ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

DEVELOPMENT OF DESIGN AID FOR TUNNEL LINING DESIGN USING ANALYTICAL METHOD(CONTINUUM)

PROJECT SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF ENGINEERING IN GEOTECHNICAL ENGINEERING

BY:TESFAMICHAEL GEBREWOLD

ADVISOR : **Dr.Ing:-HENOK FIKRE**

NOVEMBER, 2015,
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

DEVELOPMENT OF DESIGN AID FOR TUNNEL LINING DESIGN USING ANALYTICAL METHOD(CONTINUUM)

PROJECT SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF ENGINEERING IN GEOTECHNICAL ENGINEERING

BY:TESFAMICHAEL GEBREWOLD

NOVEMBER, 2015,

Approved by board examiners:

1. Dr.Ing:-Henok Fikre ________________ ______________

   Advisor Signature Date

2. Ato. Ameyu Temesgen (M.Sc.) ________________ ______________

   Examiner Signature Date

3. ____________________ ________________ ______________

   Chairman Signature Date

DECLARATION

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name: Tesfamichael G/wold

Signature: ______________________

ABSTRACT

Tunnel support design is an itterative process including assumptions on support type installed and evaluating the support pressure it provides. The design of tunnels differs significantly from the design of plant and other structures due to the difficulty of ascertaining accurate geological properties and the potential variability of these properties along the tunnel. Thus, the design is based on less reliable material property assumptions than most other designs.

There are Empirical, Analytical and numerical methods of design. In this project, an application software for continuum analytical tunnel design was developed.

The developed software analyses secondary stress around circular opening and a tunnel design by using continuum method is based on main assumptions and requirements established for this method of design.

The results of the software can be used for preliminary design and can acts as a tool for quickly verifying the numerical calculations and assessing the system behaviour during all stages of the design and construction process.

KEY WORDS: tunnel ; support design; continuum method; developed software

ACKNOWLEDGEMENTS

I would like to express my indebted appreciation to my advisor Dr. - Ing. Henok Fikre for his very kind supervision, advice and insightful suggestions throughout the preparation of this project paper.

I would like to express my special thanks to my family for their support and continuous assistance.

Table of contents

Appendix

List of Tables

List of Figures

List of Acronyms

A.....Average cross-sectional area of support per unit meter
C.....Relative stiffness of ground-support system
c.....Soil cohesion
D.....Tunnel diameter
E_R .....Elastic modulus of support
E_k .....Elasticity modulus for the three-dimensional continuum
Es.....Ground deformation modulus
h.....Tunnel crown depth
I.....Moment of inertia of lining
M.....Bending moment
N.....Normal force
P_v, P_h .....Vertical and horizontal ground pressure
P_{ro}; P_{r2} .....Radial earth pressure
P_{t2} , P_t .....Tangential earth pressure
$P^S{}_r$, $P^S{}_t$ ..... Radial and tangential load part on soil
$P^R{}_r$, $P^R{}_t$ ..... Radial and tangential load part on lining
Q.....Shear force
r.....Tunnel radius
V.....Tangential deformation
W.....Radial deformation
W_o , W_i .....Support section external and internal deformation
γ.....Unit weight
ϕ.....Soil internal friction angle
μ.....Soil poisson's ratio
θ.....Radial angle
σ_t , σ_c.....Tensile and compressive stresses
σ_{rr} , $\sigma_{\theta\theta}$ .....Radial and Tangential stresses
u_r ,u_θ .....Radial and Tangential displacements

CHAPTER ONE

INTRODUCTION

1.1. Background

In recent years, as a resulting the increases in computing power and development of many software's in the field of engineering makes a lot of tiresome, laborious and time consuming tasks can be accomplished with in a seconds.

In engineering field of study, the use of commercially available packages and software's become common for several purposes. It's because that most of engineering subjects are consist of huge equations and calculation processes, thus it will be very difficult to perform many task with in short period being out of any fault.

And the development of numerical methods like finite element, finite difference and discrete element methods make thing to be easy if it's utilized properly.

This project paper is aimed on the development of mini software application for continuum method of tunnel lining design which is one of the commonly known analytical design methods using Visual Basic .Net 2010.

There are different analytical methods to estimate the internal forces in a tunnel lining and give an indication on the type of support needed.

In this paper the focus is on tunnels which have a large overburden ($h \geq D$). This allows the ground to be treated as a continuum. It can be assumed that the area above the tunnel is not softened and can carry some load. The analysis follows basic assumptions and requirements i.e.

the tunnel to be straight tunnel, homogenous, isotropic and ideal elastic material behavior of ground and tunnel lining. [2]

The calculation of the internal forces of the analytical model is carried out using the displacement method. Equations of first order theory, for the case of a rigid bond between the tunnel support and the ground assuming infinite axial stiffness. If using segmental lining, the pre-deformations of the segments as a result of the installation can be considered using second order theory. [2]

1.2. Objective of the project

The general objective of this paper is to develop software application that analyzes continuum analytical tunnel lining design and secondary stress around circular opening.

The specific objectives are the following:

- To develop simple, fast and accurate way of analytical tunnel lining design process.
- To develop easy way to find stress around circular opening of excavation.

1.3. Problem Statement

Analytical method of tunnel lining uses long calculation process and repeated iterations, thus hand calculation would be laborious and time taking activity. The development of this mini application would enhance the calculation and might serve for rapid preliminary practical design.

1.4. Organization of the paper

This thesis has six chapters. The first chapter contains a general introduction about the study, the objective and the problem statement of the study. The second chapter is about the previous works done regarding tunnel lining design methods. In the third chapter, the methodology followed for study is provided. The fourth chapter is about the procedures to use the software with its general graphical features are presented. Parametric studies on the effects of some properties on force parameters using the software results is included in sixth chapter. The final chapter, the conclusions made from the study and the recommendations for further studies are stated.

1.5. Significance and Limitations of the paper

The software can be used for preliminary design and can acts as a tool for quickly verifying the numerical calculations and assessing the system behaviour during all stages of the design and construction process. In addition it can serve as an educational aid tool in the institutions.
The project paper is limited on the development of one of analytical tunnel deign method. Continuum method assumes the nature of the ground to be ideally elastic, homogenous and isotropic. Hence it can only be applicable for the tunnel positioned at a depth three times greater than its diameter.
There are three additional analytical methods of design which were not included in the projected called bedded beam spring, semi-continuum and convergent confinement. Since this methods uses many independent design charts and table, so it requires a long period to integrate with the software.

CHAPTER TWO

TUNNEL LINING DESIGN

2.1 Design of Tunnels

Underground structures can be designed by using many different calculation methods. While the preliminary design is dominated by analytical and empirical methods, fully modelling the entire construction process in the course of numerical calculations represents the standard procedure for detail design today. [6]

Furthermore analytical methods can act as a tool for quickly verifying the numerical calculations and assessing the system behaviour during all stages of the design and construction process. [6]

2.1.1 Empirical Design Methods

The structural elements and the excavation procedure, especially for the preliminary support of the tunnel, may be selected mainly based on experience and empirical considerations that rely more on direct observations than on numerical calculations. [8]

This procedure may be especially reasonable if experiences from a successful tunneling project can be applied to a similar, new one yet to be designed

One disadvantage of prolonged application of the empirical approach is that, lacking an incentive to apply a more appropriate tunneling design via a consistent safety assessment, the structure may be designed over conservatively, resulting in higher construction costs[8]

Some soft ground empirical rules have also been established but these tend to be based on local experience rather than being universally applicable. **[15**]

The most frequently used empirical design methods are the RMR – Rock Mass Rating – (Bieniawski, 1994) and the Q systems**.** [15]

2.1.2 Analytical Design Methods

It is impossible to take all the influences, parameters and boundary conditions that are dependent on the geology and construction phases into account in a calculation. Therefore, analytical models have been developed which simplify reality to such an extent that the remaining parameters can be dealt with in a calculation and at the same time lead to sensible results. [2]

They are unable to model the full complexity of a tunnel during construction. Specifically, they are generally two-dimensional idealizations that assume the ground is a homogeneous continuum and the tunnel is circular. [15]

There are three commonly known analytical methods are, i.e. the bedded-beam spring method, the continuum method and the tunnel support resistance method. The assessment of which method to use depend on the tunnel depth and ground condition.

two conditions can be defined as:[2]

- Shallow, h< 2D, i.e. where the ground above the tunnel crown in assumed to have no bearing capacity excavation process creates a softening zone in the crown area.
- Deep, h>3D, i.e. where the ground above the tunnel crown is acting as a support;

Where h is the tunnel crown depth and D is the tunnel diameter.

2.1.3 Bedded-beam spring method

The tunnel support is idealized as an elastically supported circular ring. The elastic bedding is achieved through radial and potentially tangentially arranged springs. The spring stiffness simulates the support behaviour of the ground. [8]

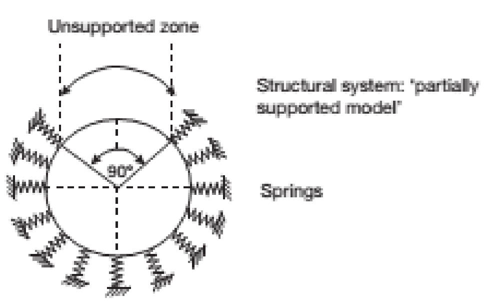


Fig 1Spring model [2]

The calculation is carried out elastically. As the ground is only represented by springs, the analysis cannot provide any information with regard to the settlement at the ground surface and to the possible stress and deformation behavior of the ground (secondary stress situation). [2]

For deep tunnels the bedded-beam spring method is generally not used because even with a supporting crown area, the supporting nature of the ground is not sufficiently taken into account

The bedded-beam spring method is the fastest and simplest calculation method and is often used to determine thickness and required reinforcement of the supporting circular ring following the results of a more sophisticated calculation method.

2.1.4 Convergence–Confinement Method (CCM)

For the support resistance method, it is assumed that the tunnel support con -strains the deformations of the ground, i.e. it provides an internal pressure (resistance) against the ground.[2]

The design criterion for this method is the limitation of the deformation of the ground. The connection between the rock deformation and the tunnel support resistance can be shown pictorially using the Fenner-Pacher curve,

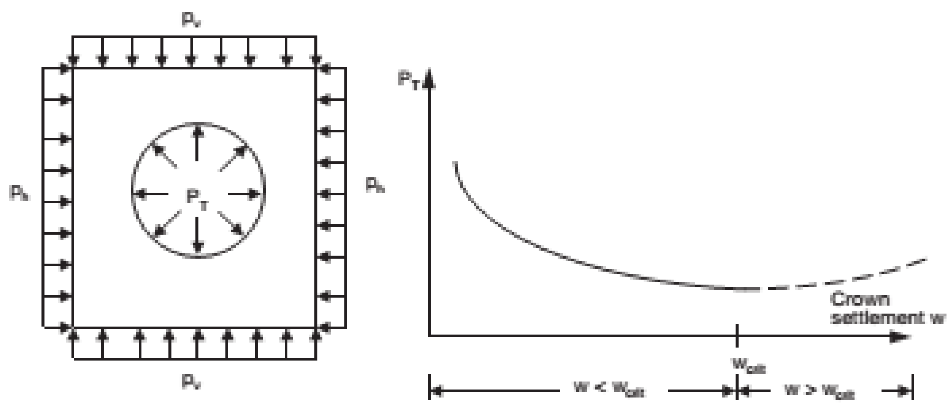


Fig 2Tunnel support resistance method [6]

The more the ground deforms (distresses) before the tunnel support is placed, the lower the load that has to be carried by the tunnel lining and the higher the self-supporting element of the ground. The required tunnel support resistance reduces with increasing deformation. [2], [6]

2.1.5 Limit-equilibrium method

Rock support systems can be designed using limit-equilibrium methods of analysis. The support requirements for individual wedges can be calculated by hand or using programs such as UNWEDGE to provide a graphical presentation of the wedge geometry and assess the distribution of support

2.1.6 Continuum method

The tunnel in deep depth of ground, in which the tunnel is constructed, is idealized as a continuum, i.e. there are no discontinuities in the material. The method assumes that the ground is an infinitely large thin section with a hole at the center.**[2]**

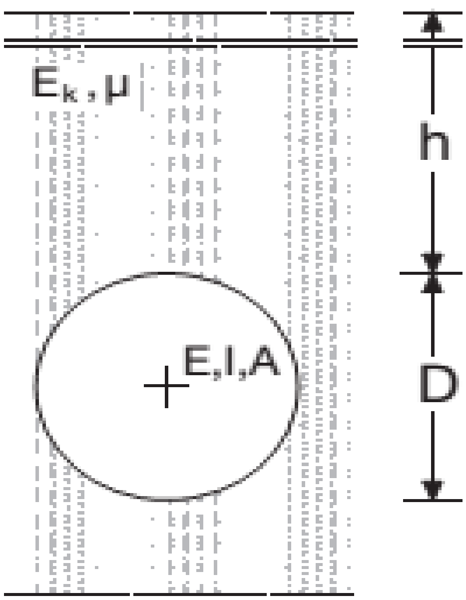


Fig 3Continuum model [2]

This calculation method allows the interpretation of the deformation and strains in the ground.

It is used in shallow tunnels in weak rock as a partial continuum method and for deep tunnels in weak rock as a continuum method with a bedded crown area.

It allows option to apply a reduction factor to the full applied ground stress; any stress relief depends on the ground conditions and the method of construction. This reduced stress can be assumed at 50–70% if the depth to tunnel axis is greater than three diameters (Duddeck and Erdman, 1985). [15]

Various closed form solution such as Ahrens, Lindner, Lux (1982), Einstein (1979), Bobet (2001), Muir Wood(1975) and Bouma (Bakker 2003) have been for estimation of the various forces carried by the tunnel lining. The closed form solutions are generally assuming a continuum of homogenous soil. [11]

2.1.7 Numerical Design Methods

As a result in the increases in computing power and the fact that there are many commercial packages available, the use of numerical models has increased significantly. [2]

It offer the ability to model explicitly complex structures, including adjacent structures, different geological strata, complex constitutive behaviour, transient and construction sequences.

The benefits of numerical methods over analytical or closed form solutions as being able to((Potts and **Zdavkovic (1999): [15]**

- simulate the construction sequence;
- deal with complex ground conditions;
- model realistic soil behaviour;
- handle complex hydraulic conditions;
- deal with ground treatment, for example compensation grouting;
- account for adjacent services and structures;
- simulate intermediate and long-term conditions;
- deal with multiple tunnels. And dynamic loading,

Numerical methods can be divided into different types depending on the computation methods adopted in the software package. For modelling continua, such as soils, the most common numerical methods are the finite element method or finite difference methods. For modelling discontinua, such as rocks, discrete element and the boundary element method are used. [7]

CHAPTER THREE

METHODOLOGY

3.1 General

As stated in chapter one, the objective of this paper is to develop software for analytical continuum tunnel design. The software is developed using Visual Basic .Net 2010 programing language. The programing language allows the coder to provide standardized, user friendly and customized software features.

As a procedure a flow chart and sequence of analysis was prepared before any actions on the designing of user interface. Then creation of interface forms have been done following code writing and debugging.

The flow chart describes general flow and steps of work. Which includes how it accepts input parameter, checking of the correctness and appropriateness of inputs according to the followed assumptions and equations used. Flow chart is attached on the annex part of the paper.

The software has simple multi- form user interface. These forms works and communicate to each other through globally declared variables. Programing codes of analysis was written by following standard declaration and inserting appropriate equations of the analysis. During the process of coding repeated debugging or checking of the accuracy of codes was performed.

The software executes stress analysis and analytical tunnel design, basic assumption and equations considered are given on the next sections below.

3.2 Continuum Method of tunnel Design

For Deep, h>3D, i.e. where the ground above the tunnel crown is acting as a support, is idealized as a continuum, i.e. there are no discontinuities in the material.

3.2.1 Input

The software requires the following parameters, namely unite weight, deformation modulus, poisson's ratio, static earth pressure coefficient, overburden depth, tunnel radius and radial angle as ground property and area of section of lining, moment of inertia, modulus of elasticity, internal, external deformations, allowable compressive and tensile strength as support material parameters

3.2.2 Conditions and process

The calculation of the internal forces of the analytical model is carried out using the displacement method. This requires that the primary stress situation or the stresses determined from the earth pressure approach is used as a load displacement condition.

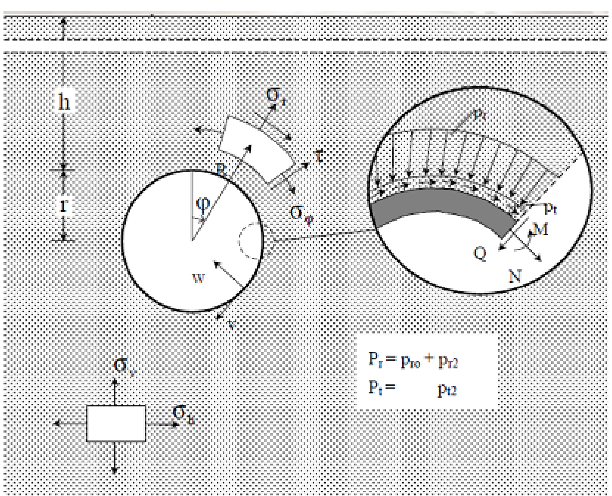


Fig 4Definition of parameters. [2]

The following are the equations, using first order theory, for the case of a rigid bond between the tunnel support and the ground assuming infinite axial stiffness. If using segmental lining, the pre-deformations of the segments as a result of the installation can be considered using second order theory. [2]

Earth pressure:

$$P_v = \gamma * h$$

$$P_h = \gamma * (h + r) * K_o$$

Transformation into polar coordinates:

$$P_r = P_{ro} + P_{r2} * \cos 2\theta$$

$$P_t = P_{t2} + \sin 2\theta$$

With

$$P_{ro} = 0.5 * \gamma * [h + (h + r) * K]$$

$$P_{r2} = P_{t2} = 0.5 * \gamma * [h - (h + r) * K]$$

Where:

γ: unit weight

r: tunnel radius

P_v : vertical ground pressure

P_h : horizontal ground pressure

P_{ro}; P_{r2} : radial earth pressure

P_{t2} ,P_t: tangential earth pressure

The load and deformation variables of the ground are denoted with a superscript ‘S’, while the load and deformation variables associated with the circular ring frame are denoted with a superscript ‘R’[2]

Elasticity modulus for the three-dimensional continuum can be determined with a Poisson’s ratio and ground stiffness parameter [2]:

$$E_k = \frac{(1 + \mu)(1 - 2\mu)}{(1 - \mu)} * E_s$$

Deformations:

$W(\theta) = \overline{w_0} + \overline{w_2}\cos 2\theta$

$V(\theta) = \overline{v_2}\sin 2\theta$

With: $EA \to \infty : Wo = 0$

$$\overline{w_2} = \frac{P_{r2} + 0.5 * P_{t2}}{\frac{1}{(3-\mu-4\mu^2)} * (2.25 - 1.5\mu) * \frac{E_k}{r} + \frac{9EI}{r^4}}$$

$\overline{v_2} = \overline{0.5 * w_2}$

Where :

μ: soil poisson's ratio

W(θ): radial deformation

V(θ): tangential deformation

θ: radial angle

The proportion of the earth pressure acting on the circular frame and the continuum is equivalent to their relative stiffness's. The circular frame load is:[2]

$$\overline{P_{ro}{}^{R}} = P_{ro} - P^{S}{}_{ro}$$

For $EA \to \infty : P^{S}{}_{ro} = 0$ as a result of the infinite axial stiffness, the complete constant load is supported by the circular ring frame

$$\overline{P_{ro}{}^{R}} = P_{ro}$$

$$\overline{P_{r2}{}^{R}} = P_{r2} - P^{S}{}_{r2}$$

$$P_{r2}{}^{S} = \frac{E_k}{r} * \frac{1}{(3-\mu-4\mu^2)} * [(5-6\mu) * w_2{}^{S} + (-4+6\mu)v_2{}^{S}]$$

$$\overline{P_{t2}{}^{R}} = P_{t2} - P^{S}{}_{t2}$$

$$P_{t2}{}^{S} = \frac{E_k}{r} * \frac{1}{(3-\mu-4\mu^2)} * [(-4+6\mu) * w_2{}^{S} + (5-6\mu)v_2{}^{S}]$$

Proportional internal force parameter:

$$N_0 = -r * P^R{}_{ro}$$

$$Q_0 = 0$$

$$M_0 = 0$$

Load Part $P^R{}_{r2}, P^R{}_{t2}$

$$N_2 = \frac{r}{3} * (2 * P^R{}_{t2} + P^R{}_{r2},) \cos 2\theta$$

$$Q_2 = -\frac{r}{3} * (P^R{}_{t2} + 2 * P^R{}_{r2},) \sin 2\theta$$

$$M_2 = \frac{r^2}{6} * (P^R{}_{t2} + 2 * P^R{}_{r2},) \cos 2\theta$$

Or Simpler from deformation:

$$Q_2 = -\frac{6EI}{r^3} * w_2{}^R * \sin 2\theta$$

$$M_2 = -\frac{3EI}{r^2} * w_2{}^R * \cos 2\theta$$

Final internal parameters and deformation:

$$N = N_0 + N_2$$

$$Q = Q_2$$

$$M = M_2$$

$$W = w_2$$

Where:

$P^S{}_r$: radial load part on soil

$P^R{}_r$ radial load part on lining

$P^S{}_t$ tangential load part on soil

$P^R{}_t$ tangential load part on lining

N: normal force

Q: shear force

M: bending moment

To consider influence of second order theory the normal force over the tunnel circumference is assumed to be constant. Formwork dependent deformation W_v, which can lead to change shape of the circular tunnel shape considered, thus it is handled with the modified stiffness of the tunnel lining [5]

$$W_v = W * \omega$$

$$\overline{EI} = EI + N(1+\omega) - \frac{r^2}{3}$$

New radial deformation is determined form modified stiffness:

$$w^{II}{}_2 = \frac{P_{r2} + 0.5 * P_{t2}}{\frac{1}{(3-\mu-4\mu^2)} * (2.25 - 1.5\mu) * \frac{E_k}{r} + \frac{9\overline{EI}}{r^4}}$$

Check the value ofω: $\omega = \frac{W_{v,F}}{W_2{}^{II}}$, here repeated iteration is required until previous and new coefficients become equal in value.[5]

Then second order moment is given as:

$$M_2^{II} = \frac{3 * EI}{r^2} * W_2{}^{II} * \cos 2\theta$$

Once the value of internal forces determined, it is possible to perform stress analysis for the selection of appropriate section of lining material.

$\sigma = \frac{N}{A} + \frac{M}{w}$ where

σ = stress

N= normal force

M= moment

A= section area

W= section deformation

3.2.3 Output

The outputs of continuum design are present numerically incorporation with pictorial representation. The results include an internal forces, displacement (deformation) and stresses of first order and second order theory calculations.

CHAPTER FOUR

APPLICATION OF THE SOFTWARE

In this portion of the paper, the graphical features, contents and procedures to use the developed software are presented.

3.1. General

As the software performs only two analyses, it may not include complex features to use. When the application runs the window shown below will appear and it is labeled as Tunnel Continuum Analysis and Design(**TCAD**)

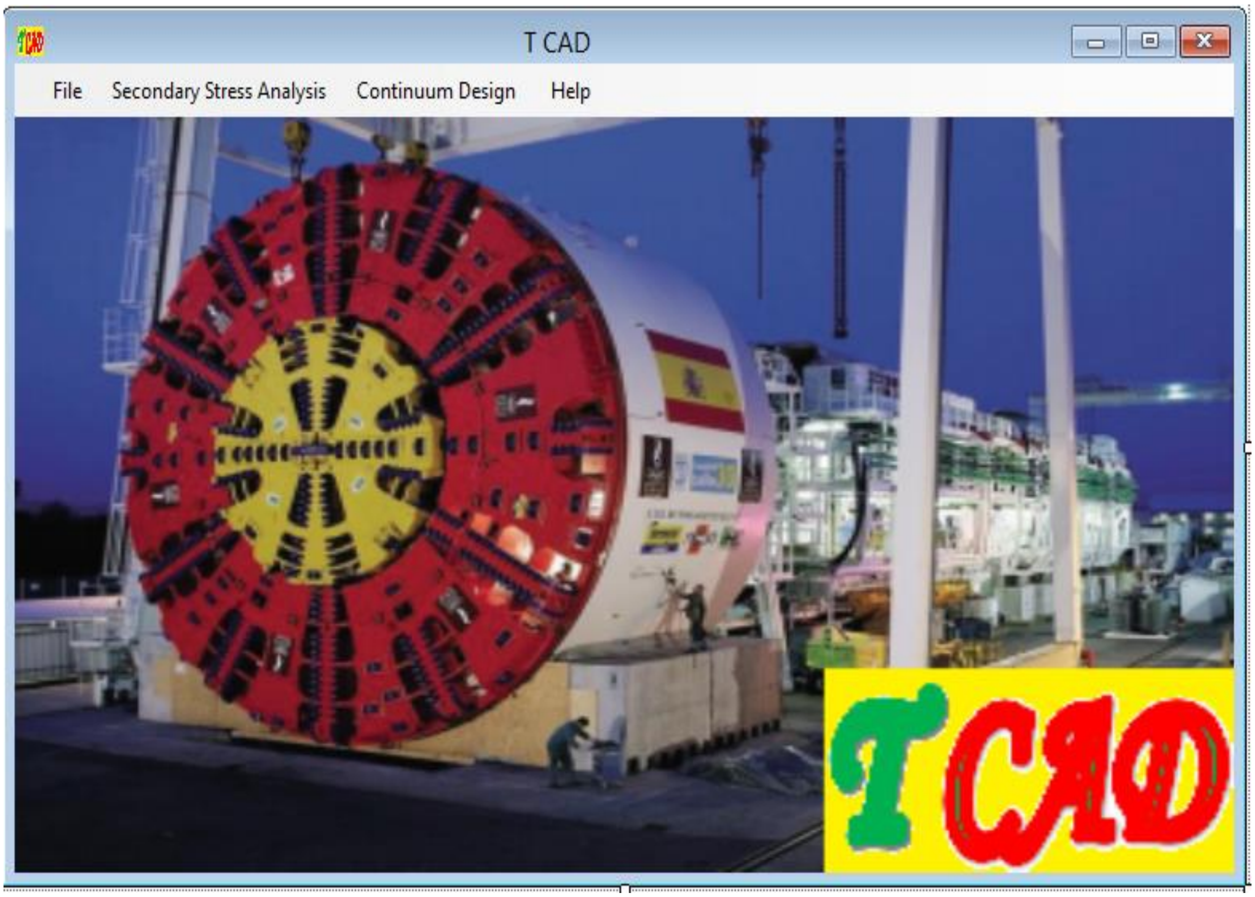


Fig 5Main starting window

The Above window contains menu bar with different annotations.

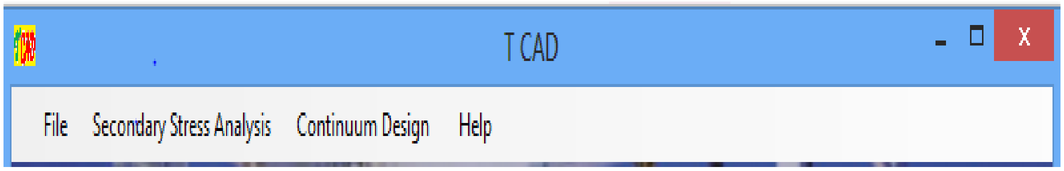


Fig 6Menu bars

3.2. Menu bar

The file menu at the left corner contains the command of exit button which lets the program to cancel its operation and closes down.

The other menus are prepared simultaneously allow the user for the selection of analysis type.

3.2.1. Stress Analysis menu

This menu contains the commands used to insert input parameters for secondary stress analysis and an output command to display the results.

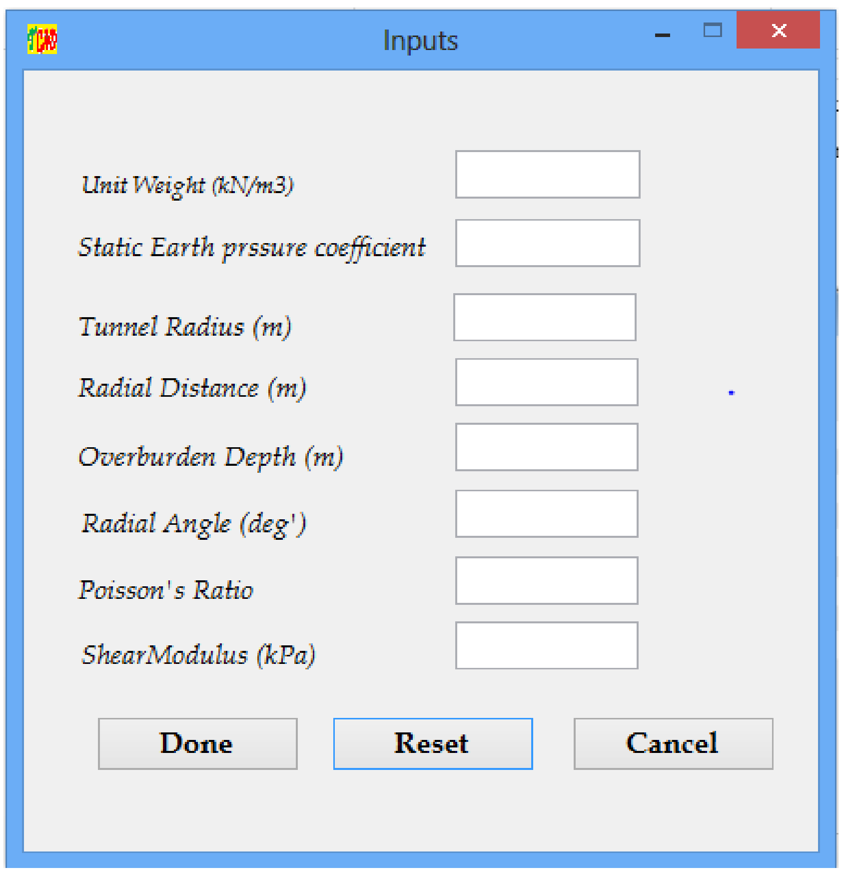


Fig 7Input parameter dialog box

Input window contains text boxes to accept inputs and buttons to perform the required action.

The text boxes are programed to accept only numerical values. It also coded not to leave blank or null, it automatically asks to insert numerical values and fill null spaces by showing small indicative warning box.

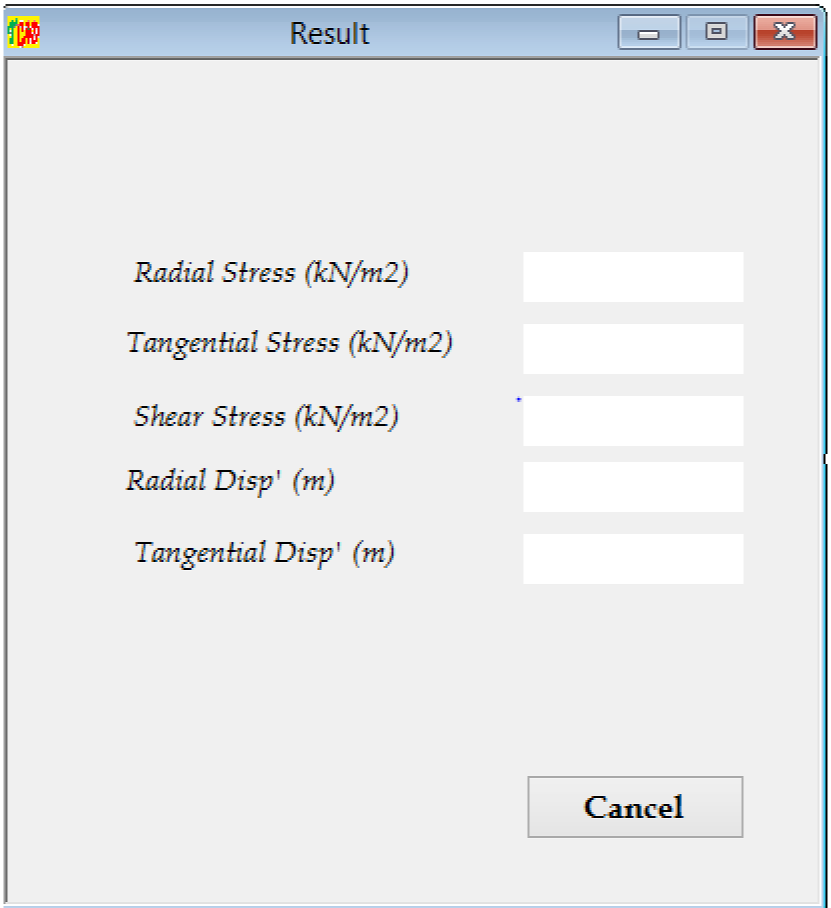


Fig 8Output window

3.2.2. Continuum Design menu

The continuum design menu lets the user to perform design. Which contains three commands of to supplement ground parameters, support parameters and an option to display the results successively.

The menu is programed to guide or lead the users to follow stepwise activities. In such way that once the continuum button is clicked, it can only activate the ground property menu. Similar to above input window the user can continue operation by feeding only numerical values. The provided buttons are used to perform actions as clearly labeled on them.

Ground parameter and lining property dialog boxes lets the user to input necessary parameters for the design process.

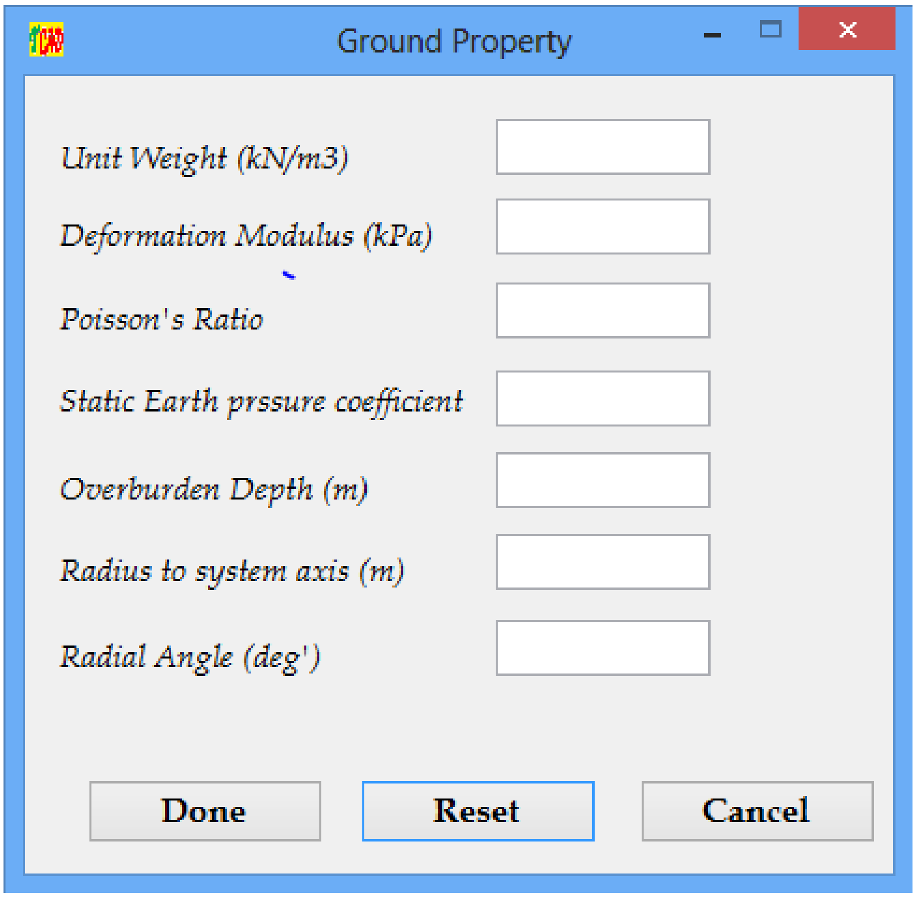


Fig 9Ground Property input dialog box

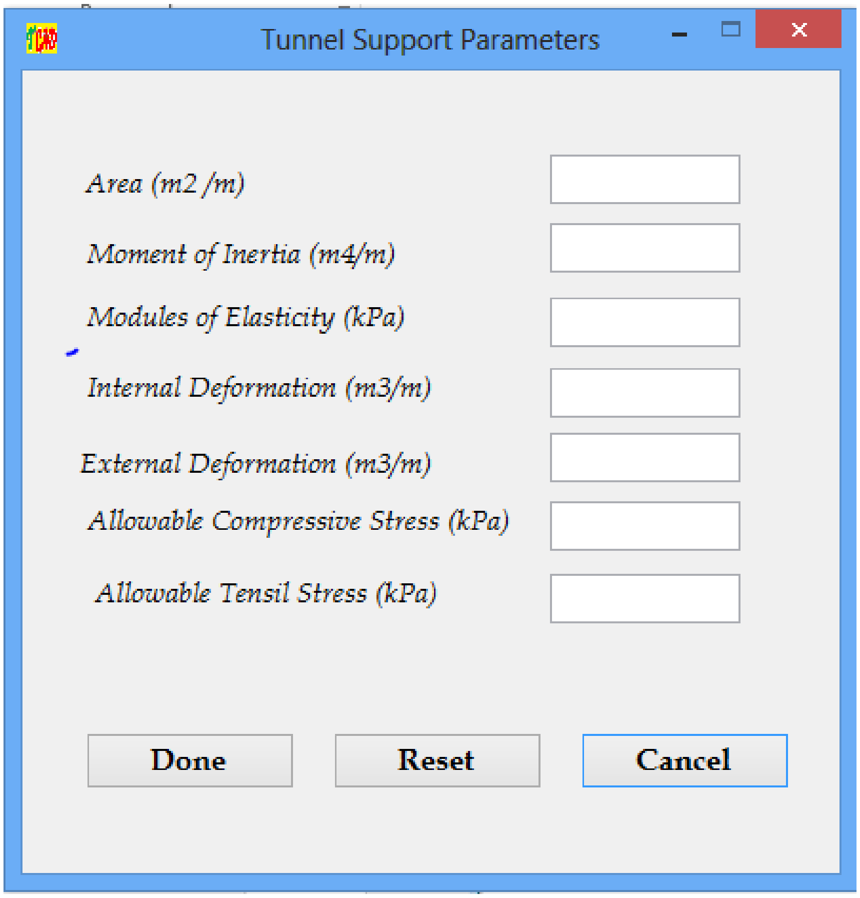


Fig 10Lining Properties input dialog box

As the figure shows below, results are displayed as numerical values and pictorial representation.

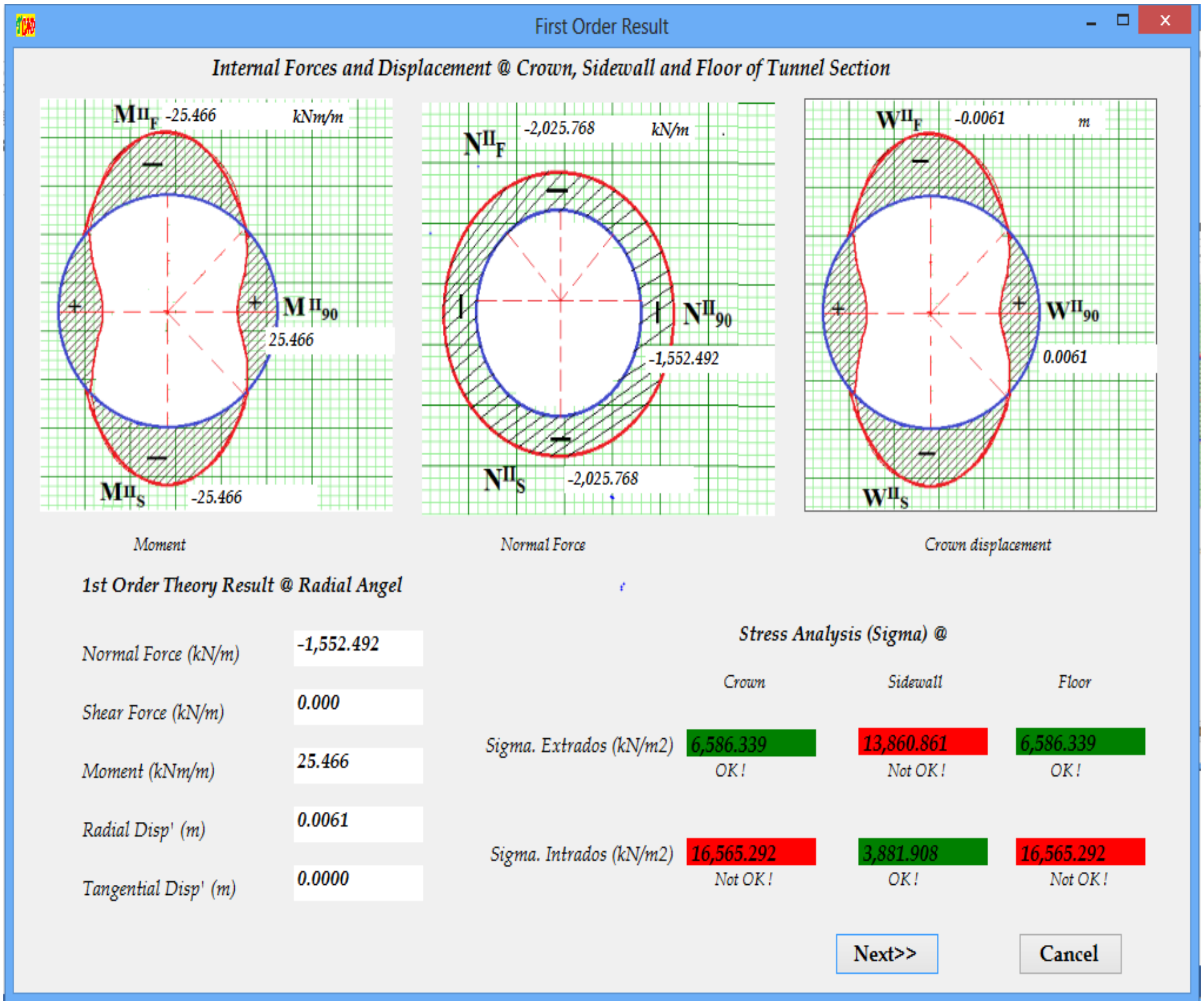


Fig 11 First order result output window

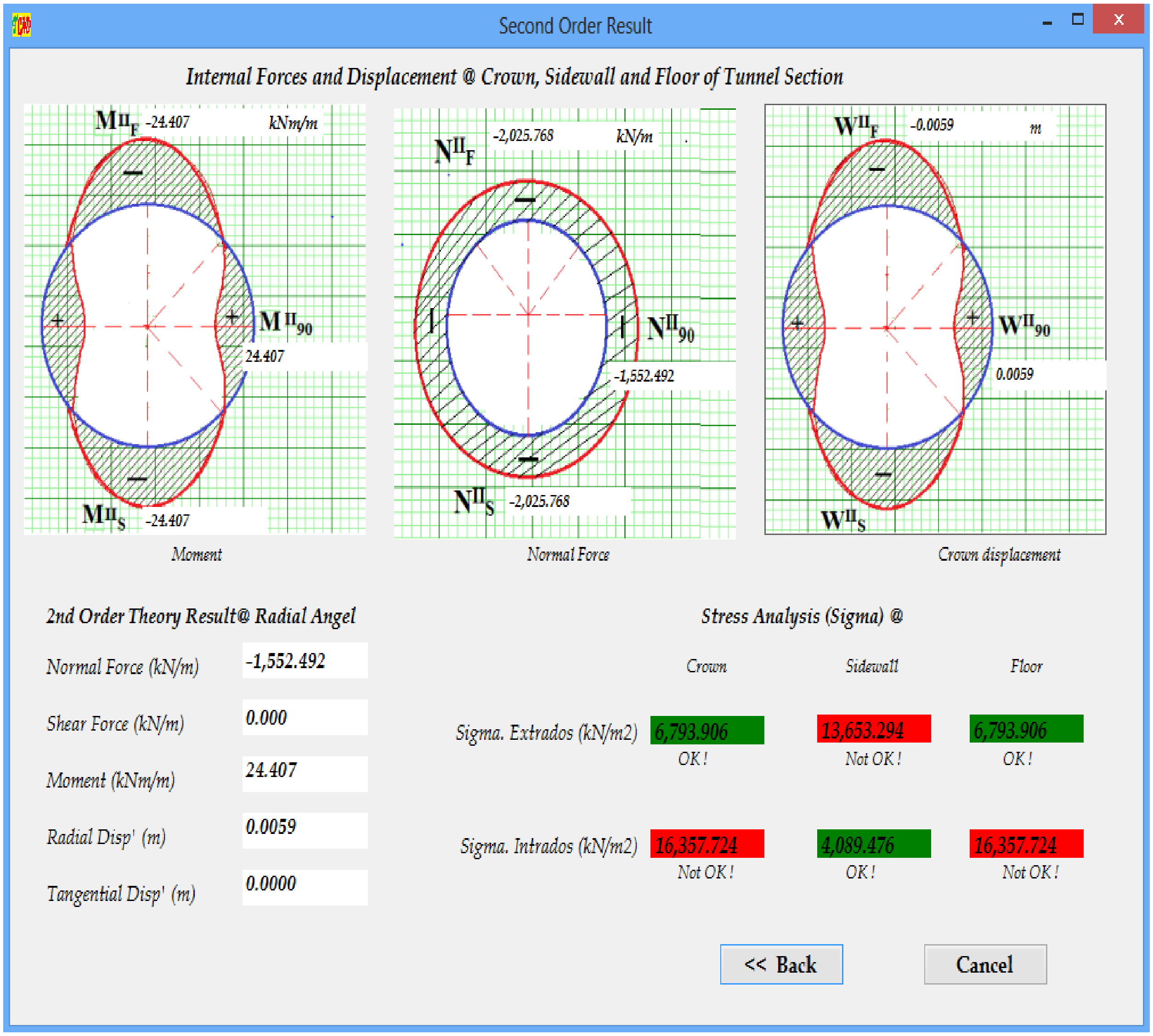


Fig 12 Second order result output window

3.2.3. Help menu

The Help menu contains the little guide about the commands and description on the software TCAD

3.3. Acceptance of the Software

To create an acceptance on the developed software, examples are provided by hand calculation and compared with software results on the appendix. The subsequent observation have been made on the results:

- Results of hand calculations and software outputs are relatively similar
- It is very simple and faster application to perform those long iteration and equations.

CHAPTER FIVE

PARAMETRIC STUDY

The lining is designed to support the weight of the overburden and a horizontal pressure equal to some fraction of it. Tunnel linings seldom carry the all total load of the overburden. The soil above the tunnel is only partly supported by the liner. What occurs in al1 tunnels is that in-situ stresses are redistributed around the opening due to the inherent shear strength and continuity of the ground. This transfer of pressure from a yielding mass of soil on to adjoining stationary parts is called the arching effect. The lining theoretically has to support only those stresses not arched to the adjacent ground. [7]

The results of parametric study are determined using the developed continuum analysis software application and all necessary assumptions for this analysis type are included for the calculation

5.1 Factors influencing the load parameters on the Liner

The prediction of lining loads and for the lining design has always been the goal of tunnel engineers.

Many factors contribute to the difference of loads on the linings of many different tunnels. It is very important to understand the influence of these factors on the tume1 lining to predict the loads on the liner more reliably. [7]

Some of factors influence force parameters on tunnel lining are introduced here. The table below shows values of soil and lining material. [12]

Table 1Factors and their values of soil properties used in parametric studies [12]

Ground parameters	Unit	Values
Unit weight (γ)	kN/m^3	18
Stiffness/Modulus of compressibility, Es	MPa	100
Shear modulus (G)	GPa	25
Poisson's ratio (μ_s)		0.3
Coefficient of lateral earth pressure (Ko)		0.5
The overburden (h)	m	30
Tunnel radius (r)	m	3
Soil cohesion (c)	kPa	30
Soil internal friction angle (ϕ)	deg	30

Table 2Factors and their values of liner material properties used in parametric studies [12]

Structural system parameters	Unit	Values
concrete (C25)		
Unit weight (γ)	kN/m^3	24
Poisson's ratio (μ_R)		0.15
young's modulus ($E_{R)}$	MPa	30000
Area (A)	m^2/m	0.25
Moment of inertia (I)	m^4/m	0.001302
External deformation ($W_{o)}$	m^3/m	5.104x10^-3
Internal deformation (W_i	m^3/m	5.104x10^-3
Allowable compressive strength (σc)	MPa	11.3

5.1.1 Relative Rigidity of the Lining with Respect to the Surrounding

The load on the lining is dependent on the stiffness of the lining relative to that of the ground. The stiffer the support is relative to the ground, the greater the support load is. The compressibility ratio, **C**, is a measure of the relative stiffness of the ground-support system under a uniform or symmetric loading condition and is defined by: [14]

$$C = \frac{E_s r(1-\mu_R{}^2)}{E_R A(1-\mu_S{}^2)}$$

where E_S, μ_S and E_R, μ_R = the elastic constants for the ground and support

A, = the average cross-sectional area of the support per unit length of tunnel

r = the tunnel radius.

There is a need to account for both the stress-strain properties of the ground and liner because a lining that may be stiff with respect to a soft clay may behave as a flexible lining in a very stiff clay or other soil type. [7]

As the stiffness of lining material is changed relative stiffness varies and change on the force parameter is calculated using the developed software by continuum method. it assumed that other parameters are kept constant.

Table 3Variation of relative stiffness Vs Maximum bending moment and Maximum normal force

Lining young's modulus (E_R) (kPa)	Modulus of compressibility (E_S) (kPa)	Relative Stiffness (C)	Bending moment (KNm/m)	Normal force (KN/M)
30000000	100000	0.053708791	106.866	965.215
28000000	100000	0.057545133	101.014	966.145
26000000	100000	0.061971682	95.011	967.098
24000000	100000	0.067135989	88.851	968.076
22000000	100000	0.073239261	82.527	969.079
20000000	100000	0.080563187	76.033	970.108

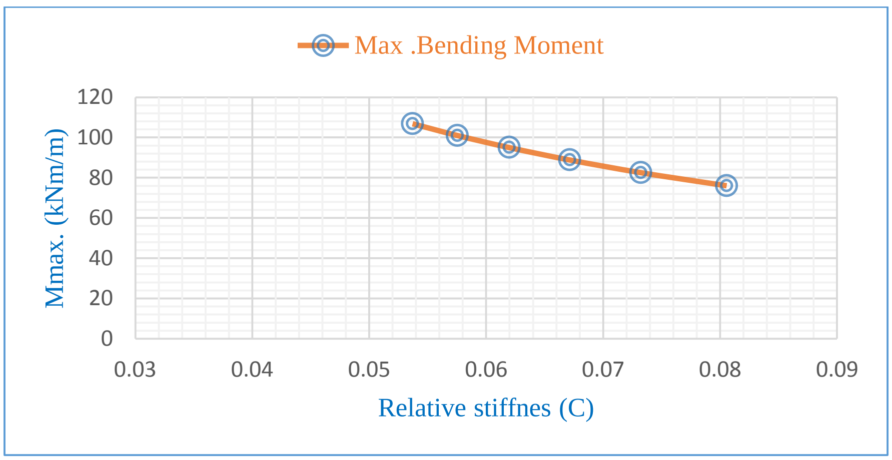


Fig 13Relative stiffness Vs Maximum Bending moment

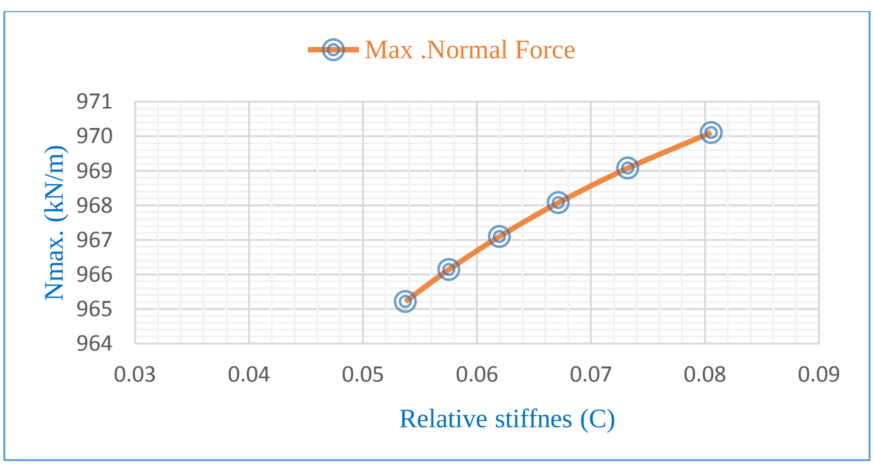


Fig 14 Relative stiffness Vs Maximum Normal force

Conclusion on result

It is observed that for a larger value of flexibility ratio (large soil modulus), the moment becomes very small. Conversely, for a small value of flexibility ratio (relatively rigid lining), the moment is large.

Or the effects of the elastic modulus of the lining an increase in the elastic modulus of the lining leads to an increase in the lining stiffness and so an increase in the lining forces.

5.1.2 Radius of the Tunnel

An important principle of tunneling is that construction difficulty and support costs increase sharply as the diameter of tunnel increases due to the existence of a fractured and jointed ground mass. [7] The load on the lining will be increased if the ground is loosened due to the existence of such a weak zone.

Keeping other parameters constant and vary the radius of tunnel shows change on the bending moment and normal force induced in the lining. And take tunnel depth to be 60m to satisfy basic requirement.

Table 4Variation of Tunnel radius Vs Maximum bending moment and Maximum normal force

Tunnel radius (r) m	Bending moment (kNm/m)	Normal force (kN/m)
3	229.137	1857.676
4	190.142	2533.901
5	156.52	3219.863
6	130.892	3918.406
7	111.425	4631.021
8	96.347	5358.443
9	84.401	6101.054

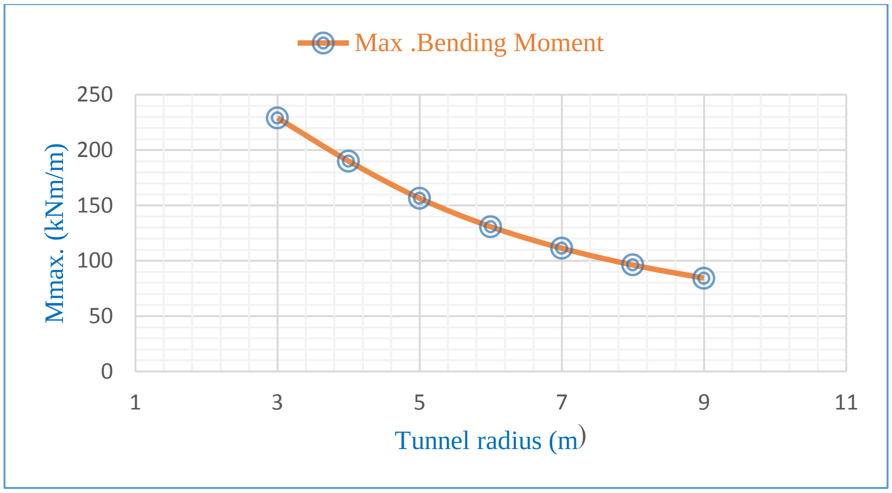


Fig 15 Maximum bending moment Vs tunnel radius

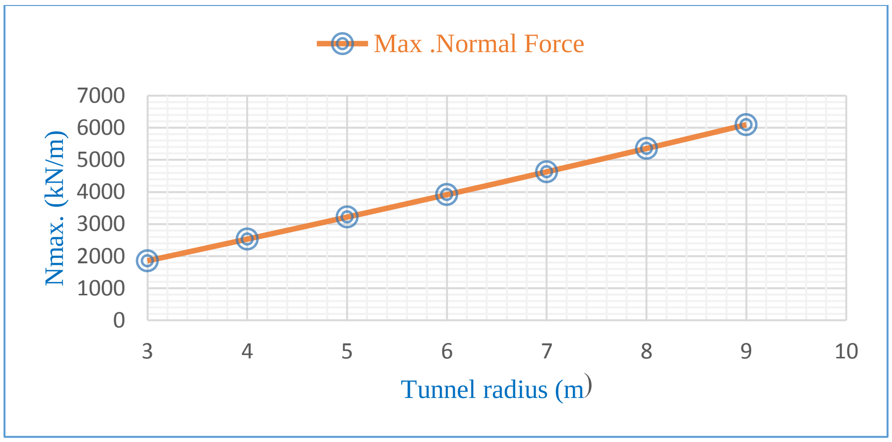


Fig 16Tunnel radius Vs Maximum Normal force

Conclusion on result

The effects of the tunnel radius are illustrated in Figures15 and 16. Figure 15 shows that the maximum bending moments decrease with the increase of the tunnel radius. This can be explained by the fact that an increase in the tunnel radius leads to a decrease in the bending stiffness of the lining.

Therefore, the lining acts as a membrane in which significant bending moments cannot develop. In the same manner, the increase of the axial forces can be explained by membrane behaviour of the lining.

5.1.3 Depth and Coefficient of Lateral Earth Pressure

Due to the difficulties of determining lateral earth pressure coefficient (Ko) and because of the issues of determining the ground mechanics, there have always been experiments to estimate Ko. Tunnel cover is one main reasons for the variation of Ko. [2]

Sheorey (1994) developed an elasto-static thermal stress model of the earth. He did provide a simplified equation which can be used for estimating the horizontal to vertical stress ratio **ko.** [18]

$$k = 0.25 + 7E_h\left(0.001 + \frac{1}{z}\right)$$

Where z (m) is the depth below surface

Eh (GPa) is the average deformation in a horizontal direction

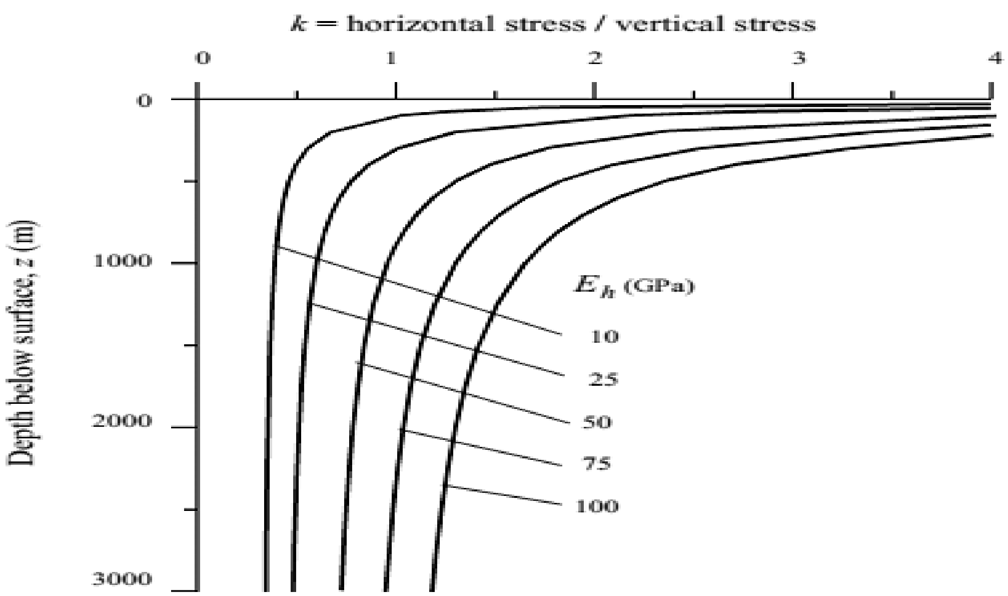


Fig 17Ratio of horizontal to vertical stress for different deformation moduli based upon Shorey's equation. [18]

For the different reasons, there is no definitive value for K0 .However, one can statistically define the range of Ko as 0.1 to 3.0.

Table 5Variation of Coefficient of lateral earth pressure Vs Maximum bending moment and Maximum normal force

Coefficient of lateral earth pressure (Ko)	Bending moment (kNm/m)	Normal force (kN/m)
0.2	181.165	485.039
0.3	156.463	645.098
0.4	131.697	805.157
0.5	106.866	965.215
0.6	81.972	1125.74
0.7	57.013	1285.332
0.8	31.989	1445.391

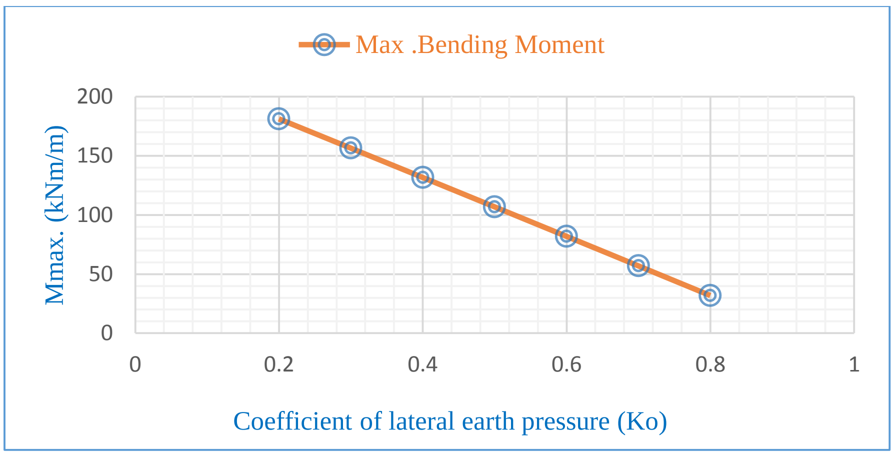


Fig 18Lateral earth pressure coefficient (Ko) Vs Maximum bending moment

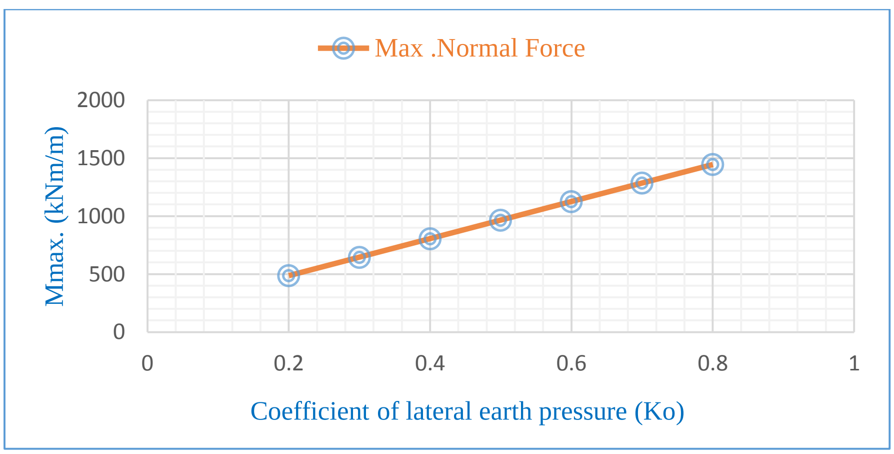


Fig 19Lateral earth pressure coefficient (Ko) Vs Maximum Normal force

Conclusion on result

Figures 17 and 18 show the effect of the lateral earth pressure. In the solutions, the maximum bending moment developed in the lining decreases rapidly with the increase in the coefficient of the lateral earth pressure because a relatively uniform compressive stress develops in the lining.

Table 6Variation of Tunnel depth Vs Maximum bending moment and Maximum normal force

Tunnel depth (m)	Bending moment (kNm/m)	Normal force (kN/m)
30	106.866	965.215
35	126.945	1113.959
40	147.141	1262.702
45	167.458	1411.446
50	187.895	1560.189
55	208.455	1708.933
60	229.137	1857.676

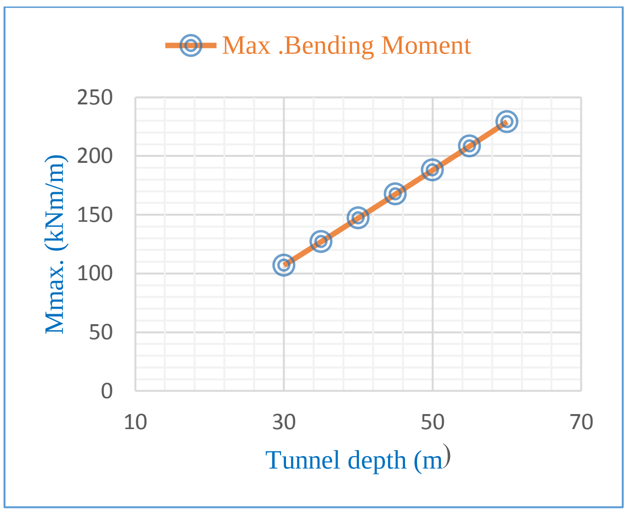


Fig 20Maximum bending moment Vs depth of tunnel

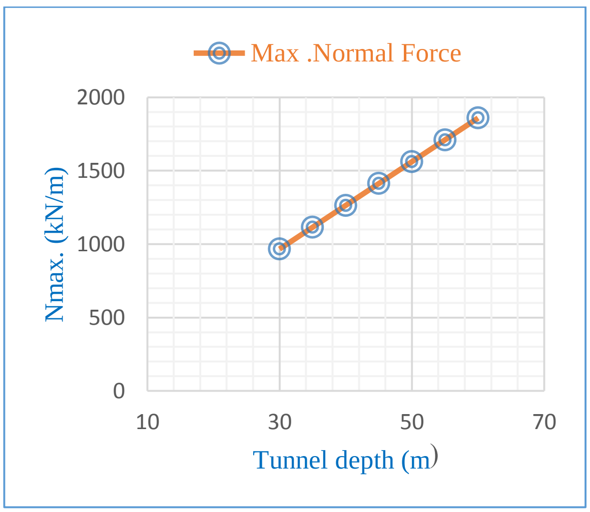


Fig 21Maximum normal force Vs. depth of tunnel

As shown in figure 19 & 20 the tunnel depth has proportional effect on the bending moment and axial force. And increase in depth increases the maximum internal forces on support.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

- Closed form solution of stress analysis can be used for preliminary analysis purpose, as it is developed by taking several assumptions.
- Analytical continuum tunnel design method is applicable for the depth to diameter ratio be three times and for weak rock ground condition
- Tunnel support design is an iterative process, process including assumptions on support type installed and evaluating the support pressure it provides
- The developed software can used as a tool for preliminary designs with simple and quick performance. In addition it can serve as an educational aid tool in the institutions.

6.2 Recommendation

The provision of other methods of analytical tunnel design with software applications and integration of additional combinations like; condition for the ground formation with stratified soil and a presence of ground water. This would led to create an option for rapid assessments and comparison on the results of numerical and analytic methods for practical purpose.

REFERENCES

[1]. B. H. G. Brady & E. T. Brown. (2004). Rock mechanics for underground mining. *3rd ed, pp173-210, Kluwer, USA.*

[2]. David Chapman, Nicole Metje and Alfred Stärk. (2010).Introduction to Tunnel Construction.*1st ed.pp91-110,383-395, USA*

[3] Ercüment .K.(2010). Design of Excavation and Support systems for the Çubukbelitunnel in Antalya. *Thesis submitted for Master of Science in Mining engineering.*

[4]. Fabrizio Barpi & Daniele Peila. (2012). Influence of the Tunnel shape on Shotcrete lining Stresses.

[5] Fikre. H. (2015). Detaille Tunnel Design. Lecture Notes, CENG 7032.Tunneling Techniques. *Addis Ababa Institute of Technology*.

[6]. Gunter G. Gschwandtner , Robert Galler. (2011). Input to the Application of the Convergence confinement method with Time-dependent Material behaviour of the Support. *ScienceDirect, Tunnelling and Underground Space Technology*

[7]. Hak J. Kim. (1997). Estimation for Tunnel Lining Loads. *A Thesis submitted for the degree of Doctor of Philosophy in Geotechnical Engineering. University of Alberta*.

[8]. International Tunneling Association (ITA) (1988). Guidelines for the Design of Tunnels. *Tunnelling and Underground Space Technology, Vol. 3, No. 3. pp. 237-249, Great Britain*

[9]. John A. Hudson.(1997). Engineering rock Mechanics an Introduction to the Principle. *1st ed pp 41-52, UK*

[10]. K.J. Bakker. (2003).Structural design of Linings for Bored Tunnels in Soft ground. *HERON,Vol.48,No, 1,Netherland.*

[11]. Leonel I. Almanzar, Mohammad R.Jafari. (2012).Design parameters Sensitivity Analysis of Precast concrete Segmental Tunnel liners. *North American Tunneling Proceedings 2012.*

[12]. Massoud palassi, Mohammad M. Mohebbi. (2008). Design of Lining of Tunnels Excavated in Soil and Soft Rock. *Electronic Journal of Geotechnical Engineering, Vol. 13, Bund. E. PP1-15*.

[13]. P.Jovanovic, FVahle. (1995). Study of literature: Structural analyses and design methods for bored tunnels in soft ground. *Bored Railway Tunnels in the Netherlands, Holland Railconsult, PP 35-40.*

[14]. P.T.Chong ,S.KTang, T.L. Lim, S.Sugawara.(2011). Bored Tunnel Lining Design in Soft Soils - A Comparison between Analytical and Numerical Analyses. *F11, PP 1-7, Singapore*.

[15] The British Tunnelling Society and the Institution of Civil Engineers. (2004).Tunnel lining design guide. *London, Great Britain*

[16]. U.S. Department of Transportation Federal Highway Administration. (2009).Technical Manual for Design and Construction of Road Tunnels-Civil Elements. *Pub.No.FHWA-NHI-10-034*

[17]. W.F.Balwden, E.Hoek, & K.Kalser. Support of Underground Excavation in Hard Rock.

[18]. WWW.home.agh.edu.pl/...cala/...chapter10.pdf . *In situ and induced stresses. September 17, 9:25am.*

Appendix 1Flow chart

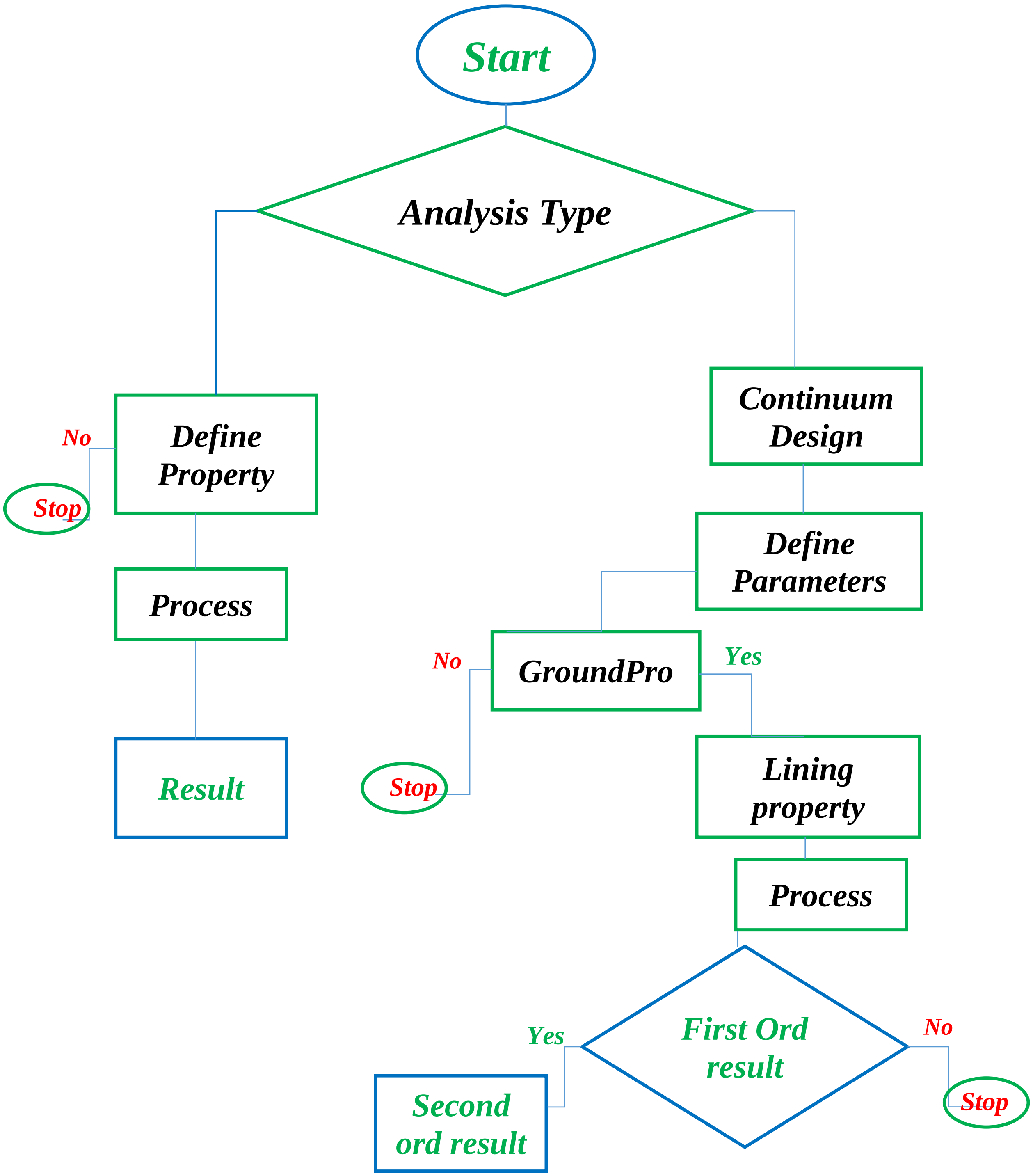


Fig 22Flow chart

Appendix 2Example Problems

This portion of the paper comprises analysis examples which have been done by hand calculation and by using the developed software. At the end the results obtained from the two methods compared and it's showed that the solutions are similar.

Example for a tunnel at King's Cross Station, London [2]

The figure below shows the schematic of the geology associated with a tunnel at King's Cross Station in London, UK.

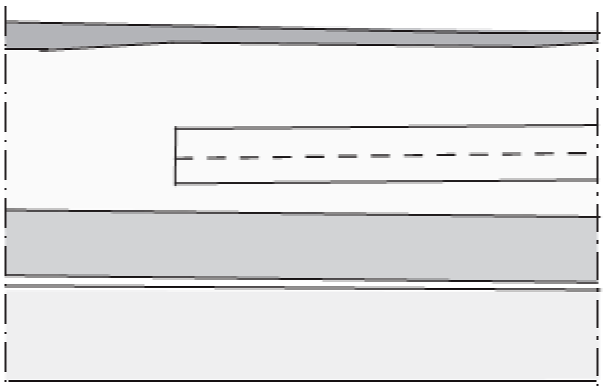


Fig 23 Schematic showing the geology around the tunnel

The overburden is assumed to be 25.0 m.

A) Ground parameters

Unit weight (γ) = 20kN/m^3

Stiffness/Modulus of compressibility (E_s) =87MN/m^2

Shear modulus (G) = 25 MPa

Poisson's ratio (μ) = 0.15

Coefficient of lateral earth pressure (K_o) = 1.2

Radius to system axis (r) = 3.05m

The overburden is assumed to be (h)=25.0 m.

B) Structural system parameters

The material is sprayed concrete (C25) with young's modulus of E=2.88x10^{4}MPa.

Profile parameters: Area (A) = 0.175m^2/m

Moment of inertia (I) = 4.466x10^{-4} m^4/m

External deformation (W_o) = 5.104x10^{-3} m^3/m

Internal deformation (W_i) = 5.104x10^{-3} m^3/m

Allowable compressive strength ($\sigma_{c,c}$) = 11.3 MPa

1) Secondary stress calculation

Use equations of Kirsch and calculated as follows:

Total stress components at a distance of r=6m, with an angle of 90°at crown of tunnel:

a=radius =3.05m

r= radial distance =6m

, $p = \gamma * h$

$p = 20 * 25$

P = 500kN/m²

$$\sigma_{rr} = \frac{500}{2}[(1+1.2)(1-\frac{3.05^2}{6^2}) - (1-1.2)(1-4\frac{3.05^2}{6^2}+3\frac{3.05^4}{6^4})\cos 2(90)$$

=**399.559 kN/m²**

$$\sigma_{\theta\theta} = \frac{500}{2}[(1+1.2)(1+\frac{3.05^2}{6^2}) + (1-1.2)(1+3\frac{3.05^4}{6^4})\cos 2(90)$$

= **752.426kN/m2**

$$\sigma_{r\theta} = \frac{500}{2}[(1-1.2)(1+\frac{2*(3.05)^2}{6^2} - 3\frac{3.05^4}{6^4})\sin 2(90)$$

= **0.0kN/m²**

Displacement induced by excavation:

$$u_r = -\frac{500*3.05^2}{4*25*10^3*6}[(1+1.2)-(1-1.2_o)\{4(1-0.15)-\frac{3.05^2}{6^2}\}\cos 2(90)]$$

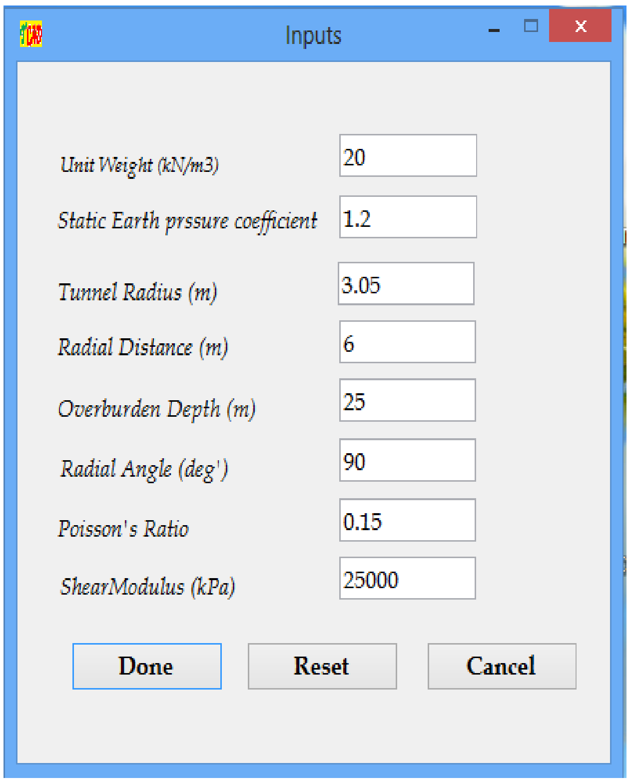
$u_r =$ **- 0.013m**

$$u_\theta = -\frac{500*3.05^2}{4*25*10^3*6}[(1-1.2)\{2(1-2*0.15)+\frac{3.05^2}{6^2}\}\sin 2(90)]$$

$u_\theta =$**0.0m**

For similar ground property, the problem is solved by the use of developed application software.

Software result displays



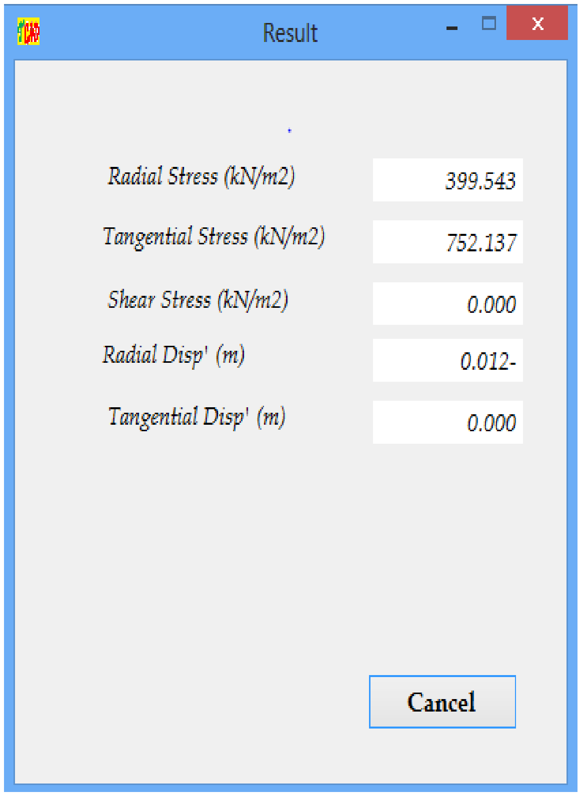


Fig 24Example stress analysis result

The final out put of the appliacation for the given problem is analyzed with single click at quick performance and displays in precise three decimal places. It is similar to the results determined from hand calculation.

2) Continuum Tunnel lining design solution

Take r= radius of system axis for this analysis =3.05m

Step 1: Earth pressure calculation

$P_{ro} = 0.5 * 20 * [25 + (25 + 3.05) * 1.2]$ = **586.97kN/m²**

$P_{r2} = P_{t2} = 0.5 * 20 * [25 - (25 + 3.05) * 1.2]$ = **-86.6kN/m²**

Tunnel support stiffness: EI = $2.88 \times 10^{4} * 4.466 \times 10^{-4}$

EI =

Elastic modulus $E_k = \dfrac{(1 + 0.15)(1 - 2 * 0.15)}{(1 - 0.15)} * 87$

E_k= 82.4MPa

Step 2: Determination of internal forces

Assumptions EA →∞ axial stiffness 1st order theory

Wo=0

Radial displacement $$\overline{w_2} = \frac{-86.6 + 0.5 * -86.6}{\dfrac{1}{(3 - 0.15 - 4 * 0.15^2)} * (2.25 - 1.5 * 0.15) * \dfrac{82400}{3.05} + \dfrac{9 * 12860}{3.05^4}}$$

$\overline{w_2}$ = **-0.0061m**

Tangential displacement $\overline{v_2} = \overline{0.5 * w_2}$

$\overline{v_2} = -0.0030 m$

Partial load $\overline{P_{ro}}^R$, (the total constant partial load P_{ro} is carried by the circular ring system (EA →∞)):

$\overline{P_{ro}}^R$ = **586.97 kN/m²**

Load parts $\overline{P_{r2}}^R$, $\overline{P_{t2}}^R$

The load part acting on the ring derived from the earth pressure load p_{r2}& p_{t2}

Load carried by continuum:

$$P_{r2}{}^{S} = \frac{82400}{3.05} * \frac{1}{(3 - 0.15 - 4 * 0.15^2)} * [(5 - 6 * 0.15) * (-0.006) + (-4 + 6 * 0.15) - 0.0030]$$

=-153.24**kN/m²**

$$\overline{P_{r2}{}^{R}} = P_{r2} - P^{S}{}_{r2}$$

$$\overline{P_{r2}{}^{R}} = \mathbf{66.4kN/m^2}$$

$$P_{t2}{}^{S} = \frac{82400}{3.05} * \frac{1}{(3 - 0.15 - 4 * 0.15^2)} * [(-4 + 6 * 0.15) * -0.006 + (5 - 6 * 0.15\mu) - 0.0030]$$

=63.11**kN/m²**

$$\overline{P_{t2}{}^{R}} = P_{t2} - P^{S}{}_{t2} = -86.6 - 63.11$$

$$\overline{P_{t2}{}^{R}} = \mathbf{-149.7kN/m^2}$$

Step 3: Proportional internal force parameter:

$$N_0 = -r * P^{R}{}_{ro}\ ,\ \ N_0 = -3.05 * P^{R}{}_{ro}$$

N_0 =-1789**kN/m²**

$$Q_0 = 0$$

$$M_0 = 0$$

Load Part $P^{R}{}_{r2}, P^{R}{}_{t2}$

$$,\ N_2 = \frac{3.05}{3} * (2 * P^{R}{}_{t2} + P^{R}{}_{r2},) \cos 2\theta\ ,\ N_2 = \frac{3.05}{3} * (2 * -149.7 + 66.64) \cos 2\theta$$

N_2 =**236.6 kN/m²**

$$Q_2 = -\frac{r}{3} * (P^{R}{}_{t2} + 2 * P^{R}{}_{r2},) \sin 2\theta,\ \ Q_2 = -\frac{3.05}{3} * (-149.7 + 2 * 66..64,) \sin 2(90)$$

Q_2 = **0.0kN/m²**

$M_2 = \frac{r^2}{6} * (P^R{}_{t2} + 2 * P^R{}_{r2},) \cos 2\theta$, $M_2 = \frac{3.05^2}{6} * (-149.7 + 2 * 66.64,) \cos 2(90)$

$M_2 =$ **25.86kNm/m**

Step 4: Final internal parameters and deformation:

$N = N_0 + N_2$ = **-1552.5 kN/m²**

$Q = Q_2$ =0.0 **kN/m²**

$M = M_2$ =**25.86kNm/m**

$W = w_2$=-**0.0064m**

Step 5: Second order theory calculation

Formwork dependent deformation Wv =r/200

Wv= 3.05/200 =0.0152

Coefficient ω: $\omega = \frac{W_{v,F}}{W_2} = \frac{1.525cm}{0.0061}$ =2.481

Equivalent stiffness: $\overline{EI} = EI + N(1+\omega) - \frac{r^2}{3} = EI + N(1+\omega) - \frac{3.05^2}{3}$

$\overline{EI} =$ **21076kPa**

New radial deformation is determined form modified stiffness:

$$w^{II}{}_2 = \frac{-86.86 + 0.5 * -86.6}{\frac{1}{(3 - 0.15 - 40.15^2)} * (2.25 - 1.5 * 0.15) * \frac{82394.11}{3.05} + \frac{9 * \overline{21076.09}}{3.05^4}} =$$

$w^{II}{}_2$=-**0.0059m**

Check the value of ω: $\omega = \frac{W_{v,F}}{W_2{}^{II}} = \frac{1.525}{0.0059}$ =2.584 =/=2.481

Second iteration **ω=2.5914**

Third itration **ω=2.5911 = 2.5914** use this value

New iterated W2=-0.00588m

Then moment from second order is given as: ,normal force is equal to firs order result

$$M_2^{II} = \frac{3*EI}{r^2} * W_2^{II} * \cos 2\theta = \frac{3*EI}{3.05^2} * W_2^{II} * \cos 2\theta$$

M_2^{II}=**24.376kNm/m**

Step 6: Stress analysis

Once the value of internal forces determined, it is possible to perform stress analysis for the selection of appropriate section of lining material.

$$\sigma = \frac{N}{A} + \frac{M}{w}$$

The crown stresses are:

Outside $\sigma_o = -\frac{2025.76}{0.175} - \frac{24.376}{0.0051}$ =**| -6.77|MPa < 11.3MPa** **Ok**

Inside $\sigma_i = -\frac{2025.76}{0.175} + \frac{24.376}{0.0051}$ =**|-16.3 |MPa >11.3Mpa** **Not ok!**

The sidewall stresses are:

Inside $\sigma_i = -\frac{1552.49}{0.175} + \frac{24.376}{0.0051}$ =**|-4.09 |MPa < 11.3MPa** **ok**

Outside $\sigma_o = -\frac{2025.76}{0.175} - \frac{24.376}{0.0051}$ = **| -13.64|MPa > 11.3MPa** **Not ok**!

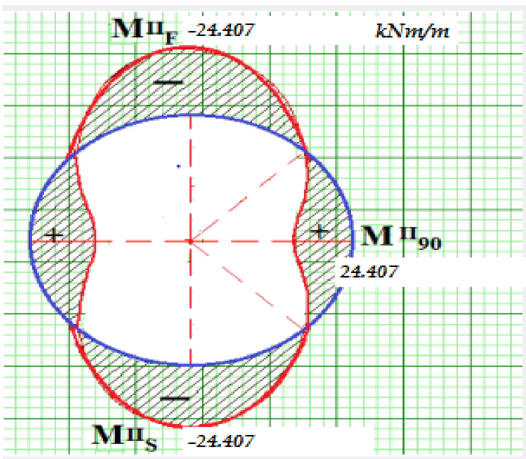
The invert (Floor) stresses are:

Outside $\sigma_o = -\dfrac{2025.76}{0.175} + \dfrac{24.376}{0.0051}$ = | **-6.79|MPa < 11.3MPa** **ok**

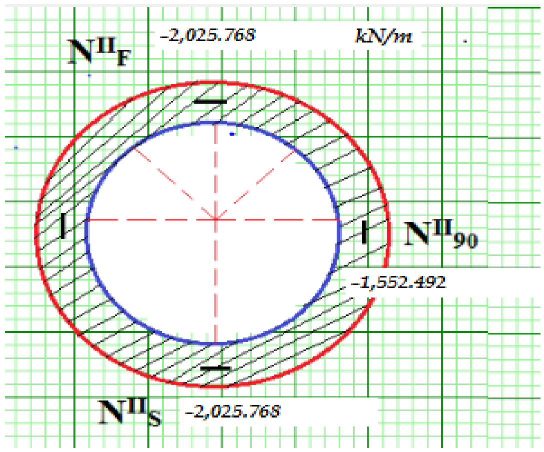
Inside $\sigma_i = -\dfrac{2025.76}{0.175} - \dfrac{24.376}{0.0051}$ = | **-16.35|MPa >11.3MPa** **Not ok**!

This results shows that an induced tensile stress in the contour of the lining is greater than the allowable tensile strength of the sprayed concrete support. But compressive stresses are in the allowable range.

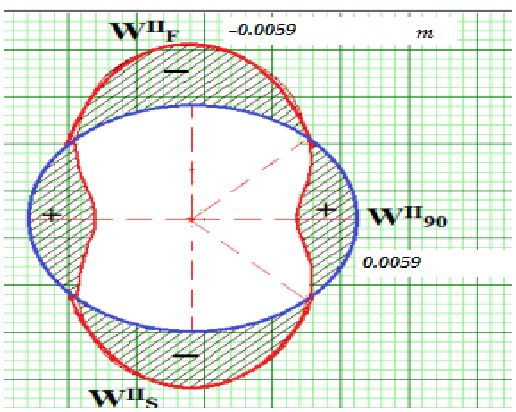
Step 7: Presentation of the internal forces and the deformation



Moment



Normal force



Radial displacements

Fig 25Moments, Normal forces and Radial displacement

❖ Software Results for a given data at radial angle of 90º

- For similar lining material parameters and ground property, the problem is solved by the use of developed application software. And the results are presented in figures below.

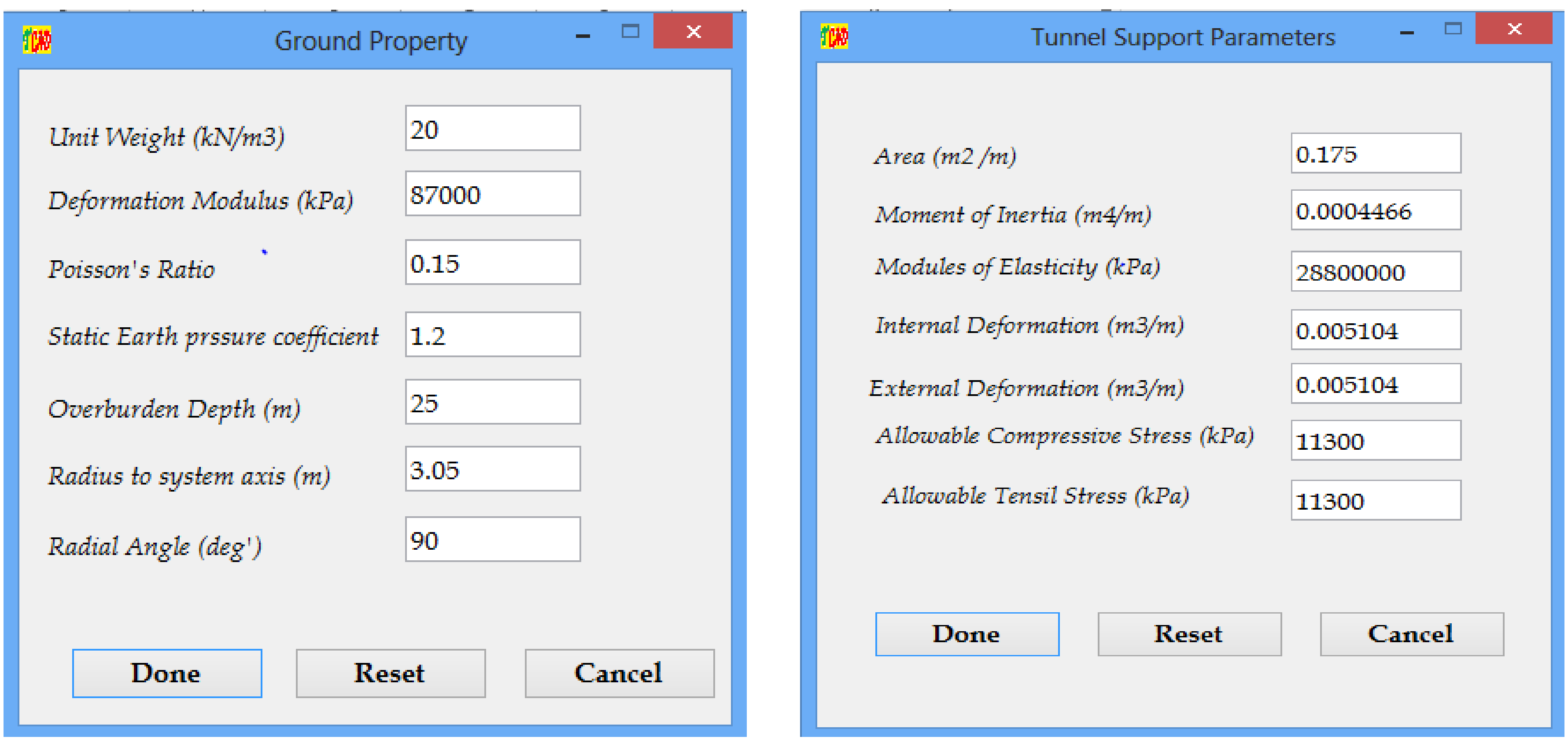


Fig 26Ground and support material properties.

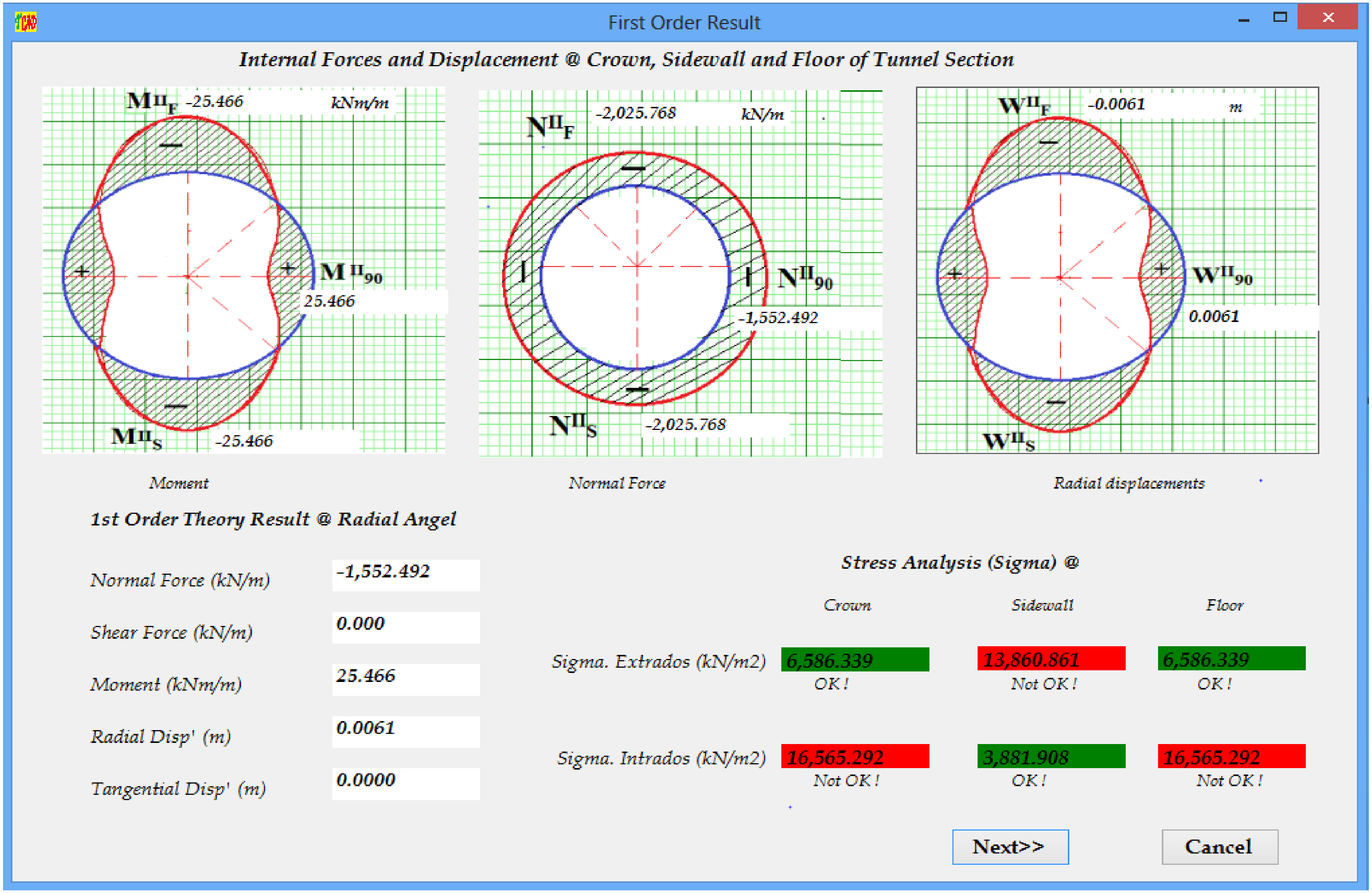


Fig 27 Firs order theory results.

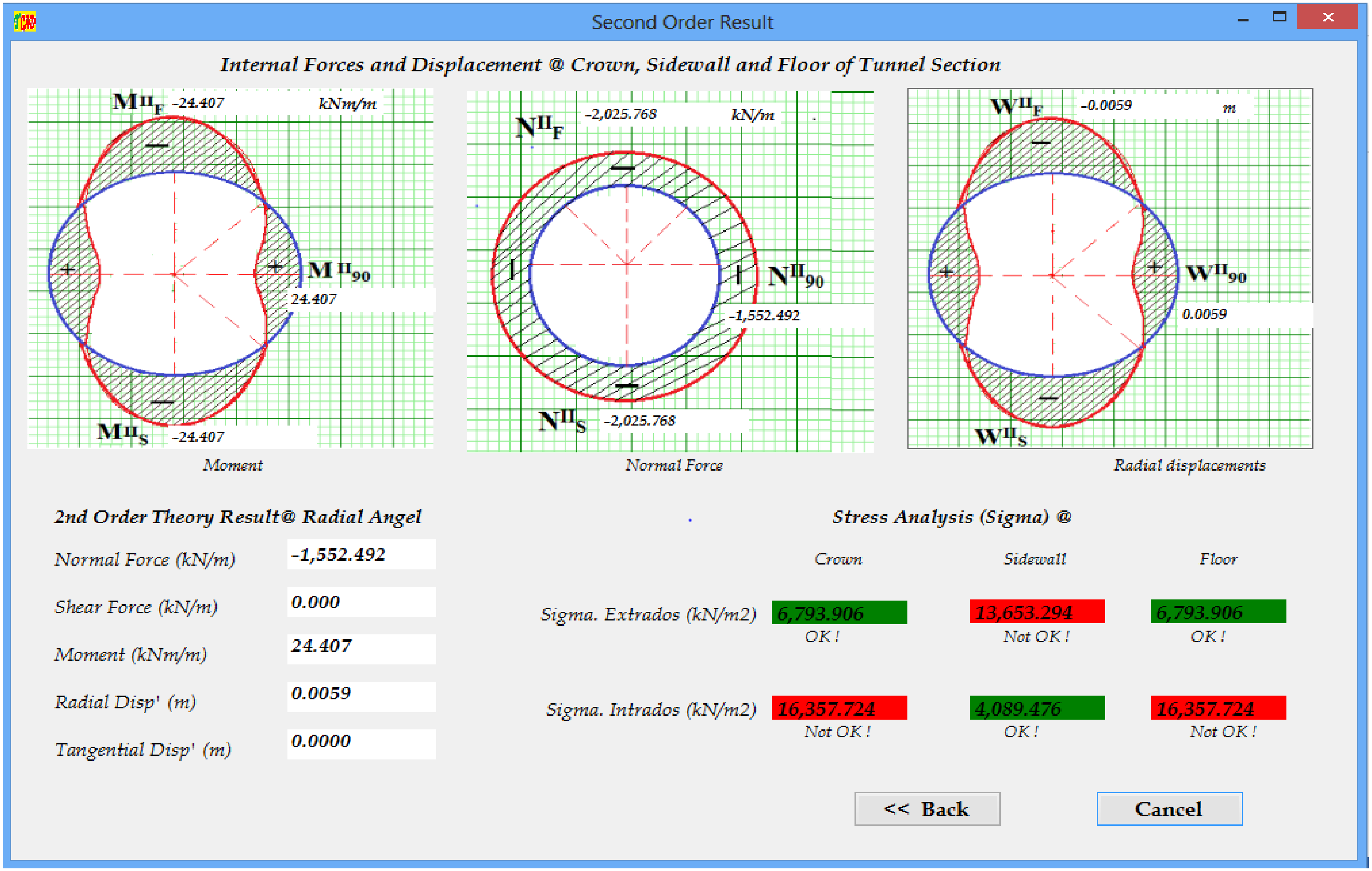


Fig 28Second order theory results.

Conclusion on results

The final output of the application for the given problem is analyzed with a single click at quick performance and displays in precise three decimal places.

Result display of the application presented in numerical values and pictorial representation of internal force parameters. Stress analysis results are displayed in numerical values with color coding and lables (Ok! Not O!) simultaneously that represent whether section is adequate to resist induced stress or not.

If green light is displays with a term “Ok!” it represents that induced stress is less than an allowable strength of liner. Conversely if red color is displays with a term “Not Ok!” shows an induced stress is out of range or greater than allowable stress of the support.

In both calculations the analysis results are similar to the values determined from hand calculation.

Appendix 3Microsoft. VB.NET programing code for the application

Continuum method of Tunnel lining design VB.Net Codes

1. Ground Property entry codes

```
Public Class frmGroundProperty
    Private Sub ContGroundReset()

        txtUnitWeight.Text = ""
        txtDeformModulus.Text = ""
        txtPoisons.Text = ""
        txtEarthpcoeff.Text = ""
        txtDepth.Text = ""
        txtRadius.Text = ""
        txtTheta.Text = ""

    End Sub
    Private Sub cmdContGroundReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdContGroundReset.Click
        ContGroundReset()
    End Sub


    Private Sub cmdContGoundDone_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdContGoundDone.Click
        'Checking Input Data
        Dim dblUnitWieght As Double
        Dim dblDeformModulus As Double
        Dim dblPoisons As Double
        Dim dblRadius As Double
        Dim dblDepth As Double
        Dim dblDiameter As Double
        Dim dblTheta As Double
        Dim dblEarthpcoeff As Double

        If txtUnitWeight.Text <> "" Then
            If IsNumeric(txtUnitWeight.Text) = True Then
                dblUnitWieght = CDbl(Trim(txtUnitWeight.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 101")
                Exit Sub
            End If
        Else
            MsgBox("Insert Unit Weight of Soil.", vbOKOnly, "Error Input 101")
            Exit Sub
        End If
        If txtDeformModulus.Text <> "" Then
            If IsNumeric(txtDeformModulus.Text) = True Then
                dblDeformModulus = CDbl(Trim(txtDeformModulus.Text))
            Else
```

```
                MsgBox("Insert a number .", vbOKOnly, "Error Input 102")
                Exit Sub
            End If
        Else
            MsgBox("Insert Deformation Modulus of Soil.", vbOKOnly, "Error Input 102")
            Exit Sub
        End If
        If txtPoisons.Text <> "" Then
            If IsNumeric(txtPoisons.Text) = True Then
                dblPoisons = CDbl(Trim(txtPoisons.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 103")
                Exit Sub
            End If
        Else
            MsgBox("Insert the Poisons Ratio.", vbOKOnly, "Error Input 103")
            Exit Sub
        End If

        If txtEarthpcoeff.Text <> "" Then
            If IsNumeric(txtEarthpcoeff.Text) = True Then
                dblEarthpcoeff = CDbl(Trim(txtEarthpcoeff.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 104")
                Exit Sub
            End If
        Else
            MsgBox("Insert Static EarthPressure coefficient.", vbOKOnly, "Error Input
104")
            Exit Sub
        End If
        If txtDepth.Text <> "" Then
            If IsNumeric(txtDepth.Text) = True Then
                dblDepth = CDbl(Trim(txtDepth.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 105")
                Exit Sub
            End If
        Else
            MsgBox("Insert Overburden Depth.", vbOKOnly, "Error Input 105")
            Exit Sub
        End If
        If txtRadius.Text <> "" Then
            If IsNumeric(txtRadius.Text) = True Then
                dblDepth = CDbl(Trim(txtRadius.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 106")
                Exit Sub
            End If
        Else
```

```
            MsgBox("Insert Tunnel Radius.", vbOKOnly, "Error Input 106")
            Exit Sub
        End If
        If txtTheta.Text <> "" Then
            If IsNumeric(txtTheta.Text) = True Then
                dblTheta = CDbl(Trim(txtTheta.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 107")
                Exit Sub
            End If
        Else
            MsgBox("Insert Radial Angle.", vbOKOnly, "Error Input 107")
            Exit Sub
        End If

        'Chcking H > 3D
        dblDepth = CDbl(txtDepth.Text)
        dblDiameter = (6 * CDbl(txtRadius.Text))
        dblRadius = CDbl(txtRadius.Text)

        If dblDepth > dblDiameter Then
        Else
            MsgBox(" Overburden depth should grater 3 times Tunnel Radius.Please check
the ratio properly!! ", vbExclamation, "Fatal Error")
            Exit Sub
        End If
        Me.Visible = False
        frmTunnHomePage.Enabled = True
        frmTunnHomePage.Visible = True
        frmTunnHomePage.Show()
        frmTunnHomePage.LinningPropertyToolStripMenuItem.Enabled = True

    End Sub



    Private Sub cmdContGoundExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdContGoundExit.Click
        'Me.cmdContGroundReset
        frmTunnHomePage.Enabled = True
        frmTunnHomePage.Show()
        Me.Visible = False
        ContGroundReset()
        frmTunnHomePage.LinningPropertyToolStripMenuItem.Enabled = False
        frmTunnHomePage.ProcessToolStripMenuItem.Enabled = False

    End Sub

    Private Sub frmGroundProperty_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
```

```
        e.Cancel = True
    End Sub

End Class
```

2. Support property entry codes

```
    Public Class frmLinningProperty
    Private Sub ContLiningReset()

        txtArea.Text = ""
        txtMomentInertia.Text = ""
        txtModulusElasticity.Text = ""
        txtInternalDeform.Text = ""
        txtExternalDeform.Text = ""
        txtAllowableComStress.Text = ""
        txtAllowableTenStress.Text = ""
    End Sub


    Private Sub cmdContLiningExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdContLiningExit.Click
        frmTunnHomePage.Visible = True
        Me.Visible = False
        ContLiningReset()
        frmTunnHomePage.ProcessToolStripMenuItem.Enabled = False
    End Sub

    Private Sub cmdContLiningReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdContLiningReset.Click
        ContLiningReset()
    End Sub


    Private Sub cmdContLiningCalculate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdContLiningCalculate.Click



        'Lining property=========

        Dim dblArea As Double
        Dim dblMomentinertia As Double
        Dim dblModuluseElasticity As Double
        Dim dblInternaldeform As Double
        Dim dblExternaldeform As Double
        Dim dblAllowableComStress As Double
        Dim dblAllowabletenstress As Double
        Dim dblInitformdeform As Double
        Dim dblRadialstress1 As Double
        Dim dblRadialstress2 As Double
        Dim dblTangentialstress2 As Double
        Dim dblYoungesmodulus As Double
```

```
If txtArea.Text <> "" Then
    If IsNumeric(txtArea.Text) = True Then
        dblArea = CDbl(Trim(txtArea.Text))
    Else
        MsgBox("Insert a number .", vbOKOnly, "Error Input 108")
        Exit Sub
    End If
Else
    MsgBox("insert Lining section Area", vbOKOnly, "Error 108")
    Exit Sub
End If

If txtMomentInertia.Text <> "" Then
    If IsNumeric(txtMomentInertia.Text) = True Then
        dblMomentinertia = CDbl(Trim(txtMomentInertia.Text))
    Else
        MsgBox("Insert a number .", vbOKOnly, "Error Input 109")
        Exit Sub
    End If
Else
    MsgBox("insert Moment Inertia", vbOKOnly, "Error 109")
    Exit Sub
End If


If txtModulusElasticity.Text <> "" Then
    If IsNumeric(txtModulusElasticity.Text) = True Then
        dblModuluseElasticity = CDbl(Trim(txtModulusElasticity.Text))
    Else
        MsgBox("Insert a number .", vbOKOnly, "Error Input 110")
        Exit Sub
    End If
Else
    MsgBox("insert ModuluseElasticity", vbOKOnly, "Error 110")
    Exit Sub
End If


If txtInternalDeform.Text <> "" Then
    If IsNumeric(txtInternalDeform.Text) = True Then
        dblInternaldeform = CDbl(Trim(txtInternalDeform.Text))
    Else
        MsgBox("Insert a number .", vbOKOnly, "Error Input 111")
        Exit Sub
    End If
Else
    MsgBox("insert Internal lining displacement", vbOKOnly, "Error 111")
    Exit Sub
End If
```

```
        If txtExternalDeform.Text <> "" Then
            If IsNumeric(txtExternalDeform.Text) = True Then
                dblExternaldeform = CDbl(Trim(txtExternalDeform.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 112")
                Exit Sub
            End If
        Else
            MsgBox("insert External lining displacement", vbOKOnly, "Error 112")
            Exit Sub
        End If


        If txtAllowableComStress.Text <> "" Then
            If IsNumeric(txtAllowableComStress.Text) = True Then
                dblAllowableComStress = CDbl(Trim(txtAllowableComStress.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 113")
                Exit Sub
            End If
        Else
            MsgBox("insert Allowable Compressive strength of support", vbOKOnly, "Error
113")
            Exit Sub
        End If

        If txtAllowableTenStress.Text <> "" Then
            If IsNumeric(txtAllowableTenStress.Text) = True Then
                dblAllowabletenstress = CDbl(Trim(txtAllowableTenStress.Text))
            Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 114")
                Exit Sub
            End If
        Else
            MsgBox("insert Allowable Tensile strength of support", vbOKOnly, "Error 114")
            Exit Sub
        End If

        Me.Visible = False
        frmTunnHomePage.Enabled = True
        frmTunnHomePage.Show()
        frmTunnHomePage.ProcessToolStripMenuItem.Enabled = True


    End Sub

    Private Sub frmLinningProperty_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        e.Cancel = True
```

```
    End Sub

End Class
```

3. Analysis (calculation) and home page codes

```
Public Class frmTunnHomePage

    Private Sub GroundPropertyToolStripMenuItem_Click(ByVal sender As System.Object, 
ByVal e As System.EventArgs) Handles GroundPropertyToolStripMenuItem.Click
        frmGroundProperty.show()
        Me.Visible = False

    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles ExitToolStripMenuItem.Click
        End
    End Sub

    Private Sub LinningPropertyToolStripMenuItem_Click(ByVal sender As System.Object, 
ByVal e As System.EventArgs) Handles LinningPropertyToolStripMenuItem.Click
        ProcessToolStripMenuItem.Enabled = False
        frmLinningProperty.Show()
        Me.Visible = False
    End Sub

    Private Sub ProcessToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles ProcessToolStripMenuItem.Click

        'calculation declaration===========

        Dim dblLiningstiffness As Double
        Dim dblRadialdeform2 As Double
        Dim dblRadialdeform As Double
        Dim dblTangentialdeform2 As Double
        Dim dblTangentialdeform As Double
        Dim dblRadiallinigload1 As Double
        Dim dblRadiallinigload2 As Double
        Dim dblRadialsoilload2 As Double
        Dim dblTangentiallininglod2 As Double
        Dim dblTangentialsoilload2 As Double
        Dim dblNormalF As Double
        Dim dblNormalload1 As Double
        Dim dblNormalload2 As Double
        Dim dblShearF As Double
        Dim dblShearload2 As Double
        Dim dblMoment As Double
        Dim dblMoment2 As Double

        'Second order theory=========

        Dim dblFormworkdeform As Double
```

```
Dim dblOmega As Double
Dim dblEIBar As Double
Dim dblOmegaNew As Double
Dim dblNewOmega As Double
Dim dblNewRadialdeform2 As Double
Dim dblSecondOrMoment As Double
Dim dblRadialdeformSe As Double
Dim dblTangentialdeformSe As Double
Dim dblTangentialdeform2Se As Double
Dim dblOmegaNew2 As Double
Dim dblOmegaNew3 As Double
'Section property checking==========
'calculated allowable first stress==========

Dim dblNormalFCF As Double
Dim dblNormalFCU As Double
Dim dblNormalFCS As Double
Dim dblSecondOrMomentCF As Double
Dim dblSecondOrMomentCU As Double
Dim dblSecondOrMomentCS As Double
Dim dblRadialdeformSeCF As Double
Dim dblRadialdeformSeCU As Double
Dim dblRadialdeformSeCS As Double
Dim dblSecondOrderMoment As Double
Dim dblSigmaFo As Double
Dim dblSigmaFi As Double
Dim dblAbsSigmaFo As Double
Dim dblAbsSigmaFi As Double
'calculated allowable ulm stress==========

Dim dblSigmaUo As Double
Dim dblSigmaUi As Double
Dim dblAbsSigmaUo As Double
Dim dblAbsSigmaUi As Double

'calculated allowable Sole stress==========

Dim dblSigmaSo As Double
Dim dblSigmaSi As Double
Dim dblAbsSigmaSo As Double
Dim dblAbsSigmaSi As Double

'First order theory results===========

Dim dblMoment1CF As Double
Dim dblMoment1CU As Double
Dim dblMoment1CS As Double
Dim dblRadialdeform1CF As Double
Dim dblRadialdeform1CU As Double
Dim dblRadialdeform1CS As Double
```

```
        Dim dblSigma1Fo As Double
        Dim dblSigma1Fi As Double
        Dim dblSigma1Uo As Double
        Dim dblSigma1Ui As Double
        Dim dblSigma1So As Double
        Dim dblSigma1Si As Double
        Dim dblMomentt As Double

        'Variable calling
        'ground property
        dblUnitWieght = frmGroundProperty.txtUnitWeight.Text
        dblDeformModulus = frmGroundProperty.txtDeformModulus.Text
        dblPoisons = frmGroundProperty.txtPoisons.Text
        dblRadius = frmGroundProperty.txtRadius.Text
        dblDepth = frmGroundProperty.txtDepth.Text
        dblTheta = frmGroundProperty.txtTheta.Text
        dblEarthpcoeff = frmGroundProperty.txtEarthpcoeff.Text

        'liningn property
        dblArea = frmLinningProperty.txtArea.Text
        dblMomentinertia = frmLinningProperty.txtMomentInertia.Text
        dblModuluseElasticity = frmLinningProperty.txtModulusElasticity.Text
        dblInternaldeform = frmLinningProperty.txtInternalDeform.Text
        dblExternaldeform = frmLinningProperty.txtExternalDeform.Text
        dblAllowableComStress = frmLinningProperty.txtAllowableComStress.Text
        dblAllowabletenstress = frmLinningProperty.txtAllowableTenStress.Text

        'Calculation Process===========

        dblRadialstress1 = (0.5 * dblUnitWieght * (dblDepth + ((dblDepth + dblRadius) *
dblEarthpcoeff)))
        dblRadialstress2 = (0.5 * dblUnitWieght * (dblDepth - ((dblDepth + dblRadius) *
dblEarthpcoeff)))
        dblTangentialstress2 = dblRadialstress2
        dblYoungesmodulus = ((((1 + dblPoisons) * (1 - (2 * dblPoisons))) / (1 -
dblPoisons)) * dblDeformModulus)
        'EI
        dblLiningstiffness = (dblModuluseElasticity * dblMomentinertia)

        dblRadialdeform2 = (dblRadialstress2 + (0.5 * dblTangentialstress2)) / (((((2.25
- (1.5 * dblPoisons)) / (3 - dblPoisons - (4 * (dblPoisons ^ 2)))) * (dblYoungesmodulus /
dblRadius)) + ((9 * dblLiningstiffness) / (dblRadius ^ 4))))
        dblTangentialdeform2 = (0.5 * dblRadialdeform2)
        'load distribution for lining and soil====

        dblRadialsoilload2 = (((dblYoungesmodulus / dblRadius) * ((1) / ((3) -
(dblPoisons) - ((4) * (dblPoisons) ^ 2)))) * ((((5) - ((6) * (dblPoisons))) *
(dblRadialdeform2)) + (((-4) + ((6) * (dblPoisons))) * (dblTangentialdeform2))))
```

```
        dblTangentialsoilload2 = ((dblYoungesmodulus / dblRadius) * ((1) / ((3) -
(dblPoisons) - ((4) * (dblPoisons) ^ 2)) * ((((-4) + ((6) * (dblPoisons))) *
(dblRadialdeform2)) + (((5) - ((6) * (dblPoisons))) * (dblTangentialdeform2)))))

        dblRadiallinigload1 = dblRadialstress1
        dblRadiallinigload2 = (dblRadialstress2 - dblRadialsoilload2)
        dblTangentialliningload2 = (dblTangentialstress2 - dblTangentialsoilload2)

        'Load portion========
        dblNormalload1 = (-(dblRadius) * (dblRadiallinigload1))
        dblNormalload2 = (((dblRadius) / (3)) * ((2 * dblTangentiallinigload2) +
(dblRadiallinigload2)) * (Math.Cos((2) * (dblTheta / 57.29578))))
        dblShearload2 = ((((-dblRadius) / (3))) * ((dblTangentiallinigload2) + (2 *
dblRadiallinigload2)) * (Math.Sin((2) * (dblTheta / 57.29578))))
 dblMoment2 = (((dblRadius) ^ 2 / (6)) * ((dblTangentiallinigload2) + (2 *
dblRadiallinigload2)) * (Math.Cos((2) * (dblTheta / 57.29578))))

        'First order theory results===========
        dblNormalF = (dblNormalload1 + dblNormalload2)
       dblShearF = dblShearload2
        dblMoment = dblMoment2

        dblRadialdeform = (((dblRadialdeform2) * (Math.Cos((2) * (dblTheta /
57.29578)))))

        dblTangentialdeform = ((dblTangentialdeform2) * (Math.Sin((2) * (dblTheta /
57.29578))))

        'First order theory results===========

        dblNormalFCF = (((-dblRadius) * (dblRadiallinigload1)) + (((dblRadius) / (3)) *
((2 * dblTangentiallinigload2) + (dblRadiallinigload2)) * (Math.Cos((2) * (0 /
57.29578)))))
        dblNormalFCU = (((-dblRadius) * (dblRadiallinigload1)) + (((dblRadius) / (3)) *
((2 * dblTangentiallinigload2) + (dblRadiallinigload2)) * (Math.Cos((2) * (90 /
57.29578)))))
        dblNormalFCS = (((-dblRadius) * (dblRadiallinigload1)) + (((dblRadius) / (3)) *
((2 * dblTangentiallinigload2) + (dblRadiallinigload2)) * (Math.Cos((2) * (180 /
57.29578)))))

        dblMoment1CF = (((dblRadius) ^ 2 / (6)) * ((dblTangentiallinigload2) + (2 *
dblRadiallinigload2)) * (Math.Cos((2) * (0 / 57.29578))))
        dblMoment1CU = (((dblRadius) ^ 2 / (6)) * ((dblTangentiallinigload2) + (2 *
dblRadiallinigload2)) * (Math.Cos((2) * (90 / 57.29578))))
        dblMoment1CS = (((dblRadius) ^ 2 / (6)) * ((dblTangentiallinigload2) + (2 *
dblRadiallinigload2)) * (Math.Cos((2) * (180 / 57.29578))))

        dblRadialdeform1CF = (((dblRadialdeform2) * (Math.Cos((2) * (0 / 57.29578)))))
        dblRadialdeform1CU = (((dblRadialdeform2) * (Math.Cos((2) * (90 / 57.29578)))))
        dblRadialdeform1CS = (((dblRadialdeform2) * (Math.Cos((2) * (180 / 57.29578)))))
```

```
        'stress analysis
        'First order moment without theta for stress analysis calc and checking

        dblMomentt = ((dblRadius) ^ 2 / (6)) * ((dblTangentialliningload2) + (2 *
dblRadiallinigload2))

        If dblMomentt > 0 Then
            'First=========
            'Outside =======
            dblSigma1Fo = ((dblNormalFCF) / (dblArea)) - ((Math.Abs(dblMoment1CF)) /
(dblExternaldeform))
            'inside====
            dblSigma1Fi = ((dblNormalFCF) / (dblArea)) + ((Math.Abs(dblMoment1CF)) /
(dblInternaldeform))
            'Ulm====
            Outside =====
            dblSigma1Uo = ((dblNormalFCU) / (dblArea)) + ((Math.Abs(dblMoment1CU)) /
(dblExternaldeform))
            'inside====
            dblSigma1Ui = ((dblNormalFCU) / (dblArea)) - ((Math.Abs(dblMoment1CU)) /
(dblInternaldeform))
            'Sole====
            'Outside ====
            dblSigma1So = ((dblNormalFCS) / (dblArea)) - ((Math.Abs(dblMoment1CS)) /
(dblExternaldeform))
           'inside=====
            dblSigma1Si = ((dblNormalFCS) / (dblArea)) + ((Math.Abs(dblMoment1CS)) /
(dblInternaldeform))

        Else
            'First=====
            'Outside ====
            dblSigma1Fo = ((dblNormalFCF) / (dblArea)) + ((Math.Abs(dblMoment1CF)) /
(dblExternaldeform))

            'Ulm====
            'Outside =====
            dblSigma1Uo = ((dblNormalFCU) / (dblArea)) - ((Math.Abs(dblMoment1CU)) /
(dblExternaldeform))
            ‘inside====
            dblSigma1Ui = ((dblNormalFCU) / (dblArea)) + ((Math.Abs(dblMoment1CU)) /
(dblInternaldeform))
            'Sole====
            'Outside ====
            dblSigma1So = ((dblNormalFCS) / (dblArea)) + ((Math.Abs(dblMoment1CS)) /
(dblExternaldeform))
            'inside=====
            dblSigma1Si = ((dblNormalFCS) / (dblArea)) - ((Math.Abs(dblMoment1CS)) /
(dblInternaldeform))
```

```
        End If
        If dblMomentt > 0 Then
            'Stress analysis at critical section============= frmFirstOrdResult
            frmFirstOrdResult2.lblSigmaFo.Text =
FormatNumber(Math.Abs(dblSigma1Fo).ToString, 3)

            If Math.Abs(dblSigma1Fo) > dblAllowableComStress Then
                frmFirstOrdResult2.lblSigmaFo.BackColor = Color.Red
                frmFirstOrdResult2.lblOk.Text = "Not OK !"
            Else
                frmFirstOrdResult2.lblSigmaFo.BackColor = Color.Green
                frmFirstOrdResult2.lblOk.Text = "OK !"
            End If

            frmFirstOrdResult2.lblSigmaFi.Text =
FormatNumber(Math.Abs(dblSigma1Fi).ToString, 3)

            If Math.Abs(dblSigma1Fi) > dblAllowabletenstress Then
                frmFirstOrdResult2.lblSigmaFi.BackColor = Color.Red
                frmFirstOrdResult2.lblOk4.Text = "Not OK !"
            Else
                frmFirstOrdResult2.lblSigmaFi.BackColor = Color.Green
                frmFirstOrdResult2.lblOk4.Text = "OK !"
            End If

            frmFirstOrdResult2.lblSigmaUo.Text =
FormatNumber(Math.Abs(dblSigma1Uo).ToString, 3)

            If Math.Abs(dblSigma1Uo) > dblAllowableComStress Then
                frmFirstOrdResult2.lblSigmaUo.BackColor = Color.Red
                frmFirstOrdResult2.lblOk2.Text = "Not OK !"
            Else
                frmFirstOrdResult2.lblSigmaUo.BackColor = Color.Green
                frmFirstOrdResult2.lblOk2.Text = " OK !"
            End If


            frmFirstOrdResult2.lblSigmaUi.Text =
FormatNumber(Math.Abs(dblSigma1Ui).ToString, 3)

            If Math.Abs(dblSigma1Ui) > dblAllowableComStress Then
                frmFirstOrdResult2.lblSigmaUi.BackColor = Color.Red
                frmFirstOrdResult2.lblOk5.Text = "Not OK !"
            Else
                frmFirstOrdResult2.lblSigmaUi.BackColor = Color.Green
                frmFirstOrdResult2.lblOk5.Text = "OK !"
            End If

            frmFirstOrdResult2.lblSigmaSo.Text =
FormatNumber(Math.Abs(dblSigma1So).ToString, 3)
```

```
            If Math.Abs(dblSigma1So) > dblAllowableComStress Then
                frmFirstOrdResult2.lblSigmaSo.BackColor = Color.Red
                frmFirstOrdResult2.lblOk3.Text = "Not OK !"
            Else
                frmFirstOrdResult2.lblSigmaSo.BackColor = Color.Green
                frmFirstOrdResult2.lblOk3.Text = "OK !"
            End If

            frmFirstOrdResult2.lblSigmaSi.Text =
FormatNumber(Math.Abs(dblSigma1Si).ToString, 3)

            If Math.Abs(dblSigma1Si) > dblAllowabletenstress Then
                frmFirstOrdResult2.lblSigmaSi.BackColor = Color.Red
                frmFirstOrdResult2.lblOk6.Text = "Not OK !"
            Else
                frmFirstOrdResult2.lblSigmaSi.BackColor = Color.Green
                frmFirstOrdResult2.lblOk6.Text = "OK !"
            End If

            'First order theory Ansuwers display 1 +ve moment===============

            frmFirstOrdResult2.lblNormaFA.Text = FormatNumber(dblNormalF.ToString, 3)
            frmFirstOrdResult2.lblShearFA.Text = FormatNumber(dblShearF.ToString, 3)
            frmFirstOrdResult2.lblMomentA.Text = FormatNumber(dblMoment.ToString, 3)
            frmFirstOrdResult2.lblRadialdeformA.Text =
FormatNumber(dblRadialdeform.ToString, 4)

            frmFirstOrdResult2.lblTangentialdeformA.Text =
FormatNumber(dblTangentialdeform.ToString, 4)

            'Internal forces At critical points , first , ulm ,sole for +ve moment

            frmFirstOrdResult2.lblNormalFCF.Text = FormatNumber(dblNormalFCF.ToString, 3)
            frmFirstOrdResult2.lblNormalFCU.Text = FormatNumber(dblNormalFCU.ToString, 3)
            frmFirstOrdResult2.lblNormalFCS.Text = FormatNumber(dblNormalFCS.ToString, 3)
            frmFirstOrdResult2.lblMoment1CF.Text = FormatNumber(dblMoment1CF.ToString, 3)
            frmFirstOrdResult2.lblMoment1CU.Text = FormatNumber(dblMoment1CU.ToString, 3)
            frmFirstOrdResult2.lblMoment1CS.Text = FormatNumber(dblMoment1CS.ToString, 3)

            frmFirstOrdResult2.lblRadialdeform1CF.Text =
FormatNumber(dblRadialdeform1CF.ToString, 4)

            frmFirstOrdResult2.lblRadialdeform1CU.Text =
FormatNumber(dblRadialdeform1CU.ToString, 4)

        Else
            'Stress analysis at critical section=============
            frmFirstOrdResult.lblSigmaFo.Text =
FormatNumber(Math.Abs(dblSigma1Fo).ToString, 3)
```

```
            If Math.Abs(dblSigma1Fo) > dblAllowableComStress Then
                frmFirstOrdResult.lblSigmaFo.BackColor = Color.Red
                frmFirstOrdResult.lblOk.Text = "Not OK !"
            Else
                frmFirstOrdResult.lblSigmaFo.BackColor = Color.Green
                frmFirstOrdResult.lblOk.Text = "OK !"
            End If

            frmFirstOrdResult.lblSigmaFi.Text =
FormatNumber(Math.Abs(dblSigma1Fi).ToString, 3)

            If Math.Abs(dblSigma1Fi) > dblAllowableComStress Then
                frmFirstOrdResult.lblSigmaFi.BackColor = Color.Red
                frmFirstOrdResult.lblOk4.Text = "Not OK !"
            Else
                frmFirstOrdResult.lblSigmaFi.BackColor = Color.Green
                frmFirstOrdResult.lblOk4.Text = "OK !"
            End If

            frmFirstOrdResult.lblSigmaUo.Text =
FormatNumber(Math.Abs(dblSigma1Uo).ToString, 3)

            If Math.Abs(dblSigma1Uo) > dblAllowableComStress Then
                frmFirstOrdResult.lblSigmaUo.BackColor = Color.Red
                frmFirstOrdResult.lblOk2.Text = "Not OK !"
            Else
                frmFirstOrdResult.lblSigmaUo.BackColor = Color.Green
                frmFirstOrdResult.lblOk2.Text = " OK !"
            End If
            frmFirstOrdResult.lblSigmaUi.Text =
FormatNumber(Math.Abs(dblSigma1Ui).ToString, 3)

            If Math.Abs(dblSigma1Ui) > dblAllowabletenstress Then
                frmFirstOrdResult.lblSigmaUi.BackColor = Color.Red
                frmFirstOrdResult.lblOk5.Text = "Not OK !"
            Else
                frmFirstOrdResult.lblSigmaUi.BackColor = Color.Green
                frmFirstOrdResult.lblOk5.Text = "OK !"
            End If
            frmFirstOrdResult.lblSigmaSo.Text =
FormatNumber(Math.Abs(dblSigma1So).ToString, 3)

            If Math.Abs(dblSigma1So) > dblAllowableComStress Then
                frmFirstOrdResult.lblSigmaSo.BackColor = Color.Red
                frmFirstOrdResult.lblOk3.Text = "Not OK !"
            Else
                frmFirstOrdResult.lblSigmaSo.BackColor = Color.Green
                frmFirstOrdResult.lblOk3.Text = "OK !"
            End If
```

```
            frmFirstOrdResult.lblSigmaSi.Text =
FormatNumber(Math.Abs(dblSigma1Si).ToString, 3)

            If Math.Abs(dblSigma1Si) > dblAllowabletenstress Then
                frmFirstOrdResult.lblSigmaSi.BackColor = Color.Red
                frmFirstOrdResult.lblOk6.Text = "Not OK !"
            Else
                frmFirstOrdResult.lblSigmaSi.BackColor = Color.Green
                frmFirstOrdResult.lblOk6.Text = "OK !"
            End If

            'First order theory Ansuwers display 1 -ve moment

            frmFirstOrdResult.lblNormaFA.Text = FormatNumber(dblNormalF.ToString, 3)

            frmFirstOrdResult.lblShearFA.Text = FormatNumber(dblShearF.ToString, 3)

            frmFirstOrdResult.lblMomentA.Text = FormatNumber(dblMoment.ToString, 3)

            frmFirstOrdResult.lblRadialdeformA.Text =
FormatNumber(dblRadialdeform.ToString, 4)

            frmFirstOrdResult.lblTangentialdeformA.Text =
FormatNumber(dblTangentialdeform.ToString, 4)

            'Internal forces At critical points , first , ulm ,sole for -ve moment

            frmFirstOrdResult.lblNormalFCF.Text = FormatNumber(dblNormalFCF.ToString, 3)
            frmFirstOrdResult.lblNormalFCU.Text = FormatNumber(dblNormalFCU.ToString, 3)
            frmFirstOrdResult.lblNormalFCS.Text = FormatNumber(dblNormalFCS.ToString, 3)
           frmFirstOrdResult.lblMoment1CF.Text = FormatNumber(dblMoment1CF.ToString, 3)
            frmFirstOrdResult.lblMoment1CU.Text = FormatNumber(dblMoment1CU.ToString, 3)
            frmFirstOrdResult.lblMoment1CS.Text = FormatNumber(dblMoment1CS.ToString, 3)
            frmFirstOrdResult.lblRadialdeform1CF.Text =
FormatNumber(dblRadialdeform1CF.ToString, 4)

            frmFirstOrdResult.lblRadialdeform1CU.Text =
FormatNumber(dblRadialdeform1CU.ToString, 4)

        End If
        If dblMomentt > 0 Then

            Me.Visible = False
            Me.Enabled = False
            frmFirstOrdResult2.Visible = True
            frmFirstOrdResult2.Show()
            frmFirstOrdResult2.Enabled = True
        Else
            Me.Visible = False
```

```
            Me.Enabled = False
            frmFirstOrdResult.Visible = True
            frmFirstOrdResult.Show()
            frmFirstOrdResult.Enabled = True
        End If


        'Second order theory calculation=============================================
        'Loop Calculation
        dblFormworkdeform = (dblRadius / 200)
        dblOmega = (dblFormworkdeform / dblRadialdeform2)
        dblOmegaNew3 = dblOmega
        dblOmegaNew2 = dblOmega

        dblEIBar = (dblLiningstiffness) + (((dblNormalload1 * (1 + dblOmega))) *
((dblRadius ^ 2) / 3))
        dblNewRadialdeform2 = (dblRadialstress2 + (0.5 * dblTangentialstress2)) /
((((2.25 - (1.5 * dblPoisons)) / (3 - dblPoisons - (4 * (dblPoisons ^ 2))) *
(dblYoungesmodulus / dblRadius)) + ((9 * dblEIBar) / (dblRadius ^ 4)))
        dblNewOmega = (dblFormworkdeform / dblNewRadialdeform2)

        Do Until dblOmegaNew2 = dblNewOmega


            dblOmegaNew2 = dblNewOmega
            dblEIBar = (dblLiningstiffness) + (((dblNormalload1 * (1 + dblNewOmega))) *
((dblRadius ^ 2) / 3))
            dblNewRadialdeform2 = (dblRadialstress2 + (0.5 * dblTangentialstress2)) /
((((2.25 - (1.5 * dblPoisons)) / (3 - dblPoisons - (4 * (dblPoisons ^ 2))) *
(dblYoungesmodulus / dblRadius)) + ((9 * dblEIBar) / (dblRadius ^ 4)))
            dblNewOmega = (dblFormworkdeform / dblNewRadialdeform2)
            dblOmegaNew3 = dblNewOmega - dblOmegaNew2
        Loop

        'After loop ======

        dblSecondOrMoment = (((3 * (dblLiningstiffness)) / (dblRadius) ^ 2) *
(dblNewRadialdeform2) * (Math.Cos((2) * (dblTheta / 57.29578))))
        dblRadialdeformSe = ((dblNewRadialdeform2) * (Math.Cos((2) * (dblTheta /
57.29578))))
        dblTangentialdeform2Se = (0.5 * dblNewRadialdeform2)
        dblTangentialdeformSe = ((dblTangentialdeform2Se) * (Math.Sin((2) * (dblTheta /
57.29578))))

        'Checking allowable stress========

        'Set Theta to be defined insid vb code as First=0deg, Ulm=90deg , Sole=180deg
Then calculat N,Q,M=============
```

```
dblNormalFCF = (((-dblRadius) * (dblRadiallinigload1)) + (((dblRadius) / (3)) * ((2 *
dblTangentialliningload2) + (dblRadiallinigload2)) * (Math.Cos((2) * (0 / 57.29578)))))

dblNormalFCU = (((-dblRadius) * (dblRadiallinigload1)) + (((dblRadius) / (3)) * ((2 *
dblTangentialliningload2) + (dblRadiallinigload2)) * (Math.Cos((2) * (90 / 57.29578)))))

dblNormalFCS = (((-dblRadius) * (dblRadiallinigload1)) + (((dblRadius) / (3)) * ((2 *
dblTangentialliningload2) + (dblRadiallinigload2)) * (Math.Cos((2) * (180 / 57.29578)))))

dblSecondOrMomentCF = ((((3) * (dblLiningstiffness)) / ((dblRadius) ^ 2)) *
(dblNewRadialdeform2) * (Math.Cos((2) * (0 / 57.29578))))

dblSecondOrMomentCU = ((((3) * (dblLiningstiffness)) / ((dblRadius) ^ 2)) *
(dblNewRadialdeform2) * (Math.Cos((2) * (90 / 57.29578))))

dblSecondOrMomentCS = ((((3) * (dblLiningstiffness)) / ((dblRadius) ^ 2)) *
(dblNewRadialdeform2) * (Math.Cos((2) * (180 / 57.29578))))

dblRadialdeformSeCF = ((dblNewRadialdeform2) * (Math.Cos((2) * (0 / 57.29578))))

dblRadialdeformSeCU = ((dblNewRadialdeform2) * (Math.Cos((2) * (90 / 57.29578))))

dblRadialdeformSeCS = ((dblNewRadialdeform2) * (Math.Cos((2) * (180 / 57.29578))))

        'Second order moment without theta for stress analysis calc and checking
        '
dblSecondOrderMoment = ((3 * (dblLiningstiffness)) / ((dblRadius) ^ 2)) *
(dblNewRadialdeform2)

        If dblSecondOrderMoment > 0 Then
            'First=====
            'Outside ====
           dblSigmaFo = ((dblNormalFCF) / (dblArea)) - ((Math.Abs(dblSecondOrMomentCF)) /
(dblExternaldeform))

            dblAbsSigmaFo = Math.Abs(dblSigmaFo)

            'inside====
            dblSigmaFi = ((dblNormalFCF) / (dblArea)) + ((Math.Abs(dblSecondOrMomentCF))
/ (dblInternaldeform))

            dblAbsSigmaFi = Math.Abs(dblSigmaFi)

            'Ulm====
            'Outside =====
            dblSigmaUo = ((dblNormalFCU) / (dblArea)) + ((Math.Abs(dblSecondOrMomentCU))
/ (dblExternaldeform))

            dblAbsSigmaUo = Math.Abs(dblSigmaUo)
```

```
            'inside====
            dblSigmaUi = ((dblNormalFCU) / (dblArea)) - ((Math.Abs(dblSecondOrMomentCU))
/ (dblInternaldeform))

            dblAbsSigmaUi = Math.Abs(dblSigmaUi)

           'Sole====
            'Outside ====

            dblSigmaSo = ((dblNormalFCS) / (dblArea)) - ((Math.Abs(dblSecondOrMomentCS))
/ (dblExternaldeform))

            dblAbsSigmaSo = Math.Abs(dblSigmaSo)

           'inside=====

            dblSigmaSi = ((dblNormalFCS) / (dblArea)) + ((Math.Abs(dblSecondOrMomentCS))
/ (dblInternaldeform))
            dblAbsSigmaSi = Math.Abs(dblSigmaSi)

        Else
            'First=====
            'Outside ====
            dblSigmaFo = ((dblNormalFCF) / (dblArea)) + ((Math.Abs(dblSecondOrMomentCF))
/ (dblExternaldeform))
           dblAbsSigmaFo = Math.Abs(dblSigmaFo)

            'inside====

            dblSigmaFi = ((dblNormalFCF) / (dblArea)) - ((Math.Abs(dblSecondOrMomentCF))
/ (dblInternaldeform))

            dblAbsSigmaFi = Math.Abs(dblSigmaFi)

            'Ulm====
            'Outside =====
            dblSigmaUo = ((dblNormalFCU) / (dblArea)) - ((Math.Abs(dblSecondOrMomentCU))
/ (dblExternaldeform))

            dblAbsSigmaUo = Math.Abs(dblSigmaUo)

            'inside====

            dblSigmaUi = ((dblNormalFCU) / (dblArea)) + ((Math.Abs(dblSecondOrMomentCU))
/ (dblInternaldeform))

            dblAbsSigmaUi = Math.Abs(dblSigmaUi)

            'Sole====
            'Outside ====
```

```
            dblSigmaSo = ((dblNormalFCS) / (dblArea)) + ((Math.Abs(dblSecondOrMomentCS))
/ (dblExternaldeform))
            dblAbsSigmaSo = Math.Abs(dblSigmaSo)

            'inside=====
            dblSigmaSi = ((dblNormalFCS) / (dblArea)) - ((Math.Abs(dblSecondOrMomentCS))
/ (dblInternaldeform))
            dblAbsSigmaSi = Math.Abs(dblSigmaSi)

        End If
        'Result Lables Assignment   ====== use formating if necessary
        If dblSecondOrderMoment > 0 Then
            'Stress analysis at critical section=============

            frmOutputDisplay.lblSigmaFo.Text =
FormatNumber(Math.Abs(dblSigmaFo).ToString, 3)

            If Math.Abs(dblSigmaFo) > dblAllowableComStress Then
                frmOutputDisplay.lblSigmaFo.BackColor = Color.Red
                frmOutputDisplay.lblOk.Text = "Not OK !"
            Else
                frmOutputDisplay.lblSigmaFo.BackColor = Color.Green
                frmOutputDisplay.lblOk.Text = "OK !"
            End If

            frmOutputDisplay.lblSigmaFi.Text =
FormatNumber(Math.Abs(dblSigmaFi).ToString, 3)

            If Math.Abs(dblSigmaFi) > dblAllowabletenstress Then
                frmOutputDisplay.lblSigmaFi.BackColor = Color.Red
                frmOutputDisplay.lblOk4.Text = "Not OK !"
            Else
                frmOutputDisplay.lblSigmaFi.BackColor = Color.Green
                frmOutputDisplay.lblOk4.Text = "OK !"
            End If

            frmOutputDisplay.lblSigmaUo.Text =
FormatNumber(Math.Abs(dblSigmaUo).ToString, 3)

            If Math.Abs(dblSigmaUo) > dblAllowableComStress Then
                frmOutputDisplay.lblSigmaUo.BackColor = Color.Red
                frmOutputDisplay.lblOk2.Text = "Not OK !"
            Else
                frmOutputDisplay.lblSigmaUo.BackColor = Color.Green
                frmOutputDisplay.lblOk2.Text = " OK !"
            End If
            frmOutputDisplay.lblSigmaUi.Text =
FormatNumber(Math.Abs(dblSigmaUi).ToString, 3)

            If Math.Abs(dblSigmaUi) > dblAllowableComStress Then
```

```
                frmOutputDisplay.lblSigmaUi.BackColor = Color.Red
                frmOutputDisplay.lblOk5.Text = "Not OK !"
            Else
                frmOutputDisplay.lblSigmaUi.BackColor = Color.Green
                frmOutputDisplay.lblOk5.Text = "OK !"
            End If
            frmOutputDisplay.lblSigmaSo.Text =
FormatNumber(Math.Abs(dblSigmaSo).ToString, 3)

            If Math.Abs(dblSigmaSo) > dblAllowableComStress Then
                frmOutputDisplay.lblSigmaSo.BackColor = Color.Red
                frmOutputDisplay.lblOk3.Text = "Not OK !"
            Else
                frmOutputDisplay.lblSigmaSo.BackColor = Color.Green
                frmOutputDisplay.lblOk3.Text = "OK !"
            End If

            frmOutputDisplay.lblSigmaSi.Text =
FormatNumber(Math.Abs(dblSigmaSi).ToString, 3)

            If Math.Abs(dblSigmaSi) > dblAllowabletenstress Then
                frmOutputDisplay.lblSigmaSi.BackColor = Color.Red
                frmOutputDisplay.lblOk6.Text = "Not OK !"
            Else
                frmOutputDisplay.lblSigmaSi.BackColor = Color.Green
                frmOutputDisplay.lblOk6.Text = "OK !"
            End If
           'Second order theory Ansuwers on disply 1 +ve moment ========

            frmOutputDisplay.lblNormaSFA.Text = FormatNumber(dblNormalF.ToString, 3)
            frmOutputDisplay.lblShearSFA.Text = FormatNumber(dblShearF.ToString, 3)
           frmOutputDisplay.lblMomentSA.Text = FormatNumber(dblSecondOrMoment.ToString,3)
frmOutputDisplay.lblRadialdeformSA.Text = FormatNumber(dblRadialdeformSe.ToString, 4)
frmOutputDisplay.lblTangentialdeformSA.Text =FormatNumber(dblTangentialdeformSe.ToString,
4)

            'Internal forces At critical points , first , ulm ,sole for +ve moment

frmOutputDisplay.lblNormalFCF.Text = FormatNumber(dblNormalFCF.ToString, 3)
frmOutputDisplay.lblNormalFCU.Text = FormatNumber(dblNormalFCU.ToString, 3)
frmOutputDisplay.lblNormalFCS.Text = FormatNumber(dblNormalFCS.ToString, 3)
frmOutputDisplay.lblSecondOrMomentCF.Text = FormatNumber(dblSecondOrMomentCF.ToString, 3)
frmOutputDisplay.lblSecondOrMomentCU.Text = FormatNumber(dblSecondOrMomentCU.ToString, 3)
frmOutputDisplay.lblSecondOrMomentCS.Text = FormatNumber(dblSecondOrMomentCS.ToString, 3)
frmOutputDisplay.lblRadialdeformSeCF.Text = FormatNumber(dblRadialdeformSeCF.ToString, 4)
frmOutputDisplay.lblRadialdeformSeCU.Text = FormatNumber(dblRadialdeformSeCU.ToString, 4)

        Else
            'Stress analysis at critical section=============
```

```
            frmOutputResult2.lblSigmaFo.Text =
FormatNumber(Math.Abs(dblSigmaFo).ToString, 3)

            If Math.Abs(dblSigmaFo) > dblAllowableComStress Then
                frmOutputResult2.lblSigmaFo.BackColor = Color.Red
                frmOutputResult2.lblOk.Text = "Not OK !"
            Else
                frmOutputResult2.lblSigmaFo.BackColor = Color.Green
                frmOutputResult2.lblOk.Text = "OK !"
            End If

frmOutputResult2.lblSigmaFi.Text = FormatNumber(Math.Abs(dblSigmaFi).ToString, 3)

            If Math.Abs(dblSigmaFi) > dblAllowableComStress Then
                frmOutputResult2.lblSigmaFi.BackColor = Color.Red
                frmOutputResult2.lblOk4.Text = "Not OK !"
            Else
                frmOutputResult2.lblSigmaFi.BackColor = Color.Green
                frmOutputResult2.lblOk4.Text = "OK !"
            End If

frmOutputResult2.lblSigmaUo.Text = FormatNumber(Math.Abs(dblSigmaUo).ToString, 3)

            If Math.Abs(dblSigmaUo) > dblAllowableComStress Then
                frmOutputResult2.lblSigmaUo.BackColor = Color.Red
                frmOutputResult2.lblOk2.Text = "Not OK !"
            Else
                frmOutputResult2.lblSigmaUo.BackColor = Color.Green
                frmOutputResult2.lblOk2.Text = " OK !"
            End If


frmOutputResult2.lblSigmaUi.Text = FormatNumber(Math.Abs(dblSigmaUi).ToString, 3)

            If Math.Abs(dblSigmaUi) > dblAllowabletenstress Then
                frmOutputResult2.lblSigmaUi.BackColor = Color.Red
                frmOutputResult2.lblOk5.Text = "Not OK !"
            Else
                frmOutputResult2.lblSigmaUi.BackColor = Color.Green
                frmOutputResult2.lblOk5.Text = "OK !"
            End If

frmOutputResult2.lblSigmaSo.Text = FormatNumber(Math.Abs(dblSigmaSo).ToString, 3)

            If Math.Abs(dblSigmaSo) > dblAllowableComStress Then
                frmOutputResult2.lblSigmaSo.BackColor = Color.Red
                frmOutputResult2.lblOk3.Text = "Not OK !"
            Else
                frmOutputResult2.lblSigmaSo.BackColor = Color.Green
                frmOutputResult2.lblOk3.Text = "OK !"
```

```
            End If

frmOutputResult2.lblSigmaSi.Text = FormatNumber(Math.Abs(dblSigmaSi).ToString, 3)

            If Math.Abs(dblSigmaSi) > dblAllowabletenstress Then
                frmOutputResult2.lblSigmaSi.BackColor = Color.Red
                frmOutputResult2.lblOk6.Text = "Not OK !"
            Else
                frmOutputResult2.lblSigmaSi.BackColor = Color.Green
                frmOutputResult2.lblOk6.Text = "OK !"
            End If

            'Second order theory Answuers on disply 2 -ve moment ========

            frmOutputResult2.lblNormaSFA.Text = FormatNumber(dblNormalF.ToString, 3)

            frmOutputResult2.lblShearSFA.Text = FormatNumber(dblShearF.ToString, 3)

            frmOutputResult2.lblMomentSA.Text = FormatNumber(dblSecondOrMoment.ToString,
3)

frmOutputResult2.lblRadialdeformSA.Text = FormatNumber(dblRadialdeformSe.ToString, 4)

frmOutputResult2.lblTangentialdeformSA.Text =FormatNumber(dblTangentialdeformSe.ToString,
4)

            'Internal forces At critical points , first , ulm ,sole for -ve moment
            frmOutputResult2.lblNormalFCF.Text = FormatNumber(dblNormalFCF.ToString, 3)
frmOutputResult2.lblNormalFCU.Text = FormatNumber(dblNormalFCU.ToString, 3)
frmOutputResult2.lblNormalFCS.Text = FormatNumber(dblNormalFCS.ToString, 3)
frmOutputResult2.lblSecondOrMomentCF.Text = FormatNumber(dblSecondOrMomentCF.ToString, 3)
frmOutputResult2.lblSecondOrMomentCU.Text = FormatNumber(dblSecondOrMomentCU.ToString, 3)
frmOutputResult2.lblSecondOrMomentCS.Text = FormatNumber(dblSecondOrMomentCS.ToString, 3)
frmOutputResult2.lblRadialdeformSeCF.Text = FormatNumber(dblRadialdeformSeCF.ToString, 4)
frmOutputResult2.lblRadialdeformSeCU.Text = FormatNumber(dblRadialdeformSeCU.ToString, 4)

        End If

    End Sub

    Private Sub AboutTSToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AboutTSToolStripMenuItem.Click
        MsgBox("This Mini Analytical Tunnel Design Aid appliction is Used to calculate
secondary stress around circular opening and design of tunnel by analytical method
(Continuum).
It is developed for a project submmited to the school of graduate of Addis Ababa
Institute of Technology in partial fulfillment of the requirements for the degree of
Master of Engineering (geotechniques) under the supervision of         Dr.Ing Henok
Fikre.
```

```
Tesfamichael G/wold
2008 E.C, 15")
    End Sub

    Private Sub TipsToUseTSToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TipsToUseTSToolStripMenuItem.Click
        MsgBox("Please Use Menu bar to select analysis type. And
Use Done Button to continue calculation process & Cancle Button on opened window to
cancel operation & return to TCAD Home Page.!! ")
    End Sub


End Class
```

4. Output display codes

'First order Result

```
Public Class frmFirstOrdResult2

    Private Sub frmFirstOrdResult2_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        e.Cancel = True
    End Sub

    Private Sub cmdFirstOrdThExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdFirstOrdThExit.Click
        Me.Visible = False
        frmTunnHomePage.Enabled = True
        frmTunnHomePage.Show()
    End Sub


    Private Sub cmdFirstOrdThNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdFirstOrdThNext.Click

        If dblSecondOrderMoment > 0 Then

            Me.Visible = False
            Me.Enabled = False
            frmOutputDisplay.Show()
            frmOutputDisplay.Enabled = True
        Else
            Me.Visible = False
            Me.Enabled = False
            frmOutputResult2.Show()
            frmOutputResult2.Enabled = True

        End If
    End Sub
End Class
```

'Second order Result

```
Public Class frmOutputDisplay

    Private Sub frmOutputDisplay_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        e.Cancel = True
    End Sub

    Private Sub cmdContOutputExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdContOutExit.Click
```

```
        Me.Visible = False
        Me.Enabled = False
        frmTunnHomePage.Visible = True
        frmTunnHomePage.Show()
        frmTunnHomePage.Enabled = True
    End Sub

    Private Sub cmdSecoOutBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSecoOutBack.Click

        Dim dblMomentt As Double

        If dblMomentt > 0 Then
            Me.Visible = False
            Me.Enabled = False
            frmFirstOrdResult2.Show()
            frmFirstOrdResult2.Enabled = True

            frmFirstOrdResult.Enabled = False
            frmFirstOrdResult.Visible = False
        Else
            Me.Visible = False
            Me.Enabled = False
            frmFirstOrdResult.Show()
            frmFirstOrdResult.Enabled = True

            frmFirstOrdResult2.Visible = False
            frmFirstOrdResult2.Enabled = False

        End If


    End Sub
End Class
```

Secondary stress analysis around circular opening

1. Input parameters entry codes

```
PublicClassfrmStressInput
PrivateSub StressReset()

        txtUnitWeight.Text = ""
        txtEarthpcoeff.Text = ""
        txtRadius.Text = ""
        txtRadialdistance.Text = ""
        txtDepth.Text = ""
        txtTheta.Text = ""
        txtShearModlus.Text = ""
        txtStresspoiss.Text = ""

EndSub

PrivateSub cmdStressReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdStressReset.Click
        StressReset()
EndSub

PrivateSub cmdStressinputExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdStressinputExit.Click
frmTunnHomePage.Enabled = True
frmTunnHomePage.Show()
Me.Visible = False
        StressReset()
EndSub

PrivateSub cmdStresscalculate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdStresscalculate.Click

'Stress around opening analysis
'variable declaration
Dim dblUnitWeight AsDouble
Dim dblDepth AsDouble
Dim dblRadius AsDouble
Dim dblRadialdistance AsDouble
Dim dblTheta AsDouble
Dim dblEarthpcoeff AsDouble
Dim dblVerticalstress AsDouble
Dim dblRadialstress AsDouble
Dim dblTangentialstress AsDouble
Dim dblShearstress AsDouble
Dim dblPoisons AsDouble
Dim dblShearModuls AsDouble

'variabble Assignment
```

```
If txtUnitWeight.Text <>""Then
If IsNumeric(txtUnitWeight.Text) = TrueThen
                dblUnitWeight = Trim(txtUnitWeight.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 115")
Exit Sub
EndIf
Else
            MsgBox("Insert Unit Weight of Soil.", vbOKOnly, "Error Input 115")
Exit Sub
EndIf

If txtEarthpcoeff.Text <>""Then
If IsNumeric(txtEarthpcoeff.Text) = TrueThen
                dblEarthpcoeff = Trim(txtEarthpcoeff.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 119")
Exit Sub
EndIf
Else
            MsgBox("Insert Static Lateral Earth pressure Coefficient.", vbOKOnly, "Error
Input 119")
Exit Sub
EndIf

If txtRadius.Text <>""Then
If IsNumeric(txtRadius.Text) = TrueThen
                dblRadius = Trim(txtRadius.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 117")
Exit Sub
EndIf
Else
            MsgBox("Insert Tunnel Radius.", vbOKOnly, "Error Input 117")
Exit Sub
EndIf

If txtRadialdistance.Text <>""Then
If IsNumeric(txtRadialdistance.Text) = TrueThen
                dblRadialdistance = Trim(txtRadialdistance.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 120")
Exit Sub
EndIf
Else
            MsgBox("Insert Radial Distance.", vbOKOnly, "Error Input 120")
Exit Sub
EndIf
```

```
If txtDepth.Text <>""Then
If IsNumeric(txtDepth.Text) = TrueThen
                dblDepth = Trim(txtDepth.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 116")
Exit Sub
EndIf
Else
            MsgBox("Insert Overberden Depth.", vbOKOnly, "Error Input 116")
Exit Sub
EndIf

If txtTheta.Text <>""Then
If IsNumeric(txtTheta.Text) = TrueThen
                dblTheta = Trim(txtTheta.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 118")
Exit Sub
EndIf
Else
            MsgBox("Insert Radial Angle.", vbOKOnly, "Error Input 118")
Exit Sub
EndIf


If txtStresspoiss.Text <>""Then
If IsNumeric(txtStresspoiss.Text) = TrueThen
                dblPoisons = Trim(txtStresspoiss.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 121")
Exit Sub
EndIf
Else
            MsgBox("Insert Poisson's Ratio.", vbOKOnly, "Error Input 118")
Exit Sub
EndIf

If txtShearModlus.Text <>""Then
If IsNumeric(txtShearModlus.Text) = TrueThen
                dblShearModuls = Trim(txtShearModlus.Text)
Else
                MsgBox("Insert a number .", vbOKOnly, "Error Input 122")
Exit Sub
EndIf
Else
            MsgBox("Insert Shear Modulus.", vbOKOnly, "Error Input 118")
Exit Sub
EndIf

Me.Visible = False
```

```
frmTunnHomePage.Enabled = True
frmTunnHomePage.Visible = True
        frmTunnHomePage.ResultToolStripMenuItem.Enabled = True


EndSub



EndClass
```

2. Calculation process codes

```
    Private Sub ResultToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ResultToolStripMenuItem.Click
        'Lable Assignment
        'variable declaration
        Dim dblUnitWeight As Double
        Dim dblDepth As Double
        Dim dblRadius As Double
        Dim dblRadialdistance As Double
        Dim dblTheta As Double
        Dim dblEarthpcoeff As Double
        Dim dblPoisons As Double
        Dim dblShearModuls As Double

        Dim dblVerticalstress As Double
        Dim dblRadialstress As Double
        Dim dblTangentialstress As Double
        Dim dblShearstress As Double
        Dim dblRadialDisplacment As Double
        Dim dblTangentialDisplacment As Double

        dblUnitWeight = frmStressInput.txtUnitWeight.Text
        dblDepth = frmStressInput.txtDepth.Text
        dblRadius = frmStressInput.txtRadius.Text
        dblRadialdistance = frmStressInput.txtRadialdistance.Text
        dblTheta = frmStressInput.txtTheta.Text
        dblEarthpcoeff = frmStressInput.txtEarthpcoeff.Text
        dblPoisons = frmStressInput.txtStresspoiss.Text
        dblShearModuls = frmStressInput.txtShearModlus.Text



        'calculation
        dblVerticalstress = (dblUnitWeight * dblDepth)

        dblRadialstress = (dblVerticalstress / 2) * (((1 + (dblEarthpcoeff)) * (1 -
(dblRadius / dblRadialdistance) ^ 2)) - ((1 - (dblEarthpcoeff)) * (1 - (4 * (dblRadius /
dblRadialdistance) ^ 2) + (3 * (dblRadius / dblRadialdistance) ^ 4)) * (Math.Cos(2 *
(dblTheta / 57.29578)))))

        dblTangentialstress = (dblVerticalstress / 2) * (((1 + (dblEarthpcoeff)) * (1 +
(dblRadius / dblRadialdistance) ^ 2)) + ((1 - (dblEarthpcoeff)) * (1 + (3 * (dblRadius /
dblRadialdistance) ^ 4)) * (Math.Cos(2 * (dblTheta / 57.29578)))))

        dblShearstress = (dblVerticalstress / 2) * ((((1 - dblEarthpcoeff) * (1 + (2 *
(dblRadius / dblRadialdistance) ^ 2) - (3 * (dblRadius / dblRadialdistance) ^ 4))) *
(Math.Sin(2 * (dblTheta / 57.29578)))))
```

```
        dblRadialDisplacment = ((-1) * ((dblVerticalstress * (dblRadius) ^ 2)) / (4 *
dblShearModuls * dblRadialdistance)) * ((1 + dblEarthpcoeff) - ((1 - dblEarthpcoeff) *
((4 * (1 - dblPoisons)) - ((dblRadius / dblRadialdistance) ^ 2)) * (Math.Cos(2 *
(dblTheta / 57.29578)))))

        dblTangentialDisplacment = ((-1) * ((dblVerticalstress * (dblRadius) ^ 2)) / (4 *
dblShearModuls * dblRadialdistance)) * ((1 - dblEarthpcoeff) * ((2 * (1 - (2 *
dblPoisons)) + ((dblRadius / dblRadialdistance) ^ 2)) * (Math.Sin(2 * (dblTheta /
57.29578)))))

        'out upt form display
        frmStressResult.lblRadialstressAns.Text = FormatNumber(dblRadialstress.ToString,
3)

        frmStressResult.lblTangentialstressAns.Text =
FormatNumber(dblTangentialstress.ToString, 3)

        frmStressResult.lblShearstressAns.Text = FormatNumber(dblShearstress.ToString, 3)

        frmStressResult.lblRadialDispAns.Text =
FormatNumber(dblRadialDisplacment.ToString, 3)

        frmStressResult.lblTangentialDispAns.Text =
FormatNumber(dblTangentialDisplacment.ToString, 3)

        frmStressResult.Show()
        Me.Visible = False

    End Sub
```

3. Output display codes

```
PublicClassfrmStressResult
PrivateSub cmdstressResExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdstressResExit.Click
Me.Visible = False
frmTunnHomePage.Show()


EndSub


EndClass
```