\delta}{dt} + \delta_{100} \tag{2-42}$$

Then equation (2-42) is the equation of straight-line OC for $3D$ consolidation, and Equation (2-40) can be written as

$$\frac{dt}{m_r} = \frac{d\delta}{\delta - \delta_{100}} = -\frac{dU_r}{1 - U_r} \tag{2-43}$$

Where U_r is the average degree of consolidation. The following relationship relates the average degeree of consolidation to the experimental settlement δ

$$U_r = \frac{\delta - \delta_0}{\delta_{100} - \delta_0} \tag{2-44}$$

Integration of the equation (2-43) results

$$\frac{1}{m_r} = \frac{\ln(1-U_r)}{t} \tag{2-45}$$

Likewise, the following equation are obtained from equations (2-41) and (2-42)

$$\frac{1}{m_v} = \frac{\ln(1-U_v)}{t} \tag{2-46}$$

$$\frac{1}{m_{rv}} = \frac{\ln(1-U_{rv})}{t} \tag{2-47}$$

Putting all values from equations (2-45) - (2-47) into equation (2-39), we get

$$(1-U_{rv}) = (1-U_r)(1-U_v) \tag{2-48}$$

The solution of general $3D$ consolidation equation (2-34) in Cartesian coordinates can be obtained by a simple analogy of the above-mentioned methods. Following similar procedure to the above process, equation (2-49) can be obtained as a solution for equation $3D$ consolidation equation (2-34)

$$(1-U_{xyz}) = (1-U_x)(1-U_y)(1-U_z) \tag{2-49}$$

Where U_{xyz} is the average degree of consolidation due to the horizontal drainage in X and Y directoins and vertical (Z) direction, simultaneously, U_x is the average degree of consolidation when only a horizontal drainage in X direction is allowed. U_y is the average degree of consolidation when only a horizontal drainage in Y direction is allowed. U_z is the average degree of consolidation when only a vertical drainage is allowed. For any two-dimensional drainage in horizontal and vertical directions, Equation (2-49) can be modified as:

$$(1-U_{yz}) = (1-U_y)(1-U_z) \tag{2-50}$$

Equations (2-49) and (2-50) are practicable when the soil has two or three drainage faces in a vertical and horizontal directions but the settlement is allowed in vertical direction only.

2.3.3 Finite Difference Solution of 2D Consolidation Equation

2.3.3.1 General

Numerical differentiation is a technique used to estimate the numerical value of a derivative at a point of some given function using only values from the function without using the analytical form of the function itself which may not even be known. To calculate the derivatives a given function $y = f(x)$ defined on a discrete finite set of grid points (x_0, x_1, x_2,...,x_N), it is assumed that the only data available for this calculation are the exact values of the function at the data points: (x_i , $y_i = f(x_i)$), and that the derivatives are required only at the discrete finite set of data points (x_i, y_i). (Passos, 2010)

The finite difference method is perhaps the easiest known technique for numerically solving PDEs and so it is often the first method chosen. The basic idea is to have functions be represented by their values at certain grid points and also have any partial derivatives be approximated through differences in these values. One disadvantage of this method is that it becomes quite complex when solving PDEs on irregular domains. Besides, it is not always easy to follow through and find solutions to the difference equations that result, evaluate their stability, or establish their convergence especially for PDEs with variable coefficients or PDEs which are non-linear. (Passos, 2010)

This method also uses a Taylor series expansion which has the additional advantage of also providing some estimate of the amount of error inevitably incurred in such calculations. With the finite difference method, the higher number of data points you use and the closer they are to each other, the better the precision of the results. In numerical analysis, there are implicit and explicit methods of approaches used for obtaining a solution with numerical approximations. These approximation solutions are used for time-dependent partial and ordinary differential equations since differential equations are required in computer simulations of physical processes. (Wikipedia, 2020)

The *finite difference method* is a procedure used to obtain approximate numerical solutions of a partial differential equation by discretizing the continuous physical domain (*z, x*) into a discrete finite difference grid (*zi, xi*) where the *zx* plane is partitioned into equally spaced grid lines parallel to the coordinate axes and defined by step sizes *h* and *k* given by

$$\Delta z = h = \frac{z_n - z_0}{n} \; and \; \Delta x = k = \frac{x_m - x_0}{m}$$

Where n and m are integers representing the number of grid points.

2.3.3.2 Explicit Finite Difference Method;

Explicit methods calculate the next value of a system based on from the current value of the system, whereas implicit methods involves both the current value and the next one to find a solution simultaneously. Mathematically, if $f(t)$ is the current value of the system and $f(t+\Delta t)$ is the value at the next time where Δt is a small-time step ,

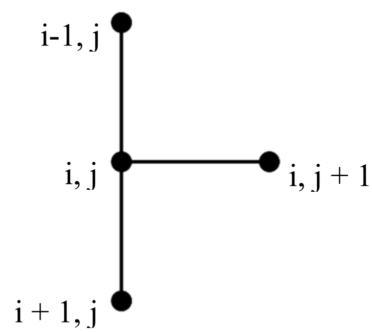


Figure 2-4: Explicit Finite Difference Method Grid

2.3.3.3 Implicit Finite Difference Method

Implicit methods require additional equations and computations for solving the same problem, and they are a little harder to implement. Implicit methods are used because many problems arising in practice are stiff, for which the use of an explicit method requires impractically small time steps Δt to keep the error in the result bounded.

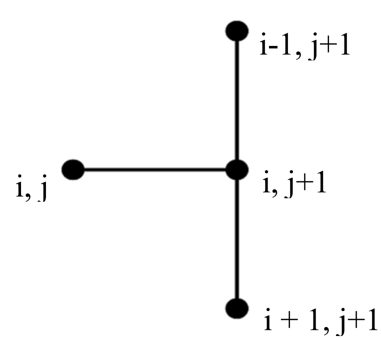


Figure 2-5: Implicit Finite Difference Method Grid

Alternating Direction Implicit (ADI) Method

Alternating Direction Implicit method was first proposed by Peaceman and Rachford in their in 1955. The Alternating Directional Implicit method is an iteration process which takes two-steps in each cycle that alternately solves the column and row spaces with an approximate solution. The Alternating Direction Implicit (ADI) method is used to solve problems in two-dimensional space.

The time-splitting ADI method is implicit in time. The general idea of the two-step time-splitting process is that the general implicit equation arising out of discretization of a two-dimensional partial differential equation can be solved by a solution of two spatially one-dimensional unsteady problems. Generally, the procedure can be implemented for all structured mesh. The stability of this method is unconditional since each of the two split steps essentially constitutes the solution of a one-dimensional problem using the backward Euler method. The ADI method has the following advantages in solving finite different equations:

i. It is convergent and unconditionally stable.
ii. It has second-order accuracy in both space and time.
iii. It leads to a tridiagonal system of equations.

3 METHODOLOGY AND ALGORITHM DEVELOPMENT

3.1 Introduction

A computer program is a collection of step-by-step commands that instructs the computer to do required specific tasks to return a required result.

Nowadays, computer programs become very important in our daily life especially in the field of engineering to enhance the power of computers, perform computations with high accuracy, and to handle rigorous calculations that cannot be done by hand.

The basic skill required to develop a computer program is a programming language and a knowledge of algorithms. An algorithm is a method to convert and breakdown a problem into a simple set of solutions that can be understood by the computer. And a programming language is a conventional rule for a way of telling what to do to the computer. There are more than 700 programming languages today. (Fowler, 2020) From these languages, C, C++, Java, Visual basics, phyton, C#, are among the most popular ones for desktop applications.

A C# programming language, pronounced as C sharp, is selected for this research to develop the software for the consolidation analysis. The major advantage of these programming languages that, like java and some others, a computer program written in C# can be deployed in any operating system like, windows, MAC, android, or ios platforms. The C# programming language is an object-oriented programming language developed by Microsoft and is entirely based on the C and C++ languages by simplifying many of the complexities of C++; hence, it is a very easy and powerful programming language.

Microsoft Visual Studio is used to write, compile, and test the codes. Microsoft Visual Studio is an Integrated Development Environment (IDE) from Microsoft, which consists of a source code editor, a debugger, and build automation tools to make software development easy.

Besides the codes and algorithms, the other major work in the development of computer software is the Graphical User Interface (GUI). A Graphical User Interface is a visual means of interacting with the user using objects such as windows, buttons, and menus. One of the main advantages of GUI is to have attractive and easy input and output system while using the software.

3.2 Computer Program for Analysis of One-Dimensional Consolidation

3.2.1 General

A one-dimensional analysis for primary consolidation of soil layers is commonly used for estimation of soil settlement due to consolidation. One of the reasons for the popularity of this analysis method over the two- dimensional and three-dimensional consolidation analysis is its easiness for computation.

The governing equation of the one-dimensional consolidation theory, from equation (2-35) is:

$$\frac{\partial \overline{u}}{\partial t} = C_v \frac{\partial^2 \overline{u}}{\partial z^2} \tag{3-1}$$

The Finite difference is commonly used for an approximate solution of the derivative for developing computer programs, The first and second-order derivatives of a function f at a point x with a finite difference method can be defined by the following approximation equations with a small interval of Δx .

First Order Forward:

$$\frac{d(f(x))}{dx} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{3-2}$$

Second Order Forward:

$$\frac{d^2(f(x))}{dx^2} \approx \frac{f(x+2\Delta x) - 2f(x+\Delta x) + f(x)}{(\Delta x)^2} \tag{3-3}$$

The same differential equation using the backward difference is defined as follows,

First Order Backward:

$$\frac{d(f(x))}{dx} \approx \frac{f(x) - f(x-\Delta x)}{\Delta x} \tag{3-4}$$

Second Order Backward:

$$\frac{d^2(f(x))}{dx^2} \approx \frac{f(x) - 2f(x-\Delta x) + f(x-2\Delta x)}{(\Delta x)^2} \tag{3-5}$$

Similarly, the central difference approximation is written as, First Order Central:

$$\frac{d(f(x))}{dx} \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2 * \Delta x} \tag{3-6}$$

Second Order Central:

$$\frac{d^2(f(x))}{dx^2} \approx \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2} \tag{3-7}$$

A Backward difference method and central difference method gives an implicit solution for derivative of *f* at *x*. Using the backward difference approximation for the first partial derivative and the central difference approximation for the second partial derivative by using a Taylor series expansion

The first and second derivatives of a function f(x) with respect to x can be written as:

$$\frac{d(f(x))}{dx} = \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{3-8}$$

$$\frac{d^2(f(x))}{dx^2} = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2} \tag{3-9}$$

Both equation (3-8) and equation (3-9) are used in the governing equation in the left-hand side and right-hand side respectively.

3.2.2 Input Data and Graphical User Interface (GUI)

The input data for one-dimensional consolidation consists of the following parameters of the soil layer:

i. Thickness of consolidating soil layer: this is the soil layer beneath the load in which the excess pore water pressure arises due to the additional induced pressure. The user is expected to input the thickness of the consolidating layer measured in meter.

ii. Coefficient of Permeability/Coefficient of Consolidation: The coefficient of permeability of soil describes how easily a liquid will move through the soil. It is also commonly referred to as the hydraulic conductivity of the soil.

    The coefficient of Consolidation is soil parameter used to measure the consolidation rate of saturated clay when it is subjected to an increase in the pressure. The unit is in square meter per year for this software. The User can input either the coefficient of permeability or the coefficient of consolidation since the can be related with a formula.

iii. Computation interval for space and time: the user can adjust the required interval for the horizontal and vertical space as well as the time increment. It can also be possible to use the default interval and increment suggested by the software.

Other required data expected from the user are the boundary conditions and initial condition of the system.

i. Initial Condition of the system is the initial excess pore water distribution which is a function of space. The function may be a polynomial or linear function.

ii. Drainage conditions are boundary conditions to be applied at the boundary of the system, if the drainage condition at a boundary is permeable the excess pore water pressure is always zero at each node of the drainage, otherwise it would have an equation to evaluate on the nodes.

The Graphical User Interface designed to get input from the user for one-dimensional consolidation analysis is shown in Figure 3-1.

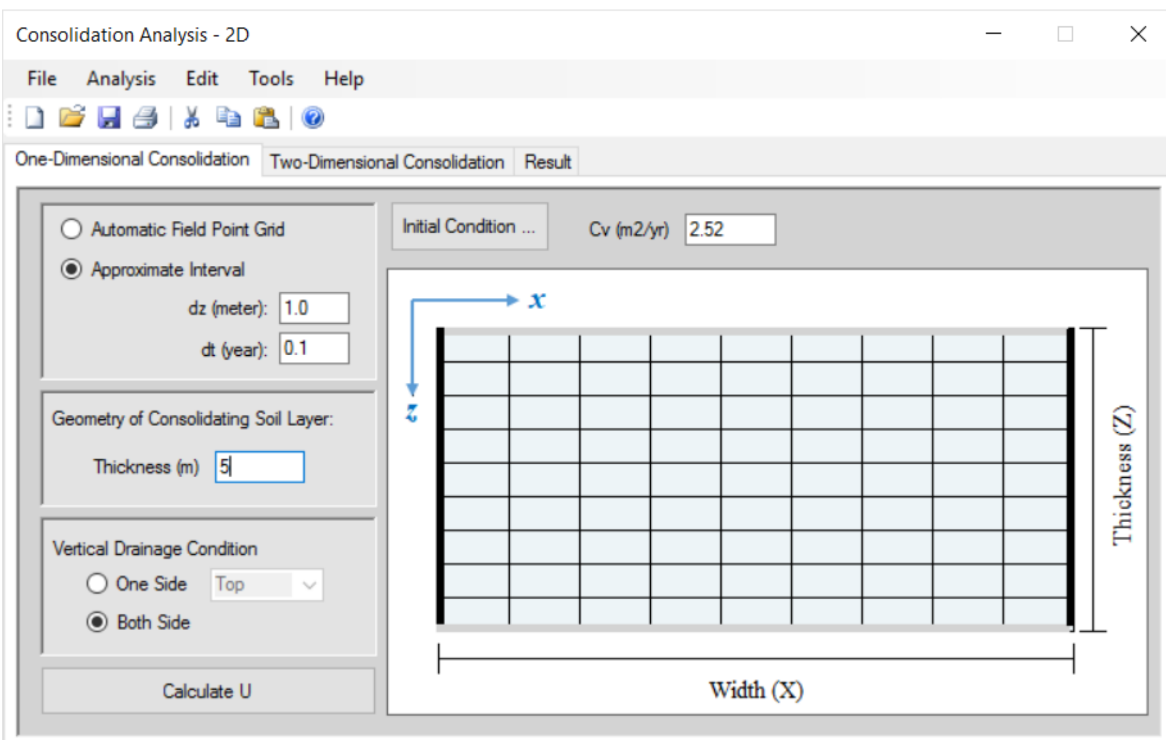


Figure 3-1: Graphical User Interface for one-dimensional Consolidation Analysis

3.2.3 Algorithm for One-Dimensional Consolidation Analysis

The first task performed in solving the governing equation using a numerical solution is to approximate the second-order derivatives by a finite difference method. From the commonly used finite difference methods: forward, backward, and central, using the backward difference method and central difference method will result in a much more desired implicit solution that is unconditionally stable. (Smith, 1985)

Using the backward difference approximation for the first partial derivative and the central difference approximation for the second partial derivative by using a Taylor series expansion as given in Equation (3-4) and (3-7), we can then write the general parabolic differential Equation (3-1) as:

$$\frac{u_{i,j}-u_{i,j-1}}{\Delta t}-C_v\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta z)^2}=0 \qquad (3\text{-}10)$$

For each $i=1,2,\ldots,m-1$ and $j=1,2,\ldots$ and where $u_{i,j}$ approximates $u(x_i,t_j)$.

By letting $\lambda=C_v\frac{\Delta t}{(\Delta z)^2}$, the previous equation can be re-arranged for $u_{i,j}$ in the following format:

$$(1+2\lambda)u_{i,j}-\lambda u_{i+1,j}-\lambda u_{i-1,j}=u_{i,j-1} \qquad (3\text{-}11)$$

For each $i = 1, 2, \ldots, m-1$ and $j = 1,2, \ldots$ using the knowledge from the initial conditions that $u_{i,0}=f(z_i)$ for each $i=1,2,\ldots,m-1$ and applying the boundary conditions to $u_{m,j}$ and $u_{0,j}$ this backward difference method leads to the following matrix representation:

$$\begin{pmatrix} (1+2\lambda) & -\lambda & \ldots & \ldots & 0 & 0 \\ -\lambda & (1+2\lambda) & -\lambda & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & . & \vdots \\ 0 & \cdots & -\lambda & (1+2\lambda) & -\lambda & 0 \\ 0 & 0 & \ldots & -\lambda & (1+2\lambda) & -\lambda \\ 0 & 0 & 0 & \ldots & -\lambda & (1+2\lambda) \end{pmatrix} \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ . \\ . \\ . \\ u_{m-1,j} \end{pmatrix} = \begin{pmatrix} u_{1,j-1} \\ u_{2,j-1} \\ . \\ . \\ . \\ u_{m-1,j-1} \end{pmatrix} \qquad (3\text{-}12)$$

Therefore, at this point, it is just needed to solve a system of linear equations to obtain $u^{(j)}$ from $u^{(j-1)}$. Since $\lambda > 0$, the matrix A is positive definite and strictly diagonally dominant.

The pseudo-code for the algorithm of a one-dimensional partial differential equation is as follows:

```
Function solve 1D PDE (input Cv, dz, dt, initialPoreWaterPressure,
verticalDrainageCondition)
{
      Declare variables Average_Degree_of_Consolidation
      Compute lambda = Cv * dt / (dZ * dZ)
      Declare and initialize a tri-diagonal Matrix [A] with coefficients
      While degree of consolidation is less than 95 % do the following
      {
        Calculate column Matrix {b} from previousTimePorewaterDistribution
        Sovle for {u} from equation [A]{u} = {b}
        Apply Boundary Conditions, ui = 0 if drainage is allowed
        Calculate the Average Degree of Consolidation
        U_avg = 1 - U_current / U_initial
      }
      Calculate the time for complete consolidation
      Duration of Consolidation = number of iteration * dt
       return Average Degree of Consolidation, Duration of Consolidation;
}
```

3.3 Computer Program for Analysis of Two-Dimensional Consolidation

3.3.1 General

It is often found that the observed rate of settlement of embankment is faster than the predicted amount from one-dimensional consolidation theory. The reason is that the width of the embankments is much smaller than the depth of the supporting soils under the embankments. Besides, the permeability of the supporting soils in the horizontal directions is usually larger than that in the vertical directions. (Tandjiria, 1999)

To overcome this limitation of one-dimensional consolidation theory, it is necessary to adopt the two-dimensional consolidation theory. The governing equation of the two-dimensional consolidation theory, from equation 2-35, is:

$$\frac{\partial \bar{u}}{\partial t} = C_{hx} \frac{\partial^2 \bar{u}}{\partial y^2} + C_{vz} \frac{\partial^2 \bar{u}}{\partial z^2} \qquad (3\text{-}13)$$

There are several ways to simplify a differential equation such as the backward, forward, and central difference approximations. Taylor series expansion is the basic concept of the difference approximations. The first derivative of a function $f(x)$ with respect to x can be written as:

$$\frac{d(f(x))}{dx} = \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{3-14}$$

While the second derivative of a function $f(x)$ with respect to x can be written as:

$$\frac{d^2(f(x))}{dx^2} = \frac{f(x+\Delta x) - 2f(x) + f(x+\Delta x)}{(\Delta x)^2} \tag{3-15}$$

Back to the governing equation (2-35) equation (3-14) is to be used in the left-hand side for the first derivative of u with respect to t and equation (3-15) is to be used in the right-hand side in place of the second derivative of u with respect to x and y.

3.3.2 Input Data and Graphical User Interface for Two-Dimensional Consolidation

The input data consists of the following parameters of the soil layer:

i. The thickness of the consolidating soil layer: this is a soil layer beneath the load in which the excess pore water pressure arises due to the additional induced pressure. The user is expected to input the thickness of the consolidating layer measured in meter.

ii. Coefficient of Permeability/Coefficient of Consolidation: The coefficient of permeability of soil measures how easily a liquid will transmit through the soil. The Coefficient of permeability is also referred as the hydraulic conductivity of the soil.

   The coefficient of Consolidation is the parameter that measures consolidation rate when the soil is subjected to an increase in the pressure. The unit is square meter per year for this software. The User can input either the coefficient of permeability or the coefficient of consolidation since the can be related with a formula.

iii. Computation interval for space and time: the user can adjust the required interval for the horizontal and vertical space as well as the time increment. It can also be possible to use the default interval and increment suggested by the software.

Other required data from the user are the boundary conditions and initial condition of the system.

i. Initial Condition of the system is the initial excess pore water distribution which is a function of space. The function may be a polynomial or a linear function.

ii. Drainage conditions are boundary conditions to be applied at the boundary of the system, if the drainage condition at a boundary is permeable the excess pore water pressure is always zero at each node of the drainage, otherwise it would have an equation to evaluate on the nodes.

The Graphical User Interface designed to get input from the user for one-dimensional consolidation analysis is shown in Figure 3-2:

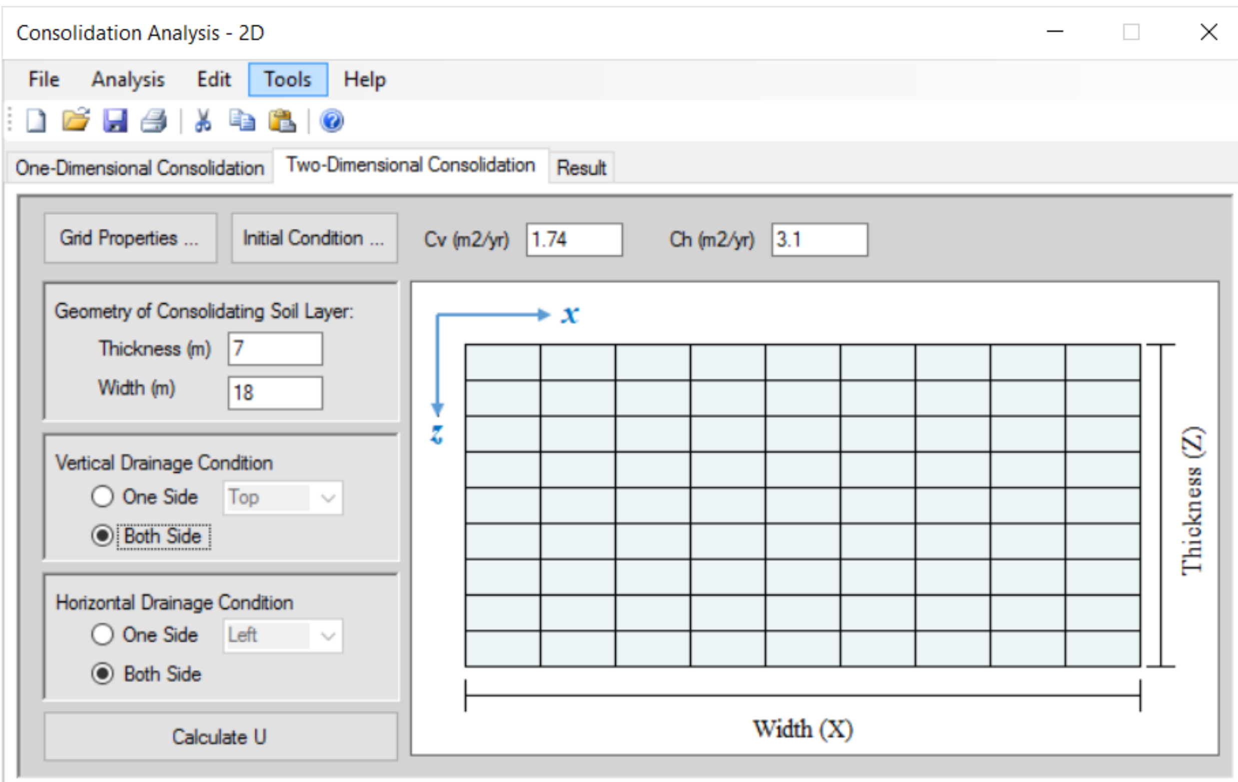


Figure 3-2: Graphical User Interface for Two-Dimensional Analysis

3.3.3 Algorithm for Two-Dimensional Consolidation Analysis

The first task here, as in one-dimensional analysis, is also converting the differential expression in the governing equation of the consolidation problem into a difference equation to be solved by numerical solution methods. As described in section 2.4.3 among several methods of finite difference method the Alternating Directional Implicit (ADI) Method is an implicit solution as its name implies, convergent, and unconditionally stable. The ADI also leads to a tridiagonal system of equations that minimizes computation time and memory usage of the computer.

Due to these advantages, the Alternating Directional Implicit method is used for solving the parabolic partial differential equation of the governing two-dimensional consolidation problem for the software development in this thesis. The ADI method is simply the extension of a backward difference method alternating in two directions at each step. The following notation is used for the analysis:

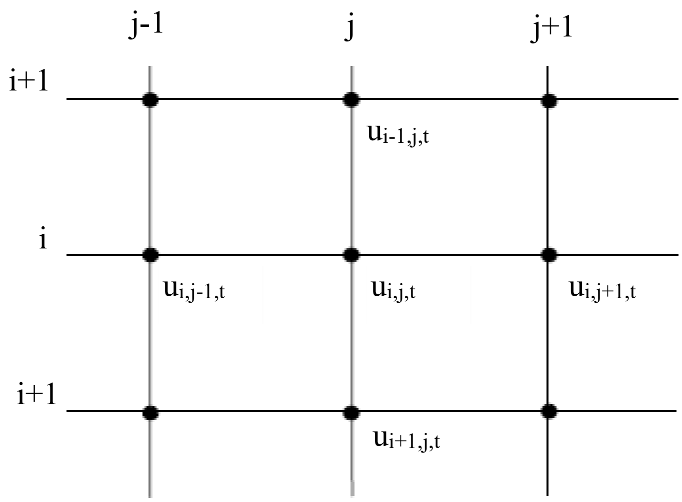


Figure 3-3: Notation for finite difference grid

Using the ADI method the equation to solve at node i,j at the time level (n+1) and (n+2) is as follows:

For the time interval between n and (n+1):

$$\frac{u_{i,j.n+1} - u_{i,j.n}}{\Delta t} = C_h \frac{u_{i,j-1,n+1} - 2u_{i,j,n+1} + u_{i+1,j,n+1}}{(\Delta x)^2} + C_v \frac{u_{i-1,j,n} - 2u_{i,j,n} + u_{i+1,j,n}}{(\Delta z)^2} \qquad (3\text{-}16)$$

For the time interval between (n+1) and (n+2):

$$\frac{u_{i,j,n+2} - u_{i,j,n+1}}{\Delta t} = C_h \frac{u_{i,j-1,n+1} - 2u_{i,j,n+1} + u_{i,j+1,n+1}}{(\Delta x)^2} + C_v \frac{u_{i-1,j,n+2} - 2u_{i,j,n+2} + u_{i+1,j,n+2}}{(\Delta z)^2} \qquad (3\text{-}17)$$

Let

$$a_x = \frac{C_h \Delta t}{(\Delta x)^2}$$
$$a_z = \frac{C_v \Delta t}{(\Delta z)^2} \qquad \textit{(3-18)}$$

Substituting Equations (3-18) and collecting the unknown values for one increment of n on the left-hand side while the known values on the right-hand side, Equation (3-16) and (3-17) can be written as:

For the time interval between n and (n+1):

$$-a_x u_{i,j,n+1} + (1+2a_x)u_{i,j,n+1} - a_x u_{i,j+1,n+1} = a_z u_{i-1,j,n} + (1-2a_z)u_{i,j,n} + a_z u_{i+1,j,n} \qquad \textit{(3-19)}$$

For the time interval between (n+1) and (n+2):

$$-a_z u_{i-1,j,n+2} + (1+2a_z)u_{i,j,n+2} - a_z u_{i+1,j,n+2} = a_x u_{i,j-1,n+1} + (1-2r_x)u_{i,j,n+1} + a_x u_{i,j+1,n+1} \qquad \textit{(3-20)}$$

It can be noted that both equations contain three known and three unknown values of excess pore water pressure. The first equation contains unknown values on a horizontal line which means that the equation is implicit in the horizontal direction. The second equation contains unknown values of excess pore water pressure on a vertical line which makes it implicit in the vertical direction for a certain time step. Both equations (3-19) and (3-20) can be solved by the system of simultaneous equation presented in the following matrix equation.

$$[A]\{u\} = \{b\} \qquad (3\text{-}21)$$

Where,

$$[A] = \begin{bmatrix} 1+2a_x & -2a_x & 0 & . & 0 & 0 & 0 \\ -a_x & 1+2a_x & -a_x & . & 0 & 0 & 0 \\ 0 & -a_x & 1+2a_x & . & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & -a_x & 1+2a_x & -a_x \\ 0 & 0 & 0 & . & 0 & -a_x & a_x \end{bmatrix}$$

$$\{u\} = \begin{Bmatrix} u_{i,1,n+1} \\ u_{i,2,n+1} \\ u_{i,3,n+1} \\ . \\ . \\ u_{i,j_b-1,n+1} \\ u_{i,j_b,n+1} \end{Bmatrix} \quad \text{and,} \quad \{b\} = \begin{Bmatrix} a_z u_{i-1,1,n} + (1-2a_z)u_{i,1,n} + a_z u_{i+1,1,n} \\ . \\ . \\ . \\ . \\ . \\ a_z u_{i-1.j_b,n} + (1-2a_z)u_{i,j_b,n} + a_z u_{i+1,j_b,n} \end{Bmatrix}$$

Then using the tridiagonal matrix elimination method (Thomas algorithm) the unknown values of excess pore water pressure u is solved for each node.

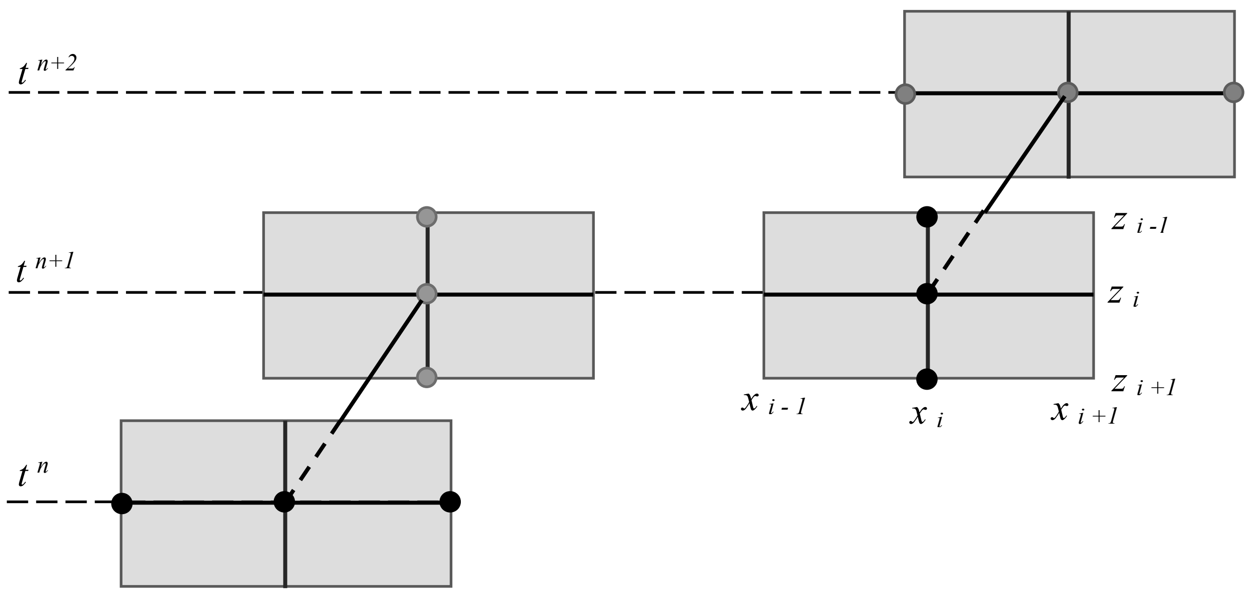


Figure 3-4: Computation grids for the two-step time increment

The pseudo-code for the algorithm to solve the two-dimensional partial differential equation is as follows:

```
function solve 2D PDE (input Cv, Ch, dz, dx, dt, initialPoreWaterPressure,
verticalDrainageCondition, horizontalDrainageCondition)
{
      Declare Variables Average_Degree_of_Consolidation
      Compute rx = C_h * intervalt / (dx * dx)
      Compute rz = C_v * intervalt / (dz * dz)
      double [,] u1 = new double[U0.RowLength, U0.ColumnLength];  //current u
      Declare and initialize Coefficient Matrices Az and Ax
```

```
        for odd time increment do
          solve {Ut} using equation (4-9) in a row direction
          Calculate column Matrix {b} from previousTimePorewaterDistribution
          Apply Boundary Conditions                C
          Calculate the Average Degree of Consolidation = 1 - U_current/U_initial
        for even time increment do
          solve {Ut} using equation (4-9) in a column direction
          Calculate column Matrix {b} from previousTimePorewaterDistribution
          Apply Boundary Conditions
          Calculate the Average Degree of Consolidation = 1 - U_current /
          U_initial
        Repeat the above two steps till Average_Degree_of_Consolidation approaches
        unity
        return Average Degree of Consolidation, Total Time of Consolidation
}
```

3.4 Validation of software output

One of the techniques for validation is using analytical solutions. Validation using analytical solutions can be applied when the numerical solution is a simulation of the analytical solution and the analytical solution for the problem is already known. The main problem of this method is that getting analytic solutions to most real problems is impossible or very difficult which is most case for a numerical solution in need. Though, if the real problem has the analytical solution and the need for a numerical solution is just for the purpose of computer computation, this method is very useful.

A one-dimensional analysis is one of these problems that have the analytic solution and the validation for the output of the software for one-dimensional analysis is done using analytical solutions from the exact solution of Terzaghi's theory. Figure 3-5 is a graph of the average degree of consolidation, U, versus dimensionless time factor, T_v for Terzaghi's exact solution taken from a research paper by Wan NurFirdaus.

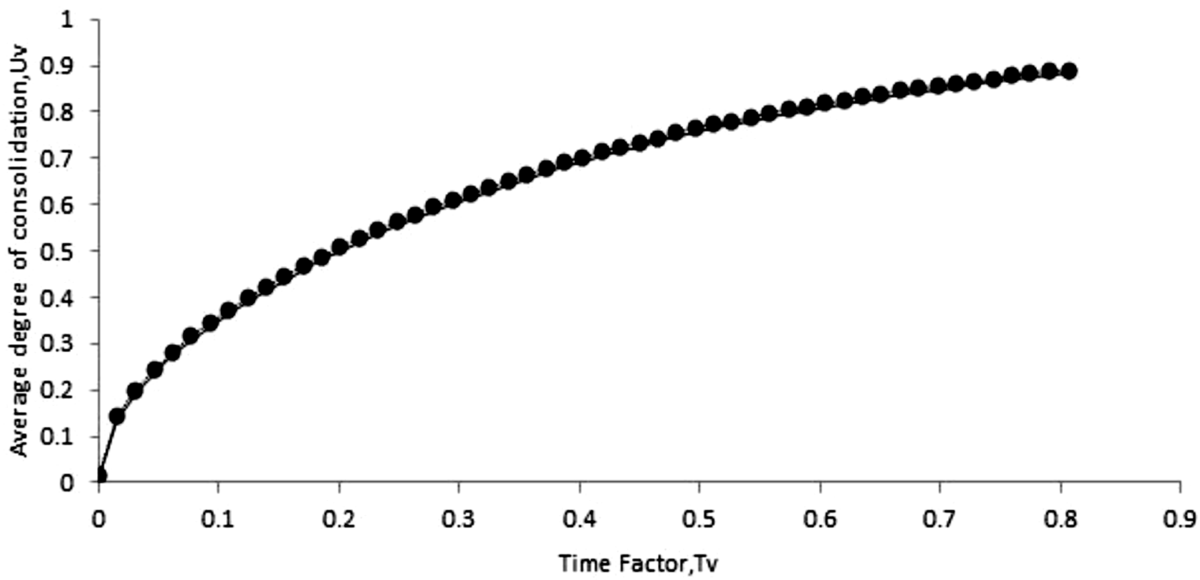


Figure 3-5: Average degree of consolidation versus dimensionless time factor (U vs T_v) of Terzaghi's Exact Solution from Wan NurFirdaus (Hassan & HishamMohamad, 2012)

Figure 3-6 shows a graph of the comparison between Terzaghi's exact solution and the software output for the average degree of consolidation, U versus the dimensionless time factor, Tv. Close results were found from the Finite Difference Solution of the software.

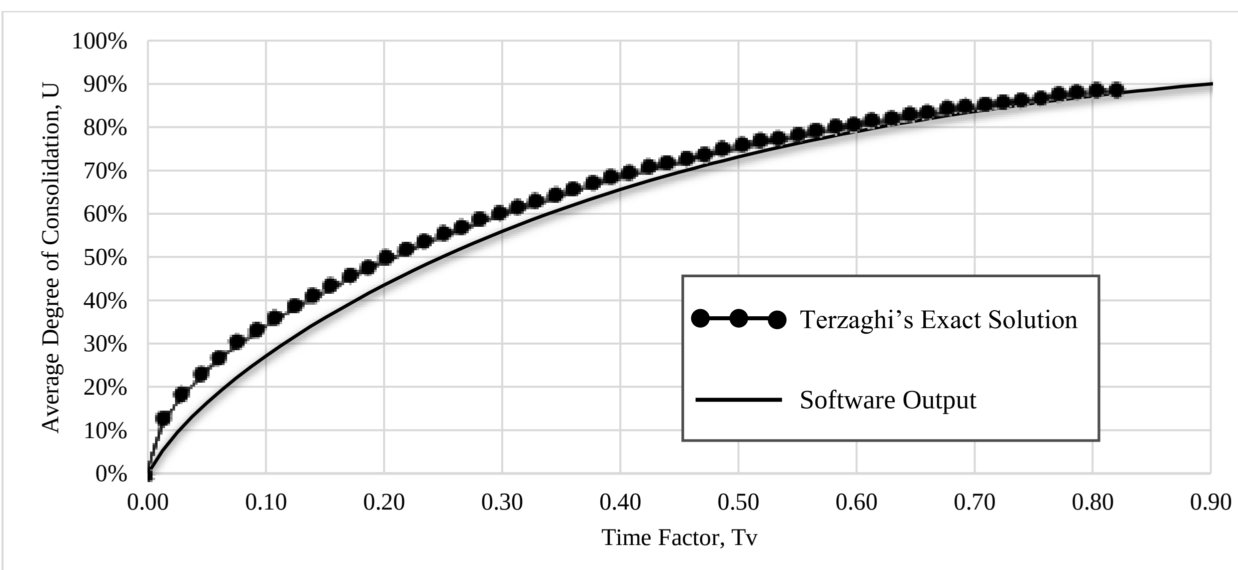


Figure 3-6: Comparison between Terzaghi's Exact Solution and Software Output

The two curves in Figure 3-6 are in a good agreement to say the software output can be applied for the analysis of consolidation problems. The possible reason for the gap between the exact solution and the software output is encountered due to the following reasons:

i. Since a finite difference method is a numerical solution, it is an approximate solution to the real problem.
ii. Different assumption of the initial pore water distribution leads to such differences especially at the beginning of the curve.

The above-mentioned procedure is deemed to validate the two-dimensional analysis too since the same Finite Difference Method is used just by extending into two-dimension space using the principle of superposition, It is also difficult to get an analytic solution for two-dimensional analysis. However, a comparison with measured consolidation data is performed to show the output from the software is not beyond the actual measure values in a three-dimensional problem. Measured data of primary consolidation from a research paper by Supangkat (Supangkat, 1994) is taken for comparison purpose.

The following graph shows excess pore water pressure, (1- U), versus dimensionless time factor, T_v measured for the experimental purpose by Supangkat. (Supangkat, 1994)

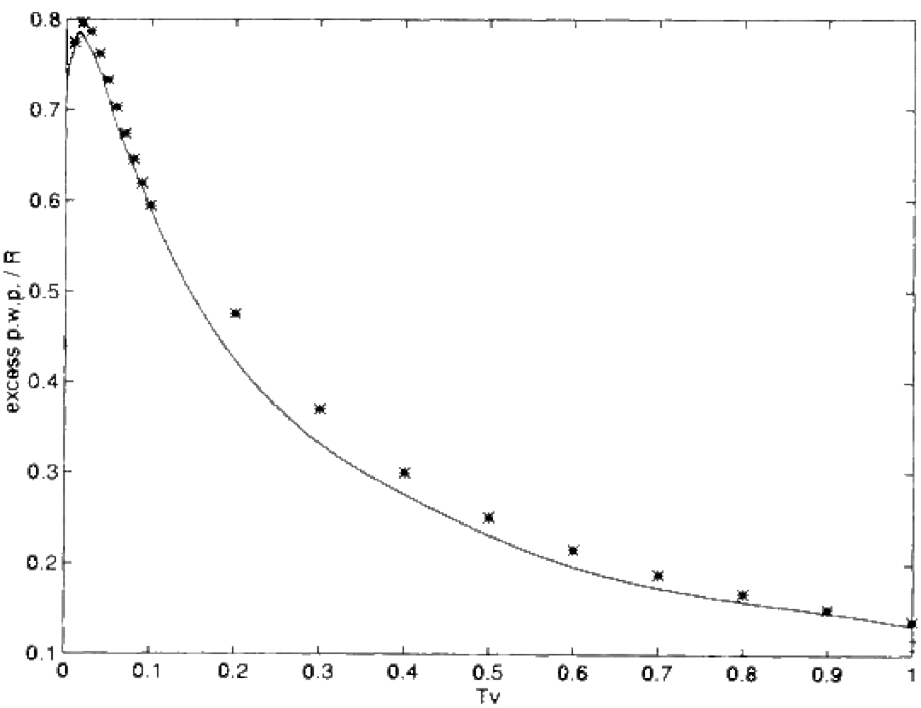


Figure 3-7: Excess pore water pressure versus dimensionless time factor from Supangkat

The average degree of consolidation is computed from the excess pore water pressure measured from the experimental data and the average degree of consolidation versus the dimensional time factor graph is plotted to compare and validate the software output. Figure 3-8 shows a graph of the comparison between the experimental data from Supangkat (Supangkat, 1994) and the software output for the average degree of consolidation, U versus the dimensionless time factor, Tv.

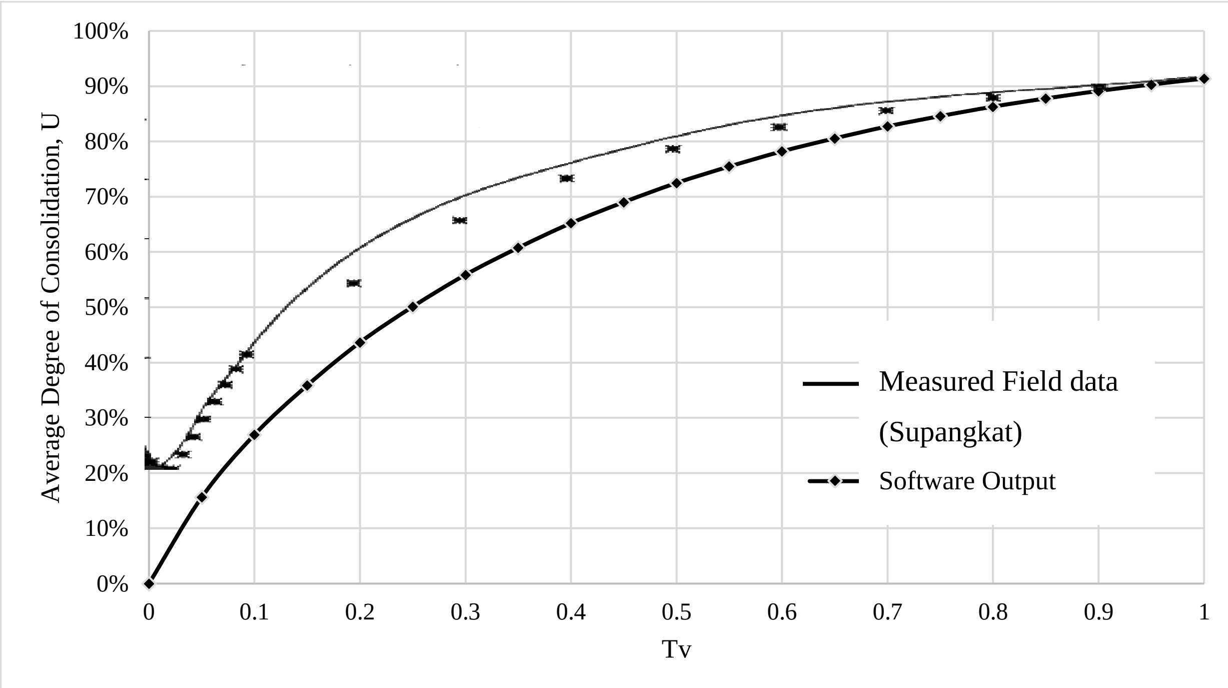


Figure 3-8: Average Degree of Consolidation versus Dimensionless time factor

The comparison shows some gap between the curves which is expected because the measured actual data is a three-dimensional problem. However, it can be seen that the software gives a closer lower bound result and it is always between the exact solution of the one-dimensional curve and the actual measured consolidation result.

4 ANALYSIS RESULTS AND DISCUSSIONS

4.1 Introduction

This section presents the comparison of results from the software analysis between one-dimensional consolidation analysis and two-dimensional analysis to observe the significance of the error from the one-dimensional analysis. From the results, cases that a one-dimensional analysis can be used with good precisions are also identified. The results are presented systematically to meet research aims and objectives by showing the differences in results between the two consolidation analysis methods.

4.2 Analysis Examples and Model Description

A step by step procedures, to do consolidation analysis using the Software, are presented on ANNEX I to illustrate how one can approach a one-dimensional and two-dimensional consolidation analysis using the software developed during the study of this research. In the previous sections, the fundamental principles of one-dimensional and two-dimensional consolidation problems have been discussed. In these examples, the basic principles for analysis purposes are used. In the next section, the results and discussions of comparison between one – dimensional and two – dimensional consolidation analysis presented.

Comparison between the results obtained from the software for two-dimensional consolidation analysis and one-dimensional consolidation analysis to show the need for the two-dimensional analysis. Several tests were run by varying geometry of the soil layer and soil parameters are examined for comparison. From the test results, the effect of these variable parameters on the significance of the errors on one-dimensional analysis is observed. From all these analyses, in conclusion, the best scenario that a one-dimensional analysis could be used without significant error is recommended.

Table 4-1 shows the various scenarios considered for the comparison of one-dimensional and two-dimensional consolidation analysis.

Table 4-1: Model Parameter Inputs

Properties	Model 1	Model 2	Model 3	Model 4	Model 5
X (m)	var	var	var	15.0	25.0
Z (m)	**3.0**	**5.0**	**8.0**	**3.0**	**5.0**
X/Z	var	var	var	5	5
Ch (m^2/yr)	4.0	4.0	4.0	**2.0**	**2.0**
Cv (m^2/yr)	2.0	2.0	2.0	2.0	2.0
Ch/Cv	2	2	2	1	1

4.3 Results and Discussion

To identify the behavior of one-dimensional and two-dimensional consolidation analysis, twenty tests grouped in five models were analyzed for different width to depth ratios, different thicknesses of the soil layer, different coefficient of consolidation, and different ratios for horizontal to a vertical coefficient of consolidation. Cases are summarized and discussed as follows.

Five models sorted into two cases are tested for comparison of one-dimensional and two-dimensional consolidation analysis. The first case examines the difference between the results from two consolidation analysis theories by varying width to depth (X/Z) ratio of the consolidating soil mass. Whereas the second case studies the effect of varying the coefficient of consolidation.

The initial pore water pressure distribution remains the same for all tests and the equation used is $U_0 (x,y) = U_o - z^2 - 2x^2$.

CASE I: varying X/Z ratio for different thickness of soil layer (Z)

Varying width to depth ratio with other factors remaining constant helps to identify the effect of the length of the lateral drainage path. It is expected that, when the width to depth ratio is getting reduced, the consolidation due to lateral drainage becomes significant and, thus, a two-dimensional analysis becomes necessary. However, several models are required to identify the minimum X/Z ratio where a one-dimensional analysis cannot give reasonably good estimation any more.

This case consists of three models to observe the effect of variation of width to depth ratio for three different thickness of consolidating layer (3 m, 5 m, and 8 m) described as follows:

Model 1: This model consists of three tests for two-dimensional analysis (with width to depth ratios of 2, 3, and 5) to compare with a one-dimensional analysis with all tests done for Z=3 m. This model shows the effect of varying width to thickness ratio on small soil layer thickness.

The result of the analysis of this model is shown in Figure 4-1 and Table 4-2:

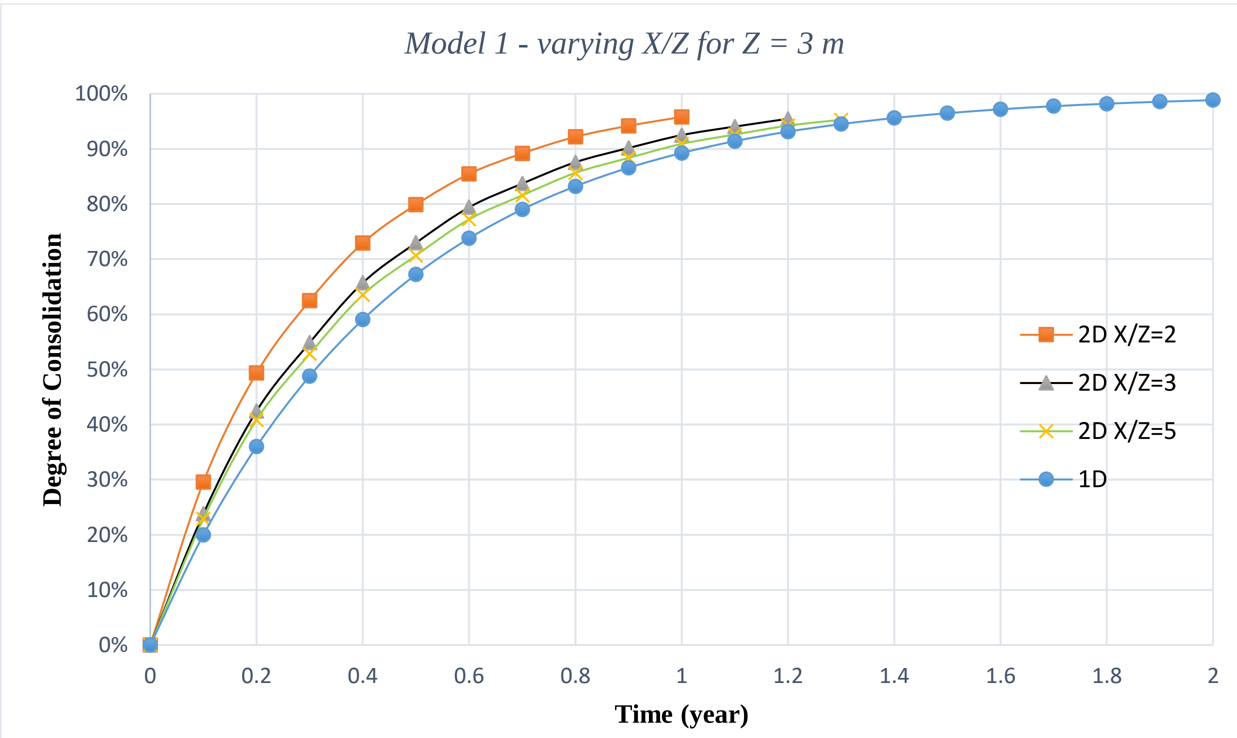


Figure 4-1: Model 1 - varying X/Z for Z = 3 m

To have a detailed look at the results, times corresponding to 50% and 90% consolidation are selected from the graph and presented in the following table:

Table 4-2: Model 1 - varying X/Z for Z = 3 m, Ch/Cv = 2, Cv=2.0 m^2/yr

Model 1	Z= 3.0 m	X/Z = 2	X/Z = 3	X/Z = 5
t_{50} (yr.)	2D Analysis	0.20	0.25	0.28
	1D Analysis	0.32	0.32	0.32
	Time Slippage	**60.0%**	**28.0%**	**14.3%**
t_{90} (yr.)	2D Analysis	0.70	0.90	1.00
	1D Analysis	1.01	1.01	1.01
	Time Slippage	**44.3%**	**12.2%**	**1.0%**

As it is seen from Figure 4-1 and Table 4-2, the results from one-dimensional and two-dimensional analysis shows a larger gap for smaller width to thickness ratio (X/Z < 5). The reason for this is the smaller the X/Z ratio means a shorter length to the lateral drainage faces which makes the consolidation due to the lateral drainages become significant. Since the one-dimensional analysis does not consider the lateral drainage, the difference gets larger with a significant time slippage.

In the next model similar model but with different layer thickness is tested to assess the difference between results of the one-dimensional and two-dimensional analyses with different thicknesses of consolidating soil.

Model 2: X/Z = 2,3, 5 for Z=5 m

Similar to Model 1, this model consists of three tests for two-dimensional analysis (with width to thickness ratios of 2, 3, and 5) to compare with a one-dimensional analysis with all tests done for **Z=5 m.** This model represents medium-thick soils and helps to assess the effect of varying width to thickness ratio on soil layers with a medium thickness.

The result of the analysis of this model is shown in Figure 4-2 and Table 4-3:

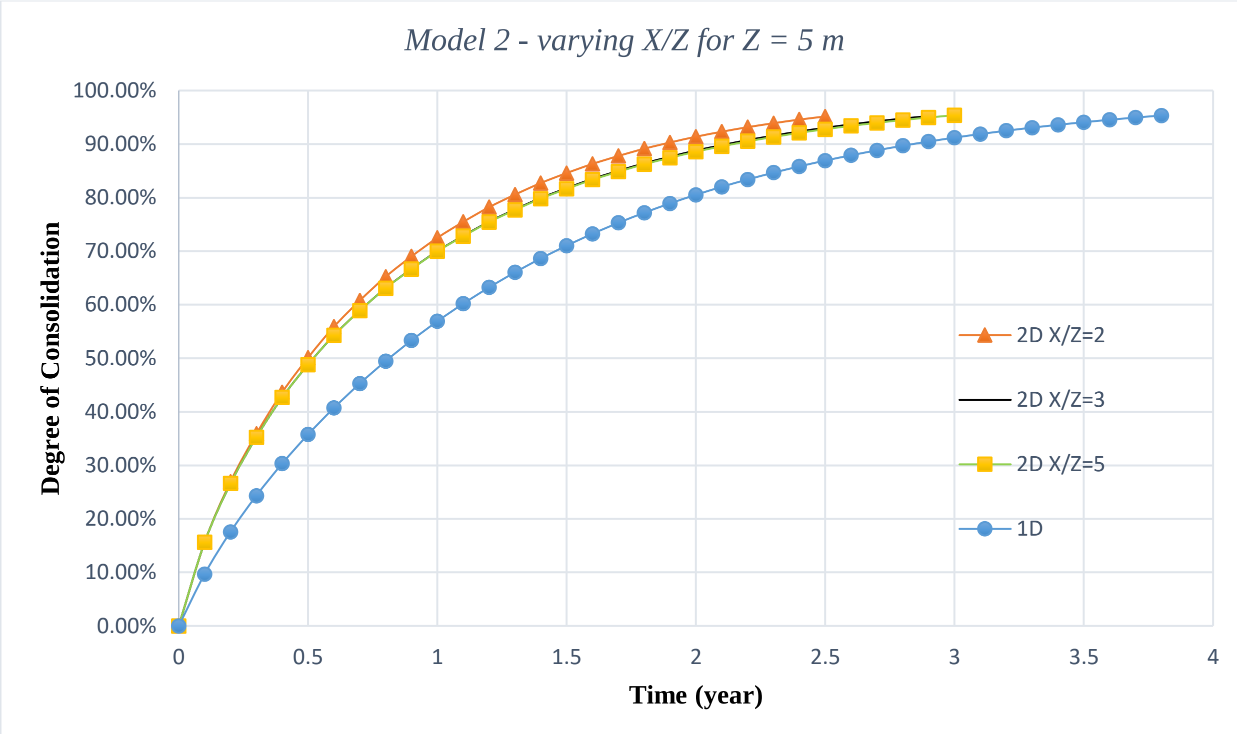


Figure 4-2: Model 2 - varying X/Z for Z = 5 m

The following table shows a comparison with selected times corresponding to 50% and 90% consolidation.

Table 4-3: Model 2 - varying X/Z for Z = 5 m, Ch/Cv = 2, Cv=2.0 m²/yr

Model 2	**Z= 5.0 m**	**X/Z = 2**	**X/Z = 3**	**X/Z = 5**
t_{50} (yr.)	2D Analysis	0.50	0.52	0.52
	1D Analysis	0.80	0.80	0.80
	Time Slippage	**60.0%**	**53.8%**	**53.8%**
t_{90} (yr.)	2D Analysis	1.90	2.11	2.12
	1D Analysis	2.90	2.90	2.90
	Time Slippage	**52.6%**	**37.4%**	**36.8%**

In this model, similar to Model 1, a larger difference between one-dimensional and two-dimensional analysis results is noted for smaller width to thickness ratio (X/Z < 5) for the same reason described for model 1.

Besides, Model 2 (i.e. Z=5 m), when compared to Model 1 (i.e. Z=3 m), results in a wider gap between one-dimensional and two-dimensional analyses results. From this, it can be observed that increasing the thickness of the consolidating layer increases the error from the one-dimensional analysis. This is because of the applied load that did not extend to the full width of the model which limits the excess pore water only in the vicinity of the load. i.e. the excess pore water pressure due to applied pressure dissipates before it reaches the lateral drainage face. From this, it can be noted that, even if the geometry of the consolidating layer has a larger width, the breadth of applied load determines the model dimensions.

The effect of a further increase in layer thickness is tested in model 3.

Model 3: X/Z = 2,3, 5 for Z=8 m

In this model, the thickness of the soil is increased to **Z=8 m** to tests the effect of further increasing the layer thickness on the difference between results from two-dimensional analysis and a one-dimensional analysis (for different width to thickness ratios of 2, 3, and 5). This model represents consolidation under soils with a large thickness.

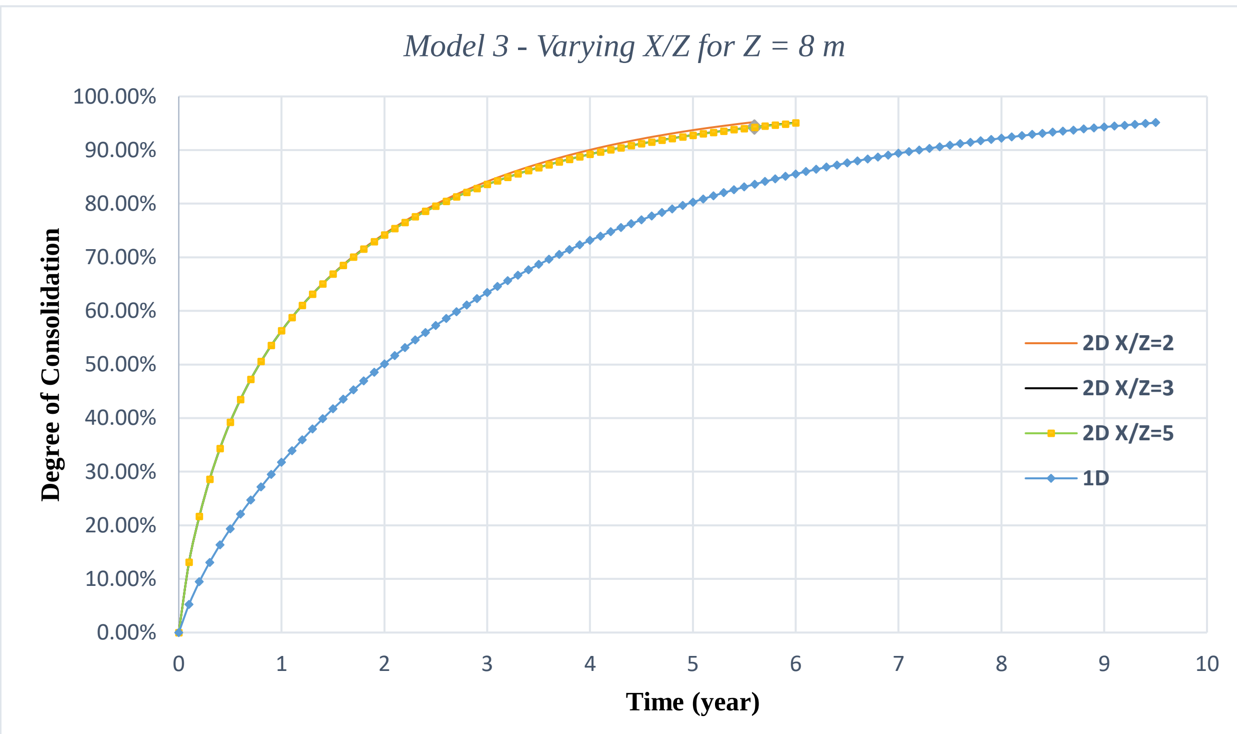


Figure 4-3: Model 3 - Varying X/Z for Z = 8 m

Model 3 further clarifies that one-dimensional analysis on a soil mass with larger layer thickness leads to a larger error. times corresponding to 50% and 90% consolidation are selected to show the gaps quantitatively and presented in the following table.

Table 4-4: Model 3 - Varying X/Z for Z = 8 m, Ch/Cv = 2, Cv=2.0 m^2/yr

Model 3	**Z= 8.0 m**	**X/Z = 2**	**X/Z = 3**	**X/Z = 5**
t_{50} (yr.)	2D Analysis	0.80	0.80	0.80
	1D Analysis	2.00	2.00	2.00
	Time Slippage	**150.0%**	**150.0%**	**150.0%**
t_{90} (yr.)	2D Analysis	4.00	4.20	4.20
	1D Analysis	7.20	7.20	7.20
	Time Slippage	**80.0%**	**71.4%**	**71.4%**

In summary, from Case I (Model I, II and III) it can be concluded that a width to thickness ratio less than five leads to a significant error in one-dimensional analysis and a larger width of a model does not always mean that the effect of being two-dimensional problem can be ignored. The problem is dependent on the dimension of the applied load too.

CASE II: varying Cv keeping Ch equals to Cv

This case is selected to further identify particular conditions to apply a one-dimensional analysis without a significant error. To have a reasonably accurate one-dimensional analysis, the condition should minimize the effect of lateral drainage. This makes the consolidation due to vertical drainage dominant. A relatively higher value of the vertical coefficient of consolidation increases the rate of consolidation due to vertical drainage. Since Cv is not greater than Ch in most cases, an isotropic condition with Cv equal to Ch is taken to increase the portion of vertical drainage as much as possible. In addition, in the previous case, it is observed that a model with a small layer thickness minimizes the effect of lateral drainage. Hence, the thickness of the soil is taken as 3m in Model 4 and 5m in Model 5.

Model 4: Ch/Cv = 1, Z=3 m, Cv=Ch = 1 m^2/yr and 3 m^2/yr

This model helps to identify the effect of variation of coefficient of consolidation on the accuracy of the one-dimensional analysis. The result of the analysis of the model is presented in the following figure:

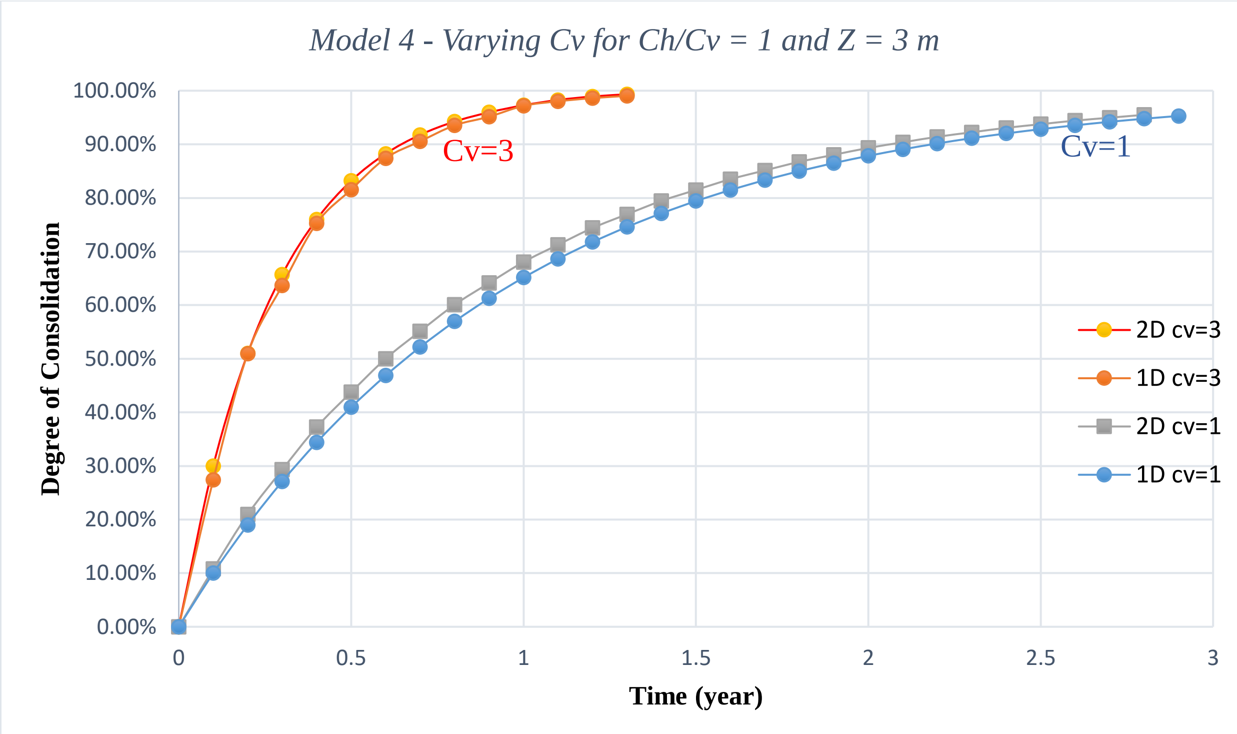


Figure 4-4: Model 4 - Varying Cv for Ch/Cv = 1 and Z = 3 m

It can be seen that the gap between one-dimensional and two-dimensional analysis seems closer for higher values of the consolidation coefficient. However, if one looks a detail on the quantitative values of the points, the gap actually did not get closer rather it proportionally shorten with the consolidation time. The following table clearly shows the time slippage in percent which is similar for the cases.

Table 4-5: Model 4 - Varying Cv for Ch/Cv = 1, X/Z=5, and Z = 3 m

Model 4	**Z= 3.0 m**	**Cv = 1**	**Cv = 3**
t_{50} **(yr.)**	2D Analysis	0.60	0.18
	1D Analysis	0.68	0.20
	Time Slippage	**13.3%**	**11.1%**
t_{90} **(yr.)**	2D Analysis	2.10	0.68
	1D Analysis	2.20	0.70
	Time Slippage	**4.8%**	**2.9%**

From this model, it can be observed that a higher coefficient of consolidation results in a shorter primary consolidation period, which is expected, and a close output between one-dimensional analysis and two – dimensional analysis.

Model 5: Cv = 1, 3 Ch/Cv = 1,Z=5

This model is similar to Model 4 except the layer thickness is changed to 5 m to study the effect of variation of coefficient of consolidation on different layer thickness.

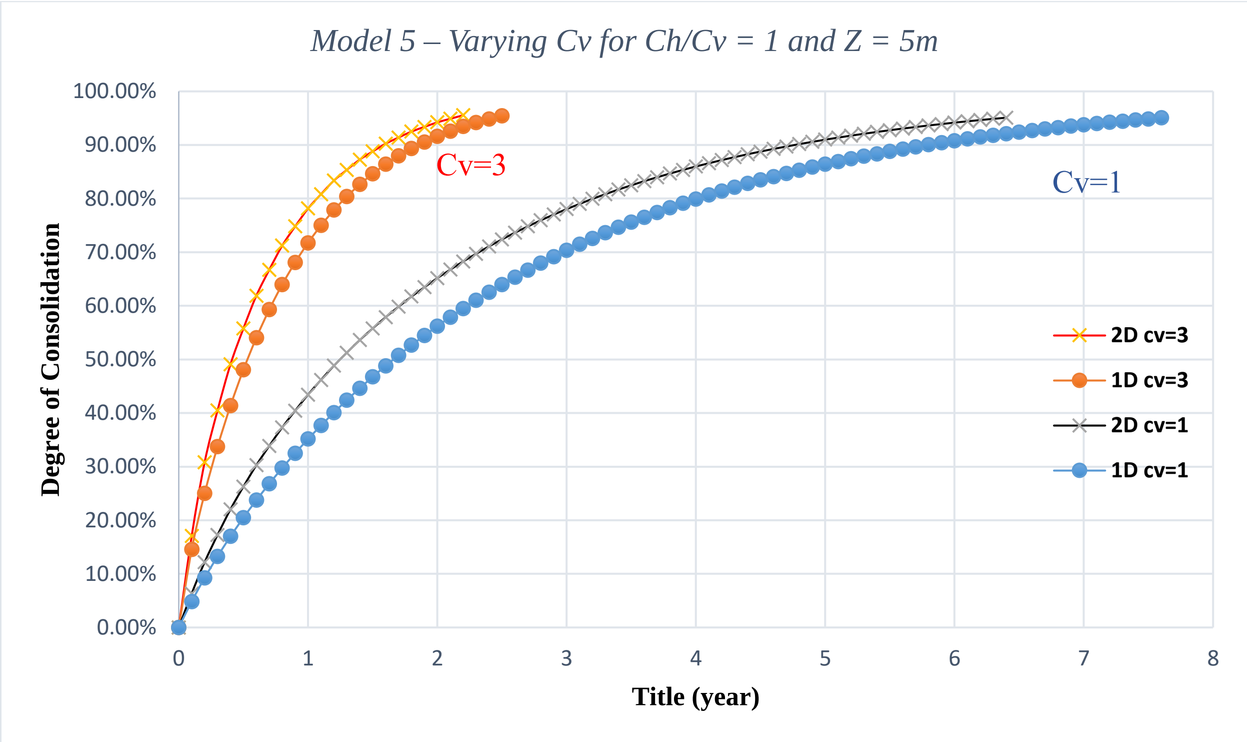


Figure 4-5: Model 5 – Varying Cv for Ch/Cv = 1 and Z = 5 m

Similar result as model 4 is obtained from model 5, that helps to conclude a higher coefficient of consolidation results a closer result between the one-dimensional analysis and two-dimensional analysis. The reason behind is a higher consolidation coefficient leads to a shorter time of complete consolidation which in turn proportionally shorten the gap between the graphs.

Table 4-6: Model 5 – Varying Cv for Ch/Cv = 1, X/Z =5 and Z = 5 m

Model 5	**Z= 5.0m**	**Cv = 1**	**Cv = 3**
t_{50} (yr.)	2D Analysis	1.25	0.41
	1D Analysis	1.70	0.57
	Time Slippage	**36.0%**	**39.0%**
t_{90} (yr.)	2D Analysis	4.90	1.60
	1D Analysis	5.80	1.90
	Time Slippage	**18.4%**	**18.8%**

Percentage of time slippage presented in Table 4-6, clearly shows that the gap between the graphs remains unchanged for different coefficients of consolidations. Therefore, the accuracy of one-dimensional analysis does not depend on the coefficient of consolidation rather on the ratio Ch/Cv. By observing the behaviors of the output result from the one-dimensional and two-dimensional analysis for different parameters, the condition that can be well estimated with one-dimensional analysis without large error can be identified. It has been observed that, a one-dimensional consolidation analysis results in a good agreement with two-dimensional consolidation analysis for small consolidating layer thickness, large width to thickness ratio, and isotropic medium (Cv = Ch). Models that do not fulfill the above conditions should be analyzed using the two-dimensional analysis.

An example of the above-mentioned condition is presented in the following figure, to illustrate the closeness of the graphs between one-dimensional and two-dimensional analysis.

The following parameters that fulfill the stipulated conditions are selected for illustration,

Z = 3, X/Z = 5, Ch/Cv = 1, Cv = 3 m^2/yr

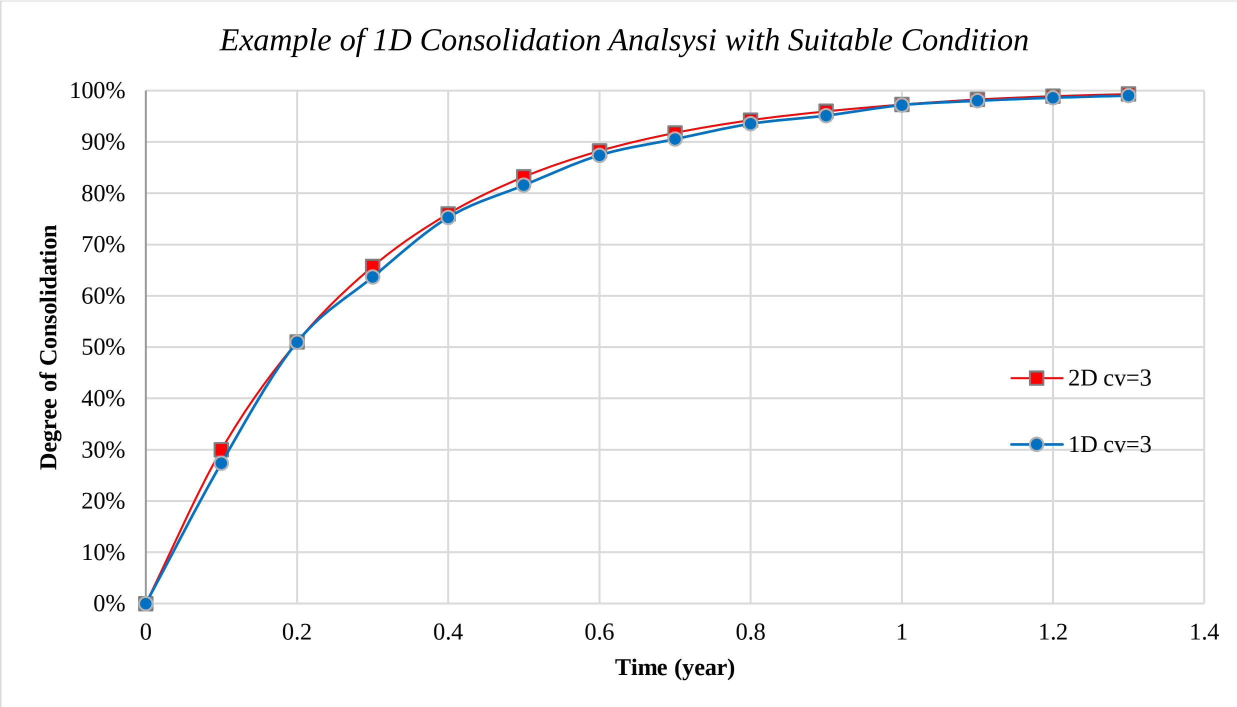


Figure 4-6: Example of 1D Consolidation Analysis with Suitable Condition

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This study presents the development of software for analysis of two-dimensional consolidation using the finite difference approach. To model two-dimensional problems as real as possible, the finite difference approach has been performed in two-dimensional directions. Based on the results obtained, the following conclusions are drawn:

- The method used which is alternating directional implicit (ADI) is a very stable finite difference method and it is convergent during the analysis process.
- Analysis of consolidation as one dimensional significantly underestimates the rate of dissipation of excess pore water pressure except for few cases as described below.
- A one-dimensional consolidation analysis gives a reasonably close estimate and with the two – dimensional consolidation analysis can be used with confident only if the following conditions are satisfied:
  - Large width to thickness ratio, $X/Z>5$
  - Small layer thickness, $Z < 5$ m
  - Soil with good permeability or the coefficient of consolidation, $C_v > 3.0\ m^2/yr$
  - Soil having an isotropic property, $C_v \sim C_h$
- Generally, a one-dimensional analysis is not recommended for two-dimensional consolidation problems where the width to thickness ratio is less than five ($X/Z < 5$) irrespective of all the other parameters.

5.2 Recommendations

In doing the research, the following points are observed as gaps in the scope of this research thesis, and the following recommendations that could expand and strengthen the application of this research are suggested for further work:

I. It is highly recommended to implement a 2D consolidation analysis with the help of the developed software for 2D problems like embankment fills.

II. In-depth exploration of the effect of lateral strain on the rate of consolidation and how it influences the two-dimensional approach is recommended for future research.

III. More methodological work is needed to further decrease the errors on estimating the degree of primary consolidation by eliminating the assumptions made in consolidation analysis such as constant coefficient of consolidation and linear relation between

IV. This research can also be extended to a 3D consolidation analysis and analysis software can be developed on the same for 3D consolidation problems like soil fill with sand drain.

REFERENCES

Biot, M. A. (1941). General Theory of Three-Dimensional Consolidation. *Journal of applied physics, 12*(2).

Budhu, M. (2011). *Soil Mechanics and Foundations.* John Wiley & Sons, inc. .

Carrillo, N. (1942). Simple two and three dimension cases in the theory of consolidation of soils. *Journal of Mathematics and Physics, 21*(1).

Das, B. M. (2008). *Advanced Soil Mechanics* (3rd ed.). NEw York: Taylor & Francis Group.

Fowler, T. (2020, April). *SOFTWARE ENGINEERING*. (Career Karma) Retrieved June 6, 2020, from https://careerkarma.com/blog/how-many-coding-languages-are-there/

Francesco, R. D. (n.d.). *Exact Solutions of Two-dimensional and Tri-dimensional Consolidation Equations.* Teramo, Italy: GEO&GEO Instruments.

Haase, M., Mario, E., & Uwe Reichel. (n.d.). *2D Modeling of the consolidation of soft soils.* Germany.

Hassan, W. N., & HishamMohamad. (2012). Evaluation of Finite Element Formulation for One-Dimensional Consolidation. *International Journal of Scientific & Engineering Research, 3*(5).

K. Morton, D. M. (2005). *Numerical Solution of Partial Differential Equations: An Introduction.* Cambridge: Cambridge University Press.

K.R.Arora, D. (2004). *Soil Mechanics and Foundation Engineering.* Delhi: Lomos Offset Press.

Karl Terzaghi, R. B. (1996). *Soil Mechanics in Engineering Practice.* New York: John Wiley & Sons Inc.

Kim, H.-J., & Mission, J. L. (2011). Numerical Analysis of One-Dimensional Consolidation in Layered Clay Using Interface Boundary Relations in Terms of Infinitesimal Strain. *International Journal of Geomechanics, 11*(1), 72-77.

Manal Salem, R. E.-S. (2014). Comparison of Measured and Calculated Consolidation Settlements of Thick Underconsolidated Clay. *Alexandria Engineering Journal, 53*, 107-117.

Mazumder, S. (2016). *Numerical Methods for Partial Differential Equations Finite Difference and Finite Volume Methods.* London: Elsevier Inc.

OBERKAMPF, W. L., & ROY, C. J. (2010). *VERIFICATION AND VALIDATION IN SCIENTIFIC COMPUTING.* New York: Cambridge University Press.

P.E.Vepsàlâine, O. &. (2017). On the stress dependence of consolidation parameters. *International Society for Soil Mechanics and Geotechnical Engineering* .

Passos, W. D. (2010). *Numerical Methods, Algorithms and Tools in C#.* New York: Taylor & Francis Group.

R.Butterfield, P. (2005). *Advanced Geotechnical Analyses: Developments in Soil Mechanics and Foundation Engineering.* London: Elsevier Science Publishers ltd. Retrieved May 2020, from https://www.sciencedirect.com/topics/engineering/secondary-consolidation

Schmertmann. (1970). *Advanced Soil Mechanics I.* SNU Geotechnical and Geoenvironmental Engineering Lab.

Smith, G. (1985). *Numerical Solution of Partial Differential Equations: Finite Difference Methods.* Oxford: Oxford University Press.

Supangkat, H. (1994). *On Finite Element Analysis of Nonlinear Consolidation.* Massachusetts Institute of Technology.

Tandjiria, V. (1999). DEVELOPMENT OF FINITE DIFFERENCE METHOD APPLIED TO CONSOLIDATION ANALYSIS OF EMBANKMENTS. *DIMENSI TEKNIK SIPIL, 1*(2).

Tewatia, S. K. (2013). Equation of 3D consolidation in cartesian coordinates. *International Journal of Geotechnical Engineering, 7*(1).

Tewatia, S. K. (2015). Principle of Super Position of Rate of Settlement in 2D and 3D Consolidations. *Geotechnical and Geological Engineering, 33*.

Ullrich, C. (1990). *Computer Arithmetic and Self-Validating Numerical Methods* . San Diego: Academic Press, Inc.

Wei Zhang, X.-f. W. (2013). Analysis on the Degree of Consolidation Based on Measured Settlements. *Electronic Journal of Geotechnical Engineering, 18*, 5441-5448.

Widipedia. (n.d.). (Widipedia) Retrieved January 28, 2020, from https://en.wikipedia.org/wiki/Superposition_principle

Wikipedia. (2020, 3 7). *Explicit and implicit methods*. Retrieved 4 29, 2020, from https://en.wikipedia.org/wiki/Explicit_and_implicit_methods

Youcef Houmadi, A. A.-H. (2012). Probabilistic analysis of a one-dimensional soil consolidation problem. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards, 6*(1), 36-49.

ANNEXES

ANNEX I - GUIDELINE AND ILLUSTRATIVE EXAMPLES

Calculating Degree of Consolidation Using 1D Analysis

Start *Consolidation 2D.exe* from the installation folder. Then change the active tab to one - Dimensional consolidation. The following window will display.

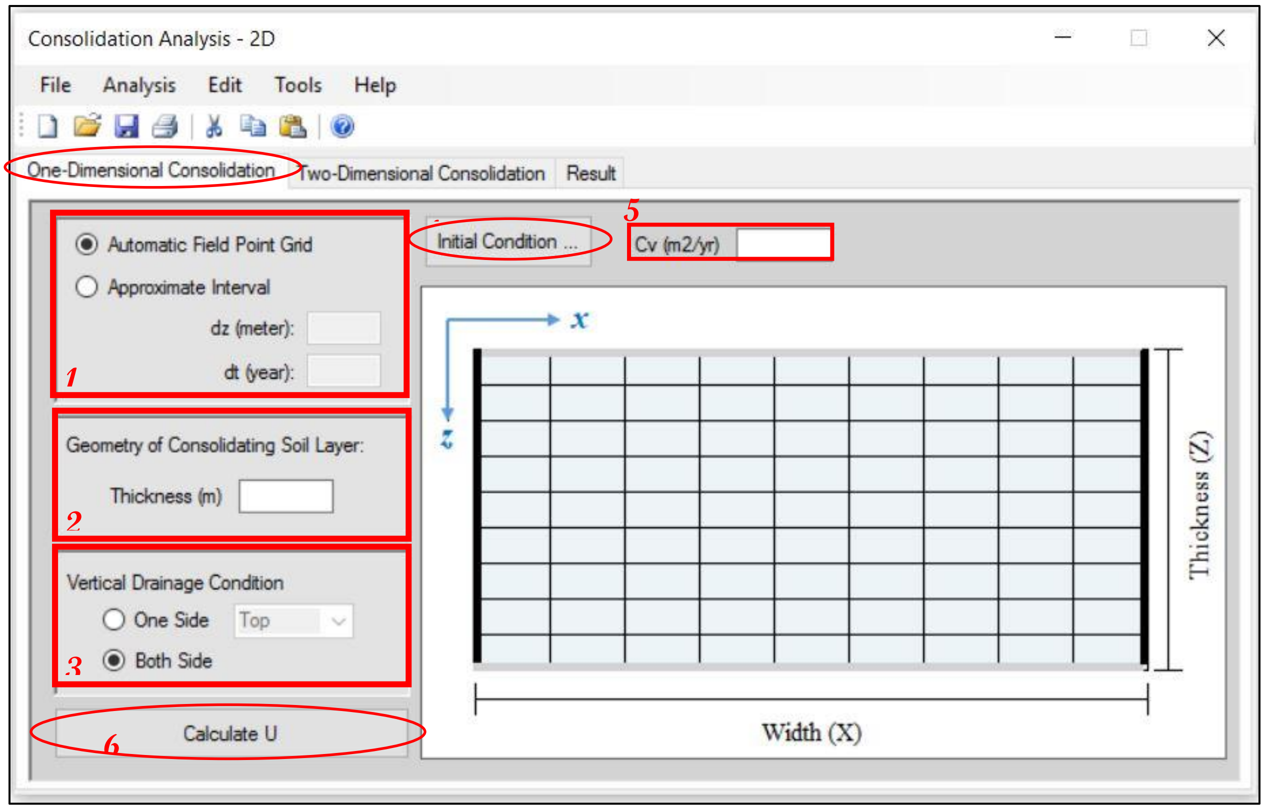


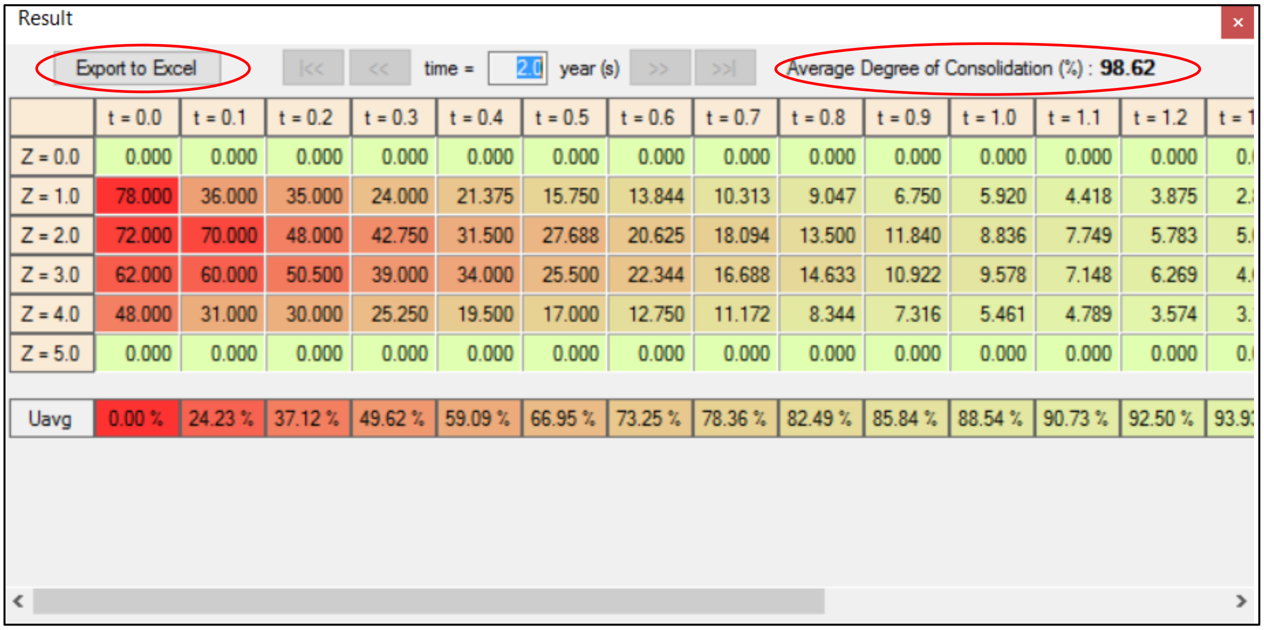
Figure A-5-1: Consolidation Analysis 1D Main Window

1. Select field point property from the top left panel
   a. Select **automatic field point grid** to let the program select the number of grid points for space and time to have good precision.
   b. Select **approximate interval:** then **the user** is required to enter the grid point intervals, **dz** and **dt.** If this option is selected the software may round to the nearest possible interval to have an equal grid point interval for the model.
2. Then enter **thickness** value in **meters** from the geometry of consolidating soil layer panel in the left.
3. Select a **vertical drainage condition**
   a. One side -> either top or bottom
   b. Or both sides

4. Click **on Initial condition** button to set the initial pore water distribution equation
5. Enter **Cv** value in **square meter per year** in the field at the top right.
6. Click **calculate U** to compute the results.

Result window for 1D consolidation

When the user clicks **Calculate U** button, a new window will appear containing the results. The **Average Degree of consolidation** is displayed at the top right of the window.



	t = 0.0	t = 0.1	t = 0.2	t = 0.3	t = 0.4	t = 0.5	t = 0.6	t = 0.7	t = 0.8	t = 0.9	t = 1.0	t = 1.1	t = 1.2	t = 1
Z = 0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.
Z = 1.0	78.000	36.000	35.000	24.000	21.375	15.750	13.844	10.313	9.047	6.750	5.920	4.418	3.875	2.
Z = 2.0	72.000	70.000	48.000	42.750	31.500	27.688	20.625	18.094	13.500	11.840	8.836	7.749	5.783	5.
Z = 3.0	62.000	60.000	50.500	39.000	34.000	25.500	22.344	16.688	14.633	10.922	9.578	7.148	6.269	4.
Z = 4.0	48.000	31.000	30.000	25.250	19.500	17.000	12.750	11.172	8.344	7.316	5.461	4.789	3.574	3.
Z = 5.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.
Uavg	0.00 %	24.23 %	37.12 %	49.62 %	59.09 %	66.95 %	73.25 %	78.36 %	82.49 %	85.84 %	88.54 %	90.73 %	92.50 %	93.9

Figure A-5-2: Result Window for 1D Analysis

This window contains **U** values of different times across the row and **U** values of different thicknesses down the column.

From the result window, it is possible to export the results to excel just by clicking the **Export to Excel** button at the top left of the window.

Calculating Degree of Consolidation Using 2D Analysis

Start ***Consolidation 2D.exe*** from the installation folder. The ***Two-Dimensional*** consolidation tab is active on the start. Here the user has to enter values in the required fields.

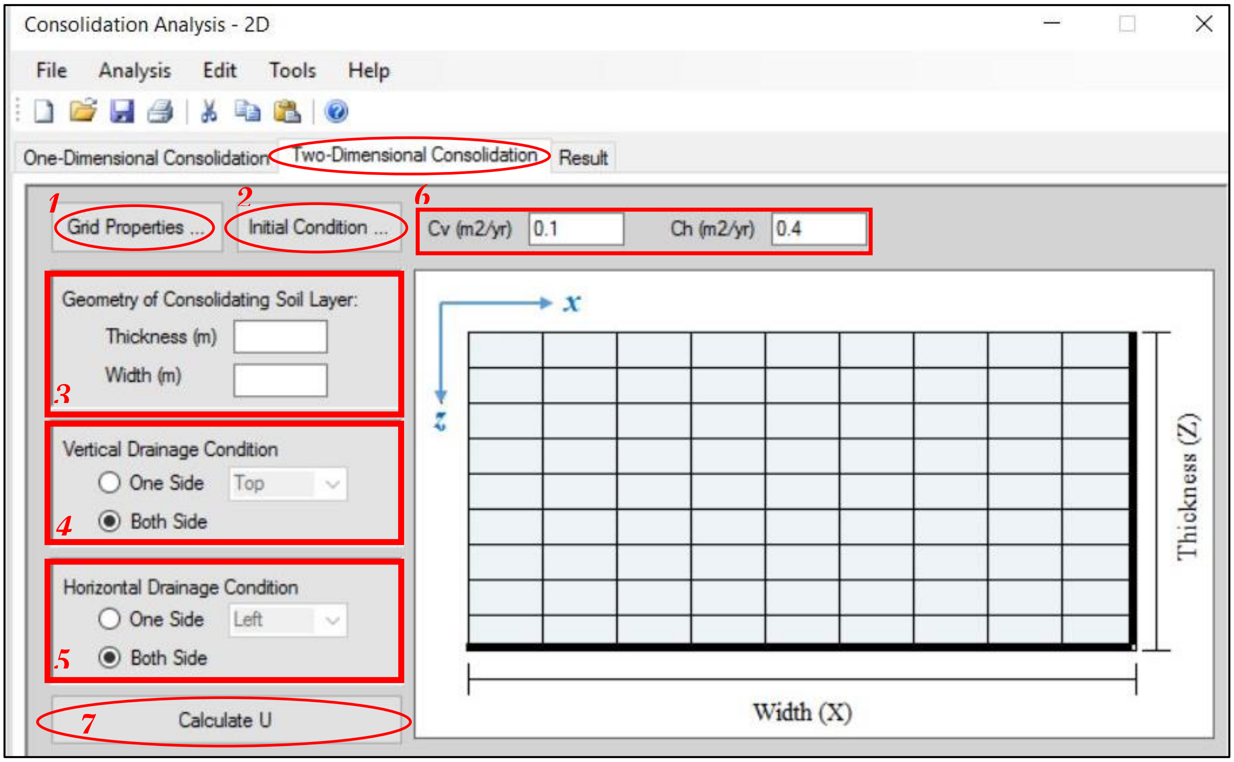


Figure A-5-3: Consolidation 2D main screen

1. Click on **Grid properties** at the top left to set the grid properties to
   a. Select ***Automatic field point grid*** to let the program select number of grid points for space and time to have good precision.
   b. The ***Approximate number of grid points*** option lets the user enter number of grid points which are **X** and **Z** in **meters**.
   c. Select **Approximate interval:** then **the user** is required to enter the grid point intervals **dx, dz,** and **dt.** If this option is selected the software may round to the nearest possible interval to have an equal grid point interval for the model.
   d. Then click **close**.

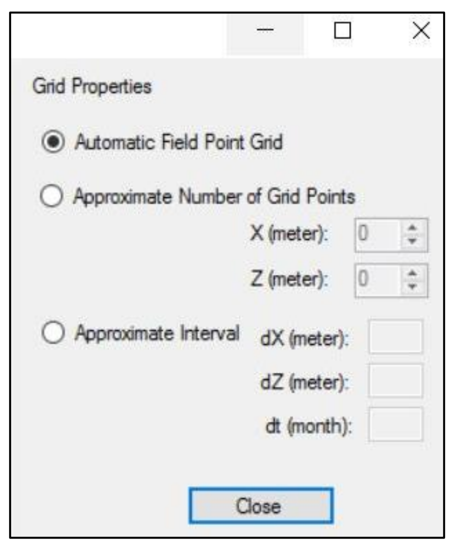
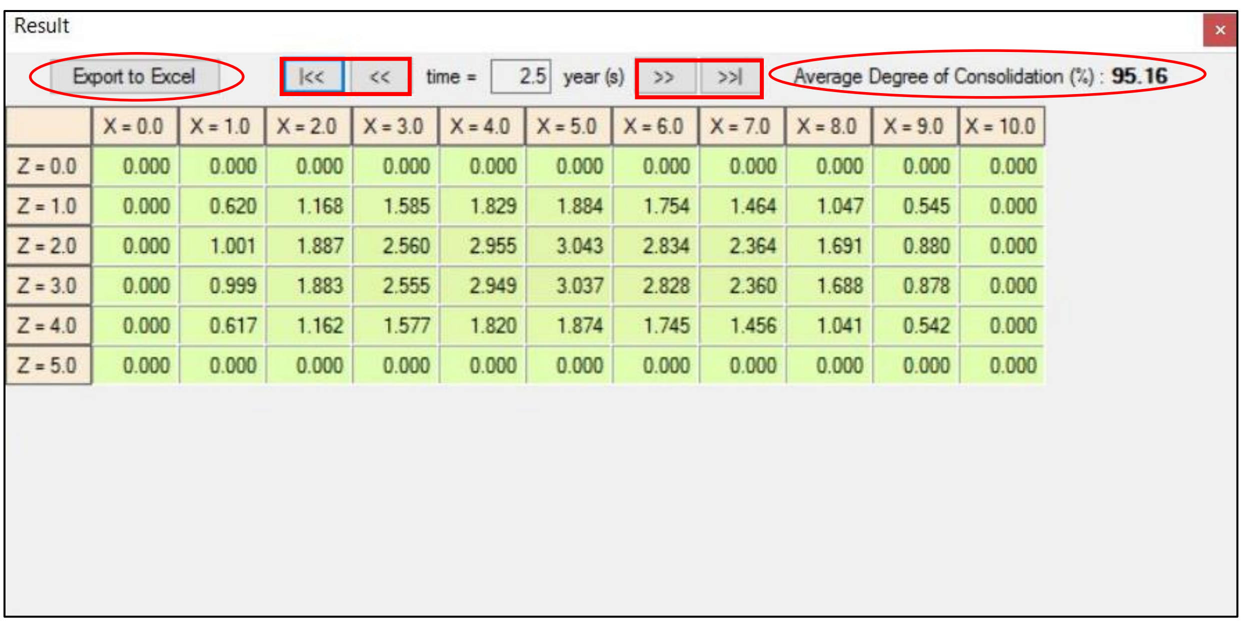


Figure A-5-4: Grid properties for 2D consolidation

2. Click **on Initial condition** button to set the initial pore water distribution equation
3. Under **Geometry of consolidating soil layer** panel enter the following properties: **thickness** and **width** in **meters**
4. Select a **vertical drainage** condition
   a. One side: choose either **top** or **bottom** from the corresponding combo box
   b. Or both sides
5. Select a **horizontal drainage** condition
   a. One side: choose either **left** or **right** from the corresponding combo box.
   b. Or both sides
6. Enter values for **Cv** and **Ch** in **meter^2/ year** in the field at the top right.
7. Click on **calculate U** to see the results.

Result window 2D consolidation

In the result window, the user can easily navigate through the desired year or time by using the navigation buttons at the top. "|<<" or ">>|" lets the user go to the initial pore water distribution or the pore water pressure distribution after the primary consolidation ends respectively. "<<" or ">>" makes a time unit forward/backward in one interval. The user can also export the result to MS Excel just by clicking **Export to Excel button.** The **Average Degree of consolidation** is displayed at the top right of the window.



	X = 0.0	X = 1.0	X = 2.0	X = 3.0	X = 4.0	X = 5.0	X = 6.0	X = 7.0	X = 8.0	X = 9.0	X = 10.0
Z = 0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Z = 1.0	0.000	0.620	1.168	1.585	1.829	1.884	1.754	1.464	1.047	0.545	0.000
Z = 2.0	0.000	1.001	1.887	2.560	2.955	3.043	2.834	2.364	1.691	0.880	0.000
Z = 3.0	0.000	0.999	1.883	2.555	2.949	3.037	2.828	2.360	1.688	0.878	0.000
Z = 4.0	0.000	0.617	1.162	1.577	1.820	1.874	1.745	1.456	1.041	0.542	0.000
Z = 5.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure A-5-5: Result Window for 2D Consolidation Analysis

ANNEX II – PROGRAM CODES OF MAJOR FUNCTIONS

```
using System;
using System.Data;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;

namespace Consolidation3D
{
    public partial class Form1 : Form
    {

        private void button1_Click(object sender, EventArgs e)
        {
            //Declare Variables
            double FH, FT, Cv;

            int n, m; //user input for no of devisions
            int N, M; //No of points calculated from no of devision (No of devision +1)

            bool drainageTop, drainageBottom;

            double h, k;

            Matrix U;

            //Read division from input
            m = 5;     //division for thickness Z ...  to be considered resonably
            n = 5;     //division for time T .... both N and T to be considered later,

            //Compute No of points from devision
            M = m + 1;     //division for thickness Z ...  to be considered resonably
            N = n + 1;     //division for time T .... both N and T to be considered later

            //Read Drainage type from User Interface
            drainageTop = true;
            drainageBottom = false;

            //Input from User Interface
            FH = 5; //Thickness of the layer in m
            FT = 0.5; //Total time in yr
            Cv = 2.52; //Cv in m2/yr

            //Initial Porewater distribution
            //Option for the user, trig or poly

            h = FH / m;
            k = FT / n;

            U = new Matrix(M, N);

            //Initial Condition at t=0

            for (int i = 0; i < M; i++)
            {
```

```
                U[i, 0] = functionPW(i * h, 2,FH);  //functionPW is initial Pore water
pressure distribution 2 for both side drainage
            }
            if (drainageTop) U[0, 0] = 0;

            if (drainageBottom) U[M - 1, 0] = 0;


            U = Solve_PDE2(U, FH, FT, Cv, M, N, drainageTop, drainageBottom);
///Access Column from the Matrix excluding the boundary layers

            Console.WriteLine(U);
        }
        static double functionPW(double z, int vertDrainageFace, double thickness)
//initial pore water distribution
        {
            //return Math.Sin(Math.PI * x);

            double u0 = 80 - 2 * z * z;
            if (((vertDrainageFace == 0) | (vertDrainageFace == 2)) & (z == 0)) u0 = 0;
            else if (vertDrainageFace == 1 | vertDrainageFace == 2 & z == thickness) u0 =
0;
            return u0;
        }
        static double functionPW(double z, double x, int horizontalDrainageFace, int
vertDrainageFace)
        {
            double delU = 80 - x*x - 2* z * z;
            //if (i==0|k==0|i)    //check boundary conditions are not initialized

            return delU;
        }

        public int MaxOrder = 10;

        public double[,] a;  // main Matrix
        public double[] d;   // solution vector
        public double[,] b;  // inverse Matrix

        //Grid Data from Dialog Box
        private int radioBox=0;
        private int numberOfGridX=0;
        private int numberOfGridZ=0;
        private double intervalSpaceX=0;
        private double intervalSpaceZ=0;
        private double intervalt=0;

        //BOUNDARY CONDITIONS
        private int horizontalDrainageFace;
        private int verticalDrainageFace;
        private double zThickness;
        private double xWidth;

        //SOIL PROPERTIES
        private double cv = 0.08; //vertical coefficient of consolidation in m2/year
        private double ch = 0.4; //horizontal coefficient of consolidation  //check
        private Matrix U0; //initial pore water distribuition
```

```
        double [, ,] u;
        double [,] u_1D;
        double[] u_avg;
        double[] u_avg_1D;

        static int numberOfGridt;  //maxnumber of time grid
        private double maxTime=2;  //end of consolidation in month //check

        //
        static double[] sumPorePressure;
        static int maxIteration = 200;

        //for 1D Consolidation
        private double[] U0_1D;        //U0_1D = new double[numberOfGridZ];
        double[,] solve_U1D(double []U0_1D, double c_v, double dZ, double dt, int
vertDrainageFace)
        {
            double[,] u_1 = new double [maxIteration,U0_1D.Length];
            double[] uAverage = new double[maxIteration];
            //computation for U_1D here
            double lambda = c_v * dt / (dZ * dZ);
            if (lambda > 0.5)
            {
                MessageBox.Show("interval is too large, may cause convergence issue!
please revise your inputs.");
                return u_1;
            }

            double[] currU = new double[U0_1D.Length];

            Matrix A = new Matrix(U0_1D.Length - 2);
            for (int j = 0; j < currU.Length; j++)
            {
                u_1[0,j] = U0_1D[j];

            }

            //[A]{u} = {b}    solve for this
            //Generate Matrix [A]
            for (int i = 0; i < A.ColumnLength; i++)
            {
                A[i, i] = 1 + 2 * lambda;
                if (i > 0) A[i, i - 1] = -1 * lambda;
                if (i < A.ColumnLength - 1) A[i, i + 1] = -1 * lambda;
            }

            //for i from 1 to iterationfortime
            //compute for b and solve for u_new
            //add u_new to u_1

            uAverage[0] = 1;
            //to be modified
            for (int j = 1; j < numberOfGridt; j++)
            {
                //b = U.getColumnSection(j - 1, 1, M2);
```

```
                //U.replaceColumnSection(j, 1, M2, Matrix.Ax_bSolve(A, b)); //Solve A*x_j
= B_j-1

                for (int p = 0; p < u_1.GetLength(1); p++)
                {
                    if (p == 0 )
                    {
                        if (vertDrainageFace == 1)  //No top darainage
                        {
                            u_1[j, p] = u_1[j - 1, p] + lambda * (u_1[j - 1, p + 1] - 2 *
u_1[j - 1, p] + u_1[j - 1, p + 1]);
                            if (u_1[j, p] < 0) u_1[j, p] = 0;
                        }
                        else
                        {
                            u_1[j, p] = 0;
                        }
                    }
                    else if (p == u_1.GetLength(1) - 1)
                    {
                        if (vertDrainageFace == 0)  //No bottom darainage
                        {
                            u_1[j, p] = u_1[j - 1, p] + lambda * (u_1[j - 1, p - 1] - 2 *
u_1[j - 1, p] + u_1[j - 1, p - 1]);
                            if (u_1[j, p] < 0) u_1[j, p] = 0;
                        }
                        else
                        {
                            u_1[j, p] = 0;
                        }
                    }
                    else
                    {
                        u_1[j, p] = u_1[j - 1, p] + lambda * (u_1[j - 1, p - 1] - 2 *
u_1[j - 1, p] + u_1[j - 1, p + 1]);
                        if (u_1[j, p] < 0) u_1[j, p] = 0;
                    }
                }

                uAverage[j] = getSum(u_1, j) / getSum(u_1, 0);
                if (uAverage[j] > .05 & j == numberOfGridt - 1) { if (numberOfGridt <
maxIteration-1) numberOfGridt++; }

            }
            maxTime = (numberOfGridt - 1) * intervalt;
            u_avg_1D =new double[numberOfGridt];
            for (int x = 0; x < numberOfGridt;x++ )
            {
                u_avg_1D[x] = uAverage[x];
            }
                return u_1;
        }

        private double getSum(double[,] array, int index)
        {
            double sum=0;
```

```
            for (int i = 0; i < array.GetLength(1); i++) sum += array[index,i];

            return sum;
        }
        static Matrix column(double rx, Matrix u0, Matrix Az)
        {
            int n = u0.RowLength - 2;
            int m = u0.ColumnLength - 2;
            double[,] b = new double[n, m];
            double[] B = new double[n];
            double[] X = new double[n];
            Matrix u1 = new Matrix(n + 2, m + 2);
            for (int i = 0; i < m + 2; i++) { u1[0, i] = u1[n + 1, i] = 0; }
            //for (int i = 0; i < n + 2; i++) { u1[i, m + 1] = 0; }   //check

            for (int i = 1; i < n + 1; i++)
            {
                for (int j = 1; j < m + 1; j++)
                {
                    b[i - 1, j - 1] = (rx * u0[i, j-1]) + ((1 - 2 * rx) * u0[i, j]) + (rx
* u0[i, j+1]);
                }
            }

            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    B[j] = b[j, i];
                }
                X = Matrix.Ax_bSolve(Az, B);
                for (int j = 0; j < n; j++)
                {
                    u1[j + 1, i + 1] = X[j];
                }
            }

            return u1;

        }

        static Matrix row(double rz, Matrix u0, Matrix Ax)
        {
            int n = u0.RowLength - 2;
            int m = u0.ColumnLength - 2;
            double[,] b = new double[n, m];
            double[] B = new double[m];
            double[] X = new double[m];
            Matrix u1 = new Matrix(n + 2, m + 2);
            for (int i = 0; i < m + 2; i++) { u1[0, i] = u1[n + 1, i] = 0; }
            //for (int i = 0; i < n + 2; i++) { u1[i, m + 1] = 0; }  //check

            for (int i = 1; i < n + 1; i++)
            {
                for (int j = 1; j < m + 1; j++)
                {
```

```
                    b[i - 1, j - 1] = (rz * u0[i-1, j]) + ((1-2 * rz) * u0[i, j]) + (rz *
u0[i+1, j]);
                }
            }

            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < m; j++)
                {
                    B[j] = b[i, j];
                }
                X = Matrix.Ax_bSolve(Ax, B);
                for (int j = 0; j < m; j++)
                {
                    u1[i + 1, j + 1] = X[j];
                }
            }

            return u1;

        }

        double[,,] solve_U(Matrix U0, double C_v, double C_h, double intervalSpaceX,
double intervalSpaceZ, double intervalt, int horizontalDrainage, int verticalDrainage)
        {

            double[,,] u = new double[maxIteration, U0.RowLength, U0.ColumnLength];
            double[] uAverage = new double[maxIteration];
            double rx = C_h * intervalt / (intervalSpaceX * intervalSpaceX);
            double rz = C_v * intervalt / (intervalSpaceZ * intervalSpaceZ);
            double [,] u1 = new double[U0.RowLength, U0.ColumnLength];  //current u

            Matrix Az = new Matrix(U0.RowLength - 2);
            Matrix Ax = new Matrix(U0.ColumnLength - 2);

            for (int i = 0; i < u.GetLength(1); i++)
            {
                for (int j = 0; j < u.GetLength(2); j++)
                {
                    u[0, i, j] = U0[i, j];
                }
            }

            //[A]{u} = {b}    solve for this
            //Generate Matrix [A]
            for (int i = 0; i < Ax.ColumnLength; i++)
            {
                Ax[i, i] = 1 + 2 * rx;
                if (i > 0) Ax[i, i - 1] = -1 * rx;
                if (i < Ax.ColumnLength - 1) Ax[i, i + 1] = -1 * rx;
            }

            for (int i = 0; i < Az.ColumnLength; i++)
            {
                Az[i, i] = 1 + 2 * rz;
                if (i > 0) Az[i, i - 1] = -1 * rz;
                if (i < Az.ColumnLength - 1) Az[i, i + 1] = -1 * rz;
```

```
            }
            numberOfGridt = (int) (maxTime / intervalSpaceX);
            for (int j = 1; j < numberOfGridt; j++)
            {
                if(j%2 ==0)
                {
                    U0 = row(rz, U0, Ax);  //row operation
                    //U0 = column(rx, U0, Az);
                }
                else
                {
                    U0 = column(rx, U0, Az); //column operation
                    // U0 = row(rz, U0, Ax); //check
                }
                sumPorePressure[j] = 0;
                for (int o=0;o<U0.RowLength;o++)
                {
                    for (int p=0;p<U0.ColumnLength;p++)
                    {
                        u[j, o, p] = U0[o, p];
                        sumPorePressure[j] += U0[o, p];
                    }
                }
                uAverage[j]=sumPorePressure[j] / sumPorePressure[0];
                if (( uAverage[j]> .05)&j==numberOfGridt-1) { if (numberOfGridt <
maxIteration-1) numberOfGridt++; }

            }
            maxTime = (numberOfGridt -1)* intervalt;
            u_avg = new double[numberOfGridt];
            for (int x = 1; x < numberOfGridt; x++)
            {
                u_avg[x] = 1-uAverage[x];
            }
                return u;
        }
        double CalcDet(double[,] m, int order)
        {
            int i, j, k;
            double sign = 1.0;
            double dm = 0;
            if (order <= 2)
            {
                return m[0, 0] * m[1, 1] - m[1, 0] * m[0, 1];
            }
            else
            {
                double[,] mm = new double[order - 1, order - 1]; // temporary matrix
                for (i = 0; i < order; i++) // index for the element of the top row
                {
                    for (j = 0; j < order - 1; j++) // index that runs down from second
row to n
                    {
                        for (k = 0; k < order - 1; k++) // index to copy the elements
into a sub matrix
                        {
                            if (j < i)
```

```
                        mm[j, k] = m[j, k + 1];
                    else
                        mm[j, k] = m[j + 1, k + 1];
                }
            }
            dm = dm + sign * CalcDet(mm, order - 1) * m[i, 0];
            sign = -sign;
        }
        return dm;
    }
}


public void button6_Click(object sender, EventArgs e)
{

    GrdProperties frm = new GrdProperties();
    frm._intervalSpaceX = intervalSpaceX;
    frm._intervalSpaceZ = intervalSpaceZ;
    frm._intervalt = intervalt;
    frm._numberOfGridX = numberOfGridX;
    frm._numberOfGridZ = numberOfGridZ;
    frm._radioBox = radioBox;

    frm.ShowDialog();

    intervalSpaceX=frm._intervalSpaceX;
    intervalSpaceZ=frm._intervalSpaceZ;
    intervalt=frm._intervalt;
    numberOfGridX=frm._numberOfGridX;
    numberOfGridZ=frm._numberOfGridZ;
    radioBox=frm._radioBox;
}
private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
{
    const int BACKSPACE = 8;
    const int DECIMAL_POINT = 46;
    const int ZERO = 48;
    const int NINE = 57;
    const int NOT_FOUND = -1;

    int keyvalue = (int)e.KeyChar; // not really necessary to cast to int

    if ((keyvalue == BACKSPACE) ||
    ((keyvalue >= ZERO) && (keyvalue <= NINE))) return;
    // Allow the first (but only the first) decimal point
    if ((keyvalue == DECIMAL_POINT) &&
    (textBoxWidth.Text.IndexOf(".") == NOT_FOUND)) return;
    // Allow nothing else
    e.Handled = true;
}

private void button8_Click(object sender, EventArgs e)
{
    //INPUTS
    //zThickness = thickness of the consolidating layer
    //xWidth = width of the consolidating layer
```

```
            //horizontalDrainageFace = 0 if drainage face is on left, 1 for right, 2 if
two sidedrainage
            //verticalDrainageFace = 0 if drainage face is on top, 1 if bottom, 2 if two
sidedrainage

            bool err = false;
            err = (!double.TryParse(textBoxThickness.Text, out zThickness) | zThickness <
2)|
                  (!double.TryParse(textBoxWidth.Text, out xWidth) | xWidth < 2)|

                  (!double.TryParse(textBoxCv.Text, out cv) | cv == 0)|
                  (!double.TryParse(textBoxCh.Text, out ch) | ch == 0);

            if (err) MessageBox.Show("Please inpute a non zero number for all inputs");

            //intialize Grid
            if (radioBox==3)            //Input from user is Approximate Interval
            {
                numberOfGridX = (int) (xWidth / intervalSpaceX)+1;
                numberOfGridZ = (int) (zThickness / intervalSpaceZ)+1;
                numberOfGridt = (int) (maxTime / intervalt)+1;

                //update intervals based number of ongrid
                intervalSpaceX = xWidth / (numberOfGridX - 1);
                intervalSpaceZ = zThickness / (numberOfGridZ - 1);
                intervalt = maxTime / (numberOfGridt - 1);
            }
            else if (radioBox == 2)    //Input from user is Approximate No of Grid
            {
                //Calculate intervals
                intervalSpaceX = xWidth / (numberOfGridX - 1);
                intervalSpaceZ = zThickness / (numberOfGridZ - 1);
                intervalt = maxTime / (numberOfGridt - 1);
            }
            else                      //Use Automatic Griding
            {
                intervalSpaceX = 1;
                intervalSpaceZ = 1;
                intervalt = 0.1;

                numberOfGridX = (int)(xWidth / intervalSpaceX) + 1;
                numberOfGridZ = (int)(zThickness / intervalSpaceZ) + 1;
                numberOfGridt = (int)(maxTime / intervalt) + 1;

                //update intervals based number of ongrid
                intervalSpaceX = xWidth / (numberOfGridX - 1);
                intervalSpaceZ = zThickness / (numberOfGridZ - 1);
                intervalt = maxTime / (numberOfGridt - 1);
            }


            if (radioButtonH2.Checked) { horizontalDrainageFace = 2; }
            else { horizontalDrainageFace = comboBoxHorizontal.SelectedIndex; }

            if (radioButtonV2.Checked) { verticalDrainageFace = 2; }
            else { verticalDrainageFace = comboBoxVertical.SelectedIndex; }
```

```
            //inintialize U0
            U0 = new Matrix(numberOfGridZ,numberOfGridX);
            sumPorePressure = new double[maxIteration];
            sumPorePressure[0] = 0;
            for (int i = 0; i < numberOfGridZ; i++ )
            {
                for (int k=0;k<numberOfGridX;k++)
                {

                    U0[i, k] = functionPW(i*intervalSpaceZ, k*intervalSpaceX,
horizontalDrainageFace, verticalDrainageFace);
                    if (horizontalDrainageFace == 2)
                    {
                        if (k == 0 | k == numberOfGridX - 1) U0[i, k] = 0;

                    }
                    else if (horizontalDrainageFace == 1)
                    {
                        if (k == numberOfGridX - 1) U0[i, k] = 0;
                    }
                    else if (horizontalDrainageFace == 0)
                    {
                        if (k == 0) U0[i, k] = 0;
                    }

                    if (verticalDrainageFace == 2)
                    {
                        if (i == 0 | i == numberOfGridZ - 1) U0[i, k] = 0;

                    }
                    else if (verticalDrainageFace == 1)
                    {
                        if (i == numberOfGridZ - 1) U0[i, k] = 0;
                    }
                    else if (verticalDrainageFace == 0)
                    {
                        if (i == 0) U0[i, k] = 0;
                    }
                    if (U0[i, k] < 0) U0[i, k] = 0;
                    sumPorePressure[0] += U0[i, k];
                }

            }
                if (err) { }
                else
                {
                    //continue computing

                    u = solve_U(U0, cv, ch, intervalSpaceX, intervalSpaceZ, intervalt,
horizontalDrainageFace, verticalDrainageFace);

                    //assign(u,numberOfGridt-1);
                    Result frmResult = new Result(u, numberOfGridt,intervalt,
intervalSpaceZ,intervalSpaceX, u_avg, maxTime);
                    //GrdProperties frm = new GrdProperties();
                    //tabControl1.SelectedIndex=2;
```

```
                frmResult.Show();
                //textBoxPgNo.Text = string.Format("{0:0.0}", maxTime);
                panel1.Visible = true;
                button2.Enabled = true;
                button3.Enabled = true;
                button4.Enabled = true;
                button7.Enabled = true;
            }

        }

        public double getLargest(double[, ,] arrray)
        {
            double largest = arrray[0, 0, 0];
            for (int k = 0; k < arrray.GetLength(0); k++)
            {
                for (int i = 0; i < arrray.GetLength(1); i++)
                {
                    for (int j = 0; j < arrray.GetLength(2); j++)
                    {
                        if (arrray[k, i, j] > largest)
                            largest = arrray[k, i, j];
                    }
                }
            }
            return largest;
        }
        public double getSmallest(double[, ,] arrray)
        {
            double smallest = arrray[0, 0, 0];
            for (int k = 0; k < arrray.GetLength(0);k++ )
            {
                for (int i = 0; i < arrray.GetLength(1); i++)
                {
                    for (int j = 0; j < arrray.GetLength(2); j++)
                    {
                        if (arrray[k, i, j] < smallest)
                            smallest = arrray[k, i, j];
                    }
                }
            }

            return smallest;
        }
        public double getSum(double[, ,] arrray, int indexOfTime)
        {
            double sum=0;

                for (int i = 0; i < arrray.GetLength(1); i++)
                {
                    for (int j = 0; j < arrray.GetLength(2); j++)
                    {

                            sum += arrray[indexOfTime, i, j];
                    }
                }
```

```
            return sum;
        }

        public void assign1D(double[,] a, double[] Uavg)
        {
            //reportViewer1.SetPageSettings

            table1.Visible = false;

            table1.RowCount = a.GetLength(1) + 2; //one for heading one for average
degree of consolidation
            table1.ColumnCount = numberOfGridt + 1; //one additional for heading
            table1.Controls.Clear();

            ToolTip ttip = new ToolTip();
            Label textBox1;

            int[] rgbMax = { 255, 50, 50 };
            int[] rgbMin = { 225, 255, 176 };
            int[] rgb = new int[3];

            table1.Padding = new Padding(0);

            for (int i = -1; i < a.GetLength(1)+1; i++)
            {
                if (i < a.GetLength(1))
                {
                    for (int j = -1; j < numberOfGridt; j++)
                    {

                        textBox1 = new Label();
                        //Color 1 rgb(255,50,50); for max u
                        //Color 2 rgb(225, 255, 176);  for min u


                        if (i < 0 | j < 0)
                        {
                            textBox1.BackColor = Color.AntiqueWhite;
                            textBox1.Width = 50;
                            textBox1.Margin = new Padding(0);
                            textBox1.TextAlign = ContentAlignment.MiddleCenter;
                            textBox1.BorderStyle = BorderStyle.FixedSingle;

                            if (j >= 0)
                            {
                                textBox1.Text = "t = " + String.Format("{0:0.0}", j *
intervalt);
                            }
                            else if (i >= 0)
                            {
                                textBox1.Text = "Z = " + String.Format("{0:0.0}", i *
intervalSpaceZ);
                            }
                            else
                            {
                                textBox1.Text = "";
                            }
```

```
                        }
                        else
                        {

                            //set alpha from 0 - 1  depending on x, alpha = [(x - A)/(B -
A)]
                            double alpha = (((a[j, i] - getSmallest(a)) / (getLargest(a)
- getSmallest(a))));

                            for (int f = 0; f < 3; f++) rgb[f] = (int)(rgbMin[f] +
(rgbMax[f] - rgbMin[f]) * alpha);

                            //textBox1.Name = "x" + j + "z" + i;
                            textBox1.BackColor = Color.FromArgb(rgb[0], rgb[1], rgb[2]);
                            textBox1.Width = 50;
                            textBox1.Text = String.Format("{0:0.000}", a[j, i]);

                            //textBox1.BackColor = Color.FromArgb(243, 243, 243);

                            textBox1.Margin = new Padding(0);
                            textBox1.TextAlign = ContentAlignment.MiddleRight;
                            textBox1.BorderStyle = BorderStyle.Fixed3D;
                            //textBox1.MouseHover += new
EventHandler(this.MyMouseHoverHandler);

                            //Controls.Add(textBox1);
                            ttip.SetToolTip(textBox1, "t = " + String.Format("{0:0.00}",
j * intervalt) + ", Z = " + String.Format("{0:0.00}", i * intervalSpaceZ));

                        }

                        table1.Controls.Add(textBox1, j + 1, i + 1);

                    }
                }
                else
                {
                    for (int j = -1; j < numberOfGridt; j++)
                    {

                        textBox1 = new Label();
                        //Color 1 rgb(255,50,50); for max u
                        //Color 2 rgb(225, 255, 176);  for min u
                        //textBox1.BackColor = Color.AntiqueWhite;
                        textBox1.Width = 50;
                        textBox1.Margin = new Padding(0,15,0,0);
                        textBox1.TextAlign = ContentAlignment.MiddleCenter;
                        textBox1.BorderStyle = BorderStyle.FixedSingle;

                        if (j < 0)
                        {
                                textBox1.Text = "Uavg";
                        }
                        else
                        {
```

```
                            //set alpha from 0 - 1  depending on x, alpha = [(x - A)/(B -
A)]
                            double alpha = getSum(a, j) / getSum(a, 0);

                            for (int f = 0; f < 3; f++) rgb[f] = (int)(rgbMin[f] +
(rgbMax[f] - rgbMin[f]) * alpha);

                            //textBox1.Name = "x" + j + "z" + i;
                            textBox1.BackColor = Color.FromArgb(rgb[0], rgb[1], rgb[2]);

                            textBox1.Text = String.Format("{0:0.00}", (1-alpha)*100) +
" %";


                            Uavg[j] = 1 - alpha;
                            //Controls.Add(textBox1);
                            ttip.SetToolTip(textBox1, "Uavg = " +
String.Format("{0:0.00}", (Uavg[j]) * 100) + ", @t = " + String.Format("{0:0.00}", j *
intervalt)+" yr");

                        }

                        table1.Controls.Add(textBox1, j + 1, i + 1);

                    }
                }
            }

            table1.Visible = true;
            lblAvgDegree.Text = String.Format("{0:0.00}", (1 - getSum(a,(numberOfGridt-
1)) / getSum(a,0)) * 100);

        }
        public void assign(double[, ,] a, int k)
        {
            //reportViewer1.SetPageSettings
            //table1.Dispose();
            table1.Visible = false;

            table1.RowCount = a.GetLength(1)+1;
            table1.ColumnCount = a.GetLength(2)+1;
            table1.Controls.Clear();

            ToolTip ttip = new ToolTip();
            Label textBox1;

            int[] rgbMax = { 255, 50, 50 };
            int[] rgbMin = { 225, 255, 176 };
            int[] rgb = new int[3];

            table1.Padding = new Padding(0);

            for (int i = -1; i < a.GetLength(1); i++)
            {
                for (int j = -1; j < a.GetLength(2); j++)
                {

                    textBox1 = new Label();
```

```
                    //Color 1 rgb(255,50,50); for max u
                    //Color 2 rgb(225, 255, 176);  for min u


                    if (i < 0 | j < 0)
                    {
                        textBox1.BackColor = Color.AntiqueWhite;
                        textBox1.Width = 50;
                        textBox1.Margin = new Padding(0);
                        textBox1.TextAlign = ContentAlignment.MiddleCenter;
                        textBox1.BorderStyle = BorderStyle.FixedSingle;

                        if (j >= 0)
                        {
                            textBox1.Text = "X = "+String.Format("{0:0.0}",j *
intervalSpaceX);
                        }
                        else if (i>=0)
                        {
                            textBox1.Text = "Z = "+String.Format("{0:0.0}",i *
intervalSpaceZ);
                        }
                        else
                        {
                            textBox1.Text = "";
                        }


                    }
                    else
                    {

                        //set alpha from 0 - 1  depending on x, alpha = [(x - A)/(B - A)]
                        double alpha = (((a[k, i, j] - getSmallest(a)) / (getLargest(a) -
getSmallest(a))));

                        for (int f = 0; f < 3; f++) rgb[f] = (int)(rgbMin[f] + (rgbMax[f]
- rgbMin[f]) * alpha);

                        //textBox1.Name = "x" + j + "z" + i;
                        textBox1.BackColor = Color.FromArgb(rgb[0], rgb[1], rgb[2]);
                        textBox1.Width = 50;
                        textBox1.Text = String.Format("{0:0.000}", a[k, i, j]);

                        //textBox1.BackColor = Color.FromArgb(243, 243, 243);

                        textBox1.Margin = new Padding(0);
                        textBox1.TextAlign = ContentAlignment.MiddleRight;
                        textBox1.BorderStyle = BorderStyle.Fixed3D;
                        //textBox1.MouseHover += new
EventHandler(this.MyMouseHoverHandler);

                        //Controls.Add(textBox1);
                        ttip.SetToolTip(textBox1, String.Format("{0:0.00}", j *
intervalSpaceX) + ", Z = " + String.Format("{0:0.00}", i * intervalSpaceZ));

                    }
```

```
                table1.Controls.Add(textBox1, j+1, i+1);

            }
        }

        table1.Visible = true;
        u_avg[k] = (1 - sumPorePressure[k] / sumPorePressure[0]);
        lblAvgDegree.Text = String.Format("{0:0.00}", u_avg[k]*100);


    }
    private void textBoxZ1D_KeyPress(object sender, KeyPressEventArgs e)
    {
        const int BACKSPACE = 8;
        const int DECIMAL_POINT = 46;
        const int ZERO = 48;
        const int NINE = 57;
        const int NOT_FOUND = -1;

        int keyvalue = (int)e.KeyChar; // not really necessary to cast to int

        if ((keyvalue == BACKSPACE) ||
        ((keyvalue >= ZERO) && (keyvalue <= NINE))) return;
        // Allow the first (but only the first) decimal point
        if ((keyvalue == DECIMAL_POINT) &&
        (textBoxZ1D.Text.IndexOf(".") == NOT_FOUND)) return;
        // Allow nothing else
        e.Handled = true;
    }

    private void textBoxt1D_KeyPress(object sender, KeyPressEventArgs e)
    {
        const int BACKSPACE = 8;
        const int DECIMAL_POINT = 46;
        const int ZERO = 48;
        const int NINE = 57;
        const int NOT_FOUND = -1;

        int keyvalue = (int)e.KeyChar; // not really necessary to cast to int

        if ((keyvalue == BACKSPACE) ||
        ((keyvalue >= ZERO) && (keyvalue <= NINE))) return;
        // Allow the first (but only the first) decimal point
        if ((keyvalue == DECIMAL_POINT) &&
        (textBoxt1D.Text.IndexOf(".") == NOT_FOUND)) return;
        // Allow nothing else
        e.Handled = true;
    }

    private void txtThickness1D_KeyPress(object sender, KeyPressEventArgs e)
    {
        const int BACKSPACE = 8;
        const int DECIMAL_POINT = 46;
        const int ZERO = 48;
        const int NINE = 57;
        const int NOT_FOUND = -1;
```

```
            int keyvalue = (int)e.KeyChar; // not really necessary to cast to int

            if ((keyvalue == BACKSPACE) ||
            ((keyvalue >= ZERO) && (keyvalue <= NINE))) return;
            // Allow the first (but only the first) decimal point
            if ((keyvalue == DECIMAL_POINT) &&
            (txtThickness1D.Text.IndexOf(".") == NOT_FOUND)) return;
            // Allow nothing else
            e.Handled = true;
        }

        private void button12_Click(object sender, EventArgs e)
        {
            //Calculate U for 1D consolidation

            //INPUTS
            //zThickness = thickness of the consolidating layer
            //xWidth = width of the consolidating layer

            //horizontalDrainageFace = 0 if drainage face is on left, 1 for right, 2 if
two sidedrainage
            //verticalDrainageFace = 0 if drainage face is on top, 1 if bottom, 2 if two
sidedrainage


            bool err, err1=false, err2 = false;
            if (radioButtonAppr1D.Checked)
            {
                err1 = (!double.TryParse(textBoxZ1D.Text, out intervalSpaceZ) |
intervalSpaceZ == 0) |
                      (!double.TryParse(textBoxt1D.Text, out intervalt) | intervalt ==
0);
            }
            err2= (!double.TryParse(txtThickness1D.Text,out zThickness)|zThickness==0) |
                  (!double.TryParse(textBoxCv1D.Text, out cv) | cv == 0);

            err = err1 | err2;

                if (err)
                {
                    MessageBox.Show("Please inpute a non zero number for all inputs");
                }
            else
            {
                //intialize Grid
                if (radioButtonAppr1D.Checked)            //Input from user is
Approximate Interval
                {

                    numberOfGridZ = (int)(zThickness / intervalSpaceZ) + 1;
                    numberOfGridt = (int)(maxTime / intervalt) + 1; ;

                    //update intervals based number of ongrid

                    intervalSpaceZ = zThickness / (numberOfGridZ - 1);
                    intervalt = maxTime / (numberOfGridt - 1);
                }
```

```
                else                     //Use Automatic Griding
                {

                    intervalSpaceZ = 1;
                    intervalt = 0.1;

                    numberOfGridZ = (int)(zThickness / intervalSpaceZ) + 1;
                    numberOfGridt = (int)(maxTime / intervalt) + 1;

                    //update intervals based number of ongrid
                    intervalSpaceZ = zThickness / (numberOfGridZ - 1);

                }

                if (radioButton3.Checked) { verticalDrainageFace = 2; }
                else { verticalDrainageFace = comboBox1D.SelectedIndex; } //0 for Top; 1
for Bottom

                //inintialize U0 for 1D
                U0_1D = new double[numberOfGridZ];
                sumPorePressure = new double[maxIteration];
                sumPorePressure[0] = 0;
                for (int i = 0; i < numberOfGridZ; i++)
                {
                    U0_1D[i] = functionPW(i*
intervalSpaceZ,verticalDrainageFace,zThickness);
                    if (U0_1D[i] < 0) U0_1D[i] = 0;
                    sumPorePressure[0] += U0_1D[i];

                }
                //continue computing

                u_1D = solve_U1D(U0_1D, cv, intervalSpaceZ, intervalt,
verticalDrainageFace);
                Result frmResult = new Result(u_1D, numberOfGridt,
intervalt,intervalSpaceZ, u_avg_1D, maxTime);
                //assign1D(u_1D, u_avg_1D);

                frmResult.Show();
                //tabControl1.SelectedIndex = 2;
                textBoxPgNo.Text = string.Format("{0:0.0}", maxTime);
                //panel1.Visible = false;
                button2.Enabled = false;
                button3.Enabled = false;
                button4.Enabled = false;
                button7.Enabled = false;
            }


        }

        private void button5_Click(object sender, EventArgs e)
        {

        }

        private void radioButton4_CheckedChanged(object sender, EventArgs e)
```

```
        {
            //on sede vertical drainage selected
            comboBox1D.Enabled = true;
            verticalDrainageFace = comboBox1D.SelectedIndex;
            if (comboBox1D.SelectedIndex == 0)
            {
                lblBottom1D.BackColor = Color.Black;
                lblTop1D.BackColor = Color.LightGray;
            }
            else
            {
                lblTop1D.BackColor = Color.Black;
                lblBottom1D.BackColor = Color.LightGray;
            }

        }
    }
}
```

ANNEX III - CERTIFICATE FROM ETHIOPIAN INTELLECTUAL PROPERTY OFFICE

የኢትዮጵያ አእምሯዊ ንብረት ጽ/ቤት
Ethiopian Intellectual Property Office

የቅጅና ተዛማጅ መብቶች የሚያስገኙ ስራዎች ምዝገባ የምስክር ወረቀት
Copyright & Neighboring Rights Registration Certificate

የማመልከቻ ቁ.፡-2340/2012
Application No.

1. የመብት ባለቤት / Right owner's
   - 1.1. ስም / Name፡- Anteneh Masresha Zerihun
   - 1.2. ጾታ / Sex፡- Male ዜግነት/ Nationality፡- Ethiopian
   - 1.3. አድራሻ /Address
     - ሐገር / Country፡- Ethiopia ክልል/Regional State ፡ Addis Ababa
     - ክ/ከተማ/Sub-city/_ዞን/Zone/: - Kolfe Keranio
     - ወረዳ /Woreda/፡- 09 ቀበሌ/Kebele፡ ቤ ቁ./ House No:- 678
     - ስልክ ቁጥር /Tel No 09 20 72 43 61 ፖ.ሳ.ቁ/ P.O.Box፡-
2. የፈጠራ ሥራው / Creative Work.
   - 2.1. ርዕስ/ Title; Consolidation2D
   - 2.2. ምድብ/ Type :- Computer Program ንዑስ ምድብ /Sub-type;-
   - 2.3. የምዝገባ ቀን / Registration Date፡- 23/07/2020 GC
   - 2.4. የምዝገባ ቁጥር / Registration No፡- 5/0444
   - 2.5. የምስክር ወረቀቱ ህጋዊነት የሚያበቃበት ቀን/ Certificate's Date of Expiry:- 22/07/2025 GC
3. የመብት ወሰን / Scope of right

በቅጂና ተዛማጅ መብት ጥበቃ ለማድረግ የወጣ አዋጅ ቁጥር 410/1996 የተዘረዘሩት የአመንጪ መብት በሙሉ፡፡

ምዝገባውን ያፀደቀው ኃላፊ /Signed by

Nassir Nuru Reshid
Copyright & Community Knowledge Protection and Development Directoret Director

ቀን/ Date፡-23/07/2020 GC

Federal Democratic Republic of Ethiopia Intellectual Property Office

ማሳሰቢያ፡- ይህ የቅጅና ተዛማጅ መብት የሚያስገኝ ስራ ምዝገባ የምስክር ወረቀት ቀዳሚ የባለቤትነት ማስረጃ በመሆን ያገለግላል፡፡*

** ይህ የምስክር ወረቀት በየአምስት ዓመቱ መታደስ አለበት፡፡*

*Note:- * This copyright & neighboring right registration certificate serves as prima facie document of right ownership.*

** This certificate has to be renewed every five years.*