HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY

DEPARTMENT OF CIVIL ENGINEERING

STRUCTURAL ENGINEERING POSTGRADUATE PROGRAM

COMPUTER PROGRAM FOR ANALYSIS AND DESIGN OF REINFORCED CONCRETE BEAM SECTIONS

MSc Thesis

By

Mikiyas Jabir

HAWASSA UNIVERSITY, HAWASSA, ETHIOPIA

August, 2018

COMPUTER PROGRAM FOR ANALYSIS AND DESIGN OF REINFORCED CONCRETE BEAM SECTIONS

MIKIYAS JABIR

THESIS SUBMITTED TO
HAWASSA UNIVERSITY
SCHOOL OF CIVIL ENGINERING,
INSTITUTE OF TECHNOLOGY, SCHOOL OF
GRADUATE STUDIES, HAWASSA UNIVERSITY
HAWASSA, ETHIOPIA

IN PARTILAL FULFILLMENT OF THE
REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE IN STRUCTURAL ENGINEERING
(SPECIALIZATION: STRUCTURAL ENGINEERING)

August, 2018

ACKNOWLEDGEMENTS

First of all I would like to Thanks to God because He has endowed me courage and strength as well as precious health throughout my life and satisfactory accomplishment of this thesis.

I would also like to express my sincere gratitude to my advisor Dr. Temesgen Wondimu. He provided me with invaluable information for the successful completion of my work. He has guided me throughout the preparation of this thesis by giving me feedback on time and continuously encouraging me.

Special thanks go to Ethiopian Road Authority (ERA) and Hawassa University Institute of Technology School of Civil Engineering coordinators for giving me master program and facilitating this program.

Finally, I would like to express my great thanks to my classmates and collogues for sharing reference books and important information.

DECLARATION

I declare that this thesis is my bonafied work and all sources of materials used for this thesis have been duly acknowledged. I solemnly declare that this thesis is not submitted to any other institution anywhere for the award of any academic degree, diploma or certificate.

Name: Mikiyas Jabir

Signature:

This MSc thesis has been submitted for examination with my approval as Thesis advisor.

Name: Temesgen Wondimu(PhD)

Signature:

LIST OF ACRONYMS

EBCS	Ethiopian building code standard
GA	Genetic algorithm
GUI	Graphical user interface
MATLAB	Matrix Laboratory
NA	Neutral axis
SLS	Serviceability limit state
ULS	Ultimate limit state

LIST OF SYMBOLS

Latin upper case letters

A	Outer circumference of the cross-section
Ac	Cross sectional area of concrete
$A_{s,prov}$	Area of steel provided
$A_{s,req}$	Area of steel required
A_s	Cross sectional area of reinforcement
$A_{s,min}$	Minimum cross sectional area of reinforcement
A_{sw}	Cross sectional area of shear reinforcement
A_k	The area enclosed by the center-lines of the connecting walls
A_{sl}	Longitudinal reinforcement for torsion
C	Compression force
E_{cm}	Secant modulus of elasticity of concrete
E_s	Design value of modulus of elasticity of reinforcing steel
K	The factor to take into account the different structural systems
M_{Ed}	Design value of the applied internal bending moment
T	Tensile force
T_{Ed}	Design value of the applied torsional moment
T_{Rd}	Design torsion resistance
$T_{Rd,c}$	Design torsional cracking moment
$T_{Rd,max}$	Maximum torsional resistance moment
V_{Ed}	Design value of the applied shear force
$V_{Ed,i}$	Shear force in a wall i due to torsion
V_{Rd}	Design shear resistance
$V_{Rd,c}$	Shear resistance of the member without shear reinforcement
$VR_{d,max}$	Shear resistance of a member limited by the crushing of compression struts
u	Circumference of outer edge of the cross-section (torsion)
u_k	The perimeter enclosed by the center-lines

Latin lower case letters

a	Distance
a	Geometrical data
b	Overall width of a cross-section, or actual flange width in a T or L beam
b_w	Width of the web on T, I or L beams. Minimum width between tension and compression chords
b_{eff}	Effective width of flange
b_t	Mean width of the tension zone. For a T beam with the flange in compression, only the width of the web is taken into account
b_1	Half of distance between adjacent webs of down stand beams
d	Effective depth of a cross-section
d_g	Largest nominal maximum aggregate size
d'	Effective depth cover
f_c	Compressive strength of concrete
f_{cd}	Design value of concrete compressive strength
f_{ck}	Characteristic compressive cylinder strength of concrete at 28 days
f_{ctk}	Characteristic axial tensile strength of concrete
f_{ctm}	Mean value of axial tensile strength of concrete
f_t	Tensile strength of reinforcement
f_{tk}	Characteristic tensile strength of reinforcement
f_y	Yield strength of reinforcement
f_{yd}	Design yield strength of reinforcement
f_{yk}	Characteristic yield strength of reinforcement
f_{ywd}	Design yield strength of shear reinforcement
h	Overall depth of a cross-section; height
h_f	Flange thickness
l (or L)	Length; Span
l_{eff}	Effective span
l_0	Distance between points of zero moment

t	Thickness
t_{ef}	The effective wall thickness
x	Neutral axis depth
z	Lever arm of internal forces
z_i	Side length of wall I (torsion)

Greek lower case letters

α	Angle; angle of shear link to the longitudinal axis; ratio
α_{cw}	Coefficient taking account of the state of the stress in the compression Chord
γ	Partial factor
γ_c	Partial factor for concrete
γ_f	Partial factor for action
γ_s	Partial factor for reinforcing steel
κ	Correction factor
ε_c	Compressive strain in the concrete
ε_{c1}	Compressive strain in the concrete at the peak stress f_c
ε_{cu}	Ultimate compressive strain in the concrete
ε_{cm}	Maximum concrete compression strain
ε_s	Strain of reinforcement or prestressing steel at maximum load
θ	Angle;angle of compression struts (shear)
ν	Strength reduction factor for concrete cracked in shear
ρ_o	Reference reinforcement ratio
ρ	Required tension reinforcement ratio
ρ'	Required compression reinforcement ratio
ρ_1	Reinforcement ratio for longitudinal reinforcement
ρ_w	Reinforcement ratio for shear reinforcement

σ_s	Tensile steel stress
σ_c	Compressive stress in the concrete
σ_{cu}	Compressive stress in the concrete at the ultimate compressive strain
τ	Torsional shear stress
ϕ	Diameter of a reinforcing bar or of a prestressing duct

TABLE OF CONTENTS

Contents **Page**

LIST OF TABLES

Tables **Page**

LIST OF FIGURES

Figures **Page**

ABSTRACT

The purpose of this study is to develop a computer program that can be used for analysis and design of a reinforced concrete beam sections based on the new Ethiopian building code, and also that can optimize double reinforced rectangular beam section. The computer program is developed in MATLAB programming language and graphical user interface is incorporated in the program by following procedures reuired for design problem by satisfying design criteria set by the code (ES EN 1992-1-1:2015). The program incorporate simple interface for section analysis and design of reinforced concrete rectangular, T or inverted L and trapezoidal beams for flexure and rectangular beam with vertical and inclined stirrup for shear, torsion and combined effect of shear with torsion that will help the designers to save time, minmize effort and avoid calculation error that commonly occur. A graphical user interface developed for flexure is programed for a particular beam cross-section dimensions and material properties satisfiying stress-strain compatibility and equilibrium on the basis of rectangular-parapolic stress distibuiton for rectangular beam and on the basis of equivalent rectangular stress distribution for T or inverted L and trapezoidal beam section. Finally, the moment resistance (M_{Rd}) *and area of reinforcement required to resist a design moment are determined. The two graphical user interface; for shear programmed based on truss model and for torsion programmed by considering beam as equivalent thin-walled sections. Using this interface design shear resistance of concrete* ($V_{Rd,c}$), *design shear resistance* (V_{Rd}), *design torsion resistance of concrete* ($T_{Rd,c}$) *and design torsion resistance* (T_{Rd}) *for rectangular reinforced concrete beam section with vertical or inclined stirrup is determined. Using graphical user interface combined effect of shear with torsion is also checked for rectangular beam with vertical and inclined stirrup. Dimensions of beam cross section, area of reinforcement, size of stirrup and spacing of stirrup are selected by satisfying design criteria for shear, torsion and combined effect respectively. Finally, application of the user interface is demonstrated with example for analysis and design of reinforced concrete beam section for flexure, shear, torsion and combined effect of shear with torsion .*

Key Words: *analysis, design, moment resistance, shear resistance, torsion reistance,optimization, MATLAB*

1. INTRODUCTION

1.1 Background of the Study

Concrete and reinforced concrete are used as building construction materials in every country including our country Ethiopia. Reinforced concrete is a dominant structural material in engineering construction but it needs to be analyzed and designed properly before construction to act as structural member. The overall goal of analysis and design of reinforced concrete structures is to construct most safe and efficient structure for the intended purpose. This can be attained for a part of structural member like beam to follow the principle provided in the design code standard. The new Ethiopian building code define beam as it is a member for which the span is not less than 3 times the overall section depth. Otherwise it should be considered as a deep beam. Reinforced concrete beams are not homogeneous in that they are made of two entirely different materials thus the methods used in the analysis and design of reinforced concrete beams are therefore different from those used in the design or investigation of beams composed entirely of steel, wood, or any other structural material but fundamental principles involved are, however, essentially the same (Darwin, et al., 2004). Applied loads and self-weight of beam resisted basically by internal bending or flexure and shear (Macgregor & Wight, 2012). According to Macgregor and Wight (2012) there are two different types of problems arise in the study of reinforced concrete:

1. Analysis: Given a cross section, concrete strength, reinforcement size and location, and yield strength, compute the resistance or strength. In analysis there should be one unique answer.
2. Design: Given a factored design moment, select a suitable cross section, including dimensions, concrete strength, reinforcement, and so on. In design there are many possible solutions.

In reinforced concrete construction beam is casted monolithically with floor or roof slab, the resulting beam in cross-section is either T for interior beam and inverted L for exterior or edge beam. Also classified on the basis of reinforcement provision as single and double reinforced section (Samuel, 2003).

As beam is the main structural part of building the failure has enormous impact on the safety and stability of building, so careful attention needs to be given during analysis and design. MATLAB programming languages is used as tool for analysis and design of reinforced concrete beam sections. Since the syntax and procedures to write script for the program is simple as compared to other programming language due to this it is selected to develop simple and easy graphical user interface for analysis and design of reinforced concrete beam sections.

MATLAB allows matrix manipulations, plotting of functions and data implementation of algorithms and creation of user interfaces. The MATLAB application is built around the MATLAB scripting language. Common usage of the MATLAB application involves using the Command window as an interactive mathematical shell or executing text files containing MATLAB code (Houcque, 2005).

1.2 Statement of the Problem

Reinforced concrete structures consist of a series of members that interact to support the loads placed on the structure.Therefore, safety and strength of every buildings or structures depends on stability and strength of individual structural components like beam, column, slab and their interaction. Beam is one of basic structural member which carries transverse load and transfers it to supporting column. Design of structural members using hand calculation is tedious. In addition to this design of beam using structural software is possible, but it requires training and clear understanding of the software interface to use it to accomplish the task needed.

To minimize the effort a computer program is necessary. Thus the program has integrated a simple graphical user interface developed in MATLAB programming languages.

1.3 Objective of the Study

1.3.1 General Objective

- To develop a computer program using MATLAB that can analyse and design reinforced concrete beam sections.

1.3.2 Specific Objective

- To develop a computer program using MATLAB that can be used to design different reinforced concrete beam sections, such as rectangular, T, inverted L and trapezoidal sections for flexure and for rectangular beam section with vertical and inclined stirrup for shear, torsion and combined effect of shear with torsion.

- To estimate flexural resistance of reinforced concrete beam section, such as rectangular, T or inverted L and trapezoidal beam.
- To estimate shear resistance of rectangular reinforced concrete beam section with vertical and inclined stirrup.
- To estimate torsional resistance of rectangular reinforced concrete beam section with vertical and inclined stirrup.
- To develop program for cost optimization of double reinforced rectangular beam section.

1.4 Scope of the Study

- The developed graphical user interface is used to analysis and design only prismatic reinforced concrete beam sections such as rectangular, T or invrted L, tarezoidal beam subjected to flexure and rectangular section with vertical and inclined stirrup subjected to shear,torsion and combined shear with torsion.
- The developed Optimization program is used only for simply supported beam and rectangular double reinforced concrete beam section.
- Normal strength concrete is considered according to table 1.1.
- The grade of reinforcement is considered from S260 to S600 according to table 1.2.
- The size of reinforcement that used is stated in table 1.3.

Table 1.1: Grade of concrete

Grade	12/15	16/20	20/25	25/30	30/37	35/45	40/50	45/55	50/60
fck (MPa)	12	16	20	25	30	35	40	45	50
$f_{ck,cube}$ (MPa)	15	20	25	30	37	45	50	55	60
f_{ctm} (MPa)	1.6	1.9	2.2	2.6	2.9	3.2	3.5	3.8	4.1
$\varepsilon 2$ ($^{o}/_{oo}$)	2.0								
$\varepsilon cu2$ ($^{o}/_{oo}$)	3.5								

Table 1.2: Grade of reinforcement

Grade	S260	S300	S360	S400	S460	S500	S560	S600
Fyk	260	300	360	400	460	500	560	600

Table 1.3: Size of Reinforcement

Size of bar (mm)	6	8	10	12	14	16	18	20	22	24	26	30	32	34	40
Area (mm2)	28.3	50.3	78.5	113	154	201	254	314	380	452.4	531	707	804.2	908	1257

2. LITERATURE REVIEW

2.1 Over view

Programming the computer is the art of producing code in a standard computer language to control the computer for calculations, data processing, or controlling a process using programming languags such as FORTRAN, BASIC, Java and MATLAB (Urroz, 2002).

MALTAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by Math Works ®, MATLAB allows matrix manipulations, plotting of functions and data implementation of algorithms, creation of user interfaces and interfacing with programs written in other languages, including C, C++, C#, Java, FORTRAN and Python. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems. In 2004, MATLAB had around one million users across industry and academia. MATLAB is used in mathematics and computer science. Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s. He designed it to give his students' access to LINPACK and EISPACK without them having to learn FORTRAN. It soon spread to other universities and found a strong audience within the applied mathematics community. Jack little, an engineer, was exposed to it during a visit Moler presentation made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded Math Works in 1984 to continue its development. These rewritten libraries were known as JACKPAC. In 2000, MATLAB was rewritten to use a newer set of libraries for matrix manipulation, LAPACK (wikipedia, 2017).

2.2 Benefit of Using MATLAB

The selection of MATLAB programming for this thesis is for the fact that the MATLAB enviroment is user-friendly because of this the objective of program is to spend time in learning the physical and mathematical principles of a problem but not to spend too many times learning about the programming language. The term friendly is used in the following sense that MATLAB software executes one instruction at a time. By analyzing the partial results and based on these results, new instructions can be executed that interact with the existing information already stored in the computer memory, without the formal compiling required by other competing high-level computer languages (Kalechman, 2009).

The effective and easy to use computing environment of MATLAB along with availability of a large number of helpful MATLAB built-in functions has rendered it the popular tool of choice for many educators in various engineering fields. Using the MATLAB interactive environment, programs placed in script files can easily be created and edited to perform the desired computations and to generate the needed output (Das & Navaee, 2002).

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows solving many technical computing problems, especially those with matrix and vector formulations, in a fraction of time it would take to write a program in a scalar non interactive programming language, such as C or FORTRAN (Moler, 1996).

MATLAB has several applications including:

- Math and computation
- Algorithm development
- Modeling, simulation and prototyping
- Data analysis, exploration and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

2.3 Available Computer Oriented Program on Analysis and Design of Structural Members

Different researcher's worldwide showed their effort for analysis and design of structural members on MATLAB. Some of these works are:

MATLAB design modules that are incorporated for reinforced concrete design based on the ACI code were developed and implemented. The main objective of the modules is to improve students' computational skills for solving routine design calculations without compromising the theoretical understanding of the subject matter (Urgessa, 2011).

Finite element formulation for eight noded brick element performance on beam element was checked for non-flexure and flexure loading conditions. MATLAB code were developed to find the deflection at the nodal points using eight noded brick element. It was found that in case of flexural model on beam member, the standard linear eight noded brick element has given poor results due to the missing of higher order shape function. Therefore, the above eight noded brick element was modified further in this work using the incompatible element. The results obtained from MATLAB using incompatible element then compared with STAAD Pro and ANSYS software (Dewangan & Mandal, 2017).

The analysis of simple supported beam subjected to only point load or only uniformly distributed load or mix load (both point load and uniformly distributed load) were done using MATLAB programming language. The shear force and bending moment were obtained by interactining with the developed graphical user interface (Venkatesh & Vivek, 2015).

Seismic analysis done using equivalent lateral force method based on IS 1893:2002 after analysis of a structure in STAAD-PRO software was completed, the maximum shear force and bending moment in the column, beam and slab were noted. Using the output from analysis MATLAB were programmed to give the required area of reinforcement, size for footing, column, beam and slab were properly designed (Harle, 2017).

A computer program were developed for analysis of continuous beam considering various types of member loads with MATLAB software based on the matrix stiffness method of structural analysis. Joint displacements, member end forces and support reactions are some of the output from the analysis of the program. Analysis of beam structure also checked with SAP 2000 software. Then, the results of analysis with MATLAB software were compared with the results obtained from SAP2000 software (Htwe & Khaing, 2014).

Beam stresses and deflections should be within the material allowable limits and therefore analysis of beam design is essential. For simply supported and cantilever beam subjected to point load and uniformly distributed load a quick and efficient way of analyzing beam design were developed with MATLAB code which can be used to solve beam design problems faster and errorless (Gajendra, et al., 2014).

A program were developed for design of reinforced concrete elements using MATLAB Programming language. The program is based on IS 456:2000 which helps to design beam, slab and column by taking analysis output from structural software (Bama & Priya, 2014).

A computer program were developed in MATLAB for analysis of beam based on finite element method. This program used to determine beam displacements or response of structural systems to both static and dynamic loading conditions (Saglar, 2009).

The analysis and design of flat slab were done using MATLAB Programming language for easy application of design of flat slab by reviewing design methods based on IS: 456-2000, NZ-3101-2006, Eurocode2-2004 and ACI-318-2008 design codes, After getting all the results of various codes with MATLAB such as bending moment, thickness of slab and quantity of reinforcement. The result obtained from each code compared eachother to select best way of practicing analysis and design of flat slab (Dahake, et al., 2016).

The nonlinear behavior of building structure subjected to seismic loading, considering torsional effect IS:1893 (Part 1)-2002 used dynamic analysis. Time history analysis carried out and response quantities viz. natural time period, base shear, torsional moment, displacement were obtained using MATLAB programming language (Awari, et al., 2016).

2.4 Computer Programs for Optimisation of Reinforced Concrete Beam

The optimisation of reinforced cement concrete doubly reinforced beam subjected to imposed load had been done. In this work the principle design objective is to minimise the total cost of beam after full filling all the requirements according to IS 456:2000 (Singh & Rai, 2014).

Genetic algorithms (GA) for the optimum cost design of reinforced concrete continuous beams based on the standard specifications of the American Concrete Institute was used . The produced optimum design satisfies the strength, serviceability, ductility, durability, and other constraints related to good design and detailing practice (Najem & Yousif , 2013)

To demonstrates application of the genetic algorithm to the design of reinforced concrete continuous beam. Genetic algorithm was used to find out the depth and width of the beam, the number and diameter of bars and the diameter and spacing of stirrups. A program is created based on genetic algorithm to carry out the design in MATLAB (Alex & Kottalil , 2015)

Optimum design of simply supported doubly reinforced beams with uniformly distributed and concentrated load had been done by incorporating actual self weight of beam, parabolic stress block, moment-equilibrium and serviceability constraints besides other constraints. The optimization techniques in general enable designers to find the best design for the structure under consideration. In this particular case, the principal design objective is to minimize the total cost of a structure (Bhalchandra & Adsul , 2012).

The application of Genetic Algorithms for the design of continuous reinforced concrete T and L beams was based on the recommendations of the current Australian design standards. This optimisation problem was implemented by writing a computer program using MATLAB. This computer program is composed of two modules, one for structural analysis and design of the beams and the second for calculating their optimum solution using the genetic algorithm code (Hadi, 2001).

Even though the above programs are used for analysis and design of structural members or elements including beam, they are not applicable for design purpose in our country because they are not based on the current Ethiopian building codes of standard. Therefore the aim of this research is to develop MATLAB program that integrate simple and friendly graphical user interface which minimize the effort that needed for design of reinforced concrete beam sections, and also that optimize simply supported rectangular double reinforced concrete beam section.

3. MATERIALS AND METHODS

3.1 Materials

3.1.1 MATLAB

MATLAB is an interactive system for matrix-based computation and a high performance language for technical computing designed for scientific and engineering uses. It integrates computation, visualization and programming for an easy way use environment where problems and solutions are expressed in familiar mathematical notations (Houcque, 2005)

In recent years with significant improvements in computer science especially in the area of software application programming for human-computer interaction new display and interface design techniques were developed resulting in graphical user interface programs. MATLAB is one of the computer program which used to develop graphical user interface that enable the user to interact with the computer through graphical icons, toolbars, buttons and other user-friendly tools. Enhancing the structural analysis software with a graphical user interface allows engineers to conveniently set up finite element models in a graphical workspace and facilitates the process of selecting the analysis options thus reducing the possibility of errors and saving time.

3.1.2 MATLAB Graphical User Interface

MATLAB is well known for its numerical problem solving power. One way of solving numerical problem is using MATLAB graphical user interface. A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components, that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a graphical user inteface need not understand the details of how the tasks are performed.

Traditionally programs written by engineers have very simple interfaces, and often only the author is the one who uses the program once it is completed. There are occasions where a more sophisticated user interface, specifically a graphical user interface (GUI) is desired. Some of these are:

- Wishing to have a nontechnical, person use your programs to perform some ongoing data analysis task, etc.
- Desire to share your tool program with other members of your work group, but want the interface to be friendly
- Hope of writing a utility function for your own use and simplify tasks.
- Third-party developer of tools for the MATLAB user community.

3.1.2.1 Buttons Used for Devlopment of Graphical User Interface (GUI)

In MATLAB graphical user interface (GUI) is developed using MATLAB GUIDE. GUIDE is an abbreviation for MATLAB Graphical User Interface Development Environment which provides a set of tools for creating graphical user interfaces (GUIs). These tools greatly simplify the process of laying out and programming GUIs. Basic GUI components used in these research are:

- Push button
- Radio button
- UI editable text
- UI static text
- Pop-up menu
- Axes

3.1.3 Design Code Selection

The analysis and design of beam, like other civil engineering structures or structural members, is dependent on certain design codes. Design of beam require a set of rigorous design specifications to ensure the safety and overall quality of the overall structures.

Design and construction of concrete structures in Ethiopia was first referenced to Ethiopian building codes of standard (EBCS-2, 1995) since it is national concrete design code throughout the country. But in recent time in Ethiopia, the code is renewed structural engineers are strongly recommended to have a good reference of the most recent version of these code ES EN 1992-1-1:2015 while designing and constructing bulding structure in the country.

3.2 Methods

3.2.1 Optimization Technique

The genetic algorithm (GA) is a heuristic search technique based on the mechanics of natural selection developed by John Holland. Koza provides a good definition of a GA: The genetic algorithm is a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects (typically fixed-length character strings patterned after chromosome strings), each with an associated fitness value, into a new population (i.e. the next generation) using operations patterned after the Darwinian principle of reproduction and survival of the fittest and after naturally occurring genetic operations (Singh & Rai, 2014).

- Genetic algorithms use a population of points at a time in contrast to the single-point approach by the traditional optimization methods. That means, at a given time, Genetic algorithms process a number of designs.
- Genetic algorithms do not require problem-specific knowledge to carry out a search. For instance, calculus-based search algorithms use derivative information to carry out a search. In contrast to this, Genetic algorithm are in different to problem-specific information.
- Genetic algorithms work on coded design variables, which are finite length strings. These strings represent artificial chromosomes. Every character in the string is an artificial gene. Genetic algorithms process successive populations of these artificial chromosomes in successive generations.
- Genetic algorithms use randomized operators in place of the usual deterministic ones.

3.2.2 Steps for Development of Program using MATLAB for Reinforced Concrete Beam Sections

First the graphical user interface window created which needed for analysis and designing of reinforced beam sections. Next, as soon as the figure file is saved automatically an editor is displayed where the scripts program is typed.

In the second step a program script that accept input arguments and return output arguments in the graphical user interface window is created using editor. Once the code or function is typed the file is saved as M-file function or M-file scripts.

Third step for input and output variables in the M-file scripts of the graphical user interface value is assigned. The variables that are used in the calculations within the M-file must have assigned values for example, for reinforced concrete beam sections values that need to be assigned includes:

- Material properties
- Geometric features and different parameters of reinforced concrete beam section

When the developed graphical user interface window is run MATLAB automatically generates a graphical user interface window. Then the required commands are executed in the user interface to obtain all the necessary parameters of reinforced concrete beam section analysis and design output such as:

- Size of cross-section of beam
- Area of reinforcement
- Moment resistance
- Shear resistance
- Torsional resistance

Finally to advance the practice of this program, a graphical user interface application is created using MATLAB application compiler. The created application has one main user interface window which consisted of six other user interface. The main interface window for analysis and design of reinforced beam sections opened when double clicking on the created short cut of the application.The other six user interface opened when selecting the radio button for the respective analysis and design problem as shown in Figure 3.1 below. The user interface window shown in Figure 3.2 below is a sample graphical user interface window for rectangular reinforced concrete beam section. In this interface to assist in understanding of what are the variables of the static boxes, edit boxes and pushbutton are referring to, a side view of a typical beam cross-section with description of the variables is integrated in the user interface window. The determination of section flexural resistance and design can be done in the same window where the cross-sectional parameters and material properties are defined. This is done by following the steps stated in the graphical user interface window. From Figure 3.3 - 3.7 shown below is graphical user interface window for the remaining analysis and design problem of reinforced concrete beam sections shown on main interface window of Figure 3.1.

Figure 3.1: Main interface window

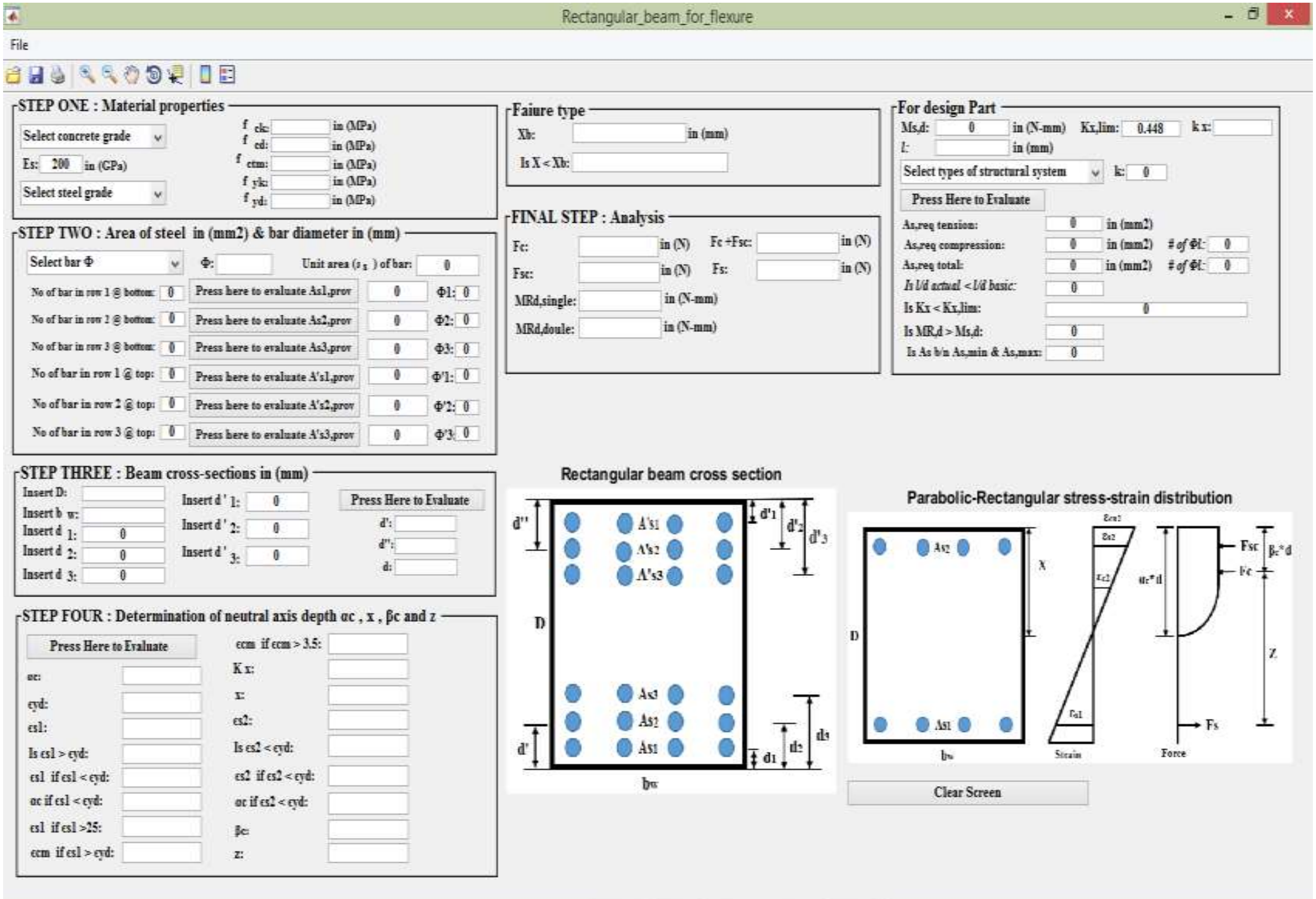


Figure 3.2: Graphical user interface window for rectangular beam

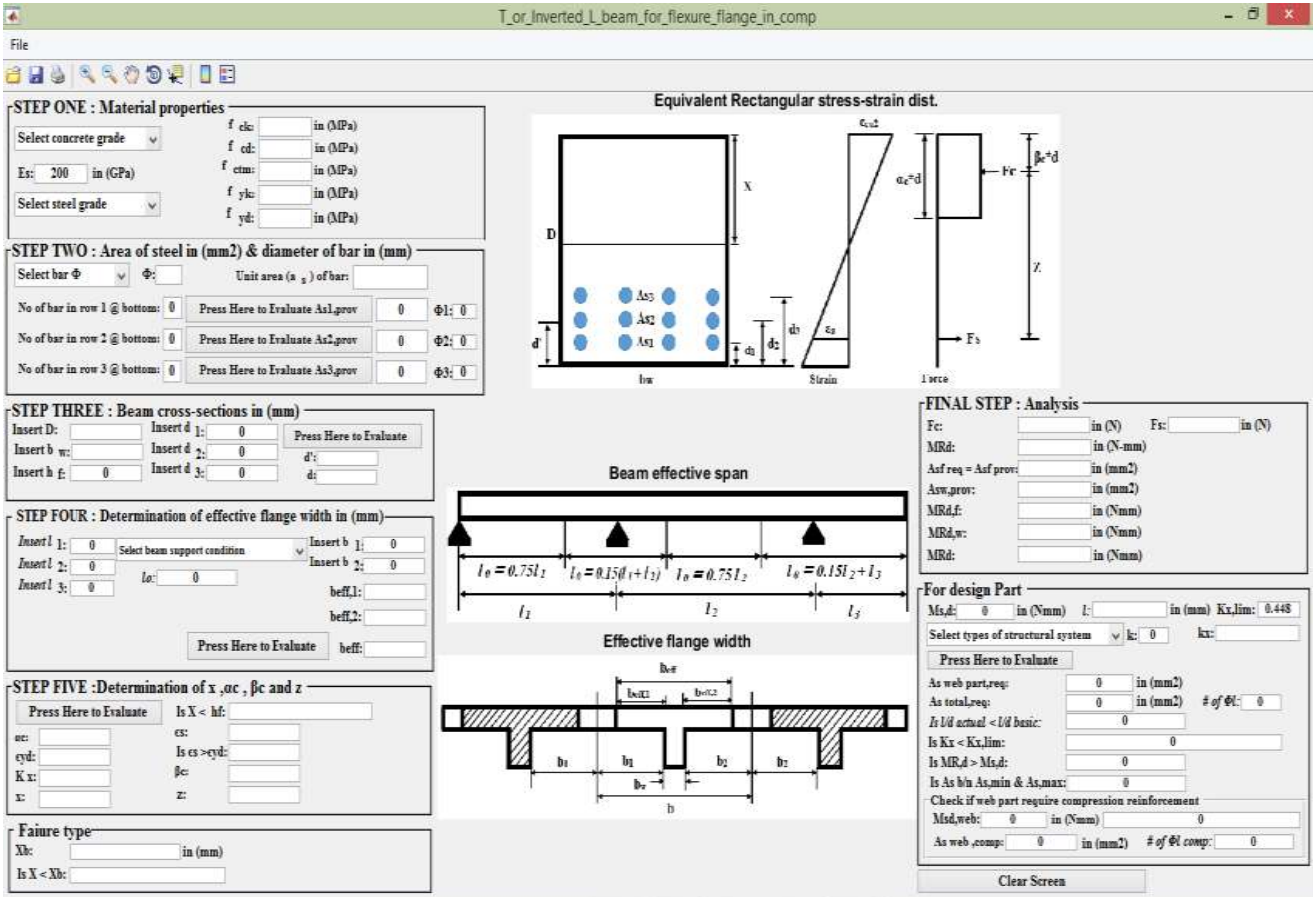


Figure 3.3: Graphical user interface window for T or inverted L beam

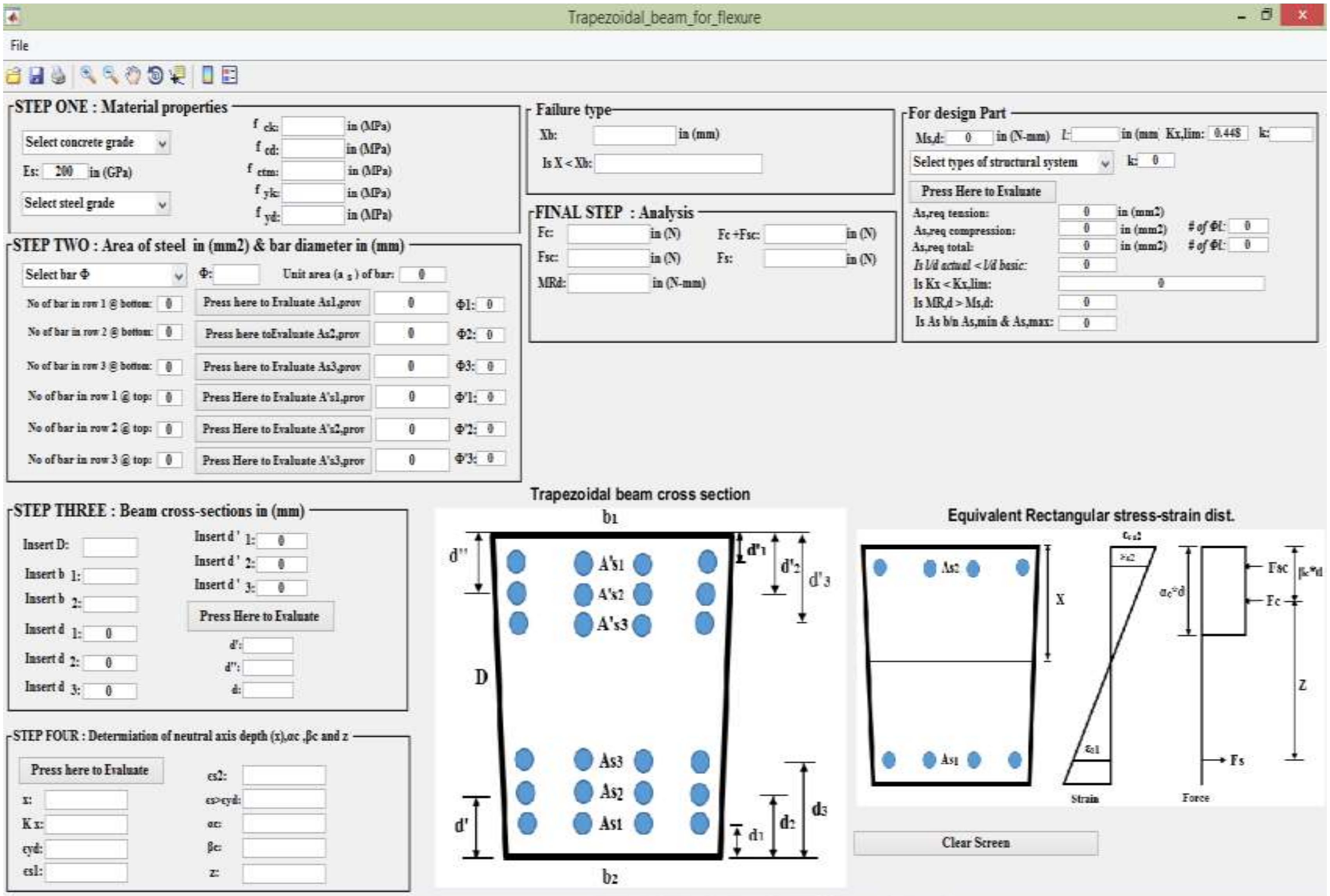


Figure 3.4: Graphical user interface window for trapezoidal beam

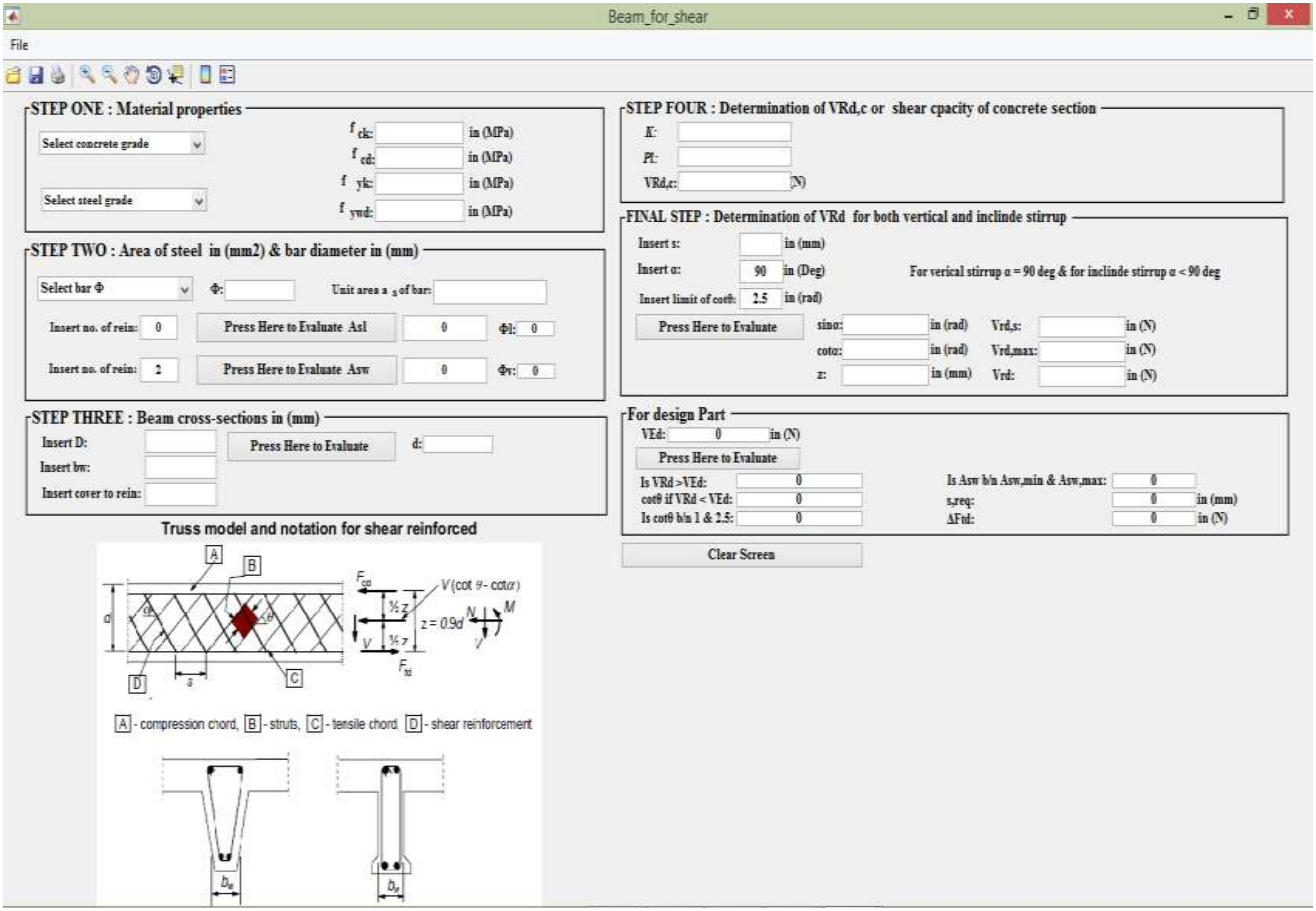


Figure 3.5: Graphical user interface window for shear

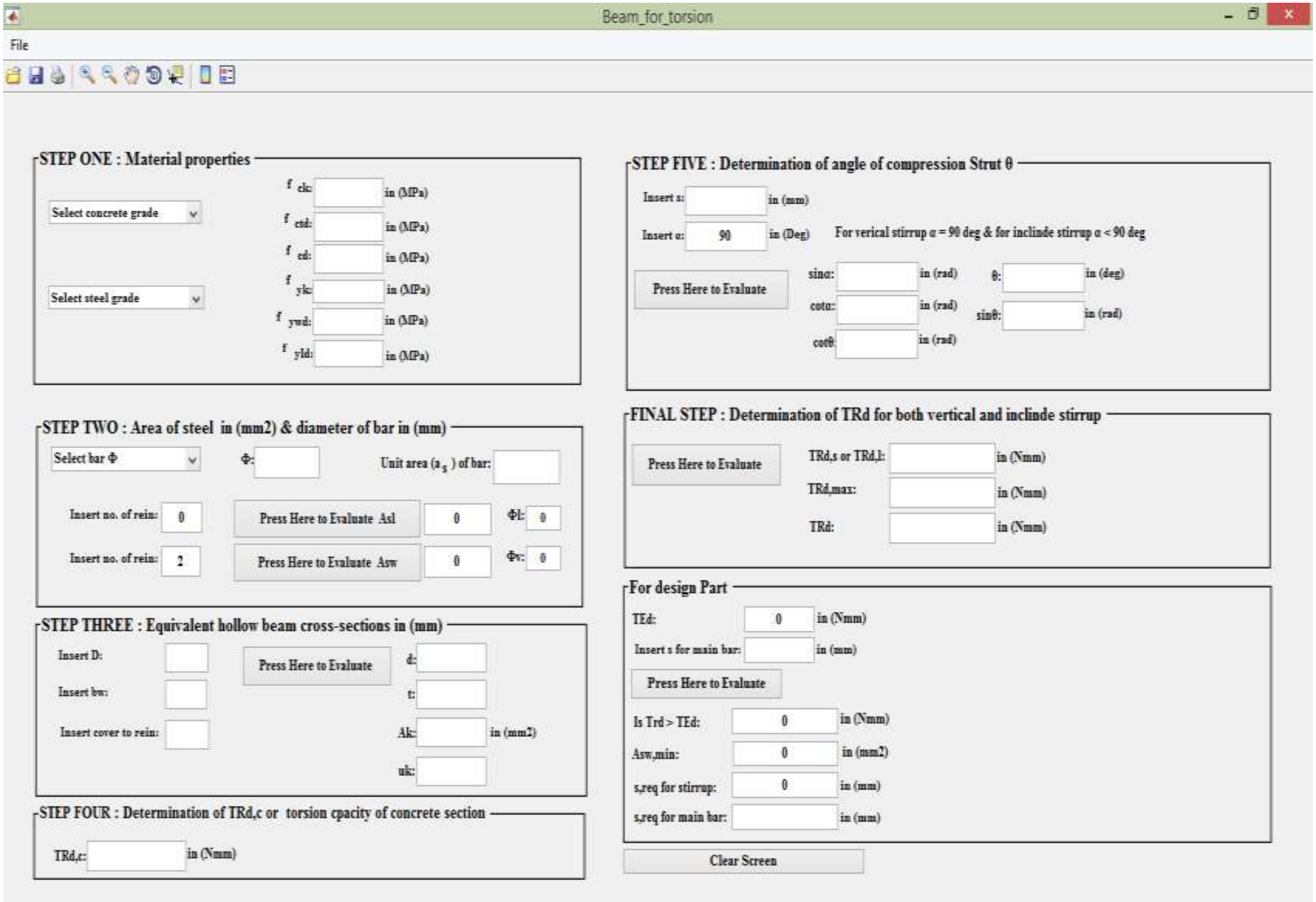


Figure 3.6: Graphical user interface window for torsion

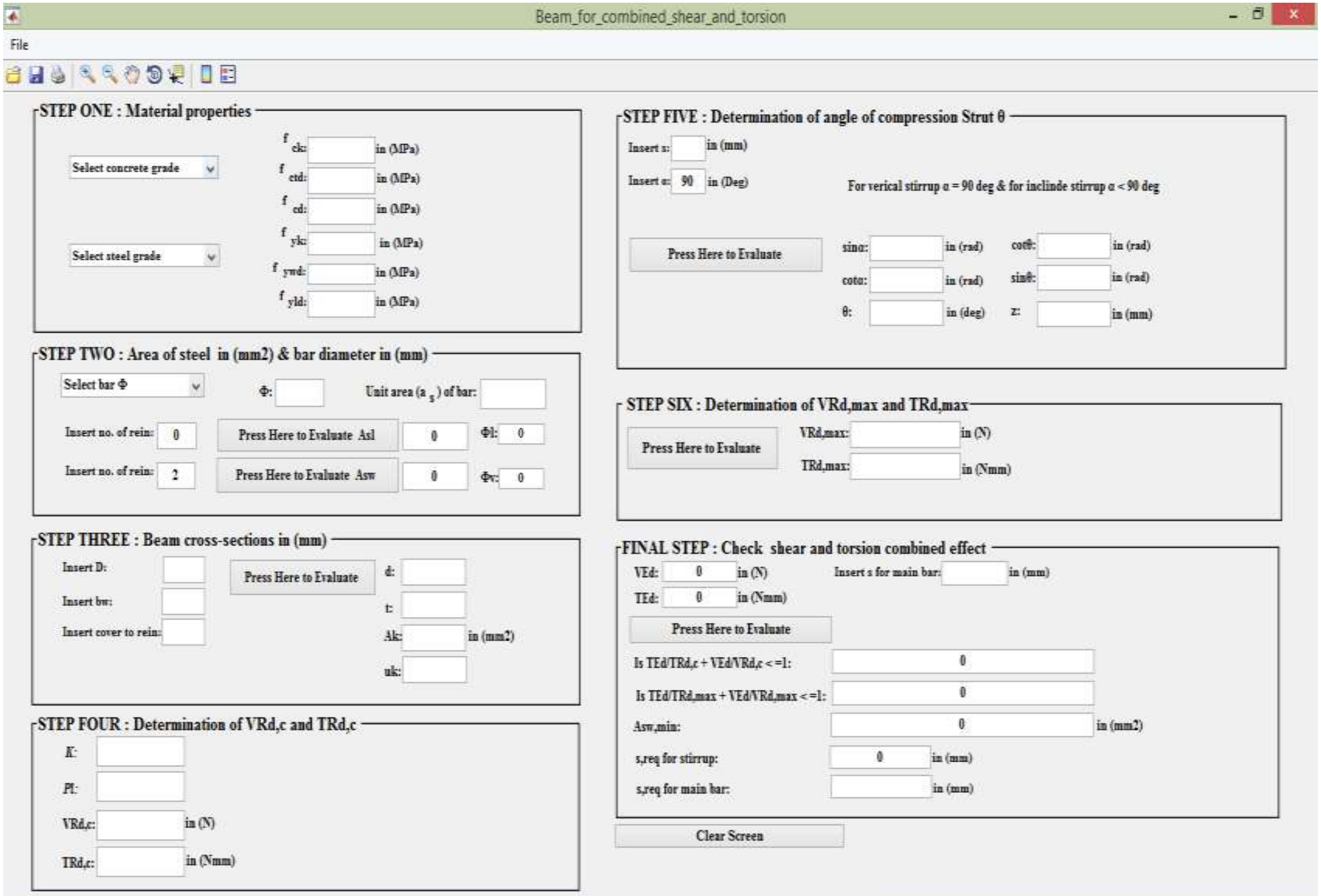
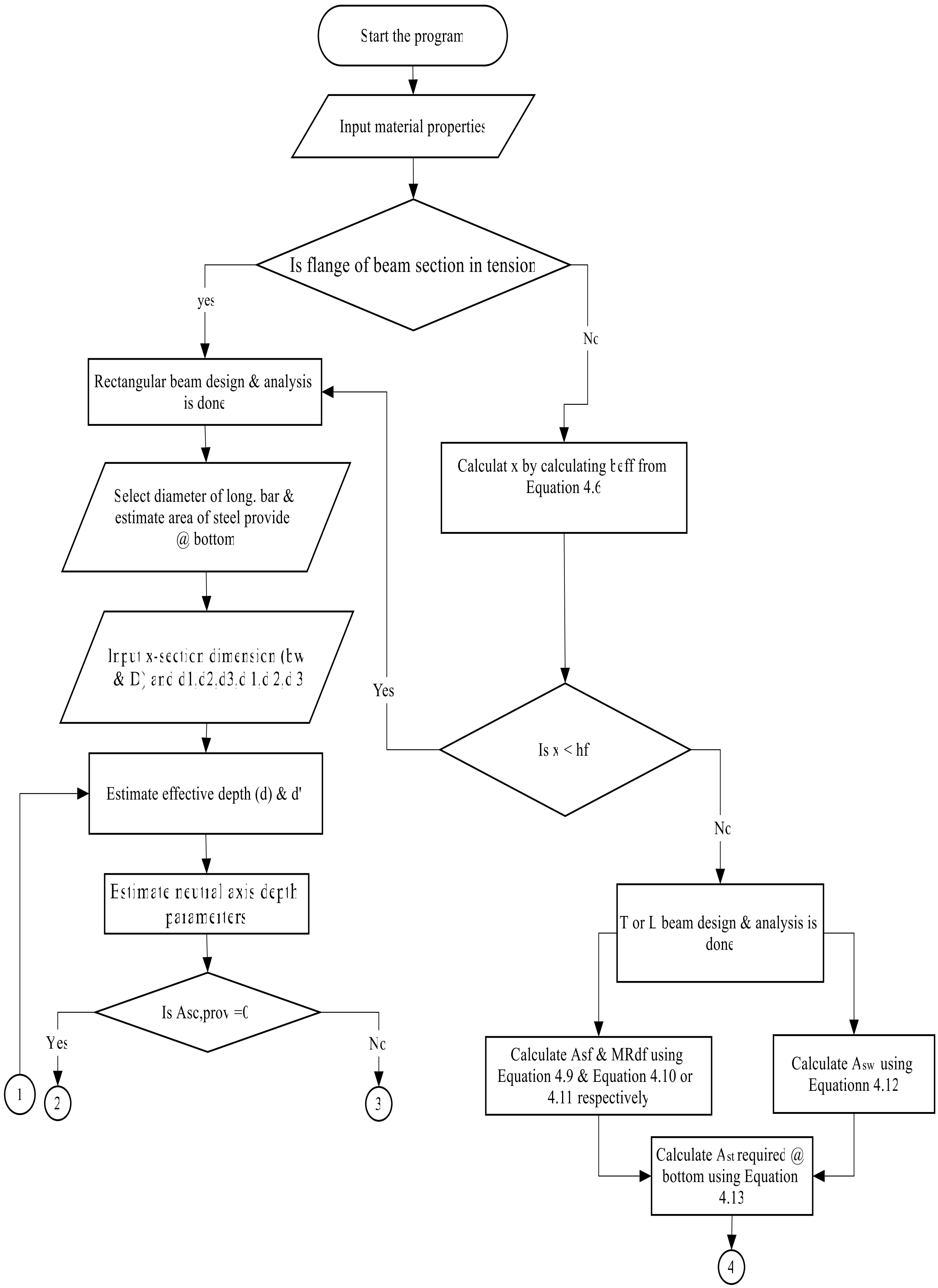


Figure 3.7: Graphical user interface window for combined effect of shear with torsion

3.2.3 Flow Chart for Analysis and Design of Reinforced Concrete Beam Sections

For a particular concrete grade, steel grade and cross-sectional dimensions the neutral axis located at x distance from the most compressed fiber of a cross-section (Figure 4.1), the corresponding strain distribution category can be known. The limiting neutral axis depth for a particular region can be described in terms of the ultimate strains in steel and concrete. For different values of beam cross-sections, steel grade, concrete grade and neutral axis depth parameters the moment resistance and area of steel required to resisist bending is evaluated. A general representation of this process is shown on Figure 3.8 for the case of rectangular, T or inverted L and trapezoidal reinforced concrete beam for flexure. Similarly graphical user interface shown on Figure 3.9, Figure 3.10 and Figure 3.11 below is adopted for evaluating parameters required for shear resistance, torsion resistance, to check combined effect of shear with torsion respectively. Inaddition, it uses for selecting beam cross-sections and evaluating area of steel provide for rectangular reinforced concrete beam with vertical or inclined stirrup respectively. A general representation of this process is shown from Figure 3.9 to Figure 3.11 for the case of rectangular beam with vertical and inclined stirrup for shear, torsion and combined effect of shear with torsion.


Start the program
Input material properties
Is flange of beam section in tension
yes
No
Rectangular beam design & analysis is done
Select diameter of long. bar & estimate area of steel provide @ bottom
Input x-section dimension (bw & D) and d1,d2,d3,d 1,d 2,d 3
Estimate effective depth (d) & d'
Estimate neutral axis depth parameters
Is Asc,prov =0
Yes
No
1
2
3
Calculat x by calculating beff from Equation 4.6
Yes
Is x < hf
No
T or L beam design & analysis is done
Calculate Asf & MRdf using Equation 4.9 & Equation 4.10 or 4.11 respectively
Calculate Asw using Equationn 4.12
Calculate Ast required @ bottom using Equation 4.13
4

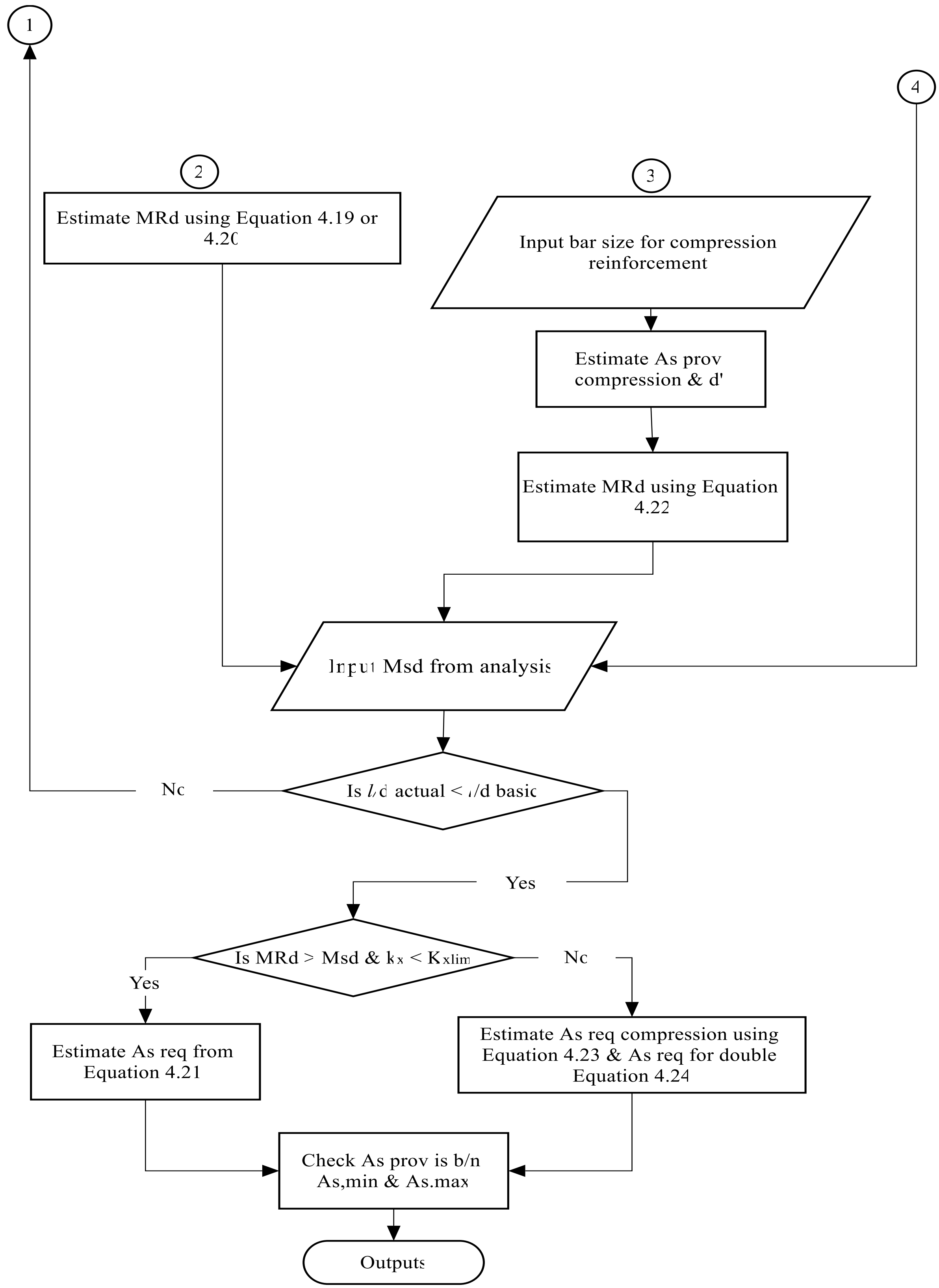


Figure 3.8: Flow chart for design of beam and estimation of flexural resistance

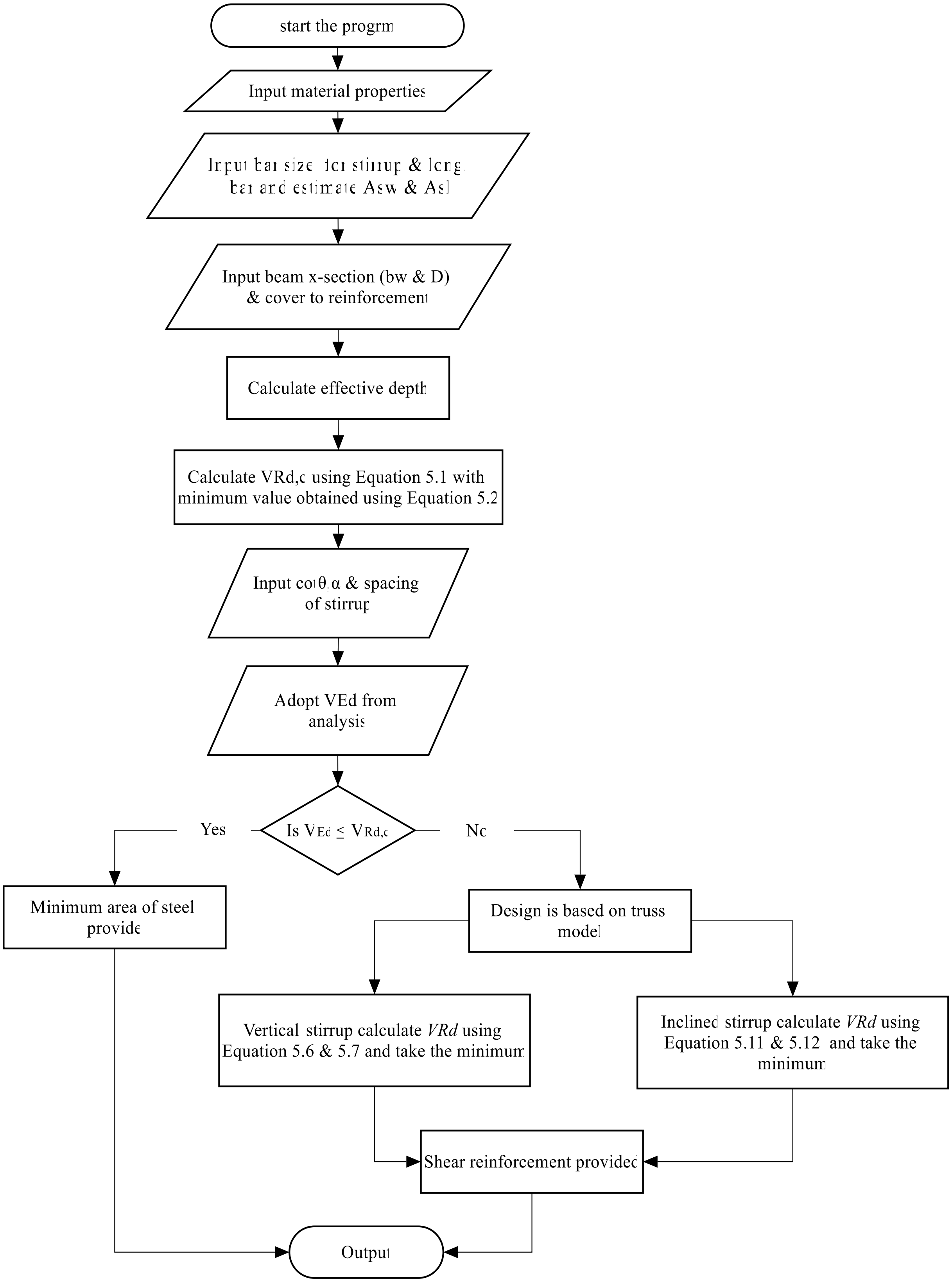


Figure 3.9: Flow chart for design of beam and estimation of shear resistance

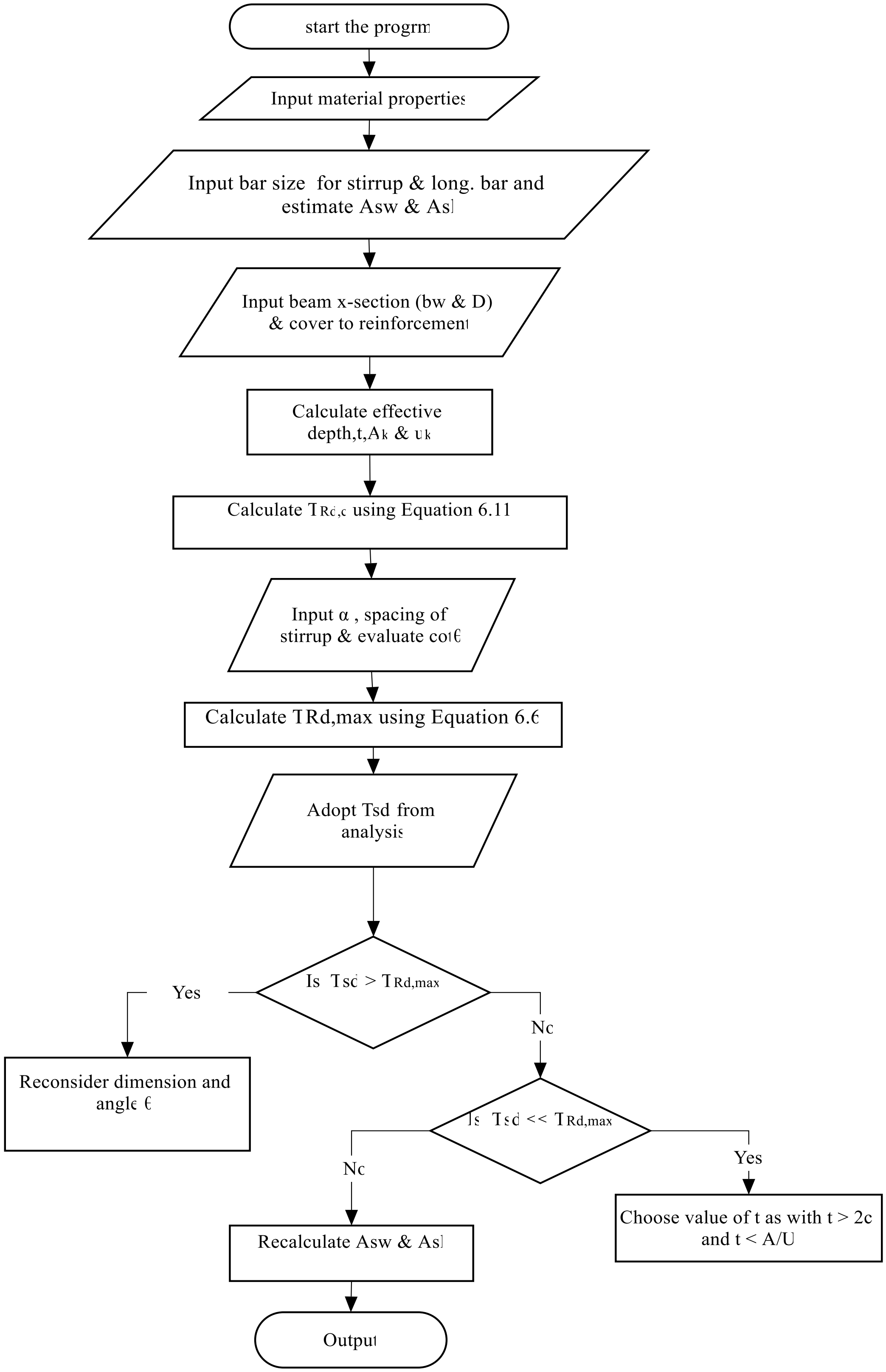


Figure 3.10: Flow chart for design of beam and estimation of torsion

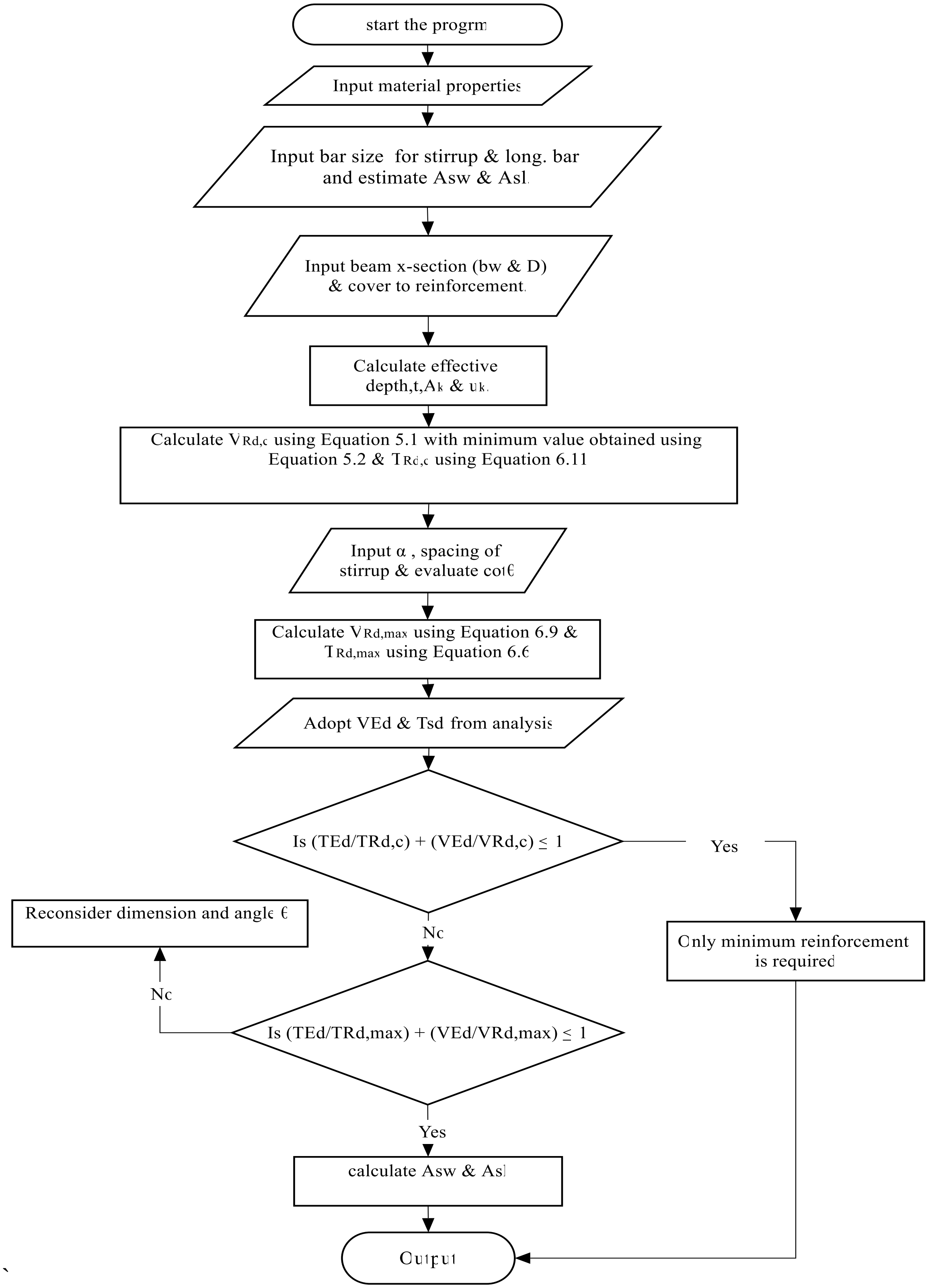


Figure 3.11: Flow chart for design and analysis of reinforced concrete beam section for combined effect of shear with torsion

3.2.4 Flow Chart for Design Optimization of Double Reinforced Concrete Rectangular Beam Sections

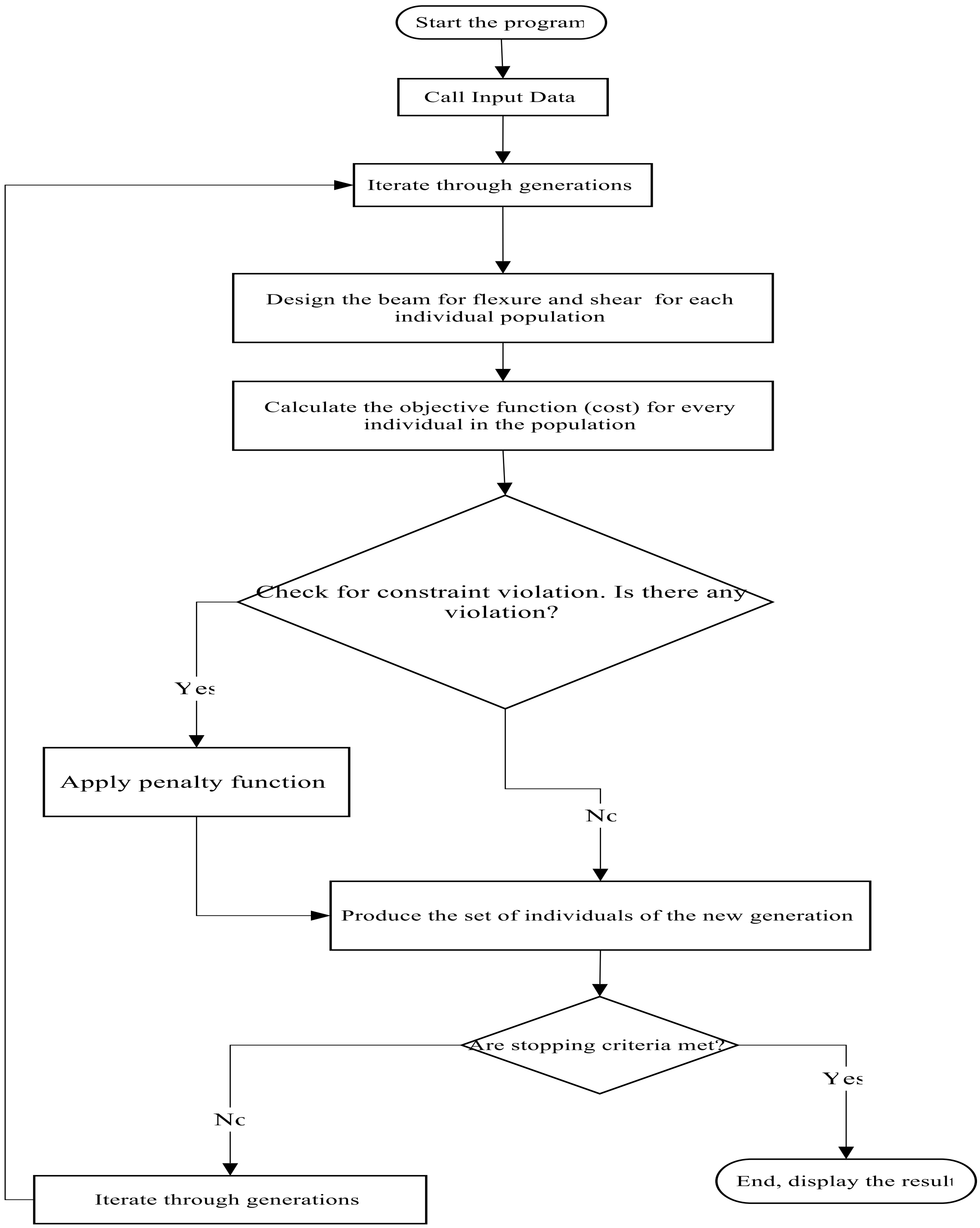


Figure 3.12: Flowchart for the developed optimization program

4. ANALYSIS AND DESIGN OF REINFORCED CONCRETE BEAM SECTIONS FOR FLEXURE

4.1 Assumptions and Criterias in Analysis and Design

The analysis and design of reinforced concrete beam sections are carried out on the bases of new Ethiopian Building Code Standard.

Unless specified exclusively the material properties and factor of safety for concrete and reinforcement are taken according to (EN, 1992-1-1:2015).

- The modulus of elasticity of the reinforcement to be taken equal to the value of 200Gpa for this study.
- The partial safety factors of materials used are: α_{cc}=0.85 is factor to account for long term effects on the compressive strength and of unfavorable effects resulting from the way the load is applied.
- γc = 1.50 is the partial safety factor for the concrete.
- γs = 1.15 is the partial safety factor for the reinforcement.

Therefore, the design values of material strength can be determined by dividing their Characteristics strength by the corresponding safety factors.

4.1.1 Linear Elastic Analysis

Analysis of elements based on theory of elasticity may be used for both the serviceability and ultimate limit states. For the determination of the action effects linear analysis may be carried out assuming

- Uncracked cross sections
- Linear stress-strain relationships
- Mean value of the modulus of elasticity

This section applies to undisturbed regions of beams, slabs and similar types of members for which sections remain approximately plane before and after loading. The basic assumption made when deriving the expression for flexural strength of reinforced concrete sections are

- Sections perpendicular to the axis of bending that are plane before bending remains plane after bending.
- The strain in the reinforcement is equal to the strain in the concrete at the same level.
- The stress in the concrete and reinforcement can be computed from the strains by using stress-strain curves for concrete and steel
- The tensile strength of the concrete is ignored.

The two main requirements that are satisfied during the analysis and design of reinforced concrete beams are:

1. Stress and strain compatibility, which ensures that the stress at any point in a member must correspond to the strain at that point.
2. Equilibrium, which ensures the internal forces must balance the external load effects.

4.1.2 Distribution of Strains and Stresses across a Section in Bending

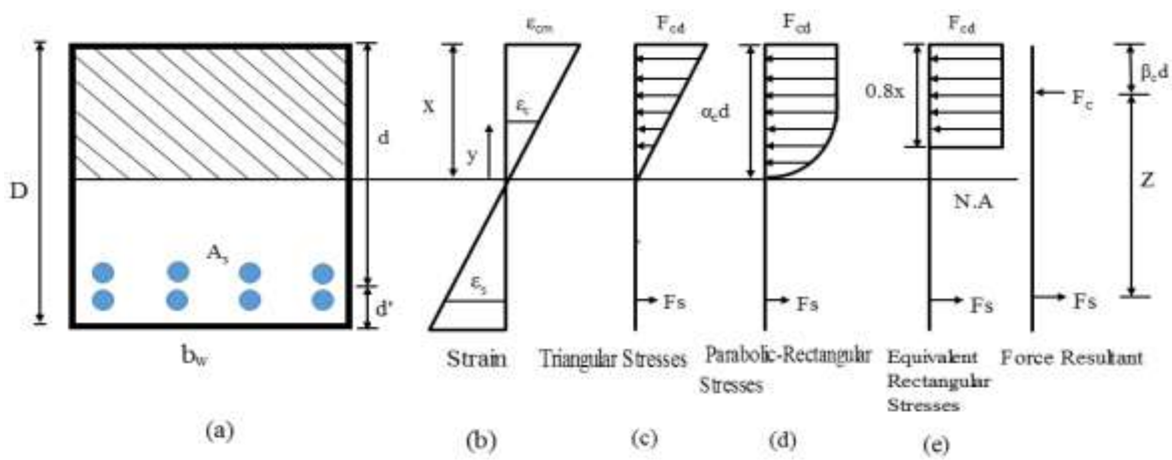


Figure 4.1: Stress strain distribution of singly reinforced beam section

The theory of bending for reinforced concrete assumes that the concrete will crack in the regions of tensile strains and that, after cracking, all the tension is carried by the reinforcement. It is also assumed that plane sections of a structural member remain plane after straining, so that across the section there must be a linear distribution of strains.

Figure 4.1 shows the cross-section of a member subjected to bending, and the resultant strain diagram, together with three different type of strain distribution in the concrete:

1. The triangular stress distribution applies when the stress is very nearly proportional to the strains, which generally occurs at the loading levels encountered under working conditions and is ,therefore, used at the serviceability limit state.
2. The parabolic-rectangular stress block represents the distribution at failure when the compressive strains are within the plastic range, and it is used in the ultimate limit state.
3. The equivalent rectangular stress block is a simplified alternative to the parabolic rectangular distribution. And the above equivalent rectangular stress block applies for rectangular, T and trapezoidal sections as shown on Figure 4.2.

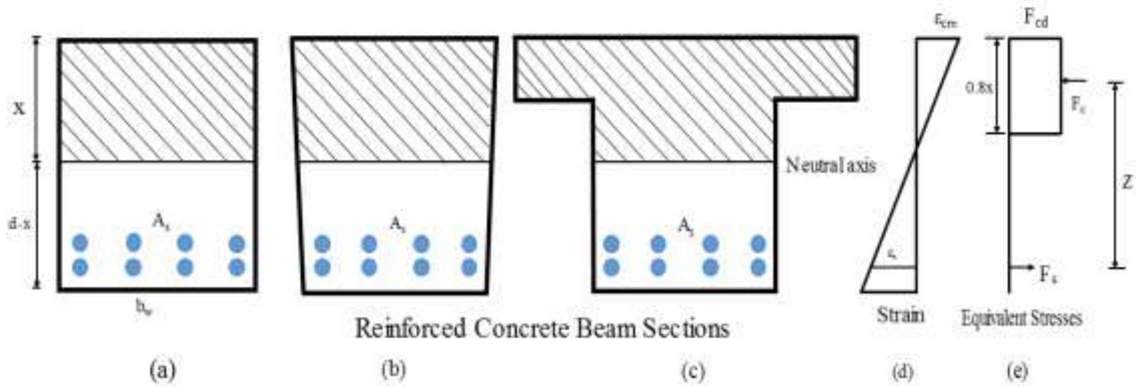


Figure 4.2: Applicability of equivalent rectangular stress block to Rectangular (a), Trapezoidal (b) and T- section (c).

4.2 Types of Flexure Failure

There are three types of flexural failures of reinforced concrete section: tension, compression and balanced failures. Knowing the types of failures may help us to choose the desirable type of failure from the three, in case failure is imminent.

1. Tension failure: If the steel content, A_s, of the section is small, the steel will yield and reach f_{yd} before the concert reaches its maximum strain, ε_{cu}, and the depth of neutral axis decreases.
2. Compression failure: If the steel content, A_s, is large, the concrete may reach its capacity before steel yields and in this case the neutral axis depth increases.
3. Balanced failure: At balanced failure the steel reaches f_{yd} and the concrete reaches a strain of ε_{cu} , simultaneously.

4.3 Design Consideration for Flexure

4.3.1 Deflection Control

1. The appearance and general utility of the structure could be impaired when the calculated sag of a beam subjected to quasi-permanent loads exceeds span/250. The sag is assessed relative to the supports. Pre-camber may be used to compensate for some or all of the deflection but any upward deflection incorporated in the formwork should not generally exceed span/250.
2. Deflections that could damage adjacent parts of the structure should be limited. For the deflection after construction, span/500 is normally an appropriate limit for quasi-permanent loads. Other limits may be considered, depending on the sensitivity of adjacent parts.

Provided that reinforced concrete beams in buildings are dimensioned so that they comply with the limits of span to depth ratio given in the code, their deflections may be considered as not exceeding the limits set out in 4.3.1(1) and 4.3.1(2) above. The limiting span/depth ratio may be estimated using expressions (4.1) or (4.2) stated below and multiplying this by correction factors to allow for the type of reinforcement used and other variables. No allowance has been made for any pre-camber in the derivation of these expressions.

$$\frac{l}{d} = k\left[11 + 1.5\sqrt{f_{ck}}\,\frac{\rho_o}{\rho} + 3.2\sqrt{f_{ck}}\left(\frac{\rho_o}{\rho} - 1\right)^{3/2}\right] \text{if } \rho \leq \rho_o \qquad (4.1)$$

$$\frac{l}{d} = k\left[11 + 1.5\sqrt{f_{ck}}\,\frac{\rho_o}{\rho - \rho'} + \frac{1}{12}\sqrt{f_{ck}}\,\frac{\rho'}{\rho_o}\right] \text{if } \rho > \rho_o \qquad (4.2)$$

Where:

$\frac{l}{d}$ is the limit span/depth

k is the factor to take into account the different structural systems and it is 1 for simply supported beam,1.3 for end span of continuous beam, 1.5 for interior span of beam and 0.4 for cantliver beam.

ρ_0 is the reference reinforcement ratio = $10^{-3}\left(\sqrt{f_{ck}}\right)$ (4.3)

ρ is the required tension reinforcement ratio at mid-span to resist the moment due to the design loads (at support for cantilevers)

$\rho^{'}$ is the required compression reinforcement ratio at mid-span to resist the moment due to design loads (at support for cantilevers)

f_{ck} is the characterstic compressive strength in MPa.

The expressions for $\frac{l}{d}$ have been derived on the assumption that the steel stress, under the appropriate design load at serviceable limit state at a cracked section at the mid-span of a beam or slab or at the support of a cantilever is 310MPa,(corresponding roughly to f_{yk}= 500MPa).

Where other stress levels are used, the values obtained using the expression (4.1) or (4.2) should be multiplied by 310/σ_s. It will normally be conservative to assume that:

$$310/\sigma_s = 500/\left(f_{yk} A_{s,req}/A_{s,prov}\right) \quad (4.4)$$

Where:

σ_s is the tensile steel stress at mid-span (at support for cantilevers) under the design load at servicablity limit state (SLS)

$A_{s,prov}$ is the area of steel provided at this section

$A_{s,req}$ is the area of steel required at this section for ultimate limit state

For flanged sections where the ratio of the flange breadth to the rib breadth exceeds 3, the values of $\frac{l}{d}$ given by above expression (4.1) or (4.2) should be multiplied by 0.8.

For beams and slabs other than flat slab, with span exceeding 7m,which support partitions liable to be damaged by excessive deflections, the value of $\frac{l}{d}$ given by above expressions (4.1) or (4.2), should be multiplied by 7/l_{eff} .The effective span, l_{eff}, of a member should be calculated using expresson (4.5) based on different support condition shown on Figure 4.3 below.

$$l_{eff} = l_n + a_1 + a_2 \qquad (4.5)$$

where:

l_n is the clear distance between the faces of the supports;

values for a_1 and a_2 , at each end of the span, may be determined from the appropriate

a_i values in Figure 4.3 where t is the width of the supporting element as shown.

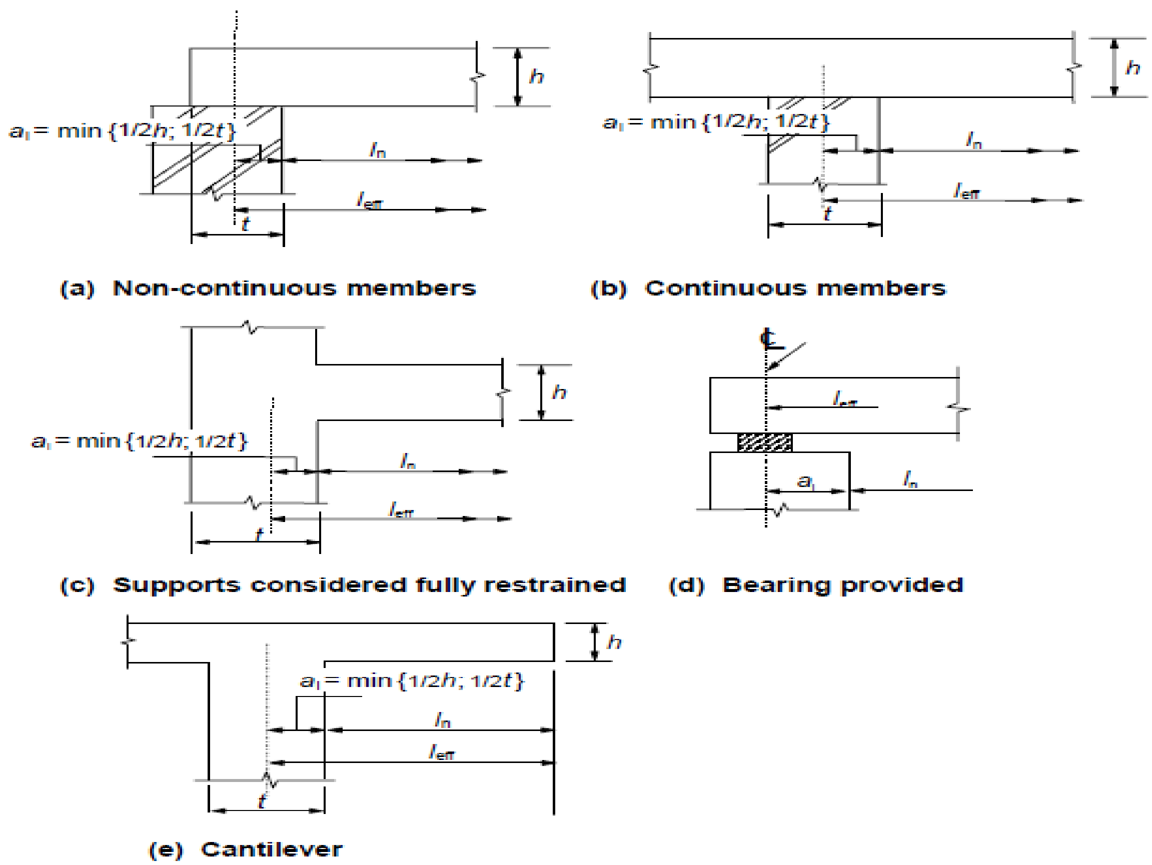


Figure 4.3: Effective span (l_{eff}) for different support conditions (EN, 1992-1-1:2015)

4.3.2 Analysis and Design of Flanged Reinforced Concrete Beam Section

4.3.2.1 Effective Flange Width Determination

Reinforced concrete floors or roofs are monolithic and hence, a part of the slab will act with the upper part of the beam to resist longitudinal compression. The resulting beam cross-section is, then, T-shaped (inverted L), rather than rectangular with the slab forming the beam flange where as part of the beam projecting below the slab forms the web or stem.

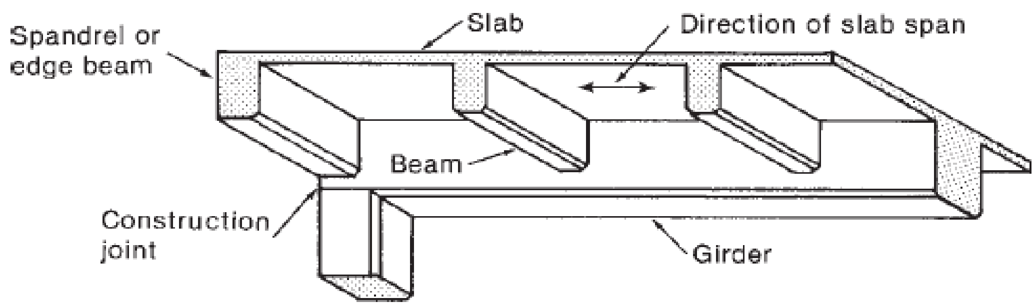


Figure 4.4: Slab, beam and girder floor system (Macgregor & Wight, 2012)

In T beams the effective flange width, over which uniform conditions of stress can be assumed, depends on the web and flange dimensions, the type of loading, the span, the support conditions and the transverse reinforcement. The effective width of flange should be based on the distance l_0 between points of zero moment, which may be obtained from Figure 4.5.

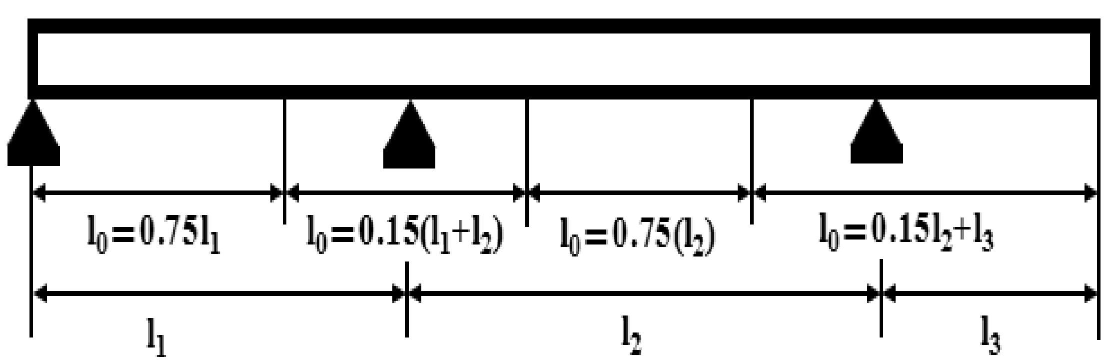


Figure 4.5: Definition of l_0, for calculation of effective flange width

Note: The length of the cantilever, l_3, should be less than half the adjacent span and the ratio of adjacent spans should lie between 2/3 and 1.5 as per ES EN 1992-1-1:2015 Section 5.3.2.1.

The effective flange width b_{eff} for a T beam or L may be derived as:

$$b_{eff} = \sum b_{eff,i} + b_w \leq b \tag{4.6}$$

Where

$$b_{eff,i} = 0.2b_i + 0.1l_o \leq 0.2l_o \tag{4.7}$$

$$b_{eff,i} \leq b_i \tag{4.8}$$

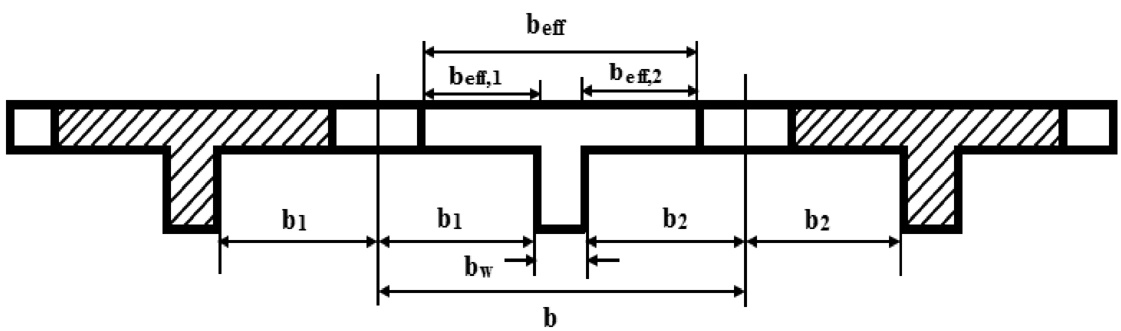


Figure 4.6: Effective flange width parameters

For a flanged beam with a negative moment, the compression zone will be the bottom rectangular part of the web, thus the procedures for design of rectangular sections will be appropriate.

If a flanged beam is subjected to positive moment, the neutral axis might remain within the flange of the beam or it might be in the web of the beam.

For the case where the neutral axis remains in the flange, the section may be treated as a rectangular section with width b as the width of the flange, and the procedures of analysis of rectangular sections can be adopted. However, if the neutral axis is in the web of the beam, a different approach for design is necessary and in doing so, adopting the rectangular stress block relationship for the concrete in compression will simplify the procedure.

Simplified equations for moment and force equilibrium obtained from stress-strain relationship of Figure 4.2 is obtained for flanged section by dividing the section into two parts: Beam flange part and beam web part to simplify the design process.

1. For beam flange part

$$A_{sf} = f_{cd}\left(b_e - b_w\right)h_f \big/ f_{yd} \qquad (4.9)$$

$$M_{Rd,f} = A_{sf}\, f_{yd}\left(d - {h_f}\big/{2}\right) \text{ or } M_{Rd,f} = f_{cd}\left(b_e - b_w\right)h_f\left(d - {h_f}\big/{2}\right) \qquad (4.10 \text{ or } 4.11)$$

2. For beam web part

$$A_{sw} = \frac{M_{Rd,w}}{f_{yd}\left(d - 0.4x\right)} \qquad (4.12)$$

Total tension reinforcement become

$$A_s = A_{sw} + A_{sf} \qquad (4.13)$$

4.3.3 Analysis and Design of Rectangular Reinforced Concrete Beam Section

Simplified equations for moment and force equilibrium obtained from stress-strain relationship of Figure 4.1 is based on the following assumptions:

1. The section is rectangular with width b and effective depth d
2. Stresses are in expressed in ‰ (x 10^{-3})
3. k_x limited to 0.448 for zero moment redistribution and calculated using expression (4.14)

$k_x = x/d$ (4.14)

4. ε_{cm} is the compression strain at the ultimate fiber in the compressed region of the section.
5. d is effective depth of the cross section defined as the distance from the center of the tensile reinforcement bars to the top most compressed fiber.
6. Assume the type of failure

Tension failure

- Rupture of steel occur at ε_s = 25‰ and $\varepsilon_s = f_{yd}/E_s$
- The strain in the steel exceeded the yield strain and the most compressed concrete has reached the crushing strain ε_{cm} = 3.5‰ , $\varepsilon_s > \varepsilon_{yd}$. then $f_s = f_{yd}$

Compression failure

- Assume ε_{cm} =3.5‰ and $\varepsilon_{yd} < \varepsilon_s$ then $f_s < f_{yd} = \varepsilon_s E_s$

Where:

E_s is modulus of elacticity of steel

From force equilibrium relations

The resultant force in the concrete determine using exprsession (4.15).

$$F_c = \alpha_c f_{cd} bd \tag{4.15}$$

The resultant force in the the compression and tension reinforcement determine using exprsession (4.16) and (4.17).

$$F_{sc} = A_{sc} f_{yd} \tag{4.16}$$

$$F_s = A_s f_{yd} \tag{4.17}$$

Then the total resultant force in the concrete can be determine from expression

$$F_{c,total} = F_c + F_{sc} \tag{4.18}$$

Where:

$$\alpha_c = \left(\frac{A_s - A_{sc}}{f_{cd} bd} \right) f_{yd}$$

$$f_{yd} = f_{yk}/\gamma_s$$

$$f_{cd} = (\alpha_{cc} f_{ck})/\gamma_c$$

From moment equilibrium

The resistance moment for rectangular single and double reinforced concrete beam section obtained using expression (4.19 or 4.20) and (4.22) respectvily.

$$M_{Rd} = \alpha_c f_{cd} bd^2 (1 - \beta_c) \text{ or } M_{Rd} = A_s f_{yd} d(1 - \beta_c) \tag{4.19 or 4.20}$$

A_s area of steel required at the tension zone for single reinforced beam section

$$A_s = {M_{sd}}/{d(1-\beta_c)} \tag{4.21}$$

$$M_{Rd} = \alpha_c f_{cd} b d^2 (1-\beta_c) + A_{sc} f_{s2} (d - d'') \tag{4.22}$$

A_{sc} or A_{s2} area of steel required at the compression zone for double reinforced beam section

$$A_{sc} = \left((M_{sd} - M_{Rd,single}) / f_{s2} (d - d'') \right) \tag{4.23}$$

$A_{s,total}$ total area of steel required at the tension zone for double reinforced concrete section

$$A_{s,total} = A_s + A_{sc} \tag{4.24}$$

Following the same design procedure as rectangule beam a trapezoidal reinforced concrete beam section can be analyized and designed using the equivalent rectangular stress-strain distribution as shown on Figure 4.2. Based on stress-strain compatiablity and equilibrium criteria the moment resistance (M_{Rd}) and area of steel required ($A_{s,req}$) can be determined for trapezoidal beam section.

Value of α_c and β_c obtain from expression below based on the assumed stress-strain distribution.

1. For parapolic-rectangular stress-strain relation

a) when $0 \leq \varepsilon_c \leq \varepsilon_{c2}$

$$\alpha_c = \varepsilon_{cm} \left[\frac{6 - \varepsilon_{cm}}{12} \right] k_x \tag{4.25}$$

$$\beta_c = k_x \left[\frac{8 - \varepsilon_{cm}}{4(6 - \varepsilon_{cm})} \right] \tag{4.26}$$

a) when $\varepsilon_{c2} \leq \varepsilon_c \leq \varepsilon_{cu2}$

$$\alpha_c = k_x \left(\frac{3\varepsilon_{cm} - 2}{3\varepsilon_{cm}} \right) \tag{4.27}$$

$$\beta_c = k_x \left[\frac{\varepsilon_{cm}(3\varepsilon_{cm} - 4) + 2}{2\varepsilon_{cm}(3\varepsilon_{cm} - 2)} \right] \tag{4.28}$$

2. For simplified rectangular stress block

$$\alpha_c = 0.8k_x \quad (4.29)$$

$$\beta_c = 0.8k_x \quad (4.30)$$

4.3.4 Reinforcement Provision for Flexure

1. The area of longitudinal tension reinforcement should not be taken as less than $A_{s,min}$.

$$A_{s,\min} = 0.26\frac{f_{ctm}}{f_{yk}}b_t d \geq 0.0013 b_t d \quad (4.31)$$

Where:

b_t denotes the mean width of the tension zone; for a T-beam with the flange in tension, only the width of the web is taken into account in calculating the value of b_t.

f_{ctm} should be determined with respect to the relevant strength class according to Table 3.1 of ES EN 1992-1-1:2015.

2. The cross-sectional area of tension or compression reinforcement should not exceed $0.04A_c$ outside lap locations.

4.4 Optimization of Double Reinforced Rectangular Beam Section

4.4.1 Problem Formulation

The optimization techniques in general enable designers to find the best design for the structure under consideration. In this particular case, the principal design objective is to minimize the total cost of structure, after full filling all the requirements according to ES EN 1992-1-1:2015. The resulting structure, should not only be marked with a low price but also comply with all strength and serviceability requirements for a given level of applied load. The reinforced concrete doubly reinforced beam subjected to imposed load is taken in this present research work, the cost optimisation is made for the structural elements.

The general form of an optimization problem is as follows

1. Given constant parameters

2. Find design variables
3. Minimize objective function
4. Satisfy design constraint

4.4.1.1 Constant Parameters

- Cost of concrete birr per m^3
- Cost of steel birr per kg
- Span of beam or L
- Dead load
- Live load
- Concrete cover
- Effective cover or dc
- Characteristics strength of steel f_{yk}
- Characteristics strength of concrete f_{ck}
- Secant modulus of elasticity of concrete E_{cm}

One can change the fixed paramers using the following dialogue box as can be seen in the Figure 4.7 and Figure 4.8 below.

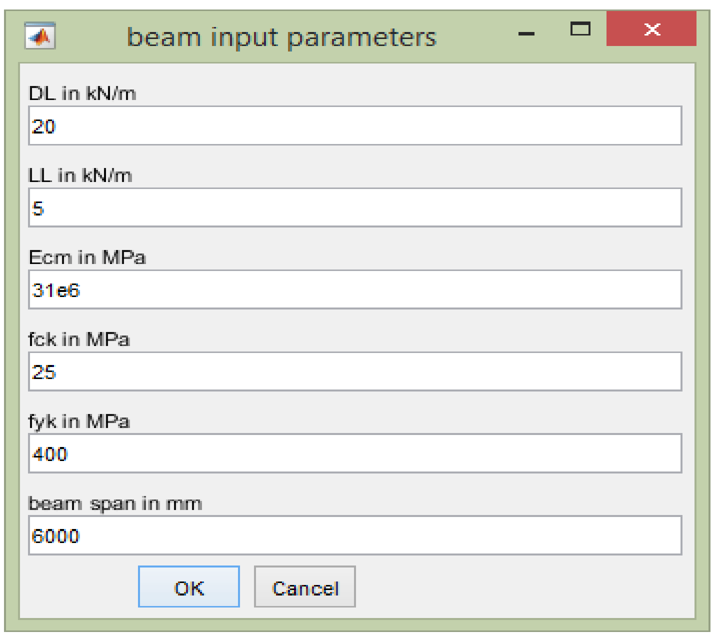


Figure 4.7: Beam input parameters dialogue box

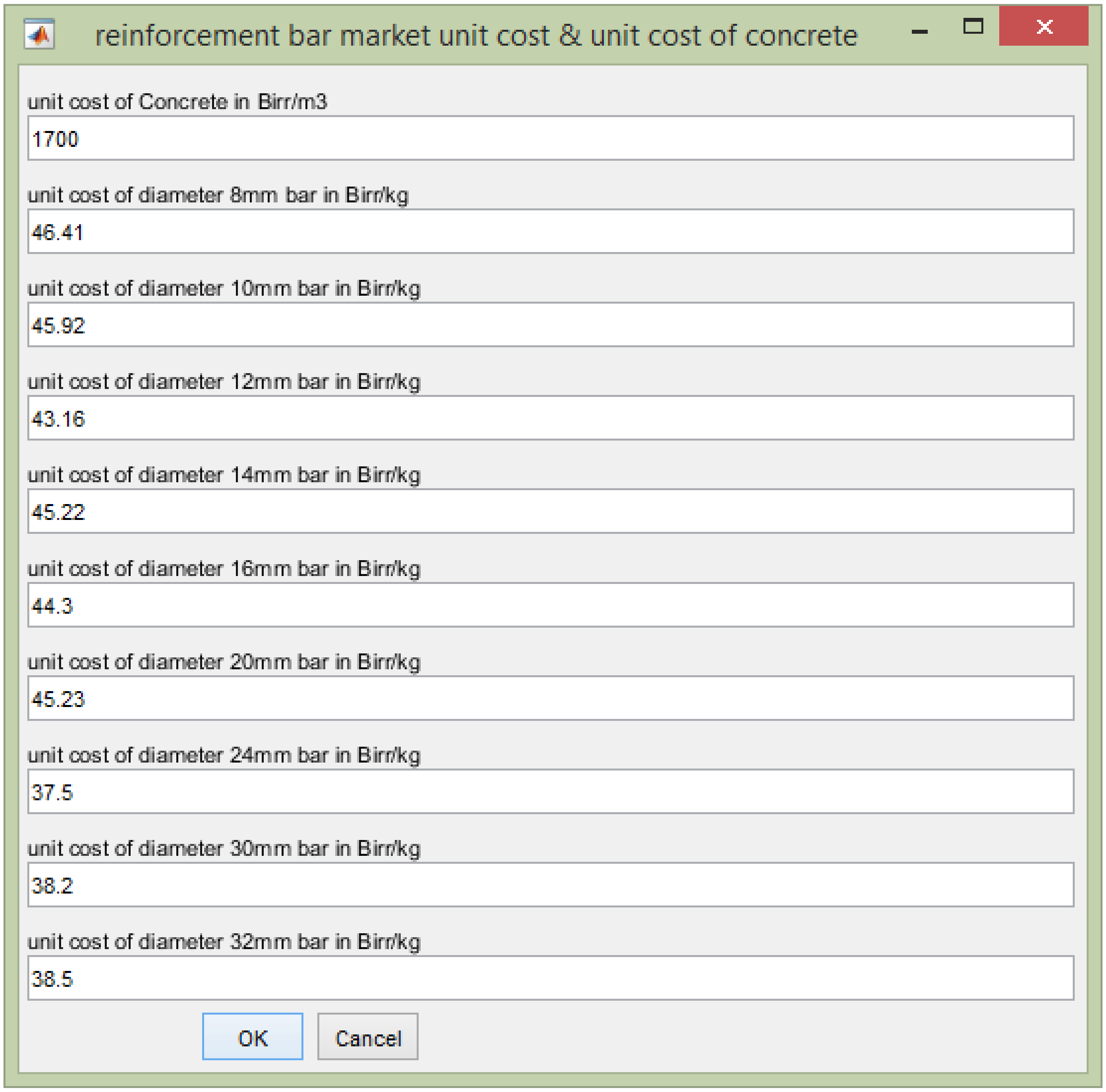


Figure 4.8: Reinforcement unit cost input dialogue box

4.4.1.2 Design Variables

Design variables In this problem are taken as discrete variables. Design variables for doubly reinforced beam are:

- Width of beam = b = x1
- Depth of beam = d = x2
- Number of bars for steel in tension zone = bars no (1) = x3
- Diameter of bars for steel in tension zone = dia1= x4
- Number of bars for steel in compression zone= bars no (2) = x5

- Diameter of bars for steel in compression zone =dia2= x6
- Diameter of bars for shear reinforcement = dia3=x7
- Spacing for shear reinforcement = sv= x8

Set of discrete values for design variables:

b in mm = (200-500) step size- 50

Diameter tension bar in mm = (10,12,14,16,20,24,30,32) bars no (1)= (2, 3, 4, 5, 6,7,8,9,10,11,12)
Diameter compression bar in mm = (10,12,14,16,20,24,30,32) bars no (2) = (2, 3, 4, 5, 6,7,8,9,10,11,12)
Diameter of stirrups = (8, 10,12)

Spacing of stirrups in mm = (180,200,220,240,260,280,300)

4.4.1.3 Objective Function

The objective function to be minimized:

$$F = C_c x V_{beamconcrete} + \gamma_s x C_s x \left(V_{steel} + V_{stirrup}\right) \quad (4.32)$$

Where:

C_c is unit cost of concrete birr/m^3

C_d is unit cost of reinforcement bar diameter birr/kg

4.4.1.4 Design Constraints

1. Ductility constraint: $x \leq x_a = 0.45xd$ (4.33)

2. Constraint for minimum area of tension reinforcement: $A_{s,\min} = 0.26 \frac{f_{ctm}}{f_{yk}} b_t d \geq 0.0013 b_t d$

3. Constraint for maximum area of tension reinforcement: $A_{st} \leq 0.04$ bxD (4.34)

4. Constraint for maximum area of compression reinforcement: $A_{sc} \leq 0.04$ bxD (4.35)

5. Depth to width ratio constraint: r = d/b; $1.5 \leq r \leq 2.5$

6. Moment-equilibrium constraint: $M_{sd} \leq M_{Rd,\lim} = 0.167 f_{ck} bd^2$ (4.36)

7.Shear constraint $V_{Sd} < V_{Rd,\max} = 0.18 bdf_{ck}\left(1 - {f_{ck}}/{250}\right)$ (4.37)

8. Deflection constraint (serviceability constraint):$\delta_{tot} \leq \delta_{all}$ =L/250

4.5 Optimization Example for Simply supported Reinforced Concrete Beam

One application of the optimization program is demonstrated with structural analysis for simply supported beam having span length of 6m and subjected to live load 5kN/m and dead load of 20 kN/m inaddition to the self weigth of the rectangular double reinforced concrete beam section Use concrete class C30/25 and steel grade S-500 for desgin optimization.

Openning MATLAB and selecting optimize fig file from MATLAB path directory and running the optimize fig file display the interface shown in figure 4.9 below.

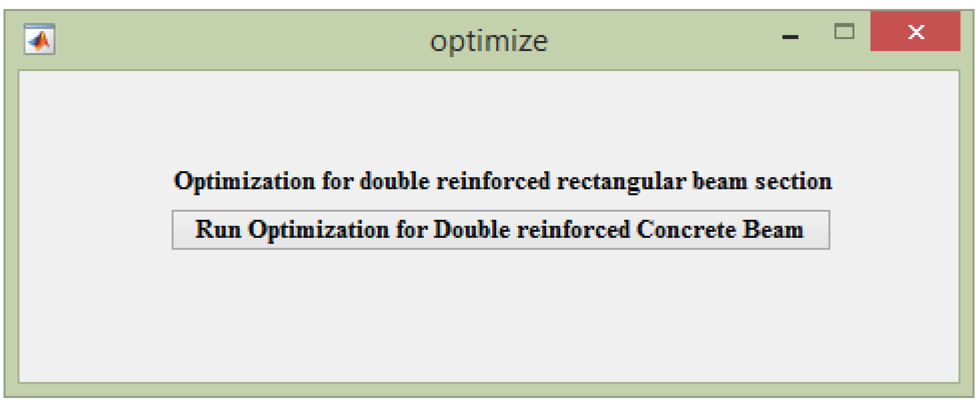


Figure 4.9: Optimzation running interface

And pressing the run optimization pushbutton of Figure 4.9 above first display beam input parameter dialogue box as can be seen in Figure 4.7 above with default value or changing the default value is possible and pressing ok button enable the second dialogue box to displayed for reinforcement unit cost input with default value or changing the default value is possible as can be seen in Figure 4.8 above. After this optimization iteration start by displaying Figure 4.10 below.

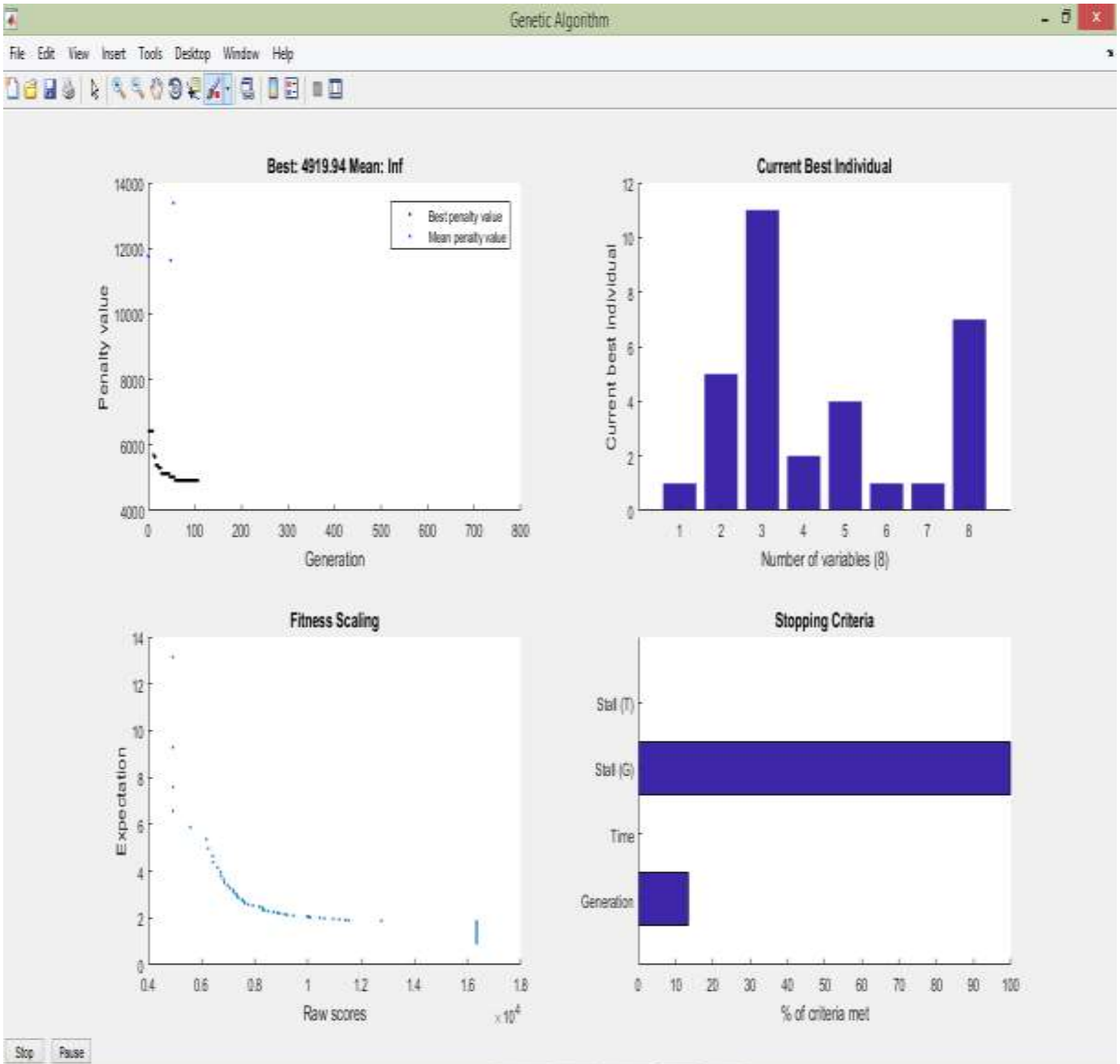


Figure 4.10: Genetic algorithm iteration display

Optimization terminated: average change in the penalty fitness value less than options.FunctionTolerance and constraint violation is less than options.ConstraintTolerance.

The resulting design variable after some iteration by genetic algorizim is gven below

b = 200mm

d = 400mm

Number of tension bar = 12

Diameter of tension bar = 12mm

Number of compression bar = 5

Diameter of compression bar = 10mm

Diameter of stirrup = 8mm

Spacing of stirrups = 300mm

Cost function returned by ga = 4919.94 in birr.

4.6 Analysis and Design Example for Flexure

One application of the graphical user interface is demonstrated with analysis and design example of rectangular, T and Trapezoidal beam section subjected to a design bending moment.

4.6.1 Steps Followed in the Graphical User Interface Window

1. Select concrete grade and steel grade from the pop-up menu.
2. Select the size of bar from pop-up menu, input the number of bar in the input box for the respective rows of bar and evaluate area of reinforcement provide using respective push button.
3. All the cross-sectional parameters are inserted in the input box and the effective depth cover and effective depth evaluated by pressing the push button.
4. Pressing the push button in step four enable to estimate the following parameter α_c, ε_{yd}, ε_{s1}, K_x, x (ε_{s2} if it is a double reinforced beam section), βc, z, enable to identifies failure type and enable to evaluate the section moment resistance (M_{Rd}) whether it is single or double reinforced beam section.

Finally for the design part iterate the above steps until the following condition satisfied.

- *l/d* actual < *l/d* basic according to the code provision.
- $K_x < K_{x,lim} = 0.448$ for ductility purpose.
- $Ms_d <= M_{Rd,lim}$

- Area of steel provided is between $A_{s,min}$ and $A_{s,max}$.

To illustrate the use of the program example for rectangular,T or L and trapezoidal sections are shown in Figure 4.11 - 4.18.

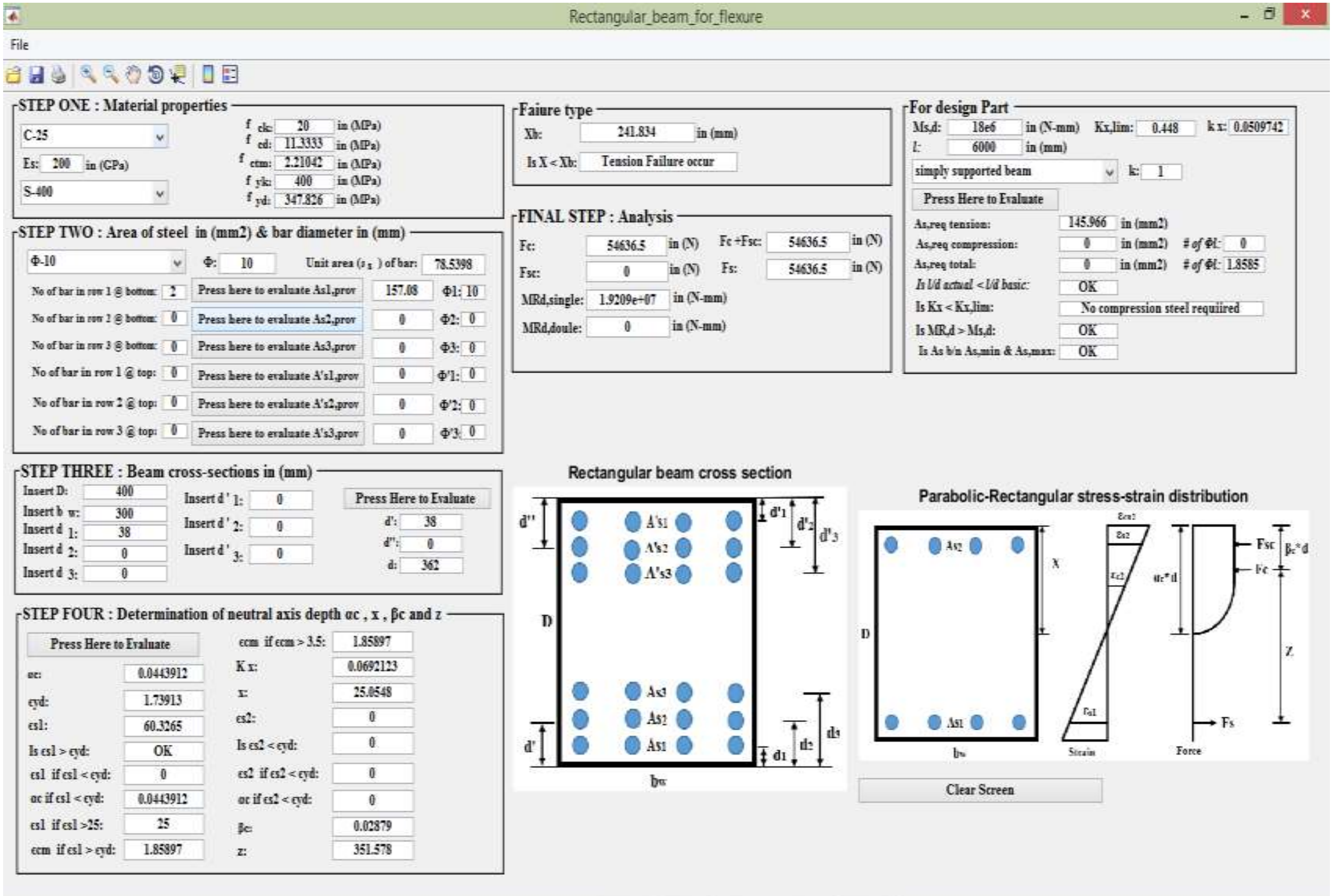


Figure 4.11: Sample design and estimation of flexural resistance of rectangular single reinforced concrete beam section

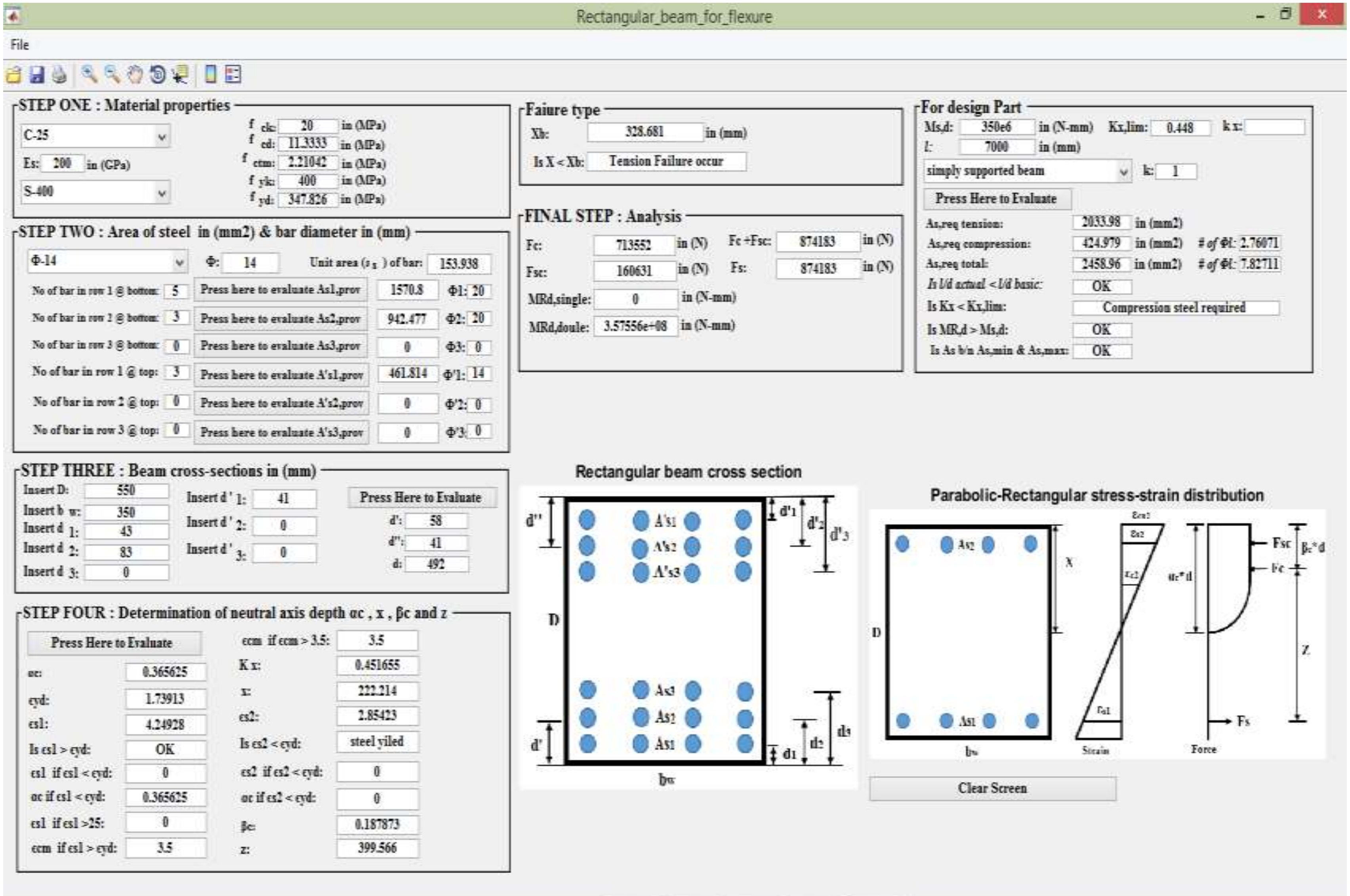


Figure 4.12: Sample design and estimation of flexural resistance of rectangular double reinforced concrete beam section

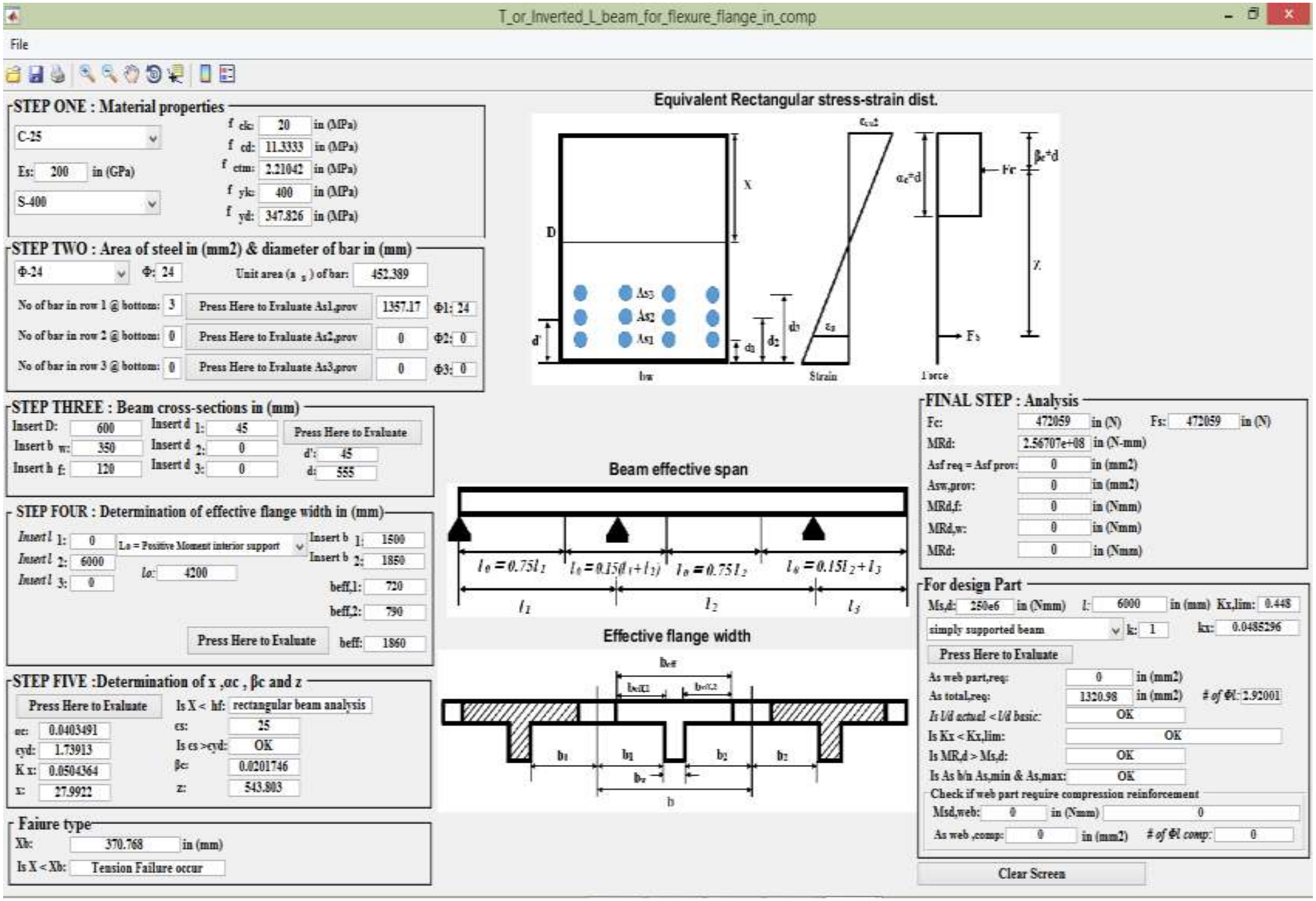


Figure 4.13: Sample design and estimation of flexural resistance of T reinforced concrete beam section treated as rectangular beam

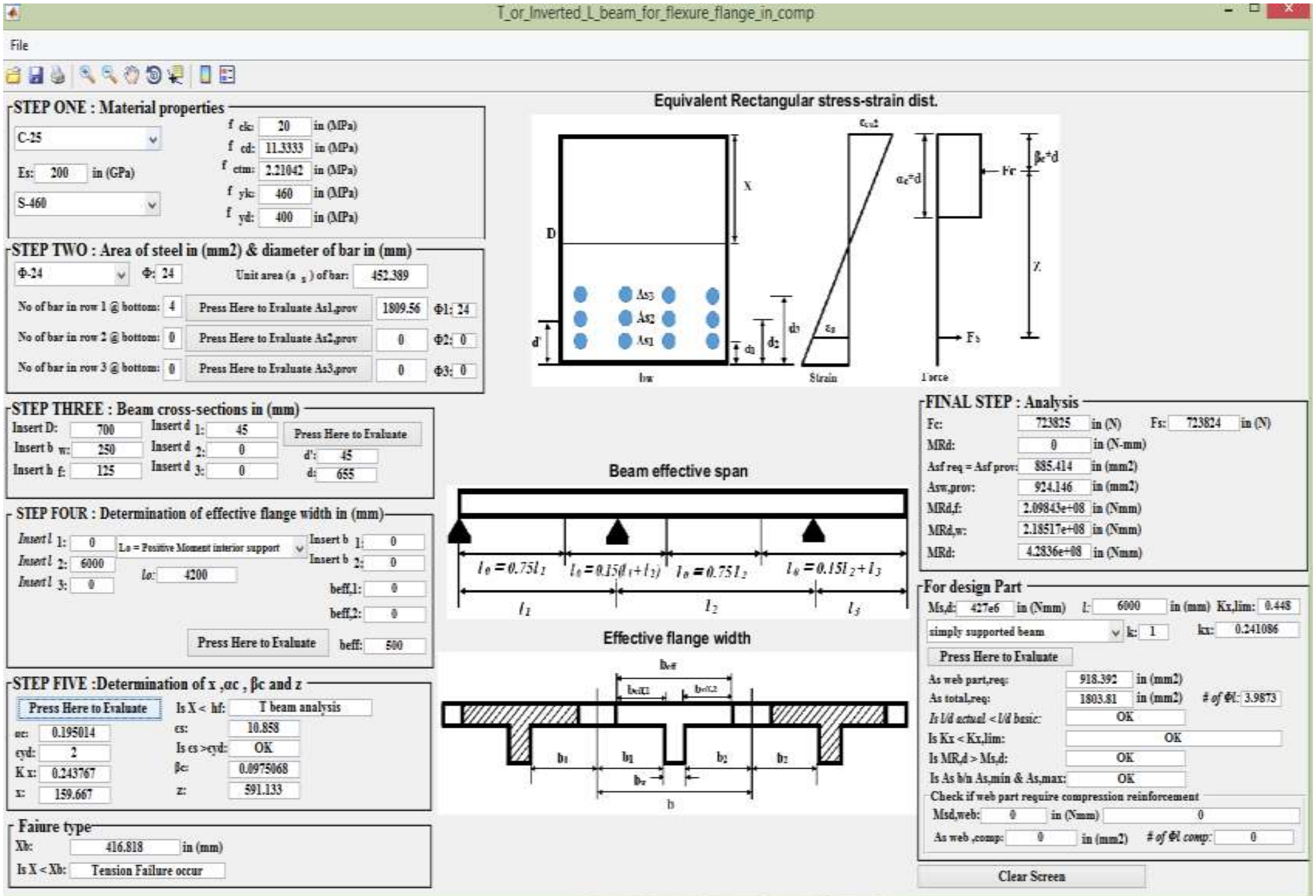


Figure 4.14: Sample design and estimation of flexural resistance of T reinforced concrete beam section treated as T beam

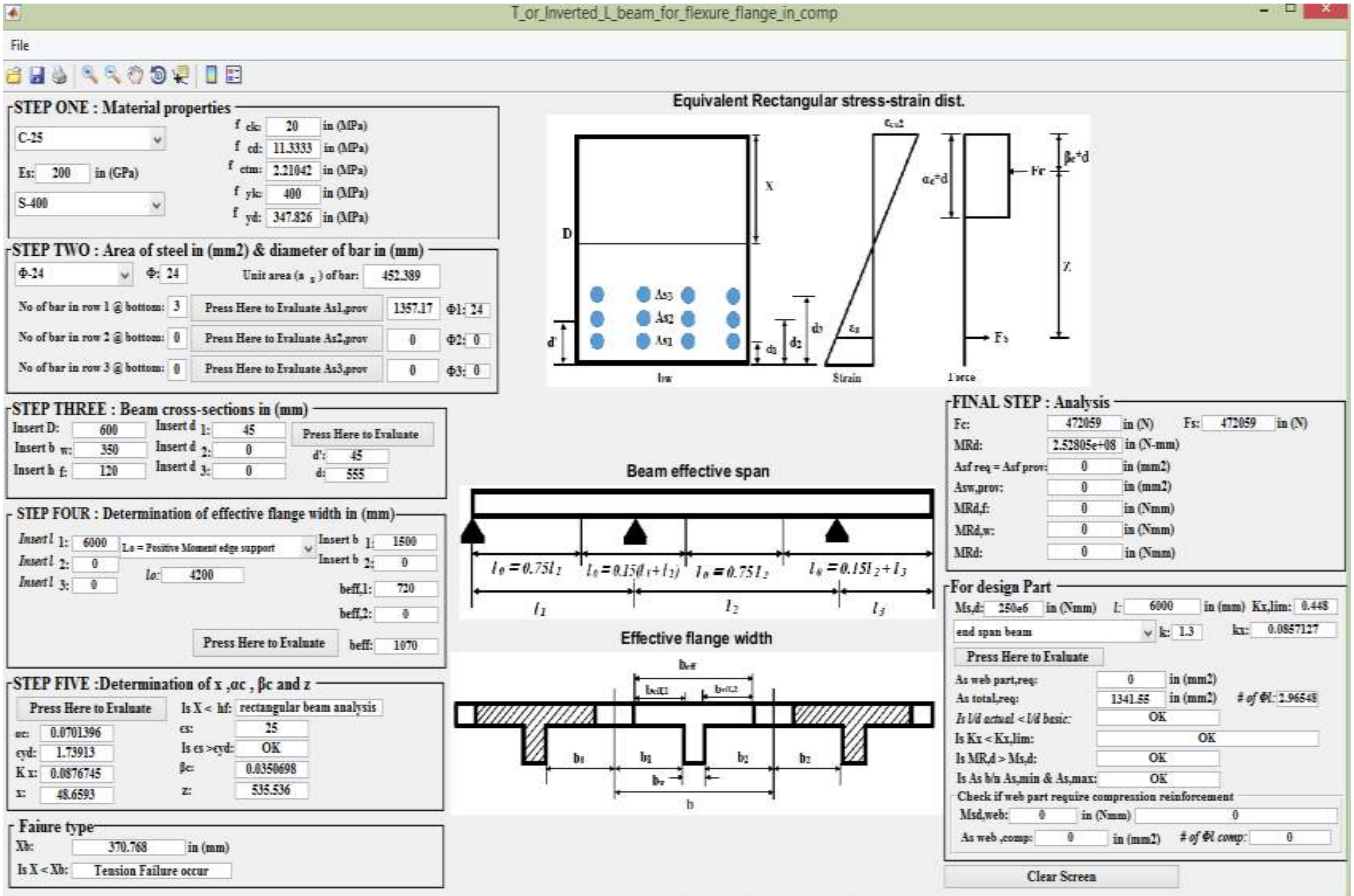


Figure 4.15: Sample design and estimation of flexural resistance of inverted L reinforced concrete beam section treated as rectangular beam

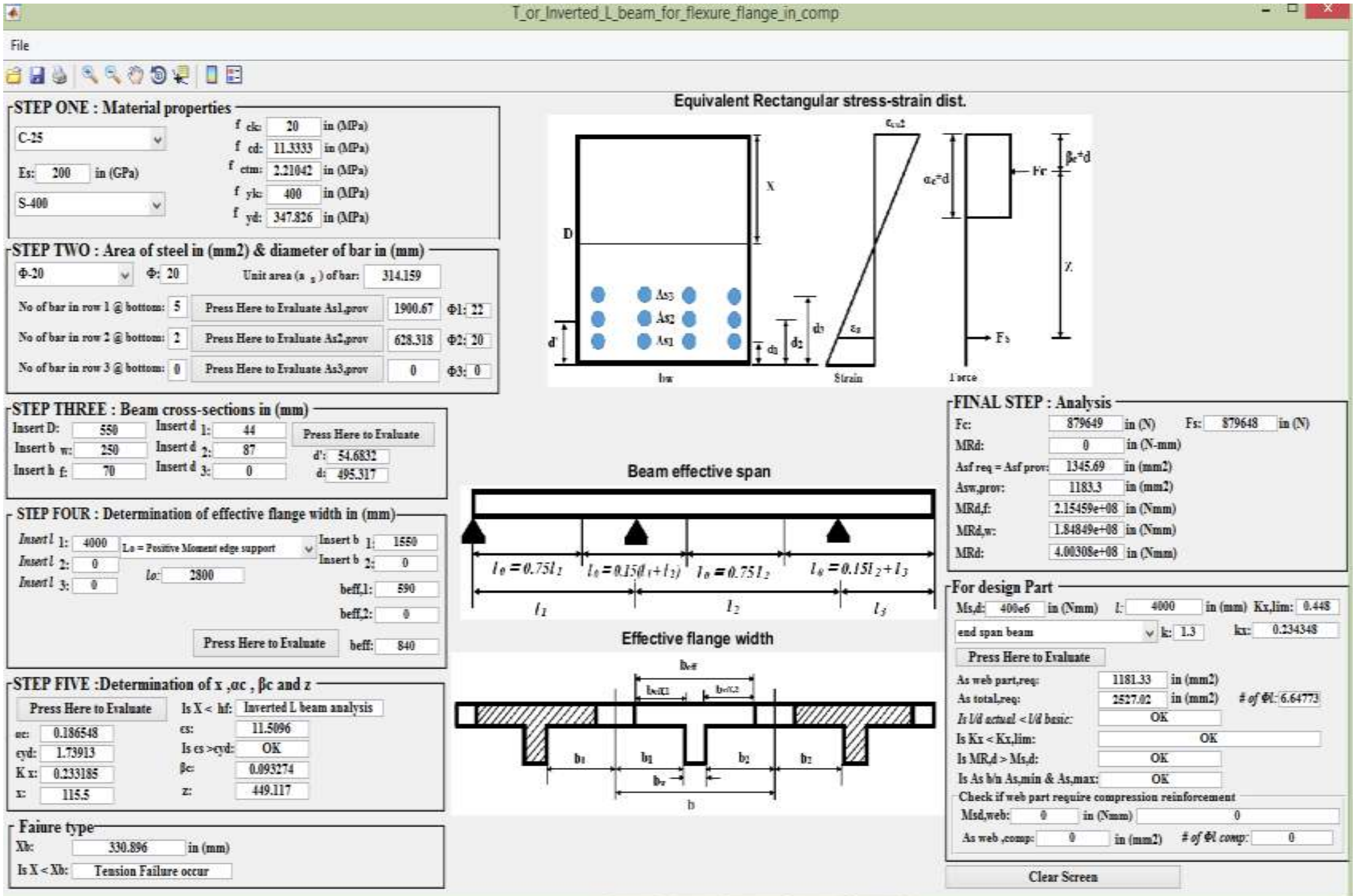


Figure 4.16: Sample design and estimation of flexural resistance of inverted L reinforced concrete beam section treated as inverted L beam

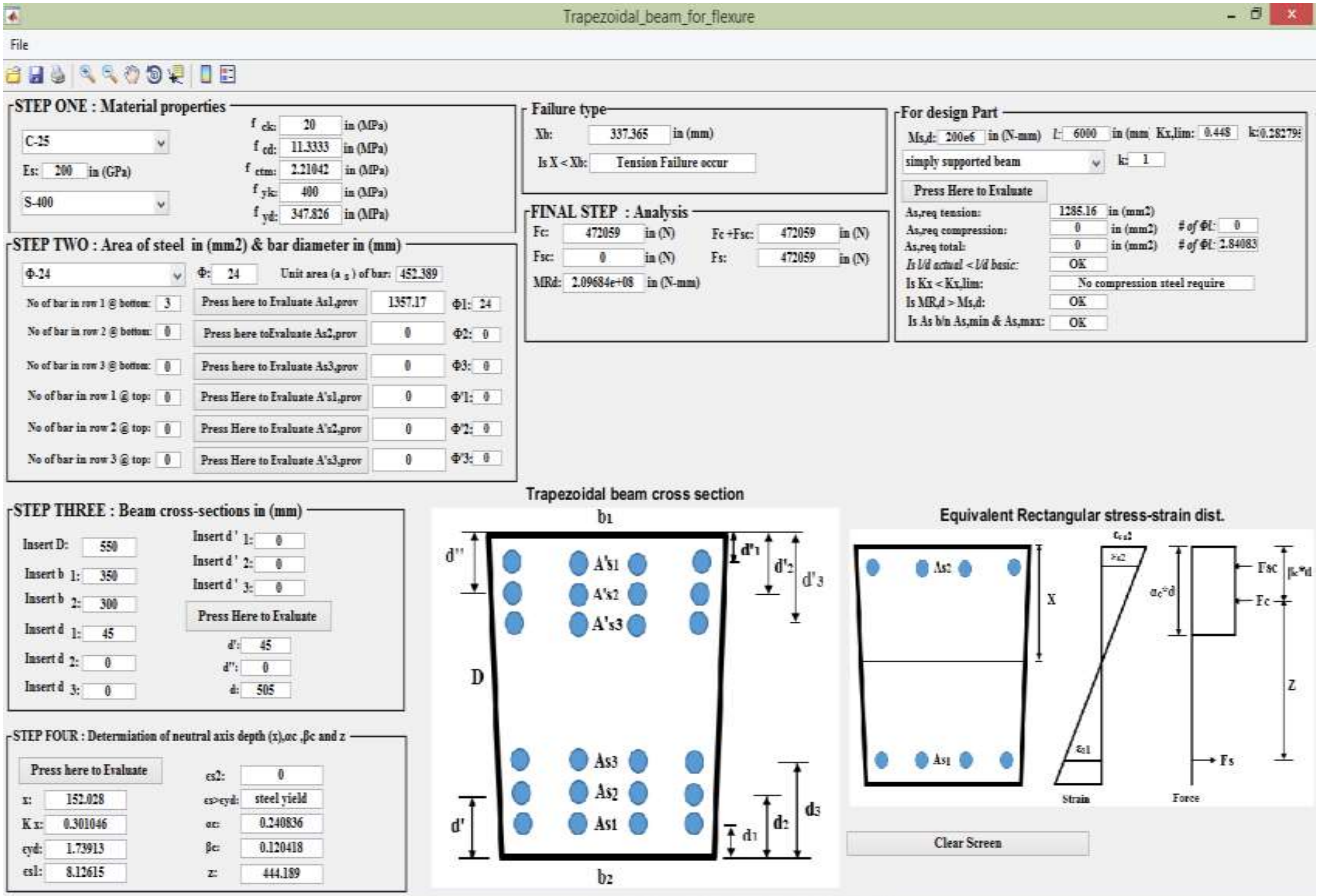


Figure 4.17: Sample design and estimation of flexural resistance of Trapezoidal single reinforced concrete beam section

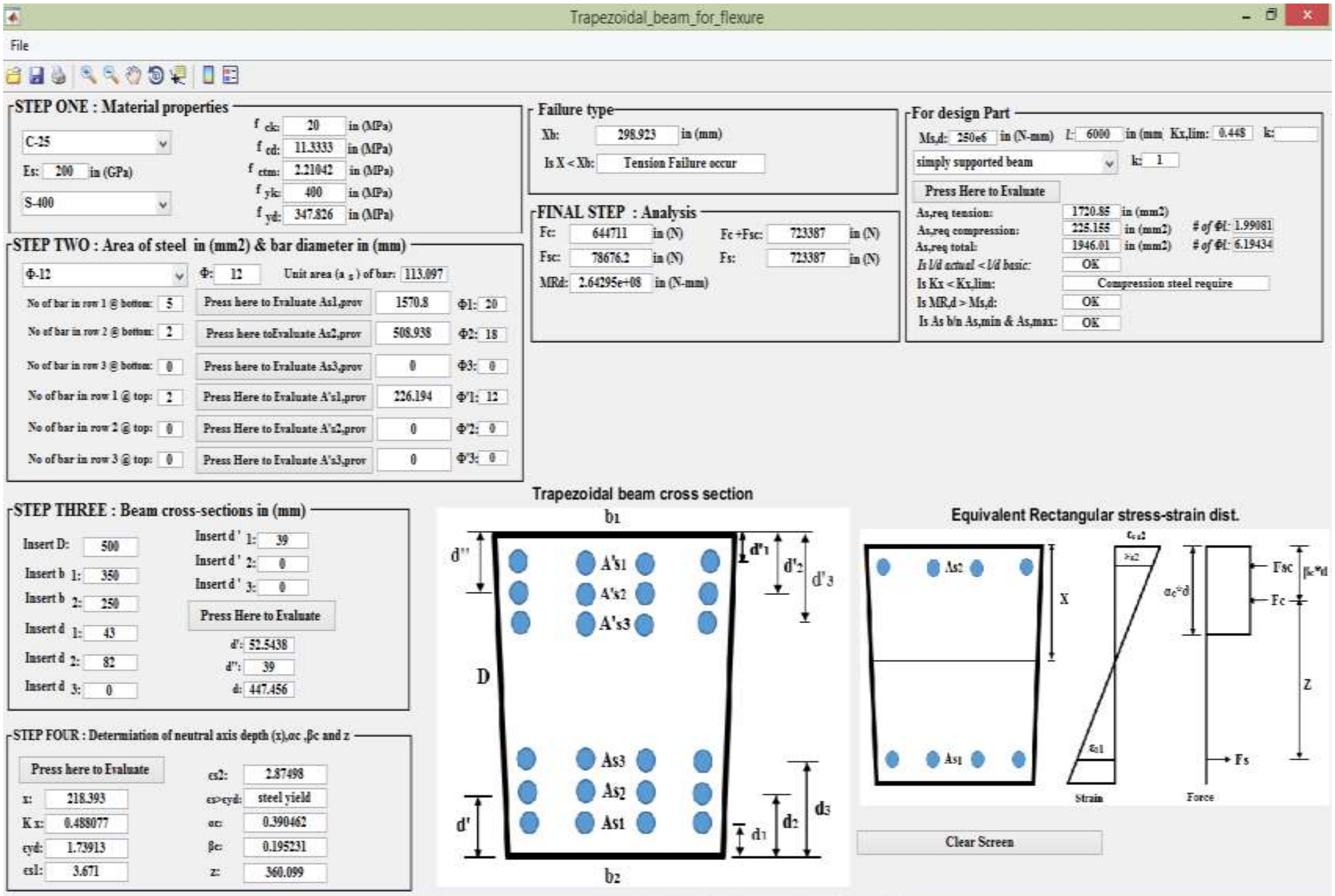


Figure 4.18: Sample design and estimation of flexural resistance of Trapezoidal double reinforced concrete beam section

5. ANALYSIS AND DESIGN OF REINFORCED CONCRETE BEAM SECTIONS FOR SHEAR

5.1 Design of Beams for Shear According to EN 1992-1-1:2015

For the verification of the shear resistance the following symbols are defined:

$V_{Rd,c}$ is the design shear resistance of the member without shear reinforcement.

$V_{Rd,s}$ is the design value of the shear force which can be sustained by the yielding of shear reinforcement.

$V_{Rd,max}$ is the design value of the maximum shear force which can be sustained by the member, limited by crushing of the compression struts.

In regions of the member where , $V_{Ed} \leq V_{Rd,c}$ no calculated shear reinforcement is necessary. V_{Ed} is the design shear force in the section considered resulting from the external loading.

In regions where $V_{Ed} > V_{Rd,c}$ sufficient shear reinforcement should be provided in order that $V_{Ed} \leq V_{Rd.}$

The design shear force should not exceed the permitted maximum value $V_{Rd,max}$, anywhere in the member.

For members subject to predominantly uniformly distributed loading, the design shear force need not be checked at a distance less than d from the face of the support. Any shear reinforcement required should continue to the support. In addition it should be verified that the shear at the support does not exceed $V_{Rd,max}$.

5.2 Members with Minimum Shear Reinforcement

The design value for the shear resistance $V_{Rd,c}$ is given by:

$$V_{Rd,c} = \left[C_{Rd,c} k (100 \rho_l f_{ck})^{1/3} + k_1 \sigma_{cp}\right] b_w d \tag{5.1}$$

with a minimum of

$$V_{Rd,c} = (v_{min} + k_1 \sigma_{cp}) b_w d \tag{5.2}$$

where:

f_{ck} is the characterstice compressive strength in MPa

$$k = 1 + \sqrt{200/d} \leq 2.0 \text{ with d in mm} \tag{5.3}$$

$$\rho_l = \frac{A_{sl}}{b_w d} \leq 0.02 \tag{5.4}$$

A_{sl} is the area of the tensile reinforcement, which extends ≥ (l_{bd} + d) beyond the section considered (see Figure 5.1)

b_w is the smallest width of the cross-section in the tensile area [mm]

A_C is the area of concrete cross section [mm^2]

$V_{Rd,c}$ is [N]

The recommended value for $C_{Rd,c}$ is $0.18/\gamma_c$, that for ν_{min} is $0.035k^{3/2}$-$f^{1/2}_{ck}$ and that for k_1 is 0.15.

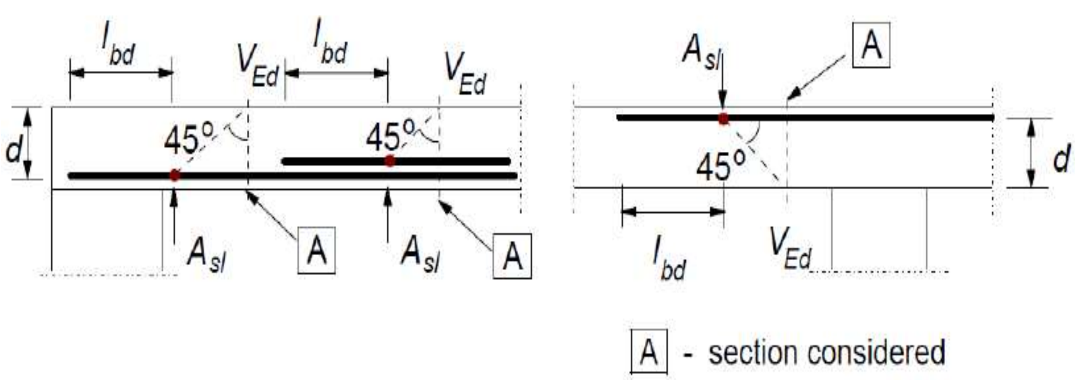


Figure 5.1: Definition of A_{sl} in Expression (5.1) (EN, 1992-1-1:2015)

5.3 Members requiring Design Shear Reinforcement

1. The design of members with shear reinforcement is based on a truss model (Figure 5.2).

In Figure 5.2 the following notations are shown:

α is the angle between shear reinforcement and the beam axis perpendicular to the shear force (measured positive as shown in Figure 5.2)

θ is the angle between the concrete compression strut and the beam axis perpendicular to the shear force

F_{td} is the design value of the tensile force in the longitudinal reinforcement

F_{cd} is the design value of the concrete compression force in the direction of the longitudinal member axis.

b_w is the minimum width between tension and compression chords

z is the inner lever arm, for a member with constant depth, corresponding to the bending moment in the element under consideration. In the shear analysis of reinforced concrete without axial force, the approximate value z = 0.9d may normally be used.

2. The angle θ should be limited. The recommended limits for cot θ is $1 \leq \cot\theta \leq 2.5$.

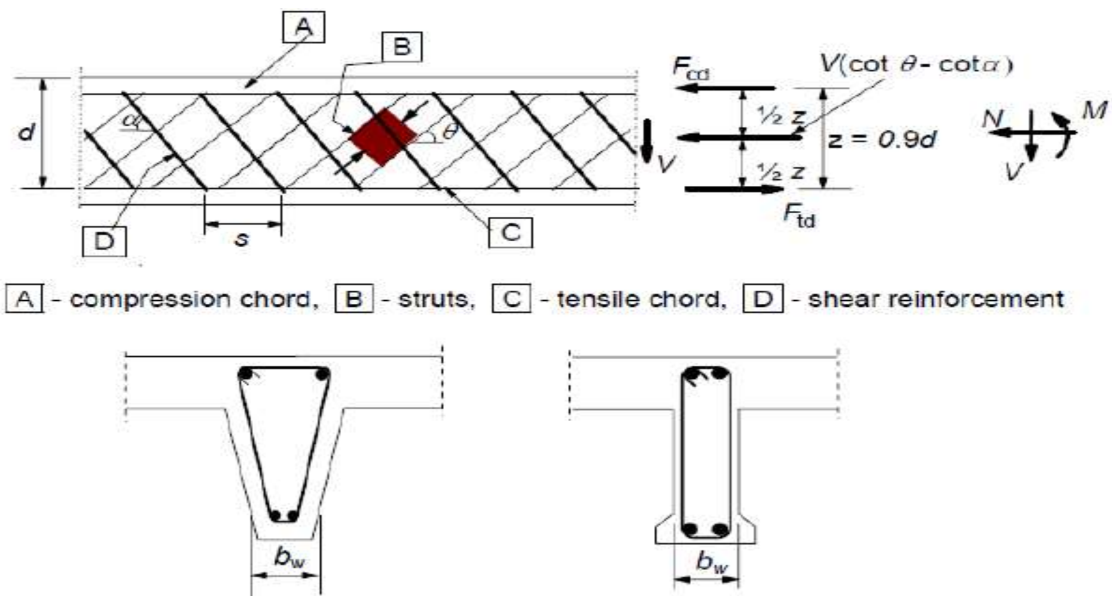


Figure 5.2: Truss model and notation for shear reinforced (EN, 1992-1-1:2015)

5.3.1 Members with Vertical Shear Reinforcement

For members with vertical shear reinforcement, the shear resistance, V_{Rd} is the smaller value of $V_{Rd,s}$ and $V_{Rd,max}$:

$$V_{Rd,s} = \frac{A_{sw}}{s} z f_{ywd} \cot\theta \qquad (5.5)$$

$$V_{Rd,\max} = \alpha_{cw} b_w z \upsilon f_{cd} \Big/ (\cot\theta + \tan\theta) \qquad (5.6)$$

where:

A_{sw} is the cross-sectional area of the shear reinforcement

s is the spacing of the stirrups

f_{ywd} is the design yield strength of the shear reinforcement

v is a strength reduction factor for concrete cracked in shear

α_{cw} is a coefficient taking account of the state of the stress in the compression chord

For reinforced and prestressed members, if the design stress of the shear reinforcement is below 80% of the characteristic yield stress f_{yk}, v may be taken as:

$$v = 0.6 \text{ for } f_{ck} \leq 60 \text{ MPa} \tag{5.7}$$

$$v = 0.9 - f_{ck}/200 > 0.5 \text{ for } f_{ck} \geq 60 \text{ MPa} \tag{5.8}$$

The recommended value of $\tan \alpha_c$ is 1 for non-prestressed structures.

The maximum effective cross-sectional area of the shear reinforcement $A_{sw,max}$ is given by:

$$\frac{A_{sw,\max} f_{ywd}}{b_w s} \leq \frac{1}{2} \alpha_c v f_{cd} \tag{5.9}$$

5.3.2 Members with Inclined Shear Reinforcement

For members with inclined shear reinforcement, the shear resistance is the smaller value of $V_{Rd,s}$ and $V_{Rd,max}$:

$$V_{Rd,s} = \frac{A_{sw}}{s} z f_{ywd} [\cot\theta + \cot\alpha] \sin\alpha < V_{Rd,max} \tag{5.10}$$

$$V_{Rd,\max} = \alpha_{cw} b_w z \upsilon_1 f_{cd} (\cot\theta + \cot\alpha) \Big/ (1 + \cot\theta^2) \tag{5.11}$$

The maximum effective shear reinforcement $A_{sw,max}$ follows from:

$$\frac{A_{sw,\max} f_{ywd}}{b_w s} \leq \frac{\frac{1}{2} \alpha_c v f_{cd} \sin\alpha}{1 - \cos\alpha} \tag{5.12}$$

5.4 Additional Tensile Force in Longitudinal Reinforcement

The longitudinal tension reinforcement should be able to resist the additional tensile force caused by shear.

The additional tensile force, ΔF_{td} in the longitudinal reinforcement due to shear V_{Ed} may be calculated from:

$$\Delta F_{td} = 0.5V_{Ed}\left(\cot\theta - \cot\alpha\right) \qquad (5.13)$$

$\left(M_{Ed}/Z\right) + \Delta F_{td}$ should be taken not greater than $M_{Ed,max}/z$

5.5 Minimum Area and Maximum Spacing of Shear Reinforcement

The ratio of shear reinforcement is given by :

$$\rho_w = A_{sw}/\left(sb_w \sin\alpha\right) \qquad (5.14)$$

where:

ρ_w is the shear reinforcement ratio
ρ_w should not be less than $\rho_{w,min}$
A_{sw} is the area of shear reinforcement within length s
s is the spacing of the shear reinforcement measured along the longitudinal axis of the member
b_w is the breadth of the web of the member
α is the angle between shear reinforcement and the longitudinal axis

When, on the basis of the design shear calculation, no shear reinforcement is required, minimum shear reinforcement should nevertheless be provided. The minimum shear reinforcement may be omitted in members such as slabs (solid, ribbed or hollow core slabs) where transverse redistribution of loads is possible. Minimum reinforcement may also be omitted in members of minor importance which do not contribute significantly to the overall resistance and stability of the structure.

The recommended value of $\rho_{w,min}$ is

$$\rho_{w,\min} = \left(0.08\sqrt{f_{ck}}\right) / f_{yk} \tag{5.15}$$

The maximum longitudinal spacing between shear assemblies should not exceed $S_{l,max}$.

The recommended value $s_{l,max}$ is

$$s_{l,\max} = 0.75d\left(1+\cot\alpha\right) \tag{5.16}$$

5.6 Analysis and Design Example for Shear

One application of this study is demonstrated with analysis and design example of rectangular beam section with vertical and inclined stirrup subjected to design shear force.

5.6.1 Steps followed in the Graphical User Interface Window

1. Select concrete grade and steel grade from the pop-up menu.
2. Select the size of bar from pop-up menu and input number of bar in the input box to evaluate A_{sl} and A_{sw} by pressing the push button.
3. All the cross-sectional parameters are inserted in the input box and then effective depth evaluated by pressing the push button.
4. Evaluate the following parameter κ and ρ_1 which used for the determination of concrete shear capacity ($V_{Rd,c}$) by pressing the push button.

Final step input s,α and the limit of cotθ in the input box and the design shear resistance(V_{Rd}) evaluated by pressing the push button.

For the design part input V_{Ed} in the input box and compare it with $V_{Rd,c}$ .If $V_{Rd,c} > V_{Ed}$ the selected cross section and area of reinforcement is ok. If not iterate the procedure from step one to the final step until it satisfied the design criteria to obtain acceptable cross-section dimensions and area of steel. To illustrate the use of the program the graphical user interface example is shown in Figure 5.3 and Figure 5.4 below for beam with vertical and inclined stirrup respectively.

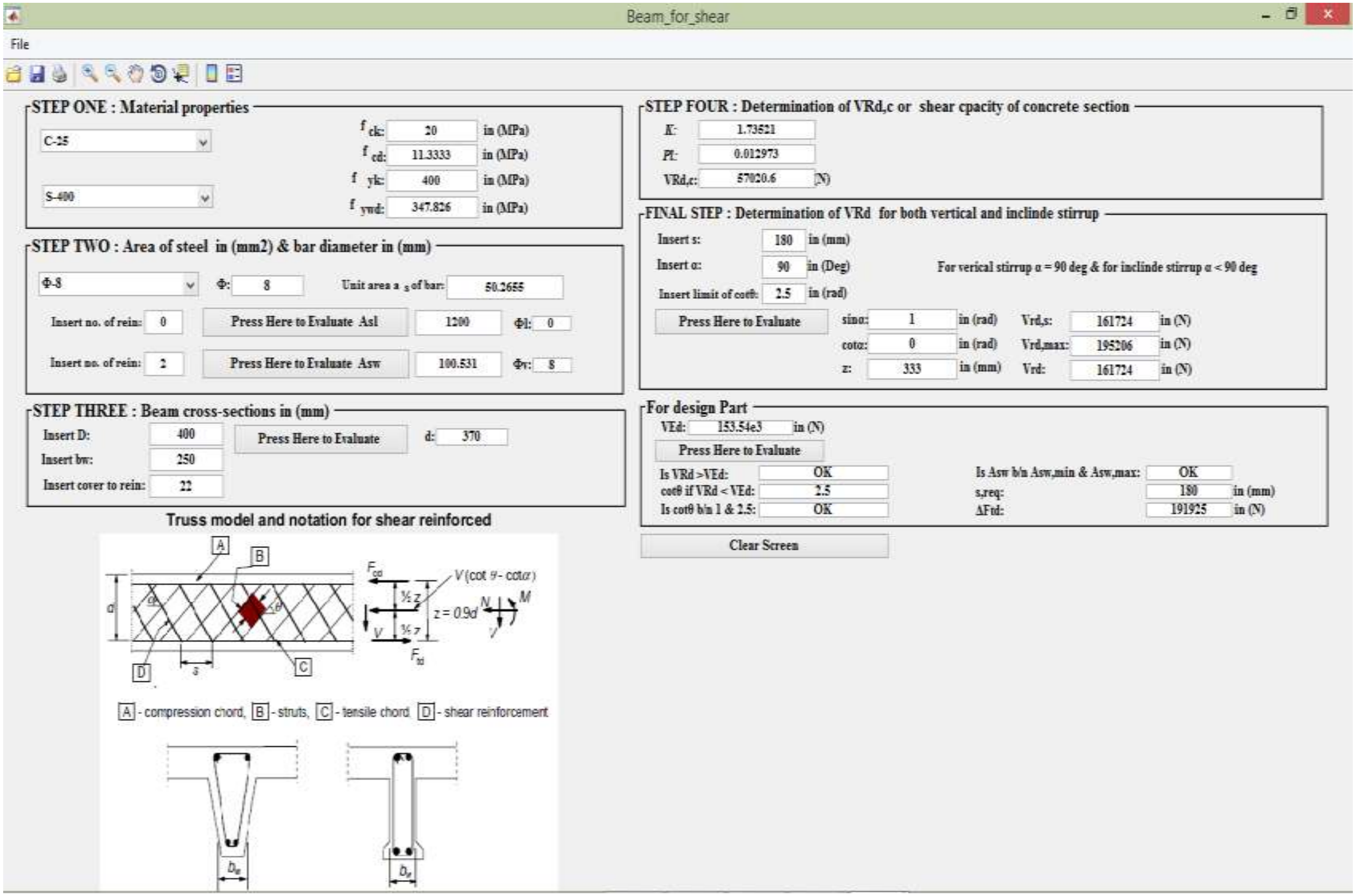


Figure 5.3: Sample design and estimation of shear resistance of rectangular reinforced concrete beam section with vertical stirrup

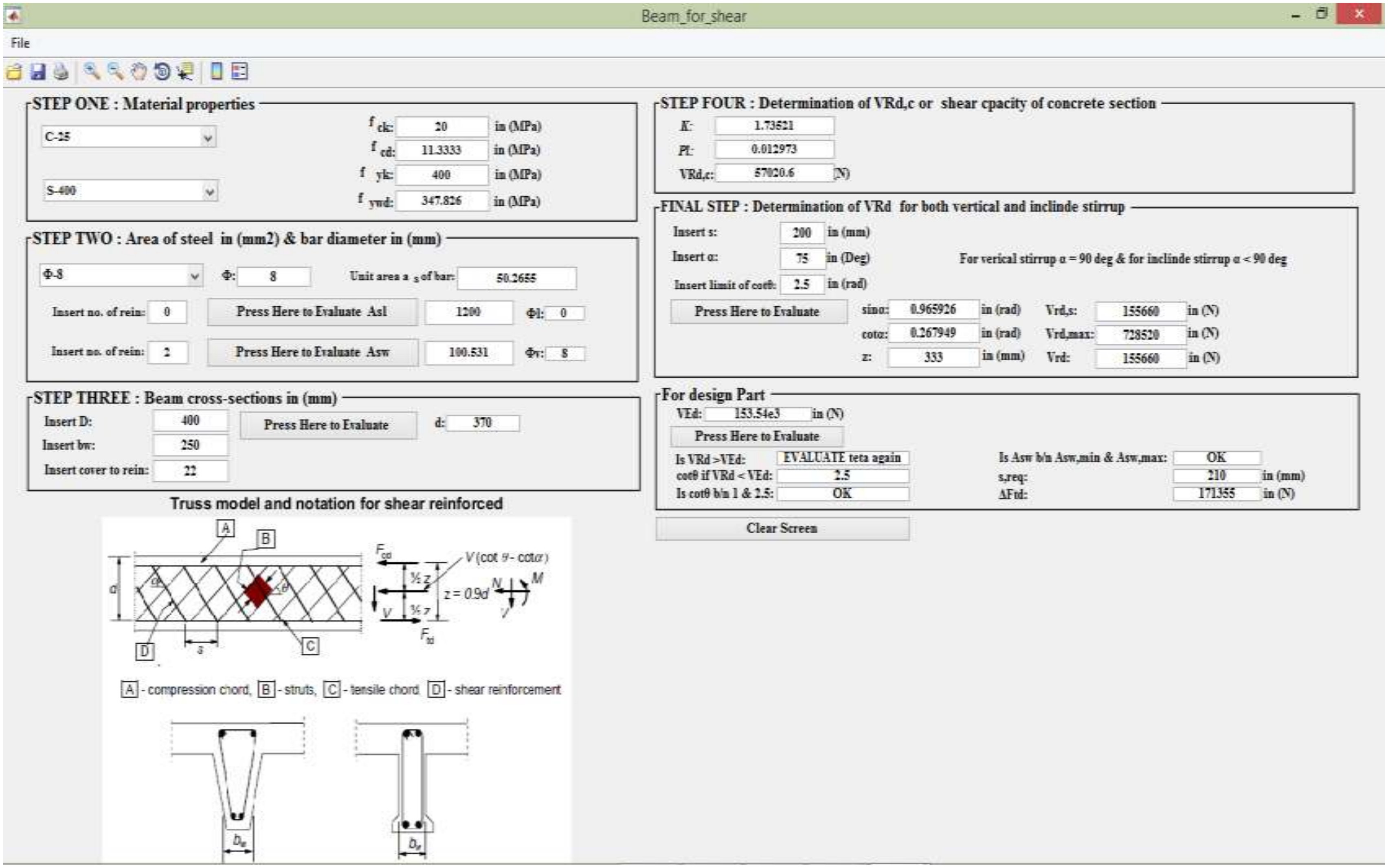


Figure 5.4: Sample design and estimation of shear resistance of rectangular reinforced concrete beam section with inclined stirrup

6. ANALYSIS AND DESIGN OF REINFORCED CONCRETE BEAM SECTIONS FOR TORSION AND COMBINED EFFECT OF SHEAR WITH TORSION

6.1 Design of Beams for Torsion According to EN 1992-1-1:2015

6.1.1 Requirements of Torsion

6.1.1.1 General

1. Where the static equilibrium of a structure depends on the torsional resistance of elements of the structure, a full torsional design covering both ultimate and serviceability limit states shall be made.
2. In statically indeterminate structures, torsion arise from consideration of compatibility only, and the structure is not dependent on the torsional resistance for its stability, then it will normally be unnecessary to consider torsion at the ultimate limit state. In such cases a minimum reinforcement in the form of stirrups and longitudinal bars should be provide to prevent excessive cracking.
3. The torsional resistance of a section may be calculated on the basis of a thin-walled closed section, in which equilibrium is satisfied by a closed shear flow. Solid sections may be modeled by equivalent thin-walled sections. Complex shapes, such as T-sections, may be divided into a series of sub-sections, each of which is modeled as an equivalent thin walled section, and the total torsional resistance taken as the sum of the capacities of the individual elements.
4. The distribution of the acting torsional moments over the sub-sections should be in proportion to their uncracked torsional stiffness. For non-solid sections the equivalent wall thickness should not exceed the actual wall thickness.
5. Each sub-section may be designed separately.

6.1.1.2 Design Consideration for Torsion

1. The shear stress in a wall of a section subject to a pure torsional moment may be calculated from:

$$\tau_{t,i} t_{ef,i} = \frac{T_{Ed}}{2A_K} \tag{6.1}$$

The shear force $V_{Ed,i}$ in a wall i due to torsion is given by

$$V_{ED,i} = \tau_{t,i} t_{ef,i} Z_i \tag{6.2}$$

Where

T_{Ed} is the applied design torsion (see Figure 6.1 below)

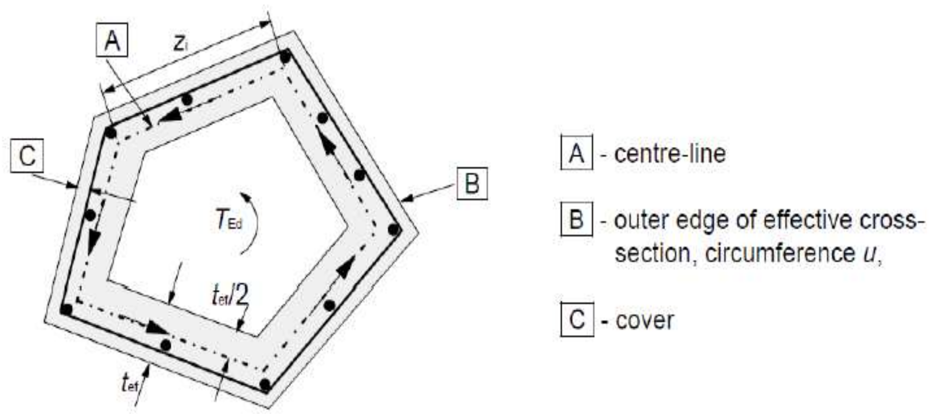


Figure 6.1: Notations and definitions used in section for torsion (EN, 1992-1-1:2015)

A_k is the area enclosed by the center-lines of the connecting walls, including inner hollow areas

$\tau_{t,i}$ is the torsional shear stress in wall i

$t_{ef,i}$ is the effective wall thickness. It may be taken as A/u, but should not be taken as less than twice the distance between edge and center of the longitudinal reinforcement. For hollow sections the real thickness is an upper limit

A is the total area of the cross-section within the outer circumference, including inner hollow areas

U is outer circumference of the cross-section

z_i is the side length of wall i defined by the distance between the intersection points with the adjacent walls

2. The effects of torsion and shear for both hollow and solid members may be superimposed, assuming the same value for the strut inclination θ. The limits for θ given in section 6.2.3(2) of EBCS EN 1992-1-1:2015 fully applicable for the case of combined shear and torsion. The maximum bearing capacity of a member loaded in shear and torsion follows from section 6.3.2(4) of EBCS EN 1992-1-1:2015.

3. The required cross-sectional area of the longitudinal reinforcement for torsion $\sum Asl$ maybe calculated from expression (6.3) below.

$$\frac{\sum A_{sl} f_{yd}}{u_k} = \frac{T_{Ed}}{2A_k} \cot\theta \qquad (6.3)$$

Where:

u_k is the perimeter of the area A_k

f_{yd} is the design yield stress of the longitudinal reinforcement A_{sl}

θ is the angle of compression struts

In compressive chords, the longitudinal reinforcement may be reduced in proportion to the available compressive force. In tensile chords the longitudinal reinforcement for torsion should be added to the other reinforcement. The longitudinal reinforcement should generally be distributed over the length of side, z_i, but for smaller sections it may be concentrated at the ends of this length.

4. The maximum resistance of a member subjected to torsion and shear is limited by the capacity of the concrete struts. In order not to exceed this resistance the expression (6.4) should be satisfied:

$$\frac{T_{Ed}}{T_{Rd,\max}} + \frac{V_{Ed}}{V_{Rd,\max}} \leq 1 \qquad (6.4)$$

Where:

T_{Ed} is the design torsional moment

V_{Ed} is the design transverse force

$T_{Rd,max}$ is the design torsional resistance moment according to expression (6.5) or (6.6) below

$$T_{Rd,\max} = 2v\alpha_{CW} f_{Cd} A_k t_{ef,i} \sin\theta \cos\theta \quad (6.5)$$

$$T_{Rd,\max} = 2v\alpha_{CW} f_{Cd} A_k t_{ef,i} (\cot\theta + \cos\alpha)\sin^2\theta \quad (6.6)$$

Where

v follows from $\nu = 0.6\left[1 - \frac{f_{ck}}{250}\right]$ (6.7)

Recommended value of α_{cw} is as follows:

- 1 for non-prestressed structures
- $\left(1 + \sigma_{cp}/f_{cd}\right) for 0 < \sigma_{cp} \leq 0.25 f_{cd}$
- 1.25 for 0.25 $f_{cd} < \sigma_{cp} \leq 0.5 f_{cd}$
- $2.5\left(1 - \sigma_{cp}/f_{cd}\right)$ for 0.5 $f_{cd} < \sigma_{cp} \leq 1.0 f_{cd}$

where:

σ_{cp} is the mean compressive stress, measured positive, in the concrete due to the design axial force. This should be obtained by averaging it over the concrete section taking account of the reinforcement. The value of σ_{cp} need not be calculated at a distance less than 0.5d cotθ from the edge of the support.

$V_{Rd,max}$ is the maximum design shear resistance determined according to expressions (6.8) or (6.9) below.

$$V_{Rd,\max} = \alpha_{cw} b_w z \nu_1 f_{cd} / (\cot\theta + \tan\theta) \quad (6.8)$$

$$V_{Rd,\max} = \alpha_{cw} b_w z \nu_1 f_{cd} (\cot\theta + \cot\alpha) / (1 + \cot^2\theta) \quad (6.9)$$

In solid cross sections the full width of the web may be used to determine $V_{Rd,max}$.

5. For approximately rectangular solid sections only minimum reinforcement is required provided that the following condition is satisfied in expression (6.10):

$$\frac{T_{Ed}}{T_{Rd,c}} + \frac{V_{Ed}}{V_{Rd,c}} \leq 1 \tag{6.10}$$

Where

$T_{Rd,c}$ is the torsional cracking moment, which may be determined by setting $\tau_{t,i} = f_{ctd}$

$$T_{Rd,c} = 2tf_{ctd}A_k \tag{6.11}$$

$V_{Rd,c}$ follows from expression (6.12) below.

$$V_{Rd,c} = \left[C_{Rd,c}k(100\rho_1 f_{ck})^{1/3} + k_1\sigma_{cp}\right]b_w d \tag{6.12}$$

With minimum value of $v_{Rd,c} = (v_{min} + k_1\sigma_{cp})b_w d$ (6.13)

Where $v_{min} = 0.035k * 3/2 * f_{ck} * \frac{1}{2}$ (6.14)

6.1.1.3 Reinforcement and Spacing Provision for Torsion

1. The provisions of ES EN-1992-1-1:2015 section 9.2.2 (5) and (6) are generally sufficient to provide the minimum torsion links required.
2. The longitudinal spacing of the torsion links should not exceed u / 8 (see section 6.3.2 of ES EN-1992-1-1:2015, Figure 6.1, for the notation), or the requirement in section 9.2.2 (6) of of ES EN-1992-1-1:2015 or the lesser dimension of the beam cross-section.
3. The longitudinal bars should be so arranged that there is at least one bar at each corner, the others being distributed uniformly around the inner periphery of the links, with a spacing not greater than 350mm.

6.2 Analysis and Design Example for Torsion

One application of this study is demonstrated with analysis and design example of rectangular beam section with vertical and inclined stirrup subjected to design torsional moment.

6.2.1 Steps followed in the Graphical User Interface Window

1. Select concrete grade and steel grade from the pop-up menu.
2. Select the size of bar from pop-up menu and input number of bar in the input box to evaluate A_{sl} and A_{sw} by pressing the push button.
3. All the cross-sectional parameters are inserted in the input box and then effective depth (d), t, u_k and A_k evaluated by pressing the push button.
4. Evaluate the concrete torsion capacity ($T_{Rd,c}$) by pressing the push button.
5. Input s and α in the input box and evaluate parameter sinα, cotα, cotθ and sinθ by pressing the push button.

Final step evaluate the design torsion resistance (T_{Rd}) by pressing push button.

For the design part input T_{sd} in the input box and compare it with $T_{Rd,c}$. If $T_{Rd,c} > T_{sd}$ the selected cross section and area of reinforcement is ok. If not iterate the procedure from step one to the final step until it satisfied the design criteria to obtain acceptable cross-section dimensions and area of steel . To illustrate the use of the program the graphical user interface example is shown in Figure 6.3 and Figure 6.4 below for beam with vertical and inclined stirrup respectively.

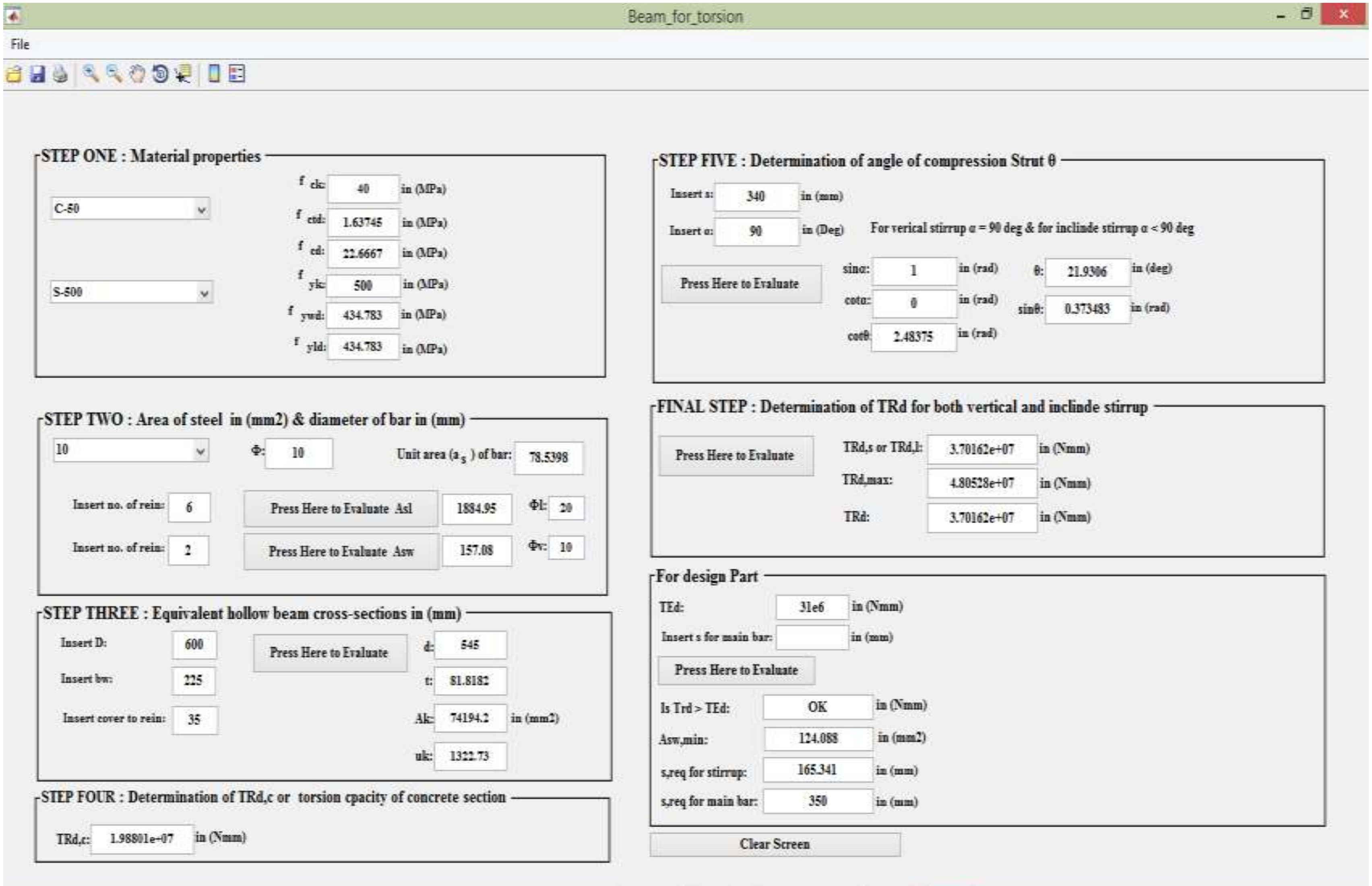


Figure 6.2: Sample design and estimation of torsion resistance of rectangular reinforced concrete beam section with vertical stirrup

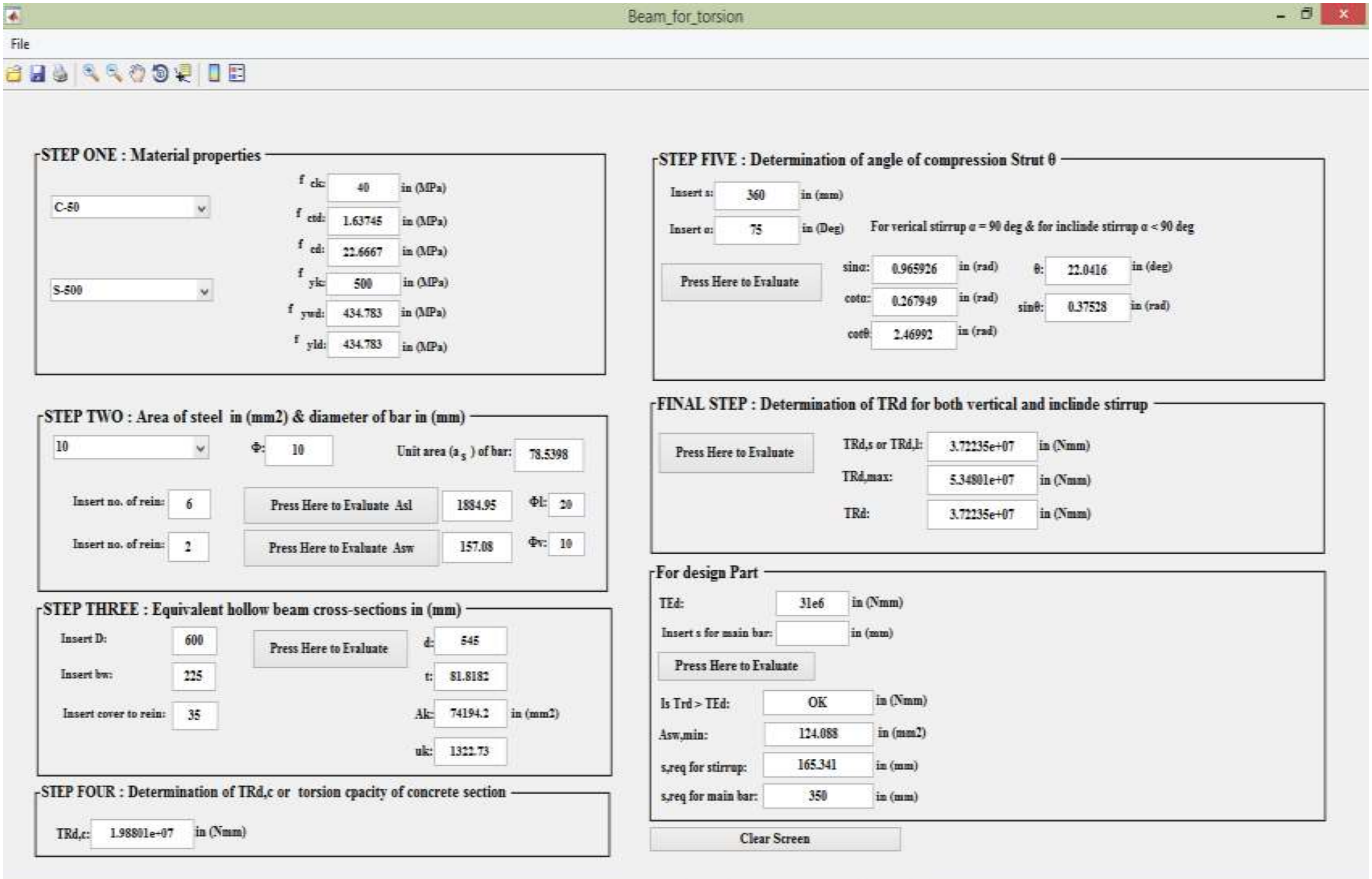


Figure 6.3: Sample design and estimation of torsion resistnace of rectangular reinforced beam section with inclined stirrup

6.3 Design Example for Combined Shear and Torsion

One application of this study is demonstrated with analysis and design example of rectangular beam section with vertical and inclined stirrup subjected to a combined design shear force and design torsional moment.

6.3.1 Steps followed in the Graphical User Interface Window

1. Select concrete grade and steel grade from the pop-up menu.
2. Select the size of bar from pop-up menu and input number of bar in the input box to evaluate A_{sl} and A_{sw} by pressing the push button.
3. All the cross-sectional parameters are inserted in the input box and then effective depth (d), t, u_k and A_k evaluated by pressing the respective push button.
4. Evaluate the following parameter κ and ρ_l which used for the determination of concrete shear capacity by pressing the push button and press the second pushbutton evaluate the concrete torsion capacity.
5. Insert s, α, cotθ in the input box and evaluate parameter sinα,cotα,sinθ and z by pressing the push button.
6. Evaluate $V_{Rd,max}$ and $T_{Rd,max}$ using push button and insert value of V_{Ed} and T_{Ed} in the input box.

Final step check for the combined effect is satisfied or not. If the deign criteria is not satisfied iterate the procedure from step one to the final step to obtain acceptable cross-section dimensions and area of steel. To illustrate the use of the program the graphical user interface example is shown in Figure 6.4 and Figure 6.5 below for beam with vertical and inclined stirrup respectively.

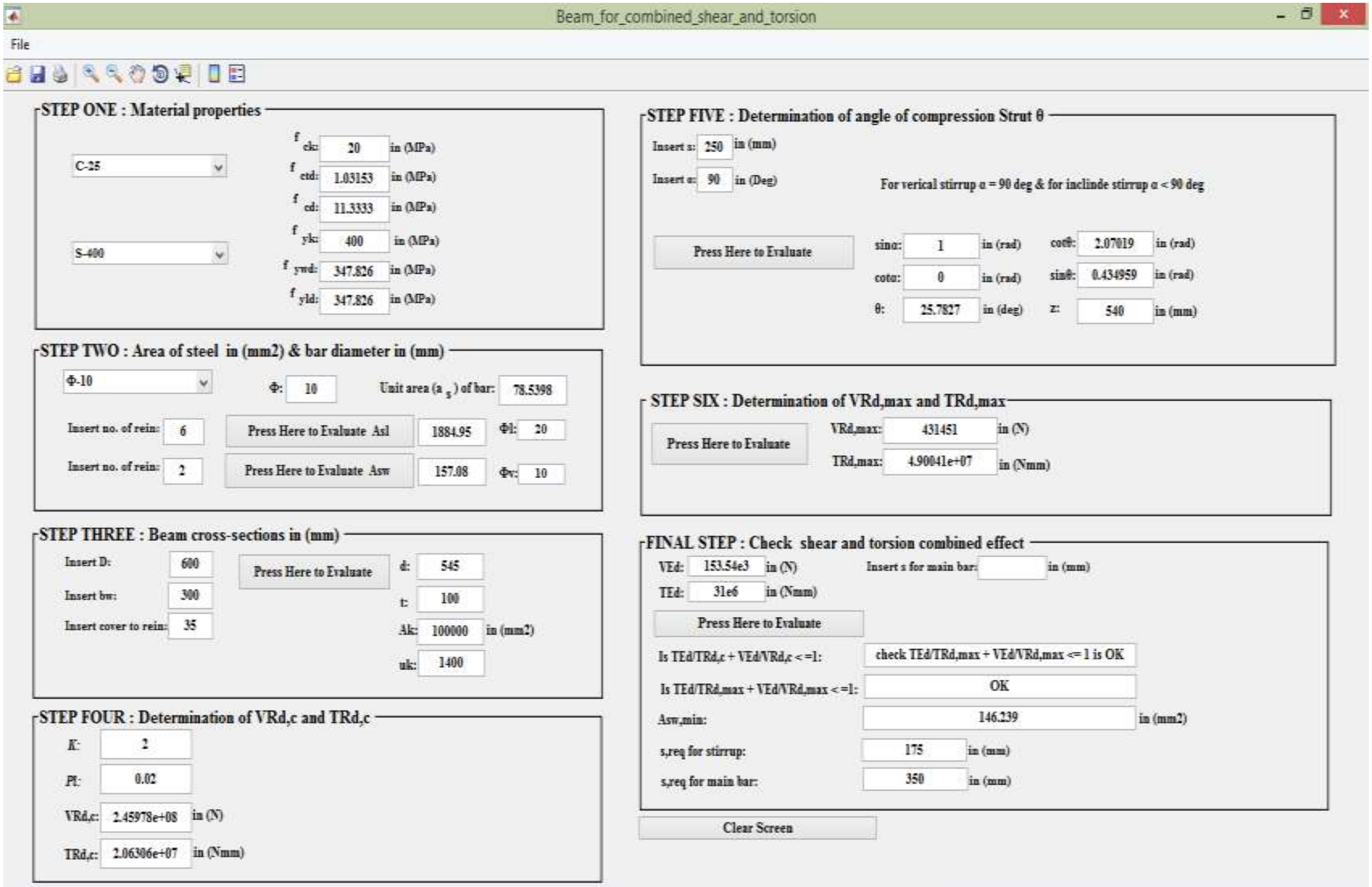


Figure 6.4: Sample design and checking of section resistance for shear combined with torsion for rectangular reinforced concrete beam section with vertical stirrup

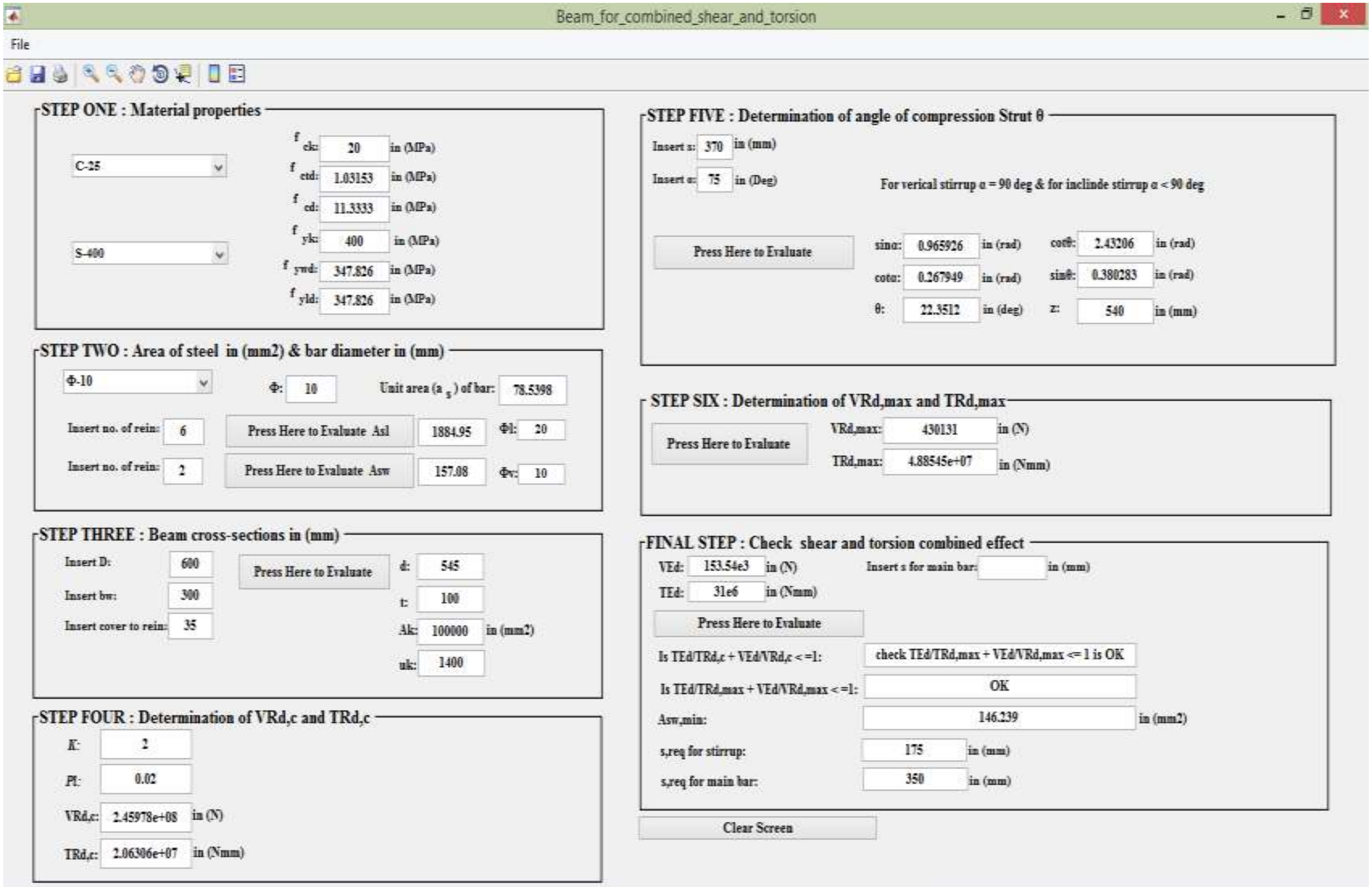
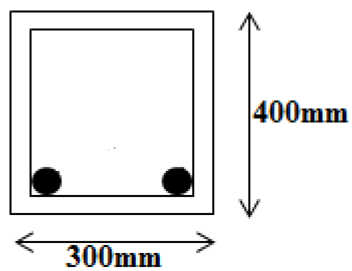


Figure 6.5: Sample design and checking of section resistance for shear combined with torsion for rectangular reinforced concrete beam section with inclined stirrup

7. NUMERICAL ANALYSIS AND DESIGN OF REINFORCED CONCRETE BEAM SECTION FOR TESTING OF THE ANALYSIS AND DESIGN INTERFACE PROGRAM

1: Determine the design flexural strength of the beam cross-section shown below. Use concrete class C20/25 and steel grade S-400 and assuming cover =25mm, stirrup ϕ_8 and 2 ϕ_{10} for longitudinal bar.



Step 1: Design Values (Changing the characteristic value to design value)

d= 400 - 25 - 8 – 5 = 362mm

$$f_{yd} = f_{yk} / \gamma_s$$

$$f_{cd} = (\alpha_{cc} f_{ck}) / \gamma_c$$

For persistent and transient design situation:

- γc = 1.50
- γs = 1.15

α_{cc} =0.85

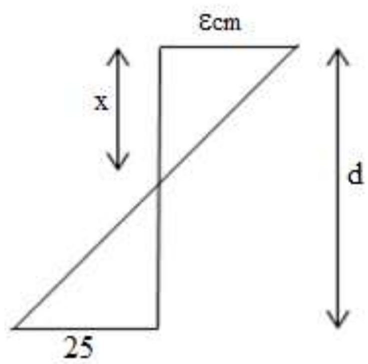
$$f_{yd} = 400/1.15 = 347.83\text{MPa}$$

$$f_{cd} = (0.85x20)/1.5 = 11.33\text{MPa}$$

Step 2: Assume the type of failure

Assume tension failure with rupture of steel and $\varepsilon_{cm} < \varepsilon_{c2} = 2 ‰$

Step 3: Draw the strain profile corresponding to the type of failure and use the similarity of triangles to develop a relationship between the unknown strain and the neutral axis.



From Similarity of Triangle

$$\frac{\varepsilon_{cm} + 25}{d} = \frac{\varepsilon_{cm}}{x} \rightarrow \frac{x}{d} = k_x = \frac{\varepsilon_{cm}}{\varepsilon_{cm} + 25} \quad (1)$$

Step 4: Use the equation of alpha corresponding to the assumption in step 2 and the relationship developed in step 3 to calculate the unknown strain.

From equilibrium of forces,

$C_c = T_s$ but $C_c = \alpha_c f_{cd} bd$ and $T_s = A_s f_{yd}$

$$\alpha_c f_{cd} bd = A_s f_{yd}$$

$$\alpha_c = \frac{A_s f_{yd}}{f_{cd} bd} = \frac{(2x\pi x5^2)x347.83}{11.33x300x362} = 0.0444$$

(2)

For $\varepsilon_{cm} < \varepsilon_{c2}$, $\alpha_c = \varepsilon_{cm}\left[\frac{6 - \varepsilon_{cm}}{12}\right]k_x$

Substituting k_X from equation (1),

$$\alpha_c = \varepsilon_{cm}\left[\frac{6-\varepsilon_{cm}}{12}\right]\left(\frac{\varepsilon_{cm}}{\varepsilon_{cm}+25}\right)$$

$$\alpha_c = \left(\frac{6\varepsilon_{cm}^{\ 2} - \varepsilon_{cm}^{\ 3}}{12\varepsilon_{cm}+300}\right)$$

$$12\varepsilon_{cm} + 300\alpha_c = 6\varepsilon_{cm}^{\ 2} - \varepsilon_{cm}^{\ 3}$$

$$-\varepsilon_{cm}^{\ 3} + 6\varepsilon_{cm}^{\ 2} - 12x0.0444x\varepsilon_{cm} - 300x0.0444 = 0$$

$$-\varepsilon_{cm}^{\ 3} + 6\varepsilon_{cm}^{\ 2} - 0.5328\varepsilon_{cm} - 13.32 = 0$$

Solving the cubic equation results three possible answers

ε_{cm1} = 5.4546 > 3.5 not ok

ε_{cm2} = -1.3136 <.0 not ok

ε_{cm3} = 1.859 < 3.5 ok

Thus, ε_{cm} = 1.859‰

Step 5: Check if the assumption in step 2 is correct and if it is, proceed to step 8. If the assumption is not correct, repeat step 2 to 5 with another assumption.

ε_{cm} = 1.859 < ε_{c2} both of the assumptions are correct

Step 6: Calculate the value of beta

For $\varepsilon_{cm} < \varepsilon_{c2}$, $\beta_c = k_x\left[\frac{8-\varepsilon_{cm}}{4(6-\varepsilon_{cm})}\right]$

$k_x = \frac{\varepsilon_{cm}}{\varepsilon_{cm}+25} = \frac{1.859}{1.859+25}$ and x=25.0504 mm

Substituting the values of k_x and ε_{cm} yields, $\beta_C = 0.025$

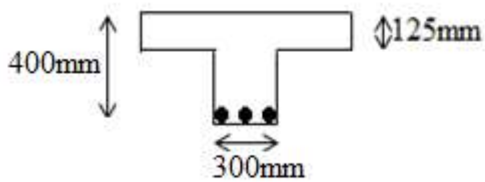
Step 7: Calculate the moment resistance

$$M = A_s f_{yd}(1-\beta_c)$$

$$M = 2x5^2 x347.83x(1-0.0257)$$

M = 19.27 *kNm*

2: Design the beam Using concrete class C20/25 and steel grade S-400 and assuming cover =25mm, and M_{sd} = -19 kNm.



Step 1: Design Values (Changing the characteristic value to design value)

d= 400 - 25 - 8 – 5 = 362mm

$$f_{yd} = 400/1.15 = 347.83\text{MPa}$$

$$f_{cd} = (0.85x20)/1.5 = 11.33\text{MPa}$$

Step 2: for negative moment, slab is in tension side and b_w is the width

Step 3: Calculate μ_{sd} and k_z

$$\mu_{sd} = \frac{M_{sd}}{f_{cd}bd^2} = \frac{19000000}{11.33x300x362^2} = 0.0426$$

$\mu_{sd} = 0.0426 < \mu_{sd}^* = 0.295$ it has sufficient ductility and single reinforced is enough

From chart

$K_z = 0.98$

$K_z = z/d$

$z = dxk_z = 0.98x362 = 354.76mm$

Step 4: Calculate A_{st}

$$A_{st} = \frac{M_{sd}}{f_{yd}z} = \frac{19000000}{347.83x354.76mm} = 157.975mm^2$$

Step 5: Check $A_{s,max}$ and $A_{s,min}$

$$A_{s,\min} = \max\begin{cases} 0.26\frac{f_{ctm}}{f_{yk}}b_t d \\ 0.0013b_t d \end{cases}$$

$$\max\begin{cases} 0.26x\frac{2.21x}{400}300x362 = 156.004mm^2 \\ 0.0013x300x362 = 141.18mm^2 \end{cases}$$

$A_{s,min} = 156mm^2$

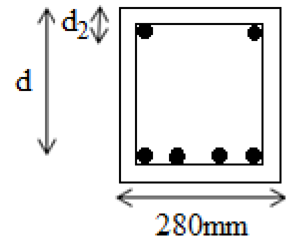
$A_{s,max} = 0.04A_c$

$=0.04x300x400$

$A_{s,max} = 4800mm^2$

Thefore $A_{s,req}$ is $157.975mm^2$

3: Determine the moment resistance of the doubly reinforced section shown in the figure below. d = 500 mm, d_2 = 50 mm $A_{s1} = 2581mm^2$ and $A_{s2} = 645mm^2$, C20/25 and S300.



Step 1: Design Values (Changing the characteristic value to design value)

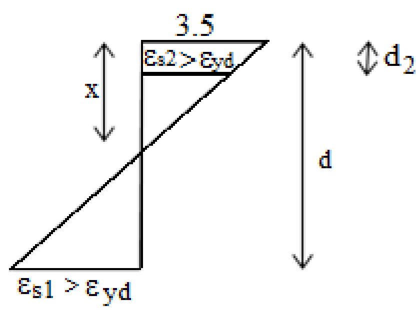
$f_{yd} = {}^{300}\!/_{1.15}$ =260.87MPa

$f_{cd} = {}^{(0.85x20)}\!/_{1.5}$ =11.33MPa

Step 2: Assume the type of failure

Assume tension failure with crushing of concrete and all of the provided steel has yielded.

Step 3: Draw the strain profile corresponding to the type of failure and use the similarity of triangles to develop a relationship between the unknown strain and the neutral axis.



From Similarity of Triangle

$$\frac{3.5+\varepsilon_{s1}}{d} = \frac{3.5}{x} \rightarrow \frac{x}{d} = k_x = \frac{3.5}{3.5+\varepsilon_{s1}} \tag{1}$$

$$\frac{3.5}{x} = \frac{\varepsilon_{s2}}{x-50} \rightarrow \varepsilon_{s2} = \frac{3.5(x-50)}{x} \tag{2}$$

Step 4: Use the equation of alpha corresponding to the assumption in step 2 and the relationship developed in step 3 to calculate the unknown strain.

From equilibrium of forces,

$C_c = T_s$ but $C_c = \alpha_c f_{cd} bd + A_{s2} f_{yd}$ and $T_s = A_{s1} f_{yd}$

$$\alpha_c f_{cd} bd + A_{s2} f_{yd} = A_{s1} f_{yd}$$

$$\alpha_c = \left(\frac{A_{s1} f_{yd} - A_{s2} f_{yd}}{f_{cd} bd} \right)$$

$$\alpha_c = \left(\frac{2851 - 645}{11.33x280x500} \right) x260.87 = 0.318 \qquad (3)$$

For $\varepsilon_{cm} > \varepsilon_{c2}$ $\alpha_c = k_x \left(\frac{3\varepsilon_{cm} - 2}{3\varepsilon_{cm}} \right)$

Substituting k_X from equation (1),

$$\alpha_c = \left(\frac{3.5}{3.5 + \varepsilon_{s1}} \right) \left(\frac{3\varepsilon_{cm} - 2}{3\varepsilon_{cm}} \right)$$

$$\alpha_c = \left(\frac{3.5}{3.5 + \varepsilon_{s1}} \right) \left(\frac{3x3.5 - 2}{3x3.5} \right)$$

$$\alpha_c = \left(\frac{8.5}{10.5 + 3\varepsilon_{s1}} \right)$$

$$10.5x\alpha_c + 3\varepsilon_{s1}\alpha_c = 8.5$$

$$\varepsilon_{s1} = \left(\frac{8.5 - 10.5x\alpha_c}{3\alpha_c} \right) = \left(\frac{8.5 - 10.5x0.318}{3x0.318} \right) = 5.41$$

Thus, $\varepsilon_{s1} = 5.41‰$

$k_x = \frac{3.5}{3.5 + \varepsilon_{s1}} = \; = \frac{3.5}{3.5 + 5.41} = 0.3928$ and x = 196.41mm

$$\varepsilon_{s2} = 3.5x\left(\frac{x-50}{x}\right) = 3.5x\left(\frac{196.41-50}{196.41}\right) = 2.6090 \text{‰}$$

Step 5: Check if the assumption in step 2 is correct and if it is, proceed to step 8. If the assumption is not correct, repeat step 2 to 5 with another assumption.

$$\varepsilon_{yd} = \frac{260.87}{200x10^3} = 1.304 \text{‰}$$

$$\varepsilon_{s1} = 5.41 \text{‰} > \varepsilon_{yd}$$

$$\varepsilon_{s2} = 2.609 \text{‰} > \varepsilon_{yd}$$

Both of the assumptions are correct

Step 6: Calculate the value of beta

For $\varepsilon_{cm} > \varepsilon_{c2}$, $\beta_c = k_x \left[\frac{\varepsilon_{cm}(3\varepsilon_{cm}-4)+2}{2\varepsilon_{cm}(3\varepsilon_{cm}-2)}\right]$

Substituting the values of k_x and ε_{cm} yields, $\beta_c = 0.1634$

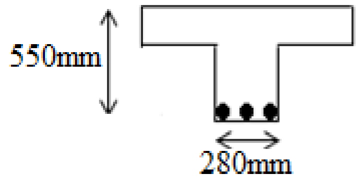
Step 7: Calculate the moment resistance

$$M = \alpha_c f_{cd} bd^2 (1-\beta_c) + A_{s2} f_{yd} (d-d_2)$$

$$M = 0.318x11.33x280x500^2 (1-0.1634) + 645x260.87(500-50)$$

M = 286.712 kNm

4: Design the beam Using concrete class C20/25 and steel grade S-300 and assuming cover =25mm, d = 500mm, d_2 = 50mm and M_{sd} = -286 kNm.



Step 1: Design Values (Changing the characteristic value to design value)

d= 500mm

$f_{yd} = 300/1.15$ =260.87MPa

$f_{cd} = (0.85x20)/1.5$ =11.33MPa

Step 2: for negative moment, slab is in tension side and b_w is the width

Step 3: Calculate μ_{sd} and k_z

$$\mu_{sd} = \frac{M_{sd}}{f_{cd}bd^2} = \frac{286000000}{11.33x280x500^2} = 0.3606$$

$\mu_{sd} = 0.3606 > \mu_{sd}^* = 0.295$ it require compresson reinforcement for sufficient ductility

From chart

K_z = 0.744

K_z = z/d

z = dxk_z = 0.744x500 = 372mm

$$M_{Rd} = 0.295xf_{cd}bd^2 = 0.295x11.33x280x500^2 = 233.9645kNm$$

Step 4: Calculate A_{stot}

$$A_{sc} = \frac{M_{sd} - M_{Rd}}{f_{yd}(d - d_2)} = \frac{286000000 - 233964500}{260.87x(500 - 50)} = 443.268mm^2$$

$$A_{st} = \frac{M_{sd}}{f_{yd} z} = \frac{233964500}{260.87x372mm} = 2410.92mm^2$$

$A_{s,tot}$ = 443.268mm^2 + 2410.92mm^2 = 2854.18mm^2

Step 5: Check $A_{s,max}$ and $A_{s,min}$

$$A_{s,\min} = \max\begin{cases} 0.26\frac{f_{ctm}}{f_{yk}} b_t d \\ 0.0013 b_t d \end{cases}$$

$$\max\begin{cases} 0.26x\frac{2.21x}{300} 280x500 = 268.146mm^2 \\ 0.0013x280x500 = 182mm^2 \end{cases}$$

$A_{s,min}$ = 268.146mm^2

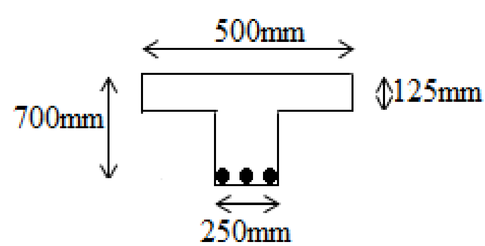
$A_{s,max}$ = 0.04A_c

=0.04x280x550

$A_{s,max}$ = 6160mm^2

Thefore $A_{s,req}$ is 2854.18mm^2

5: Compute the design resistance of the T – beam shown in the figure below. The concrete and steel grades are C20/25 and S460 respectively. (Cover to stirrup = 25 mm, stirrup of diameter 8 c/c 200 and 4 ϕ_{24} for longitudinal bar was provided)



Step 1: Design Values (Changing the characteristic value to design value)

d= 700 – (25 + 8 + 12) = 655mm

$f_{yd} = 460/1.15$ =400MPa

$f_{cd} = (0.85x20)/1.5$ =11.33MPa

Step 2: Assume the neutral axis to be in the flange

Step 3: Assume the strain the tension reinforcement to be greater than the yield strain

Step 4: Use the procedure of analysis of singly reinforced concrete sections to estimate neutral axis depth

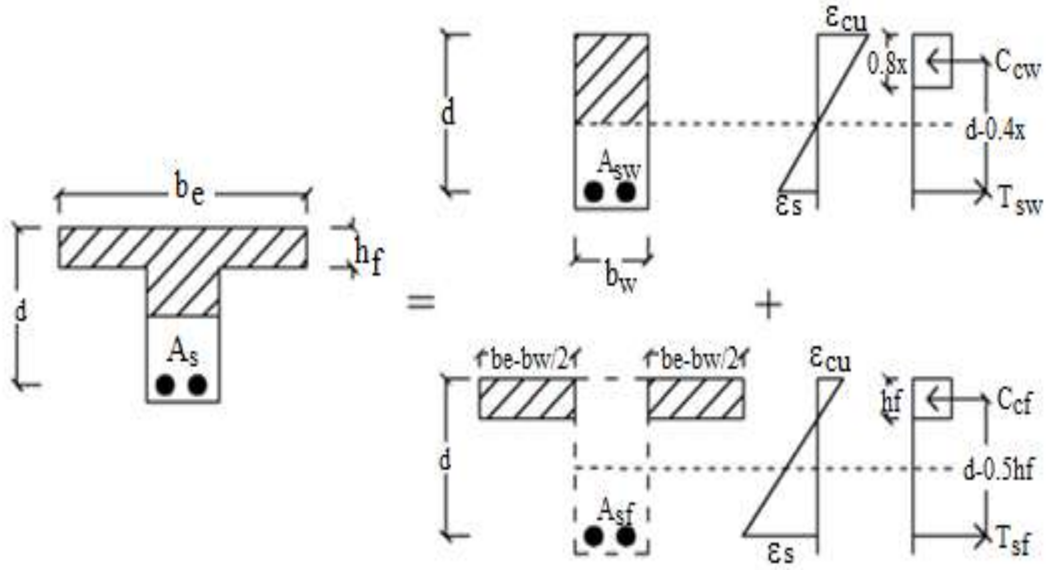
$$\alpha_c = \frac{A_s f_{yd}}{f_{cd} bd} = \frac{4x\pi x12^2 x400}{11.33x500x655} = 0.195$$

Using $\alpha_c = 0.8k_x$, k_x = 0.2437 and using k_x = x/d, x = 159.656 mm

Step 5: Check if the assumption in step 3 is correct

X = 159.656 > 125, the assumption is not correct

Step 6: Take the neutral axis to be below the flange and divide the section into two parts: Beam W and Beam F to simplify the analysis process



Step 7: Take the rectangular stress strain relationship for the concrete under compression and calculate the moment resistance using force equilibrium.

Beam F

$$A_{sf} = \frac{f_{cd}(b_e - b_w)h_f}{f_{yd}} = \frac{11.33x(500-250)x125}{400} = 885.16mm^2$$

$$M_{Rd,f} = A_{sf}\, f_{yd}\left(d - {}^{h_f}\!/_{2}\right) \text{ or } M_{Rd,f} = f_{cd}(b_e - b_w)h_f\left(d - {}^{h_f}\!/_{2}\right)$$

$$M_{Rd,f} = 11.33x(500-250)x125x\left(655 - {}^{125}\!/_{2}\right) = 209.78kNm$$

The force in the remaining steel area A_{sw} is balanced by compression in the rectangular portion of the beam. (i.e. $A_{sw} = A_s - A_{sf}$)

Beam W

$$Asw = As - Asf = 4x\pi x12^2 - 885.16 = 924.4mm^2$$

$$A_{sw} f_{yd} = 0.8x f_{cd} b_w$$

$$x = \frac{A_{sw} f_{yd}}{0.8 f_{cd} b_w}$$

$$x = \frac{924.4x400}{0.8x11.33x250} = 163.18mm$$

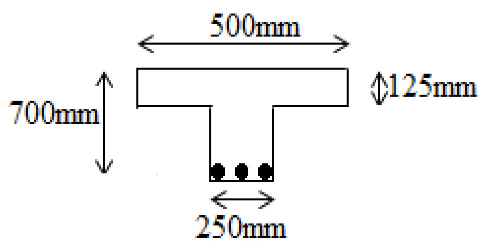
$$M_{Rd,w} = A_{sw} f_{yd}(d - 0.4x) \text{ or } M_{Rd,w} = f_{cd} b_w (0.8x)(d - 0.4x)$$

$$M_{Rd,w} = 11.33x250x0.8x163.18x(655 - 0.4x163.18) = 218.06kNm$$

The total moment capacity of the section now becomes,

M_{Rd} = 209.78 +218.06 = 427.84 *kNm*

6: Design the T – beam shown in the figure below given Msd =427kNm. use concrete and steel grades are C20/25 and S460 respectively.



Step 1: Design Values (Changing the characteristic value to design value)

d= 700 – (25 + 8 + 12) = 655mm

$f_{yd} = 460/1.15$ =400MPa

$f_{cd} = (0.85x20)/1.5$ =11.33MPa

Step 2: Assume the neutral axis to be in the flange

Step 3: Assume the strain the tension reinforcement to be greater than the yield strain

Step 4: Use the procedure of analysis of singly reinforced concrete sections to estimate neutral axis depth

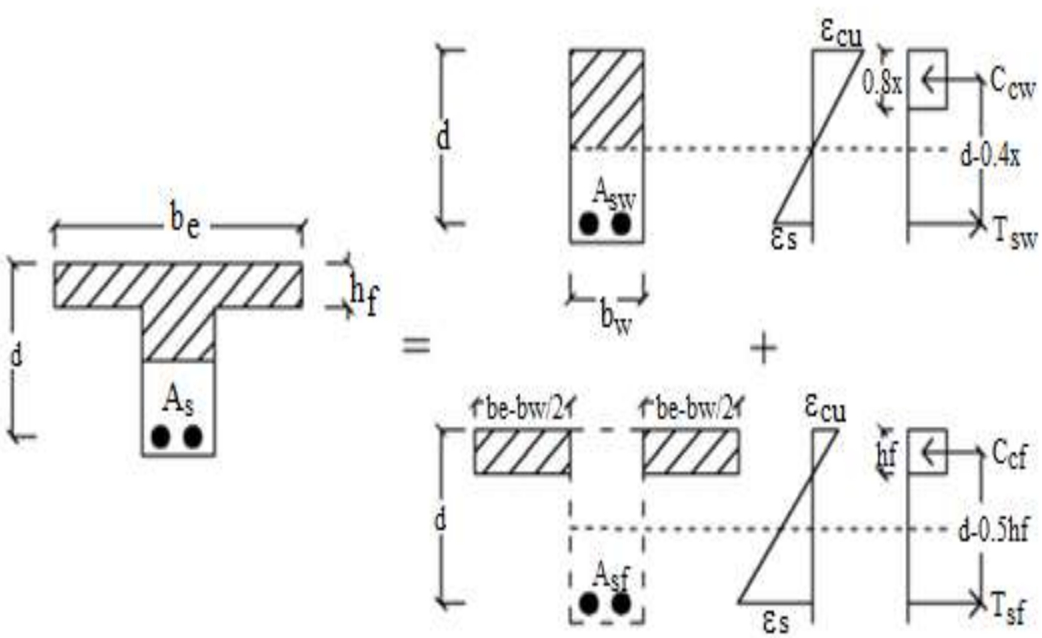
$$\alpha_c = \frac{A_s f_{yd}}{f_{cd} bd} = \frac{4x\pi x12^2 x400}{11.33x500x655} = 0.195$$

Using $\alpha_c = 0.8k_x$, k_x = 0.2437 and using k_x = x/d, x = 159.656 mm

Step 5: Check if the assumption in step 3 is correct

X = 159.656 > 125, the assumption is not correct

Step 6: Take the neutral axis to be below the flange and divide the section into two parts: Beam W and Beam F to simplify the analysis process



Step 7: Calculate the area of steel for each beam part.

Beam F

$$A_{sf} = \frac{f_{cd}(b_e - b_w)h_f}{f_{yd}} = \frac{11.33x(500-250)x125}{400} = 885.16mm^2$$

$$M_{Rd,f} = 11.33x(500-250)x125x\left(655 - 125/2\right) = 209.78kNm$$

Beam W

$$A_{sw} = \frac{M_{Rd,w}}{f_{yd}(d - 0.4x)} = \frac{427 - 209.78}{400(655 - 0.4x159.656)} = 918.652mm^2$$

The total area of steel therfore,

A_{stot}= 885.16mm^2 +918.652mm^2 = 1803.812mm^2

Step 8: Check $A_{s,max}$ and $A_{s,min}$

$$A_{s,\min} = \max\begin{cases} 0.26\dfrac{f_{ctm}}{f_{yk}}b_t d \\ 0.0013b_t d \end{cases}$$

$$\max\begin{cases} 0.26x\dfrac{2.21x}{400}250x655 = 235.226mm^2 \\ 0.0013x250x500 = 212.875mm^2 \end{cases}$$

$A_{s,min}$ = 235.226 mm^2

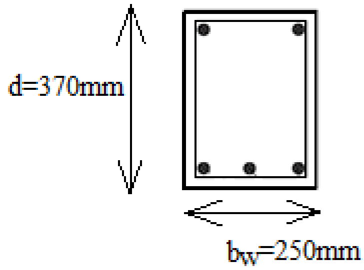
$A_{s,max}$ = 0.04A_c +A_{sf}

=0.04x250x700 + 885.16

$A_{s,max}$ = 7885.16mm^2

Thefore $A_{s,req}$ is 1803.812mm^2

8: Design the beam shown below for shear. Use C20/25 concrete and S400 steel. Assuming A_{st} = 1200mm^2



Step 1: Design Values (Changing the characteristic value to design value)

$$f_{yd} = f_{yk}/\gamma_s$$

$$f_{cd} = (\alpha_{cc} f_{ck})/\gamma_c$$

$$f_{yd} = 400/1.15 = 347.83\text{MPa}$$

$$f_{cd} = (0.85x20)/1.5 = 11.33\text{MPa}$$

Step 2: Calculate design value of shear

Crushing of compression struts should be checked at the face of the support and diagonal tension should be checked at a distance, d, from the face of the column

$V_{Ed,max}$ = 175.14kN and V_{Ed} = 153.54kN from analysis and similarity of triangle

Step 3: Check compression failure of diagonal struts

- Determine $V_{Rd,max}$ with $\cot\theta = 2.5$

$$V_{Rd,max} = \alpha_{cw} b_w z \upsilon f_{cd} / (\cot\theta + \tan\theta)$$

$\alpha_c = 1$

b_w = 250mm

z = 0.9d

ν = 0.6

$$V_{Rd,\max} = \left((1x250x0.9x370x0.6x11.33) \Big/ \left(2.5 + 1/2.5\right) \right) = 195.15kN > V_{Ed,\max}$$

Step 4: Check diagonal tension failure

- Calculate the shear resistance contribution of the concrete

$$V_{Rd,c} = \max \begin{cases} \left[C_{Rd,c} k (100 \rho_l f_{ck})^{1/3} + k_1 \sigma_{cp}\right] b_w d \\ (\nu_{\min} + k_1 \sigma_{cp}) b_w d \end{cases}$$

$C_{Rd,c} = 0.18/\gamma_c = 0.18/1.5 = 0.12$

$k_1 = 0.15$

$$k = 1 + \sqrt{200/d} \leq 2.0 = 1 + \sqrt{200/370} = 1.375 \leq 2.0$$

$\nu_{min} = 0.035k^{3/2}\text{-}f^{1/2}_{ck} = 0.035x\ 1.375^{3/2}x20^{1/2} = 0.358$

$$\rho_l = \frac{A_{sl}}{b_w d} \leq 0.02 = \frac{1200}{250x370} = 0.013$$

$\sigma_{cp} = 0$

$$V_{Rd,c} = \max \begin{cases} \left[0.12x1.735(100x0.013x20)^{1/3} + 0.15x0\right] 250x370 = 57.05 \\ (0.358 + 0.15x0) 250x370 = 33.09 \end{cases}$$

$V_{Rd,c}$ = 57.05kN < V_{Ed} web reinforcement should be provided

Step 5: Calculate the necessary amount of shear reinforcement

$$V_{Rd,s} = \frac{A_{sw}}{s} z f_{ywd} \cot\theta \rightarrow s = \frac{A_{sw}}{v_{Rd,s}} z f_{ywd} \cot\theta$$

Take a single rectangular loop of diameter 8 stirrup,

$$s = \frac{2x\pi x4^2 x0.9x370x347.83x2.5}{153.54x1000} = 189.6mm$$

Use ϕ_8 c/c 180mm

9: Calculate the area of steel required in the web of the edge beam. Use C40/50 and S-500 bars for stirrup and longitudinal reinforcement.Assume cover to longitudinal bars to be 35mm and cotθ=2.5.

Solution:

Step1: Summarize the given parameters

Material C40/50

f_{ck}=40MPa; f_{cd}=22.66MPa; f_{ctk},0.05=2.5MPa; f_{ctd}=1.4MPa E_{cm}=35,000MPa

S-500 f_{yk}=500MPa; f_{yd}=434.78MPa; Es=200,000MPa; ε_y=2.17‰

Action T_{Ed}=31 kNm

Step2: Compute the thickness of the equivalent thin-walled section for the web.

Thickness= area/perimeter

t=A/u

Therefore,

t=600x225/1650= 81mm

Minimum is twice the cover to the longitudinal bars 2x35= 70mm

Therefore, t=81mm

Step3: Compute the maximum torsional moment that can be resisted by the compressive struts.

$$T_{Rd,\max} = 2v\alpha_{CW} f_{Cd} A_k t_{ef,i} (\cot\theta + \cos\alpha)\sin^2\theta$$

$$\nu = 0.6\left[1 - \frac{f_{ck}}{250}\right]$$

α_{cw}=1

$$\nu = 0.6\left[1 - \frac{40}{250}\right] = 0.5$$

t = 81mm

A_k = (600-81)x(225-81) = 74736mm^2

f_{cd} = 0.85x40/1.15 = 22.67MPa

$T_{Rd,max}$ =2x74736x81x0.5x22.67x0.3714x0.9285 = 47.7kNm

This exceeds T_{Ed} (31kNm), therefore section is ok

Step4: Compute the area of longitudinal reinforcement for torsion required to satisfy the design torsional moment, T_{Ed}

$$A_{sl} = \frac{T_{Ed} u_k \cot\theta}{2A_k f_{yd}}$$

A_{sl} = 31 x 10^6 x 1326 x 3 / (2 x 74736 x 500/1.15) = 1583mm^2

Step5: Compute the minimum amount of link reinforcement required to resist the design torsional moment.

Minimum area of torsional link reinforcement is:

$$\frac{A_{sw}}{s} = \frac{T_{Ed}}{2A_k f_{ydw} \cot\theta}$$

$A_{sw/s}$ = 31 x10^6 / (2 x 74736x500/1.15 x 2.5)

= 0.19 mm^2/m

Since the program developed based on numerical analysis and design problem the result obtained is exact for section analysis and approximately exact for design problem.

From the above numerical analysis and design problem shown on chapter seven (from example:1 upto example:9) one can conclude that the program give the exact output for section analysis and design of single and double reinforced concrete rectangular beam, T beam subjected to design action suchas:

- Design moment
- Design shear force
- Design torsional moment

8. CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

- The developed graphical user interface gives satisfactory section analysis and design outputs for rectangular, T or Inverted L and Trapezoidal beam. The analysis and design outputs are solely based on ES EN 1992-1-1:2015.
- The developed graphical user interface application can be used without installing MATLAB program.
- The program is developed independently for rectangular, T or inverted L and trapezoidal reinforced concrete beam sections by considering design bending moment.
- The program is also developed independently for rectangular beam with vertical and inclined stirrup by considering effects of shear, torsion and combined effect of shear with torsion.
- The section analysis and design commences from a particular strain distribution in the ULS. The possible strain distributions in the ULS were adopted from ES EN 1992-1-1:2015. Having a particular strain distribution in the ULS, the stress in the materials was found using the stress-strain relationships. For the rectangular reinforced concrete beam ,a parabolic rectangular stress-strain relation was taken. For T or inverted L and trapezoidal beam equivalent rectangular stress-strain relation was taken according ES EN 1992-1-1:2015.
- It can be observed that, the time required for manual design is much greater than time required for using the application which gives the results within short time.
- The steps stated on the graphical user interface window need to be followed accordingly to arrive at the output needed.
- The dimensions to be used for every parameter should be consistent thus, it is necessary to strictly follow the dimensions specified on the user interface window.
- Additionally, the graphical user interface is particularly useful in the analysis and design of beam as instruction tool for instructors to save their time.

- It can be said that researches carried out for finding optimum design of concrete structures are of great value to practicing engineers. The optimum solution need to satisfies the provisions of the code and minimizes the cost of the structure.
- Genetic algorithm based design of simple span beam gave reasonable results, satisfying the design constraints. This method has the advantage that the cost of concrete and steel can be incorporated into the design. This will help in obtaining reasonable sections and steel based on the cost.
- The optimization program also help the user to update the cost of reinforcement and cost of concrete as the market price varies through time.

8.2 Recommendations

A computer program for the analysis and design of non-prismatic reinforced concrete beam sections subjected to flexure, shear and torsion can be developed further, other beam cross-sections shape can be considered other than cross-sections that considered in this thesis. Furthermore, the structural analysis for simple and multiple span beam which is prismatic and non prismatic subjectd to different load condition can also be integrated in the graphical user interface program.

Furthermore, the cost optimization for multiple span beam which is subjectd to different load condition can also be integrated in the computer program.

REFERENCES

Alex , D. M., & Kottalil , D. (2015). Genetic Algorithm based Design of a Reinforced Concrete Continuous Beam. *International Journal of Engineering Research & Technology* , 224-227.

Bhalchandra, S. A., & Adsul , P. K. (2012). Cost Optimization Of Doubly Reinforced Rectangular Beam Section. *International Journal of Modern Engineering Research* , 3939-3942.

Singh, B., & Rai, H. S. (2014). Optimisation of RCC Beam. *International Journal of Engineering, Business and Enterprise Applications*, 20-39.

Awari, U. R., Gohdke, R. B., & Waghmare, M. V. (2016). Torsional Effect for Unsymmetrical R.C.Frames. *International research journal of engineering and technology, 03*(07), 812-818.

Bama, S. S., & Priya, K. B. (2014). Matlab Program For Design Of Rc Elements. *International Journal of Science and Engineering Research, 2*(12).

Dahake, A. G., Kakade, D. N., & Nirkha, S. P. (2016). Design of Flat Slab with MATLAB. *International Journal of Engineering Research, 5*(3), 537-540.

Darwin, D., Dolan, C. W., & Nilson, A. H. (2004). *Design of Concrete Structures.* Kansas and Wyoming University: Thirteenth Edition. The Mc Graw-hill companies.

Das, K. N., & Navaee, S. (2002). Utilization of MATLAB in Structural Analysis. Georgia Southern University.

Dewangan, U. K., & Mandal, R. R. (2017). Finite Element Modeling of Beam with Eight Noded Brick Element Using Matlab. *International Journal of Civil Engineering and Technology, 5*(8), 646–656.

EN, E. (1992-1-1:2015). *Design of Concrete Structures.* Addis Ababa: Ministry of construction.

Gajendra, A., Pamnani, G., Rajput, D. S., & Tiwari, N. (2014). Beam Analysis in Matlab. *1*(2), 27-36.

Hadi, M. (2001). Optimum Design of Reinforced Concrete Continuous Beams by Genetic Algorithms. *Proceedings of the Eighth International Conference on Civil and Structural Engineering Computing* (p. Paper 57). University of Wollongong, Australia: Civil-Comp Press, Stirlingshire, UK.

Harle, S. M. (2017). Analysis by STAAD-PRO and Design of Structural Element by. *Journal of Asian Scientific Research, 7*(2), 145-164.

Houcque, D. (2005). *Introduction to MATLAB for Engineering Students.* Illinois: McCormick.

Htwe, T., & Khaing, S. Y. (2014). Analysis of Beam Structure with Matlab Software. *International Journal of Scientific Engineering and Technology Research, Vol.03*,(Issue.10), Pages:2064-2069.

Kalechman, M. (2009). *Practical MATLAB® Basics for Engineers.* City University of New York: by Taylor & Francis Group, LLC.

Macgregor, J. G., & Wight, J. K. (2012). *Reinforced Concrete Mechanics and Design.* New Jersey: sixth edition.

Moler, C. (1996). *MATLAB The Language of Technical Computing, Computation and Visualization ProgrammingMATLAB* (first ed.). University of New Mexico: COPYRIGHT to The Math Works, Inc.

Najem , M. R., & Yousif , T. S. (2013). Optimum cost design of reinforced concrete continuous beams. *Int. Journal of Applied Sciences and Engineering Research*, 79-92.

Saglar, M. A. (2009). *Development of a Finite Element Analysis Program for Structural Dynamics Applications.* Istanbul Technical University.Istanbul: ProQuest LLC.

Samuel, M. (2003). *Reinforced concrete I Lecture notes.* Civil Engineering Department Faculty of Technology, Addis Ababa University: AAU.

Silver, D. H. (2013). Creating Graphical User Interfaces with MATLAB. *Mid-Atlantic Section Conference of the American Society of Engineering Education*, (pp. 189-221). Fairleigh Dickinson University.

Urgessa, G. S. (2011). Incorporating MATLAB Modules in Reinforced Concrete Design Instruction. 1-7.

Urroz, G. E. (2002). *Programming with MATLAB.*

Venkatesh, B. B., & Vivek, M. B. (2015). Developing Graphical User Interface for Analysis of Beam by Using MATLAB Software. *International Journal of Research in Engineering and Technology, 04*(04), 676-679.

wikipedia, M. (2017, July 26). *MATLAB*. Retrieved July 26, 2017, from MATLAB wikipedia: https://en.wikipedia.org/wiki/MATLAB

APPENDIX

1. Source Code for Main Interface

```
function varargout =
Main_interface_window(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
'gui_Singleton',  gui_Singleton,
...               'gui_OpeningFcn',
@Main_interface_window_OpeningFcn,
...
'gui_OutputFcn',
@Main_interface_window_OutputFcn,
...            'gui_LayoutFcn',  []
, ...              'gui_Callback',
[]);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State,
varargin{:});
else
gui_mainfcn(gui_State,
varargin{:});
end
function
Main_interface_window_OpeningFcn(h
Object, eventdata, handles,
varargin)
imshow('hawassa.PNG');
axes(handles.axes2)
imshow('hawassa.PNG');
axes(handles.axes5)
imshow('beam_detail.PNG');
axes(handles.axes9)
imshow('beam_view.PNG');
guidata(hObject, handles);
function varargout =
Main_interface_window_OutputFcn(hO
bject, eventdata, handles)
varargout{1} = handles.output;
function
radiobutton8_Callback(hObject,
eventdata, handles)
run
('Beam_for_combined_shear_and_tors
ion.m');
function
radiobutton7_Callback(hObject,
eventdata, handles)
run ('Beam_for_torsion.m');
function
radiobutton6_Callback(hObject,
eventdata, handles)
run ('Beam_for_shear.m')
radiobutton9_Callback(hObject,
eventdata, handles)
run ('main.m')
```

2. Source Code for Rectangular Beam Interface for Flexure

```
handles.output = hObject;
axes(handles.axes2)
imshow('beam
section1.PNG');title('Rectangular
beam cross section')
axes(handles.axes1)
imshow('strain
dist1.PNG');title('Parabolic
Rectangular stress strain
distribution')
guidata(hObject, handles);
function varargout =
Rectangular_beam_for_flexure_Outpu
tFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
end
format short;
%Step one Material Property
function
popupmenu2_Callback(hObject,
eventdata, handles)
%for persistent and transient
design situation:partial seafty
factor for
%concrete=1.5 and partial seafty
factor for steel=1.15
switch
get(handles.popupmenu2,'Value')
case 1
concreteclass= 'select concrete
grade';
fck_in_MPa=0;
fctm_in_MPa=0;
fcd_in_MPa=0;
case 2
concreteclass= '15';
fck_in_MPa=0.8*15
fctm_in_MPa=0.3*(12)^(2/3)
fcd_in_MPa=0.8*0.85*15/1.5
case 3
concreteclass= '20';
fck_in_MPa=0.8*20
fctm_in_MPa=0.3*(16)^(2/3)
fcd_in_MPa=0.8*0.85*20/1.5
case 4
concreteclass= '25';
fck_in_MPa=0.8*25
fctm_in_MPa=0.3*(20)^(2/3)
fcd_in_MPa=0.8*0.85*25/1.5
case 5
concreteclass= '30';
fck_in_MPa=0.8*30
fctm_in_MPa=0.3*(24)^(2/3)
fcd_in_MPa=0.8*0.85*30/1.5
case 6
concreteclass= '37';
fck_in_MPa=0.8*37
fctm_in_MPa=0.3*(29.6)^(2/3)
fcd_in_MPa=0.8*0.85*37/1.5
case 7
concreteclass= '45';
fck_in_MPa=0.8*45
fctm_in_MPa=0.3*(36)^(2/3)
fcd_in_MPa=0.8*0.85*45/1.5
case 8
concreteclass= '50';
fck_in_MPa=0.8*50
fctm_in_MPa=0.3*(40)^(2/3)
fcd_in_MPa=0.8*0.85*50/1.5
otherwise
```

```
end
set(handles.edit132,'string',fck_i
n_MPa);
set(handles.edit131,'string',fctm_
in_MPa);
set(handles.edit38,'string',fcd_in
_MPa);
function
popupmenu2_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
popupmenu1_Callback(hObject,
eventdata, handles)
%fyk &fyd are characterstices &
design yield strength of
reinforcement
%respectvely
switch
get(handles.popupmenu1,'Value')
case 1
steelclass= 'select steel grade';
fyk_in_MPa=0;
fyd_in_MPa=0;
case 2
steelclass= '260';
fyk_in_MPa=260
fyd_in_MPa=260/1.15
case 3
steelclass= '300';
fyk_in_MPa=300
fyd_in_MPa=300/1.15
case 4
steelclass= '360';
fyk_in_MPa=360
fyd_in_MPa=360/1.15
case 5
steelclass= '400';
fyk_in_MPa=400
fyd_in_MPa=400/1.15
case 6
steelclass= '460'
fyk_in_MPa=460
fyd_in_MPa=460/1.15
case 7
steelclass= '500';
fyk_in_MPa=500
fyd_in_MPa=500/1.15
case 8
steelclass= '560';
fyk_in_MPa=560
fyd_in_MPa=560/1.15
case 9
steelclass= '600';
fyk_in_MPa=600
fyd_in_MPa=600/1.15
otherwise
end
set(handles.edit130,'string',fyk_i
n_MPa)
set(handles.edit39,'string',fyd_in
_MPa)
function
popupmenu1_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
```

```
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
% Step Two Area of steel in mm2
function
popupmenu3_Callback(hObject,
eventdata, handles)
switch
get(handles.popupmenu3,'Value')
case 1
Diameter= 'select bar size';
bar_diametr_in_mm=0;
unit_area_of_reinforcement_in_mm2=
0;
case 2
Diameter= '6';
bar_diametr_in_mm=6;
unit_area_of_reinforcement_in_mm2=
(pi*6^2)/(4);
case 3
Diameter= '8';
bar_diametr_in_mm=8;
unit_area_of_reinforcement_in_mm2=
(pi*8^2)/(4);
case 4
Diameter= '10';
bar_diametr_in_mm=10;
unit_area_of_reinforcement_in_mm2=
(pi*10^2)/(4);
case 5
Diameter= '12';
bar_diametr_in_mm=12;
unit_area_of_reinforcement_in_mm2=
(pi*12^2)/(4);
case 6
Diameter= '14';
bar_diametr_in_mm=14;
unit_area_of_reinforcement_in_mm2=
(pi*14^2)/(4);
case 7
Diameter= '16';
bar_diametr_in_mm=16;
unit_area_of_reinforcement_in_mm2=
(pi*16^2)/(4);
case 8
Diameter= '18';
bar_diametr_in_mm=18;
unit_area_of_reinforcement_in_mm2=
(pi*18^2)/(4);
case 9
Diameter= '20';
bar_diametr_in_mm=20;
unit_area_of_reinforcement_in_mm2=
(pi*20^2)/(4);
case 10
Diameter= '22';
bar_diametr_in_mm=22;
unit_area_of_reinforcement_in_mm2=
(pi*22^2)/(4);
case 11
Diameter= '24';
bar_diametr_in_mm=24;
unit_area_of_reinforcement_in_mm2=
(pi*24^2)/(4);
case 12
Diameter= '26';
bar_diametr_in_mm=26;
unit_area_of_reinforcement_in_mm2=
(pi*26^2)/(4);
case 13
Diameter= '30';
```

```
bar_diametr_in_mm=30;
unit_area_of_reinforcement_in_mm2=
(pi*30^2)/(4);
case 14
Diameter= '32';
bar_diametr_in_mm=32;
unit_area_of_reinforcement_in_mm2=
(pi*32^2)/(4);
case 15
Diameter= '34';
bar_diametr_in_mm=34;
unit_area_of_reinforcement_in_mm2=
(pi*34^2)/(4);
case 16
Diameter= '40';
bar_diametr_in_mm=40;
unit_area_of_reinforcement_in_mm2=
(pi*40^2)/(4);
otherwise
end
set(handles.edit21,'string',unit_a
rea_of_reinforcement_in_mm2);
set(handles.edit200,'string',bar_d
iametr_in_mm);
function
popupmenu3_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
pushbutton59_Callback(hObject,
eventdata, handles)
%As1 ,As2 & As3 are area of steel
for rectangular beam at the
tension side
As1_prov_at_bottom_row1_in_mm2=str
2num(get(handles.edit21,'string'))
*str2num(get(handles.edit79,'strin
g'))
set(handles.edit80,'string',As1_pr
ov_at_bottom_row1_in_mm2)
number_of_bar_at_row1=str2num(get(
handles.edit79,'string'))
diameter_of_bottom_bar_in_mm=str2n
um(get(handles.edit200,'string'))
diameter=0;
if number_of_bar_at_row1>0
disp(diameter_of_bottom_bar_in_mm)
set(handles.edit201,'string',diame
ter_of_bottom_bar_in_mm)
elseif number_of_bar_at_row1==0
disp(0)
set(handles.edit201,'string',diame
ter);
end
function
pushbutton73_Callback(hObject,
eventdata, handles)
As2_prov_at_bottom_row2_in_mm2=str
2num(get(handles.edit21,'string'))
*str2num(get(handles.edit51,'strin
g'))
set(handles.edit55,'string',As2_pr
ov_at_bottom_row2_in_mm2)
number_of_bottom_bar_at_row2=str2n
um(get(handles.edit51,'string'))
```

```
diameter_of_bottom_bar_in_mm=str2n
um(get(handles.edit200,'string'))
diameter=0;
if number_of_bottom_bar_at_row2>0
disp(diameter_of_bottom_bar_in_mm)
set(handles.edit202,'string',diame
ter_of_bottom_bar_in_mm)
elseif
number_of_bottom_bar_at_row2==0
disp(0)
set(handles.edit202,'string',diame
ter);
end
function
pushbutton74_Callback(hObject,
eventdata, handles)
As3_prov_at_bottom_row3_in_mm2=str
2num(get(handles.edit21,'string'))
*str2num(get(handles.edit52,'strin
g'))
set(handles.edit56,'string',As3_pr
ov_at_bottom_row3_in_mm2)
number_of_bottom_bar_at_row3=str2n
um(get(handles.edit52,'string'));
diameter_of_bottom_bar_in_mm=str2n
um(get(handles.edit200,'string'));
diameter=0;
if number_of_bottom_bar_at_row3>0
disp(diameter_of_bottom_bar_in_mm)
set(handles.edit203,'string',diame
ter_of_bottom_bar_in_mm);
elseif
number_of_bottom_bar_at_row3==0
disp(0)
set(handles.edit203,'string',diame
ter);
end
function
pushbutton75_Callback(hObject,
eventdata, handles)
As1c_prov_at_top_row1_in_mm2=str2n
um(get(handles.edit21,'string'))*s
tr2num(get(handles.edit53,'string'
))
set(handles.edit40,'string',As1c_p
rov_at_top_row1_in_mm2)
number_of_top_bar_at_row1=str2num(
get(handles.edit53,'string'))
diameter_of_top_bar_in_mm=str2num(
get(handles.edit200,'string'))
diameter=0;
if number_of_top_bar_at_row1>0
disp(diameter_of_top_bar_in_mm)
set(handles.edit204,'string',diame
ter_of_top_bar_in_mm)
elseif
number_of_top_bar_at_row1==0
disp(0)
set(handles.edit204,'string',diame
ter);
end
function
pushbutton76_Callback(hObject,
eventdata, handles)
As2c_prov_at_top_row2_in_mm2=str2n
um(get(handles.edit54,'string'))*s
tr2num(get(handles.edit21,'string'
))
set(handles.edit57,'string',As2c_p
rov_at_top_row2_in_mm2)
number_of_top_bar_at_row2=str2num(
get(handles.edit54,'string'))
diameter_of_top_bar_in_mm=str2num(
get(handles.edit200,'string'))
```

```
diameter=0;
if number_of_top_bar_at_row2>0
disp(diameter_of_top_bar_in_mm)
set(handles.edit205,'string',diame
ter_of_top_bar_in_mm);
elseif
number_of_top_bar_at_row2==0
disp(0)
set(handles.edit205,'string',diame
ter);
end
function
pushbutton55_Callback(hObject,
eventdata, handles)
As3c_prov_at_top_row3_in_mm2=str2n
um(get(handles.edit102,'string'))*
str2num(get(handles.edit21,'string
'))
set(handles.edit101,'string',As3c_
prov_at_top_row3_in_mm2)
number_of_top_bar_at_row3=str2num(
get(handles.edit102,'string'))
diameter_of_top_bar_in_mm=str2num(
get(handles.edit200,'string'))
diameter=0;
if number_of_top_bar_at_row3>0
disp(diameter_of_top_bar_in_mm)
set(handles.edit206,'string',diame
ter_of_top_bar_in_mm);
elseif
number_of_top_bar_at_row3==0
disp(0)
set(handles.edit206,'string',diame
ter);
end
% Step Three beam cross-section
property in mm
function
pushbutton36_Callback(hObject,
eventdata, handles)
%dt,dc is efective depth cover at
tension & compression zone and d
is
%effective depth
effecive_depth_cover_at_bottom=(st
r2num(get(handles.edit58,'string')
)*str2num(get(handles.edit80,'stri
ng'))+str2num(get(handles.edit59,'
string'))*str2num(get(handles.edit
55,'string'))+str2num(get(handles.
edit60,'string'))*str2num(get(hand
les.edit56,'string')))/(str2num(ge
t(handles.edit55,'string'))+str2nu
m(get(handles.edit80,'string'))+st
r2num(get(handles.edit56,'string')
))
set(handles.edit50,'string',effeci
ve_depth_cover_at_bottom);
A31=0;
As_compression_total=str2num(get(h
andles.edit40,'string'))+str2num(g
et(handles.edit57,'string'));
effecive_depth_cover_at_top=(str2n
um(get(handles.edit61,'string'))*s
tr2num(get(handles.edit40,'string'
))+str2num(get(handles.edit62,'str
ing'))*str2num(get(handles.edit57,
'string'))+str2num(get(handles.edi
t103,'string'))*str2num(get(handle
s.edit101,'string')))/(str2num(get
(handles.edit40,'string'))+str2num
(get(handles.edit57,'string'))+str
2num(get(handles.edit101,'string')
))
```

```
if As_compression_total==0
disp(A31);
set(handles.edit44,'string',A31)
elseif As_compression_total>0
disp(effecive_depth_cover_at_top)
set(handles.edit44,'string',effeci
ve_depth_cover_at_top)
end
D_in_mm=str2num(get(handles.edit84
,'string'))
effecive_depth_cover_at_bottom_in_
mm=str2num(get(handles.edit50,'str
ing'))
d_in_mm=D_in_mm-
effecive_depth_cover_at_bottom_in_
mm
set(handles.edit5,'string',d_in_mm
);
%Step Four Determination of
Neutral Axis Depth alphac , X , ac
and Z
function
pushbutton20_Callback(hObject,
eventdata, handles)
%Ac1=alphac for single
reinforcement and Ac2=alphac for
double reinforcement
As_compression_in_mm2=str2num(get(
handles.edit40,'string'))+str2num(
get(handles.edit57,'string'))+str2
num(get(handles.edit101,'string'))
;
Alpha_c1=(str2num(get(handles.edit
80,'string'))+str2num(get(handles.
edit55,'string'))+str2num(get(hand
les.edit56,'string')))*str2num(get
(handles.edit39,'string'))/(str2nu
m(get(handles.edit38,'string'))*st
r2num(get(handles.edit5,'string'))
*str2num(get(handles.edit4,'string
')))
Aclpha_c2=(((str2num(get(handles.e
dit80,'string'))+str2num(get(handl
es.edit55,'string'))+str2num(get(h
andles.edit56,'string')))-
(str2num(get(handles.edit40,'strin
g'))+str2num(get(handles.edit57,'s
tring'))+str2num(get(handles.edit1
01,'string'))))*str2num(get(handle
s.edit39,'string')))/(str2num(get(
handles.edit38,'string'))*str2num(
get(handles.edit5,'string'))*str2n
um(get(handles.edit4,'string')))
if As_compression_in_mm2==0
disp(Alpha_c1);
set(handles.edit23,'string',Alpha_
c1);
elseif As_compression_in_mm2>0
disp(Aclpha_c2);
set(handles.edit23,'string',Aclpha
_c2);
end
E_in_GPa=str2num(get(handles.edit1
49,'string'))
Eyd=str2num(get(handles.edit39,'st
ring'))/E_in_GPa
set(handles.edit77,'string',Eyd);
As_compresssion=str2num(get(handle
s.edit40,'string'))+str2num(get(ha
ndles.edit57,'string'))+str2num(ge
t(handles.edit101,'string'))
ecm11=(-2-
75*str2num(get(handles.edit23,'str
```

```
ing')))/(3*str2num(get(handles.edi
t23,'string'))-3)
es11=(8.5-
10.5*str2num(get(handles.edit23,'s
tring')))/(3*str2num(get(handles.e
dit23,'string')))
if ecm11 >3.5
disp(es11);
set(handles.edit76,'string',es11);
end
if
es11>str2num(get(handles.edit77,'s
tring'))
disp(es11);
set(handles.edit76,'string',es11);
end
es11=(8.5-
10.5*Aclpha_c2)/(3*Aclpha_c2)
if As_compresssion>0
disp(es11);
set(handles.edit76,'string',es11);
end
%es11,es12 & es13 are strain of
steel on the tension side
design_criteria1='OK';
design_criteria2='NOT OK';
es11=str2num(get(handles.edit76,'s
tring'));
if es11>Eyd
disp(design_criteria1);
set(handles.edit78,'string',design
_criteria1);
elseif es11<Eyd
disp(design_criteria2);
set(handles.edit78,'string',design
_criteria2);
end
if
es11>str2num(get(handles.edit77,'s
tring'))
disp(es11);
set(handles.edit91,'string',es11);
end
p1=
[(3*(str2num(get(handles.edit80,'s
tring'))+str2num(get(handles.edit5
5,'string'))+str2num(get(handles.e
dit56,'string'))-
(str2num(get(handles.edit40,'strin
g'))+str2num(get(handles.edit57,'s
tring'))))*E_in_GPa)/(str2num(get(
handles.edit38,'string'))*str2num(
get(handles.edit4,'string'))*str2n
um(get(handles.edit5,'string')))
(10.5*(str2num(get(handles.edit80,
'string'))+str2num(get(handles.edi
t55,'string'))+str2num(get(handles
.edit56,'string'))-
(str2num(get(handles.edit40,'strin
g'))+str2num(get(handles.edit57,'s
tring'))))*E_in_GPa)/(str2num(get(
handles.edit38,'string'))*str2num(
get(handles.edit4,'string'))*str2n
um(get(handles.edit5,'string'))) -
8.5];
pes2 = roots(p1);
es12=pes2(1,1);
es13=pes2(2,1);
A44=0;
Ecu2=3.5;
if es12>0
disp(es12);
set(handles.edit91,'string',es12);
end
```

```
if es11<Eyd
disp(es12);
set(handles.edit91,'string',es12);
set(handles.edit75,'string',Ecu2);
elseif es11>Eyd
disp(A44)
set(handles.edit91,'string',A44);
end
if es13>0
disp(es13);
set(handles.edit91,'string',es13);
end
if es11<Eyd
disp(es13);
set(handles.edit91,'string',es13);
set(handles.edit75,'string',Ecu2);
elseif es11>Eyd
disp(A44)
 set(handles.edit91,'string',A44);
end
%Ac1=alphac for single
reinforcement and Ac2=alphac for
double reinforcement and ecm1 is
maximum compressive strain of
steel
es11=str2num(get(handles.edit76,'s
tring'));
Alpha_c=str2num(get(handles.edit23
,'string'));
Alpha_c11=(str2num(get(handles.edi
t80,'string'))+str2num(get(handles
.edit55,'string'))+str2num(get(han
dles.edit56,'string')))*200*str2nu
m(get(handles.edit91,'string'))/(s
tr2num(get(handles.edit38,'string'
))*str2num(get(handles.edit5,'stri
ng'))*str2num(get(handles.edit4,'s
tring')));
if es11<Eyd
disp(Alpha_c11);
set(handles.edit89,'string',Alpha_
c11);
elseif es11>Eyd
disp(Alpha_c);
set(handles.edit89,'string',Alpha_
c);
end
es11=str2num(get(handles.edit76,'s
tring'));
es111=str2num(get(handles.edit91,'
string'));
es1=25
if es11>25
disp(es1);
set(handles.edit68,'string',es1);
elseif es11<es111
disp(es111);
set(handles.edit68,'string',es111)
;
elseif es111<25
set(handles.edit68,'string',es111)
;
end
ecu2=3.5
p= [-1 6 -
12*str2num(get(handles.edit23,'str
ing')) -
300*str2num(get(handles.edit23,'st
ring'))];
pecm = roots(p);
ecm12=pecm(1,1)
ecm13=pecm(2,1)
ecm14=pecm(3,1)
```

```
if pecm(1,1)<=2
disp(ecm12);
set(handles.edit75,'string',ecm12)
;
elseif pecm(2,1)<=2
disp(ecm13);
set(handles.edit75,'string',ecm13)
;
elseif pecm(3,1)<=2
disp(ecm14);
set(handles.edit75,'string',ecm14)
;
end
if ecm11 >=2
disp(ecm11);
set(handles.edit75,'string',ecm11)
;
elseif ecm11<=3.5
disp(ecm11);
set(handles.edit75,'string',ecm11)
;
end
if As_compresssion>0
disp(ecu2)
set(handles.edit75,'string',ecu2);
end
ecm=str2num(get(handles.edit75,'st
ring'))
ecu2=3.5
if ecm<ecu2
disp(ecm);
set(handles.edit69,'string',ecm);
elseif ecm>=ecu2
disp(ecu2);
set(handles.edit69,'string',ecu2);
end
ecm1=str2num(get(handles.edit75,'s
tring'));
Kx1=(str2num(get(handles.edit75,'s
tring'))/(str2num(get(handles.edit
75,'string'))+25));
if ecm1<=3.5
disp(Kx1);
set(handles.edit72,'string',Kx1);
x_in_mm=Kx1*str2num(get(handles.ed
it5,'string'))
set(handles.edit24,'string',x_in_m
m);
elseif ecm1>=3.5&&es>Eyd
Kx2=(3.5/(3.5+es))
disp(Kx2);
set(handles.edit72,'string',Kx2);
x_in_mm=Kx2*str2num(get(handles.ed
it5,'string'))
set(handles.edit24,'string',x_in_m
m);
elseif es<Eyd
es1=str2num(get(handles.edit91,'st
ring'));
Kx3=(3.5)/(es1+3.5)
disp(Kx3);
set(handles.edit72,'string',Kx3);
x_in_mm=Kx3*str2num(get(handles.ed
it5,'string'))
set(handles.edit24,'string',x_in_m
m);
end
if As_compression_in_mm2>0
Aalpha_c2=((str2num(get(handles.ed
it80,'string'))+str2num(get(handle
s.edit55,'string'))+str2num(get(ha
ndles.edit56,'string')))*str2num(g
et(handles.edit39,'string'))-
```

```
(str2num(get(handles.edit40,'strin
g'))+str2num(get(handles.edit57,'s
tring'))+str2num(get(handles.edit1
01,'string')))*str2num(get(handles
.edit39,'string')))/(str2num(get(h
andles.edit38,'string'))*str2num(g
et(handles.edit5,'string'))*str2nu
m(get(handles.edit4,'string')));
es1=(8.5-
10.5*Aalpha_c2)/(3*Aalpha_c2);
Kx=(3.5)/(3.5+es1)
disp(Kx)
set(handles.edit72,'string',Kx);
x_in_mm=Kx*str2num(get(handles.edi
t5,'string'))
set(handles.edit24,'string',x_in_m
m);
end
es21=(3.5*(str2num(get(handles.edi
t24,'string'))-
str2num(get(handles.edit44,'string
'))))/(str2num(get(handles.edit24,
'string')))
es22=0;
if As_compression_in_mm2==0
disp(es22);
set(handles.edit43,'string',es22);
set(handles.edit47,'string',es22);
set(handles.edit70,'string',es22);
elseif
As_compression_in_mm2>0&&es21>Eyd
disp(es21);
set(handles.edit43,'string',es21);
set(handles.edit47,'string',es22);
set(handles.edit70,'string',es22);
elseif
As_compression_in_mm2>0&&es21<Eyd
disp(es21);
set(handles.edit43,'string',es21);
p=[10.5*E_in_GPa*(str2num(get(hand
les.edit40,'string'))+str2num(get(
handles.edit57,'string'))+str2num(
get(handles.edit101,'string'))) (-
36.75*E_in_GPa*(str2num(get(handle
s.edit57,'string'))+str2num(get(ha
ndles.edit40,'string'))+str2num(ge
t(handles.edit101,'string')))-
10.5*(str2num(get(handles.edit80,'
string'))+str2num(get(handles.edit
55,'string'))+str2num(get(handles.
edit56,'string')))*str2num(get(han
dles.edit39,'string'))) (-
29.75*str2num(get(handles.edit44,'
string'))*(str2num(get(handles.edi
t38,'string'))*str2num(get(handles
.edit4,'string')))+36.75*(str2num(
get(handles.edit55,'string'))+str2
num(get(handles.edit56,'string'))+
str2num(get(handles.edit80,'string
')))*str2num(get(handles.edit39,'s
tring')))];
pes3=roots(p);
es23=pes3(1,1);
es24=pes3(2,1);
set(handles.edit47,'string',es23);
set(handles.edit47,'string',es24);
Alpha_c2=((str2num(get(handles.edi
t80,'string'))+str2num(get(handles
.edit55,'string'))+str2num(get(han
dles.edit56,'string')))*str2num(ge
t(handles.edit149,'string'))*str2n
um(get(handles.edit47,'string'))-
(str2num(get(handles.edit40,'strin
g'))+str2num(get(handles.edit57,'s
```

```
tring'))+str2num(get(handles.edit1
01,'string')))*str2num(get(handles
.edit149,'string'))*str2num(get(ha
ndles.edit47,'string')))/(str2num(
get(handles.edit38,'string'))*str2
num(get(handles.edit5,'string'))*s
tr2num(get(handles.edit4,'string')
))
set(handles.edit70,'string',Alpha_
c2);
end
%es2 strain at the compression
reinforcement side
design_criteria1='steel yiled'
design_criteria11='steel not
yield'
A43=0;
es2=str2num(get(handles.edit43,'st
ring'));
if es2==0
disp(A43);
set(handles.edit83,'string',A43);
elseif es2>Eyd
disp(design_criteria1);
set(handles.edit83,'string',design
_criteria1);
elseif es2<Eyd
disp(design_criteria11);
set(handles.edit83,'string',design
_criteria11);
end
%Bc=betac,ecm=maximum concrete
compressive strain,Z lever arm and
Asc is
%reinforcement at compression zone
Kx=str2num(get(handles.edit72,'str
ing'));
ecm=str2num(get(handles.edit75,'st
ring'));
Beta_c1=Kx*(((8-ecm)/(4*(6-ecm))))
Beta_c2=Kx*((((ecm*(3*ecm-
4))+2)/(2*ecm*(3*ecm-2))))
if ecm<2
disp(Beta_c1);
set(handles.edit86,'string',Beta_c
1);
end
if ecm>=2
disp(Beta_c2);
set(handles.edit86,'string',Beta_c
2);
end
if ecm<=3.5
disp(Beta_c2);
set(handles.edit86,'string',Beta_c
2);
end
if ecm>=3.5
disp(Beta_c2);
set(handles.edit86,'string',Beta_c
2);
end
ecm=3.5;
Beta_c2=Kx*((((ecm*(3*ecm-
4)+2))/(2*ecm*(3*ecm-2))))
if As_compression_in_mm2>=0
disp(Beta_c2);
set(handles.edit86,'string',Beta_c
2);
end
es2=0;
A45=0;
if As_compression_in_mm2==0
disp(es2)
```

```
set(handles.edit47,'string',es2);
disp(A45)
set(handles.edit70,'string',A45);
end
Z_in_mm=str2num(get(handles.edit5,
'string'))*(1-
str2num(get(handles.edit86,'string
')))
set(handles.edit11,'string',Z_in_m
m);
% check for failure type
%x is nuetral axis depth
Xb_in_mm=(3.5/(3.5+str2num(get(han
dles.edit77,'string'))))*str2num(g
et(handles.edit5,'string'))
set(handles.edit71,'string',Xb_in_
mm);
Xb_in_mm=str2num(get(handles.edit7
1,'string'));
X=str2num(get(handles.edit24,'stri
ng'));
design_criteria_5='Compression
Failure occur'
design_criteria_6='Tension Failure
occur'
design_criteria_7='Balanced
Failure ocuur'
if X>Xb_in_mm
disp(design_criteria_5);
set(handles.edit73,'string',design
_criteria_5);
elseif X==Xb_in_mm
disp(design_criteria_7);
set(handles.edit73,'string',design
_criteria_7);
else
disp(design_criteria_6);
set(handles.edit73,'string',design
_criteria_6);
end
% Final step analysis for concrte
& steel resultant force & moment
capacty
% Fc is compression
force,compression force of
reinforcement & Fs tension
% reinforcement force
Fc_in_N=str2num(get(handles.edit23
,'string'))*str2num(get(handles.ed
it38,'string'))*str2num(get(handle
s.edit4,'string'))*str2num(get(han
dles.edit5,'string'))
Fc1_in_N=str2num(get(handles.edit8
9,'string'))*str2num(get(handles.e
dit38,'string'))*str2num(get(handl
es.edit4,'string'))*str2num(get(ha
ndles.edit5,'string'))
es1=str2num(get(handles.edit76,'st
ring'));
Eyd=str2num(get(handles.edit77,'st
ring'));
if es1<Eyd
disp(Fc1_in_N);
set(handles.edit65,'string',Fc1_in
_N)
end
if es1>Eyd
disp(Fc_in_N);
set(handles.edit65,'string',Fc_in_
N)
end
Fsc_in_N=str2num(get(handles.edit3
9,'string'))*str2num(get(handles.e
dit40,'string'))+str2num(get(handl
```

```
es.edit39,'string'))*str2num(get(h
andles.edit57,'string'))+str2num(g
et(handles.edit39,'string'))*str2n
um(get(handles.edit101,'string'))
Fsc1_in_N=str2num(get(handles.edit
47,'string'))*str2num(get(handles.
edit149,'string'))*str2num(get(han
dles.edit40,'string'))+str2num(get
(handles.edit47,'string'))*str2num
(get(handles.edit149,'string'))*st
r2num(get(handles.edit57,'string')
)+str2num(get(handles.edit47,'stri
ng'))*str2num(get(handles.edit149,
'string'))*str2num(get(handles.edi
t101,'string'))
es2=str2num(get(handles.edit43,'st
ring'));
if es2>Eyd
disp(Fsc_in_N);
set(handles.edit64,'string',Fsc_in
_N);
end
if es2<Eyd
disp(Fsc1_in_N);
set(handles.edit64,'string',Fsc1_i
n_N);
end
C_in_N=str2num(get(handles.edit65,
'string'))+str2num(get(handles.edi
t64,'string'))
set(handles.edit67,'string',C_in_N
);
Fs_in_N=str2num(get(handles.edit39
,'string'))*str2num(get(handles.ed
it80,'string'))+str2num(get(handle
s.edit39,'string'))*str2num(get(ha
ndles.edit55,'string'))+str2num(ge
t(handles.edit39,'string'))*str2nu
m(get(handles.edit56,'string'))
Fs1_in_N=str2num(get(handles.edit9
1,'string'))*str2num(get(handles.e
dit149,'string'))*str2num(get(hand
les.edit80,'string'))+str2num(get(
handles.edit91,'string'))*str2num(
get(handles.edit149,'string'))*str
2num(get(handles.edit55,'string'))
+str2num(get(handles.edit91,'strin
g'))*str2num(get(handles.edit149,'
string'))*str2num(get(handles.edit
56,'string'))
if es1>Eyd
disp(Fs_in_N);
set(handles.edit63,'string',Fs_in_
N);
elseif es1<Eyd
disp(Fs1_in_N);
set(handles.edit63,'string',Fs1_in
_N);
end
%MRdr=MRd=total moment capacity of
rectangular beam
%As2=reinforcement at the
compression side
As_compression=str2num(get(handles
.edit40,'string'))+str2num(get(han
dles.edit57,'string'))+str2num(get
(handles.edit101,'string'));
Af=0;
es1=str2num(get(handles.edit76,'st
ring'));
Eyd=str2num(get(handles.edit77,'st
ring'));
MRdr11_in_Nmm=(str2num(get(handles
.edit80,'string'))+str2num(get(han
```

```
dles.edit55,'string'))+str2num(get
(handles.edit56,'string')))*str2nu
m(get(handles.edit39,'string'))*st
r2num(get(handles.edit5,'string'))
*(1-
str2num(get(handles.edit86,'string
')))
MRdr12_in_Nmm=(str2num(get(handles
.edit80,'string'))+str2num(get(han
dles.edit55,'string'))+str2num(get
(handles.edit56,'string')))*str2nu
m(get(handles.edit149,'string'))*s
tr2num(get(handles.edit91,'string'
))*str2num(get(handles.edit5,'stri
ng'))*(1-
str2num(get(handles.edit86,'string
')))
if  es1>Eyd
disp(MRdr11_in_Nmm);
set(handles.edit25,'string',MRdr11
_in_Nmm);
end
if es1<Eyd
disp(MRdr12_in_Nmm);
set(handles.edit25,'string',MRdr12
_in_Nmm);
end
Af1=0;
es2=str2num(get(handles.edit43,'st
ring'));
MRdr13_in_Nmm=str2num(get(handles.
edit23,'string'))*str2num(get(hand
les.edit38,'string'))*str2num(get(
handles.edit4,'string'))*str2num(g
et(handles.edit5,'string'))*str2nu
m(get(handles.edit5,'string'))*(1-
str2num(get(handles.edit86,'string
')))+(str2num(get(handles.edit40,'
string'))+str2num(get(handles.edit
57,'string'))+str2num(get(handles.
edit101,'string')))*str2num(get(ha
ndles.edit39,'string'))*(str2num(g
et(handles.edit5,'string'))-
str2num(get(handles.edit44,'string
')))
MRdr14_in_Nmm
=str2num(get(handles.edit70,'strin
g'))*str2num(get(handles.edit38,'s
tring'))*str2num(get(handles.edit4
,'string'))*str2num(get(handles.ed
it5,'string'))*str2num(get(handles
.edit5,'string'))*(1-
str2num(get(handles.edit86,'string
')))+(str2num(get(handles.edit40,'
string'))+str2num(get(handles.edit
57,'string'))+str2num(get(handles.
edit101,'string')))*str2num(get(ha
ndles.edit47,'string'))*str2num(ge
t(handles.edit149,'string'))*(str2
num(get(handles.edit5,'string'))-
str2num(get(handles.edit44,'string
')))
if  As_compression==0
disp(Af1);
set(handles.edit90,'string',Af1);
elseif As_compression>0
disp(Af);
set(handles.edit25,'string',Af);
end
if es2<Eyd
disp(MRdr14_in_Nmm);
set(handles.edit90,'string',MRdr14
_in_Nmm);
elseif es2>Eyd
```

```
disp(MRdr13_in_Nmm);
set(handles.edit90,'string',MRdr13
_in_Nmm);
end
%step checked for design part
function
pushbutton150_Callback(hObject,
eventdata, handles)
% Ast area of reinforcement at
tension zone
dc=str2num(get(handles.edit44,'str
ing'));
bw=str2num(get(handles.edit4,'stri
ng'));
fyk=str2num(get(handles.edit130,'s
tring'));
fcd=str2num(get(handles.edit38,'st
ring'));
l=str2num(get(handles.edit170,'str
ing'));
d=str2num(get(handles.edit5,'strin
g'));
k=str2num(get(handles.edit172,'str
ing'));
M_in_Nmm=str2num(get(handles.edit1
36,'string'));
m=((fyk/1.15)/(0.8*fcd));
pr=(1.25/m)*(1-sqrt(1-
((2*M_in_Nmm)/(fcd*bw*(d)^2))));
Kxlim=str2num(get(handles.edit209,
'string'));
Ms_in_Nmm=Kxlim*(8.5/10.5)*fcd*bw*
d^2*(1-Kxlim*(24.75/59.5));
As_req_single_in_mm2=Ms_in_Nmm/(0.
814*(fyk/1.15)*d)
As_req_compression_in_mm2=(M_in_Nm
m-Ms_in_Nmm)/((fyk/1.15)*(d-dc))
As_req_double_in_mm2=As_req_single
_in_mm2+As_req_compression_in_mm2
As_req_in_mm2=pr*bw*d
Ad=0;
As_compression=str2num(get(handles
.edit40,'string'))+str2num(get(han
dles.edit57,'string'));
if  M_in_Nmm<Ms_in_Nmm
disp(As_req_in_mm2)
set(handles.edit207,'string',As_re
q_in_mm2);
elseif M_in_Nmm>Ms_in_Nmm
disp(As_req_single_in_mm2);
set(handles.edit207,'string',As_re
q_single_in_mm2);
set(handles.edit211,'string',As_re
q_compression_in_mm2);
set(handles.edit210,'string',As_re
q_double_in_mm2);
end
dia1t=str2num(get(handles.edit201,
'string'));
dia1c=str2num(get(handles.edit204,
'string'));
Asreq11=str2num(get(handles.edit20
7,'string'));
Asreq21=str2num(get(handles.edit21
0,'string'));
Asunit=0.25*pi*(dia1t)^2;
Asunitc=0.25*pi*(dia1c)^2;
Noofbars=Asreq11/Asunit
Noofbard=Asreq21/Asunit
Noofbarc=As_req_compression_in_mm2
/Asunitc
if  M_in_Nmm<Ms_in_Nmm
set(handles.edit208,'string',Noofb
ars);
```

```
else
set(handles.edit208,'string',Noofb
ard);
set(handles.edit256,'string',Noofb
arc)
end
if As_compression==0
set(handles.edit211,'string',Ad);
set(handles.edit210,'string',Ad);
set(handles.edit256,'string',Ad)
end
As_tension=str2num(get(handles.edi
t80,'string'))+str2num(get(handles
.edit55,'string'))+str2num(get(han
dles.edit56,'string'));
f1=min(As_tension/As_req_in_mm2,1.
5);
if l>=7000
f2=7000/l;
elseif l<=7000
f2=1;
else
end
p=(As_req_in_mm2)/(str2num(get(han
dles.edit4,'string'))*str2num(get(
handles.edit5,'string')));
p1=0;
po=0.001*sqrt(str2num(get(handles.
edit132,'string')));
a=l/d;
if p<=po
b=k*(11+1.5*sqrt(str2num(get(handl
es.edit132,'string')))*po/p+3.2*sq
rt(str2num(get(handles.edit132,'st
ring')))*(po/p-
1)^1.5)*(500/fyk)*f1*f2
elseif p>po
b=k*(11+1.5*sqrt(str2num(get(handl
es.edit132,'string')))*(po/(p-
p1))+1/12*sqrt(str2num(get(handles
.edit132,'string')))*sqrt(p1/po))*
(500/fyk)*f1*f2
end
design_criteria1='OK'
design_criteria11='NOT OK'
if a<b
disp(design_criteria1);
set(handles.edit171,'string',desig
n_criteria1);
else
disp(design_criteria11);
set(handles.edit171,'string',desig
n_criteria11);
end
kx= [0.4159 -1
(1.235*M_in_Nmm)/(str2num(get(hand
les.edit38,'string'))*str2num(get(
handles.edit4,'string'))*(str2num(
get(handles.edit5,'string'))^2))]
pkx = roots(kx);
kx11=pkx(1,1);
kx12=pkx(2,1);
kx13=[]
if kx11>0&&kx11< Kxlim
disp(kx11);
set(handles.edit500,'string',kx11;
elseif kx12>0&&kx12< Kxlim
disp(kx12);
set(handles.edit500,'string',kx12;
elseif kx11> Kxlim
disp(kx13);
set(handles.edit500,'string',kx13;
elseif kx12> Kxlim
```

```
disp(kx13);
set(handles.edit500,'string',kx13;
end
Kxlim=str2num(get(handles.edit209,
'string'));
Kx=str2num(get(handles.edit500,'st
ring'));
design_criteria2='No compression
steel require'
design_criteria22=Compression
steel require'
if Kx<Kxlim
disp(design_criteria2)
set(handles.edit134,'string',desig
n_criteria2);
else
disp(design_criteria22)
set(handles.edit134,'string',desig
n_criteria22);
end
Asmin=max((0.26*str2num(get(handle
s.edit131,'string'))*str2num(get(h
andles.edit4,'string')))/str2num(g
et(handles.edit130,'string')),0.00
13*str2num(get(handles.edit4,'stri
ng'))*str2num(get(handles.edit5,'s
tring')));
Asmax=0.04*str2num(get(handles.edi
t4,'string'))*str2num(get(handles.
edit84,'string'));
As_tension=str2num(get(handles.edi
t80,'string'))+str2num(get(handles
.edit55,'string'))+str2num(get(han
dles.edit56,'string'));
design_criteria3='OK'
design_criteria33='NOT OK'
if As_tension>Asmin
disp(design_criteria3);
set(handles.edit133,'string',desig
n_criteria3);
else
disp(design_criteria33);
set(handles.edit133,'string',desig
n_criteria33);
end
if As_tension<Asmax
disp(design_criteria3);
set(handles.edit133,'string',desig
n_criteria3);
else
disp(design_criteria33);
end
if As_compression<Asmax
disp(design_criteria3)
set(handles.edit133,'string',desig
n_criteria3);
else
disp(design_criteria33);
end
MEd=str2num(get(handles.edit136,'s
tring'));
MRds=str2num(get(handles.edit25,'s
tring'));
MRdd=str2num(get(handles.edit90,'s
tring'));
design_criteria4='OK'
design_criteria44='NOT OK'
if MRds>MEd
disp(design_criteria4)
set(handles.edit135,'string',desig
n_criteria4);
elseif MRdd>MEd
```

```
disp(design_criteria4)
set(handles.edit135,'string',desig
n_criteria4);
else
disp(design_criteria44)
set(handles.edit135,'string',desig
n_criteria44);
end
function
popupmenu8_Callback(hObject,
eventdata, handles)
switch
get(handles.popupmenu8,'Value')
case 1
d = 'beam support cond';
k=0;
case 2
a = 'simply supported beam';
k=1;
case 3
b = 'interior span beam';
k=1.5;
case 4
c = 'end span beam';
k=1.3;
case 5
d = 'cantliver beam';
k=0.4;
otherwise
end
set(handles.edit172,'string',k);
function
popupmenu8_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
pushbutton80_Callback(hObject,
eventdata, handles)
c=[];
d=[0];
set(handles.edit65,'string',c);
set(handles.edit64,'string',c);
set(handles.edit67,'string',c);
set(handles.edit63,'string',c);
set(handles.edit25,'string',c);
set(handles.edit90,'string',c);
set(handles.edit136,'string',c);
set(handles.edit170,'string',c);
set(handles.edit171,'string',c);
set(handles.edit207,'string',c);
set(handles.edit208,'string',c);
set(handles.edit172,'string',c);
set(handles.edit256,'string',c);
set(handles.edit211,'string',c);
set(handles.edit210,'string',c);
set(handles.edit134,'string',c);
set(handles.edit135,'string',c);
set(handles.edit133,'string',c);
set(handles.edit500,'string',c);
set(handles.edit130,'string',c);
set(handles.edit131,'string',c);
set(handles.edit132,'string',c);
set(handles.edit38,'string',c);
set(handles.edit39,'string',c);
set(handles.edit200,'string',c);
set(handles.edit21,'string',c);
set(handles.edit44,'string',c);
```

```
set(handles.edit50,'string',c);
set(handles.edit5,'string',c);
set(handles.edit84,'string',c);
set(handles.edit4,'string',c);
set(handles.edit70,'string',c);
set(handles.edit47,'string',c);
set(handles.edit69,'string',c);
set(handles.edit75,'string',c);
set(handles.edit68,'string',c);
set(handles.edit76,'string',c);
set(handles.edit77,'string',c);
set(handles.edit11,'string',c);
set(handles.edit91,'string',c);
set(handles.edit89,'string',c);
set(handles.edit86,'string',c);
set(handles.edit83,'string',c);
set(handles.edit43,'string',c);
set(handles.edit23,'string',c);
set(handles.edit78,'string',c);
set(handles.edit72,'string',c);
set(handles.edit24,'string',c);
set(handles.edit71,'string',c);
set(handles.edit80,'string',d);
set(handles.edit55,'string',d);
set(handles.edit56,'string',d);
set(handles.edit40,'string',d);
set(handles.edit57,'string',d);
set(handles.edit101,'string',d);
set(handles.edit79,'string',d);
set(handles.edit51,'string',d);
set(handles.edit52,'string',d);
set(handles.edit53,'string',d);
set(handles.edit54,'string',d);
set(handles.edit102,'string',d);
set(handles.edit201,'string',d);
set(handles.edit202,'string',d);
set(handles.edit203,'string',d);
set(handles.edit204,'string',d);
set(handles.edit205,'string',d);
set(handles.edit206,'string',d);
set(handles.edit58,'string',d);
set(handles.edit59,'string',d);
set(handles.edit60,'string',d);
set(handles.edit61,'string',d);
set(handles.edit62,'string',d);
set(handles.edit103,'string',d);
```

3. Source Code for T or Inverted L Beam Interface for Flexure

```
handles.output = hObject;
axes(handles.axes4)
imshow('beam
span1.PNG');title('Beam effective
span')
axes(handles.axes5)
imshow('section for
tbeam1.PNG');title('Equivalent
Rectangular stress-strain dist.')
axes(handles.axes6)
imshow('T beam
width1.PNG');title('Effective
flange width')
guidata(hObject, handles);
function varargout =
T_or_Inverted_L_beam_for_flexure_f
end
function
pushbutton44_Callback(hObject,
eventdata, handles)
As1_prov_at_bottom_row1_in_mm2=str
2num(get(handles.edit79,'string'))
*str2num(get(handles.edit21,'strin
g'))
set(handles.edit80,'string',As1_pr
ov_at_bottom_row1_in_mm2);
number_of_bar_at_row1=str2num(get(
handles.edit79,'string'))
diameter_of_bottom_bar_in_mm=str2n
um(get(handles.edit200,'string'))
diameter=0;
if number_of_bar_at_row1>0
disp(diameter_of_bottom_bar_in_mm)
set(handles.edit201,'string',diame
ter_of_bottom_bar_in_mm);
```

```
lange_in_comp_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;
function edit1_Callback(hObject,
eventdata, handles)
function edit1_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
```

```
elseif number_of_bar_at_row1==0
disp(0)
set(handles.edit201,'string',diame
ter);
end
function
pushbutton57_Callback(hObject,
eventdata, handles)
As2_prov_at_bottom_row2_in_mm2=str
2num(get(handles.edit93,'string'))
*str2num(get(handles.edit21,'strin
g'))
set(handles.edit94,'string',As2_pr
ov_at_bottom_row2_in_mm2);
number_of_bottom_bar_at_row2=str2n
um(get(handles.edit93,'string'))
diameter_of_bottom_bar_in_mm=str2n
um(get(handles.edit200,'string'))
diameter=0;
```

```
if number_of_bottom_bar_at_row2>0
disp(diameter_of_bottom_bar_in_mm)
set(handles.edit202,'string',diame
ter_of_bottom_bar_in_mm);
elseif
number_of_bottom_bar_at_row2==0
disp(0)
set(handles.edit202,'string',diame
ter);
end
function
pushbutton58_Callback(hObject,
eventdata, handles)
As3_prov_at_bottom_row3_in_mm2=str
2num(get(handles.edit95,'string'))
*str2num(get(handles.edit21,'strin
g'))
set(handles.edit96,'string',As3_pr
ov_at_bottom_row3_in_mm2);
number_of_bottom_bar_at_row3=str2n
um(get(handles.edit95,'string'))
diameter_of_bottom_bar_in_mm=str2n
um(get(handles.edit200,'string'))
diameter=0;
if number_of_bottom_bar_at_row3>0
disp(diameter_of_bottom_bar_in_mm)
set(handles.edit203,'string',diame
ter_of_bottom_bar_in_mm);
elseif
number_of_bottom_bar_at_row3==0
disp(0)
set(handles.edit203,'string',diame
ter);
end
function
pushbutton56_Callback(hObject,
eventdata, handles)
effecive_depth_cover_at_bottom=(st
r2num(get(handles.edit80,'string')
)*str2num(get(handles.edit97,'stri
ng'))+str2num(get(handles.edit94,'
string'))*str2num(get(handles.edit
98,'string'))+str2num(get(handles.
edit96,'string'))*str2num(get(hand
les.edit99,'string')))/(str2num(ge
t(handles.edit80,'string'))+str2nu
m(get(handles.edit94,'string'))+st
r2num(get(handles.edit96,'string')
));
set(handles.edit92,'string',effeci
ve_depth_cover_at_bottom)
D_in_mm=str2num(get(handles.edit84
,'string'))
d_in_mm=D_in_mm-
str2num(get(handles.edit92,'string
'))
set(handles.edit5,'string',d_in_mm
)
function
popupmenu14_Callback(hObject,
eventdata, handles)
switch
get(handles.popupmenu14,'Value')
case 1
   effective_length_lo_in_mm= 0 ;
case 2
    effectivespan = 'postive
moment interior suppourt' ;
effective_length_lo_in_mm=0.7*str2
num(get(handles.edit49,'string'))
case 3
    effectivespan = 'postive
moment edge suppourt' ;
```

```
effective_length_lo_in_mm=0.7*str2
num(get(handles.edit48,'string'))
case 4
    effectivespan = 'negative
moment interior suppourt' ;
effective_length_lo_in_mm=0.15*(st
r2num(get(handles.edit48,'string')
)+str2num(get(handles.edit49,'stri
ng')))
case 5
    effectivespan = 'negative
moment cantliver suppourt' ;
effective_length_lo_in_mm=0.15*str
2num(get(handles.edit48,'string'))
+str2num(get(handles.edit50,'strin
g'))
otherwise
end
set(handles.edit65,'string',effect
ive_length_lo_in_mm);
function
pushbutton37_Callback(hObject,
eventdata, handles)
beff_1_in_mm=min(0.2*str2num(get(h
andles.edit55,'string'))+0.1*str2n
um(get(handles.edit65,'string')),s
tr2num(get(handles.edit55,'string'
)))
set(handles.edit68,'string',beff_1
_in_mm);
beff_2_in_mm=min(0.2*str2num(get(h
andles.edit56,'string'))+0.1*str2n
um(get(handles.edit65,'string')),s
tr2num(get(handles.edit56,'string'
)))
set(handles.edit69,'string',beff_2
_in_mm);
beff_in_mm=str2num(get(handles.edi
t68,'string'))+str2num(get(handles
.edit69,'string'))+str2num(get(han
dles.edit4,'string'))
set(handles.edit70,'string',beff_i
n_mm);
function
pushbutton23_Callback(hObject,
eventdata, handles)
Alpha_c1=(str2num(get(handles.edit
80,'string'))+str2num(get(handles.
edit94,'string'))+str2num(get(hand
les.edit96,'string')))*str2num(get
(handles.edit39,'string'))/(str2nu
m(get(handles.edit38,'string'))*st
r2num(get(handles.edit5,'string'))
*str2num(get(handles.edit70,'strin
g')))
set(handles.edit23,'string',Alpha_
c1);
Es_in_GPa=str2num(get(handles.edit
149,'string'))
Eyd=str2num(get(handles.edit39,'st
ring'))/Es_in_GPa
set(handles.edit77,'string',Eyd);
Kx=str2num(get(handles.edit23,'str
ing'))/0.8
set(handles.edit72,'string',Kx);
X=str2num(get(handles.edit72,'stri
ng'))*str2num(get(handles.edit5,'s
tring'))
set(handles.edit24,'string',X);
analysis_type1='rectangular beam
analysis'
analysis_type2='T beam analysis'
analysis_type3='Inverted L beam
analysis'
```

```
X=str2num(get(handles.edit24,'stri
ng'));
hf=str2num(get(handles.edit6,'stri
ng'))
l1=str2num(get(handles.edit49,'str
ing'))
l2=str2num(get(handles.edit49,'str
ing'))
l3=str2num(get(handles.edit50,'str
ing'))
if X<hf
disp(analysis_type1);
set(handles.edit83,'string',analys
is_type1);
elseif X>hf&&l1>0
disp(analysis_type3);
set(handles.edit83,'string',analys
is_type3);
elseif X>hf&&l2>0
disp(analysis_type2);
set(handles.edit83,'string',analys
is_type2);
elseif X>hf&&l3>0
disp(analysis_type2);
set(handles.edit83,'string',analys
is_type2);
end
es1=25;
es=(3.5*str2num(get(handles.edit5,
'string'))/str2num(get(handles.edi
t24,'string')))-3.5;
if es>25
disp(25);
set(handles.edit91,'string',es1);
elseif es<25
disp(es);
set(handles.edit91,'string',es);
end
AA='NOT OK'
BB='OK'
es=str2num(get(handles.edit91,'str
ing'));
Eyd=str2num(get(handles.edit77,'st
ring'));
if es<Eyd
disp(AA);
set(handles.edit255,'string',AA);
elseif es>Eyd
disp(BB);
set(handles.edit255,'string',BB);
end
betac=0.4*str2num(get(handles.edit
72,'string'));
set(handles.edit86,'string',betac)
;
Z=str2num(get(handles.edit5,'strin
g'))*(1-
str2num(get(handles.edit86,'string
')));
set(handles.edit107,'string',Z);
Xb=(3.5/(3.5+str2num(get(handles.e
dit77,'string'))))*str2num(get(han
dles.edit5,'string'));
set(handles.edit105,'string',Xb);
Xb=str2num(get(handles.edit105,'st
ring'));
X=str2num(get(handles.edit24,'stri
ng'));
design_criteria1='Compression
Failure occur';
design_criteria2='Tension Failure
occur';
design_criteria3='Balanced Failure
ocuur';
```

```
if X>Xb
disp(design_criteria1);
set(handles.edit106,'string',desig
n_criteria1);
elseif X==Xb
disp(design_criteria3);
set(handles.edit106,'string',desig
n_criteria3);
else
disp(design_criteria2);
set(handles.edit106,'string',desig
n_criteria2);
end
Fc_in_N=str2num(get(handles.edit23
,'string'))*str2num(get(handles.ed
it38,'string'))*str2num(get(handle
s.edit5,'string'))*str2num(get(han
dles.edit70,'string'))
set(handles.edit7,'string',Fc_in_N
);
Fs_in_N=str2num(get(handles.edit39
,'string'))*str2num(get(handles.ed
it80,'string'))+str2num(get(handle
s.edit39,'string'))*str2num(get(ha
ndles.edit94,'string'))+str2num(ge
t(handles.edit39,'string'))*str2nu
m(get(handles.edit96,'string'))
Fs1=str2num(get(handles.edit91,'st
ring'))*str2num(get(handles.edit14
9,'string'))*str2num(get(handles.e
dit80,'string'))+str2num(get(handl
es.edit91,'string'))*str2num(get(h
andles.edit149,'string'))*str2num(
get(handles.edit94,'string'))+str2
num(get(handles.edit91,'string'))*
str2num(get(handles.edit149,'strin
g'))*str2num(get(handles.edit96,'s
tring'))
es=str2num(get(handles.edit91,'str
ing'));
Eyd=str2num(get(handles.edit77,'st
ring'));
if es>Eyd
disp(Fs_in_N);
set(handles.edit18,'string',Fs_in_
N);
elseif es<Eyd
disp(Fs1);
set(handles.edit18,'string',Fs1);
end
asf=0;
asw=0;
mrdf=0;
mrdw=0;
mrdt=0;
x=str2num(get(handles.edit24,'stri
ng'));
hf=str2num(get(handles.edit6,'stri
ng'));
MRdr_in_Nmm
=(str2num(get(handles.edit80,'stri
ng'))+str2num(get(handles.edit94,'
string'))+str2num(get(handles.edit
96,'string')))*str2num(get(handles
.edit39,'string'))*str2num(get(han
dles.edit5,'string'))*(1-
str2num(get(handles.edit86,'string
')))
Mrdrr=0;
if x>hf
disp(Mrdrr) ;
set(handles.edit25,'string',Mrdrr)
;
```

```
elseif x<hf
disp(MRdr_in_Nmm);
set(handles.edit25,'string',MRdr_i
n_Nmm);
disp(asf);
set(handles.edit1,'string',asf);
disp(asw);
set(handles.edit2,'string',asw);
disp(mrdf);
set(handles.edit8,'string',mrdf);
disp(mrdw);
set(handles.edit9,'string',mrdw);
disp(mrdt);
set(handles.edit113,'string',mrdt)
;
end
asf=0;
asw=0;
mrdf=0;
mrdw=0;
mrdt=0;
x=str2num(get(handles.edit24,'stri
ng'));
hf=str2num(get(handles.edit6,'stri
ng'));
MRdr_in_Nmm
=(str2num(get(handles.edit80,'stri
ng'))+str2num(get(handles.edit94,'
string'))+str2num(get(handles.edit
96,'string')))*str2num(get(handles
.edit39,'string'))*str2num(get(han
dles.edit5,'string'))*(1-
str2num(get(handles.edit86,'string
')))
Mrdrr=0;
if x>hf
Asf_in_mm2
=str2num(get(handles.edit38,'strin
g'))*(str2num(get(handles.edit70,'
string'))-
str2num(get(handles.edit4,'string'
)))*str2num(get(handles.edit6,'str
ing'))/(str2num(get(handles.edit39
,'string')))
set(handles.edit1,'string',Asf_in_
mm2);
C='increase As';
Asw_in_mm2=(str2num(get(handles.ed
it80,'string'))+str2num(get(handle
s.edit94,'string'))+str2num(get(ha
ndles.edit96,'string')))-
str2num(get(handles.edit1,'string'
))
set(handles.edit2,'string',Asw_in_
mm2);
if Asw_in_mm2 > 0
    disp (Asw_in_mm2);
set(handles.edit2,'string',Asw_in_
mm2);
elseif Asw_in_mm2<0
    disp(C);
set(handles.edit2,'string',C);
end
MRdf_in_Nmm
=str2num(get(handles.edit1,'string
'))*str2num(get(handles.edit39,'st
ring'))*(str2num(get(handles.edit5
,'string'))-
0.5*str2num(get(handles.edit6,'str
ing')))
set(handles.edit8,'string',MRdf_in
_Nmm);
```

```
MRdw_in_Nmm=str2num(get(handles.edit2,'string'))*str2num(get(handles.edit39,'string'))*str2num(get(handles.edit5,'string'))*(1-str2num(get(handles.edit86,'string')))
set(handles.edit9,'string',MRdw_in_Nmm);
MRdt_in_Nmm=str2num(get(handles.edit8,'string'))+str2num(get(handles.edit9,'string'))
set(handles.edit113,'string',MRdt_in_Nmm);
disp(Mrdrr) ;
set(handles.edit25,'string',Mrdrr);
elseif x<hf
disp(MRdr_in_Nmm);
set(handles.edit25,'string',MRdr_in_Nmm);
disp(asf);
set(handles.edit1,'string',asf);
disp(asw);
set(handles.edit2,'string',asw);
disp(mrdf);
set(handles.edit8,'string',mrdf);
disp(mrdw);
set(handles.edit9,'string',mrdw);
disp(mrdt);
set(handles.edit113,'string',mrdt);
end
popupmenu14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
function pushbutton73_Callback(hObject, eventdata, handles)
Kxlim=str2num(get(handles.edit209,'string'));
bw=str2num(get(handles.edit4,'string'));
be=str2num(get(handles.edit70,'string'));
fyk=str2num(get(handles.edit130,'string'));
fcd=str2num(get(handles.edit38,'string'));
x=str2num(get(handles.edit24,'string'));
hf=str2num(get(handles.edit6,'string'));
M=str2num(get(handles.edit136,'string'));
l=str2num(get(handles.edit170,'string'));
d=str2num(get(handles.edit5,'string'));
k=str2num(get(handles.edit172,'string'));
m=((fyk/1.15)/(0.8*fcd));
pr1=(1.25/m)*(1-sqrt(1-(2*M)/(fcd*be*d^2)));
Mdf=fcd*(be-bw)*hf*(d-0.5*hf);
Ast=str2num(get(handles.edit80,'string'))+str2num(get(handles.edit94
```

```
,'string'))+str2num(get(handles.ed
it96,'string'));
Asf_req_in_mm2=(fcd*(be-
bw)*hf)/((fyk/1.15))
Asw_req_in_mm2=(M-
Mdf)/((fyk/1.15)*(d-0.4*x))
Asw1=0;
Mweb=M-Mdf
Mweb1=0;
if x>hf
disp(Asw_req_in_mm2);
set(handles.edit215,'string',Asw_r
eq_in_mm2);
set(handles.edit250,'string',Mweb)
;
elseif x<hf
disp(Asw1);
disp(Mweb1)
set(handles.edit215,'string',Asw1)
;
set(handles.edit250,'string',Mweb1
);
set(handles.edit251,'string',Mweb1
)
set(handles.edit252,'string',Mweb1
);
set(handles.edit254,'string',Mweb1
);
end
Ms_in_Nmm=Kxlim*(8.5/10.5)*fcd*bw*
d^2*(1-Kxlim*(24.75/59.5));
Mweb2=Mweb-Ms_in_Nmm
pw=(1.25/m)*(1-sqrt(1-
((2*Mweb2)/(fcd*bw*d^2))));
Aswebcompression=pw*bw*d
Nobar=Aswebcompression/str2num(get
(handles.edit21,'string'));
AA= 'compression reinforcement
require'
AAA=0;
if Mweb>Ms_in_Nmm
    disp(AA)
set(handles.edit250,'string',Mweb)
;
set(handles.edit251,'string',AA);
    disp(Aswebcompression)
set(handles.edit252,'string',Asweb
compression);
set(handles.edit254,'string',Nobar
);
else
set(handles.edit251,'string',AAA);
set(handles.edit252,'string',AAA);
set(handles.edit250,'string',AAA);
set(handles.edit254,'string',AAA);
end
As_req_total_in_mm2=Asf_req_in_mm2
+Asw_req_in_mm2;
As_req1_in_mm2=pr1*be*d;
if x<hf
disp(As_req1_in_mm2);
set(handles.edit207,'string',As_re
q1_in_mm2);
elseif x>hf
disp(As_req_total_in_mm2);
set(handles.edit207,'string',As_re
q_total_in_mm2);
end
dia=str2num(get(handles.edit201,'s
tring'));
Asreq11=str2num(get(handles.edit20
7,'string'));
Asunit=0.25*pi*(dia)^2;
Noofbar=Asreq11/Asunit;
```

```
set(handles.edit208,'string',Noofb
ar)
if (be/bw)>3
f3=0.8;
elseif (be/bw)<3
f3=1;
else
end
if x>hf
f1=min(Ast/As_req_total_in_mm2,1.5
);
elseif x<hf
f1=min(Ast/As_req1_in_mm2,1.5);
else
end
if x> hf
p=((As_req_total_in_mm2)/(str2num(
get(handles.edit4,'string'))*str2n
um(get(handles.edit5,'string'))));
elseif x<hf
p=((As_req1_in_mm2)/(str2num(get(h
andles.edit70,'string'))*str2num(g
et(handles.edit5,'string'))));
end
if l>=7000
f2=7000/l;
elseif l<=7000
f2=1;
else
end
po=0.001*sqrt(str2num(get(handles.
edit132,'string')));
a=l/d;
if p<=po
b=k*(11+1.5*sqrt(str2num(get(handl
es.edit132,'string')))*po/p+3.2*sq
rt(str2num(get(handles.edit132,'st
ring')))*(po/p-
1)^1.5)*(500/fyk)*f1*f2*f3;
elseif p>po
b=k*(11+1.5*sqrt(str2num(get(handl
es.edit132,'string')))*po/p)*(500/
fyk)*f1*f2*f3;
end
design_criteria4='OK'
design_criteria5='NOT OK'
if a<b
disp(design_criteria4);
set(handles.edit171,'string',desig
n_criteria4);
else
disp(design_criteria5);
set(handles.edit171,'string',desig
n_criteria5);
end
kx= [0.4159 -1
(1.235*M_in_Nmm)/(str2num(get(hand
les.edit38,'string'))*str2num(get(
handles.edit70,'string'))*(str2num
(get(handles.edit5,'string'))^2))]
pkx = roots(kx);
kx11=pkx(1,1);
kx12=pkx(2,1);
kx13=[]
if kx11>0&&kx11< Kxlim
disp(kx11);
set(handles.edit500,'string',kx11)
;
elseif kx12>0&&kx12< Kxlim
disp(kx12);
set(handles.edit500,'string',kx12)
;
elseif kx11> Kxlim
disp(kx13);
```

```
set(handles.edit500,'string',kx13)
;
elseif kx12> Kxlim
disp(kx13);
set(handles.edit500,'string',kx13)
;
end
Kx=str2num(get(handles.edit500,'st
ring'));
design_criteria6='OK '
design_criteria7='NOT OK'
if Kx<Kxlim
disp(design_criteria6);
set(handles.edit134,'string',desig
n_criteria6);
else
disp(design_criteria7)
set(handles.edit134,'string',desig
n_criteria7);
end
Asmin=max((0.26*str2num(get(handle
s.edit131,'string'))*str2num(get(h
andles.edit4,'string')))/str2num(g
et(handles.edit130,'string')),0.00
13*str2num(get(handles.edit4,'stri
ng'))*str2num(get(handles.edit5,'s
tring')));
mx=0.04*str2num(get(handles.edit4,
'string'))*str2num(get(handles.edi
t84,'string'))+Asf_req_in_mm2;
mx1=0.04*str2num(get(handles.edit8
4,'string'))*str2num(get(handles.e
dit70,'string'));
if x>hf
Asmax=mx;
elseif x<hf
Asmax=mx1;
end
Ast=str2num(get(handles.edit80,'st
ring'))+str2num(get(handles.edit94
,'string'))+str2num(get(handles.ed
it96,'string'));
design_criteria8='OK'
design_criteria9='NOT OK'
if Ast>Asmin
disp(design_criteria8);
set(handles.edit133,'string',desig
n_criteria8);
else
disp(design_criteria9);
set(handles.edit133,'string',desig
n_criteria9);
end
if Ast<Asmax
disp(design_criteria8);
set(handles.edit133,'string',desig
n_criteria8);
elseif Ast>Asmax
disp(design_criteria9);
set(handles.edit133,'string',desig
n_criteria9);
end
MEd=str2num(get(handles.edit136,'s
tring'));
MRdr=str2num(get(handles.edit25,'s
tring'));
MRdt=str2num(get(handles.edit113,'
string'));
design_criteria10='OK'
design_criteria11='NOT OK'
if MRdr>MEd
disp(design_criteria10);
set(handles.edit135,'string',desig
n_criteria10);
```

```
elseif MRdt>MEd
disp(design_criteria10);
set(handles.edit135,'string',desig
n_criteria10);
else
disp(design_criteria11);
set(handles.edit135,'string',desig
n_criteria11);
end
function
popupmenu18_Callback(hObject,
eventdata, handles)
switch
get(handles.popupmenu18,'Value')
case 1
    d = 'beam support cond';
    k=0;
case 2
    a = 'simply supported beam';
    k=1;
case 3
    b = 'interior span beam';
    k=1.5;
case 4
    c = 'end span beam';
    k=1.3;
case 5
    d = 'cantliver beam';
    k=0.4;
otherwise
end
set(handles.edit172,'string',k);
function
popupmenu18_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
pushbutton75_Callback(hObject,
eventdata, handles)
c=[];
set(handles.edit65,'string',c);
set(handles.edit25,'string',c);
set(handles.edit136,'string',c);
set(handles.edit170,'string',c);
set(handles.edit171,'string',c);
set(handles.edit207,'string',c);
set(handles.edit208,'string',c);
set(handles.edit172,'string',c);
set(handles.edit134,'string',c);
set(handles.edit135,'string',c);
set(handles.edit133,'string',c);
set(handles.edit500,'string',c);
set(handles.edit130,'string',c);
set(handles.edit131,'string',c);
set(handles.edit132,'string',c);
set(handles.edit38,'string',c);
set(handles.edit39,'string',c);
set(handles.edit200,'string',c);
set(handles.edit21,'string',c);
set(handles.edit5,'string',c);
set(handles.edit84,'string',c);
set(handles.edit4,'string',c);
set(handles.edit70,'string',c);
set(handles.edit69,'string',c);
set(handles.edit68,'string',c);
set(handles.edit77,'string',c);
set(handles.edit91,'string',c);
```

```
set(handles.edit86,'string',c);
set(handles.edit83,'string',c);
set(handles.edit23,'string',c);
set(handles.edit78,'string',c);
set(handles.edit72,'string',c);
set(handles.edit24,'string',c);
set(handles.edit255,'string',c);
set(handles.edit107,'string',c);
set(handles.edit105,'string',c);
set(handles.edit106,'string',c);
set(handles.edit7,'string',c);
set(handles.edit18,'string',c);
set(handles.edit1,'string',c);
set(handles.edit2,'string',c);
set(handles.edit8,'string',c);
set(handles.edit9,'string',c);
set(handles.edit113,'string',c);
set(handles.edit215,'string',c);
set(handles.edit250,'string',c);
set(handles.edit252,'string',c);
set(handles.edit251,'string',c);
set(handles.edit254,'string',c);
set(handles.edit80,'string',d);
set(handles.edit94,'string',d);
set(handles.edit96,'string',d);
set(handles.edit201,'string',d);
set(handles.edit202,'string',d);
set(handles.edit203,'string',d);
set(handles.edit6,'string',d);
set(handles.edit97,'string',d);
set(handles.edit98,'string',d);
set(handles.edit99,'string',d);
set(handles.edit48,'string',d);
set(handles.edit49,'string',d);
set(handles.edit50,'string',d);
set(handles.edit65,'string',d);
set(handles.edit55,'string',d);
set(handles.edit56,'string',d);
```

4. Source Code for Beam Interface for Shear

```
handles.output = hObject;
axes(handles.axes1)
imshow('beam section for
shear.PNG');title('Truss model and
notation for shear reinforced')
guidata(hObject, handles);
function
popupmenu1_Callback(hObject,
eventdata, handles)
switch
get(handles.popupmenu1,'Value')
case 1
steelclass= 'select steel grade';
fyk_in_MPa=0;
fywd_in_MPa=0;
case 2
steelclass= '260';
fyk_in_MPa=260;
fywd_in_MPa=260/1.15;
case 3
steelclass= '300';
fyk_in_MPa=300;
fywd_in_MPa=300/1.15;
case 4
steelclass= '360';
fyk_in_MPa=360;
fywd_in_MPa=360/1.15;
case 5
steelclass= '400';
fyk_in_MPa=400;
fywd_in_MPa=400/1.15;
case 6
steelclass= '460';
fyk_in_MPa=460;
fywd_in_MPa=460/1.15;
case 7
steelclass= '500';
fyk_in_MPa=500;
fywd_in_MPa=500/1.15;
case 8
steelclass= '560';
fyk_in_MPa=560;
fywd_in_MPa=560/1.15;
case 9
steelclass= '600';
fyk_in_MPa=600;
fywd_in_MPa=600/1.15;
otherwise
end
set(handles.edit130,'string',fyk_i
n_MPa);
set(handles.edit39,'string',fywd_i
n_MPa);
function
popupmenu1_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
popupmenu2_Callback(hObject,
eventdata, handles)
switch
get(handles.popupmenu2,'Value')
case 1
concreteclass= 'select concrete
grade';
```

```
fck_in_MPa=0;
fcd_in_MPa=0;
case 2
concreteclass= '15';
fck_in_MPa=0.8*15
fcd_in_MPa=0.8*0.85*15/1.5
case 3
concreteclass= '20';
fck_in_MPa=0.8*20
fcd_in_MPa=0.8*0.85*20/1.5
case 4
concreteclass= '25';
fck_in_MPa=0.8*25
fcd_in_MPa=0.8*0.85*25/1.5
case 5
concreteclass= '30';
fck_in_MPa=0.8*30
fcd_in_MPa=0.8*0.85*30/1.5
case 6
concreteclass= '37';
fck_in_MPa=0.8*37
fcd_in_MPa=0.8*0.85*37/1.5
case 7
concreteclass= '45';
fck_in_MPa=0.8*45
fcd_in_MPa=0.8*0.85*45/1.5
case 8
concreteclass= '50';
fck_in_MPa=0.8*50
fcd_in_MPa=0.8*0.85*50/1.5
otherwise
end
set(handles.edit24,'string',fck_in
_MPa);
set(handles.edit38,'string',fcd_in
_MPa);
function
popupmenu2_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function edit24_Callback(hObject,
eventdata, handles)
function edit24_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
popupmenu3_Callback(hObject,
eventdata, handles)
switch
get(handles.popupmenu3,'Value')
case 1
Diameter= 'select bar size';
bar_diametr_in_mm=0;
unit_area_of_reinforcement_in_mm2=
0;
case 2
Diameter= '6';
bar_diametr_in_mm=6;
```

```
unit_area_of_reinforcement_in_mm2=
(pi*6^2)/(4);
case 3
Diameter= '8';
bar_diametr_in_mm=8;
unit_area_of_reinforcement_in_mm2=
(pi*8^2)/(4);
case 4
Diameter= '10';
bar_diametr_in_mm=10;
unit_area_of_reinforcement_in_mm2=
(pi*10^2)/(4);
case 5
Diameter= '12';
bar_diametr_in_mm=12;
unit_area_of_reinforcement_in_mm2=
(pi*12^2)/(4);
case 6
Diameter= '14';
bar_diametr_in_mm=14;
unit_area_of_reinforcement_in_mm2=
(pi*14^2)/(4);
case 7
Diameter= '16';
bar_diametr_in_mm=16;
unit_area_of_reinforcement_in_mm2=
(pi*16^2)/(4);
case 8
Diameter= '18';
bar_diametr_in_mm=18;
unit_area_of_reinforcement_in_mm2=
(pi*18^2)/(4);
case 9
Diameter= '20';
bar_diametr_in_mm=20;
unit_area_of_reinforcement_in_mm2=
(pi*20^2)/(4);
case 10
Diameter= '22';
bar_diametr_in_mm=22;
unit_area_of_reinforcement_in_mm2=
(pi*22^2)/(4);
case 11
Diameter= '24';
bar_diametr_in_mm=24;
unit_area_of_reinforcement_in_mm2=
(pi*24^2)/(4);
case 12
Diameter= '26';
bar_diametr_in_mm=26;
unit_area_of_reinforcement_in_mm2=
(pi*26^2)/(4);
case 13
Diameter= '30';
bar_diametr_in_mm=30;
unit_area_of_reinforcement_in_mm2=
(pi*30^2)/(4);
case 14
Diameter= '32';
bar_diametr_in_mm=32;
unit_area_of_reinforcement_in_mm2=
(pi*32^2)/(4);
case 15
Diameter= '34';
bar_diametr_in_mm=34;
unit_area_of_reinforcement_in_mm2=
(pi*34^2)/(4);
case 16
Diameter= '40';
bar_diametr_in_mm=40;
unit_area_of_reinforcement_in_mm2=
(pi*40^2)/(4);
otherwise
end
```

```
set(handles.edit21,'string',unit_a
rea_of_reinforcement_in_mm2);
set(handles.edit200,'string',bar_d
iametr_in_mm);
function
popupmenu3_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
pushbutton1_Callback(hObject,
eventdata, handles)
As_provide_at_tension_zone=str2num
(get(handles.edit21,'string'))*str
2num(get(handles.edit79,'string'))
set(handles.edit80,'string',As_pro
vide_at_tension_zone);
number_of_bar_provided_at_tension_
zone=str2num(get(handles.edit79,'s
tring'))
diameter_of_tension_bar_in_mm=str2
num(get(handles.edit200,'string'))
diameter=0;
if
number_of_bar_provided_at_tension_
zone>0
disp(diameter_of_tension_bar_in_mm
)
set(handles.edit201,'string',diame
ter_of_tension_bar_in_mm);
elseif
number_of_bar_provided_at_tension_
zone==0
    disp(0)
set(handles.edit201,'string',diame
ter);
end
function
pushbutton2_Callback(hObject,
eventdata, handles)
Asw_provided_fof_shear=str2num(get
(handles.edit21,'string'))*str2num
(get(handles.edit52,'string'))
diameter_of_stirrup_in_mm=str2num(
get(handles.edit200,'string'))
diameter=0;
Asw_provided_fof_shear1=0;
if diameter_of_stirrup_in_mm==6
disp(diameter_of_stirrup_in_mm)
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
elseif
diameter_of_stirrup_in_mm==8
disp(diameter_of_stirrup_in_mm)
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
    elseif
diameter_of_stirrup_in_mm==10
disp(diameter_of_stirrup_in_mm)
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
```

```
    elseif
diameter_of_stirrup_in_mm==12
disp(diameter_of_stirrup_in_mm)
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
else
    disp(diameter);
set(handles.edit53,'string',Asw_pr
ovided_fof_shear1);
set(handles.edit202,'string',diame
ter);
end
function
pushbutton3_Callback(hObject,
eventdata, handles)
D_in_mm=str2num(get(handles.edit84
,'string'))
d_in_mm=D_in_mm-
(str2num(get(handles.edit58,'strin
g'))+0.5*str2num(get(handles.edit2
01,'string'))+str2num(get(handles.
edit202,'string')))
set(handles.edit5,'string',d_in_mm
);
d=str2num(get(handles.edit5,'strin
g'));
k=1+sqrt(200/d)
kmax=2
if k<=2
    disp(k)
set(handles.edit31,'string',k)
elseif k>2
    disp(kmax)
set(handles.edit31,'string',kmax)
end
p1=str2num(get(handles.edit80,'str
ing'))/(str2num(get(handles.edit4,
'string'))*str2num(get(handles.edi
t5,'string')))
p1max=0.02
if p1<p1max
    disp(p1)
set(handles.edit37,'string',p1)
elseif p1>p1max
    disp(p1max)
set(handles.edit37,'string',p1max)
;
end
crdc=0.18/1.5
vmin=0.035*str2num(get(handles.edi
t31,'string'))^1.5*str2num(get(han
dles.edit24,'string'))^0.5
Vmin=vmin*str2num(get(handles.edit
4,'string'))*str2num(get(handles.e
dit5,'string'))
VRdc=(crdc*str2num(get(handles.edi
t31,'string'))*(100*str2num(get(ha
ndles.edit37,'string'))*str2num(ge
t(handles.edit24,'string')))^(1/3)
)*str2num(get(handles.edit4,'strin
g'))*str2num(get(handles.edit5,'st
ring'))
if VRdc>Vmin
    disp(VRdc)
set(handles.edit49,'string',VRdc)
elseif VRdc<Vmin
    disp(Vmin)
set(handles.edit49,'string',Vmin)
end
function
pushbutton15_Callback(hObject,
eventdata, handles)
```

```
sinalpha_in_rad=sind(str2num(get(h
andles.edit32,'string')))
set(handles.edit68,'string',sinalp
ha_in_rad);
cotalpha_in_rad=cotd(str2num(get(h
andles.edit32,'string')))
set(handles.edit2,'string',cotalph
a_in_rad);
z=0.9*str2num(get(handles.edit5,'s
tring'))
set(handles.edit40,'string',z);
alpha_in_deg=str2num(get(handles.e
dit32,'string'))
VRds=(str2num(get(handles.edit53,'
string'))*str2num(get(handles.edit
40,'string'))*str2num(get(handles.
edit39,'string'))*str2num(get(hand
les.edit47,'string')))/str2num(get
(handles.edit48,'string'));
VRds1=(str2num(get(handles.edit53,
'string'))*str2num(get(handles.edi
t40,'string'))*str2num(get(handles
.edit39,'string'))*(str2num(get(ha
ndles.edit47,'string'))+str2num(ge
t(handles.edit2,'string')))*str2nu
m(get(handles.edit68,'string')))/s
tr2num(get(handles.edit48,'string'
));
if alpha_in_deg==90
    disp(VRds)
set(handles.edit50,'string',VRds);
elseif alpha_in_deg<90
    disp(VRds1)
set(handles.edit50,'string',VRds1)
;
end
acw=1%for non prestressed member
v=0.6%for fck<60MPa
alpha1=str2num(get(handles.edit32,
'string'));
VRdmax_in_N=(acw*str2num(get(handl
es.edit4,'string'))*str2num(get(ha
ndles.edit40,'string'))*v*str2num(
get(handles.edit38,'string')))/(st
r2num(get(handles.edit47,'string')
)+(1/str2num(get(handles.edit47,'s
tring'))))
VRdmax1_in_N=(acw*str2num(get(hand
les.edit4,'string'))*str2num(get(h
andles.edit40,'string'))*v*str2num
(get(handles.edit38,'string'))*(1/
str2num(get(handles.edit2,'string'
))))/(str2num(get(handles.edit47,'
string'))+(1/str2num(get(handles.e
dit47,'string'))))
 if alpha1==90
     disp(VRdmax_in_N)
set(handles.edit46,'string',VRdmax
_in_N)
 elseif alpha1<90
     disp(VRdmax1_in_N)
set(handles.edit46,'string',VRdmax
1_in_N)
 end
VRds11=str2num(get(handles.edit50,
'string'));
VRdmax_in_N=str2num(get(handles.ed
it46,'string'));
if VRdmax_in_N<VRds11
    disp(VRdmax_in_N)
set(handles.edit45,'string',VRdmax
_in_N)
elseif VRds11<VRdmax_in_N
    disp(VRds11)
```

```
set(handles.edit45,'string',VRds11
)
end
function
pushbutton22_Callback(hObject,
eventdata, handles)
VEd_in_N=str2num(get(handles.edit1
31,'string'))
Vrd_in_N=str2num(get(handles.edit4
5,'string'))
design_criteria1=('OK');
design_criteria2=('EVALUATE teta
again');
if Vrd_in_N>VEd_in_N
    disp(design_criteria1)
set(handles.edit134,'string',desig
n_criteria1);
else
    disp(design_criteria2)
set(handles.edit134,'string',desig
n_criteria2);
end
acw=1%for non prestressed member
v=0.6 %for fck<60MPa
teta=0.5*(asin((2*str2num(get(hand
les.edit131,'string')))/(acw*str2n
um(get(handles.edit4,'string'))*st
r2num(get(handles.edit40,'string')
)*v*str2num(get(handles.edit38,'st
ring'))))));
if
str2num(get(handles.edit46,'string
'))>str2num(get(handles.edit131,'s
tring'))
b=2.5;
set(handles.edit132,'string',b);
elseif
str2num(get(handles.edit46,'string
'))<str2num(get(handles.edit131,'s
tring'))
    b=cot(teta);
set(handles.edit132,'string',b);
end
aa=str2num(get(handles.edit132,'st
ring'));
a=1;
b1=2.5;
design_criteria3=('OK');
design_criteria4=('NOT OK');
if aa>= a && aa<= b1
    disp(design_criteria3)
set(handles.edit133,'string',desig
n_criteria3);
elseif aa<=a && aa>=b1
    disp(design_criteria4)
set(handles.edit133,'string',desig
n_criteria4);
end
acw=1;%for non prestressed member
v=0.6; %for fck<60MPa
alpha=str2num(get(handles.edit32,'
string'));
pwmin=(0.08*sqrt(str2num(get(handl
es.edit24,'string'))))/(str2num(ge
t(handles.edit130,'string')));
Aswmin=pwmin*str2num(get(handles.e
dit5,'string'))*str2num(get(handle
s.edit4,'string'));
Asmax=0.5*acw*v*str2num(get(handle
s.edit38,'string'))*sind(alpha)*st
r2num(get(handles.edit4,'string'))
*str2num(get(handles.edit24,'strin
g'))*str2num(get(handles.edit48,'s
```

```
tring'))/((1-
cosd(alpha))*str2num(get(handles.e
dit39,'string')));
;
design_criteria5=('OK');
design_criteria6=('NOT OK');
Asw=str2num(get(handles.edit53,'st
ring'));
if Asw>Aswmin && Asw < Asmax
    disp(design_criteria5)
set(handles.edit137,'string',desig
n_criteria5);
else
    disp(design_criteria6)
set(handles.edit137,'string',desig
n_criteria6);
end
Smax=0.75*str2num(get(handles.edit
5,'string'))*(1+str2num(get(handle
s.edit2,'string')));
S=str2num(get(handles.edit48,'stri
ng'));
Sd=min([Smax,S]);
set(handles.edit136,'string',Sd);
deltaFtd=0.5*str2num(get(handles.e
dit131,'string'))*(str2num(get(han
dles.edit47,'string'))-
str2num(get(handles.edit2,'string'
)));
set(handles.edit180,'string',delta
Ftd);
function
pushbutton28_Callback(hObject,
eventdata, handles)
c=[];
d=[0];
set(handles.edit58,'string',c);
set(handles.edit4,'string',c);
set(handles.edit84,'string',c);
set(handles.edit5,'string',c);
set(handles.edit31,'string',c);
set(handles.edit50,'string',c);
set(handles.edit46,'string',c);
set(handles.edit45,'string',c);
set(handles.edit130,'string',c);
set(handles.edit137,'string',c);
set(handles.edit131,'string',c);
set(handles.edit132,'string',c);
set(handles.edit133,'string',c);
set(handles.edit24,'string',c);
set(handles.edit38,'string',c);
set(handles.edit39,'string',c);
set(handles.edit48,'string',c);
set(handles.edit68,'string',c);
set(handles.edit2,'string',c);
set(handles.edit40,'string',c);
set(handles.edit49,'string',c);
set(handles.edit200,'string',c);
set(handles.edit21,'string',c);
set(handles.edit134,'string',c);
set(handles.edit136,'string',c);
set(handles.edit37,'string',c);
set(handles.edit180,'string',c);
set(handles.edit79,'string',d);
set(handles.edit80,'string',d);
set(handles.edit53,'string',d);
set(handles.edit201,'string',d);
set(handles.edit202,'string',d);
```

5. Source Code for Beam Interface for Torsion

```
function edit39_Callback(hObject,
eventdata, handles)
function edit39_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
pushbutton13_Callback(hObject,
eventdata, handles)
As_provide_at_tension_zone=str2num
(get(handles.edit21,'string'))*str
2num(get(handles.edit79,'string'))
set(handles.edit80,'string',As_pro
vide_at_tension_zone);
number_of_bar_provided_at_tension_
zone=str2num(get(handles.edit79,'s
tring'))
diameter_of_tension_bar_in_mm=str2
num(get(handles.edit200,'string'))
diameter=0;
if
number_of_bar_provided_at_tension_
zone>0
disp(diameter_of_tension_bar_in_mm
)
set(handles.edit201,'string',diame
ter_of_tension_bar_in_mm);
elseif
number_of_bar_provided_at_tension_
zone==0
    disp(0)
set(handles.edit201,'string',diame
ter);
end
Asw_provided_fof_shear=str2num(get
(handles.edit21,'string'))*str2num
(get(handles.edit52,'string'))
diameter_of_stirrup_in_mm=str2num(
get(handles.edit200,'string'))
diameter=0;
Asw_provided_fof_shear1=0;
if diameter_of_stirrup_in_mm==6
disp(diameter_of_stirrup_in_mm)
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
elseif
diameter_of_stirrup_in_mm==8
disp(diameter_of_stirrup_in_mm)
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
    elseif
diameter_of_stirrup_in_mm==10
disp(diameter_of_stirrup_in_mm)
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
    elseif
diameter_of_stirrup_in_mm==12
disp(diameter_of_stirrup_in_mm)
set(handles.edit53,'string',Asw_pr
ovided_fof_shear);
```

```
set(handles.edit202,'string',diame
ter_of_stirrup_in_mm);
else
    disp(diameter);
set(handles.edit53,'string',Asw_pr
ovided_fof_shear1);
set(handles.edit202,'string',diame
ter);
end
function
pushbutton25_Callback(hObject,
eventdata, handles)
D_in_mm=str2num(get(handles.edit14
,'string'))
d_in_mm=D_in_mm-
(str2num(get(handles.edit58,'strin
g'))+0.5*str2num(get(handles.edit2
01,'string'))+str2num(get(handles.
edit202,'string')))
set(handles.edit84,'string',d_in_m
m);
t_in_mm=(str2num(get(handles.edit4
,'string'))*str2num(get(handles.ed
it14,'string')))/(2*(str2num(get(h
andles.edit4,'string'))+str2num(ge
t(handles.edit14,'string'))))
t1=2*str2num(get(handles.edit58,'s
tring'));
if t_in_mm>t1
    disp(t_in_mm)
set(handles.edit5,'string',t_in_mm
);
elseif t_in_mm<t1
    disp(t1)
set(handles.edit5,'string',t1);
end
Ak_in_mm2=(str2num(get(handles.edi
t4,'string'))-
str2num(get(handles.edit5,'string'
)))*(str2num(get(handles.edit14,'s
tring'))-
str2num(get(handles.edit5,'string'
)));
set(handles.edit11,'string',Ak_in_
mm2);
Uk_in_mm=2*((str2num(get(handles.e
dit4,'string'))-
str2num(get(handles.edit5,'string'
)))+(str2num(get(handles.edit14,'s
tring'))-
str2num(get(handles.edit5,'string'
))));
set(handles.edit31,'string',Uk_in_
mm);
TRdc_in_Nmm=2*str2num(get(handles.
edit56,'string'))*str2num(get(hand
les.edit5,'string'))*str2num(get(h
andles.edit11,'string'))
set(handles.edit49,'string',TRdc_i
n_Nmm);
function
pushbutton20_Callback(hObject,
eventdata, handles)
sinalpha_in_rad=sind(str2num(get(h
andles.edit32,'string')))
set(handles.edit68,'string',sinalp
ha_in_rad);
cotalpha_in_rad=cotd(str2num(get(h
andles.edit32,'string')))
set(handles.edit2,'string',cotalph
a_in_rad);
p=
[(str2num(get(handles.edit53,'stri
```

```
ng'))*str2num(get(handles.edit39,'
string'))*str2num(get(handles.edit
68,'string'))*str2num(get(handles.
edit11,'string')))/(str2num(get(ha
ndles.edit48,'string')))
(str2num(get(handles.edit53,'strin
g'))*str2num(get(handles.edit39,'s
tring'))*str2num(get(handles.edit2
,'string'))*str2num(get(handles.ed
it68,'string'))*str2num(get(handle
s.edit11,'string')))/(str2num(get(
handles.edit48,'string'))) (-
2*str2num(get(handles.edit80,'stri
ng'))*str2num(get(handles.edit55,'
string'))*str2num(get(handles.edit
11,'string')))/(str2num(get(handle
s.edit31,'string')))];
pteta = roots(p);
design_criteria='Change s';
teta11_in_rad=pteta(1,1);
teta12_in_rad=pteta(2,1);
if
teta11_in_rad>=1&&teta11_in_rad<=2
.5
disp(teta11_in_rad);
set(handles.edit40,'string',teta11
_in_rad);
elseif
teta12_in_rad>=1&&teta12_in_rad<=2
.5
    disp(teta12_in_rad);
set(handles.edit40,'string',teta12
_in_rad);
else
set(handles.edit40,'string',design
_criteria);
end
```

```
teta_in_deg=acotd(str2num(get(hand
les.edit40,'string')));
set(handles.edit89,'string',teta_i
n_deg);
sinteta_in_rad=sind(str2num(get(ha
ndles.edit89,'string')));
set(handles.edit90,'string',sintet
a_in_rad);
function edit68_Callback(hObject,
eventdata, handles)
function edit68_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject,'BackgroundCol
or'),
get(0,'defaultUicontrolBackgroundC
olor'))
set(hObject,'BackgroundColor','whi
te');
end
function
pushbutton16_Callback(hObject,
eventdata, handles)
TRds=(str2num(get(handles.edit53,'
string'))*str2num(get(handles.edit
39,'string'))*str2num(get(handles.
edit40,'string'))*str2num(get(hand
les.edit11,'string')))/(str2num(ge
t(handles.edit48,'string')));
TRds1=(str2num(get(handles.edit53,
'string'))*str2num(get(handles.edi
t39,'string'))*(str2num(get(handle
s.edit2,'string'))+str2num(get(han
dles.edit40,'string')))*str2num(ge
t(handles.edit68,'string'))*str2nu
m(get(handles.edit11,'string')))/(
```

```
str2num(get(handles.edit48,'string
')));
alpha_in_rad=str2num(get(handles.e
dit32,'string'));
if alpha_in_rad==90
    disp(TRds)
set(handles.edit50,'string',TRds);
elseif alpha_in_rad<90
    disp(TRds1)
set(handles.edit50,'string',TRds1)
;
end
a=1;
TRdmax=2*(str2num(get(handles.edit
5,'string'))*a*str2num(get(handles
.edit11,'string'))*str2num(get(han
dles.edit38,'string'))*0.6*(1-
str2num(get(handles.edit24,'string
'))/250)*(str2num(get(handles.edit
40,'string')))*(str2num(get(handle
s.edit90,'string')))^2)
TRdmax1=2*(str2num(get(handles.edi
t5,'string'))*a*str2num(get(handle
s.edit11,'string'))*str2num(get(ha
ndles.edit38,'string'))*0.6*(1-
str2num(get(handles.edit24,'string
'))/250)*(str2num(get(handles.edit
40,'string'))+str2num(get(handles.
edit2,'string')))*(str2num(get(han
dles.edit90,'string')))^2)
alpha_in_rad=str2num(get(handles.e
dit32,'string'))
if alpha_in_rad==90
    disp(TRdmax)
set(handles.edit46,'string',TRdmax
);
elseif alpha_in_rad<90
    disp(TRdmax1)
set(handles.edit46,'string',TRdmax
1);
end
TRds=str2num(get(handles.edit50,'s
tring'));
TRdmax=str2num(get(handles.edit46,
'string'));
if TRdmax<TRds
    disp(TRdmax)
set(handles.edit45,'string',TRdmax
)
elseif TRds<TRdmax
    disp(TRds)
set(handles.edit45,'string',TRds)
end
function edit89_Callback(hObject,
eventdata, handles)
function
pushbutton41_Callback(hObject,
eventdata, handles)
pwmin=(0.08*sqrt(str2num(get(handl
es.edit24,'string'))))/(str2num(ge
t(handles.edit66,'string')));
Aswmin=pwmin*str2num(get(handles.e
dit4,'string'))*str2num(get(handle
s.edit84,'string'));
set(handles.edit165,'string',Aswmi
n)
TEd_in_Nmm=str2num(get(handles.edi
t130,'string'))
Trd=str2num(get(handles.edit45,'st
ring'));
design_criteria1=('OK');
design_criteria2=('NOT OK');
if Trd>TEd_in_Nmm
    disp(design_criteria1)
```

```
set(handles.edit131,'string',desig
n_criteria1);
else
    disp(design_criteria2)
set(handles.edit131,'string',desig
n_criteria2);
end
D=str2num(get(handles.edit14,'stri
ng'))
b=str2num(get(handles.edit4,'strin
g'))
s=str2num(get(handles.edit48,'stri
ng'))
s1=min([b,D])
s2=0.75*str2num(get(handles.edit84
,'string'))*(1+str2num(get(handles
.edit2,'string')));
s3_in_mm=str2num(get(handles.edit3
1,'string'))/8
sv_in_mm=min([s,s1,s2,s3_in_mm])
set(handles.edit132,'string',sv_in
_mm);
sl_in_mm=str2num(get(handles.edit1
37,'string'));
slm_in_mm=350
sl_in_mm=min([sl_in_mm,slm_in_mm])
set(handles.edit133,'string',sl_in
_mm);
function
pushbutton42_Callback(hObject,
eventdata, handles)
c=[];
d=[0];
set(handles.edit58,'string',c);
set(handles.edit4,'string',c);
set(handles.edit14,'string',c);
set(handles.edit84,'string',c);
set(handles.edit5,'string',c);
set(handles.edit11,'string',c);
set(handles.edit31,'string',c);
set(handles.edit50,'string',c);
set(handles.edit46,'string',c);
set(handles.edit45,'string',c);
set(handles.edit130,'string',c);
set(handles.edit137,'string',c);
set(handles.edit131,'string',c);
set(handles.edit132,'string',c);
set(handles.edit133,'string',c);
set(handles.edit24,'string',c);
set(handles.edit56,'string',c);
set(handles.edit38,'string',c);
set(handles.edit39,'string',c);
set(handles.edit55,'string',c);
set(handles.edit48,'string',c);
set(handles.edit68,'string',c);
set(handles.edit2,'string',c);
set(handles.edit40,'string',c);
set(handles.edit89,'string',c);
set(handles.edit90,'string',c);
set(handles.edit49,'string',c);
set(handles.edit200,'string',c);
set(handles.edit21,'string',c);
set(handles.edit79,'string',d);
set(handles.edit80,'string',d);
set(handles.edit53,'string',d);
set(handles.edit201,'string',d);
set(handles.edit202,'string',d);
```

6. Source Code for Optimization of Simply Supported Double Reinforced Rectangular Beam Section

Optimization running file

```
function varargout =
optimize(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
'gui_Singleton',  gui_Singleton,
...
'gui_OpeningFcn',
@optimize_OpeningFcn, ...
'gui_OutputFcn',
@optimize_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
                   'gui_Callback',
[]);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State,
varargin{:});
end
function
optimize_OpeningFcn(hObject,
eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout =
optimize_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;
function
pushbutton1_Callback(hObject,
eventdata, handles)
run('main.m')
```

Input parameter

```
propmt={'DL in kN/m','LL in
kN/m','Ecm in MPa','fck in
MPa','fyk in MPa','beam span in
mm'};
name='beam input parameters';
lines=1;
def={'20','5','31e6','25','400','6
000'};
options.Resize='on';
options.WindowStyle='normal';
options.interpreter='tex';
answer=inputdlg(propmt,name,lines,
def,options);
DL=str2double(answer{1});
LL=str2double(answer{2});
E=str2double(answer{3});
fck=str2double(answer{4});
fyk=str2double(answer{5});
L=str2double(answer{6});
assignin('base', 'DL', DL);
assignin('base', 'LL', LL);
assignin('base', 'E', E);
assignin('base', 'fck', fck);
assignin('base', 'fyk', fyk);
```

```
assignin('base', 'L', L);
save('DL.mat','DL');
save('LL.mat','LL');
save('E.mat','E');
save('fck.mat','fck');
save('fyk.mat','fyk');
save('L.mat','L');
```

Cost of Steel and unit cost of concrete

```
prompt={'unit cost of Concrete in
Birr/m3','unit cost of diameter
8mm bar in Birr/kg','unit cost of
diameter 10mm bar in Birr/kg
','unit cost of diameter 12mm bar
in Birr/kg','unit cost of diameter
14mm bar in Birr/kg','unit cost of
diameter 16mm bar in
Birr/kg','unit cost of diameter
20mm bar in Birr/kg','unit cost of
diameter 24mm bar in Birr/kg',
'unit cost of diameter 30mm bar in
Birr/kg','unit cost of diameter
32mm bar in Birr/kg'};
name='reinforcement bar market
unit cost & unit cost of
concrete';
lines=1;
def={'1700','46.41','45.92','43.16
','45.22','44.3','45.23','37.5','3
8.2','38.5'};
options.Resize='on';
options.WindowStyle='normal';
options.interpreter='tex';
answer=inputdlg(prompt,name,lines,
def,options);
Cc=str2double(answer{1});
Cd8=str2double(answer{2});
Cd10=str2double(answer{3});
Cd12=str2double(answer{4});
Cd14= str2double(answer{5});
Cd16=str2double(answer{6});
Cd20=str2double(answer{7});
Cd24=str2double(answer{8});
Cd30=str2double(answer{9});
Cd32=str2double(answer{10});
assignin('base', 'Cc', Cc);
assignin('base', 'Cd8',Cd8);
assignin('base', 'Cd10',Cd10);
assignin('base', 'Cd12',Cd12);
assignin('base', 'Cd14',Cd14);
assignin('base', 'Cd16',Cd16);
assignin('base', 'Cd20',Cd20);
assignin('base', 'Cd24',Cd24);
assignin('base', 'Cd30', Cd30);
assignin('base', 'Cd32',Cd32);
save('Cc.mat','Cc');
save('Cd8.mat','Cd8');
save('Cd12.mat','Cd12');
save('Cd14.mat','Cd14');
save('Cd16.mat','Cd16');
save('Cd20.mat','Cd20');
save('Cd24.mat','Cd24');
save('Cd30.mat','Cd30');
save('Cd32.mat','Cd32');
```

Discretization script

```
function x=disc(x)
allx1=200:50:500;
allx2=2:12;
allx3=[10,12,14,16,20,24,30,32];
ds=[8,10,12];
sv=[180,200,220,240,260,280,300];
x([1,2]) = allx1(x([1,2]));
x([3,5]) = allx2(x([3,5]));
x([4,6]) = allx3(x([4,6]));
```

```
x(7)=ds(x(7));
x(8)=sv(x(8));
```

analysis script for simple span beam

```
function
[Msd,Vsd,delta_max,swt]=analysis(x
)
load('DL.mat','DL');
load('LL.mat','LL');
load('L.mat','L');
load('E.mat','E');
swt=x(1)*(x(2)+50)*25*10^-6;
Pd=1.35*(DL+swt)+1.5*LL;
Msd=(Pd*L^2)/(8*10^6);
Vsd=(Pd*L)/(2*1000);
I=(x(1)*(x(2)+50)^3)/12;
delta_max=(5*(DL+swt+LL)*L^4*10^3)
/(384*E*I);% maximum deflect
```

Constraint script

```
function [c,ceq]=constraint(x)
load('fck.mat','fck');
load('fyk.mat','fyk');
load('L.mat','L');
fctm=0.3*fck^(2/3);
dg=20;Cc=30;
Ast=(pi/4)*x(3)*x(4)^2;
Asc=(pi/4)*x(5)*x(6)^2;
h=x(2)+50;
[Msd,Vsd,delta_max,~]=analysis(x);
As_min=(2.6*fctm*x(1)*x(2))/fyk;
As_max=0.04*x(1)*h;
MRd_lim=0.167*fck*x(1)*x(2)^2*10^-
6;
% Minimum Spacing
s_min=max([x(4),dg+5,20]);
x(8)=(x(1)-2*(Cc+x(7))-
x(3)*x(4))/(x(3)-1);
if Msd <= MRd_lim
a=(0.87*fyk*Ast)/(0.567*fck*x(1));
    x_a=1.25*a;
    MRd=0.87*fyk*Ast*(x(2)-
a/2)*10^-6;
else
    a=(0.87*fyk*(Ast-
Asc))/(0.567*fck*x(1));
    x_a=1.25*a;
    MRd=(0.567*fck*x(1)*a*(x(2)-
a/2)+0.87*fyk*Asc*x(2))*10^-6;
end
VRd_max=0.18*x(1)*x(2)*fck*(1-
fck/250)*10^-3;
c(1,1)=Msd/MRd-1;
c(2,1)=Ast/As_max-1;
c(3,1)=Asc/As_max-1;
c(4,1)=As_min/Ast-1;
c(5,1)=Vsd/VRd_max-1;
c(6,1)=x_a/(0.45*x(2))-1;
c(7,1)=1.5/(h/x(1))-1;
c(8,1)=(h/x(1))/2.5-1;
c(9,1)=delta_max/(L/250)-1;
c(10,1)=s_min/x(8)-1;
ceq=[];
function
[c,ceq]=constraintwithdisc(x)
x=disc(x);
[c,ceq]=constraint(x);
```

Objective script

```
function F= objective(x)
load('L.mat','L');
load('Cc.mat','Cc');
load('Cd8.mat','Cd8');
load('Cd10.mat','Cd10');
load('Cd12.mat','Cd12');
load('Cd14.mat','Cd14');
```

```
load('Cd16.mat','Cd16');
load('Cd20.mat','Cd20');
load('Cd24.mat','Cd24');
load('Cd30.mat','Cd30');
load('Cd32.mat','Cd32');
gamma_s=7850;
Ast=(pi/4)*((x(3)*x(4)^2)+x(5)*x(6
)^2);
Vbg=x(1)*(x(2)+50)*L;
Av=(pi/4)*x(7)^2;
ns=L/x(8)+1;
Ls=2*(x(1)+(x(2)+50));
Vstr=Av*ns*Ls;
Vs=Ast*L;
Vbc=Vbg-Vs-Vstr;
if (x(4)==10)
 F=Cc*Vbc*10^-
9+gamma_s*Cd10*(Vs+Vstr)*10^-9;
 elseif (x(4)==12)
 F=Cc*Vbc*10^-
9+gamma_s*Cd12*(Vs+Vstr)*10^-9;
 elseif (x(4)==14)
 F=Cc*Vbc*10^-
9+gamma_s*Cd14*(Vs+Vstr)*10^-9;
  elseif (x(4)==16)
 F=Cc*Vbc*10^-
9+gamma_s*Cd16*(Vs+Vstr)*10^-9;
 elseif (x(4)==20)
 F=Cc*Vbc*10^-
9+gamma_s*Cd20*(Vs+Vstr)*10^-9;
 elseif (x(4)==24)
 F=Cc*Vbc*10^-
9+gamma_s*Cd24*(Vs+Vstr)*10^-9;
 elseif (x(4)==30)
F=Cc*Vbc*10^-
9+gamma_s*Cd30*(Vs+Vstr)*10^-9;
elseif (x(4)==32)
F=Cc*Vbc*10^-
9+gamma_s*Cd32*(Vs+Vstr)*10^-9;
 elseif (x(6)==10)
F=Cc*Vbc*10^-
9+gamma_s*Cd10*(Vs+Vstr)*10^-9;
elseif (x(6)==12)
 F=Cc*Vbc*10^-
9+gamma_s*Cd12*(Vs+Vstr)*10^-9;
elseif (x(6)==14)
 F=Cc*Vbc*10^-
9+gamma_s*Cd14*(Vs+Vstr)*10^-9;
elseif (x(6)==16)
 F=Cc*Vbc*10^-
9+gamma_s*Cd16*(Vs+Vstr)*10^-9;
elseif (x(6)==20)
 F=Cc*Vbc*10^-
9+gamma_s*Cd20*(Vs+Vstr)*10^-9;
elseif (x(6)==24)
 F=Cc*Vbc*10^-
9+gamma_s*Cd24*(Vs+Vstr)*10^-9;
elseif (x(6)==30)
 F=Cc*Vbc*10^-
9+gamma_s*Cd30*(Vs+Vstr)*10^-9;
elseif (x(6)==32)
 F=Cc*Vbc*10^-
9+gamma_s*Cd32*(Vs+Vstr)*10^-9;
 elseif(x(7)==8)
 F=Cc*Vbc*10^-
9+gamma_s*Cd8*(Vs+Vstr)*10^-9;
elseif (x(7)==10)
 F=Cc*Vbc*10^-
9+gamma_s*Cd10*(Vs+Vstr)*10^-9;
elseif (x(7)==12)
 F=Cc*Vbc*10^-
9+gamma_s*Cd12*(Vs+Vstr)*10^-9;
End
F=min(F);
```

```
function F = objectivewithdisc(x)
x = disc(x);
F = objective(x);
```

Main script

```
clear
close all
clc
run('input_parameter.m');
run('cost_of_steel_and_unit_cost_o
f_concrete.m');
LB=[1,1,1,1,1,1,1,1];
UB=[7,7,11,8,11,8,3,7];
obj=@objectivewithdisc;
cons=@constraintwithdisc;
intcon=1:8;
rng(0,'twister') ;
options =
gaoptimset('PlotFcns',{@gaplotbest
f,@gaplotbestindiv,@gaplotexpectat
ion,@gaplotstopping},'PopulationSi
ze',200,'generations',800,'TolFun'
,1e-15,'TolCon',1e-
15,'StallGenLimit',50,'Display','i
ter');
[x,fval]=ga(obj,8,[],[],[],[],LB,U
B,cons,intcon,options);
x=disc(x);
disp(x')
disp(fval)
fprintf('\nCost function returned
by ga = %g\n', fval);
```