Metropolia
University of Applied Sciences

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Automation in Structural Engineering

Master Thesis

International Master of Science in Construction and Real Estate Management

Joint Study Programme of Metropolia UAS and HTW Berlin

from

Udham Singh

S0572561

Berlin, 26.07.2021

1st Supervisor: Prof. Sunil Suwal

2nd Supervisor: Dr. Paula Naukkarinen

Acknowledgement

Following people, institutions and researchers have tremendously contributed to the successful completion of this thesis.

I wish to express my deepest gratitude to my supervisors, Dr. Paula Naukkarinen and professor Sunil Suwal for their continuous guidance and support throughout my thesis. Without their thought-provoking feedback, it would not have been possible to improve my work and take it to a higher level. Also, I thank you for your invaluable insights in helping me to formulate my research questions.

In addition, I would like to thank my parents, colleagues and my girlfriend to keep me motivated and providing emotional support through hard times during my thesis. Without their love, help and support, it would not have been possible to get through the difficult situation that Covid-19 has brought.

I would like to extend my sincere thanks to prof. Dr.-Ing. Nicole Riediger and to the universities for regular guidance and understanding in dealing with the Covid-19 situation and making the studies as smooth as possible.

In the end, I would like to thank all the researchers who have presented such a tremendous amount of great work in the field of automation making it possible for me to understand the topic inside out and moving in the right direction.

Conceptual Formulation

International Master of Science in Construction and Real Estate Management

Joint Study Programme of Metropolia Helsinki and HTW Berlin

Date: 06.02.2020

Conceptual Formulation

Master Thesis for: Mr Udham Singh

Student number: 1913005

Topic: Automation in Structural Engineering

Background:

The initial planning and designing stages of a project are very crucial for deciding the most efficient and effective design solution from various available options. Therefore, every possible solution should be analysed and calculated to decide the best suited for the requirements of the client and achieving sustainability. The design process can include a lot of repetitive calculations for finding the most optimised and feasible solution. Therefore, automation of the design process saves time which also saves money in construction and real estate industry. The one such method of automation is optimising using genetic algorithm. Also, it has been shown by previous optimisation studies that the use of metaheuristic methods have proved to give most optimised results. Therefore, to better analyse the structure in the preliminary phase, the automation can be a very reliable, precise, time-saving and cost-efficient solution which will lead to a best-optimised design of a structure and contributing to sustainability. The optimisation can be done with one of the many possible meta-heuristic optimizing methods available according to our convenience and competence. The whole process can be organised with the help of BIM (Building Information Modelling) with a well-developed framework to efficiently include BIM model with Tekla or Revit, FEM Structural analysis software (such as RFEM, StadPro or RSAP) and the optimization process using MATLAB. The best possible optimised solutions can then be utilised to answer different possible research questions.

Research Question:

1) How to extract and utilise the data from Building information modelling (BIM) and Finite element method software (RFEM) to the optimisation process?
2) What is the difference in results of the optimisation process and the initial results of the BIM and RFEM software in terms of cost, carbon footprint and steel reinforcement?
3) What is the relationship between different structural element sizes on the cost and carbon footprint of the structure?
4) What is the relationship between cost-optimised, carbon optimised, and steel reinforcement optimised structural models and how to choose the final model which can be efficient in all the three optimisations?

Methodology:

The thesis will be more of a practical work which will involve designing a simple model in AutoCAD, Revit and RFEM to understand the geometry and structural calculations for safety. The data from a BIM model is then systematically taken from Revit which is used as an algorithm to find the best possible solutions for the model in terms of steel reinforcement, cost and carbon footprint. The analysis will be done on how the optimisation helps attain a sustainable solution and if it makes a big difference in achieving the sustainability goals. The

Metropolia

htw. Hochschule für Technik und Wirtschaft Berlin University of Applied Sciences

methodology for optimisation is still open but for now, MATLAB seems a most appropriate option.

Time Scale:

Task	Start	End	Outcome
Research on the topic	3/1/2020	5/1/2020	Clarity on the thesis topic
Develop BIM Framework	5/1/2020	6/1/2020	Organising the tasks to be performed and connecting them
Learn MATLAB and computer programming	6/1/2020	8/1/2020	Familiar with MATLAB and genetic algorithm
Develop the AutoCAD model with preliminary calculations	8/1/2020	9/1/2020	Achieving the final model
Export the model in Revit and RFEM for Structural Analysis	9/1/2020	11/1/2020	Final data for optimizing process and comparison
Mathematical Objective functions to be optimised	11/1/2020	1/1/2021	Final equations which must be used for Genetic algorithm
Computing in MATLAB/Computer language	1/1/2021	3/1/2021	Final optimised solutions should be available
Analysing the Results	3/1/2021	6/1/2021	All questions must be answered with all the comparisons
Writing Report and preparing presentation	6/1/2021	8/1/2021	Everything should be ready for submission

Resources:

Metropolia University of Applied Sciences library: https://metropolia.finna.fi/?lng=en-gb
HTW Berlin library: https://bibliothek.htw-berlin.de/en/
Academia website: https://www.academia.edu/
Google Scholar: https://scholar.google.com/
Research gate: https://www.researchgate.net/

References:

Adeli, H., & Sarma, K. (2006). Cost Optimization of Structures: fuzzy logic, genetic algorithm and parallel computing. John Wiley & Sons Ltd.

Eleftheriadis, S., Duffour, P., Greening, P., James, J., & Mumovic, D. (2017). Multilevel Computational Model for Cost and Carbon Optimisation of Reinforced Concrete Floor Systems. *Automation and Robotics in Construction*.

Goldberg, D. E. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, Inc.

Melanie, M. (1999). *An Introduction to Genetic Algorithm.* Massachusetts: A Bradford Book.

Signature of the Supervisor

Abstract

The initial planning and designing stage of a project is very crucial for deciding the most efficient and effective design solution from various available options that directly impact sustainability. The optimal solution for reinforced concrete structures can include a lot of repetitive calculations for finding the most optimized and feasible solution, which is impractical manually and semi-automatically using structural analysis software. Therefore, the aim of the research is the automation of the design process which may well be the answer for solving repetitive problems efficiently and reliably by providing optimal solutions in terms of time, cost and embodied carbon emissions leading to an optimized design of a structure and contributing to sustainability in construction and real estate industry. One such method of automation is optimizing using a genetic algorithm in MATLAB which is a metaheuristic method and has proved to provide optimal and robust results in past research. To understand the results of the genetic algorithm, it is compared with the manual calculations for various elements such as beam, column, slabs and building frame. The building information modelling tools such as Revit and Tekla are used for visualizing the results. The automation of the design process by incorporating Eurocode principles and constraints indicate that the cost saving is in the range of 7-40% and embodied carbon emission saving is between 5-52% depending on the elements. The use of a genetic algorithm indicates the saving of costs and embodied carbon emissions in all the analysed elements and frames demonstrating it to be a robust, cost-effective and time-efficient solution to achieve optimization of structural designing in the early design phase.

Keywords: automation, *design process, genetic algorithm*, *optimization*

Table of Contents

List of Figures

List of Tables

List of Abbreviations

GA - Genetic Algorithm

MATLAB – Matrix Laboratory

RFEM – Dlubal Structural Analysis Software using Finite Element Method

RSAP – Robot Structural Analysis Product

STAAD PRO – Structural Analysis and Designing Program

SQP - Sequential quadratic programming

BIM – Building Information Modelling

FEM – Finite Element Method

FRP – Fibre Reinforced Polymer

CO_2 – Carbon Dioxide

BS – British Standards

RCC – Reinforced Cement Concrete

ACI - American Concrete Institute

ULS – Ultimate Limit State

COA – Cuckoo Optimization Algorithm

API – Application Programming Interface

ACO – Ant Colony Optimization

SA – Simulated Annealing

ANN – Artificial Neural Network

LP – Linear Programming

GRP – Generalized Reduced Gradient

HS – Harmony Search

TLBO - Teaching Learning Based Optimization

BA – Bat Algorithm

BS – British Standard

CSI – Computers and Structures Inc.

ETABS - Extended Three-Dimensional Analysis of Building Systems

EA - Evolutionary Algorithms

NSGA - Nondominated Sorting Genetic Algorithm

FORTRAN – Formula Translation

LMM - Lagrangian Multipliers Method

UDL - Uniformly Distributed Load

PL - Point Load

DRB - Doubly Reinforced Beam

GRG - Generalized Reduced Gradient

IP - Interior point

DSSPF - Decision Support System for Precast Floors

MGA1 - Messy Genetic Algorithm

PLOOTO - Parametric Layout Organization Generator

LCCF - Life Cycle Carbon Footprint

LCC – Life Cycle Cost

MCDM - Multi-Criteria Decision Making

LCEI - Life Cycle Environmental Impact

MOO - Multi-Objective Optimization

GWO - Gray Wolf Optimization

SUMT - Sequential Unconstrained Minimization Technique

NLPP - Non-Linear Programming problem

GHG - Green House Gases

List of Symbols

Ø - The diameter of the steel bar

θ – Angle

ρ – Ratio of steel reinforcement to concrete

δ – Moment redistribution ratio

α_{sx} – Moment redistribution factor in the x-direction

α_{sy} – Moment redistribution factor in the y-direction

ρ_o – Reference reinforcement percentage

ρ'– Required compression reinforcement percentage

u_i – Reference reinforcement percentage

1 Introduction

The construction industry is one of the consistent growing industries in the world contributing significantly to the total carbon dioxide emissions, energy usage for material manufacturing such as concrete, steel, timber and glass. The United Nations have laid out various sustainability goals which are aimed to be achieved in different industries to improve sustainability which also includes the building and construction sector. Also, the Paris Agreement in 2015 has indicated that the building industry must reduce its carbon emission and energy usage to achieve the global goal of keeping the global warming temperature below 2ºC as its contribution is significant. Therefore, research on the integration of growing technology with current construction practices has been identified as a progressive way to move towards better decision making which can ultimately lead to sustainability. To achieve the goals set for 2050, energy usage and carbon emission must be reduced globally by at least 60%. As per reports by United Nations, the goal achievement requires huge investment for energy-efficient building construction which on the contrary saw a reduction of about 2% from 2017 to 2018. Therefore, a decline in investment towards energy-efficient buildings is taking us further away from the goals of 2050. (United Nations Environment Programme, 2020)

The building and construction sector currently use 36% of total global energy usage and emits around 39% of global GHG (Green House Gas) emissions which is around 9.7 $GtCO_2$. The current future projections indicate that the population will increase by around 2.5 billion by the year 2050 which will ultimately increase the demand for the building and construction sector industry. Keeping this projection trend in mind, the stocks for the building sector will increase by 90% and its contribution to GHG and energy usage will climb as well. The projections have made the task even harder for the building and construction sectors. Therefore, future projected contributions have challenged the professionals in the construction industry to come up with ideas to reduce this contribution over a certain period systematically. (GlobalABC , 2019)

The focus of the industry at the moment is only on the energy-efficient materials and techniques used during the life cycle of the building and completely ignoring the fact that concrete and steel are producing significant carbon emissions that can be tackled at an early stage of designing. In the building and construction sector, the initial design phase should be optimal so that the material consumption, cost, embodied carbon emission and energy usage can be reduced.

The most used materials in structural design are concrete and steel. Concrete is the second most used material for various purposes after water which in itself is momentous information. Current statistics show that steel consumption is growing every year and touched approximately 1808 million metric tons in 2018 which indicates a consistent increment every year. Also, every ton of steel produced emits 1.83 tons of CO_2 emission. It accounts for 7-9 % of the total global direct emissions from fossil fuels production. (World Steel Association, 2019)

The knowledge of these materials contributing the most to the construction industry's carbon emissions makes it even more important to incorporate the design process which makes it possible to optimize it. Therefore, there has been significant research on optimizing the design process through various methods that will be explored in the coming sections and then the most suitable option will be further researched to optimize the various building elements and the structural frame. The optimal design will have a significant reduction in carbon emissions and cost which in turn will have a huge impact on the industry. To have a perspective of what this reduction would mean we can have a look at this picture below to realise the responsibility of structural engineers and how much they can contribute.

Figure 1: Embodied carbon emission savings (Gibbons & Orr, 2020)

1.1 Problem Formulation

The industry has faced a grand question of how to optimize the cost and energy used in construction. The building can be divided broadly into the construction phase and the life cycle phase. Both heavily plays a role in how efficient the building will be over its life cycle. Therefore, the optimization of both the design phase and the maintenance and operation phase is extremely essential.

The optimization of the design phase seems to have a significant effect on the embodied carbon emissions and energy efficiency of a building. Therefore, optimization in designing the structures is a practical and effective way to contribute. Now, widely, and essentially used materials in the building and construction sector are concrete and steel. Therefore, various techniques of sustainable consumption of these materials need to be implemented for achieving sustainability. The distribution of steel usage in the industry is shown below:

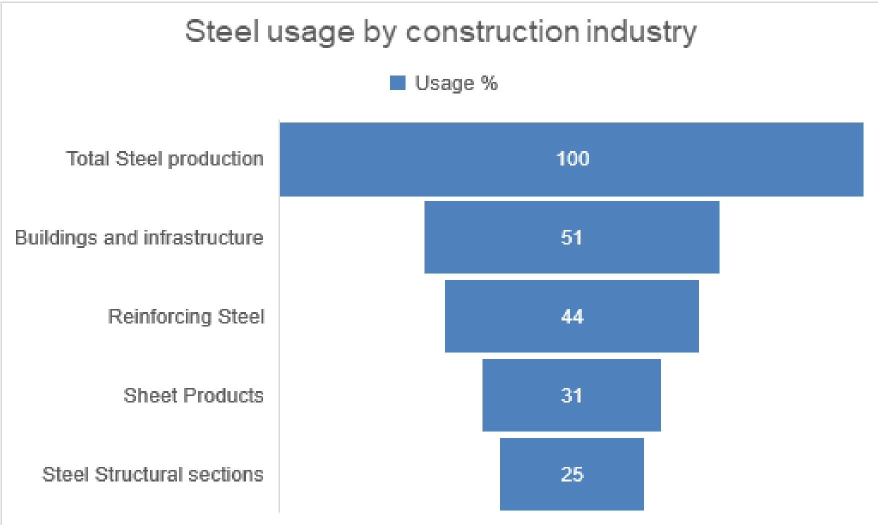


Figure 2: Steel usage (World Steel Association, 2019)

The initial planning and designing stages of a project are very crucial for deciding the most efficient and effective design solution from various available options. Therefore, every possible solution should be analysed and calculated to decide the best suited for the requirements of the client and achieving sustainability. The design process can include a lot of repetitive calculations for finding the most optimized and feasible solution for a given building. Therefore, efficient analyses of the structure in the preliminary phase should be achieved. The automation technique can be a very reliable, precise, time-saving and cost-efficient solution that will lead to a best-optimized design of a structure and contributing to sustainability at the same time.

We must figure out the strategy on how we can implement the optimizing strategy in the industry. we know that the safety of any structure cannot be compromised in any scenario otherwise, it loses the basic motive of construction in the first place. So, there is a need for the industry to come up with a system where the safety of a structure is intact and, we can increase the efficiency of the concrete and steel used satisfying the minimum requirement given by the codes.

The optimization of materials used such as steel reinforcement and concrete is essential. Also, the embodied carbon emission at the design phase can be achieved by optimization. Therefore, optimization should be done for embodied carbon emission and the cost of the structure.

1.2 Problem Identification

The construction industry has developed and accepted technological advancement to an extent where it has become easy to develop the design with the help of various tools. Nowadays, it is possible to put your mind out and develop the most innovative, sustainable, and fancy design for a structure. Additionally, we have reached a point where we have selfishly consumed a huge number of natural resources without giving a thought to future generations and we have failed to plan a world where we could conserve most of our resources and use them efficiently and effectively. Therefore, it is our social, moral and professional responsibility to give our best to achieve the required level of sustainability for our resources. The current way of analyzing the structure for cost and carbon emission gives an insight into the issues we are facing in the current market and some of them are listed below:

- Hand calculations become harder and time inefficient as the size and complexity of the structure increases.
- The complexity and professional competence requirements in implementing automation techniques have made the building and construction industry reluctant to use them for the design phase.
- The time required for analyzing big structures manually is huge which adds to an increase in project cost significantly.
- There is not enough repetitive designing for various combinations that are available to structural engineers because it is a tedious and ineffective strategy. Therefore, there is a high possibility to miss out on the most optimum design solution. Also, human beings tend to incline towards more safety and use extra resources than required which possibly impacts the material consumption for steel and concrete etc.
- Designers and clients do not widely focus on the GRG (Green House Gas) aspect at the early stages of the construction because the sustainable options are usually higher in cost.

1.3 Problem Statement and Objectives

The above-mentioned issues in subsection 1.2 are improved by implementing the automation concept and it has also been extensively studied by various researchers. Therefore, the current study will implement optimization in the design phase and compare it with the present design results. The main research questions which will be answered in the further chapters are as follows:

1) How to develop an interaction between various tools such as Building information modelling (BIM), MATLAB and Finite element method software (RFEM) to extract and utilise the data for the optimization process?

2) What are the impacts of different structural element dimensions on the cost and embodied carbon emissions of the structure?

3) What are the advantages of the optimization process over the manual designing methodology in terms of cost, embodied carbon emissions and steel reinforcement?

4) What is the relationship between cost-optimized and carbon optimized structural models and how does it provide a different perspective for decision making in the design phase?

In the process of answering the above-mentioned research questions, some new concepts are also investigated. The concepts are as follows:

- Use of BIM in developing automation framework.
- Optimizing Steel reinforcement design, cost and carbon emission in the structure considering various parameters.
- Comparing the best optimum models and structural elements which provide the best solution for cost, carbon emission and steel reinforcement.
- Efficiency and effectiveness of modelling using genetic algorithm function in MATLAB optimization toolbox.
- Sustainability achievements with automation.

2 Literature Survey and Review

The most important aspect of any progressive academic research is to have a thorough insight into what has already been done in the past. Therefore, to achieve this objective, surveying and collecting the available research is a must. This will also help the researcher define his/her contribution to the research topic broadly. We must collect and survey all the literature available on the relevant topic as much as possible. By reviewing the past literature, we can conclude what we can research so that it gives additional knowledge for anyone wanting to read about a similar topic. The literature survey is possible through a wide range of possibilities such as collecting journals, articles, books etc from various academic platforms and online websites which gives all the research papers for free. The literature survey for the current thesis has been developed in such a way that it provides all the possible knowledge about optimization of steel reinforcement for different structural members and structures, optimization of cost and carbon of a structure or structural members, the various techniques and tools already identified and implemented by the researchers.

The literature surveyed and collected has been reviewed for a deeper understanding of the topic. The review also suggested the precise area which needs academic contribution which will have a tremendous effect on the current research. The current literature review gives deep knowledge of the implemented automation techniques in the construction and building sector at various levels. Depending on the past research developments there will be a plan made for further contribution to the current research topic.

2.1 Literature Collection and Segregation

The research journal has used fibre reinforced concrete instead of normal concrete to test the seismic retrofitting of a framed structure. The fibre reinforced polymer (FRP) has shown good strength and ductility to improve the typical design against the seismic effect. The model has been calculated with the finite element method (FEM) for internal forces. The multi-objective genetic algorithm (GA) approach has been used to maximize the ductility of the frame and minimize the volume of the frame to achieve a cost-efficient retrofitting model. The case study has been done on a framed structure for further optimization in a practical way. The results have shown the improved and easy optimization achievement for FRP jacketing by using multiple objective functions and different variables. (Chisari & Chiara, 2016)

The optimization of RCC (Reinforced Cement Concrete) flat slab has been studied satisfying the British standards (BS8110). The main aim of the optimization process is to minimize the slab thickness and find out the perfect percentage of steel reinforcement which will give the least cost of the slab keeping the safety criteria intact. The method used for this purpose is the genetic algorithm (GA) which is an inbuilt function of MATLAB. The GA is used with the help of objective functions which needs to be minimized. The objective functions have certain variables which can be tried in the algorithm combinations. The most cost-efficient slab thickness in the middle strip and the column strip has been achieved with the best possible percentage of steel reinforcement. (Olawale, et al., 2019)

The cost optimization of the flat slab according to British standards has been carried out. The cost objective function includes the cost of foundations, columns and floors, with each of them covering the cost of concrete, labour cost for reinforcement, material cost and formwork cost. The structural analysis is done using the equivalent frame method and optimization is done in three steps with column layout optimization, column dimensions and slab thickness optimization, number and sizes of the reinforcement for reinforced concrete members. Finally, the examples of three flat slabs have been taken for optimization and results are compared to each other. The results indicate the cost-saving by optimization as compared to the conventional designing. The cost saving is directly proportional to the number of structural elements, which means the bigger the amount of RCC the more is the saving (Sahab, et al., 2005).

The very detailed paper gives the design of flat one way and two-way slabs using the ACI (American Concrete Institute) codes. A very described method of constructing various objective functions for both types of slabs with all the equations is explained very clearly. The ACI codes defining the required resistance conditions to clear ULS (Ultimate limit state) and SLS (Serviceability limit state) criteria have been satisfied in the examples. The cost optimization objective function which has to be minimised include the cost of concrete and reinforcement steel. The metaheuristic method of the cuckoo optimization algorithm is used to minimize the cost function. The results obtained from the COA (Cuckoo optimization algorithm) has been compared to the other optimization algorithms for the same slab example to give an insight into the efficiency of COA. (Ghandi, et al., 2017)

The author has worked on optimizing the design for cost and carbon. The structural analysis is done using the finite element method which has been provided with constraints through genetic analysis. The building information modelling is also used in the whole optimization process. The software used for BIM and structural analysis were Autodesk Revit and Autodesk robots respectively. Also, the computer language used was C# and developed a .NET framework for extracting data from the robot application programming interface (API). The key areas which were chosen to be optimized are the grid layout of the building, floor slab thickness, column sizes and steel reinforcement. The objective function was developed in an expression that included all the variables targeted to be optimized. Multilevel optimization includes three different levels. Firstly, the structural grid layout has been given many variations possible. Secondly, the column sizes and slab thickness were optimized at this level. Thirdly, reinforcement data for column and slab. The total area of the structure is 15m*16m with a core area of 4m*5m. The results indicated different models which are best in terms of cost and carbon optimization. The best possible solution for saving cost has three per cent less compared to the carbon optimized solution but it also has greater carbon emission of seven per cent. As expected, the total contribution of the slab is 75 per cent of the total cost and 90 per cent of the total carbon emission of the structure. (Eleftheriadis, et al., 2017)

The author (Paya-Zaforteza, et al., 2009) has used Spanish building codes for the optimization process which includes reducing the carbon do oxide emission and cost of the frame to its minimum. The frame chosen for the study was of different heights which sat at 2,4,6 and 8 m height. The frame with 8m height had the biggest number of variables which were 156 and the total combinations for them were 10^{232}. The optimization was carried out used the metaheuristic method called simulated annealing (SA). The result of the study indicated that a CO_2 optimized solution is more expensive than a cost-optimized solution but the difference between them is of utmost 2.77 per cent which is not very huge in a practical scenario and could be accommodated in the construction industry. Also, the cost-optimized solution gives an increment of 3.8 per cent for the CO_2 emission. (Paya-Zaforteza, et al., 2009)

The paper deals with the optimization of the cost function for a 3D frame structure. The cost function is heuristic with the use of the Ant Colony Optimization (ACO) algorithm. The criteria followed for the safety are that the resistance axial force, shear force and moment of members are greater than the ones coming onto the members through loading. Also, the serviceability limit state is satisfied by controlling the maximum deflection less than deflection by serviceable load case. The author has concluded that by using this technique there is a further cost saving of around 4.8 per cent compared to the paper provided by (Sahab, et al., 2005). In a nutshell, two examples are considered for applying this technique of Ant Colony Optimization (ACO) algorithm and both of them shows very efficient and effective cost optimization results. (Hadi, et al., 2012)

The paper opts for a genetic algorithm (GA) technique for optimization of the 2D frame structure satisfying the American Concrete Institute (ACI) building code. The cost objective function is the total sum of the cost of concrete, steel and formwork plus labor cost used for the construction. The penalty function is introduced to introduce the constrained part of the problem into GA which is originally apt for unconstrained problem-solving. The selection of the population in the GA is done using tournament selection and repairing operator is also used for better convergence in beams and columns. These changes make the algorithm better as compared to the traditional GA according to the authors. The optimization design has been supported with a 2D frame example and it has clearly shown cost-effectiveness. Also, the time required for the design process was short and effective as compared to the other research done on the same frame. (Niaki, et al., 2016)

The author has attempted to minimize the cost of the frame structure which included the cost of concrete and steel used in the structural members. The Neuroshell-2 software program is adapted to provide the channel for the Artificial Neural Network (ANN) computational model for optimization. The ACI-318-08 code is referred for safety and serviceability purposes. The various constraints for beams and columns are geometric constraints, capacity constraint, minimum steel area requirement constraint, maximum steel area limit constraint, flexural capacity constraint, shear strength requirement constraints, maximum beam width constraint and crack width constraint. The frame chosen for the designing had various variables which create 50 different kind of models which needs to be analyzed for cost optimization. The author concludes that the modelling with the Neuroshell-2 program is efficient and gives out decent results for multiple story frames. (Aga & Adam, 2015)

The study has been performed for optimization of the concrete beam complying with Brazilian standard ABNT NBR 6118:2014. The technique used for this purpose is the analytic solver program with the help of an excel spreadsheet. Also, upon running the option for analyzing the problem and check which solver would fit best for the solving process among LP Simplex, nonlinear GRG and Evolutionary, it was found that the most appropriate for the used problem would be Evolutionary solver because the problem is non-smooth and nonconvex. The considered variables that were suspected to impact the optimization were the span length of the beam, concrete compressive strength and loading. It was seen that the solver was able to provide more cost-efficient solutions as compared to the conventional design. (Correia, et al., 2019)

The study is done to optimize the reinforced cement concrete beam to save the cost of the structural element. The cost objective function is the total sum of concrete, steel reinforcement and formwork used in the beam. The main idea of the optimization is to find out the best grade of concrete and steel which would give the most optimal solution. The optimization is done using an SQP (Sequential Quadratic Programming) algorithm which was carried out using MATLAB. The optimization has given an economical solution to the problem as compared to the conventional method of calculations. (Thomas & Arulraj.G, 2017)

The reinforced cement concrete continuous beam is designed optimally accordi-oning to the Indian Standards satisfying the ULS (Ultimate Limit State), SLS (Serviceability Limit State), ductility and durability. The GA (Genetic algorithm) has opted for the optimization process and the variable is chosen for this study is the cross-section of the beam. The results of the study have been compared to the literature available in the public space. The author has concluded that the GA (Genetic algorithm) algorithm is cost-efficient, time-saving and mathematically convenient to carry out. (Govindaraj & Ramasamy, 2005)

The paper gives an insight into the RCC (Reinforced cement concrete) beam optimization. Fact that there can be a various number of combinations between beam dimensions and the steel reinforcement ratios to give out a similar resistance against the internal forces, the conventional method is not effective as there are so many iterations with changing variables, and it would consume a lot of time in the design phase. Therefore, to tackle this issue GA (Genetic algorithm) is adopted to search for the best solution. The author has concluded that the GA (Genetic algorithm) approach has shown some magnificent results in terms of reducing the cost of the RCC rectangular beam. (Coello Coello , et al., 1997)

The author has worked on optimizing a singly reinforced concrete beam using a nonlinear mathematical expression. The cost and weight of the structural element are adopted for minimizing as an objective function. The ACI 318S-13 (American Concrete Institute) building code is used for designing structural concrete members. The optimization technique is supported with a design example and then comparing it to the current conventional designing methodology. The problem was successfully drafted by a nonlinear programming methodology. The designing and optimization are done complying with all the ULS (Ultimate Limit State) and SLS (Serviceability Limit State) conditions required to be satisfied for the structural member. The objective functions are minimized for various four different examples with different variables and were analyzed for the best suitable solution. The author also highlighted the change in optimal steel ratios in different examples and how the optimal ratio is different as compared to the ratio taken for conventional designing. (Rojas, 2016)

The study is performed for minimizing the construction cost and material used in the designing of simply supported beams, columns, and multi-storey frame structures. The heuristic method used is the GA (Genetic Algorithm). The building codes used are from ACI (American Concrete Institute) for strength and serviceability requirements. The constraints according to the ACI building codes are applied as a penalty function to the fittest solution so that the result is analyzed realistically. The examples worked in the study indicates that the GA (Genetic Algorithm) is an effective and efficient technique for optimization and it also minimizes the cost objective function. (Camp, et al., 2003)

This study is done on optimizing an RCC (Reinforced Cement Concrete) beam and besides the effect of slab thickness on the designing of the beam is studied. The design procedure follows the standards set by ACI318 (American Concrete Institute). The iterative optimization process is formulated by using a metaheuristic method called HS (Harmony Search) which is a music-inspired algorithm search technique. The objective function used is the cost of the beam considering variables such as dimensions and steel reinforcement rebars required for the flexural design. The four different scenarios have been considered with different slab thicknesses. It has been concluded that the slab thickness does not have any significant effect on the most optimal design, but the beam must be considered a T section. (Nigdeli & Bekdas, 2019)

The study has used the new metaheuristic method called TLBO (Teaching Learning Based Optimization) for the optimization of an RC beam. The study is done to check the efficiency of this method as compared to other optimization methods such as HS (Harmony Search) and BA (Bat algorithm). The design procedure complies with ACI 318-05 (American Concrete Institute) building design codes. The optimization process is done for ten different flexural moments. It has been found that the TLBO optimization technique is very robust and efficient for optimization as compared to HS (Harmony Search) and BA (Bat Algorithm). (Bekdaş & Niğdeli, 2016)

The study illustrates the optimization of an existing building in terms of energy efficiency for the best retrofitting of the structure. The study uses GA (Genetic

Algorithm) and energy plus for the energy simulations for nineteen different countries in Europe which have different climatic conditions from each other. The energy evaluation is done using the existing residential building as a benchmark. The various optimization variables are considered for the study such as greenhouse gas reduction, annual energy operating cost reduction, installation and construction cost reduction and annual specific energy demand reduction. The optimized results are hugely dependent on the climatic condition, energy cost and carbon emission as per the country of analysis. (Salataa, et al., 2020)

The study emphasizes the importance of automation on the steel reinforcement calculation for the building structural frame. The BIM (Building Information Modelling) framework have been developed by the author to systematically manage the information for the structure coming from various platforms providing geometric information, structural analysis data and finally the optimization process. The hybrid GA (Genetic Algorithm) has been implemented in three different levels for optimization which are optimizing longitudinal tensile steel reinforcement, optimizing longitudinal compressive steel reinforcement and optimizing the shear steel reinforcement. The building codes used are BS8110 (British Standard) for the structural analysis of the structure. The optimization is done on two examples which are a three-storey building frame and RC beam. The GA (Genetic Algorithm) optimization has shown efficient results as compared to calculated by Autodesk RSA, CSI ETABS and Manual Calculations. The best results by hybrid GA (Genetic Algorithm) for percentage difference between the required minimum required steel reinforcement and the provides steel reinforcement is just 0.004% which is an amazing result as compared to the other methodology. (Mangal & Cheng, 2018)

The study analyzes the hybrid GA (Genetic Algorithm) and then compares it to the conventional GA (Genetic Algorithm). The process of GA works in two different stages which are global search from the search space using the hybrid GA (Genetic Algorithm) and local search using Hooks and Jeeves method (Hooke & JEEVES , 1961) by implementing GA (Genetic Algorithm). The hybrid GA (Genetic Algorithm) is further analyzed by working on a truss problem and to a flat slab problem. The results of the truss problem show that the hybrid GA (Genetic Algorithm) with some more functions are more efficient as compared to the normal GA (Genetic Algorithm). Also, in the optimization of the flat slab, the hybrid GA (Genetic Algorithm) takes more time and evaluated more functions but the result is more optimized as compared to the normal GA (Genetic Algorithm). (Sahab, et al., 2004)

The author implements the new metaheuristic method called CS (Cuckoo Search) which was developed in 2009. The study analyzes the thirteen different examples of optimization and then compares them to the other studies which used different search algorithms. The authors strongly conclude that the CS (Cuckoo Search) algorithm is easy to implement as compared to other algorithms such as GA (Genetic Algorithm) and shows more efficient results for the examples worked in the study. Also, the CS (Cuckoo Search) has comparatively a smaller number of essential parameters which are population size n and Pa. (Gandomi, et al., 2013)

The study indicates the issues with multi-objective evolutionary algorithms (EA) which are its complexity, non-elitism and sharing parameters specification. The authors have come up with a nondominated sorting genetic algorithm (NSGA-II) which eliminates the above-mentioned issues for EA (Evolutionary Algorithms). The methodology is implemented on different problems, and it has been concluded that it reduces the complexity to a good amount but have faced similar issues with epistatic problems as compared to the EA (Evolutionary Algorithm). (Deb, et al., 2002)

The most important part of designing, in the beginning, is to figure out the best topology of the structure. The topology changes the dimensions of all the structural elements and therefore it hugely affects the cost, material consumption and displacements in the structure. The authors have worked on finding the best topology with the use of GA (Genetic Algorithm) for iterative hit and trial options and then the material alterations to minimize the cost function. The cost function includes concrete, steel reinforcement, column placements and the displacements of the structure. The structural analysis is performed using the Autodesk RSAP (Robot Structural Analysis Program). The precise algorithm used for the study is NSGA-II (Non-dominated Sorting Genetic Algorithm) for five different examples with different data. The conclusion was that the optimization certainly helps the structural designer to find the best possible solution in terms of column placement for the given topology, but it also indicates the complexity of the optimization and the similar results for all five examples. (Oliveira & Miranda, 2020)

The author has broadly analyzed the effectiveness of GA (Genetic Algorithm) for optimization as compared to the traditional methodologies. All the characteristics involved in the optimization using GA (Genetic Algorithm) is studied in the current study. The clear algorithm with all its operators and function is explained by the author Also, the explanation is supported by an example of optimizing an equation for maximum and minimum values. (Premalatha, 2015)

The study presents the optimization of the structural frame and beam using GA (Genetic Algorithm). The beam satisfies all the criteria set by the building codes such that flexure, shear, axial and torsion. The internal forces of the beam are done by computing it in STAAD Pro software which gives the result of shear forces, moments and axial forces. The structural analysis is done according to the ACI code (American Concrete Institute) for all the structural members. The analysis is done semi-automatic with the use of STAAD Pro to provide the initial dimensions for columns and beams. This initial data is fed to the GA (Genetic Algorithm) for further optimization and the structure is reanalyzed with the optimized data to check the building code design procedure. (Shariati, et al., 2019)

The study presents optimization of steel reinforcement for an RC (Reinforced Cement) flat slab. The BIM (Building Information Modelling) is used to extract data related to the geometry of the slab and the structural analysis is done using FEM (Finite element method). The optimization is carried out in two steps. Firstly, the steel reinforcement is decided based on the three-dimensional aspect of the FEM (Finite Element Method). Secondly, the realistic approach of providing the steel reinforcement was adopted where a constructability constraint was developed which controlled the bars in each zone and the spacing between them. An example with six different simulations with different constraints but the same geometric data as per conventional design is illustrated for practical application. The first two simulation does not include the constructability constraints at all but the simulations 3 and 4 include only zoning constraint function and simulation 5 and 6 are done with both zoning and spacing constraints. Each pair takes steel reinforcement and spacing from two different databases D1 which has diameters 8mm to 32 mm and spacing from 175 mm to 250 mm with an increment of 25 mm whereas D2 has diameters 8mm to 32 mm and spacing can be from 50 mm to 250 mm with increments of 10 mm. This provides database D2 with many more options for selecting optimized steel weight. The results of automation were compared to the conventional design procedure to get an insight into how the results of providing steel reinforcement weight differ. The authors concluded that the optimization technique studied in first and second simulation has shown a reduction in slab reinforcement by 21 per cent and 23 per cent respectively, simulation 3 and 4 shows a reduction of around 12 per cent, simulation 5 and 6 shows a reduction of 3 per cent and 5 per cent as compared to the conventional design. Therefore, the constraints help achieve a more realistic design that is easier to install on the construction site. (Eleftheriadis, et al., 2018)

The study optimizes the one-way slab for total cost including the cost of concrete and steel reinforcement with satisfying all the design criteria according to the American standards ACI 318-M08 (American Concrete Institute). The heuristic method names PSO (particle Swarm Optimization) is used with a penalty function to satisfy constraints. The technique is illustrated with four different examples having different slab lengths ranging from two meters to five meters and different support conditions. The authors have very clearly identified the equations for flexural constraint, shear constraint, serviceability constraint, deflection constraint and constraint normalization. After studying the four examples the authors concluded that the cost of the slab is directly proportional to the span length. Also, the steel reinforcement is different for each example but for most commonly used spans (7-10 feet) it is between 0.43 per cent to 0.47 per cent. The proposed algorithm can be applied to any location having a different cost for concrete and steel reinforcement. (Ahmadi-Nedushan & Varaee, 2011)

The study optimizes a rectangular RC beam according to the cost function rather than weight minimization. All the constraints have been applied in the optimization process for minimizing cost and it is done using the LMM (Lagrangian Multipliers Method). The minimum cost has been achieved without any iterations. The example of singly and the doubly reinforced beam is illustrated using the ACI building code. After finding some variables such as steel reinforcement ratio in the beam using LMM method, the achieved data is used further for finding out the optimum depth and the steel reinforcement to be provided for best cost-efficient results using the ANN (Artificial Neural Network). The ANN is an inbuilt optimization function of MATLAB and can be easily modelled with algorithm and data input. (Yousif, et al., 2010)

The longitudinal reinforcement design is optimized using MATLAB coding. The author has compared design examples using ACI 318 and EC2 to illustrate the differences in methodology for each building code designing. The reinforcement and neutral axis depth are calculated to satisfy the flexure and axial loading with minimum and maximum limitations according to the codes. A total of four different examples was analyzed by computing for minimizing the cost. The authors concluded that the use of computers and MATLAB has become a fairly easy process for experts in various fields. The time taken for computing was found to be one-tenth of a second is extremely less compared to the manual calculations. (Tomás & Alarcón, 2012)

The optimization of an RCC cantilever beam is carried out using the GA (Genetic Algorithm). The beam is designed using IS 459-2000 (Indian Standards) for strength and serviceability. The objective function for optimization includes the cost of concrete and steel reinforcement. The GA has been applied on a simple beam and it has been found that the beam optimization result using GA gives fairly efficient results compared to the manually calculated. (Alex & Kottalil, 2015)

The study presents optimization of structural RCC beam with different support conditions such as cantilever beam, simply supported beam and continuous beam and subjected to UDL (Uniformly Distributed Load) and PL (Point Load). The design follows the standards provided by the IS 456-2000 (Indian Standards). The design examples have shown effective and efficient results. (Kumar & Shanthi, 2018)

The study performs the optimum design of a one-way slab according to the IS456-2000 (Indian Standard). The optimization technique used in the study is GA (Genetic Algorithm) and MATLAB. The object function is defined for cost minimization which includes the cost of concrete, cost of steel reinforcement and cost of formwork for the given structural element. The cost function with all the constraints is applied to various examples of a slab having different span lengths and loading values. The design example is divided into four main divisions such as available parameters, design variables, building code satisfying criteria and the constraints. After running the GA (Genetic Algorithm), it has been concluded that the ratio of the length of the slab to that of the thickness of the slab should be around 29-30 for achieving the most optimized solution. Also, the authors indicated that the use of higher-grade materials does not always comply with the most optimized design. (Singh, et al., 2014)

The study is done for optimizing a DRB (Doubly Reinforced Beam) complying with strength and serviceability design conditions given in IS 456-2000. The object function for optimization is minimizing the cost of the beam which includes the cost of concrete, steel and formwork. The authors have used three different approaches such as GA using MATLAB, GRG (Generalized Reduced Gradient) method in excel and IP (Interior point) method for problem-solving. The results obtained from each method have been compared to study the best result options. The conclusion was that the GA gave the most optimum results compared to the other two approaches with excellent and smart searching of the space. (Bhalchandra & Adsul, 2012)

The study has been done in the multi-storey frames to optimize the structural members such as beams and columns using ACI 318-05 2005 building codes. The optimization method of sequential quadratic programming which is one of the tools in MATLAB has been used for the study. The objective cost function includes the cost of formwork, material cost for concrete and steel, concrete placing cost, vibrating cost, equipment cost and labour cost. The three examples have been illustrated with a different number of bays and stories in each example. The results from various design examples have concluded that the optimization technique can save up to 23 per cent of the cost as compared to the traditional calculation methodology. (Guerra & Kiousis, 2006)

The study reflects on energy saving in the initial phases of the design process. Energy is an important aspect when it comes to the environment, and it has always been minimized in a building through techniques and methods. The authors in this study have tried to optimize the building design for minimizing embodied energy. The embodied energy includes the total energy consumption during construction and the entire material life cycle. The example of a rectangular beam is taken for cost optimization and analyzed for embodied energy. The objective function used is the cost function and the embodied energy function for minimizing. The building code used for the RCC beam is ACI 318-08M. The results indicate that concrete contributes to much larger embodied energy as compared to steel, but steel is much costlier than concrete. Therefore, the embodied energy-optimized beam solution has more amount of steel as it possesses less embodied energy but on the other hand the cost is also increased as it is expensive than concrete. According to the author's conclusion, the energy-optimized solution is more expensive. The cost increases to around five per cent compared to if we neglect the embodied energy optimization model. Also, the cost-optimized solution is slightly more ductile than the embodied energy-optimized solution. The results have a significant difference when the R which is the cost ratio of steel to that of the concrete changes. (Yeo & Gabbai, 2011)

The focus of the study is on the effect of structural analysis on the life cycle cost of the structure. The efficient design can have a huge reduction in overall energy used for structures over their entire life span. Also, the embodied energy is kept in balance with the right choice of materials and design leading to better sustainability. The authors have used the BIM framework to plan and execute the whole process of optimization. The approach is also tested on a realized structure which is a multistory RC (Reinforced Concrete) building. The contribution of slabs in a building for carbon emission is highest compared to the other structural members. Therefore, optimized and comprehensive design can reduce it to a greater extent. The BIM framework is developed for the optimization process and the life cycle of carbon emission. The quantity of the materials used is directly taken from BIM software which is Autodesk Revit 2017. The focus of the study has been on the optimization of embodied carbon emission of the superstructure such as slabs, columns and walls. It was obvious that the slab contributes a bigger amount to the cost and carbon compared to the columns. The slab design has shown a reduction of 50 per cent of steel reinforcement after optimization as compared to the manual calculations. The authors have analyzed the current structural system of the building and found that 78 per cent of the total life cycle carbon emissions are from the structural system which means optimization at the initial phase is crucial to minimize the carbon emission. The eight different structural systems have been tested and optimized for the same building and it has been concluded that the structural elements contribute 90 per cent of the embodied carbon emissions and architectural elements contribute 10 per cent of the remaining embodied carbon emissions. The optimized results for new structural systems indicate a reduction of 16-19 per cent of the embodied carbon emission as compared to the original building design. (Eleftheriadis, et al., 2018)

The study has been performed on precast concrete floor design using the tool DSSPF (Decision Support System for Precast Floors) and GA (Genetic Algorithm). The integrated design phases such as conceptual design, embodiment design and detailed design are all considered to achieve the highest level of optimization. Also, it considers all the construction phases which include manufacturing of materials/structural elements, transportation to the sites and the erection cost. There are many alternatives for finding the optimized solution such as layout design, the design of dimensions for structural elements, the reinforcement design and concrete strength considerations for cast in place and precast. To accommodate and consider all these alternatives, the cost objective function is introduced so that all these factors can be analyzed, and the overall cost can be minimized. The building code used for satisfying strength and serviceability criteria is ACI (American Concrete Institute). The cost function includes the cost of manufacturing, indirect cost, cast in place concrete, transportation, assembly and connection. The MGA1 (Messy Genetic Algorithm) is used with the rank concept where the first two best ranked is retained using the concept of elitism for further generations. A total of eleven design variables were used for formulating the GA algorithm. The technique is implemented on the realized building called commercial Carvalho which is situated in Brazil. Three design variations are studied for the current building and then it has been compared to the original design of the realized building. The concept of DSSPF has shown a time reduction for the initial structural layout designing phase for structural engineers and this time could be spent on detailing the elements and the design verification phase. (de Albuquerque, et al., 2012)

The study performs integrated optimization for cost and carbon emission in a building. The approach uses three-level analysis which broadly implies optimization of structure layout, generation of architectural layout as well as internal spacing and optimizing various building components. The strength and serviceability are satisfied according to the EC2 (Eurocodes2) for a reinforced concrete structure. The various levels use different techniques for getting the best possible solutions. The first level uses a multilevel and multi-objective purpose optimization approach and to implement it NSGA II (Non-Sorting Genetic Algorithm) is used. The first level finds out the grid layout for columns, dimensions of the column and thickness of the slab and finally the steel reinforcement details of both column and slab. Level two uses PLOOTO (Parametric Layout Organization Generator) tool for generating spatial configuration. The file format achieved from the PLOOTO tool is further used for energy analysis with a software called Energy Plus. Level three involves defining various design properties for the building components, window orientation and the ratio of window to the wall. Furthermore, this data is used to find out the LCCF (Life Cycle Carbon Footprint) and LCC (Life Cycle Cost). The optimization technique is implemented on a building of 22*22 m plot dimension and the cutout area of 7.5*7 m. The nine different variations were analyzed for cost and carbon emissions. The slab seems to contribute 80-85% of the cost and carbon to the total values of a building. Also, concrete is the highest contributor to the carbon emission in the building which was above 60% and formwork had a significant contribution to the total cost. The cost optimization model is found to be inversely proportional to the carbon emission optimized model which indicated that the most cost-efficient design solution does not exhibit the least carbon emission. (Eleftheriadis, et al., 2018)

The paper reviews the state of art decision making in civil engineering regarding sustainability. The review is done with consideration of various literary publications and the existing MCDM (Multi-Criteria Decision Making) approach. The analysis is done on journal articles from 2015-2017. The MCDM decision-making approach seems to be developing and growing at a significant pace over the years and pick up in research articles have been observed since 2010. The various research indicates that the MSDM approach is very robust and flexible in the assessment of various possible alternatives available in terms of sustainability and selecting the most rational option by considering all the possible tradeoffs. Also, the review indicated that the various decision-making approaches have been researched over the years to find the best possible approach for concrete problems. Therefore, the authors conclude that there is a need for more comparative studies between various available decision-making methods so that its possible to relate their advantages and disadvantages with each other and the most efficient and effective approach for concrete problems can be implemented in the future giving the best results for sustainability. (Zavadskas, et al., 2018)

The paper optimizes the life cycle cost of a single-family house in Polish climatic conditions. The influence of various choices selection is studies that can have an impact on the life cycle cost of the house. Some of these variations are ceiling to the unheated attic, ceiling to the ground floor, the orientation of the building, external wall insulations, different types and sizes of windows. The tools used for analysis and programming in MATLAB and Energy Plus. The authors have optimized the same problem with three different metaheuristic methods namely PSO (Particle Swarm Optimization), GA (Genetic Algorithm) and Optimization based on teaching and learning concepts which are TLBO (Teaching Learning Based Optimization). All the results obtained from these different search algorithms are compared and analyzed for optimum solutions. The optimization approach for the house is comprehensive and it is performed individually with different parameters for each room separately depending on what would be best for that particle room. The energy details and cost of all the analyzed materials are available. The objective function for optimization is cost function and it is the summation of present investment value, building and services operating cost, replacement and maintenance cost over the life span of the house. The results depending on the simulation were best achieved by TBLO for this house which saves about 32% of the energy. The GA has provided the least optimized results as the problem is of local search and a smaller number of simulations were used. (Grygierek & Ferdyn-Grygierek, 2019)

The author has implemented the GA (Genetic Algorithm) for a singly reinforced concrete beam for finding the most optimized solution. The cost is taken as an objective function that needs to be minimized and it includes the cost of steel reinforcement, cost of concrete and cost of formwork. The constraints are provided for the beam in the algorithm so that it satisfies all the strength and serviceability criteria according to the IS 456 (Indian Standards) building codes. Some of these constraints are deflection limitations, amount of maximum and minimum reinforcement that can be sued in the singly reinforced beam, the resistance bending moment of the beam section should be less than the factored bending moment due to the loading, designed shear strength should counteract the shear forces due to loading, limitations for maximum and minimum dimensions of the beam and finally the stirrups spacing limitations. There is a total of five design variables taken for optimization out of which four are continuous such as breadth, the effective depth of the beam, area of steel, stirrups spacing, and one is a discrete set of variables which is the strength of concrete. The constant parameters are the cost of concrete, steel, formwork, effective cover and area of stirrups. All these set of a variable defines the beam section as a nonlinearly constrained problem with mixed variables. The beam was set up in the variations of span length and loading for optimization. The various relationships and usage of materials were observed from the optimized results. The author has concluded that the results vary depending on the size and loading. Therefore, the design can choose the best suited for their needs and also the GA (Genetic Algorithm) can easily be modified depending on the site conditions and requirements. (Ajmal , 2017)

The authors presented a cost minimization of the concrete box frames which are being used in the construction of roads. The problem of the concrete box frame is formulated with a higher number of variables which are 50. To solve this high variable problem the authors have used two metaheuristic methods such as SA (Simulated Annealing) and threshold accepting. Also, two heuristic methods have been used which are random walk and descent local search. All the results obtained from all these methods have been compared with each other to analyse which of them suits the best for optimizing the design example of the box frame. The box frame has a span length of 13 meters. The economic function includes the cost of concrete, steel and formwork. The constraints that need to be satisfied are provided from the ULS (Ultimate limit state) design, SLS (Serviceability Limit State) design, geometric constraints and constructability constraints on the construction site. The authors have concluded that the threshold accepting approach for optimization provides the most optimum solution from all four methods. The results of the threshold accepting method give improved efficiency of 7.5% and 1.4% from that achieved by random walk and descent local search respectively. (Perea, et al., 2007)

The optimization study is performed on the reinforced concrete continuous beam having UDL (Uniformly Distributed Load) throughout the span length of the beam.

The optimization technique used is GA (Genetic Algorithm) which is an inbuilt tool of optimization in MATLAB. The objective function used for minimization is the cost function which includes the cost of concrete and steel used in the beam section. The algorithm is provided with constant parameters which are fix throughout the problem and given as input so that the algorithm can utilize them to find out the unknown variables using the equations such as dimensions of the beam, size and number of the steel reinforcement bars and size and the number of the stirrups. The design constraints are formulated according to the IS 456 2000 (Indian Standards) building codes for strength and serviceability criteria for beams. The technique is illustrated with a design example and the result of the GA is compared to the manually calculated results. The GA optimized results have shown less amount of steel used for the beam. (Alex & Kottalil, 2015)

The research review is done to figure the applicability of the optimization done by the engineers on the real site. There has been a lot of advancement in technology and the construction industry has also implemented some of it very successfully in practical applications. One such advancement is the optimization of structural design. Various studies have been done over the years to minimize the cost of the structure using different techniques and the authors have reviewed their use in the construction site. The various problems of optimization over the years has been divided into few categorized depending on their objective. These categories are mainly optimization of the topology of the structure, optimization of shape, optimization of size and lastly optimization of topography. The review is done for the last fifteen years, and the authors have identified that there are various assumptions made by different researchers based on their building codes. Therefore, it was not possible to directly compare them with each other. Although, authors have made a various assumption that gives a good insight about their optimization technique and their limitations for use on building site. The basic conclusion is that the advancement in technology has led the designers to opt for automatic optimization but in most cases, the solutions have been purely mathematical, and it is hard to implement them on the construction site. Also, the optimization is not done on real world building design which is much more complex and trickier compared to the simple design examples taken up for these studies. Therefore, the inclusion of more particle problems in the optimization and consideration of site practicalities are the two essential areas the design engineers have to focus on to implement it in the real world. (Aleksandar, et al., 2013)

The study presents a cost optimization of a doubly reinforced concrete beam section. The strength and serviceability criteria are satisfied according to the IS 456:2000 (Indian Standards) design codes. Also, the use of IS 13920:1993 is done for any design constraints for detailing the purpose for ductility requirements. The authors have implemented all the design constraints as discrete variables and then the solutions with the consideration of ductile detailing and without ductile detailing have been performed. The optimization tool used is the GA (Genetic Algorithm) toolbox in MATLAB. The problem is formulated as a cost objective function having constant parameters as input and having to find the various design variables satisfying all the design constraints. The design constraints used to satisfy design codes are flexure strength, maximum and minimum tension reinforcement limitations, compression reinforcement limitations, shear reinforcement limitations and spacing of shear reinforcement limitations. The results were obtained from running GA in MATLAB for various combinations of design input parameters such as span length, the strength of concrete, loading values and strength of steel. The results are satisfactory according to the authors in terms of optimization and GA is a decent tool for solving discrete problems. Also, the results of optimization have shown a difference of 3-6% between doubly reinforced beams with and without ductile detailing. (Singh & Rai, 2014)

The study indicates the importance of multi-objective optimization because of the mere fact that optimization must be performed on cost function and environmental and energy efficiency function. The cost and energy efficiency are both very important aspects of construction especially now when the high emphasis is on saving the environment and natural resources available. Also, according to the authors, the solutions obtained from optimization could be accepted based on the requirements of the project rather than just accepting the most optimized solution which may not be feasible for the ongoing project. Therefore, practicality is equally important as optimization. The study uses GA (Genetic Algorithm) for multi-objective optimization so that cost and environmental objectives can be satisfied. Therefore, the two-objective function for the study is LCC (Life Cycle Cost) which includes construction cost plus the operating cost and LCEI (Life Cycle

Environmental Impact) which includes the environmental impact of a construction phase plus life cycle environment impact due to operating of the building. The implementation technique is studied on a case study for a building in Serbia having pentagon shape geometry and various other available data including that of energy plus software. The authors concluded that efficient optimization is done using GA but there are many solutions and the designer must choose the best suited for them according to the requirement. (Milajić, et al., 2019)

The study is performed to optimize an RCC (Reinforced Cement Concrete) beam which satisfies all the criteria for strength and serviceability set by the building design codes IS456-2000 (Indian Standards). The authors have come up with a hybrid approach that includes two different search algorithms namely PSO (Particle Swarm Optimization) and GSA (Gravitational Search Optimization). Therefore, both these approaches together give out an algorithm name as PSOGSA which uses the social search from PSO and good local search capabilities of the GSA. Also, all the algorithm writing has been done using C++ computer language. The objective function to be minimized is the cost function which includes the cost of concrete, steel reinforcement for shear and flexure and cost of formwork. The optimization problem formulated by using is unconstrained and continuous. Therefore, to get better results for the given constraints, the penalty function is used which penalize the objective function if there is any kind of constraint breaking using the optimization. So, by using the penalty function the problem with constraints is converted to an unconstrained problem. The cost of steel and concrete depending on the place of construction is the same. Therefore, the cost objective function has been more simplified to a variable of the ratio between steel and concrete cost and volume variables. The constraints used in this problem is provided by the IS456-2000 (Indian Standard) building code which is for moment resistance, deflection of the beam, beam dimension, depth of neutral axis and tensile steel. The hybrid approach of PSOGSA is tested on a beam from a framed structure. The span length range between 5 to 9 m and the loading range is between 30 to 50 kN/m which gives us the five different combinations to analyze. A flowchart of different steps and processes is also proposed in the study from the start to the end of the optimization. The algorithm seems to find the optimized solution for the beam with consideration of practical aspects and the computation time is two seconds. (Chutani & Singh, 2017)

The study aims to implement a MOO (Multi-Objective Optimization) function using NSGA II (Non-dominated Sorting Genetic Algorithm). The two objective functions which need to be minimized considered in the study are LCC (Life Cycle Cost) and the LCCF (Life Cycle Carbon Footprint). Both these objective functions include the construction and lifetime cost and carbon emission throughout its life cycle which is around 60 years. Also, the authors have studied both functions with and without the renovation done on the model building and how they affect the overall values of LCC and LCCF. The optimized solution with multi-objective functions gives many optimal solutions at different points and according to the research, all of them are considered equally efficient. Therefore, the concept of the Pareto front is introduced in NSGA II so that it’s easy to pair the optimal solution from many possibilities and produce the best possible generation in the specified number of iterations. The approach is tested in a high-rise residential building built in London and the aim is to analyze the feasibility of analyzing LCC and LCCF with renovation done during the life of the building, applicability of optimization and decision-making comparison between optimization approach and traditional renovation approach. The modelling of the building geometry and thermal zones are done with SketchUp from Trimble and the studio legacy plugin. The model is exported to Energy Plus for further evaluation. A vast variety of results were obtained, and the authors have concluded the best result for LCC and LCCF from different possibilities. The results have indicated that the optimization method used for LCCF calculation has a big difference in reducing carbon footprint and the optimal solution shows a reduction of 21% in the case of the refurbished solution and around 67% in the case of un-refurbished solution. The LCC has not shown as great a reduction as it was in LCCF but there was a reduction in the optimal solution of around 5% from refurbished and 16% from un-refurbished solutions. All in all, the optimization process shows efficient solutions and results in practical construction practice. (Vasinton & Raslan, 2016)

The study presents a cost optimization of an RC slab with different support conditions namely simply supported slab, cantilever slab, one support continuous and both ends continuous. The design of strength and serviceability is performed based on the ACI (American Concrete Institute) building codes. There are three different metaheuristic optimization techniques are used which are GA (Genetic Algorithm), GWO (Gray Wolf Optimization) and PSO (Particle Swarm

Optimization). The tool used for optimization is MATLAB where the problem is formulated and computed. The cost objective function of one way reinforced concrete slab includes the cost of steel reinforcement bars, cost of concrete and cost of formwork plus finishing material and it is subjected to various constraints from ACI. These constraints are flexural, shear constraints, serviceability constraints and defection constraints. The variables to be considered for optimization are the slab thickness, reinforcement bard spacing and reinforcement diameter. The four different slab end support condition design results using three optimization methods were compared with each other and with the previous studies done on the flat slab by different authors. The study concludes that the GWO technique uses the minimum number of iterations for finding the optimal results which are 15. Therefore, GWO provides the best convergence in comparison to GA and PSO methods. The GWO and PSO optimization results for the slab with three end support conditions namely cantilever, one end continuous and both end continuous are more optimal and superior as compared to the results from GA but the result for simply supported slab is similar for all the three methods. The author's comparison with previous slab optimization studies shows that the proposed methods in the current study show the minimum values of the cost function among all the studies. (Suryavanshi & Akhtar, 2019)

The authors have used ACI 318-05 (American Concrete Institute) building codes for reinforced concrete one-way slab ribbed slab in a one-way joist floor system. The objective is to minimize the cost of the slab using the HS (Harmony Search) algorithm. The technique is illustrated with an example with six design variables namely thickness of the top slab, rib depth, rib width at the top and bottom end, rib spacing and bar diameter. The design constraints that needed to be satisfied according to ACI are shear, flexural constraints, deflection constraints, serviceability constraints and other additional constraints. The optimal design is achieved from variable possible combinations using HS and it has been found that the thickness of the top slab and the rib spacing has the greatest impact on the total cost. (Kaveh & Shakouri Mahmud Abadi, 2011)

The design of the flat slab with a drop panel is optimized using SUMT (Sequential Unconstrained Minimization Technique) in MATLAB. The design is done using IS 456-2000 (Indian Standards) building codes. The objective of the study is to minimize the total cost of the slab which includes the cost of concrete, steel reinforcement and formwork. Each of these costs is the summation of the cost of materials and the labour cost for placing them on the construction site. An example is illustrated to use the proposed technique in which the problem is formulated as NLPP (Non-Linear Programming problem). The modelling and design analysis is done using the direct design method. The penalty function and constraints are also used in problem-solving. The reduction of around 33.91% has been observed in the current example as compared to the conventional approach without optimization. The concrete and steel reinforcement material grades influence the result to a great extent. Also, the reduction in the cost has been directly proportional to the number of spans. (Patil, et al., 2013)

The design optimization of a reinforced concrete beam using GA (Genetic Algorithm) is proposed in the study. The ACI (American Concrete Institute) building codes are used to formulate the design constraints related to strength, serviceability, ductility, practicality and durability. The internal forces such as moments, forces and deformations are analyzed. The cost function is formulated to be analyzed which includes the cost of steel reinforcement and the cost of concrete. The database of steel reinforcement bars is made which had all the available bar diameters which are available in the market and are practical to install on construction sites. The methodology is illustrated on a design example of a cantilever beam with varying material properties and loading conditions. The authors concluded that the GA (Genetic Algorithm) is efficient and effective for finding the optimum solution for a constrained problem. (Yousif & Najem, 2012)

The current scenario of carbon emission in building and construction has challenged all professionals to come up with ways to reduce the emission of carbon at various levels. One such area where a significant reduction of embodied carbon can be achieved is in the designing phase. Optimal design can lead to cost savings and reduce carbon emissions. The study has presented optimization of a composite beam that satisfy all the ULS (Ultimate Limit State) and SLS (Serviceability Limit State) conditions as per Eurocode 2,3,4 and UK national annexes which are BS EN 1992-1-1, BS EN 1993-1-1, BS EN 1994-1-1:2004. MATLAB is used for optimization with its inbuilt global optimization GA (Genetic Algorithm) toolbox. The design example with five different objective functions is considered for optimization. The objectives are to minimize beam section, the overall weight of the composite beam, depth of the concrete slab, deflection of the beam and to maximize the span length. The optimization was achieved in four objective functions for embodied carbon emissions where the reduction could be seen but for the span length objective, it gave slightly higher values as the length was increased. All in all, the approach to optimize multiple objectives gives designers a very precise overview of cost and embodied carbon emission factors. (Whitworth & Tsavdaridis, 2020)

The author has performed GA (Genetic Algorithm) optimization using the inbuilt toolbox in MATLAB to optimize industrial steel building. The objective was to minimize the cost of the structure which included the cost of labour and materials, find the best topology with optimal portal frame, purlins and steel cross-sections. The design example considered for analysis in the current study is a single storey industrial building with each portal frame with two columns and two beams taken from the Indian standards database. The purlin runs horizontally connecting the two portal frames. Both horizontal and vertical bracings were not considered for optimization in the objective function. The cost objective function includes the cost of the materials, installation cost, labour cost and the corrosion resistance painting cost for steel sections, fabrication cost and fire protection cost. The constraints are provided as per the IS 800-1984 building codes and broadly covers slenderness limitations of beams, columns and purlins, axial compression, axial bending, deflection and bending stress for purlins. The optimization method has been able to achieve the most optimal cost design with appropriate topology for the design example. (Kumar, 2013)

The view that the building industry uses a big chunk of total world energy has driven the authors to work on energy optimization. The minimization of LCC (life cycle cost) is performed with a variety of various options such as different window types, window dimensions, the orientation of the building, insulation of roof, wall and ground floor. The optimization is done using multi-variable GA (Genetic Algorithm) and the tool used for its implementation in MATLAB which has incorporated optimization toolbox. The energy-based simulations are performed in Energy Plus software. The optimization is done on seven different available possibilities to figure out the optimal among them. The design example is a family house located in a temperate climatic condition. The multiple zones were created depending on the various criteria and are simulated in the Energy Plus program. The programming is done is in MATLAB where the data from energy simulation and GA have interacted. The polish standards were used for energy simulations. The building life cycle was considered at around 30 years. The results indicate that the position of the windows hugely impact the energy consumption in the building whereas the number and size of the building do not hugely correspond to the energy savings. Also, the orientation of the building can impact energy usage and the LCC (Life Cycle Cost). In the current example, the optimal orientation of the building seems to save around 1% of the total LCC. The building with a heating and cooling system does add to the initial building cost but with optimal variable selection, it could save somewhere between 7% to 34% of the energy and the cost associated with it during its lifetime. (Ferdyn-Grygierek & Grygierek, 2017)

The study is done to design and optimize a singly reinforced beam and an axially loaded column. The design code used for strength and serviceability is Indian Standards. The method of GRG (Generalized Reduced Gradient) and SQP (Sequential Quadratic Programming) are used to formulate the problem in an advanced excel program and for optimizing the solver toolbox is used. The advantages of the automatic formulation of a design problem are that it makes repetition with variable parameters easy and time-saving. The objective function for the current study is the minimization of total weight. The results of both the methods GRG and SQP are compared with each other and analyzed. The results of the SQP optimization method gives more optimal results as compared to the GRG method but the difference is not big. Therefore, the authors chose to work with the GRG method for further comparison with IS (Indian Standard) design methodology as it is easy to implement. The results show that the wider the range of variables used, the better is the optimization as the options are more. The reduction of 25% in self-weight is observed for the GRG method compared to the IS design and 37% reduction for beam and 29.57% for columns after further optimization. (Gare & Angalekar, 2016)

The flat slab is optimized using GA (Genetic Algorithm) and the designing method and constraints are taken from the IS:456 2000 (Indian Standards). The idea is to save the total cost and total weight of the slab. The problem formulation is with unconstrained minimization and constrained maximization. The result indicated saving of around 20% to 30% in material usage. (Raje & Patel, 2017)

The paper discusses the use of API (Application Program Interface) to develop an interaction between energy simulation software called Energy Plus and MATLAB. The current estimation is that the big chuck of GHG (Green House Gases) is contributed by the building and construction industry. Therefore, the energy-efficient techniques are heavily researched and more ideas on how to implement them for all the data available. Developing an API brings together the two most essential parts for energy studies in a building which are energy simulations for buildings and its management and design for research purposes. The API can be developed in C# computer language and it is very easily exported to other tools which work with the .NET library. Therefore, MATLAB can import developed API easily in .NET libraries. A small example of how to use the API in a real problem is illustrated. It has been concluded that MATLAB can optimize certain data which can then be used as an input for Energy plus through editing before the simulation in the .idf file. Also, the use of od MATLAB makes it possible to use many numbers of optimization process running parallel reducing the total time required for the optimization process. All in all, the API (Application Program Interface) makes it possible to use both tools such as MATLAB and Energy Plus to work on the same platform allowing to use of the strength of optimization toolbox and then results of the simulation to improve the energy consumption in the building and reducing carbon emission. (Gordillo, et al., 2020)

The optimization of a flat slab is performed using a genetic algorithm. The design of the flat slab is done using BS 8110 (British Standard) for strength and serviceability. The objective function adopted for the optimization is the total cost including the cost of the concrete, cost of the steel reinforcement and cost of formwork. The technique is implemented using four different slab design examples which are flat slab with edge beam, flat slab without edge beam, flat slab with edge beam and flat beam without edge beam. The results indicate that the slab without edge beam provides the most optimal solutions for different span length range under the live load of 3.5 kN/m^2 as compared to the other three slabs. Also, the span length is directly proportional to the total cost of the slab and the column dimensions are inversely proportional to the thickness of the slab. Also, the flat slab without an edge beam gives out the most efficient solution when the span length and loading is kept constant, and the dimensions of the columns are changed. The authors also analyzed the effective depth to span ratio for each design example such that they give the optimum result and the percentage of cost contribution from formwork so that the particular type of slab is optimized. (Galeb & Jennam, 2015)

The total cost of the beam is optimized which includes the cost of steel, cost of concrete and cost of formwork. The constraints are developed from the design criteria of ACI (American Concrete Institute) building codes. The different cross-section types of RC (Reinforced Concrete) beams are considered for optimization namely rectangular, trapezoidal, triangular, inverted triangular and inverted trapezoidal. All the beams are provided with an external bending moment with varying values of safety factors which implement a different range of moment values. The two numerical examples are also illustrated for simply rectangular beam, continuous rectangular and triangular beam. The results have stated that the margin of safety has a direct impact on the cost of the beam materials. The triangular beam cross-section shows an optimum design solution for total material cost. The cost reduction is 12% and 37% as compared to the rectangular and trapezoidal sections respectively. (Al-Ansari, 2013)

The cost optimization using the GA (Genetic Algorithm) from MATLAB optimization toolbox is performed for a space frame and plane frame made from RCC (Reinforced cement concrete) satisfying ACI 2011 (American Concrete Institute) codes. Various design variables are considered during the optimization which relates to the dimensions of the cross-sections, steel reinforcement and topology of the frames. The axial loading, biaxial loading and uniaxial loading are considered for designing beams and columns. The cost objective function is minimized which includes the cost of concrete, steel reinforcement and formwork. The structural analysis is done using STAAD Pro 2016. Two numerical examples for space frame and plane frame are analyzed with the formulated problem equations. The results of the cost function are increased to 2% if the breath of the column and beam is kept constant during the optimization. (Chen, et al., 2019)

2.2 Critical Review of Literature

The various studies have concluded that the optimization process certainly gives out better results than the results obtained with traditional methodology. All the optimization techniques that have been developed over the years has a different concept and logic of working. Some of these methods are GA (Genetic Algorithm), HA (Harmony Search), SA (Simulated Annealing), SQP (Sequential quadratic programming), COA (Cuckoo Optimization Algorithm), ACO (Ant Colony Optimization), ANN (Artificial Neural Network), TLBO (Teaching Learning Based Optimization), BA (Bat Algorithm) etc. All these metaheuristic techniques depend upon the type of problem formulation, constraints that need to be met and the type of variables. The optimization has shown improved results for concrete designing in individual elements. Although, the various methods have different positives and negatives of their own and selection of optimization methods according to the problem is crucial. The BIM (Building Information Modelling) has been widely implemented in the studies to accommodate all the processes involved from start to finish. The commonly used structural engineering software used in the research are Autodesk RSAP, Dlubal RFEM, STAAD Pro. Finally, the optimization algorithm can be successfully written on various programming language platforms such as python, C#, MATLAB, FORTRAN, Java etc.

2.3 Research Gap

The complexity and number of variables in concrete designing make it harder to automate frame structures or any specific structural elements such as beams, columns and slabs. Therefore, there are significantly a smaller number of studies done on concrete structure automation as compared to steel structures. The design problem can be formulated in various ways with different logics to achieve the target of cost and carbon optimization which makes it appealing to solve the design differently by automation through coding. Therefore, the research gap for analysing the entire design exists and will be explored in this research. The current research indicates that the genetic algorithm would be a robust optimizing technique to work with when the constraints are non-linear and can be implemented in MATLAB.

3 Methodology

The GA (Genetic Algorithm) has been indicated by many above studies to be a successful metaheuristic method with great precision, efficient optimization, ease of implementation and robustness. Also, multi-objective, multi-search and different types of constraints such as linear, nonlinear and discrete problems can be implemented effectively. Therefore, further research is done using the concept of GA (Genetic Algorithm) with the help of the inbuilt optimization toolbox function in MATLAB. The structural analysis is done using a FEM (Finite Element Method) software named Dlubal RFEM for the structural element. The BIM (Building Information Modelling) software such as Revit/Tekla is used to visualize the data related to geometry, loading and support conditions etc. Also, the BIM model provides better visualization for the involved professionals and clients. The optimization process is implemented using MATLAB programming. The technique is used in various examples such as the design of an RC (Reinforced Concrete) beam, column, slab and structural frame. The problems can be formulated in mathematical equations with various concepts and there are a different number of techniques and methods to achieve the optimization. Therefore, further research will present various objective functions with different constraints to achieve optimization. Also, these three are significantly interconnected with each other. In formulating the design problem, the Eurocode is applied, and all the requirements of ultimate limit state and serviceability limit states will be successfully met. In the end, the comparison will be made with the manual calculations and the different objective functions to access the quality of the results and the optimization process as a whole.

4 Automation Concept

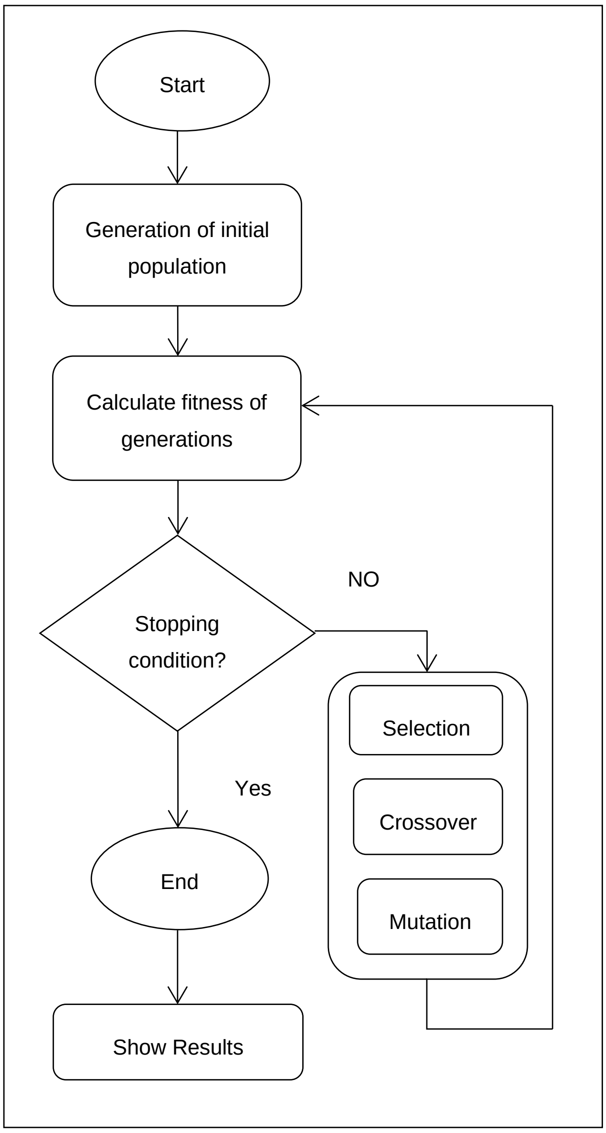
4.1 Genetic Algorithm

The theory of evolution depends on the principle of ‘survival of the fittest’ proposed by Charles Darwin in 1859. This theory explains that every species evolves to be the better version of itself in terms of competing, surviving and reproducing using the natural inheritance and selection possibilities. The theory gave birth to its application in solving engineering problems by using algorithms. This evolutionary computation was applied where the optimal solution was searched from various possibilities within a certain time frame. (Sivanandam & Deepa, 2008)

The Genetic algorithm was developed by John Holland in 1975 from the theory of evolution principle proved to be successful in solving the vast majority of engineering problems in various disciplines. Structural engineering is one such field where it has been implemented over the years and delivered some effective results as can be seen from the literature review section. There is a big scope for optimization in structural engineering problems which can help engineers to make better decisions in every way possible.

The genetic algorithm in numerical problems requires an objective function that needs to be searched for the optimal solution. The GA process has three main operators namely selection, crossover and mutation. The selection in GA happens depending on the fitness of the objection function. The selection can be controlled with various methods such as roulette wheel selection method, tournament selection, Boltzmann selection etc. The crossover is done from the selected parents to produce a better solution (child) and can be executed by using various methods such as uniform crossover, heuristic crossover, multi-point crossover etc. Finally, the mutation is done in the new solution where one or more genes are altered to improve the solution and can be performed using methods such as uniform mutation, non-uniform mutation, gaussian, power mutation, varying probability of mutation etc. (Mirjalili, 2019)

The generalized flowchart for the GA and its pseudo-code follows:




```
INPUT DATA
  OBJECTIVE FUNCTION
  CONSTRAINTS
RUN GA
  POPULATION INITIALIZATION
  EVALUATE FITNESS
  WHILE STOPPING CRITERIA NOT MET DO
    SELECTION OF FITTEST PARENT
    CROSSOVER
    MUTATION
    GENERATION OF NEW CHILD
    EVALUATE FITNESS OF NEW CHILD
    REPLACE NEW CHILD WITH LEAST FIT
  END
OUTUT RESULT FOR BEST SOLUTION
END GA
```

Figure 3: GA flowchart (left) and GA pseudocode (right)

The basic GA can solve only unconstrained problems, but the structural engineering problems have various constraints set by the building design codes such as EC, ACI, IS, BS etc. These constraints provide lower and upper bound for almost all the equations involved. Therefore, when the objective function is constrained, it needs to be converted into an unconstrained problem so that GA can solve it. To implement the constraints, the penalty function can be used for the required results. The MATLAB optimization toolbox is an efficient and effective tool to formulate an optimization problem for non-computer science professionals as it has almost all the inbuilt options for generations, selection, crossover, mutation, penalty function implementation.

4.2 BIM (Building Information Modelling)

The building information modelling (BIM) is defined by (NBIMS-US, 2005) committee as “a *digital representation of physical and functional characteristics of a facility. As such it serves as a shared knowledge resource for information about a facility, forming a reliable basis for decisions during its life cycle from inception onwards*”.

The BIM is a tool that enables industry professionals to build a 3D model-based approach such that it helps them to coordinate between different specialists, manage the documentation and simulation during the building life cycle. Therefore, it has bridged a gap between the professionals and the investors or clients by providing them with a visualization of the project before its built and later using the data for the construction of the project. It serves as a data source for everyone involved in the project. The BIM can be broadly supporting the different processes such as Planning, Designing, Building and maintenance for the construction of a project by the data it contains. Therefore, BIM is a perfect tool for the industry and its use has been proved to be of great assistance for all the professionals working on a project.

The use of BIM has played a vital role in structural engineering by providing the structural engineers with broad information and visualization of the project so that they can make better decisions. Also, BIM data helps structural designers for making the design more optimum, accurate, time-efficient, error less and having a high level of constructability. The design professionals can efficiently document, detail and fabricate structural systems of a project.

The current project will make use of the BIM software Revit for extracting the data related to the geometry of the structural frame model and structural elements, loading data, support conditions and material properties. This data will be then analysed for internal forces in the FEM software from (Dlubal Software , 1987) called RFEM. The exchange of data between REVIT and RFEM takes place through inbuilt RF-COM which imports data from REVIT for structural analyses to RFEM and then the analysed results are imported back to REVIT automatically. Now, the data from this REVIT and RFEM software will be exported to MATLAB with the help of (*.csv)* file format which is easily readable to formulate optimization problems and solve them using the inbuilt optimization toolbox. The framework provides an insight into the progression of the project at various levels and interaction between different software. The general BIM framework is shown below:

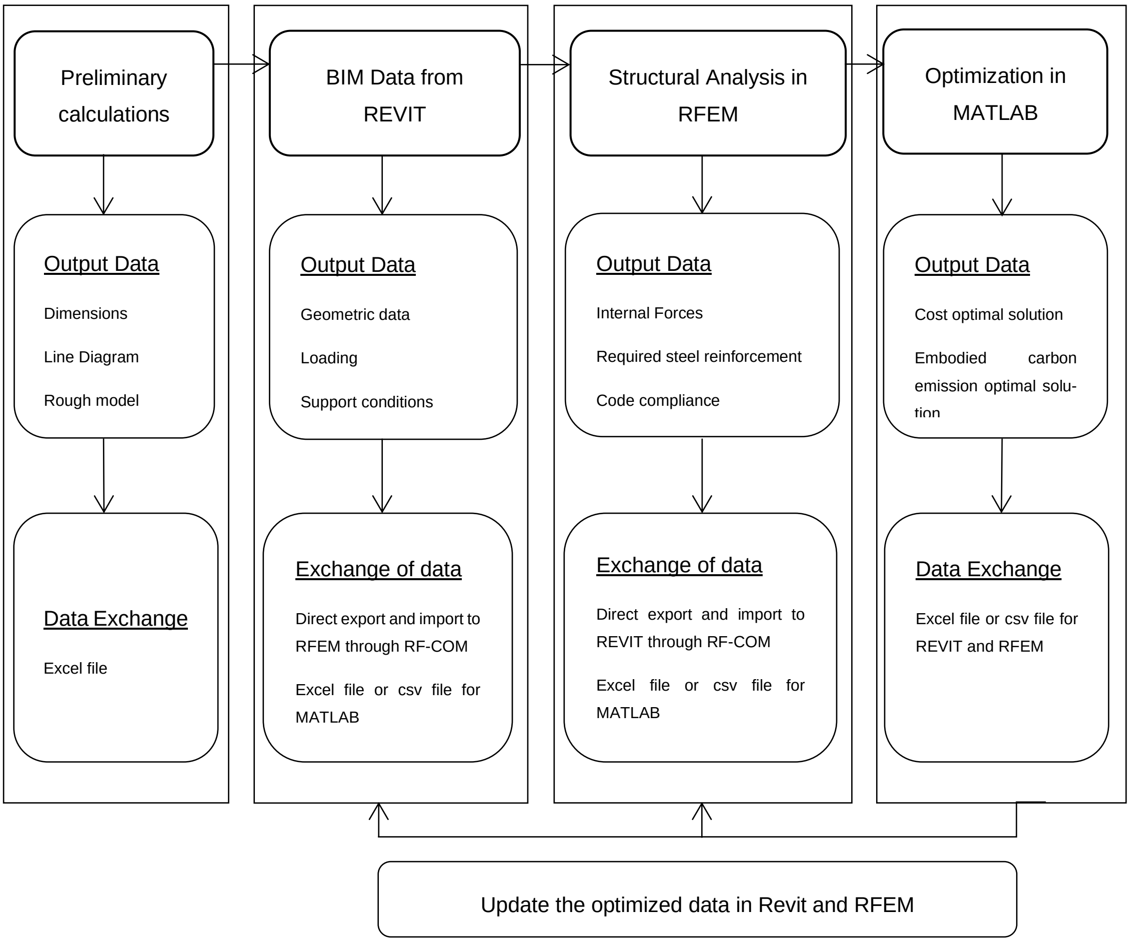


Figure 4: Generalized BIM-based framework

There is a possibility to depict the entire methodology with the help of a general flowchart showing the idea of the thesis from start till the end including all the processes such as BIM modelling in Revit, structural analysis in RFEM, formulation of objection function, formulation of constraints as per the building design code EC (Eurocodes), optimization process and finally analysing the results obtained for cost-optimized model and the embodied carbon emission optimized model. Also, we can get an insight into better decision making with the help of the results obtained. The flowchart is shown below:

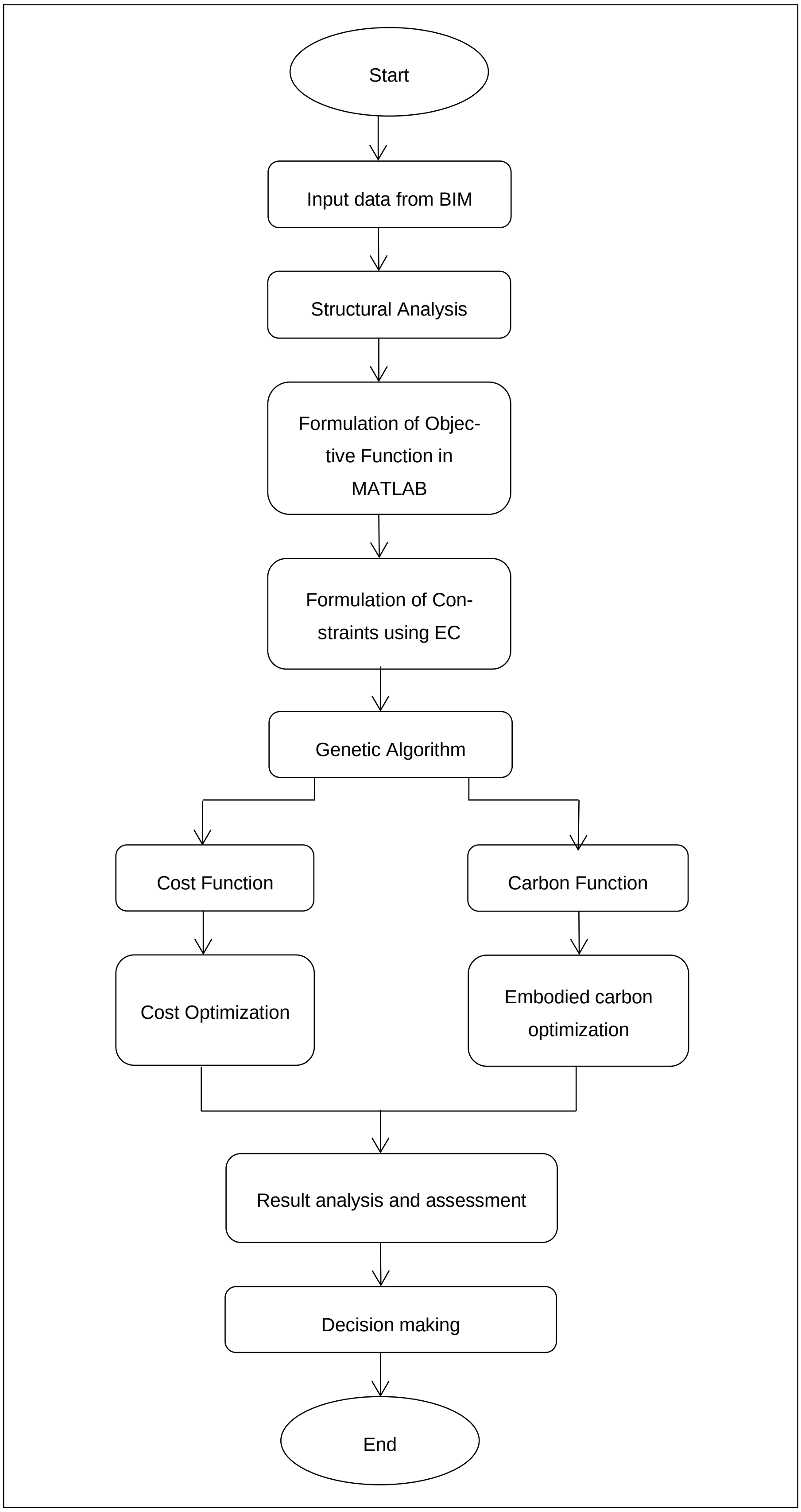


Figure 5: Flowchart for the optimization process

4.3 MATLAB, RFEM and REVIT

MATLAB is a platform that uses matrix and vector-based programming to develop algorithms for solving various mathematical and engineering problems. The computing of a wide variety of problems can be achieved with the help of interactive environments and programming platforms. The optimization toolbox provides various solvers such as GA, PS, SA, pattern search, the surrogate, the multi-start and global search for searching a single or multi-objective function with various types of constraints for an optimal solution which has multiple numbers of local maxima and minima. For our multi-objective problem for cost and carbon optimization, the technique of the Pareto front can be implemented with the use of GA to search for the optimal solution. (The Matworks Inc., 2004-2018)

The RFEM is a structural analysis software developed by Dlubal which analyse the structures using finite element analysis and it has a modular system. The 2D and 3D structural models consisting of members, walls, plates, solids etc. can be modelled with ease for calculation of internal forces, deformations, stresses etc. This information can then be utilized in the add-on modules for a specific purpose such as calculation of concrete members, surfaces, column, steel model calculation, timber analysis etc. The modular system allows the user to modify the model recalculate it in the add-on modules. (Dlubal Software , 1987)

The Revit is building information modelling software that helps collaborate and coordinate between professionals such as architects, engineers, designers, builders and other construction specialists. The Revit can help realize multiple disciplines of a project such as designing, planning, fabrication, optimization and lastly building. The cloud services help all the individuals involved in the project to work independently at the same time and update the changes or corrections in any area including scheduling. Also, it brings strong accountability in the profession as everyone is assigned with tasks and it is transparent for everyone to see the results. The interaction of Revit with other construction-related software are highly developed and made easy. We will be using its interaction with structural analysis software RFEM for the current project. (Autodesk Revit, 2002)

5 Modelling and Calculations

5.1 Beam Design

A beam is designed for flexure and shear by using stress blocks from Eurocode and UK national annexes. The procedure for the concrete class less than C50/60 is as follows:

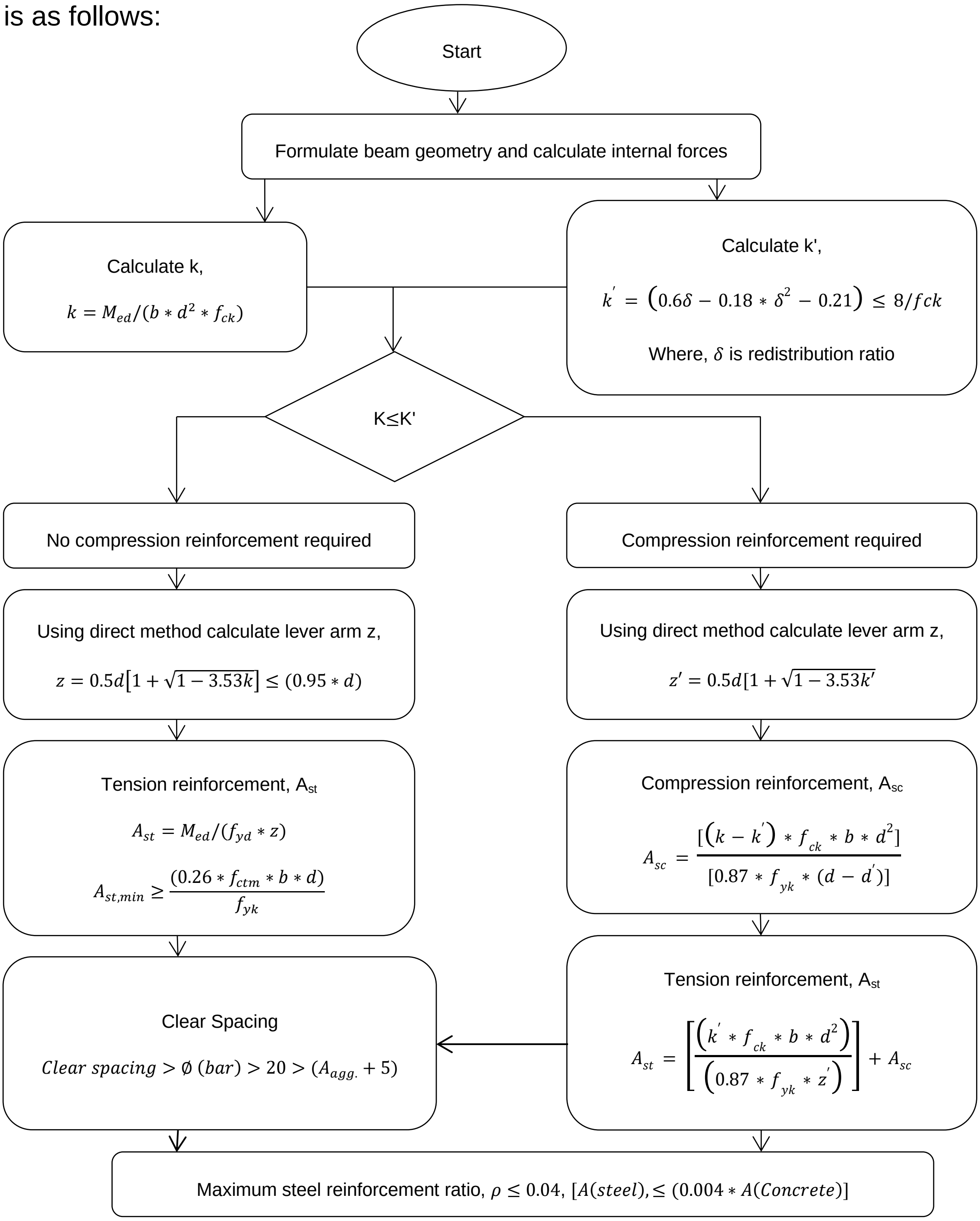


Figure 6: Flexure design flowchart

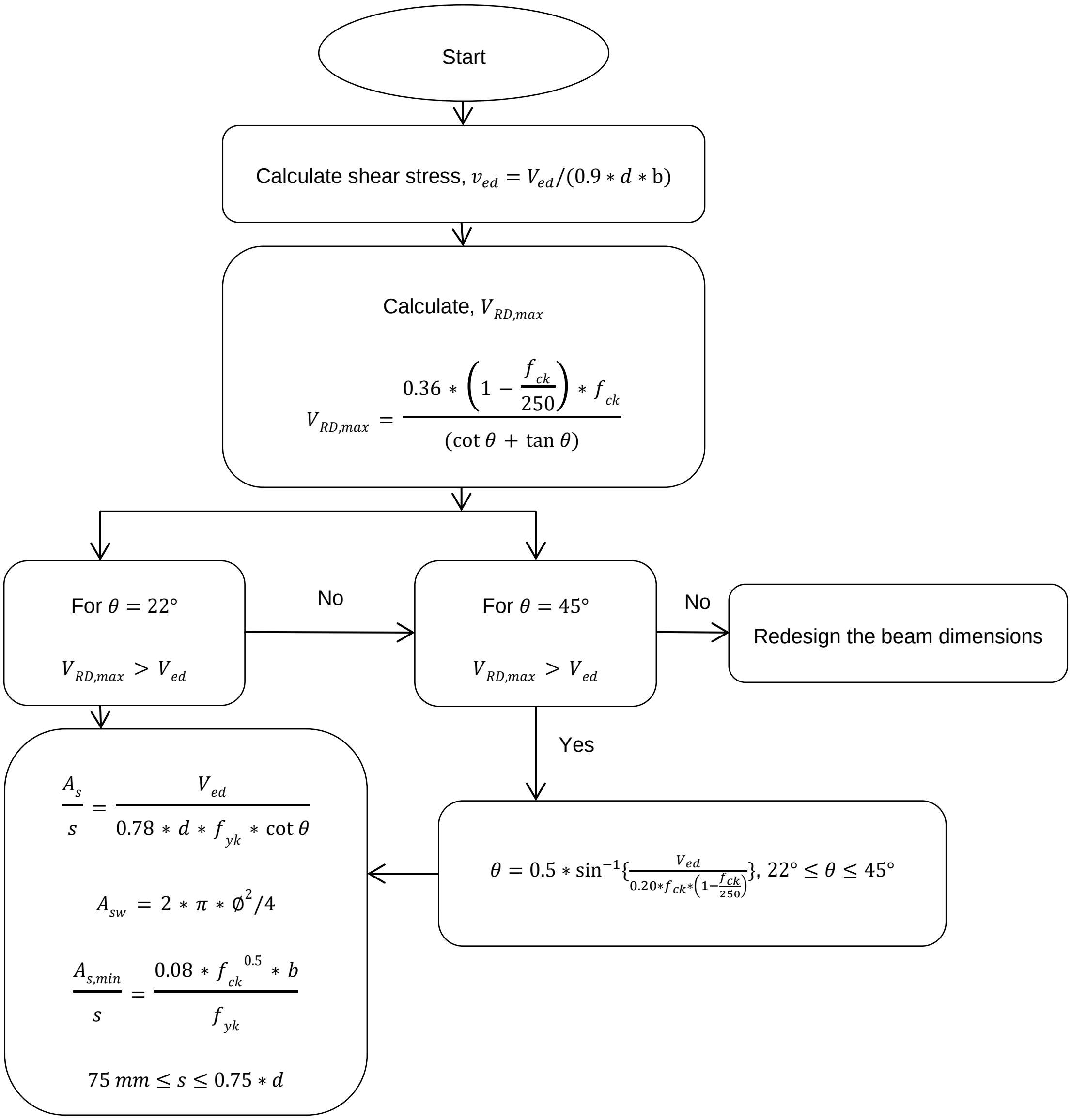


Figure 7: Shear design flowchart

The above design helps us formulate a beam design problem with input data, variables, constants, constraints and known values of cost and embodied carbon of each material. The preliminary design is done, and the beam is modelled in Revit and analysed in RFEM. Further optimization is carried out in MATLAB with the detailed design given by Eurocode. The objective function is formulated for cost and embodied carbon for the problem to be solved as two single objective GA or GA multi-objective optimization. The MATLAB is provided with constant parameters to find variable parameters which in turn minimize the objective function satisfying design constraints.

A reinforced cement concrete beam is taken as an example to illustrate the proposed optimization procedure. The beam data used, and the line diagram of the beam is given below:

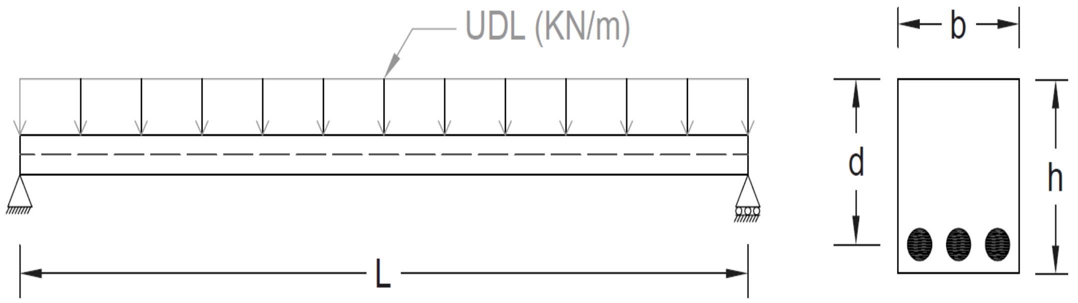


Figure 8: Beam line diagram

Given data:

Length of the beam (l) = 6000 mm

Live load on the beam (L.L) = 20 kN/m

Variable data:

Breadth of the beam (b) = (200, 225, 250, 275, 300, 325, 350, 375, 400, 425,450, 475, 500) mm

Depth of beam (h) = (500, 525, 550, 575, 600, 625, 650, 675, 700, 725, 750) mm

Characteristic strength of concrete (f_{ck}) = (20, 25, 30, 35, 40, 45, 50) N/mm^2

Characteristic strength of steel (fyk) = (500, 550, 600) N/mm^2

Diameter for longitudinal tensile reinforcement ($\emptyset_{st}$) = (12, 14, 16, 20, 25, 28, 32) mm

Diameter for longitudinal compressive reinforcement ($\emptyset_{sc}$) = (12, 16, 20, 25, 28, 32) mm

Diameter of shear reinforcement ($\emptyset_{s}$) = (6, 8, 10, 12, 14) mm

Number of bars (n) = (2, 3, 4,5,6)

Theta (θ) = (22 - 45) degrees

Spacing (s) = (75 – 600) mm with step size of 5 mm

<u>Objective function:</u>

Cost objective function f(x) = [(V_c*C_c) +(V_s*C_s) +(V_f*C_f)]

Embodied carbon objective function h(x) = [(V_c*E_c) +(V_s*E_s) +(V_f*E_f)]

Where, V_c, V_s, V_f is the volume of concrete, steel and formwork respectively

C_c, C_s, C_f is the cost of concrete, steel and formwork respectively

E_c, E_s, E_f, embodied carbon emission for concrete, steel and formwork

<u>Constraints:</u>

Constraint function, g(x) = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13})

g(X_1) for lever arm, $0.82 * d \le 0.5d\left[1 + \sqrt{1 - 3.53k}\right] - (0.95 * d) \le 0$

g(X_2) for the area of tensile steel, $\frac{(0.26 * f_{ctm} * b * d)}{f_{yk}} \le \frac{M_{ed}}{f_{yd} * z} \le 0.04 * (b * h - A_{st})$

g(X_3), $(0.6 * \delta - 0.18 * \delta^2 - 0.21) - 0.168 \le 0$

g(X_4), $\left\{0.5d\left[1 + \sqrt{1 - 3.53k'}\ \right]\right\} - (0.82 * d) \le 0$

g(X_5), $\frac{d'}{d} - 0.171 \le 0$

g(X_6), $\left(n * \pi * \frac{\emptyset_{st,prov.}{}^2}{4}\right) - (M_{ed}/(f_{yd} * z)) > 0$

g(X_7), $\left[\frac{(k' * f_{ck} * b * d^2)}{(0.87 * f_{yk} * z')}\right] + \left[\frac{[(k - k') * f_{ck} * b * d^2]}{[0.87 * f_{yk} * (d - d')]}\right] - 0.04 * (b * h - A_{steel}) \le 0$

g(X_8), $\left(n * \pi * \frac{\emptyset_{st,prov.}{}^2}{4}\right) + \left(n * \pi * \frac{\emptyset_{sc,prov.}{}^2}{4}\right) - \left[\frac{(k' * f_{ck} * b * d^2)}{(0.87 * f_{yk} * z')}\right] + \left[\frac{[(k - k') * f_{ck} * b * d^2]}{[0.87 * f_{yk} * (d - d')]}\right] \ge 0$

g(X_9), $\left(n * \pi * \frac{\emptyset_{st,prov.}{}^2}{4}\right) * \left(f_{yd} * z\right) - M_{ed} > 0$

g(X_{10}), $22° \le 0.5 * \sin^{-1}\left\{\frac{V_{ed}}{0.20 * f_{ck} * \left(1 - \frac{f_{ck}}{250}\right)}\right\}] \le 45°$

g(X_{11}), $\left[\frac{V_{ed}}{0.78 * d * f_{yk} * \cot\theta}\right] - \left[\frac{0.08 * f_{ck}{}^{0.5} * b}{f_{yk}}\right] \ge 0$

g(X_{12}), $s - 0.75 * d \le 0$

g(X_{13}), $\left(\frac{A_{s,prov.}}{s}\right) * 0.78 * d * f_{yk} * \cot\theta] - V_{ed} > 0$

The cost of the materials such as concrete (Mister Concrete, 2017), steel (MEPS International Ltd., 1979) used is average as it is fluctuated hugely with a lot of variables and is shown in the table below. Also, the embodied carbon footprint according to the materials such as concrete (Kim, et al., 2016), steel (Clark & Bradley, 2013) is provided in the table.

Concrete Strength, f_{ck}	Cost	Average Concrete Cost	Steel Reinforcement Cost	Formwork
N/mm^2	$Euros/m^3$	$Euros/m^3$	Euros/kg	$Euros/m^2$
20	95	110	0.8	6
25	100			
30	105			
35	110			
40	115			
45	120			
50	125			

Table 1: Cost of materials

Concrete Strength, f_{ck}	Carbon	Average Concrete Embodied Carbon	Steel Reinforcement Embodied Carbon	Formwork (Aluminum)
N/mm2	kg-Co_{2e}/m^3	kg-Co_{2e}/m^3	kg-Co_{2e}/kg	kg-Co_{2e}/kg
20	245	338.88	0.87	0.79
25	295.127			
30	355.6			
35	358.5			
40	362.7			
45	369.66			
50	385.6			

Table 2: Embodied carbon emissions of materials

The design is formulated with all the given design data and the constraints according to Eurocode in MATLAB. The best solution for cost and embodied carbon emission is analysed separately using single-objective GA optimization with elitism. The elite count of individuals that survive to the next generation is given by the formula:

$$EC = 0.05 * maximum\ [minimum\big((10 * number\ of\ variabes, 100), 40\big)]$$

Due to the availability of nonlinear constraint, the population is taken as a double vector and the initial population is created using constraint dependent creation function which automatically selects the starting population best suited for the constraints provided. The fitness of the population is sorted using the rank scaling where all the individuals are given a rank based on their performance for the objective function. The best-ranked individual is one and next best with increasing orders. The scaled individuals are then chosen for next-generation using the stochastic uniform method. The mutation and crossover function are constraints dependent as well. The results of the optimization process can be seen in the pictures below.

The augmented Lagrangian penalty function is also implemented if the constraints are not satisfied. The initial penalty used is 10 with a penalty factor of 100. The number of iterations performed by the GA is given by the formula:

$$Generation = 100 * Number\ of\ variables$$

The stopping criteria are set by the average change in values of each iteration. The algorithm stops if the function tolerance value is less than 10^-6 and the constraint tolerance is less than 10^-3. The number of iterations taken to achieve the results is 224 in the case of cost optimization.

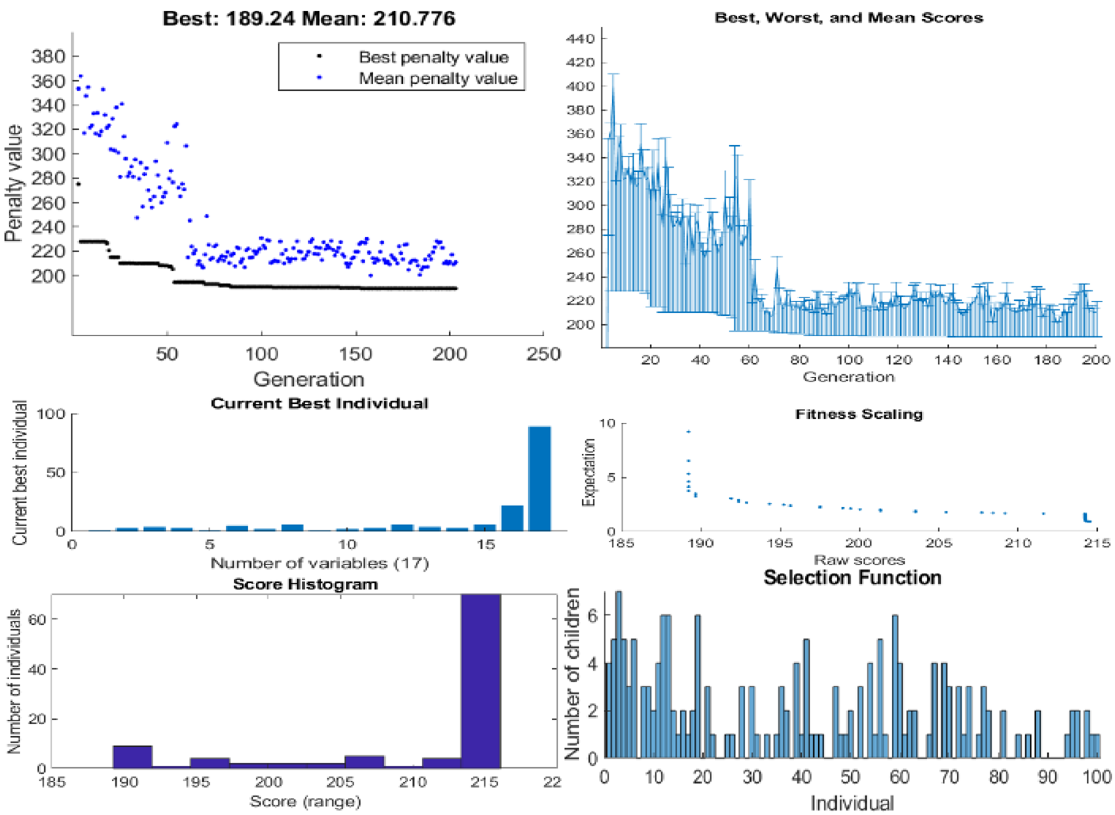


Figure 9: Cost single objective GA results (Mathworks, 1984)

The beam is designed manually, using the structural analysis software RFEM and MATLAB to check the effectiveness of the GA. The results obtained for the cost objective function are as follows:

Description	Manual	RFEM Calculations	GA Optimization (Cost)
Breadth (mm)	250	200	200
Depth (mm)	600	500	550
Tensile Steel	#4 of 16 mm	#3 of 12 mm	#4 of 14 mm
Compressive Steel	NA	NA	NA
Hanger bars	#2 of 8 mm	#2 of 8 mm	#2 of 8 mm
Spacing (mm)	15 links at 400	21 links at 300	12 links at 515
Shear Reinforcement	2-legged 8 mm	2-legged 6 mm	2-legged 8 mm
Total Area (mm^2)	2411.52	2549.68	1921.68
Cost (Euros)	240	204	191
Carbon (kg-CO_2)	406	308	305

Table 3: Cost single objective beam results (Mathworks, 1984)

The detailing of the beam data obtained initially with manual calculations and later using GA for cost and carbon optimization can be done in Revit for better visualization and detailing. The data can be used for further decision making about which design to be considered according to the requirements of the project.

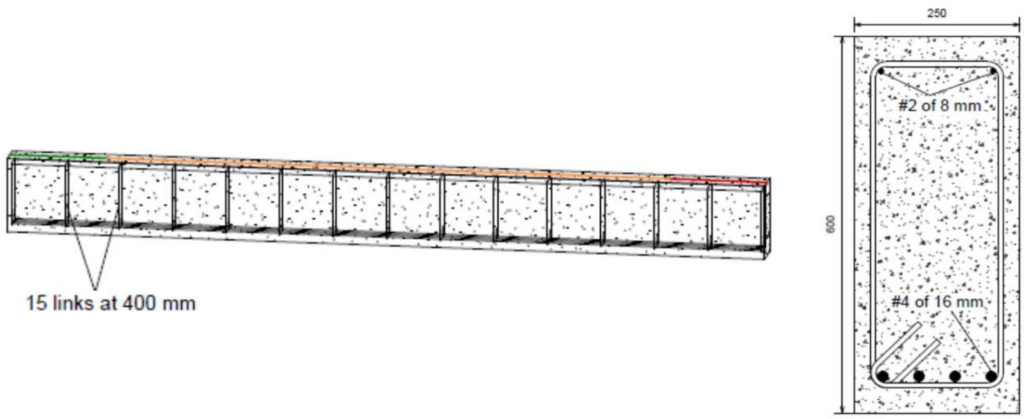


Figure 10: Revit detailing for manually calculated solution

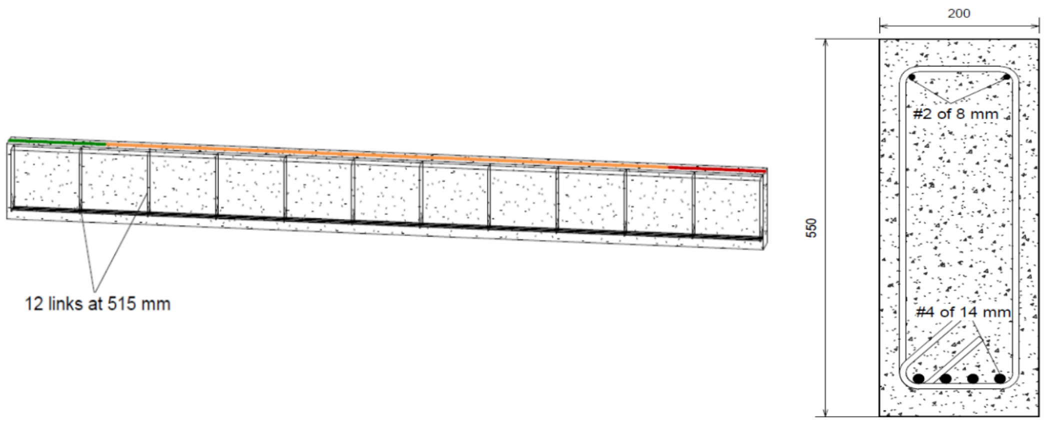


Figure 11: Revit detailing for cost optimized solution

The MATLAB results for embodied carbon fitness function using single-objective GA are shown below. The optimization is performed with a higher number of initial populations which is 500 as compared to the population size of the cost objective optimization where the initial population was just 100. This is done to try and achieve better efficiency with a much greater number of possible populations. As can be seen from the results the GA has found a slightly better solution for both cost and carbon when the population size is increased but it does take a few seconds more to compute. The number of iterations taken to achieve the optimized result is 118. A total of 17 variables was used in the optimization and all the constraints were satisfied which are mentioned in the design procedure using Eurocodes.

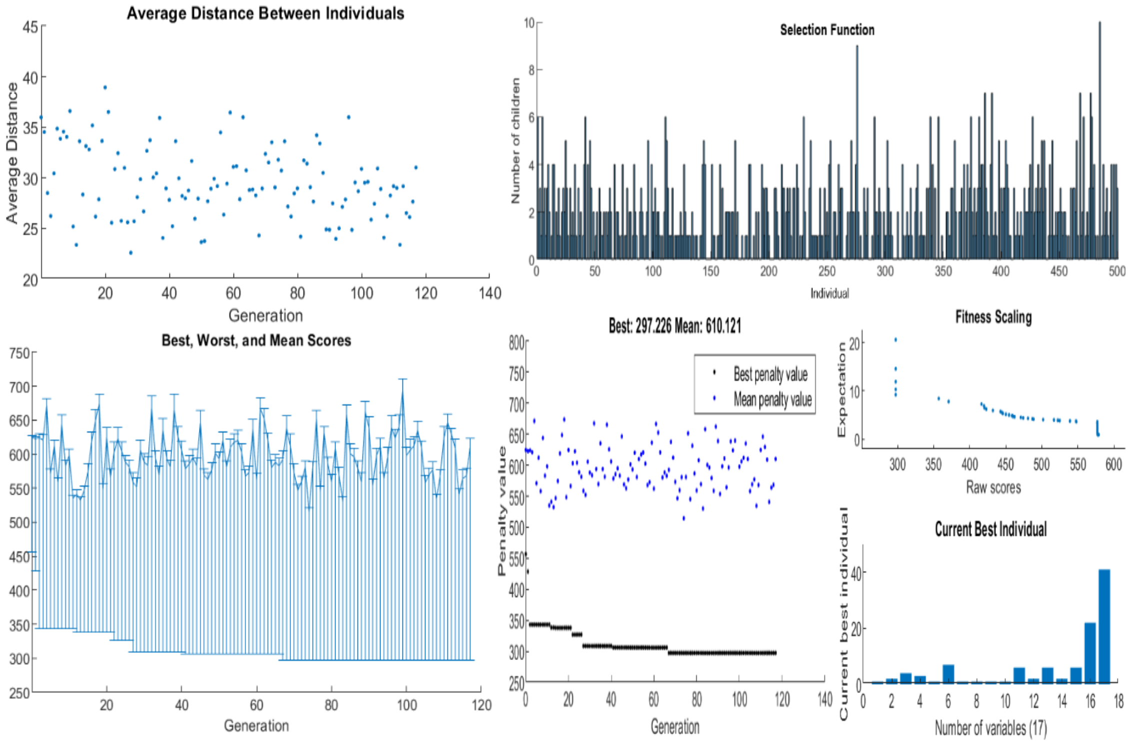


Figure 12: Carbon single objective GA results (Mathworks, 1984)

Description	Manual	RFEM Calculations	GA Optimization (Carbon)
Breadth (mm)	250	200	200
Depth (mm)	600	500	525
Tensile Steel	#4 of 16 mm	#3 of 12 mm	#6 of 12 mm
Compressive Steel	NA	NA	NA
Hanger bars	#2 of 8 mm	#2 of 8 mm	#2 of 8 mm
Spacing (mm)	15 links at 400	21 links at 300	22 links at 275
Shear Reinforcement	2-legged 8 mm	2-legged 6 mm	2-legged 6 mm
Total Area (mm^2)	2411.52	2549.68	2022.16
Cost (Euros)	240	204	189
Carbon (kg-CO_2)	406	308	298

Table 4: Carbon single objective beam results (Mathworks, 1984)

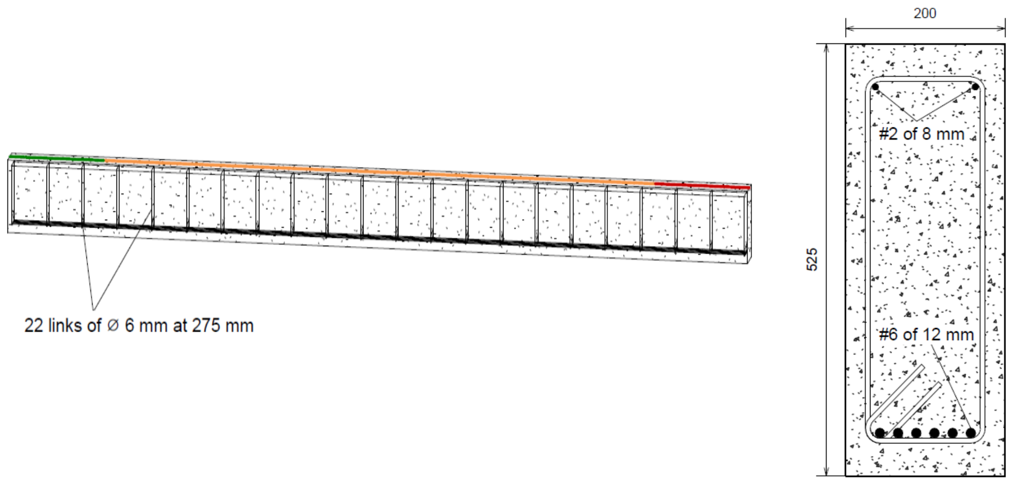


Figure 13: Revit detailing for carbon optimized solution

Description	Manual	RFEM	Cost Optimized	Carbon Optimized
Concrete Strength fck (N/mm^2)	30	30	40	25
Steel Strength f_{yk} (N/mm^2)	500	500	600	600
Theta (θ) degree	22	22	22	22

Table 5: Additional variable data

5.2 Column Design

The column is a structural member which transfers loads from slab and beams to the foundation. Therefore, the columns are primarily compressive members. The design of the columns using Eurocodes are broadly decided into two different structural systems which are braced and unbraced columns. The braced columns are considered where all the horizontal loading is transferred to the foundations using additional lateral load-bearing elements such as shear walls and bracings. Whereas unbraced columns have no lateral load-bearing additional elements, and the columns alone are used to transfer the lateral loads to the foundation. The design procedure for both types of the structural system is different and can be seen from the below figures:

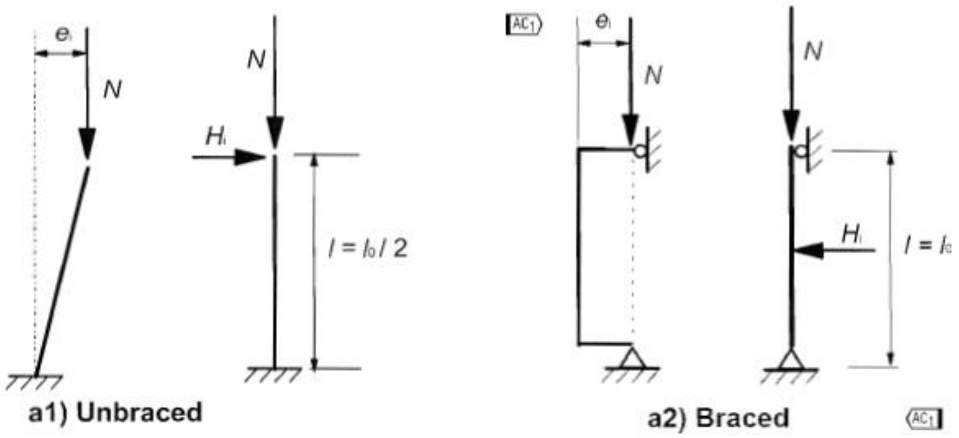


Figure 14: Isolated column member (EN-1992-1-1 (CEN), 2004)

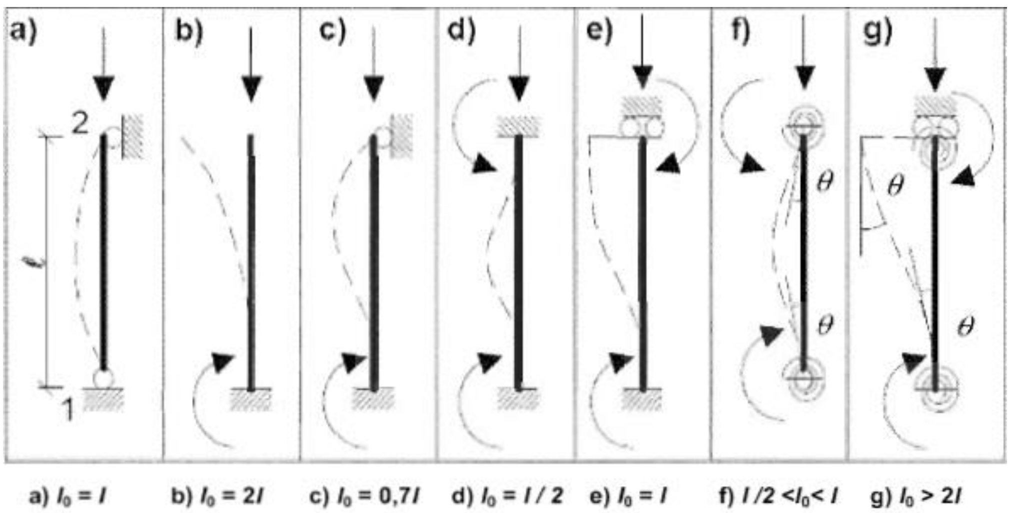


Figure 15: Effective length of column (EN-1992-1-1 (CEN), 2004)

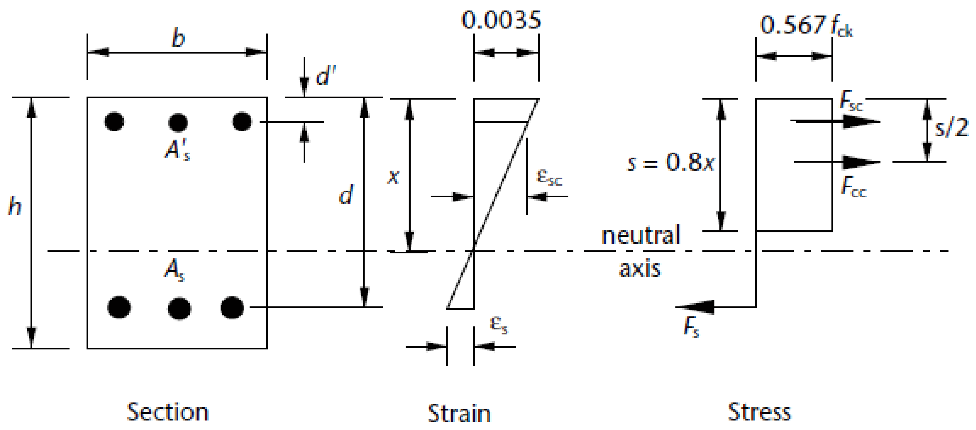


Figure 16: Column section (Mosley, et al., 2012)

The equations for calculating steel reinforcement can be given by:

$$N_{ED} = F_{cc} + F_{sc} + F_s$$

$$N_{ED} = 0.567 f_{ck} * b * s + f_{sc} * A_s' + f_s * A_s$$

$$s = 0.8 * x$$

$$M_{ED} = F_{cc}\left(\frac{h}{2} - \frac{s}{2}\right) + F_{sc}\left(\frac{h}{2} - d'\right) - F_s\left(d - \frac{h}{2}\right)$$

$$N_{Rd} = 0.567 f_{ck} A_c + 0.87 f_{yk} A_s$$

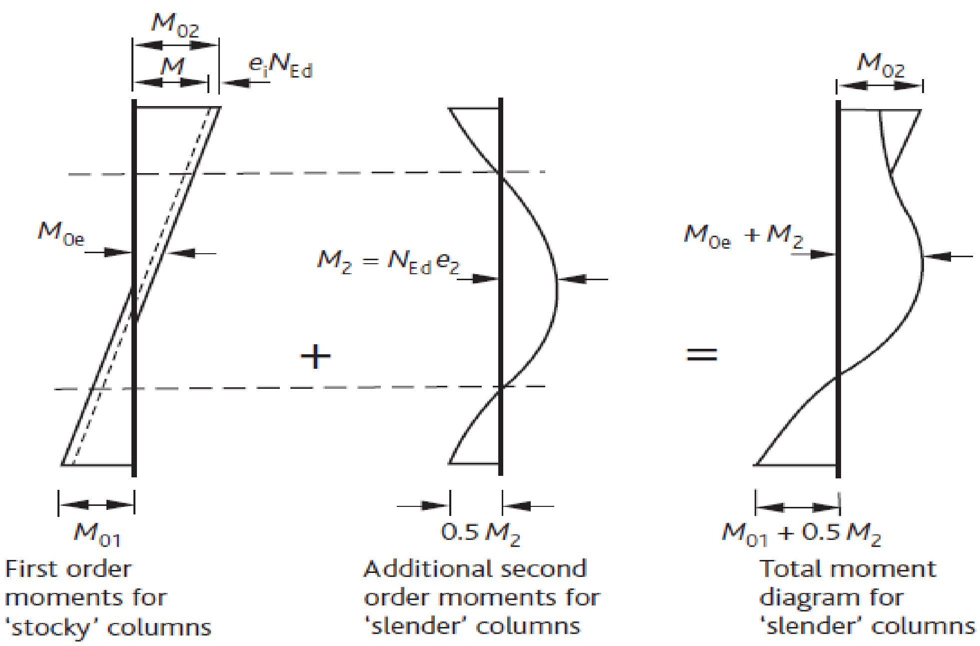


Figure 17: Design bending moment diagram (Bond, et al., 2006)

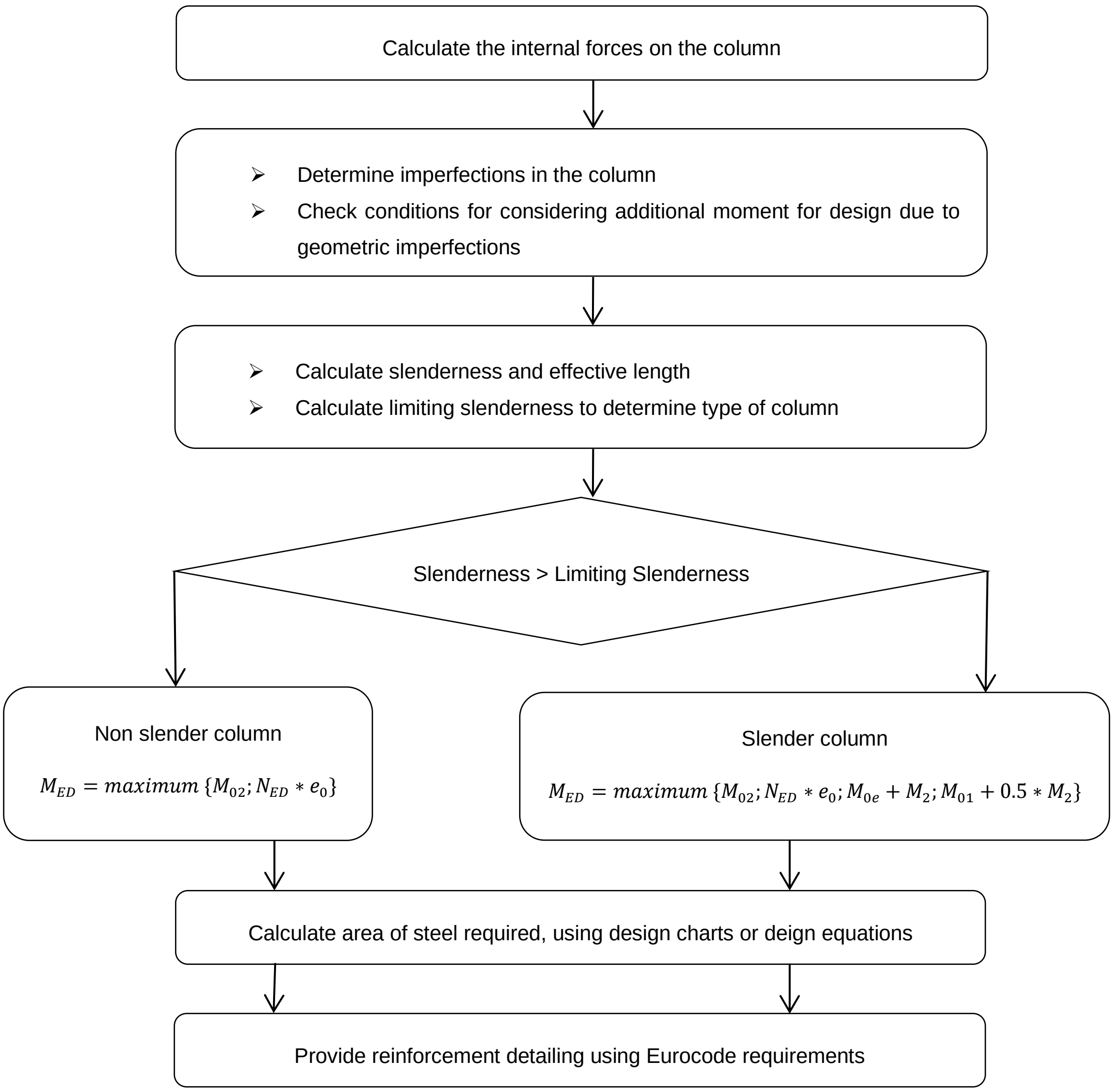


Figure 18: Generalized column design

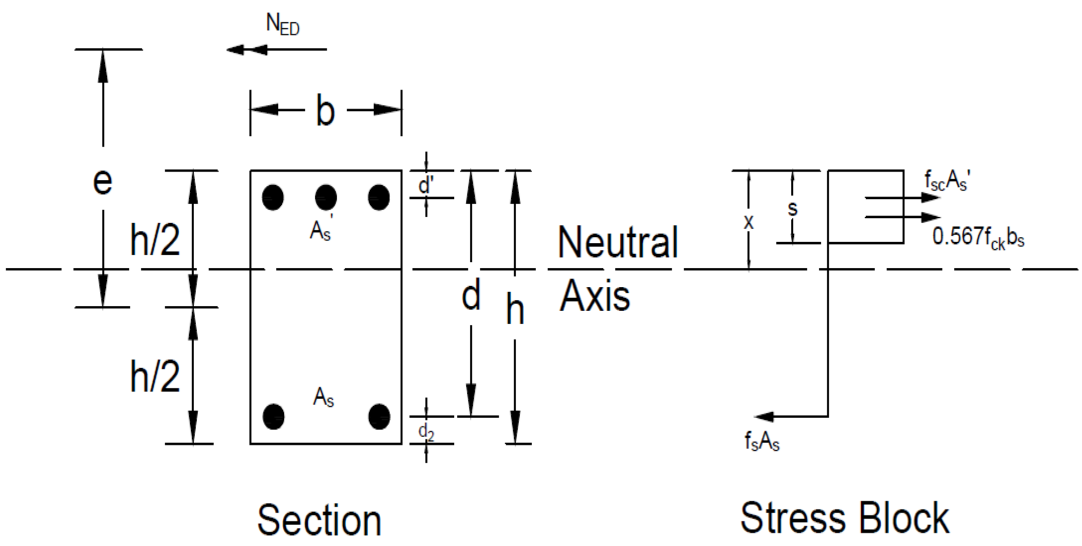


Figure 19: Stress block for unsymmetrical reinforcement

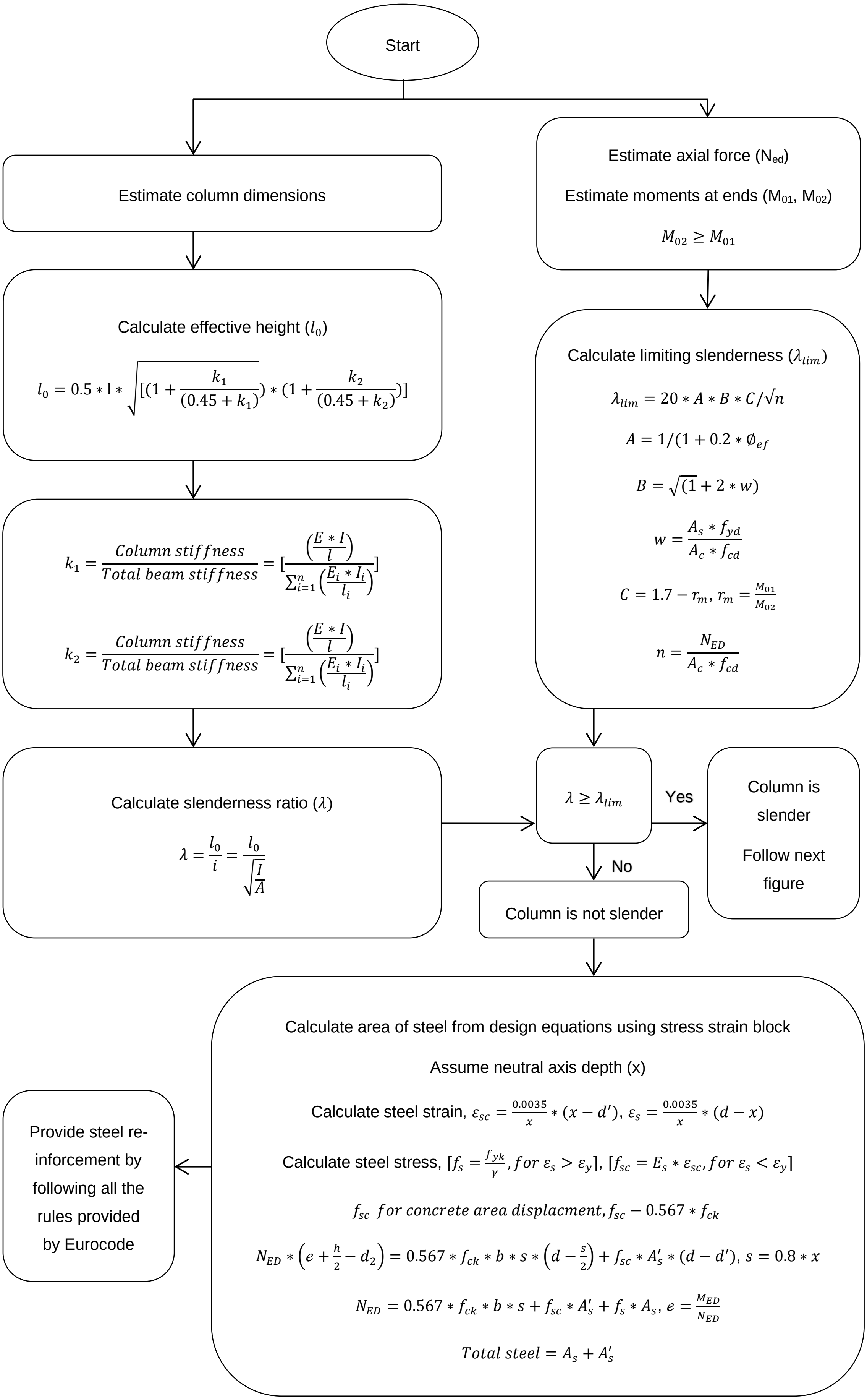


Figure 20: Braced non-slender/short column design

$\lambda \geq \lambda_{lim}$

Yes

Use nominal curvature method,

Consider second order effect

$$M_t = N_{ED} * e_{total}$$

$$e_{total} = e_0 + e_a + e_2$$

$$e_0 = \max\left(0.6 * e_{02} + 0.4 * e_{01}, 0.4 * e_{02}\right)$$

$$e_a = \mathcal{V} * \frac{l_0}{2}, \mathcal{V} = \frac{1}{100 * \sqrt{l}} > \frac{1}{200}$$

$$e_2 = k_1 * k_2 * \left(\frac{l_0^2}{\pi^2}\right) * \left(\frac{\varepsilon_{yd}}{0.45 * d}\right)$$

$$k_1 = 1 + \left(0.35 + \frac{f_{ck}}{200} - \frac{\lambda}{150}\right) * \phi_{ef} \geq 1$$

$$\phi_{ef} = \phi(\infty, t_0) * \frac{M_{0Eqp}}{M_{0ED}}$$

$$k_2 = \frac{N_{ud} - N_{ED}}{N_{ud} - N_{bal}} \leq 1.0$$

$$N_{ud} = 0.567 * f_{ck} * A_c + 0.87 * f_{yk} * A_{sc}$$

$$N_{bal} = 0.29 * f_{ck} * A_c$$

Calculate total steel using (M_t, N_{ED})

Follow steps in brace short column

Figure 21: Braced slender column Design

$$l_0 = l\sqrt{\left(1 + 10 * \frac{k_1 * k_2}{k_1 + k_2}\right)}$$

$$l_0 = l * \left(1 + \frac{k_1}{1 + k_1}\right) * \left(1 + \frac{k_2}{1 + k_2}\right)$$

$$l_0 = maximum\left[l\sqrt{\left(1 + 10 * \frac{k_1 * k_2}{k_1 + k_2}\right)}, \quad l * \left(1 + \frac{k_1}{1 + k_1}\right) * \left(1 + \frac{k_2}{1 + k_2}\right)\right]$$

$$\lambda_{lim} = \frac{10.8}{\sqrt{\left(\frac{N_{ED}}{A_c * f_{cd}}\right)}}$$

Follow design procedure given in figure 21 and 22

Figure 22: Unbraced column design (EN-1992-1-1 (CEN), 2004)

The design of the column is illustrated with manual calculations and MATLAB programming. The example of a column taken for problem formulation in MATLAB is shown in the diagram below:

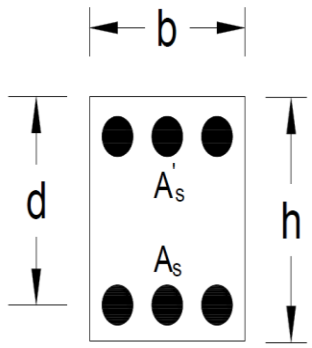


Figure 23: Column section

<u>Given Data:</u>

Height of the column = 3 m

Live Load (N) = 500 kN

Density of concrete = 25 kN/m^3

<u>Variable data:</u>

Breadth of the column (b) = (200 – 700) mm with step size of 25 mm

Depth of the column (h) = (200 – 700) mm with step size of 25 mm

Characteristic strength of concrete (f_{ck}) = (20, 25, 30, 35, 40, 45, 50) N/mm^2

Characteristic strength of steel (fyk) = (500, 550, 600) N/mm^2

Diameter for bottom reinforcement ($\emptyset_{sb}$) = (12, 16, 20, 25, 28, 32) mm

Number of bars (n_b) = (2,3,4,5)

Diameter for top reinforcement ($\emptyset_{st}$) = (12, 16, 20, 25, 28, 32) mm

Number of bars (n_t) = (2,3,4,5)

Diameter of shear reinforcement ($\emptyset_{s}$) = (6, 8, 10, 12, 14) mm

Diameter for prov. bottom reinforcement ($\emptyset_{sb,prov}$) = (12, 16, 20, 25, 28, 32) mm

Diameter for provided top reinforcement ($\emptyset_{sb,prov}$) = (12, 16, 20, 25, 28, 32) mm

Spacing (s) = (75 – 800) mm with step size of 5 mm

Depth of neutral axis= 150-250 mm with step size of 5 mm

Objective function:

Cost objective function f(x) = [(V_c*C_c) +(V_s*C_s) +(V_f*C_f)]

Embodied carbon objective function h(x) = [(V_c*E_c) +(V_s*E_s) +(V_f*E_f)]

Where, V_c, V_s, V_f is the volume of concrete, steel and formwork respectively.

C_c, C_s, C_f is the cost of concrete, steel and formwork respectively.

E_c, E_s, E_f, embodied carbon emission for concrete, steel and formwork

Constraints:

Constraint function, g(x) = (X_1, X_2, X_3, X_4, X_5, X_6)

g(X_1) for eccentricity, $20 - e \leq 0$

g(X_2) for top reinforcement, $\left[\frac{N_{ED}\left(e+\frac{h}{2}-d_2\right)-0.567*f_{ck}*b*s\left(d-\frac{s}{2}\right)}{f_{sc}*(d-d')}\right] - \left[n_t * \pi * \frac{{\emptyset_{st}}^2}{4}\right] \leq 0$

g(X_3) for bottom reinforcement,

$$\left\{\frac{\{N_{ED}-(0.567*f_{ck}*b*s)-\left(f_{sc}*\left[\frac{N_{ED}\left(e+\frac{h}{2}-d_2\right)-0.567*f_{ck}*b*s\left(d-\frac{s}{2}\right)}{f_{sc}*(d-d')}\right]\right)}{f_s}\right\} - \left[n_b * \pi * \frac{{\emptyset_{sb}}^2}{4}\right] \leq 0$$

g(X_4) for minimum reinforcement, $(0.002 * A_C) - \left(\frac{0.10*N_{ED}}{0.87*f_{yk}}\right) \leq 0$

g(X_5) for maximum reinforcement, $\left[\frac{Area\ of\ steel\ (A_{s,total})}{Area\ of\ Concrete\ (A_C)}\right] - 0.04 \leq 0$

g(X_6) for minimum shear reinforcement diameter, $[6 - \emptyset_s] \leq 0$

The design is formulated with all the given design data and the constraints according to Eurocode in MATLAB. The best solution for cost and embodied carbon emission is analysed separately using single-objective GA optimization with elitism. The elite count of individuals that survive to the next generation is given by the formula:

$$EC = 0.05 * maximum\ [minimum((10 * number\ of\ variabes, 100), 40)]$$

Due to the availability of nonlinear constraint, the population is taken as a double vector and the initial population is created using constraint dependent creation function which automatically selects the starting population best suited for the constraints provided. The fitness of the population is sorted using the rank scaling where all the individuals are given a rank based on their performance for the objective function. The best-ranked individual is one and next best with increasing orders. The scaled individuals are then chosen for next-generation using the stochastic uniform method. The mutation and crossover function are constraints dependent as well. The results of the optimization process can be seen graphically in the pictures below.

The augmented Lagrangian penalty function is also implemented if the constraints are not satisfied. The initial penalty used is 10 with a penalty factor of 100. The number of iterations performed by the GA is 2000 and can be also given by the formula:

$$Generation = 100 * Number\ of\ variables$$

The stopping criteria are set by the average change in values of each iteration. The algorithm stops if the function tolerance value is less than 10^-6 and the constraint tolerance is less than 10^-3. The number of iterations taken to achieve the results is 91with a function count of 9201 in the case of cost optimization.

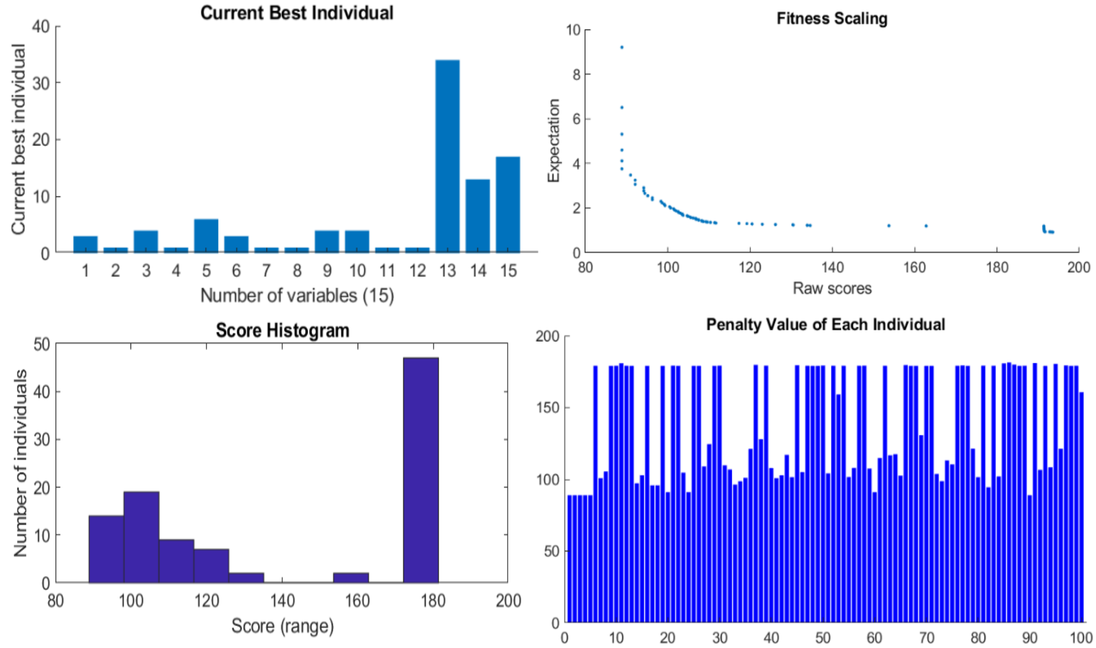


Figure 24: MATLAB cost optimization graphs (Mathworks, 1984)

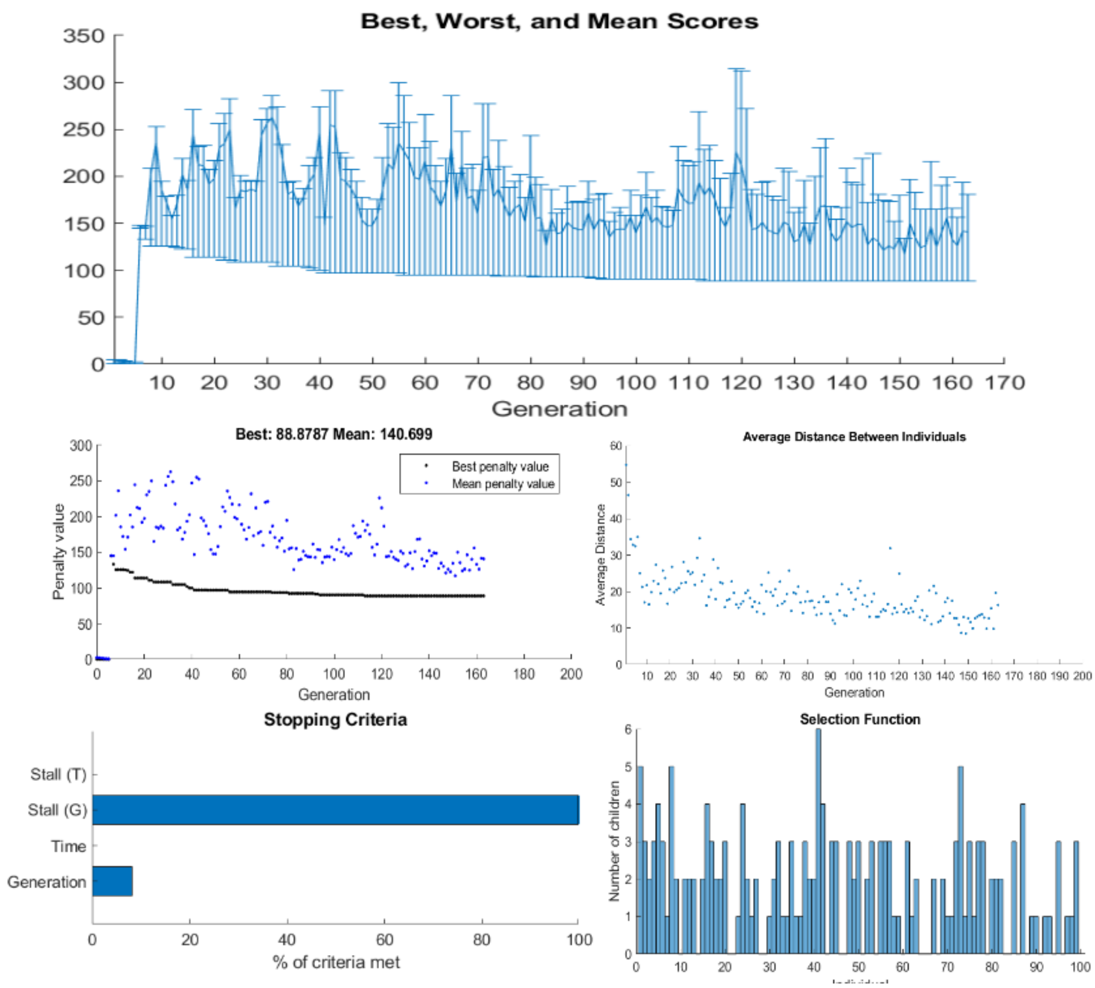


Figure 25: MATLAB cost optimization graphs (Mathworks, 1984)

Description	Manual	GA Optimization (Cost)
Breadth (mm)	400	250
Depth (mm)	400	200
Top Steel Reinforcement (mm)	#3 of 25 mm	#2 of 12 mm
Bottom Steel Reinforcement (mm)	#4 of 25 mm	#2 of 12 mm
Spacing (mm)	6 links at 500	12 links at 240
Shear Reinforcement	2 legged 8 mm	2 legged 12 mm
Total Area (mm^2)	4037.255	3278.16
Cost (Euros)	149	89
Carbon (kg-CO_2)	244	116

Table 6: MATLAB cost optimization result (Mathworks, 1984)

The MATLAB results for embodied carbon fitness function using single-objective GA are shown below. The optimization is performed with a higher number of initial populations which is given by the formula:

$$Population\ size = \max(\min(10*\text{number of variables}, 100), 40)$$

The number of iterations taken to achieve the carbon optimized result is 125 with a function count of 12601. A total of 15 variables was used in the optimization and all the constraints were satisfied which are mentioned in the design procedure using Eurocodes.

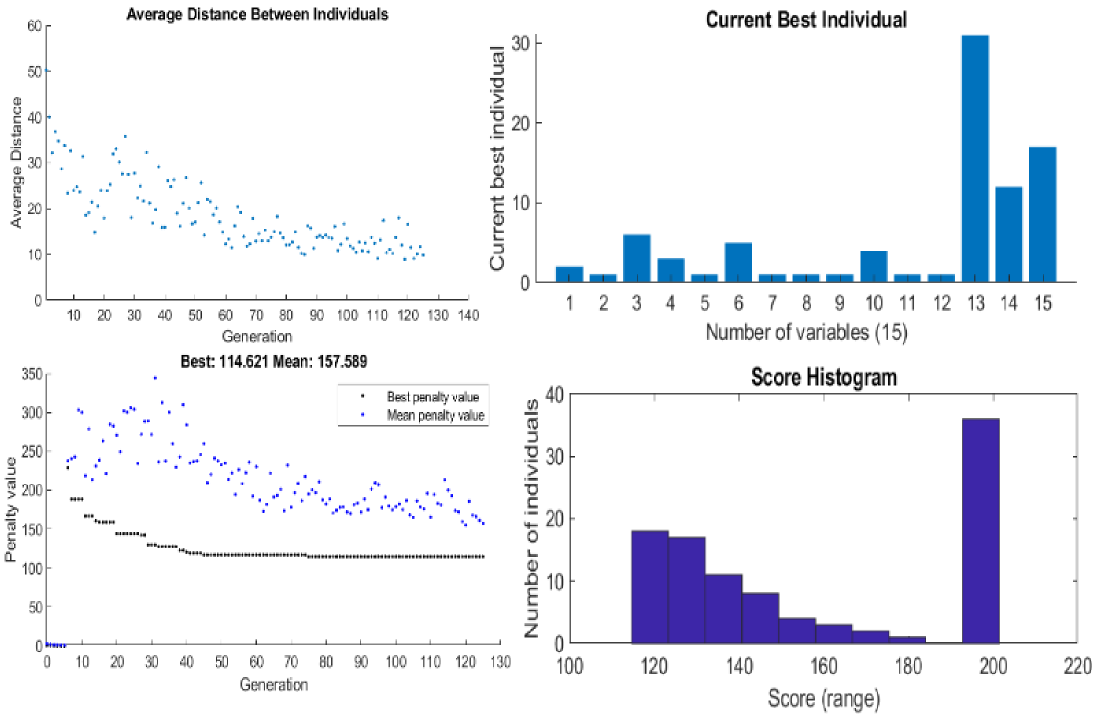


Figure 26: MATLAB carbon optimization graphs (Mathworks, 1984)

Description	Manual	Cost Optimized	Carbon Optimized
Concrete Strength, f_{ck} (N/mm^2)	30	40	50
Steel Strength, f_{yk} (N/mm^2)	500	500	600
Depth of neutral axis (mm)	190	210	205
Eccentricity (mm)	20	22	22

Table 7: Additional variable data

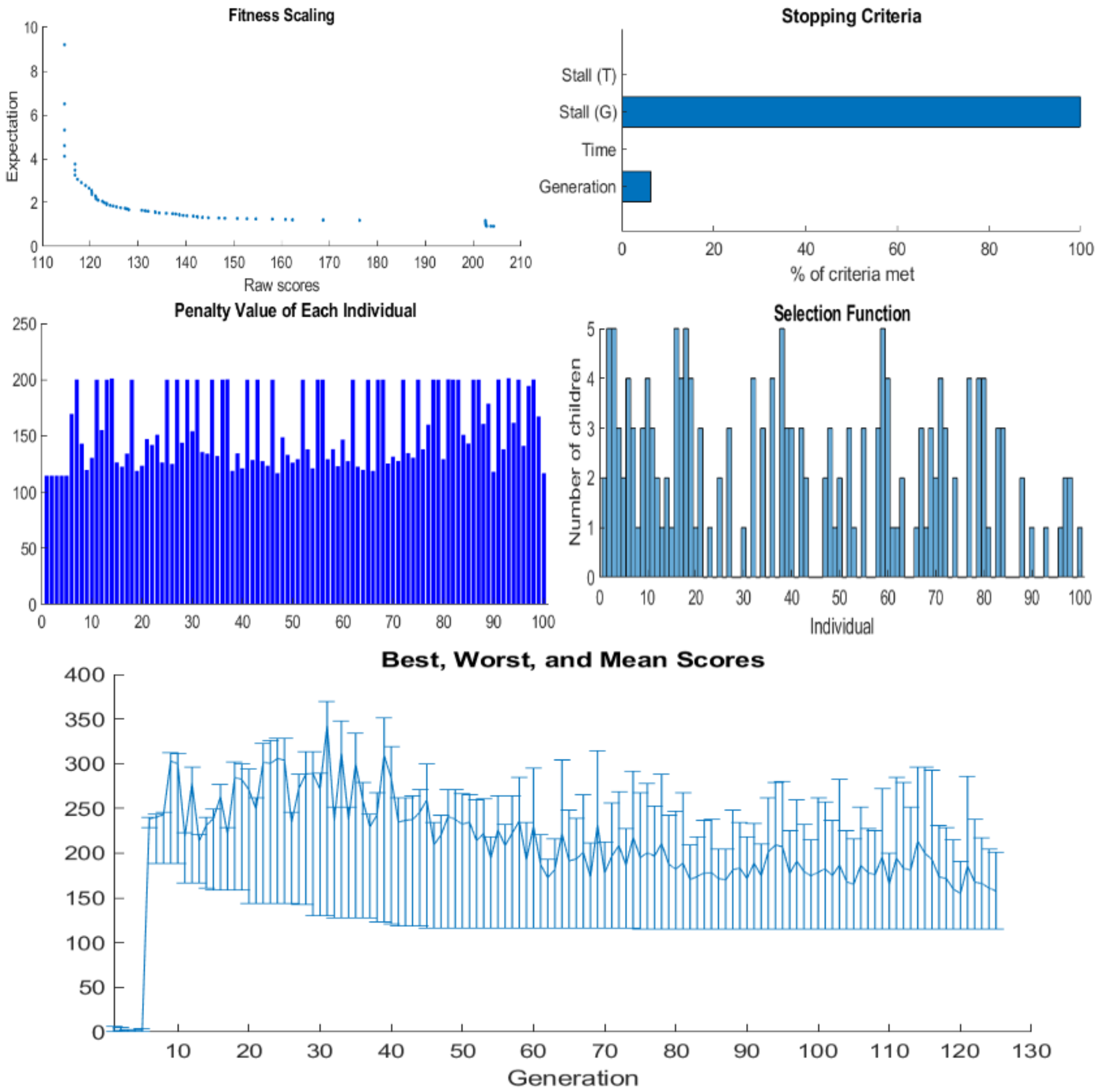


Figure 27: MATLAB carbon optimization graphs (Mathworks, 1984)

Description	Manual	GA Optimization (Carbon)
Breadth (mm)	400	225
Depth (mm)	400	200
Top Steel Reinforcement (mm)	#3 of 25 mm	#2 of 12 mm
Bottom Steel Reinforcement (mm)	#4 of 25 mm	#2 of 12 mm
Spacing (mm)	6 links at 500	13 links at 225
Shear Reinforcement	2 legged 8 mm	2 legged 12 mm
Total Area (mm^2)	4037.255	3459.024
Cost (Euros)	149	90
Carbon (kg-CO_2)	244	115

Table 8: MATLAB carbon optimized result

5.3 Slab Design

The RCC slab is a horizontal structural member which serves as the first point of contact for loads coming on the structure and transmits it to the beams, walls or column depending on the type of structural system. The slabs can be used for floors, roofs, bridge decks and RCC walls. There are different types of slabs used in the construction industry but two broad classifications depending on the load transfer directions are as follows:

1) One-way slab: The ratio of the length of the longer edge to the shorter edge is more than 2 and the bending and deflection caused by the loading is in one direction. The one-way slabs are mostly supported on two edges and the loads are carried in the perpendicular direction to the supports. Therefore, the main reinforcement is always provided in the deflected direction i.e., shorter span and the distribution reinforcement is in the direction of larger span.
2) Two-way slab: The ratio of the length of the longer edge to the shorter edge is less than 2 and the deflection caused by the loading is in two directions. The two-way slabs are supported on the four sides and the loads are carried in both directions. Therefore, the main reinforcement is provided in both directions.

The slabs are further divided into various types depending on their purpose of usage. These slabs can be identified as follows:

1) Conventional/Simply supported solid slabs: These slabs can be continuous or discontinuous on the edges and are supported by beams. The beam depth is larger compared to the thickness of the slab. These can be one way or two ways slabs. The load transfer takes place from the slab to the beams and then to the columns.
2) Continuous slabs: These types of slabs are continuous over multiple supports, and the design moments are shared between the slabs.
3) Flat slab: These slabs are supported directly by the columns in a structural system that excludes beams. There are many types of flat slabs i.e. flat slab with a column head, flat slab with drop panels and flat slab with both column head and drop panel.

4) Ribbed slab: It is a one-way slab system that consists of joists/ribs at equal spacing making the slab look like T beams resting on the beam girders and transferring the load to the columns. The rib section of the slab is re-inforced, and it acts as a small beam.
5) Waffle slab: Waffle slab is a grid-like slab with horizontal and vertical gaps between the pods where the reinforcement is provided during the form-work. The appearance of the slab from inside the building looks like a waf-fle when pods are removed. Because of the availability of the pods the concrete usage is very less. These slabs can resist heavy loading com-pared to the conventional slabs and are used where the spans are bigger i.e., in cinema halls, auditoriums or big hotels. Waffle slabs are two-way slabs.

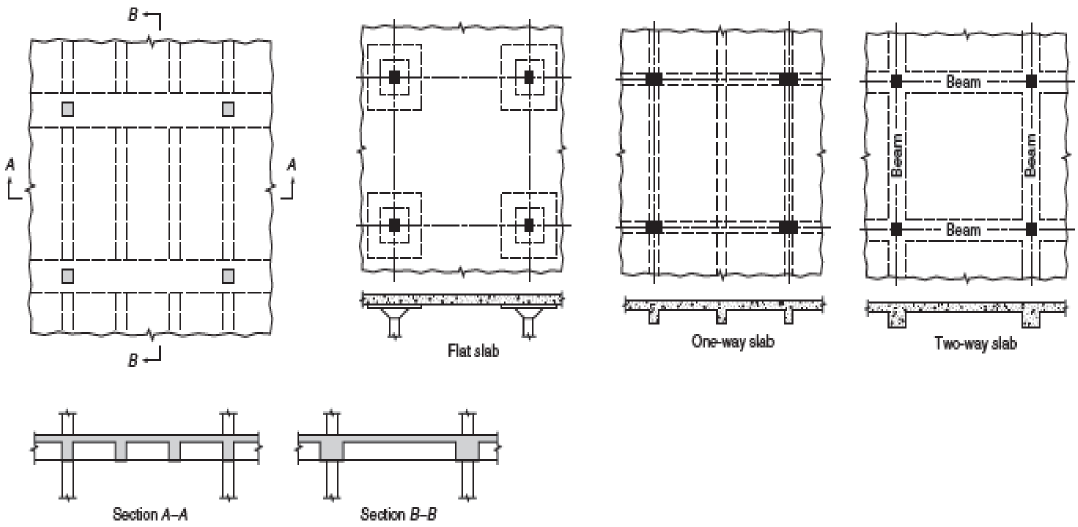


Figure 28: Types of slabs (Darwin, et al., 2016)

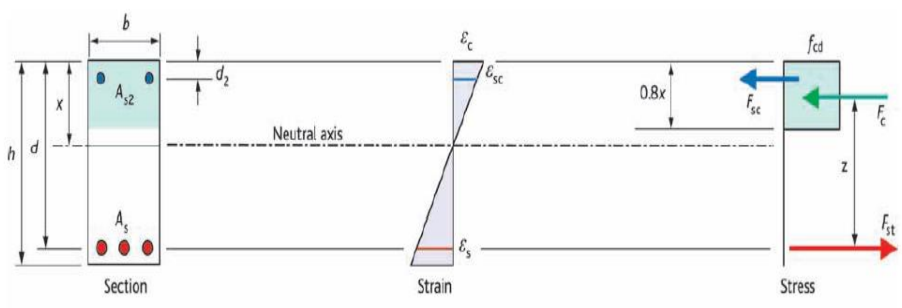


Figure 29: Rectangular stress block (www.concretecenter.com)

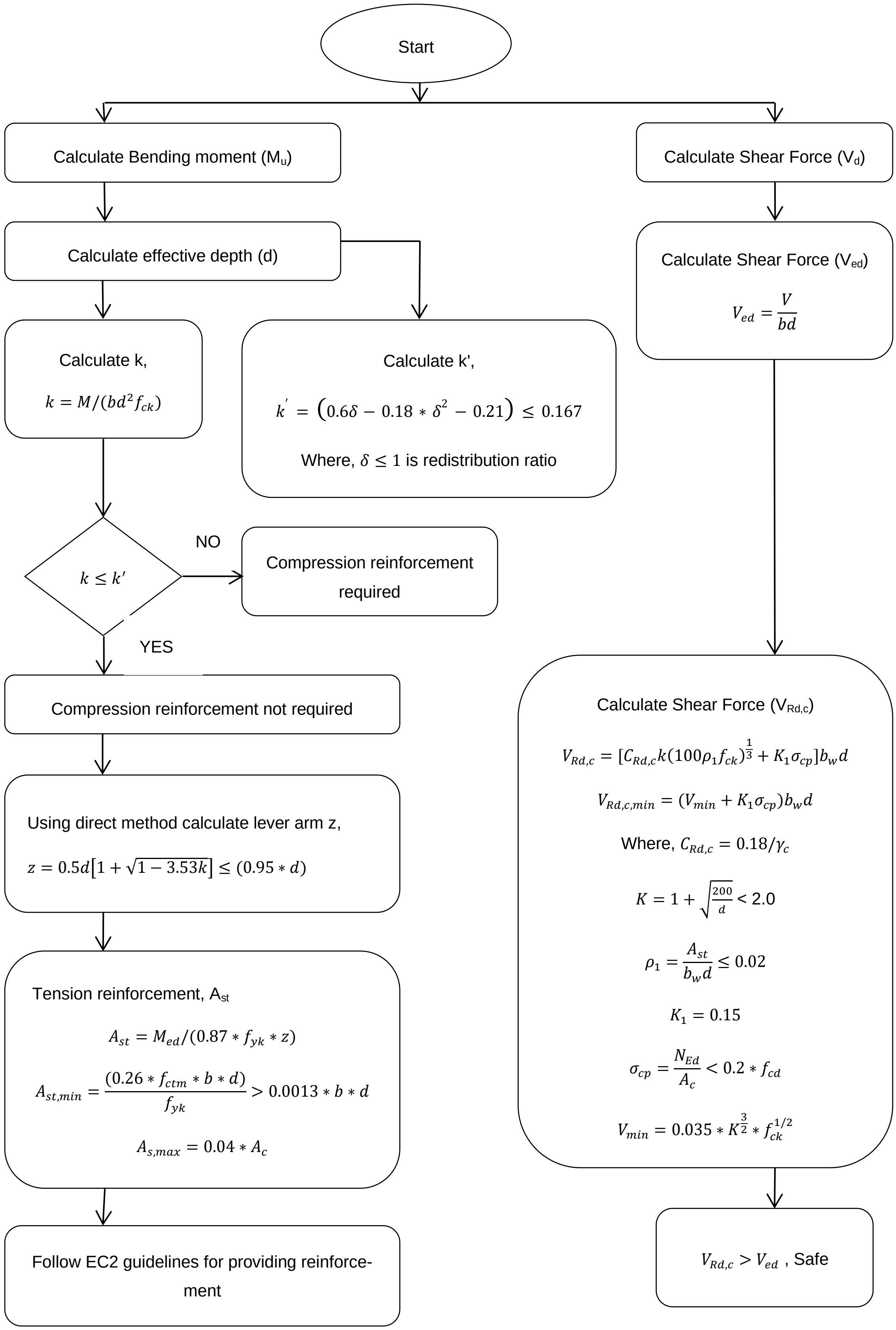


Figure 30: Slab flexure and shear design (European Commission, 2004)

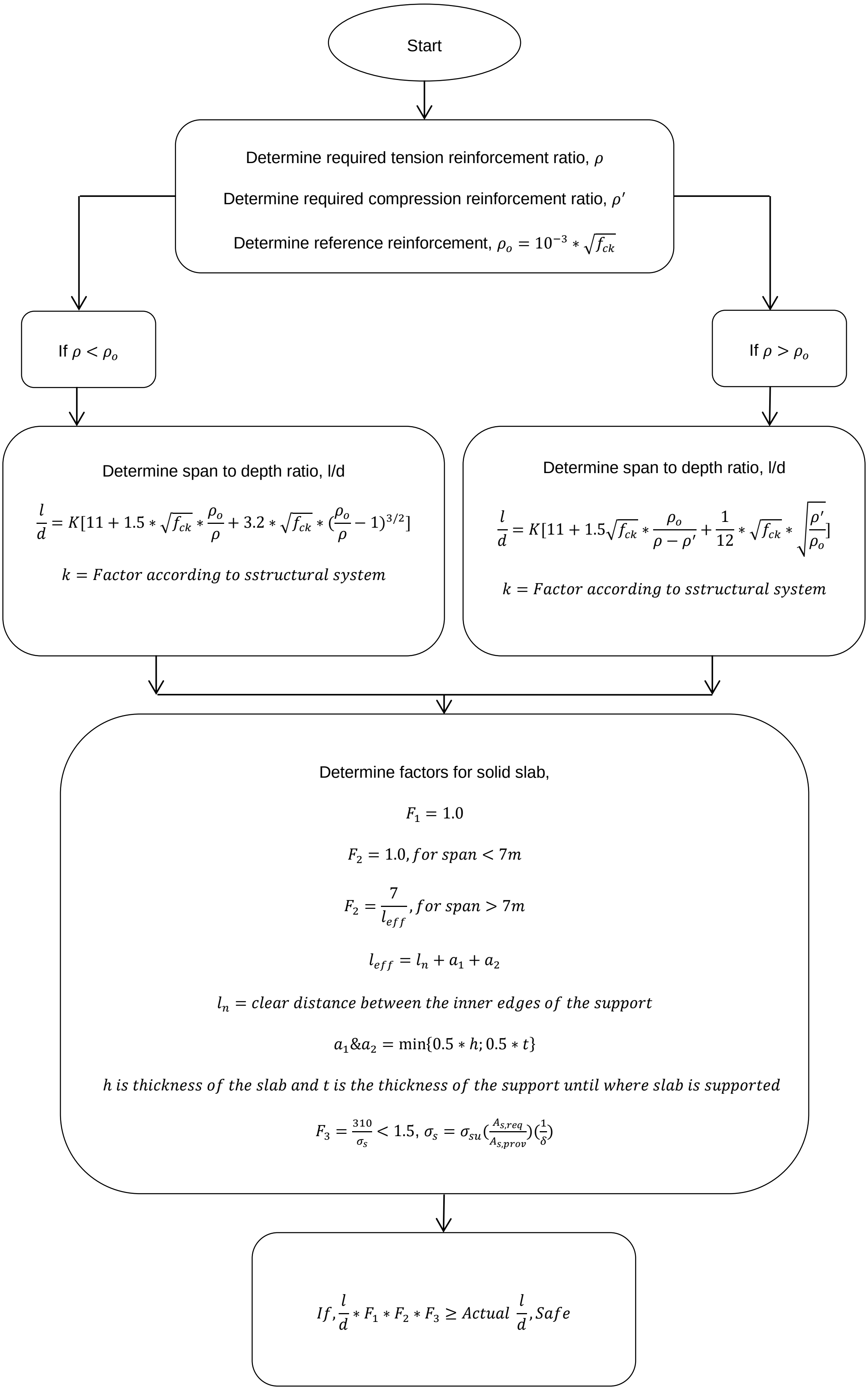


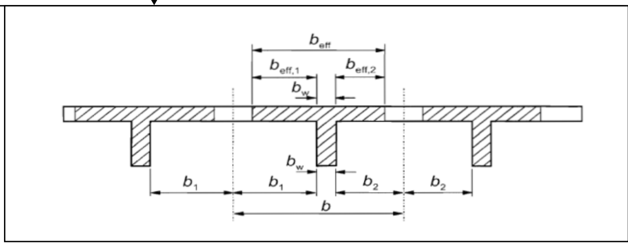
Figure 31: Deflection check for slab (EN-1992-1-1 (CEN), 2004)

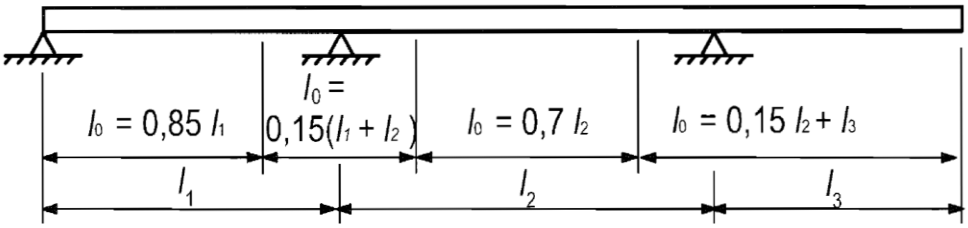
Start

Calculate effective breadth, b_{eff}

$$b_{eff} = b_w + \sum b_{eff,i}$$

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





Calculate, $k = \frac{M}{bd^2 f_{ck}}$

$$k' = (0.6\delta - 0.18 * \delta^2 - 0.21) \leq 0.167$$

Where, $\delta \leq 1$ is redistribution ratio

Using direct method, calculate lever arm

$$z = 0.5d\left[1 + \sqrt{1 - 3.53k}\right] \leq (0.95 * d)$$

Determine neutral axis depth

$$X = 2.5 * (d - z)$$

If, $X \leq 1.25 * h_f$

YES

Neutral axis depth lies in flange of slab. Therefore, design as rectangular section

NO

Neutral axis lies in web of slab section

Determine resistance moment capacity

$$M_{R,f} = 0.57 f_{ck}(b_{eff} - b_w)h_f(d - 0.5h_f)$$

$$K_f = \frac{M - M_{R,f}}{f_{ck} b_w d^2}$$

$K_f \leq K'$

NO

Redesign the slab

YES

Determine required reinforcement

$$A_{s,req} = \frac{M_{R,f}}{0.87 f_{yk}(d - 0.5h_f)} + \frac{M - M_{R,f}}{0.87 f_{yk} z}$$

$$A_{s,min} = \frac{0.26 f_{ctm} b_t d}{f_{yk}}$$

$$A_{s,min} \geq 0.0013 b_t d$$

$$A_{s,req} \leq 0.04 A_c$$

Figure 32: Ribbed slab design (EN-1992-1-1 (CEN), 2004)

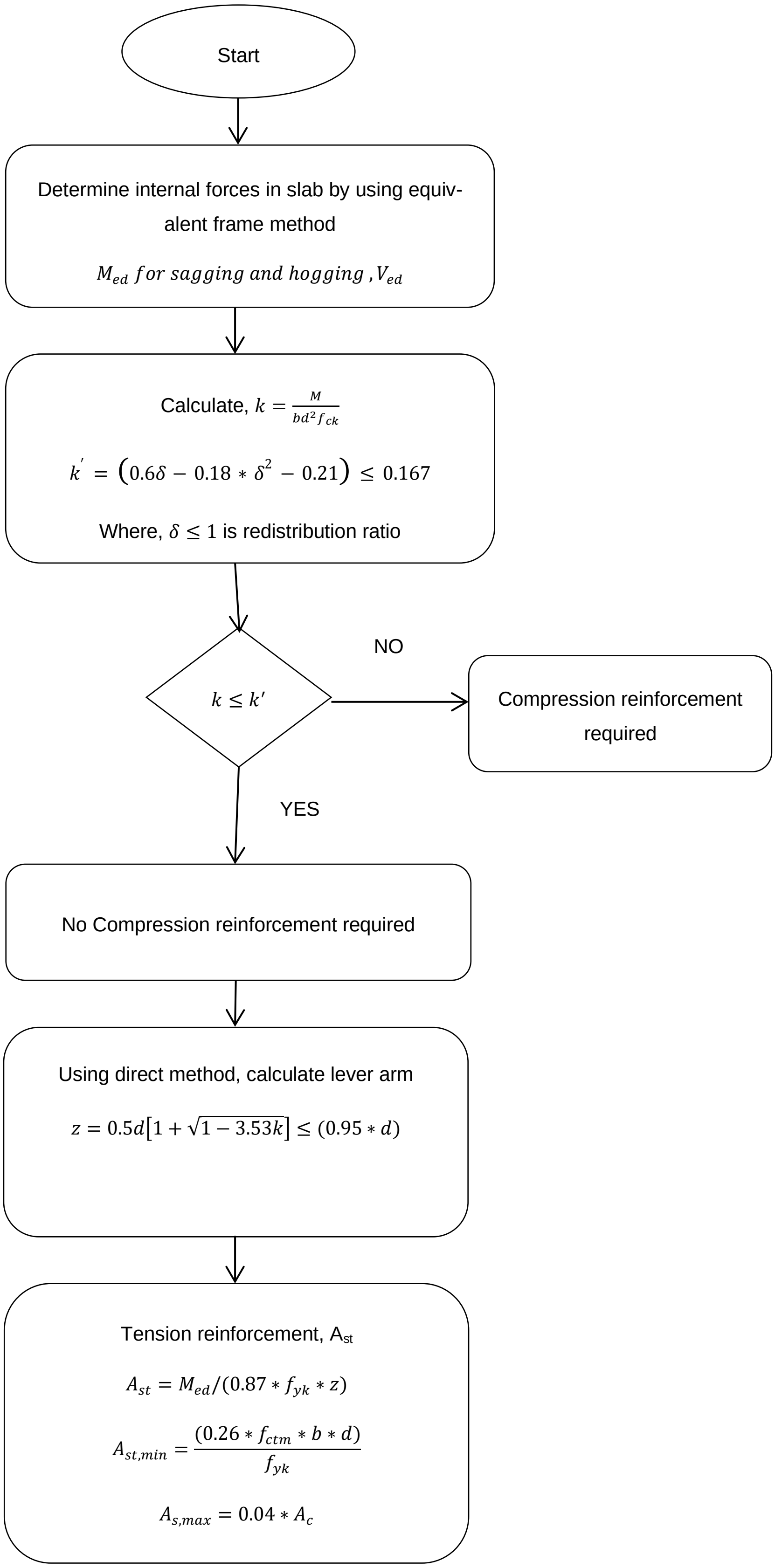


Figure 33: Flat slab design (EN-1992-1-1 (CEN), 2004)

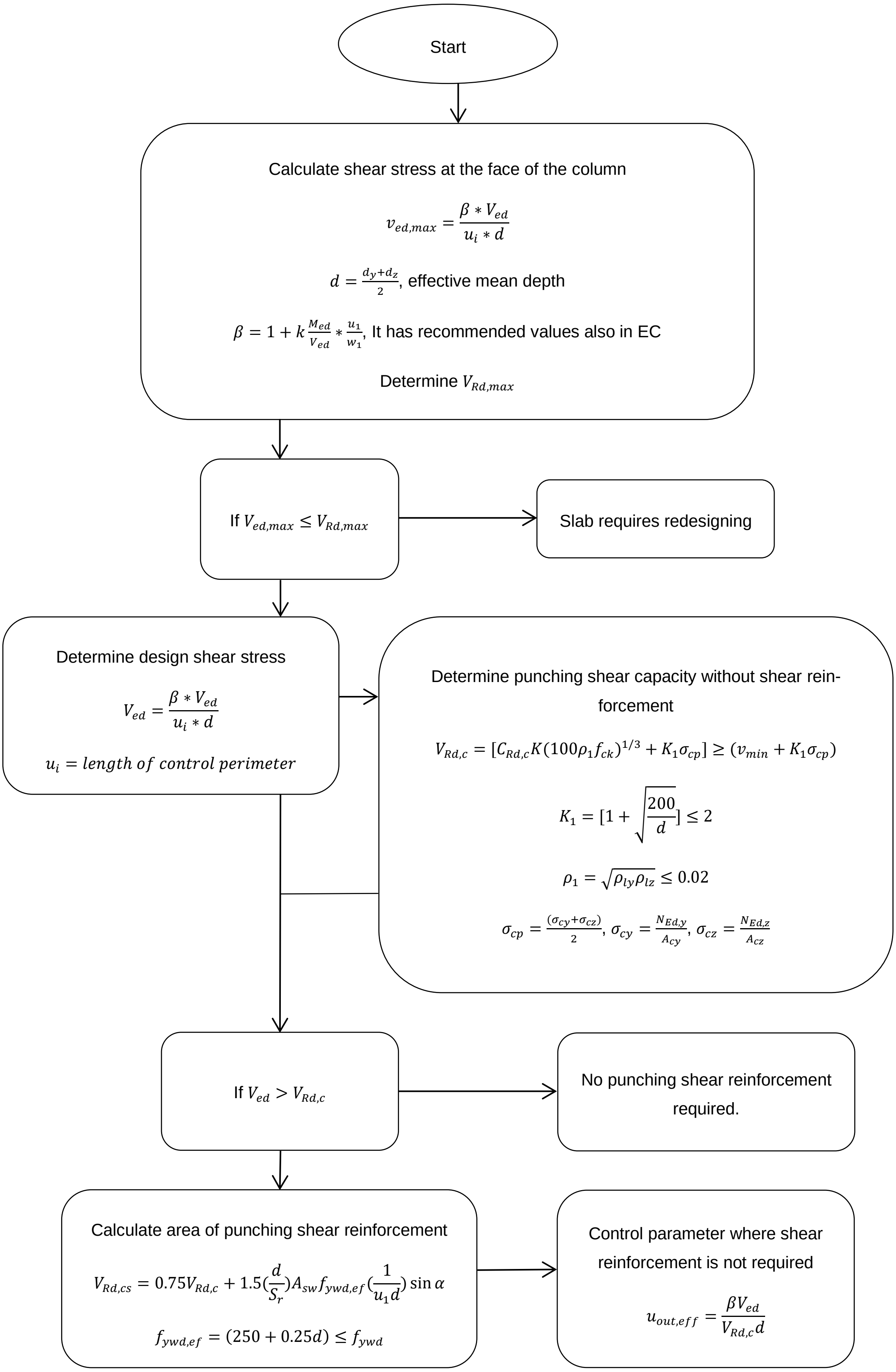


Figure 34: Punching shear design (EN-1992-1-1 (CEN), 2004)

The design for different slabs is implemented in MATLAB for four different slabs i.e., one way slab, two-way slab, ribbed slab and flat slab. The problem formulation for each slab can be seen below separately:

5.3.1 One Way Slab

Given data:

Length of the slab in x-direction = 5 m

Length of the slab in y-direction = 10 m

Density of concrete = 25 kN/m^3

Live load (q_k) = 3 kN/m^2

Variable data: x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)

Depth of the slab (h), x_1 = (125 - 350) mm with step size of 25 mm.

Characteristic strength of concrete (f_{ck}), x_2 = (20, 25, 30, 35, 40, 45, 50) N/mm^2

Characteristic strength of steel (f_{yk}), x_3 = (400,420,450,500) N/mm^2

Diameter for bottom reinforcement ($\emptyset_{sb}$), x_4 = (10,12, 16, 20, 25, 28, 32) mm

Diameter for transverse reinforcement, x_5 = (8,10,12, 16, 20, 25, 28, 32) mm

Number of bars (n_b), x_6 = (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)

Number of bars (n_t), x_7 = (2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)

Objective function:

Cost objective function f(x) = [(V_c*C_c) +(V_s*C_s) +(V_f*C_f)]

Embodied carbon objective function h(x) = [(V_c*E_c) +(V_s*E_s) +(V_f*E_f)]

Where, V_c, V_s, V_f is the volume of concrete, steel and formwork respectively.

C_c, C_s, C_f is the cost of concrete, steel and formwork respectively.

E_c, E_s, E_f, embodied carbon emission for concrete, steel and formwork

Constraints:

Constraint function, g(x) = (X_1, X_2, X_3, X_4, X_5, X_6)

g(x_1) for slab thickness for fire safety, $(h_s - x(1)) < 0$

g(x_2) for minimum cover, $(20 - axisdistance) < 0$

g(x_3), $(\frac{M}{bd^2 f_{ck}} - 0.168) < 0$

g(x_4) for reinforcement, $(\frac{M}{0.87*x(3)*z} - \frac{x(6)*\pi*x(4)^2}{4}) < 0$

g(x_5) for minimum reinforcement, $(\frac{0.26*0.3*f_{ck}^{\frac{2}{3}}*b*d}{f_{yk}} - \frac{M}{0.87*x(3)*z}) < 0$

g(x_6) for maximum reinforcement,

$$\left(\frac{x(6)*\pi*x(4)^2}{4} - 0.04*\left(breadth*x(1) - \frac{x(6)*\pi*x(4)^2}{4}\right)\right) < 0$$

g(x_7), $(\rho - 0.02) < 0$

g(x_8) shear check, $(V_{ed} - \left(0.12 * k(100\rho_1 f_{ck})^{\frac{1}{3}}\right) bd) < 0)$

g(x_9) $1 < f_3 < 1.5$

g(x_{10}) deflection check, $\left(actual\ \frac{l}{d} - \left(\frac{l}{d} * f_1 * f_2 * f_3\right)\right) < 0$

g(x_{11}) for transverse steel, $\left(0.02 * \frac{0.26*0.3*f_{ck}^{\frac{2}{3}}*b*d}{f_{yk}} - \frac{x(7)*\pi*x(5)*x(5)}{4}\right) < 0$

g(x_{12}), $(K_1 - 2) < 0$

g(x_{13}) for effective depth, $(\frac{span}{20*modification\ factor} - (x(1) - cover - \frac{\emptyset}{2})) < 0$

g(x_{14}) for main reinforcement spacing, $(maxspacingmain - maxspacing) < 0$

g(x_{15}) for transverse steel, $(maxspacingtrannverse - maxspacing) < 0$

The basic settings for a genetic algorithm such as population size, fitness scaling, selection, reproduction, elitism, mutation, crossover, the penalty function is taken through the same formulas as for column and beam. The number of iterations taken for cost optimization was 107 and carbon optimization was 77. The cost and carbon optimized solutions have the same variables which give the best solution for both objectives.

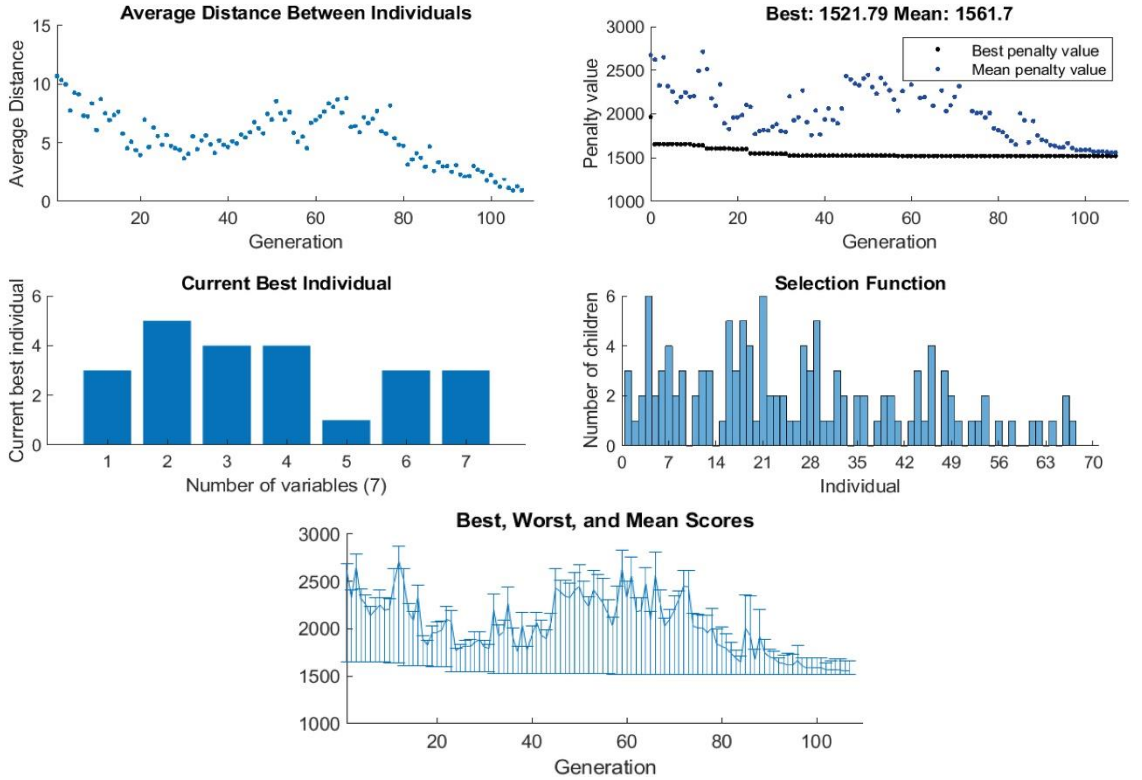


Figure 35:MATLAB cost optimization result (Mathworks, 1984)

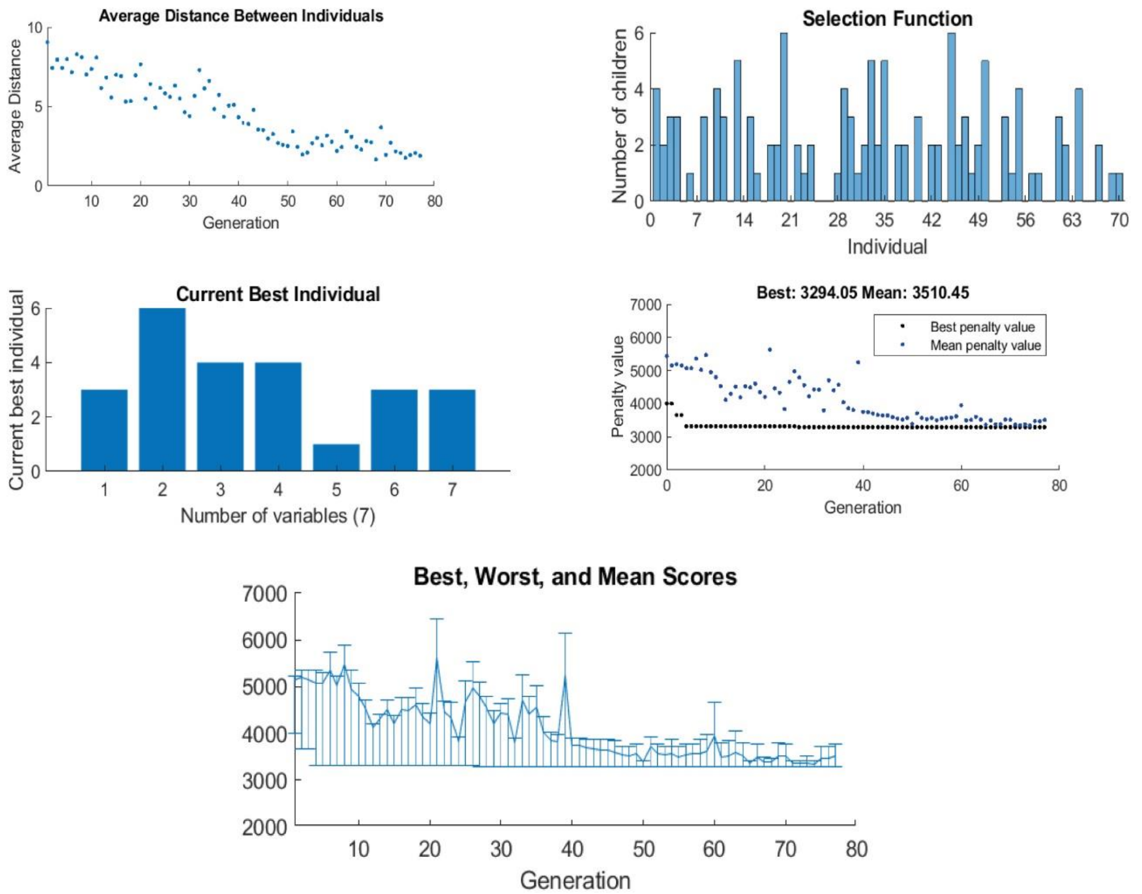


Figure 36: MATLAB carbon optimization result (Mathworks, 1984)

Variables	Manual Calculations	Cost Optimization	Carbon Optimization
Depth (mm)	200	175	175
Concrete strength (MPA)	30	45	45
Steel strength (MPA)	500	500	500
Bottom steel (mm)	12	16	16
Transverse steel (mm)	10	10	10
Number of bottom bars	5	3	3
Number of transverse bars	3	3	3
Cost (Euros)	1649	1522	1522
Carbon (kg-CO_2)	3725	3294	3294

Table 9: Result comparison for one way slab

5.3.2 Two Way Slab

The two-way slab is designed as per the Eurocodes and the design procedure is like that of a one-way slab. The moments in different direction used factors and the data which is other than the one used in one way slab are given below.

Given data:

Length of the slab in x-direction = 5 m

Length of the slab in y-direction = 8 m

Density of concrete = 25 kN/m^3

Live load (q_k) = 10 kN/m^2

Additional formulas:

Moment is x-direction, $M_{sx} = \alpha_{sx} * n * {l_x}^2$

Moment in the y-direction, $M_{sy} = \alpha_{sy} * n * {l_x}^2$

Additional variables:

Diameter for y-direction reinforcement, x_5 = (8,10,12, 16, 20, 25, 28, 32) mm

Number of bars (n_y), x_6 = (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)

The basic settings for a genetic algorithm such as population size, fitness scaling, selection, reproduction, elitism, mutation, crossover, the penalty function is taken through the same formulas as for column and beam. The number of iterations taken for cost optimization was 115 and carbon optimization was 93. The cost and carbon optimized solutions have the same variables which give the best solution for both objectives.

Variables	Manual Calculations	Cost Optimization	Carbon Optimization
Depth (mm)	200	200	200
Concrete strength (MPA)	30	50	50
Steel strength (MPA)	500	500	500
X-direction steel dia. (mm)	16	12	12
Y-direction steel dia. (mm)	16	10	10
Number of x-direction bars	7	8	8
Number of y-direction bars	2	5	5
Cost (Euros)	1760	1441	1441
Carbon (kg-CO_2)	3258	3099	3099

Table 10: Result comparison for two-way slab

5.3.3 Ribbed Slab

<u>Given data:</u>

Length of the slab in x-direction = 5 m

Length of the slab in y-direction = 7 m

Density of concrete = 25 kN/m^3

Live load (q_k) = 2.5 kN/m^2

<u>Variable data:</u> x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11})

Depth of the slab (h), x_1 = (125 - 400) mm with step size of 25 mm.

Characteristic strength of concrete (f_{ck}), x_2 = (20, 25, 30, 35, 40, 45, 50) N/mm^2

Characteristic strength of steel (f_{yk}), x_3 = (500,550,600) N/mm^2

Diameter for bottom reinforcement ($\emptyset_{sb}$), x_4 = (12, 14, 16, 20, 25, 28, 32) mm

Diameter for transverse reinforcement, x_5 = (12, 14, 16, 20, 25, 28, 32) mm

Number of bars (n_b), x_6 = (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)

Number of bars (n_t), x_7 = (2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)

Rib breadth, x_8 = (100, 125, 150, 175, 200, 225, 250)

Clear rib distance, x_9 = (300 - 1500) with step size of 50mm

Flange depth, x_{10} = (50, 60, 70, 80, 90, 100)

Mesh area, x_{11} = (98, 142, 193, 252, 393)

<u>Constraints:</u>

Constraint function, g(x) = (X_1, X_2, X_3, X_4, X_5, X_6)

g(x_1) for slab thickness for fire safety, $(h_s - x(1)) < 0$

g(x_2) for minimum cover, $(25 - axisdistance) < 0$

g(x_3), $(\frac{M}{bd^2 f_{ck}} - 0.168) < 0$

g(x_4) for reinforcement, $(\frac{M}{0.87*x(3)*z} - \frac{x(6)*\pi*x(4)^2}{4}) < 0$

g(x_5) for minimum reinforcement, $(\frac{0.26*0.3*f_{ck}^{\frac{2}{3}}*b*d}{f_{yk}} - \frac{M}{0.87*x(3)*z}) < 0$

g(x_6) for maximum reinforcement,

$$\left(\frac{x(6)*\pi*x(4)^2}{4} - 0.04*\left(breadth*x(1) - \frac{x(6)*\pi*x(4)^2}{4}\right)\right) < 0$$

g(x_7), $(\rho - 0.02) < 0$

g(x_8) shear check, $(V_{ed} - \left(0.12*k(100\rho_1 f_{ck})^{\frac{1}{3}}\right)bd) < 0)$

g(x_9) $1 < f_3 < 1.5$

g(x_{10}) deflection check, $\left(actual\ \frac{l}{d} - \left(\frac{l}{d}*f_1*f_2*f_3\right)\right) < 0$

g(x_{11}) for mesh steel, $\left(\frac{0.13*1000*x(10)}{100} - x(11)\right) < 0$

g(x_{12}), $(K_1 - 2) < 0$

g(x_{13}) for effective depth, $(\frac{span}{20*modification\ factor} - (x(1) - cover - \frac{\emptyset}{2})) < 0$

g(x_{14}) for rib depth, $\big((x(1) - x(10)\big) - (4 * x(8)) < 0$

g(x_{15}) for neutral axis depth, $\big((2.5 * (effectivedepth - z)\big) - (1.25 * x(10)) < 0$

Data	Manual	Cost Optimized	carbon Optimized
Thickness (mm)	200	200	175
Concrete Strength (MPA)	30	45	50
Steel Strength (MPA)	500	550	600
Rib breadth (mm)	150	100	100
Clear distance (mm)	400	500	500
Flange depth (mm)	60	50	50
Rib depth (mm)	140	150	125
Cost (Euros)	752	618	630
Carbon (kg-CO_2)	1563	1120	1076

Table 11: Result comparison for ribbed slab

5.3.4 Flat Slab

<u>Given data:</u>

Length of the slab in x-direction = 6.5 m

Length of the slab in y-direction = 6.5 m

Density of concrete = 25 kN/m^3

Live load (q_k) = 5 kN/m^2

<u>Variable data:</u> x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14})

Depth of the slab (h), x_1 = (125 - 400) mm with step size of 25 mm.

Depth of the drop panel, x_2 = (85 - 150) mm with a step size of 5 mm.

Dimension of the drop panel, x_3 = (2200 - 3200) mm with a step size of 100 mm.

Dimension of column head, x_4 = (1000 - 1500) mm with a step size of 100 mm.

Characteristic strength of concrete (f_{ck}), x_5 = (20, 25, 30, 35, 40, 45, 50) N/mm^2.

Characteristic strength of steel (f_{yk}), x_6 = (500,550,600) N/mm^2.

Diameter for middle strip ($\emptyset_{ms}$), x_7 = (12, 14, 16, 20, 25, 28, 32) mm.

Number of middle strip bars (n_{ms}), x_8 = (1 - 20) with step size of 1.

Diameter for column strip ($\emptyset_{cs}$), x_9 = (12, 14, 16, 20, 25, 28, 32) mm.

Number of column strip bars (n_{cs}), x_{10} = (1 - 20) with step size of 1.

Dia. for middle strip at interior span ($\emptyset_{mi}$), x_{11} = (12, 14, 16, 20, 25, 28, 32) mm.

Number of middle strip bars at interior span (n_{mi}), x_{12} = (1 - 20) step size of 1.

Dia. for column strip at interior span ($\emptyset_{ci}$), x_{13} = (12, 14, 16, 20, 25, 28, 32) mm.

Number of column strip bars at interior span (n_{ci}), x_{14} = (1 - 20) step size of 1.

<u>Constraints:</u>

Constraint function, g(x) = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}, X_{19}, X_{20}, X_{21}, X_{22}, X_{23}, X_{24}, X_{25})

g(x_1) for slab thickness for fire safety, $(h_s - x(1)) < 0$

g(x_2) for minimum cover, $(15 - axisdistance) < 0$

g(x_3) for bay area, $(30 - (lx * ly) < 0$

g(x_4) for load ratio, $(1.25 - \left(\frac{g_k}{q_k}\right)) < 0$

g(x_5) for middle strip at centre, $(\frac{M_{cm}}{b_m {d_{span}}^2 f_{ck}} - 0.168) < 0$

g(x_6) for middle strip at centre, $(\frac{M_{cm}}{0.87*x(6)*z} - \frac{x(8)*\pi*x(7)^2}{4}) < 0$

g(x_7) for minimum reinforcement, $(\frac{0.26*0.3*{f_{ck}}^{\frac{2}{3}}*b_m*d_{span}}{f_{yk}} - \frac{M_{cm}}{0.87*x(6)*z}) < 0$

g(x_8) for maximum reinforcement, $\left(\frac{x(8)*\pi*x(7)^2}{4} - 0.04 * area\ of\ concrete\right) < 0$

g(x_9) for column strip at the centre, $(\frac{M_{cc}}{b_c {d_{support}}^2 f_{ck}} - 0.168) < 0$

g(x_{10}) for column strip at centre, $(\frac{M_{cc}}{0.87*x(6)*z} - \frac{x(10)*\pi*x(9)^2}{4}) < 0$

g(x_{11}) for minimum reinforcement, $(\frac{0.26*0.3*{f_{ck}}^{\frac{2}{3}}*b_c*d_{support}}{f_{yk}} - \frac{M_{cc}}{0.87*x(6)*z}) < 0$

g(x_{12}) for maximum reinforcement, $\left(\frac{x(10)*\pi*x(9)^2}{4} - 0.04 * area\ of\ concrete\right) < 0$

g(x_{13}) for middle strip at interior span, $(\frac{M_{im}}{b_m d_{span}{}^2 f_{ck}} - 0.168) < 0$

g(x_{14}) for middle strip at interior span, $(\frac{M_{im}}{0.87*x(6)*z} - \frac{x(12)*\pi*x(11)^2}{4}) < 0$

g(x_{15}) for minimum reinforcement, $(\frac{0.26*0.3*f_{ck}{}^{\frac{2}{3}}*b_m*d_{span}}{f_{yk}} - \frac{M_{cc}}{0.87*x(6)*z}) < 0$

g(x_{16}) for maximum reinforcement, $\left(\frac{x(12)*\pi*x(11)^2}{4} - 0.04 * area\ of\ concrete\right) < 0$

g(x_{17}) for column strip at interior span, $(\frac{M_{ic}}{b_c d_{support}{}^2 f_{ck}} - 0.168) < 0$

g(x_{18}) for column strip at interior span, $(\frac{M_{ic}}{0.87*x(6)*z} - \frac{x(14)*\pi*x(13)^2}{4}) < 0$

g(x_{19}) for minimum reinforcement, $(\frac{0.26*0.3*f_{ck}{}^{\frac{2}{3}}*b_c*d_{support}}{f_{yk}} - \frac{M_{ic}}{0.87*x(6)*z}) < 0$

g(x_{20}) for maximum reinforcement, $\left(\frac{x(14)*\pi*x(13)^2}{4} - 0.04 * area\ of\ concrete\right) < 0$

g(x_{21}) punching shear at drop panel, $(V_{ed} - \left(0.12 * k(100\rho_1 f_{ck})^{\frac{1}{3}}\right) u * d_{span}) < 0)$

g(x_{22}) punching shear at control perimeter,

$$(V_{ed} - \left(0.12 * k(100 * \rho_1 * f_{ck})^{\frac{1}{3}}\right) u_1 * d_{span}) < 0)$$

g(x_{23}) punching shear at column head,

$$\left(V_{ed} - 0.5u_0 d_{span}\left(0.6\left(1 - \frac{f_{ck}}{250}\right)\right) * \frac{f_{ck}}{1.5}\right) < 0)$$

g(x_{24}) deflection check for middle strip, $\left(actual\ \frac{l}{d_{span}} - \left(\frac{l}{d_{span}} * f_1 * f_2 * f_3\right)\right) < 0$

g(x_{25}) deflection check for column strip, $\left(actual\ \frac{l}{d_{support}} - \left(\frac{l}{d} * f_1 * f_2 * f_3\right)\right) < 0$

The design is formulated with all the given design data and the constraints according to Eurocode in MATLAB. The best solution for cost and embodied carbon emission is analysed separately using single-objective GA optimization with elitism. The elite count of individuals that survive to the next generation is given by the formula:

$$EC = 0.05 * maximum\ [minimum\big((10 * number\ of\ variabes, 100), 40\big)]$$

Due to the availability of nonlinear constraint, the population is taken as a double vector and the initial population is created using constraint dependent creation function which automatically selects the starting population best suited for the constraints provided. The fitness of the population is sorted using the rank scaling where all the individuals are given a rank based on their performance for the objective function. The scaled individuals are then chosen for next generation using the stochastic uniform method. The mutation and crossover function are constraints dependent as well. The augmented Lagrangian penalty function is also implemented if the constraints are not satisfied. The initial penalty used is 10 with a penalty factor of 100. The number of iterations performed by the GA is given by the formula:

$$Generation = 100 * Number\ of\ variables$$

The stopping criteria are set by the average change in values of each iteration. The algorithm stops if the function tolerance value is less than 10^-6 and the constraint tolerance is less than 10^-3. Iterations for cost optimization and carbon optimization are 211 and 96 respectively.

Data	Manual	Cost Optimized	Carbon Optimized
Thickness (mm)	250	200	200
Thickness of drop panel (mm)	100	85	135
Dimension of drop panel (mm)	2500	2200	2500
Dimension of column head (mm)	1200	1500	1300
Concrete Strength (MPA)	30	20	25
Steel Strength (MPA)	500	600	500
Cost (Euros)	3073	2526	2702
Carbon (kg-CO_2)	7586	6594	6021

Table 12: Result comparison for flat slab

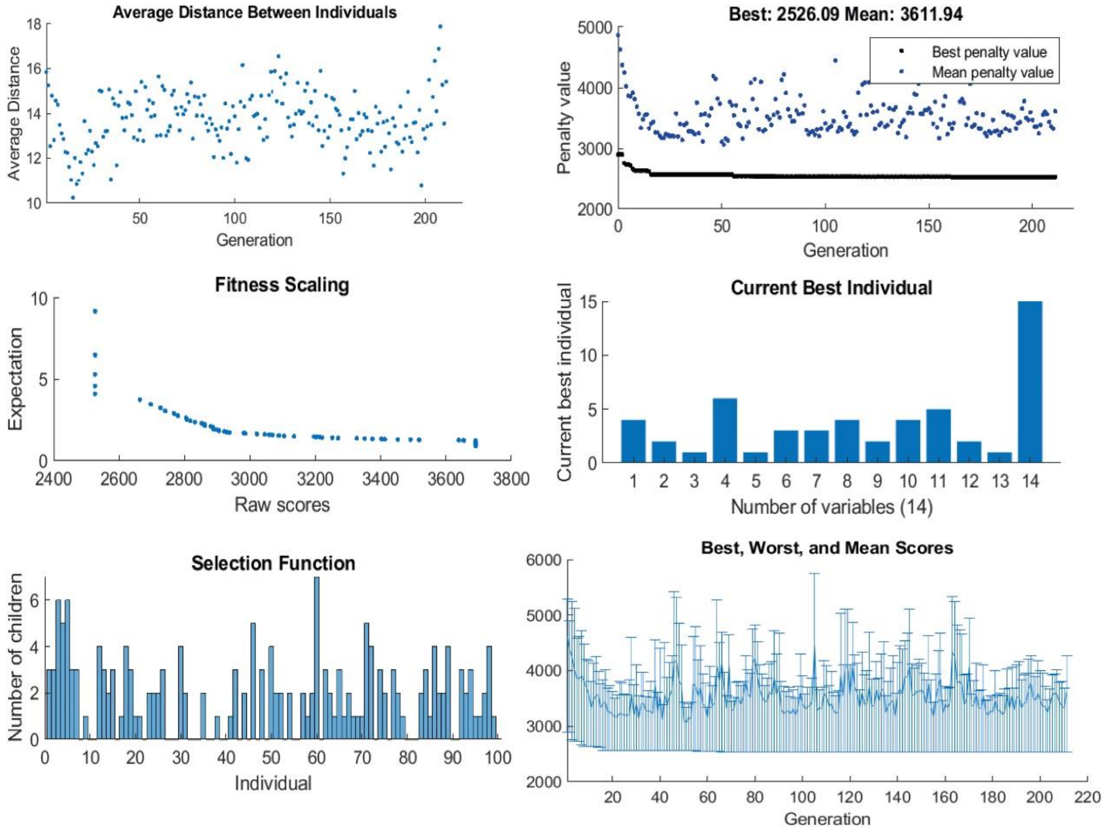


Figure 37: MATLAB cost optimization results (Mathworks, 1984)

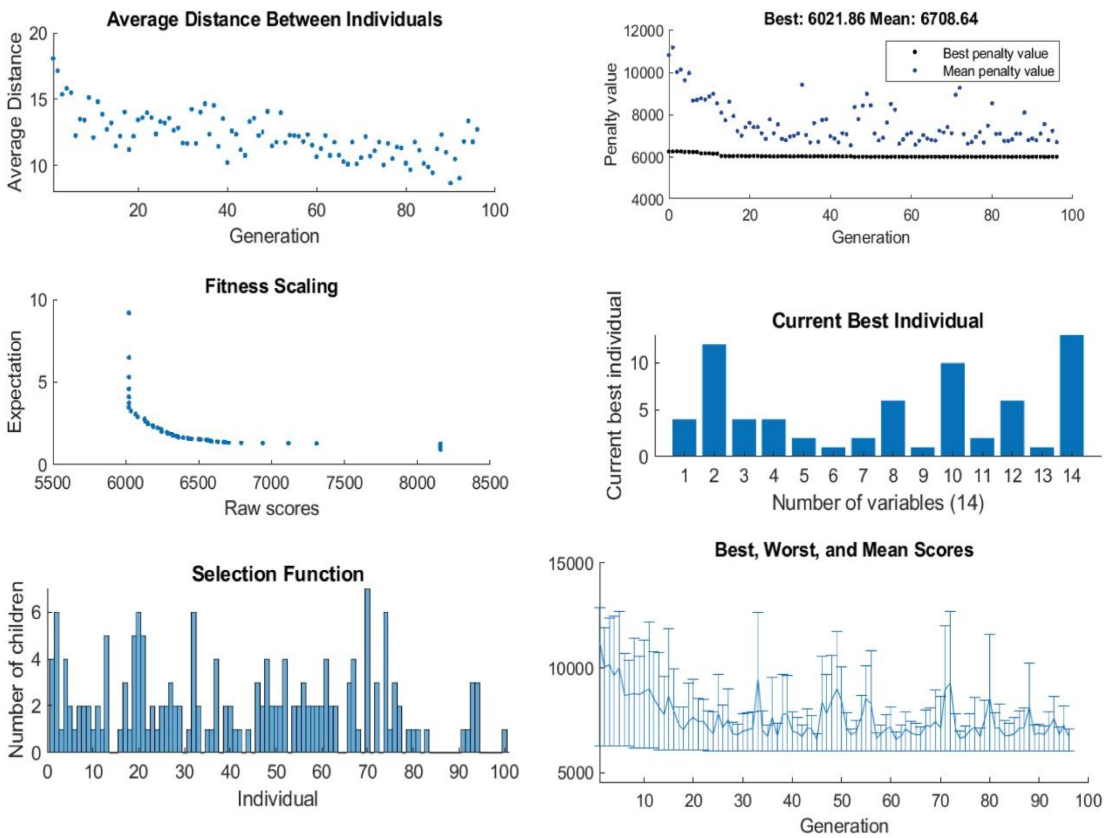


Figure 38: MATLAB carbon optimization results (Mathworks, 1984)

5.4 Design of RCC Frame

The RCC (reinforced cement concrete frame) frame is a combination of horizontal and vertical structural members which are mainly slabs, beams and columns. The slabs and beams are horizontal members, and the column is a vertical member. The loading of the structure is taken by the slab and transferred to the beams which further transfer it to the columns. The columns transfer the load to the foundations. The design of each member is given in the Eurocodes. The analysis of the frame is usually done by the substitute frame method. The design procedures for slabs, beams and columns are given in previous sections in detail level. The overview of the design procedure can be seen in the figure below.

The MATLAB code is formulated with two different objectives to reduce the cost and embodied carbon emissions. The design and its constraint are prepared as per the Eurocode guidelines to make sure that the safety of the frame is not compromised. The design for slab, beams, columns in total are provided with a total of 77 variables with each one having various options which vary between 3-20 in numbers. The manually calculated results are shown in the tables below and the entire design is available in the appendix along with MATLAB codes. The result of the optimization is also depicted in the tables below. The results take 89 iterations to calculate the optimum solution. The results and typologies are shown below:

Element	Manual Calculations	MATLAB Calculations (Carbon)	MATLAB Calculations (Cost)
Slab	150 mm	125 mm	125 mm
Beams	525x300 mm	400x200	500*250
Column	300x300 mm	200x200	250*250
Span in x	7 @ 4.5 m	6 @ 5 m	6 @ 5 m
Span in y	3 @ 7 m	4 @ 5 m	5 @ 4 m

Table 13: Element geometric data comparison

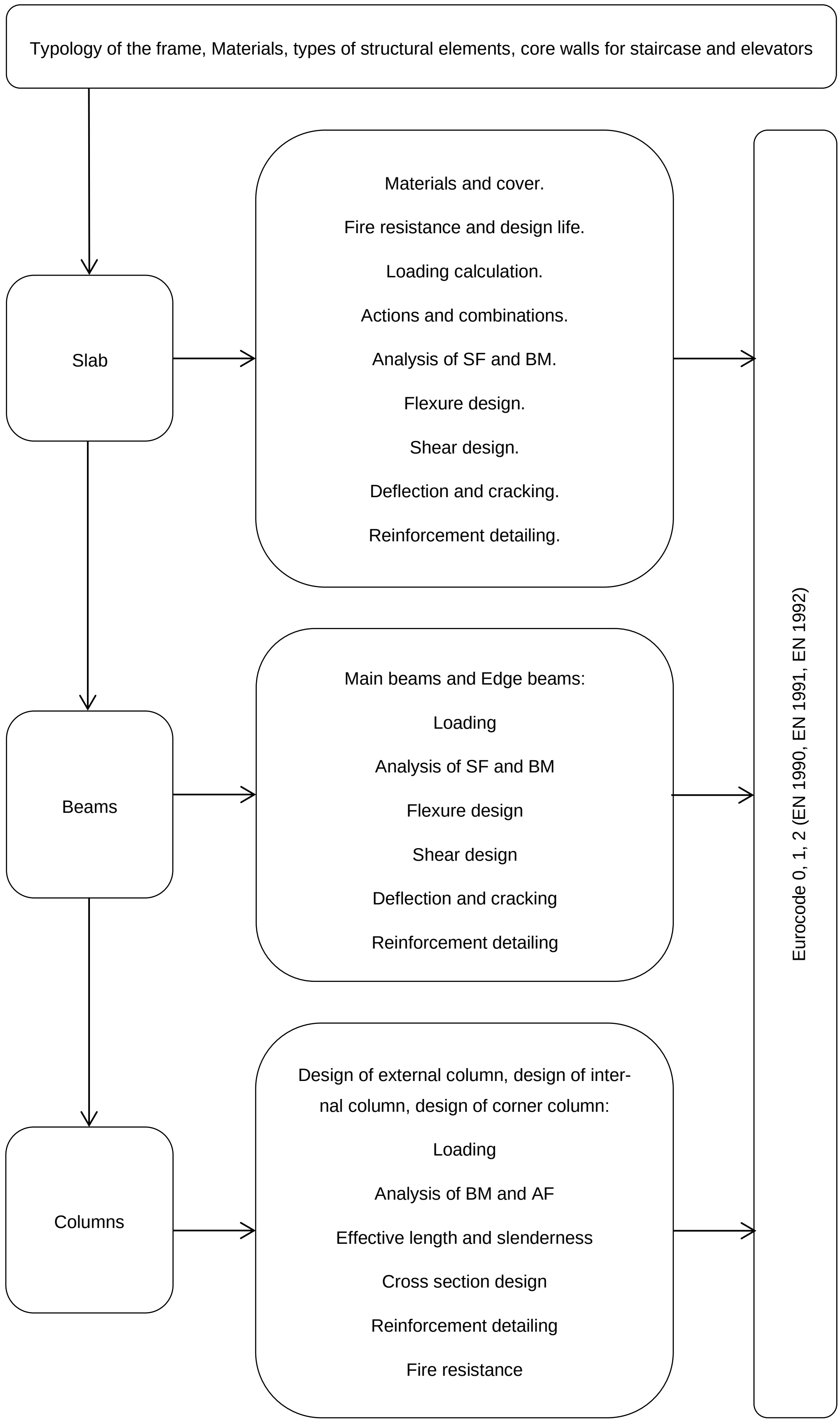


Figure 39: Frame analysis according to Eurocodes

Element	Manual Calculations				
	Steel (Kg)	Concrete (m^3)	Formwork (m^2)	Cost (Euros)	Carbon (Kg-Co_{2e})
Slab	25,546.80	598.96	315.00	86,329.34	225,210.27
Beams	11,443.00	209.30	1,039.56	32,198.19	80,912.67
Column	7,104.35	44.96	672.00	10,642.52	21,436.01
Total	44,094.15	853.22	2,026.56	129,170.05	327,558.95

Table 14: Frame manual calculation results

Element	MATLAB Calculations (Carbon)				
	Steel (Kg)	Concrete (m^3)	Formwork (m^2)	Cost (Euros)	Carbon (Kg-Co_{2e})
Slab	33,120.66	474.46	669.10	80,292.52	189,617.65
Beams	12,875.95	135.78	429.20	25,751.90	57,228.38
Column	5,108.41	46.44	510.84	8,514.02	20,196.52
Total	51,105.02	656.68	1,609.14	114,558.43	267,042.55

Table 15: Frame carbon objective results

Element	MATLAB Calculations (Cost)				
	Steel (Kg)	Concrete (m^3)	Formwork (m^2)	Cost (Euros)	Carbon (Kg-Co_{2e})
Slab	24,546.80	502.96	650.00	74,976.04	191,817.36
Beams	8,544.50	178.00	582.30	26,427.25	67,770.98
Column	2,648.35	54.96	570.00	8,175.68	20,945.18
Total	35,739.65	735.92	1,802.30	109,578.97	280,533.51

Table 16: Frame cost objective results

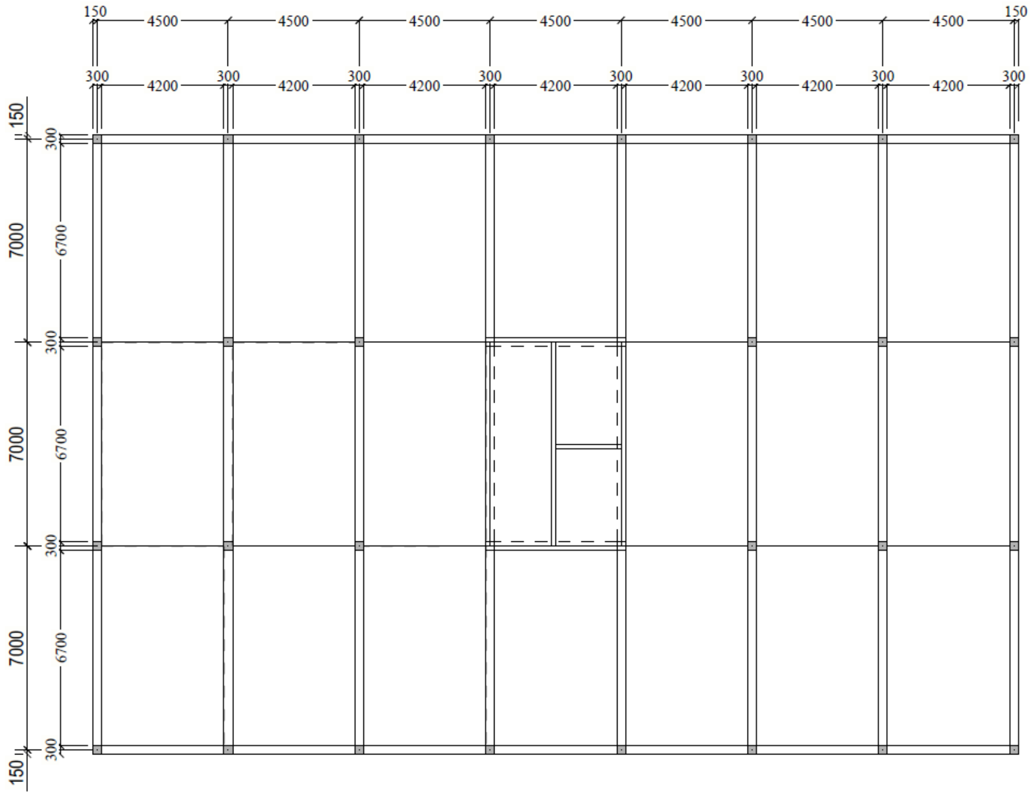


Figure 40: Plan for manual calculations (Trimble, 2021)

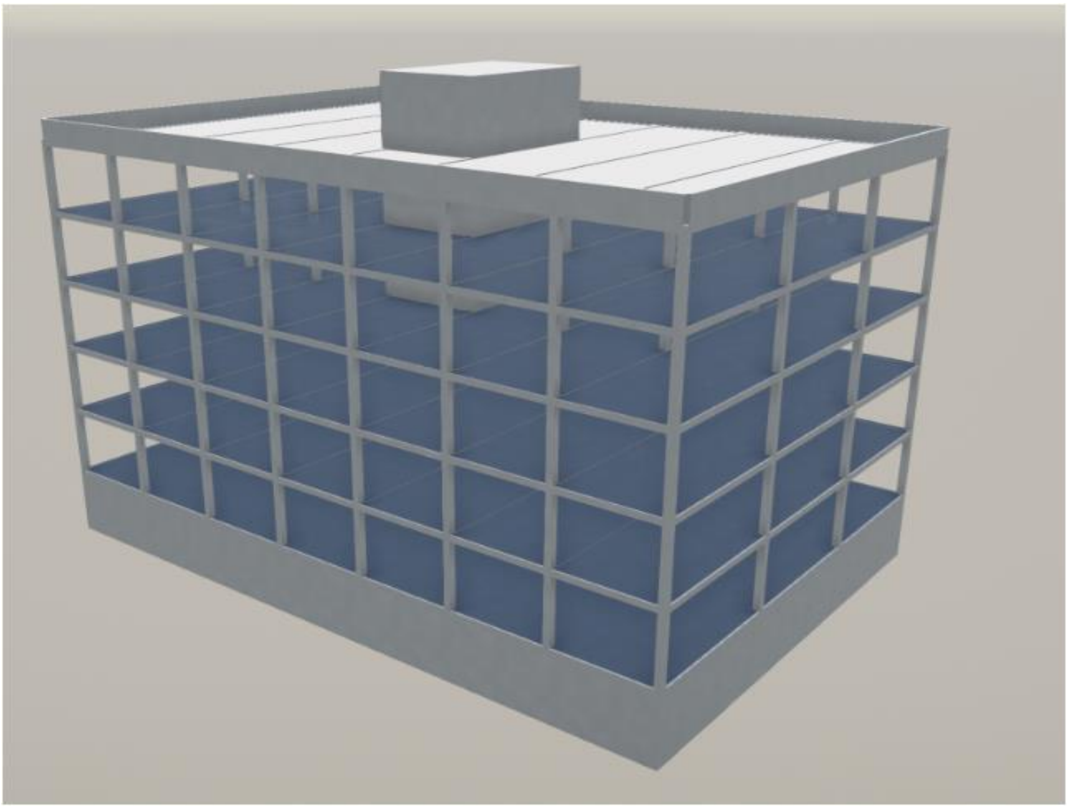


Figure 41: 3d view for manual calculation (Trimble, 2021)

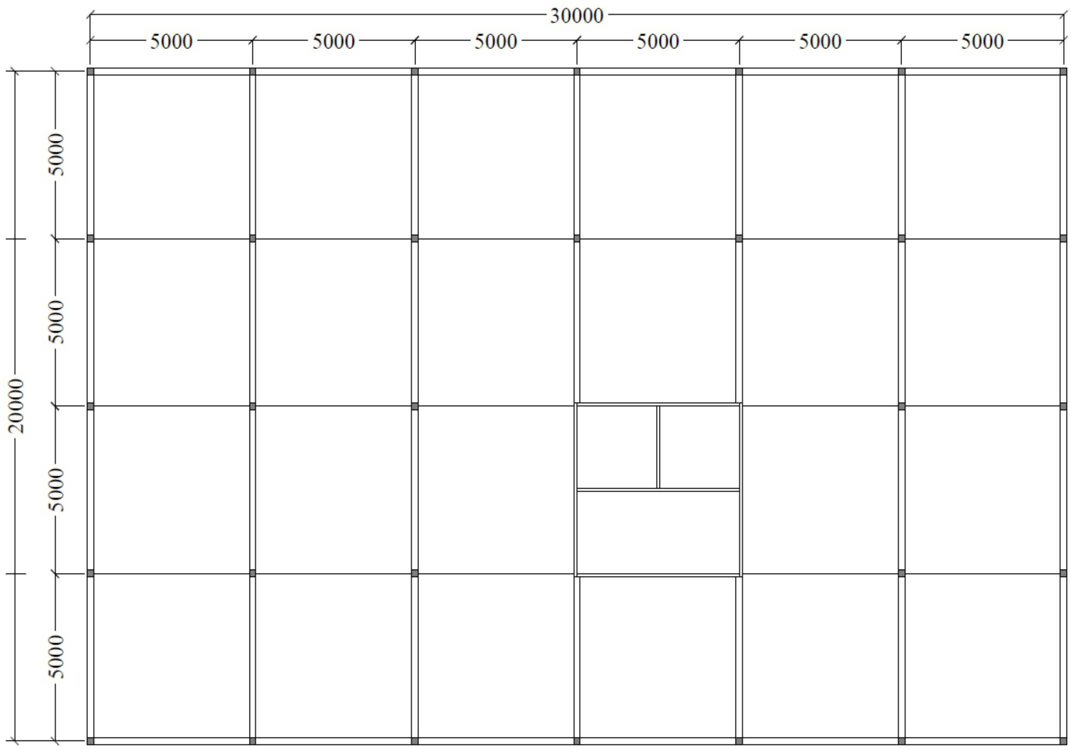


Figure 42: Plan for carbon optimization solution (Trimble, 2021)

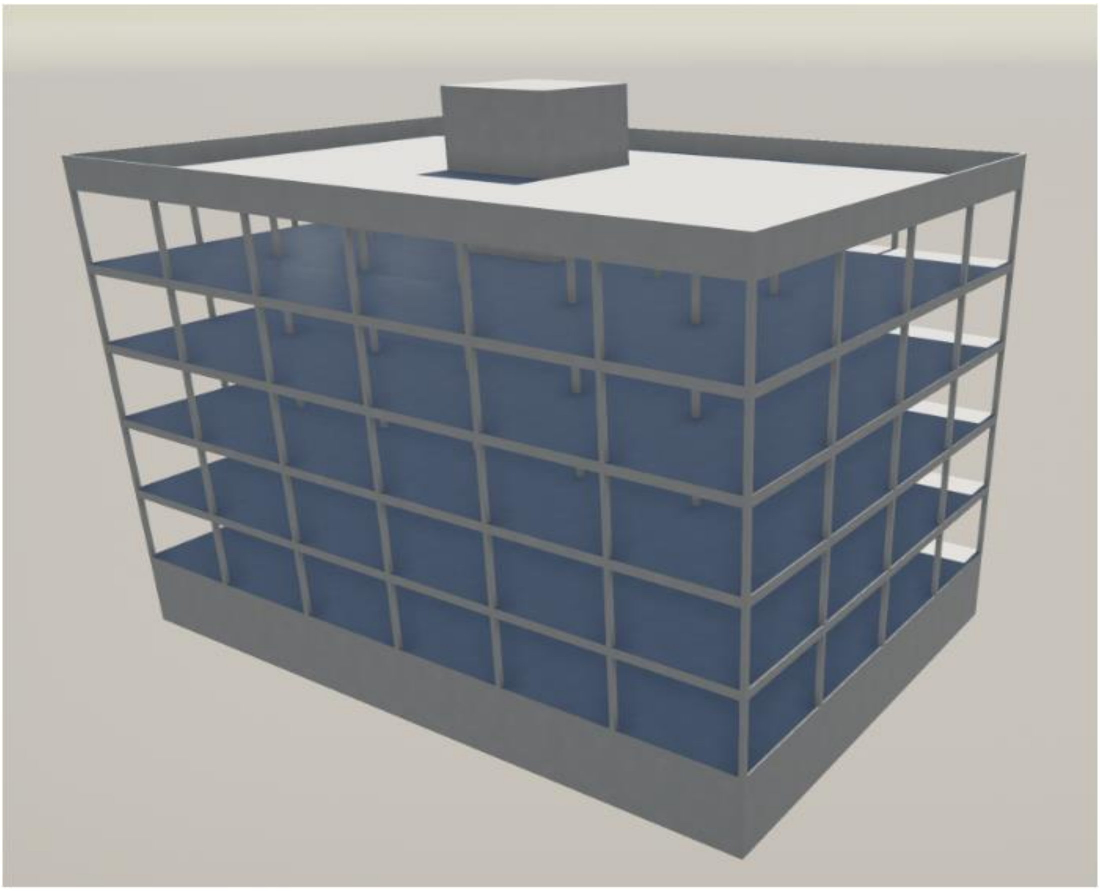


Figure 43: 3d view for carbon optimization solution (Trimble, 2021)

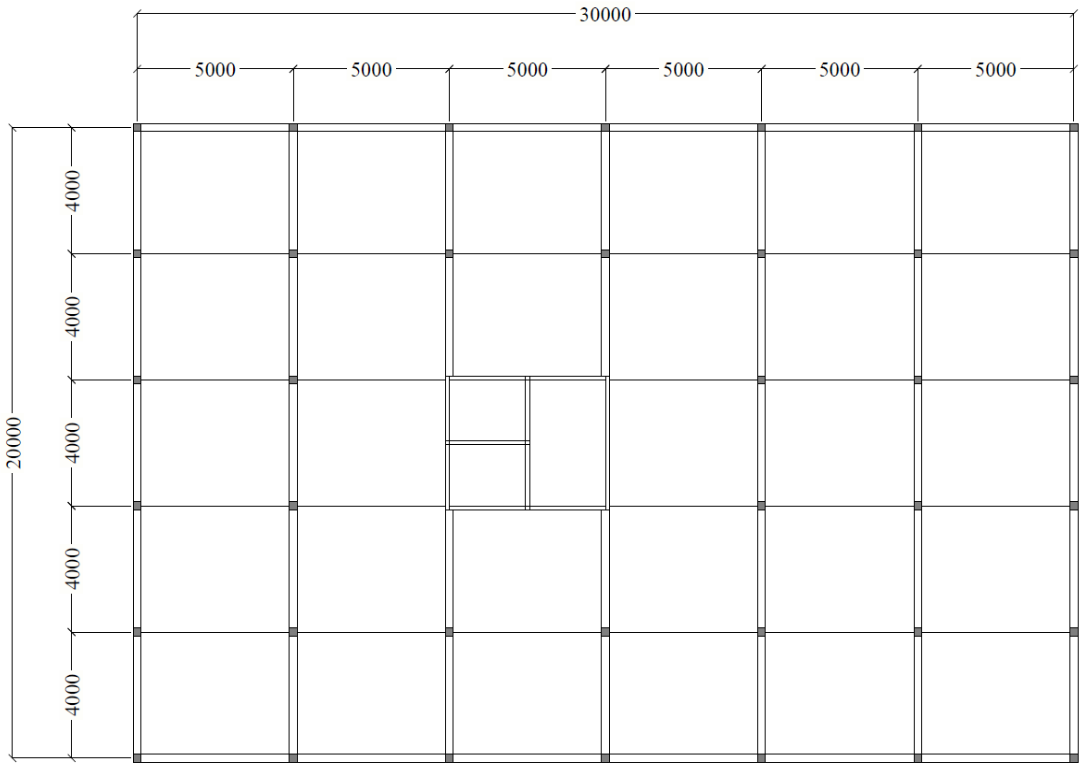


Figure 44: Plan for cost optimization solution (Trimble, 2021)

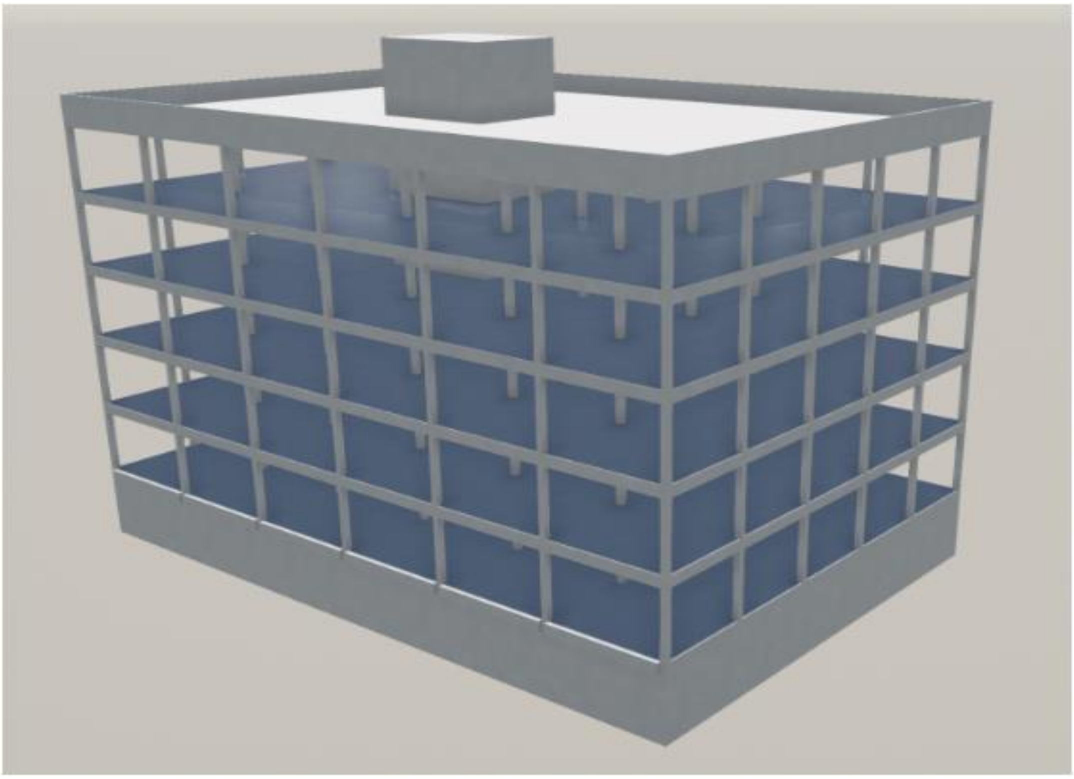
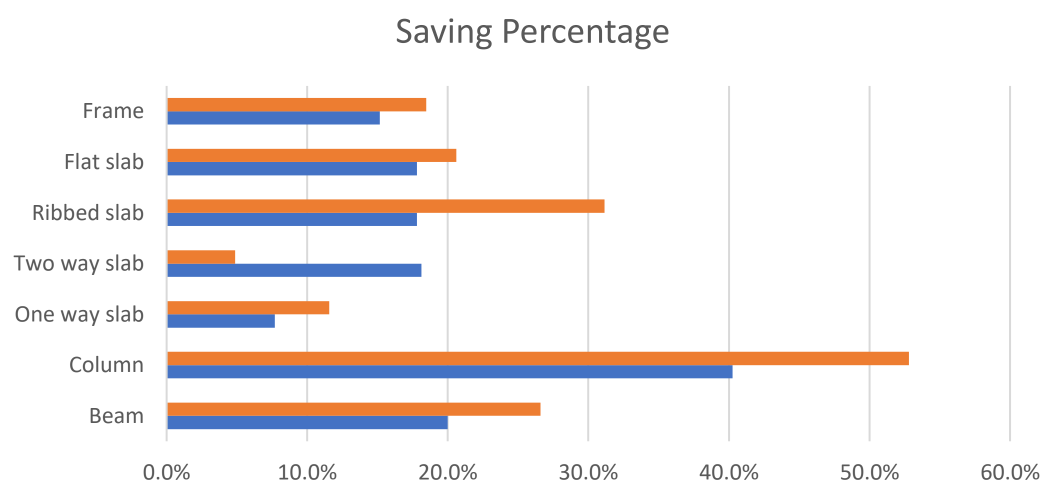


Figure 45: 3d view for cost optimization solution (Trimble, 2021)

6 Results and Analysis

The results received from the calculations will be analysed in this section with all the analyses perspectives and a critical view. The total savings in percentage can be seen from the graph below which depicts the results of optimization using genetic algorithm toolbox in MATLAB compared to that of manual results. All of the tables resulting in this graph can be found in individual sections and all the codes are attached in the appendix. The slabs tend to show the least saving among all which is mostly due to the high quantity of concrete and lower thickness differences with the manual calculations. The beams and columns demonstrate higher savings because they have a wide number of combinations for breadth and depth keeping their lengths and height constant. The frame design also shows significant savings which also happens to have the greatest number of variables and hence much more possibilities for more combinations. At the same time, the frame design has the greatest number of constraints also which are required to satisfy the design as per Eurocodes. Further analysis is done in the below subsections of each element diving deeper into understanding the results that have been achieved.



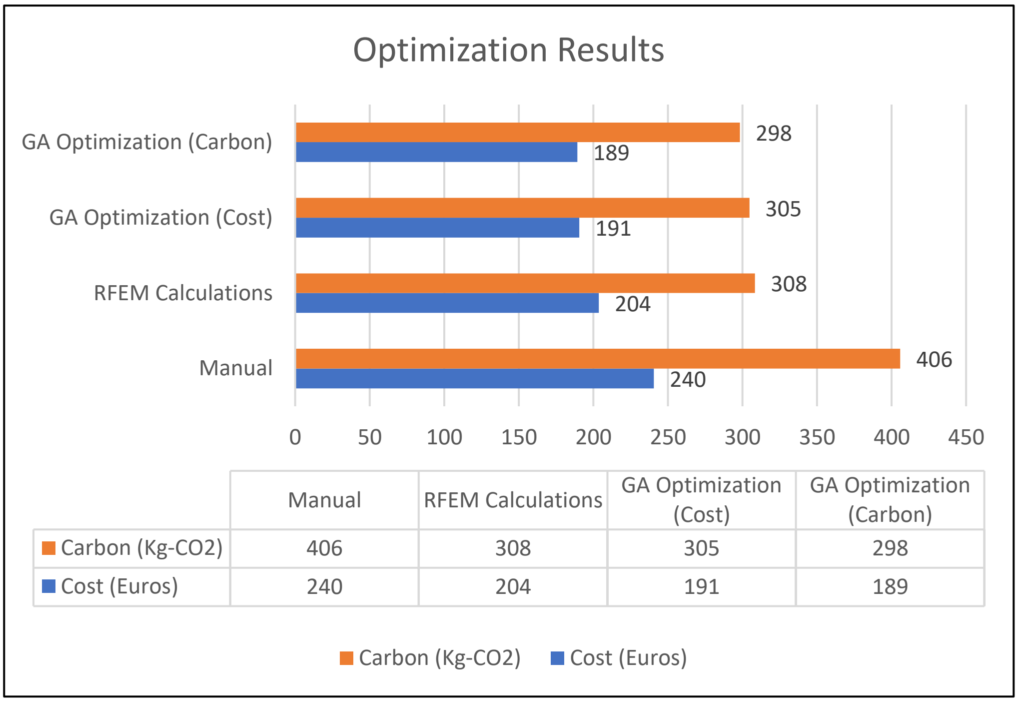
	Beam	Column	One way slab	Two way slab	Ribbed slab	Flat slab	Frame
Carbon	26.6%	52.8%	11.6%	4.9%	31.2%	20.6%	18.5%
Cost	20.0%	40.3%	7.7%	18.1%	17.8%	17.8%	15.2%

Carbon Cost

Figure 46: Total percentage saving using GA

6.1 Reinforced Concrete Beam

The beam is the only structural element, which is designed manually, using RFEM and MATLAB. The RFEM was used to see what difference it would make and whether it is worth using a genetic algorithm or sticking with software is the best option. The software indicates significantly different from manual calculations, but the optimization seems to increase the saving further therefore, the option of coding is incorporated. The results can be seen in the graph below which are calculated by using two objective functions each of which concentrate on the optimization of cost and carbon separately:



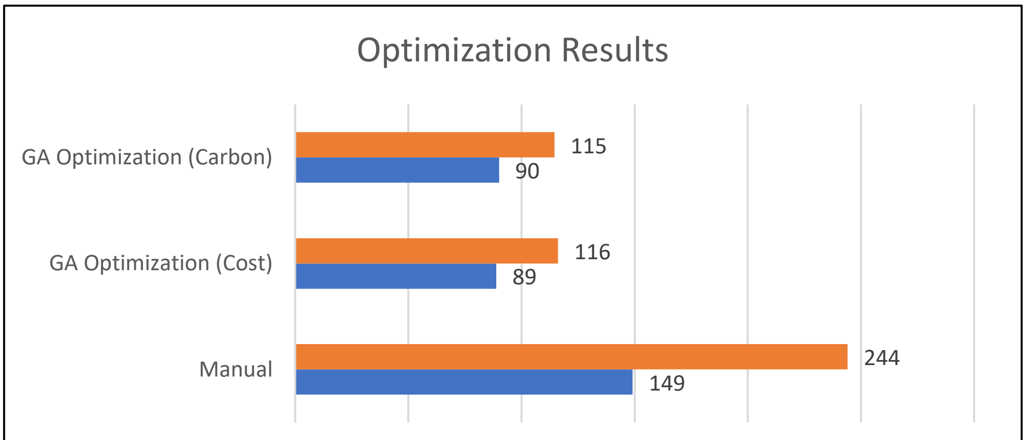
	Manual	RFEM Calculations	GA Optimization (Cost)	GA Optimization (Carbon)
Carbon (Kg-CO2)	406	308	305	298
Cost (Euros)	240	204	191	189

Figure 47: Beam optimization result comparison

The results depict that the number of variables has a great impact on the results of cost and carbon optimization. The manual calculations and RFEM calculations are both using fixed variable values and the iteration is only one whereas in MATLAB more iterations can be done with a greater number of variables which calculate all the options which are best suited for reducing cost and embodied carbon. In beams, the length is the same for all the calculations therefore the results are hugely dependent on the breadth and depth of the beam. The MATLAB optimization achieved a reduction of about 21% in cost and 27% in embodied carbon compared to the manual calculations. Also, the reduction of 15% and 25% in cost and carbon when structural analysis software RFEM is used.

6.2 Reinforced Concrete Column

The design of the column is done as a braced column with automation to decide on the slender column or short column within the code. The results from the column are very impressive with carbon almost reduced to half making it the most optimized element. However, despite the column being similar to beam in the sense that it also depends greatly on the choice of breadth and depth of its geometry shows much better results comparatively. This might be because of different loading values and better theoretical results of the interaction diagrams which is incorporated in a formula calculation instead of graphical or tabulated values in MATLAB. On the contrary to beams the different objective function seems to provide identical solution meaning the genetic algorithm is not able to find any better solution which saves more embodied carbon compared to the best cost-optimized solution. The saving of about 50% and 40% in carbon and cost respectively can be achieved with columns which is fairly a huge number.

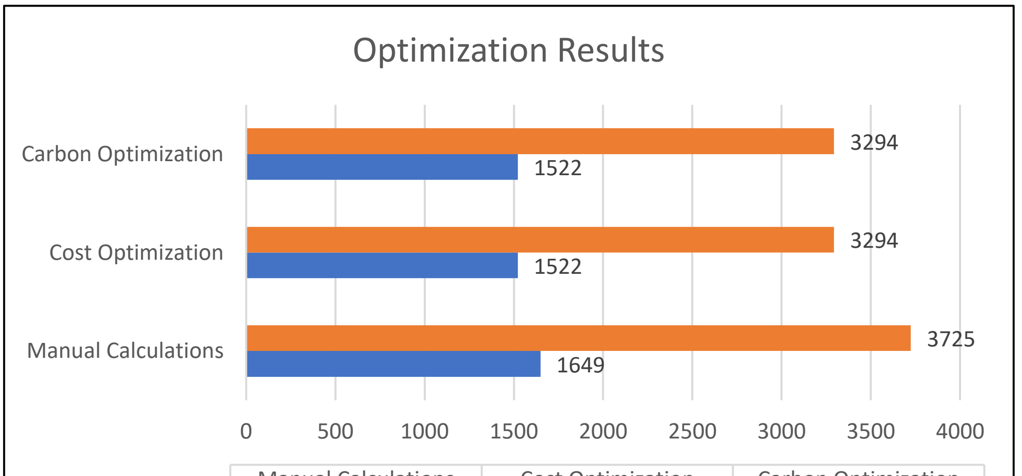


	Manual	GA Optimization (Cost)	GA Optimization (Carbon)
Carbon (Kg-CO2)	244	116	115
Cost (Euros)	149	89	90

Figure 48: Column optimization result comparison

6.3 Reinforced Concrete Slabs

The slabs are designed as conventional one way and two-way slabs, ribbed slabs and flat slabs. The span dimensions are different for each slab and therefore it would not be possible to compare them directly as per the cost and carbon values. However, they can be compared based on the percentage saved from manual design. The results are quite interesting for slabs and are to be expected in this pattern. The conventional slabs being thicker and having beams as well makes them use more quantities of concrete as well as steel making it the least element in terms of saving percentage compared to other slabs. The one-way slab and two-way slabs show a difference of about 7% and 18% for cost and 11% and 5% for carbon. This result is the perfect example of why designing using an optimization toolbox can provide a broad perspective is to what is the trade-off between cost and carbon optimized solutions and whether the decision-makers are ready to give more weightage to the environment and risk saving less on designed elements. Also, the less cost saving in two-way slabs when the objective is to optimize embodied carbon can be related to the use of more steel in both directions with top and bottom reinforcements as per the Eurocodes and to the fact that there are more constraints for two-way slab compared to that of the one-way slab. The steel tends to have more embodied carbon emissions compared to that of concrete which is the case in the two-way slab.



	Manual Calculations	Cost Optimization	Carbon Optimization
Carbon (Kg-CO2)	3725	3294	3294
Cost (Euros)	1649	1522	1522

Figure 49: One-way slab optimization result comparison

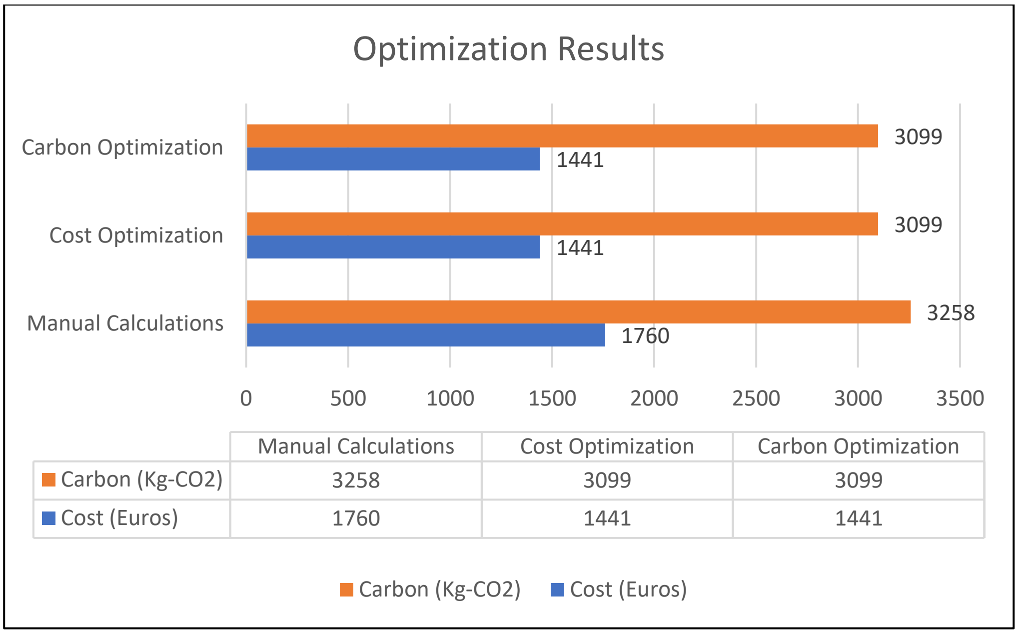


Figure 50: Two-way slab optimization result comparison

The ribbed slabs are an alternative to the conventional slabs which are way lighter uses fewer materials and are terrific in handling heavier loads. Therefore, it is no surprise that the cost of the ribbed slab is almost half of one way and two-way slabs even though the bay area is just 5 m^2 less. The fact that the ribbed slabs are used to have lighter slabs makes them already efficient to conventional slabs. Also, interestingly that the coding is further able to achieve 18% saving in cost and about 31% in embodied carbon.

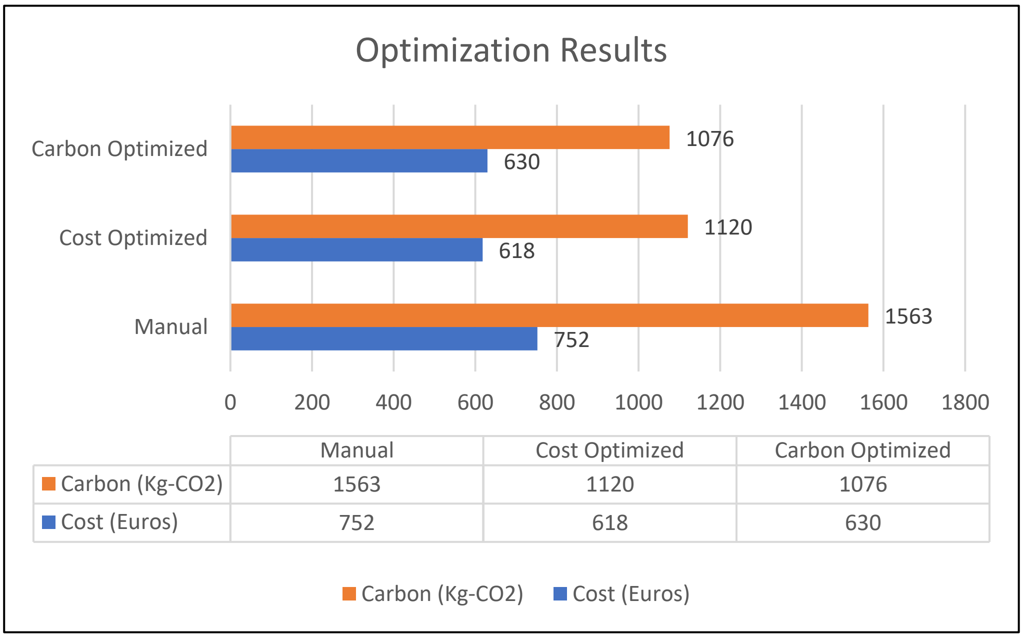


Figure 51: Ribbed slab optimization result comparison

The flat slabs generally tend to be cheaper than conventional slabs and rightly so in our case it seems almost double which might be because the bay area of the slab is 42.3 m^2 compared to that of the two-way slab which is about 40 m^2. Also, the imposed load is almost double which helps to make sense as to why the costs are so high compared to other slabs. The optimization is significant in flat slabs reducing about 18% in cost and 20% in embodied carbon which is only slightly inferior to ribbed slabs.

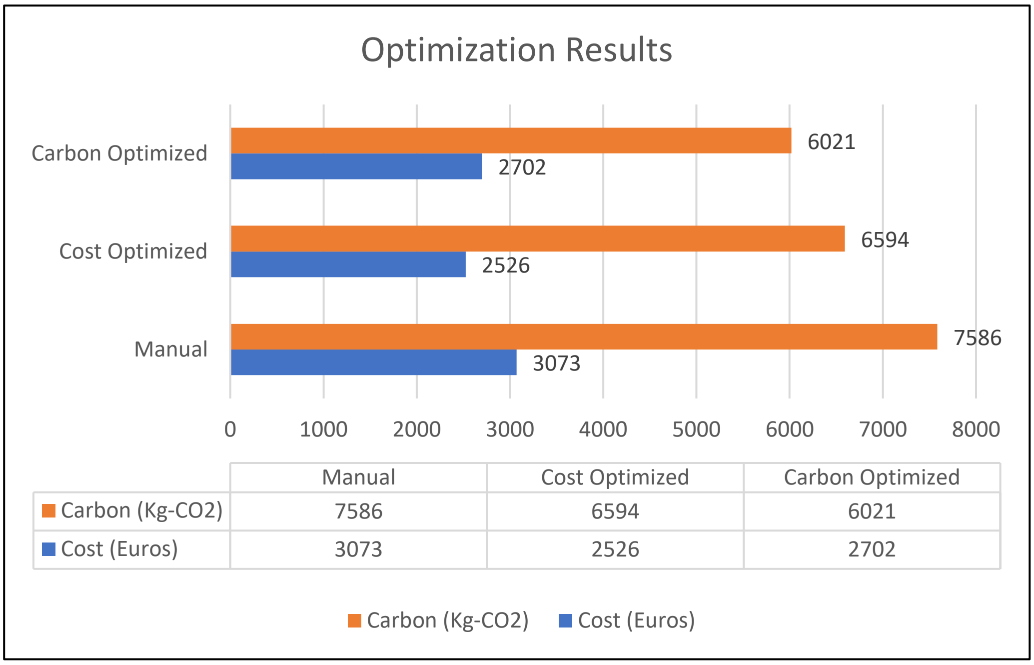


Figure 52: Flat slab optimization result comparison

6.4 RCC Frames

The frames are the hardest to analyse directly with the initial data as it is an extremely complicated and bigger design compared to that of individual elements. The frame does however show expected results as the slabs constitute the biggest portion followed by beams and then column. The individual saving for each element can be seen from the graphs below in separate graphs for cost and embodied carbon optimization objectives. The total frame cost saving is about 15% which is slightly less than 18% in the case of carbon. Also, it is worth noting that the different objectives produce different results such as geometric values and typology of the frame. The significant difference in the frame is for columns which shows only 24% saving in cost and 6% in carbon as compared to the individual calculations where they showed the most amount of saving among all elements.

It is challenging to make sense of it however the most acceptable reason might be the differences in loading because of the higher length of spans in x-direction compared to that of manual design spans. The detailed results can be seen from the below graphs for individual elements.

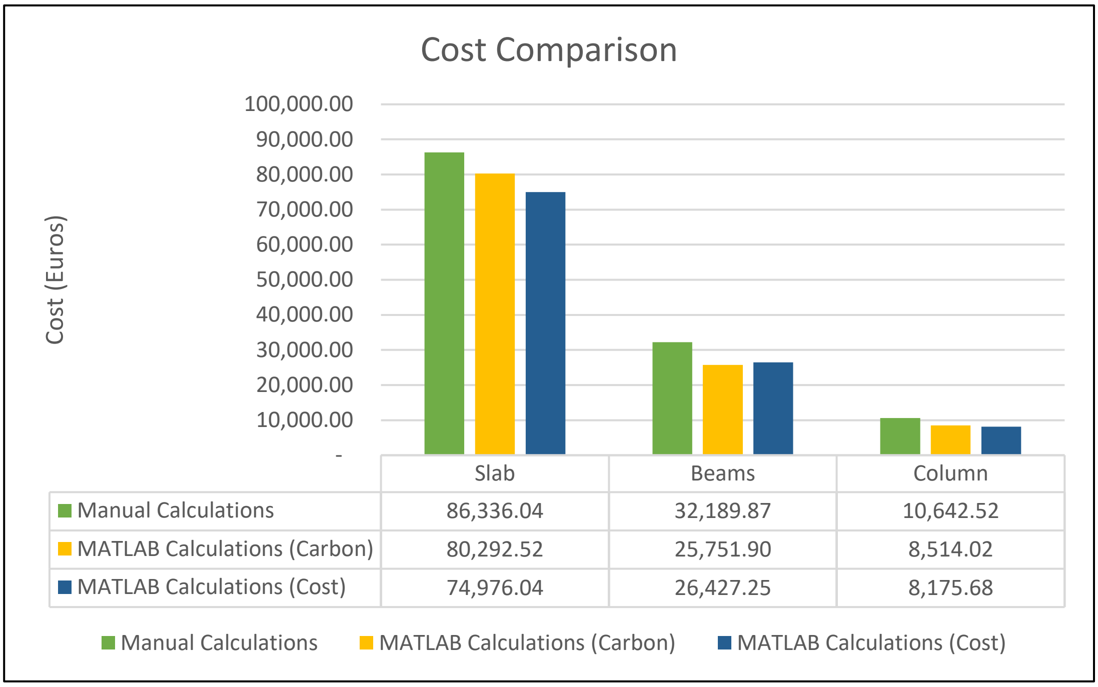


Figure 53: Elements cost comparison

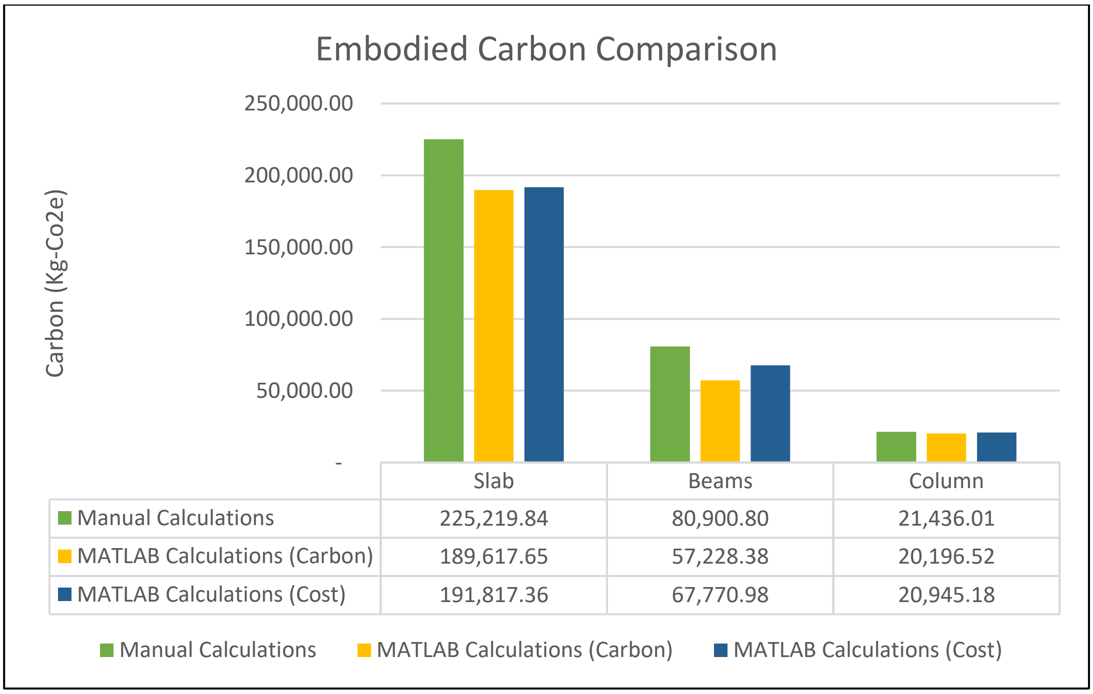
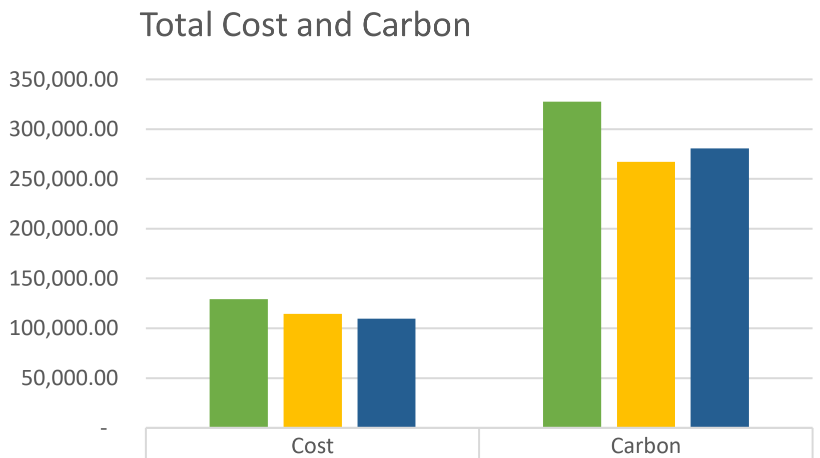


Figure 54: Elements embodied carbon comparison

The comparison of the frame with other individual elements in terms of percentages might not seem to be an attractive difference but it is important to remember that the costs of the frames are hugely higher and cannot be compared with that of the individual elements. To put it in perspective the 15% saving of cost leads to saving of about 20,000 euros which is significant. Also, the most interesting result is that of the carbon saving which is only 18% but leads to the saving of about 60,000 kg-co_{2e} which cannot be neglected and compared with other individual elements in terms of percentages because the amount involved in the frame is much higher in scale.



	Cost	Carbon
Manual Calculations	129,168.43	327,556.65
MATLAB Calculations (Carbon)	114,558.43	267,042.55
MATLAB Calculations (Cost)	109,578.97	280,533.51

Figure 55: Total frame cost and embodied carbon

Depending on these results and the typologies of different solutions that are shown in Figures 40-45, the designer can make a call as to what option suits the needs of the client and the principles of the company. Also, it is simpler to see the results visually with the models for each solution making it easier for the designer to convey his points clearly and effectively.

7 Conclusion

The argument of using automation techniques in looking for optimized solutions when it comes to reducing cost and embodied carbon emission is strongly supported by the results of this research. This research aimed to optimize the structural elements such as beam, column, slab, and the building frame in terms of cost and embodied carbon at an early stage of the design phase using a genetic algorithm. The results strongly show that the cost and embodied carbon are significantly reduced in solutions found using the genetic algorithm approach compared to the manual design. The biggest reason for this is the powerful computation ability of the genetic algorithm toolbox in MATLAB to solve problems with a greater number of variables and extremely high iteration requirements which is next to impossible to achieve manually and in a short time frame. Also, the optimization approach showed that it can solve a huge number of combinations for extremely difficult design and successfully present the best solution.

The idea that optimization can support decision making early during the design phase seems to be true as it is clear from the results that the cost and carbon optimized solutions are mostly different which leads to having manual design solutions, cost-optimized solutions and carbon optimized solutions for a same structural element or frame. The mere knowledge of these solutions can help senior managers to choose an option that relates to their vision as to whether they prefer to have low carbon emissions from their structure or want to just save as much money as they can and make a compromise with the carbon emissions. Therefore, it makes a significant difference for making an efficient and effective decision which we do not get from manual design or even from the use of structural analysis software.

Also, the different solutions are a direct result of choosing different geometric values for the elements such as breadth, depth, thickness etc. This relation has been explored beautifully with this research playing extensively with a huge number of variable options which can be seen from the codes and the solutions received. The result indicates that the more the variables and options within that variable are the better chances are there for the genetic algorithm to search for the most optimized solution. Also, when the geometric values are of similar importance, they have a greater impact on the solution i.e., the breadth and depth in the case of beams have almost equal weightage and hence they can be played with to have a greater impact on the results.

The use of BIM was not very significant in this approach as there were not much constant data that was required for the genetic algorithm coding because all the data was taken to be variable to provide the greater option to the algorithm to have more combinations. The use of BIM software such as Revit and Tekla is being used in the case of the beam and frame to have a visual understanding of the different solutions. The data is exchanged through the csv or excel files. Although the use of BIM was thought to be a powerful tool in this research it did not have a significant impact on this particular methodology.

The approach of using the genetic algorithm by optimization toolbox of MATLAB for computation is a very powerful, efficient, and effective technique compared to the vast number of optimization approaches used around in the academic world. The fact that the concrete design has a huge number of variables that need to be interpreted throughout the design made it extremely hard to automate the optimization and create a logic for it which also generated a research gap that there were not a lot of research being done for concrete structures compared to the steel structural design. Therefore, the current research is a significant contribution to the optimization of concrete structural elements and the entire building structural frame making the gap slightly less.

In a nutshell, the methodology of choosing a metaheuristic approach of using a genetic algorithm that belongs to the evolutionary algorithm class gave impressive results when it came to finding the most optimized solution from a bunch of various solutions. The genetic algorithm toolbox in MATLAB allowed the implementation of Eurocode and its design constraints successfully making the design very safe which is the first and foremost aspect of construction. The achievement of optimized results also signifies that the materials being used are less and emissions are reducing which helps to lower the emissions of the construction industry as a whole and contribute to the sustainability goals.

7.1 Limitations

There were few limitations in this research that were significant. Firstly, the computation of complicated and big structures is extremely hard to solve in the coding logic as there are a high number of variables and the design is significantly longer with a huge number of constraints that need to be met as per Eurocodes safety point of view requiring a high level of focus and time. Secondly, the knowledge of coding required to solve the complicated design problems is very high and hence at some point hindered the optimal and efficient writing of codes making them longer. Thirdly, there is not enough research that works with the entire design as it makes the application of genetic algorithms significantly harder, Therefore, most of the research done on concrete structures using genetic algorithms are addressing only one key relation of the design which have an impact on the solution and not the entire design which also made it hard to establish comparative study with my research.

7.2 Future Work

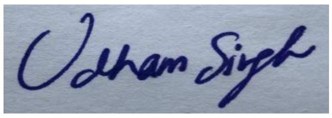
The current research is a great addition to the previous body of work in the optimization field which indicated genetic algorithms to be a powerful tool. However, to enlarge the scope of work for more complicated and real-life structures, the use of structural analysis software for design implementation as per Eurocode can be explored through an application programming interface (API) which would significantly increase the workability with complicated structures and the time requirement for coding will reduce significantly as well. Also, the analysis software is equipped with Eurocodes so the MATLAB code does not require that all the constraints must be written in the code as they will be already satisfied in the software making the coding precisely focused on the optimization.

Statutory Declaration

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper.

I clearly marked and separately listed all the literature and all of the other sources which I employed when producing this academic work, either literally or in content.

I am aware that the violation of this regulation will lead to the failure of the thesis.

Udham Singh	
Student’s name	Student’s signature
S0572561	30.July.2021
Matriculation number	Berlin, date

Consent of Publishing the Master's Thesis

Freely Given Consent according to Art. 7 GDPR

■ I agree to the publication of my final thesis in the university library of HTW Berlin and to the inclusion in the online catalogue (webOPAC) of HTW Berlin by publishing the following data.

■ I agree to the digitization of the front page and the table of contents of the thesis in Dandelon.

Family Name: Singh **First Name:** Udham

Student-ID number: S0572561 **Date of Degree:** 30.07.2021

Faculty: Engineering Sciences-Technology and Life

Study Programme: Construction and Real Estate Management

I fully understand that this allows the machine-based research in indexes for all users of the online catalogue, and that the data processed in the online catalogue of HTW Berlin is also capable of being searched and referenced by global web search engines/webcrawlers, which do not fall within the power of control of HTW Berlin.

2021-07-30 Udham Singh

Date and Signature of the Data Subject

Confirmation of the Primary Reviewer

I confirm that the final thesis of the aforementioned candidate is publishable.

Date and Signature of the Primary Reviewer

Bibliography

1) Bekdaş, G. & Niğdeli, S. M., 2016. Optimum Design Of Reinforced Concrete Columns Employing Teaching-Learning Based Optimization. *Challenge Journal Of Structural Mechanics,* November, 2(4), pp. 216-219.
2) Adeli, H. & Sarma, K. C., 2006. Cost Optimization of Structures: fuzzy logic, genetic algorithm and parallel computing. In: s.l.:John Wiley $ Sons Ltd..
3) Aga, A. A. & Adam, F. M., 2015. Design Optimization of Reinforced Concrete Frames. *Open Journal of Civil Engineering,* March.pp. 74-83.
4) Ahmadi-Nedushan, B. & Varaee, H., 2011. Minimum Cost Design of Concrete Slabs Using Particle Swarm Optimization With Time Varying Acceleration Coefficients. *World Applied Sciences Journal,* January, 13(12), pp. 2484-2494.
5) Ajmal , H. P. C., 2017. A study on cost optimised structural design of reinforced concrete beams. *International journal of scientific & engineering research (IJSER),* November.08(11).
6) Al-Ansari, M. S., 2013. Flexural Safety Cost of Optimized Reinforced Concrete Beam. *International Journal of Civil Engineering and Technology (IJCIET),* March-April, 04(02), pp. 15-35.
7) Aleksandar, M., Goran, P. & Dejan, B., 2013. Optimal structural design of reinforced concrete structures - Review of existing solutions considering applicability aspect. *Archives for Technical Sciences,* 09(01), pp. 53-60.
8) Alex, D. M. & Kottalil, L., 2015. Genetic Algorithm Based Design Of a Reinforced Concrete Cantilever Beam. *International Research Journal of Engineering and Technology (IRJET),* October, 02(07), pp. 1249-1252.
9) Alex, D. M. & Kottalil, L., 2015. Genetic algorithm based design of a reinforced concrete continuous beam. *International Journal of Engineering Research & Technology (IJERT),* September, 04(09), pp. 224-227.
10) Autodesk Revit, 2002. *Autodesk Revit.* [Online] Available at: https://www.autodesk.com/

11) Bhalchandra, S. & Adsul, P., 2012. Cost Optimization Of Doubly Reinforced Rectangular Beam Section. *International Journal of Modern Engineering Research (IJMER),* September-October, 02(05), pp. 3939-3942.

12) Bond, A. J. et al., 2006. *How to Design Concrete Structures using Eurocode 2,* s.l.: The Concrete Centre.

13) Camp, C. V., Pezeshk, S. & Hansson, H., 2003. Flexural Design of Reinforces Concrete Frames Using a Genetic Algorithm. *Journal of Structural Engineering,* January .129(1).

14) Chen, C. et al., 2019. Optimum cost design of frames using genetic algorithms. *Steel and Composite Structures,* February, 30(03), pp. 293-304.

15) Chisari, C. & Chiara, B., 2016. Multi-Objective Optimization of FRP Jackets for Improving the Seismic Response of Reinforced Concrete Frames. *American Journal of Engineering and Applied Sciences,* 27 September.

16) Chutani, S. & Singh, J., 2017. Use of Hybrid PSOGSA Search Algorithm for Optimum Design of RC Beam. *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET),* August, 03(05), pp. 480-487.

17) Clark, D. & Bradley, D., 2013. Embodied Carbon of Steel versus Concrete Buildings. In: *What Color is Your Book.* s.l.:s.n.

18) Coello Coello , C., Hernandez, F. S. & Farrera, F. A., 1997. Optimal Design of Reinforced Concrete Beams Using Genetic Algorithms. *Expert Systems with applications,* pp. 101-108.

19) Correia, R., Bono, G. & Bono, G., 2019. Optimization of Reinforced Concrete Beams Using Solver Tool. *IBRACON Structures and Materials Journal,* August.pp. 910-931.

20) de Albuquerque, A. T., El Debs, M. K. & Melo, A. M., 2012. A cost optimization-based design of precast concrete floors using genetic algorithms. *Automation in Construction,* Volume 22, p. 348–356.

21) Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions On Evolutionary Computation,* April.Volume 6.

22) Dlubal Software , 1987. [Online] Available at: https://www.dlubal.com/en

23) Eleftheriadis, S. et al., 2017. Multilevel Computational Model for Cost and Carbon Optimisation of Reinforced Concrete Floor Systems. *Automation and Robotics in Construction.*

24) Eleftheriadis, S., Duffour, P. & Mumovic, D., 2018. BIM-embedded life cycle carbon assessment of RC buildings using optimised structural design alternatives. *Energy & Buildings,* May, Volume 173, p. 587–600.

25) Eleftheriadis, S., Duffour, P., Stephenson, B. & Mumovic, D., 2018. Automated Specification Of Steel Reinforcement To Support The Optimisation Of RC Floor. *Automation In Constructuion ,* October, Volume 96, pp. 366-377.

26) Eleftheriadis, S. et al., 2018. *Integrated Building Life Cycle Carbon and Cost Analysis Embedding Multiple Optimisation levels.* Cambridge, s.n.

27) EN-1992-1-1 (CEN), 2004. *Eurocodes 2:Design of Concrete Structures Part 1-1: General rules and rules for buildings,* Brussels: The European Union.

28) European Commission, 2004. *Eurocode 2: Design of Concrete Structures - Part 1-2,* Brussels: CEN.

29) Ferdyn-Grygierek, J. & Grygierek, K., 2017. Multi-Variable Optimization of Building Thermal Design Using Genetic Algorithms. *Energies,* October, Volume 10, p. 1570.

30) Galeb, A. C. & Jennam, M. A., 2015. Optimum Design of Reinforced Concrete Flat Slabs. *Eng. &Tech.Journal,* January, 33(09), pp. 2049-2065.

31) Gandomi, A. . H., Yang, X.-S. & Alavi , A. H., 2013. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers,* April.Volume 29.

32) Gare, P. & Angalekar, S. S., 2016. Design of Structural Element Employing Optimization Approach. *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET),* July, 05(07), pp. 13808-13816.

33) Ghandi, E., Shokrollahi, N. & Nasrolahi, M., 2017. Optimum Cost design of Reinforced Concrete Slabs Using Cuckoo Search Optimization Algorithm. *International Journal of Optimization in Civil Engineering,* pp. 539-564.

34) Gibbons, O. & Orr, J., 2020. *How to calculate embodied carbon,* London: The institution of structural engineers.

35) GlobalABC , 2019. *2019 global status report for buildings and construction: Towards a zero-emission,efficient and resilient buildings and construction sector,* s.l.: Global Alliance for Buildings and Construction, International Energy Agency and the United Nations.

36) Goldberg, D. E., 1989. *Genetic Algorithm in Search, Optimization and MAchine Learning.* s.l.:Addison-Wesley Publishing Company Inc..

37) Gordillo, G. C. et al., 2020. EplusLauncher: An API to Perform Complex EnergyPlus Simulations in MATLAB® and C#. *Sustainability,* 16 January, 12(02), p. 672.

38) Govindaraj, V. & Ramasamy, J., 2005. Optimum Detailed Design of Reinforced Concrete Continuous Beams Using Genetic Algorithms. *Computers and Structures,* September.pp. 34-48.

39) Grygierek, K. & Ferdyn-Grygierek, J., 2019. Multi-variable optimization models for building envelope design using energyplus simulation and metaheuristic algorithms. *Architecture civil engineering environment,* June.

40) Guerra, A. & Kiousis, P. D., 2006. Design optimization of reinforced concrete structures. *Computers and concrete,* August, 03(05), pp. 313-334.

41) Hadi, M. N., Sharafi, P. & Teh, L. H., 2012. A Heuristic Approach for Optimim Cost and Layout Design of 3D Reinforced Concrete Frames. *Journal od Structural Engineering,* July.pp. 853-863.

42) Hooke, R. & JEEVES , T. A., 1961. Direct Search Solution Of Numerical And Statistical Problems. *Journal Of Association for Computing Machinery,* April.pp. 212-229.

43) Kaveh, A. & Shakouri Mahmud Abadi, A., 2011. Cost Optimization of Reinforced Concrete One-Way Ribbed Slabs Using Harmony Search

Algorithm. *Arabian Journal for Science and Engineering,* 19 October, Volume 36, p. 1179–1187.

44) Kim, T. H., Chae, C. U., Kim, G. H. & Jang, H. J., 2016. Analysis of CO2 Emission Characteristics of Concrete Used at Construction Sites. *Sustainability,* 08 April.p. 348.

45) Kumar, K. & Shanthi, R., 2018. Cost Optimization Using GA for Structural Beams With Different and Conditions. *International Journal of Innovative Science and Research Technology,* March.03(03).

46) Kumar, R., 2013. Cost Optimization of Industrial Building using Genetic Algorithm. *International Journal of Scientific Engineering and Technology,* April, 02(04), pp. 185-191.

47) Mangal, M. & Cheng, J. C., 2018. Automated Optimization Of Steel Reinforcement In RC Building Frames Using building Information Modeling And Hybrid Genetic Algorithm. *Automation in Construction,* June, Volume 90, pp. 39-57.

48) Mathworks, 1984. *www.mathworks.com.* [Online] Available at: https://se.mathworks.com/products/matlab.html

49) MEPS International Ltd., 1979. *www.meps.co.uk.* [Online].

50) Milajić, A., Milošević, P. & Beljaković, D., 2019. Optimal green building design using multi-objective genetic algorithm. *Reporting for sustainability,* April.pp. 119-123.

51) Mirjalili, S., 2019. *Evolutionary Algorithms and Neural Networks.* s.l.:Springer International Publishing AG, part of Springer Nature 2019.

52) Mister Concrete, 2017. *www.misterconcrete.co.uk.* [Online].

53) Mosley, B., Bungey, J. & Hulse, R., 2012. *Reinforced concrete design to Eurocode 2.* Seventh ed. s.l.:PALGRAVE MACMILLAN.

54) Najem, R. M. & Yousif, S. T., 2013. Optimum cost design of reinforced concrete continuous beams using Genetic Algorithms. *Int. Journal of Applied Sciences and Engineering Research (IJASER),* February, 02(01), pp. 79-92.

55) NBIMS-US, 2005. *National Institue of Building Sciences.* [Online] Available at: https://www.nibs.org/page/factsheets

56) Niaki, M. I., Maheri, M. R. & Bagheri, M., 2016. Optimum Design of 2-D Reinforced Concrete Frames Using a Genetic Algorithm. *The Turkish Online Journal of Design, Art and Communication,* pp. 1909-1919.

57) Nigdeli, S. M. & Bekdas, G., 2019. The Effect Of The Consideration Of Slab Dimensions On Optimum Design Of Reinforced Concrete Beams. *European Journal Of Engineering and Natural Sciences,* October, 3(2), pp. 81-85.

58) Olawale, S. O. A. et al., 2019. Cost Optimisation of the Design of Reinforced Concrete Flat Slab to BS8110. *Materials Science and Engineering 640 (2019) 012052.*

59) Oliveira, J. I. F. & Miranda, A. C. O., 2020. Structural Optimization Using Multi-Objective Genetic Algorithm. *D. Chiranjevulu Journal of Engineering Research and Application,* March, 10(3), pp. 1-12.

60) Patil, K. S., Gore, N. & Salunke, P., 2013. Optimum Design of Reinforced Concrete Flat Slab with Drop Panel. *International Journal of Recent Technology and Engineering (IJRTE),* September.02(04).

61) Paya-Zaforteza, I., Yepes, V., Hospitaler, A. & Gonzalez-Vidosa, F., 2009. CO2- Optimization of reinforced concrete frames by simulated annealing. *Engineering Structures,* Volume 31, pp. 1501-1508.

62) Perea, C. et al., 2007. Design of reinforced concrete bridge frames by heuristic optimization. *Advances in Engineering Software,* 19 July.

63) Premalatha, C., 2015. Genetic Algorithm For Optimization Problems. *International Journal Of Engineering Sciences & Research Technology,* May.

64) Raje, S. S. & Patel, R. B., 2017. Column Capital and Drop Panel Optimization of Flat Slab Using Genetic Algorithm. *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET),* April, 06(04), pp. 5960-5965.

65) Rojas, A.-L., 2016. Numerical Experimentation for teh Optimal Design of Reinforced Rectangular Concrete Beams for Singly Reinforced Sections. 10 March, Volume DYNA 83(196), pp. 134-142.

66) Sahab, M. G., Ashour, A. F. & Toropov, V. V., 2005. Cost Optimisation of Reinforced Concrete Flat Slab Building. *Engineering Structures,* pp. 313-322.

67) Sahab, M. G., Toropov, V. V. & Ashour, A. F., 2004. A Hybrid Genetic Algorithm For Structural Optimization Problems. *Asian Journal of Civil Engineering (Building And Housing),* Volume 5, pp. 121-143.

68) Salataa, F. et al., 2020. Effects Of Local Conditions On The Multi-Variable And Multi-Objective Energy optimization of Residential Buildings Using Genetic Algorithms. *Applied Energy,* February.Volume 260.

69) Shariati, A. et al., 2019. Using Genetic Algorithms Method For The Paramount Design Of Reinforced Concrete Structures. *Structural Engineering and Mechanics,* April, Volume 71, pp. 503-513.

70) Singh, B. & Rai, H. S., 2014. Optimisation of RCC Beam. *International Journal of Engineering, Business and Enterprise applications (IJEBEA),* June-August, 09(01), pp. 21-34.

71) Singh, H., Rai, H. S. & Singh, J., 2014. Discrete Optimisation of One way Slab using Genetic Algorithm. *International Journal of Engineering, Business and Enterprise Applications (IJEBEA),* June-August, 09(02), pp. 116-121.

72) Sivanandam, S. & Deepa, S., 2008. *Introduction to Genetic Algorithms.* s.l.:Springer-Verlag Berlin Heidelberg.

73) Suryavanshi, D. K. & Akhtar, S., 2019. Design optimization of reinforced concrete slabs using various optimization techniques. *International Journal of Trend in Scientific Research and Development (IJTSRD),* August, 03(05), pp. 45-58.

74) The Matworks Inc., 2004-2018. *Global Optimization Toolbox User's Guide.* [Online]
Available at: https://se.mathworks.com/products/global-optimization.html

75) Thomas, S. M. & Arulraj.G, P., 2017. Optimization of Singly Reinforced RC Beam. *International Journal of Research-Granthaalayah,* February.pp. 199-207.

76) Tomás, A. & Alarcón, A., 2012. Automated Design of Optimum Longitudinal Reinforcement For Flexural And Axial Loading. *Computers and Concrete,* August, 10(02), pp. 149-171.

77) United Nations Environment Programme, 2020. *2020 Global Status Report for Buildings and Construction: Towards a Zero-emission, Efficient and Resilient Buildings and Construction Sector,* Nairobi: s.n.

78) Vasinton, S. & Raslan, R., 2016. *Multi-Objective optimisation for the minimisation of life cycle carbon footprint and life cycle cost using NSGA II: A refurbished high rise residntial building case study..* Newcastle, Building Performance and Optimization Conference BSO16.

79) Whitworth, A. . H. & Tsavdaridis, K. D., 2020. Genetic Algorithm for Embodied Energy Optimisation of Steel-Concrete Composite Beams. *Sustainability,* April, 12(08), p. 3102.

80) World Steel Association, 2019. [Online] Available at: https://www.worldsteel.org/

81) www.concretecenter.com, n.d. [Online].

82) Yeo, D. & Gabbai, R. D., 2011. Sustainable design of reinforced concrete structures through embodied energy optimization. *Energy and Buildings,* April, Volume 43, p. 2028–2033.

83) Yousif, S. T., Alsaffar, I. S. & Ahmed, S. M., 2010. Optimum Design of Singly and Doubly Reinforced Concrete Rectangular Beam Sections: Artificial Neural Networks Application. *Iraqi Journal of Civil Engineering,* June, 6(3), pp. 1-19.

84) Yousif, S. T. & Najem, R. M., 2012. Optimum cost design of reinforced concrete beams using genetic algorithm. *The Iraqi Journal For Mechanical And Material Engineering,* January, 12(04), pp. 680-693.

85) Zavadskas, E. K., Adeli, H., Antucheviciene, J. & Vilutiene, T., 2018. Sustainable Decision-Making in Civil Engineering, Construction and Building Technology. *Sustainability,* 10(14).

Appendix

Beam Problem Formulation MATLAB Codes:

```
function y = udhambeam(x)

x = udhambeammapvariables(x);

concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per  m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=(0.79/1000000); % embodied carbon emission of formwork per mm3

deadload=(x(1)*x(2)*25)/(1000000); %dead load in  N/mm
liveload=(20); %live load 20*10^6 N/mm
totalload=((1.35*deadload)+(1.5*liveload)); %total load on the beam
length=(6000); %length of beam 6000mm fixed
moment=((totalload*(length*length))/8); %maximum design moment on beam in Nmm
shearforce=((totalload*length)/2); %maximum shear design force in N
cover=(x(5)+10); % cover simplified in mm
effectivedepth=(x(2)-cover-(x(5)/2)-x(9)); %effective depth in mm
k=((moment)/((x(1)*effectivedepth*effectivedepth*x(3)))); %k to check if ↙
compression reinforcement is required

leverarm=0.5*effectivedepth*(1+(((1-(3.53*k)).^(1/2)))); %leverarm distance in mm
tensionreinforcement=((moment)/((x(4)/1.15)*leverarm)); % required tension ↙
reinforcement
minimumtensionreinforcement=((0.26*(0.30*((x(3))^(2/3))*x(1)*effectivedepth))/(x(4) ↙
/1.15)); %minimum reinforcement required
providedtensionreinforcement=(x(13)*3.14*((x(7)*x(7))/4)); %provided tension ↙
reinforcement
shearstress=((shearforce)/(0.9*effectivedepth*x(1))); %shear stress for check
resistanceshear=(0.36*(1-((x(3)/250)))*x(3))/(cotd(x(16))+tand(x(16)));
shearreinforcement=((shearforce*x(17))/(0.78*effectivedepth*(x(4)/1.15)*(cotd(x ↙
(16))))); %required shear reinforcement
providedshearreinforcement=(2*3.14*((x(10)*x(10))/4)); %provided shear ↙
reinforcement in mm
minimumshearreinforcement=(0.08*(x(3)^(0.5))*x(1)*x(17))/(x(4)/1.15); %minimum ↙
required steel reinforcement
maximumspacing=0.75*effectivedepth; %maximum possible spacing between shear links
nominalhangingbar=((2*3.14*8*8)/4);
linknumbers=(length/x(17));
totalsteel=(nominalhangingbar+(providedshearreinforcement*linknumbers) ↙
+providedtensionreinforcement);%total steel used
maximumsteel=((0.04*x(1)*x(2))-totalsteel); %maximum limit of  steel

%y(1)=(((x(1).*x(2))-totalsteel)*length*concretecost)+(totalsteel*length* ↙
(7850/1000000000)*steelcost)+(((2.*x(2))+x(1)).*length*formworkcost); %objective ↙
function for minimizing cost
y(1)=(((x(1).*x(2))-totalsteel)*length*concretecarbon)+(totalsteel*length* ↙
(7850/1000000000)*steelcarbon)+(((2.*x(2))+x(1)).*length*formworkcarbon); %objectie ↙
function for minimizing cost


end
```

Code 1: MATLAB code for beam objective function (Mathworks, 1984)

```
function [Cineq,Ceq] = udhambeamnonlcon(x)

x = udhambeammapvariables(x);

 deadload=(x(1)*x(2)*25)/(1000000); %dead load in  N/mm
 liveload=(20); %live load 20*10^6 N/mm
 totalload=((1.35*deadload)+(1.5*liveload)); %total load on the beam
 length=(6000); %length of beam 6000mm fixed
 moment=(((totalload*(length*length)))/8); %maximum design moment on beam in Nmm
 cover=(x(5)+10); % cover simplified in mm
 effectivedepth=(x(2)-cover-(x(5)/2)-x(9)); %effective depth in mm
 shearforce=((totalload*length)/2); %maximum shear design force in N
 k=(moment)/((x(1)*effectivedepth*effectivedepth*x(3)));    %k to check if compression ↙
 reinforcement is required

 %kdash=(0.168); %fixed condition

 leverarm=0.5*effectivedepth*(1+(((1-(3.53*k))^(1/2)))); %leverarm distance in mm
 tensionreinforcement=((moment)/((x(4)/1.15)*leverarm)); %required tension  ↙
 reinforcement
 providedtensionreinforcement=(x(13)*3.14*((x(7)*x(7))/4)); %provided tension  ↙
 reinforcement
 minimumtensionreinforcement=((0.26*(0.30*((x(3))^(2/3))*x(1)*effectivedepth))/(x(4)   ↙
 /1.15)); %minimum reinforcement required
 nominalhangingbar=((2*3.14*8*8)/4);
 shearreinforcement=((shearforce*x(17))/(0.78*effectivedepth*(x(4)/1.15)*(cotd(x   ↙
 (16))))); %required shear reinforcement
 providedshearreinforcement=(2*3.14*((x(10)*x(10))/4)); %provided shear  ↙
 reinforcement in mm
 minimumshearreinforcement=(0.08*(x(3)^(0.5))*x(1)*x(17))/(x(4)/1.15); %minimum  ↙
 required steel reinforcement
 linknumbers=(length/x(17));
 totalsteel=(nominalhangingbar+(providedshearreinforcement*linknumbers)  ↙
 +providedtensionreinforcement);%total steel used
 maximumsteel=((0.04*x(1)*x(2))-totalsteel); %maximum limit of steel
 maximumleverarm=(0.95*effectivedepth); %maximum lever arm
 minimumleverarm=(0.82*effectivedepth); %minimum leverarm

 Cineq =[k-0.168,leverarm-maximumleverarm,minimumleverarm-leverarm,  ↙
 minimumtensionreinforcement-tensionreinforcement,shearreinforcement-  ↙
 providedshearreinforcement,minimumshearreinforcement-providedshearreinforcement,  ↙
 tensionreinforcement-providedtensionreinforcement,totalsteel-maximumsteel,  ↙
 minimumtensionreinforcement-providedtensionreinforcement,(0.3*x(2))-x(1),x(1)-(0.5  ↙
 *x(2))];
 Ceq = [];

 end
```

Code 2: MATLAB code for beam constraints (Mathworks, 1984)

```
function x = udhambeammapvariables(x)

allX1 = [200,225,250,275,300,325,350,375,400,425,450,475,500];
allX2 = [500,525,550,575,600,625,650,675,700,725,750];
allX3 = [20,25,35,40,45,50];
allX4 = [500,550,600];
allX5 =  [12,14,16,20,25,28,32];
allX6 =  [12,14,16,20,25,28,32];
allX7 =  [12,14,16,20,25,28,32];
allX8 =  [12,14,16,20,25,28,32];
allX9 = [6,8,10,12,14];
allX10 = [6,8,10,12,14];
%allX16 =↙
[22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45];
allX17 =↙
[75,80,85,90,95,100,105,110,115,120,125,130,135,140,145,150,155,160,165,170,175,180  ↙
,185,190,195,200,205,210,215,220,225,230,235,240,245,250,255,260,265,270,275,280,28  ↙
5,290,295,300,305,310,315,320,325,330,335,340,345,350,355,360,365,370,375,380,385,3  ↙
90,395,400,405,410,415,420,425,430,435,440,445,450,455,460,465,470,475,480,485,490,  ↙
495,500,505,510,515,520,525,530,535,540,545,550,555,560,565,570,575,580,585,590,595  ↙
,600];

% The possible values for x(4) and x(6)
%allX4_6 = 45:5:60;

% Map x(3), x(4), x(5) and x(6) from the integer values used by GA to  the
% discrete values required.

x(1)  =  allX1(x(1));
x(2)  =  allX2(x(2));
x(3)  =  allX3(x(3));
x(4)  =  allX4(x(4));
x(5)  =  allX5(x(5));
x(6)  =  allX6(x(6));
x(7)  =  allX7(x(7));
x(8)  =  allX8(x(8));
x(9)  =  allX9(x(9));
x(10)  =  allX10(x(10));
x(17)  =  allX17(x(17));
end
```

Code 3: MATLAB code for beam variables (Mathworks, 1984)

Column problem formulation codes:

```
function y = udhamcolumn(x)

x = udhamcolumnmapvariables(x);

%x(1) = 200;
%x(2) = 200;
%x(3) = 20;
%x(4) = 500;
%x(5) = 12;
%x(6) = 12;
%x(7) = 12;
%x(8) = 12;
%x(9) = 6;
%x(10) = 6;
%x(11) = 2;
%x(12) = 2;
%x(13) = 75;
%x(14) = 300;

concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per  m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=(0.79/1000000); % embodied carbon emission of formwork per mm3


deadload=(x(1)*x(2)*25)/(10000); %dead load in KN/m
liveload=(500); %live load 500*10^3 N/mm
totalload=((1.35*deadload)+(1.5*liveload)); %total load on the column
length=(3000); %length of column 3000mm fixed
columnstiffness= (x(1)*x(2)*x(2)*x(2))/12;
beamstiffness= (200*550*550*550)/12; % beam dimensions
kone= ((1*columnstiffness)/3000)/((4*beamstiffness)/6000);
ktwo=  ((1*columnstiffness)/3000)/((4*beamstiffness)/6000);
effectivelength= 0.5*length*(((1+(kone/(0.45+kone))*(1+(ktwo/(0.45+ktwo)))))^ ↙
(1/2));
eone= x(2)/30;
etwo= length/400;
ethree= 20;
e=x(15);
%clear eone;
%clear etwo;
%clear ethree;
moment= totalload*e;
area= x(1)*x(2);
lambda= effectivelength/((columnstiffness/area)^(1/2));
a= 0.7;
b= 1.1;
c=0.7;
n= totalload/(area*(x(3)/1.5));
lambdalimit= (20*a*b*c)/(n^(1/2));
nominalcover= 40;
```

```
covertop= (nominalcover+x(10)+(x(8)/2));
coverbottom= (nominalcover+x(10)+(x(7)/2));
esc= (0.0035/x(14))*(x(14)-covertop);
effectivedepth= x(2)-coverbottom;
es= (0.0035/x(14))*(effectivedepth- x(14));
modulus= 200*1000;
fsc= modulus*esc;
fscforconcrete= fsc-(0.567*x(3));
fs=  x(4)/1.15;
s= 0.8*x(14);
%e1= (moment*1000000)/(totalload*1000);
topreinforcement= (((totalload*1000)*(e+(x(2)/2)-coverbottom))-(0.567*x(3)*x(1)*s*
(effectivedepth-(s/2))))/(fscforconcrete*(effectivedepth-covertop))
bottomreinforcement= ((totalload*1000)-(0.567*x(3)*x(1)*s)-
(fscforconcrete*topreinforcement))/fs
topprovidedreinforcement= (x(12)*3.14*x(8)*x(8))/4
bottomprovidedreinforcement=  (x(11)*3.14*x(7)*x(7))/4
totalsteel= (topprovidedreinforcement+bottomprovidedreinforcement);
areaofconcrete= (area-totalsteel);
minimumsteelarea= (0.10*totalload*1000)/(0.87*x(4));
maximumsteel= (0.04*areaofconcrete);


requiredsheardiaonearea= (3.14*x(8)*x(8))/4;
requiredsheardiatwoarea= (3.14*x(7)*x(7))/4;
requiredspacingone= 20*x(8);
requiredspacingtwo= 20*x(7);
requiredspacingthree= x(1);

%requiredsheardia= max(max(requiredsheardiaone),requiredsheardiatwo);
%requiredspacing= min(min(requiredspacingone,requiredspacingtwo),
requiredspacingthree);
%clear  requiredsheardiaone;
%clear  requiredsheardiatwo;
%clear  requiredspacingone;
%clear  requiredspacingtwo;
%clear requiredspacingthree;
providedshearreinforcementdiaarea=(3.14*x(10)*x(10))/4;
providedspacing= x(13);
providedshearreinforcement= 2*providedshearreinforcementdiaarea*
(length/providedspacing)

absolutetotalsteel=
topprovidedreinforcement+bottomprovidedreinforcement+providedshearreinforcement;

%y(1)=(((x(1).*x(2))-absolutetotalsteel)*length*concretecost)+
(absolutetotalsteel*length*(7850/1000000000)*steelcost)+(((2.*x(2))+x(1)).
*length*formworkcost); %objective function for minimizing cost
y(1)=(((x(1).*x(2))-absolutetotalsteel)*length*concretecarbon)+
(absolutetotalsteel*length*(7850/1000000000)*steelcarbon)+(((2.*x(2))+x(1)).
*length*formworkcarbon); %objectie function for minimizing cost

end
```

Code 4: MATLAB code for column objective function (Mathworks, 1984)

```
function [Cineq,Ceq] = udhamcolumnnonlcon(x)

x = udhamcolumnmapvariables(x);

%x(1) = 200;
%x(2) = 200;
%x(3) = 20;
%x(4) = 500;
%x(5) = 12;
%x(6) = 12;
%x(7) = 12;
%x(8) = 32;
%x(9) = 6;
%x(10) = 10;
%x(11) = 2;
%x(12) = 2;
%x(13) = 210;
%x(14) = 300;

deadload=(x(1)*x(2)*25)/(10000); %dead load in KN/m
liveload=(500); %live load 500*10^3 N/mm
totalload=((1.35*deadload)+(1.5*liveload)); %total load on the column
length=(3000); %length of column 3000mm fixed
columnstiffness= (x(1)*x(2)*x(2)*x(2))/12;
beamstiffness= (200*550*550*550)/12; % beam dimensions
kone= (((1*columnstiffness)/3000)/((4*beamstiffness)/6000));
ktwo=  (((1*columnstiffness)/3000)/((4*beamstiffness)/6000));
effectivelength= 0.5*length*(((1+(kone/(0.45+kone)))*(1+(ktwo/(0.45+ktwo))))^ ↙
(1/2));
eone= x(2)/30;
etwo= length/400;
ethree= 20;
e=x(15);
%clear eone;
%clear etwo;
%clear ethree;
moment= totalload*e;
area= x(1)*x(2);
lambda= effectivelength/((columnstiffness/area)^(1/2));
a= 0.7;
b= 1.1;
c= 0.7;
n= totalload/(area*(x(3)/1.5));
lambdalimit= (20*a*b*c)/(n^(1/2));
nominalcover= 40;
covertop= (nominalcover+x(10)+12);
coverbottom= (nominalcover+x(10)+x(7));
esc= (0.0035/x(14))*(x(14)-covertop);
effectivedepth= x(2)-coverbottom;
es= (0.0035/x(14))*(effectivedepth- x(14));
modulus= 200*1000;
fsc= modulus*esc;
fscforconcrete= fsc-(0.567*x(3));
```

```
fs=  x(4)/1.15;
s= 0.8*x(14);
topreinforcement= (((totalload*1000)*(e+(x(2)/2)-coverbottom))-(0.567*x(3)*x(1)*s*
(effectivedepth-(s/2))))/(fscforconcrete*(effectivedepth-covertop));
bottomreinforcement= ((totalload*1000)-(0.567*x(3)*x(1)*s)-
(fscforconcrete*topreinforcement))/fs;
topprovidedreinforcement= (x(12)*3.14*x(8)*x(8))/4;
bottomprovidedreinforcement=  (x(11)*3.14*x(7)*x(7))/4;
totalsteel= (topprovidedreinforcement+bottomprovidedreinforcement);
areaofconcrete= (area-totalsteel);
minimumsteelarea= (0.10*totalload*1000)/(0.87*x(4));
maximumsteel= (0.04*areaofconcrete);


requirequiredsheardiaonearea= (3.14*x(8)*x(8))/4;
requiredsheardiatwoarea= (3.14*x(7)*x(7))/4;
requiredspacingone= 20*x(8);
requiredspacingtwo= 20*x(7);
requiredspacingthree= x(1);

%requiredsheardia= max(max(requiredsheardiaone),requiredsheardiatwo);
%requiredspacing= min(min(requiredspacingone,requiredspacingtwo),
requiredspacingthree);
%clear  requiredsheardiaone;
%clear  requiredsheardiatwo;
%clear  requiredspacingone;
%clear  requiredspacingtwo;
%clear requiredspacingthree;
providedshearreinforcementdiaarea=(3.14*x(10)*x(10))/4;
providedspacing= x(13);
providedshearreinforcement= providedshearreinforcementdiaarea*
(length/providedspacing);

absolutetotalsteel=
topprovidedreinforcement+bottomprovidedreinforcement+providedshearreinforcement;


Cineq = [20-e,eone-e,etwo-e,ethree-e,topreinforcement-topprovidedreinforcement,
bottomreinforcement-bottomprovidedreinforcement,(0.002*areaofconcrete)-
minimumsteelarea,totalsteel-maximumsteel,lambda-lambdalimit,
requirequiredsheardiaonearea-providedshearreinforcementdiaarea,
requiredsheardiatwoarea-providedshearreinforcementdiaarea,providedspacing-
requiredspacingone,providedspacing-requiredspacingtwo,providedspacing-
requiredspacingthree];
Ceq = [];

end
```

Code 5: MATLAB code for column constraints (Mathworks, 1984)

```
function x = udhamcolumnmapvariables(x)

allX1 =
[200,225,250,275,300,325,350,375,400,425,450,475,500,525,550,575,600,625,650,675,70
0];  %cloumn breadth
allX2 =
[200,225,250,275,300,325,350,375,400,425,450,475,500,525,550,575,600,625,650,675,70
0];  %column depth
allX3 = [20,25,35,40,45,50]; %concrete strength
allX4 = [500,550,600];   %steel strength
allX5 = [12,14,16,20,25,28,32]; % diameter of bottom reinforcment
allX6 = [12,14,16,20,25,28,32]; % diameter of top reinforcment
allX7 = [12,14,16,20,25,28,32]; %provided diameter of bottom reinforcment
allX8 = [12,14,16,20,25,28,32]; %provided diameter of top reinforcment
allX9 = [6,8,10,12,14]; %diameter of shear links
allX10 = [6,8,10,12,14]; % provided diameter of shear links
allX11 = [2,3,4,5,6]; %number of bottom reinforcement bars
allX12 = [2,3,4,5,6]; %%number of top reinforcement bars
allX13 =
[75,80,85,90,95,100,105,110,115,120,125,130,135,140,145,150,155,160,165,170,175,180
,185,190,195,200,205,210,215,220,225,230,235,240,245,250,255,260,265,270,275,280,28
5,290,295,300,305,310,315,320,325,330,335,340,345,350,355,360,365,370,375,380,385,3
90,395,400,405,410,415,420,425,430,435,440,445,450,455,460,465,470,475,480,485,490,
495,500,505,510,515,520,525,530,535,540,545,550,555,560,565,570,575,580,585,590,595
,600,605,610,615,620,625,630,635,640,645,650,655,660,665,670,675,680,685,690,695,70
0,705,710,715,720,725,730,735,740,745,750,755,760,765,770,775,780,785,790,795,800];
%spacing
allX14 =
[150,155,160,165,170,175,180,185,190,195,200,205,210,215,220,225,230,235,240,245,25
0]; %depth of neutral axis
allX15 = [6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24];

x(1)  =  allX1(x(1));
x(2)  =  allX2(x(2));
x(3)  =  allX3(x(3));
x(4)  =  allX4(x(4));
x(5)  =  allX5(x(5));
x(6)  =  allX6(x(6));
x(7)  =  allX7(x(7));
x(8)  =  allX8(x(8));
x(9)  =  allX9(x(9));
x(10)  =  allX10(x(10));
x(11)  =  allX11(x(11));
x(12)  =  allX12(x(12));
x(13)  =  allX13(x(13));
x(14)  =  allX14(x(14));
x(15)  =  allX15(x(15));

end
```

Code 6: MATLAB code for column variables (Mathworks, 1984)

One way slab problem formulation codes:

```
function x = onewayslabmapvariables(x)

allX1 = [125,150,175,200,225,250,275,300,325,350,375,400];  %Depth of the slab
allX2 = [20,25,35,40,45,50]; %concrete strength
allX3 = [400,420,450,500];  %steel strength
allX4 = [10,12,14,16]; %provided diameter of bottom/tensile reinforcment
allX5 = [10,12,14,16]; %provided diameter of transverse reinforcment
allX6 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; %number of bottom/tensile reinforcement bars
allX7 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; %%number of transverse reinforcement bars

x(1) = allX1(x(1));
x(2) = allX2(x(2));
x(3) = allX3(x(3));
x(4) = allX4(x(4));
x(5) = allX5(x(5));
x(6) = allX6(x(6));
x(7) = allX7(x(7));
end
```

Code 7: MATLAB code for one way slab variables (Mathworks, 1984)

```
function y = onewayslab(x)

x = onewayslabmapvariables(x);
concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=(0.79/1000000); % embodied carbon emission of formwork per mm3

% Loading
density = 24; % concrete in KN/m2
finishingload = 1; %load in KN/m2
slabweight = x(1)*density/1000;
totaldeadload = slabweight+finishingload;
liveload = 3; % in KN/m2
ultimateload = (1.35*totaldeadload)+(1.5*liveload); % in KN/m2

% Moment and Shear Force
lengthinx = 5; % in m
lengthiny = 10; % in m
moment = (ultimateload*lengthinx*lengthinx)/8; %in KN-m
shearforce = (ultimateload*lengthinx)/2; % in KN

% Effective Depth
assumedbardia = 10; %in mm
cover = 25; % in mm
effectivedepth = x(1)-cover-assumedbardia/2; % in mm

% Flexure
breadth = 1000; % in mm
k = (moment*1000000)/(breadth*effectivedepth*effectivedepth*x(2));
kdash = 0.168;
leverarm = 0.5*effectivedepth*(1+(1-(3.53*k))^(1/2)); % in mm
maxleverarm = 0.95*effectivedepth;
 if leverarm <= maxleverarm
    z = leverarm;
 else
    z = maxleverarm;
 end
requiredsteel = (moment*1000000)/(0.87*x(3)*z); % Tensile steel
minimumsteel = (0.26*0.3*x(2)^(2/3)*breadth*effectivedepth)/x(3);
transverserequiredsteel = minimumsteel*0.2;
providedsteel = x(6)*3.14*x(4)*x(4)/4;
maximumsteel = 0.04*(breadth*x(1)-providedsteel);
```

```
% Shear Check
row = requiredsteel/(breadth*effectivedepth);
k1 = 1+((200/effectivedepth)^(1/2));
 if k1>2
    k1=2;
 else
    k1= 1+ ((200/effectivedepth)^(1/2));
 end
resistanceshear = (0.12*k1*((100*row*x(2))^(1/3))/1000)*breadth*effectivedepth;

% Deflection
k2 = 1; % for one way solid slab
rowzero = (x(2)^(1/2))/1000;
rowone = requiredsteel/(breadth*effectivedepth);
rowtwo = 0;
 if rowone<=rowzero
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/rowone)))+(3.2*(x(2)^(1/2))*(((rowzero/rowone)-1)^(3/2))));
 else
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/(rowone-rowtwo))))+((1/12)*(x(2)^(1/2))*((rowtwo/rowzero)^(1/2))));
 end
f1 = 1;
f2 = 1; % span less than7m
f3 = (500*providedsteel)/(x(3)*requiredsteel);
basicspantodepthratio = spantodepthratio*f1*f2*f3;
actualspantodepthratio = lengthinx*1000/effectivedepth;
transversesteel = (x(7)*3.14*x(5)*x(5))/4;
totalsteel = transversesteel+providedsteel; % is mm2 per m of breadth

% Objective function
y(1)=(((x(1)*breadth*10)-((providedsteel*10)+(transversesteel*5)))*lengthinx*1000*concretecost)+(((providedsteel*5000*10)
+(transversesteel*10000*5))*(7850/1000000000)*steelcost)+(lengthiny*lengthinx*1000000*formworkcost); %objective function for minimizing cost
%y(1)=(((x(1)*breadth*10)-((providedsteel*10)+(transversesteel*5)))*lengthinx*1000*concretecarbon)+(((providedsteel*5000*10)
+(transversesteel*10000*5))*(7850/1000000000)*steelcarbon)+(lengthiny*lengthinx*1000000*formworkcarbon); %objectie function for minimizing carbon

end
```

Code 8: MATLAB code for one way slab (Mathworks, 1984)

```
function [Cineq,Ceq] = onewayslabcon(x)

x = onewayslabmapvariables(x);
concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=(0.79/1000000); % embodied carbon emission of formwork per mm3

% Loading
density = 24; % concrete in KN/m2
finishingload = 1; %load in KN/m2
slabweight = x(1)*density/1000
totaldeadload = slabweight+finishingload;
liveload = 3; % in KN/m2
ultimateload = (1.35*totaldeadload)+(1.5*liveload); % in KN/m2

% Moment and Shear Force
lengthinx = 5; % in m
lengthiny = 10; % in m
moment = (ultimateload*lengthinx*lengthinx)/8; %in KN-m
shearforce = ultimateload*lengthinx/2; % in KN

% Effective Depth
assumedbardia = 10; %in mm
cover = 25; % in mm
effectivedepth= x(1)-cover-(assumedbardia/2); % in mm
axisdistance = cover+assumedbardia/2;
```

```
% Flexure
breadth = 1000; % in mm
k = (moment*1000000)/(breadth*effectivedepth*effectivedepth*x(2));
kdash = 0.168;
leverarm = 0.5*effectivedepth*(1+(1-(3.53*k))^(1/2)); % in mm
maxleverarm = 0.95*effectivedepth;
 if leverarm <= maxleverarm
    z = leverarm;
 else
    z = maxleverarm;
 end
requiredsteel = (moment*1000000)/(0.87*x(3)*z); % Tensile steel
minimumsteel = (0.26*0.3*x(2)^(2/3)*breadth*effectivedepth)/x(3);
transverserequiredsteel = minimumsteel*0.2;
providedsteel = x(6)*3.14*x(4)*x(4)/4;
maxspacingmain = 1000/x(6);
maximumsteel = 0.04*(breadth*x(1)-providedsteel);

% Shear Check
row = requiredsteel/(breadth*effectivedepth);
k1 = 1+((200/effectivedepth)^(1/2));
 if k1>2
    k1=2;
 else
    k1= 1+ ((200/effectivedepth)^(1/2));
 end
resistanceshear = (0.12*k1*((100*row*x(2))^(1/3))/1000)*breadth*effectivedepth;
```

```
% Deflection
k2 = 1; % for one way solid slab
rowzero = (x(2)^(1/2))/1000;
rowone = requiredsteel/(breadth*effectivedepth);
rowtwo = 0;
 if rowone<=rowzero
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/rowone)))+(3.2*(x(2)^(1/2))*(((rowzero/rowone)-1)^(3/2))));
 else
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/(rowone-rowtwo))))+((1/12)*(x(2)^(1/2))*((rowtwo/rowzero)^(1/2))));
 end
f1 = 1;
f2 = 1; % span less than7m
f3 = (500*providedsteel)/(x(3)*requiredsteel);
basicspantodepthratio = spantodepthratio*f1*f2*f3;
actualspantodepthratio = lengthinx*1000/effectivedepth;
transversesteel = x(7)*3.14*x(5)*x(5)/4;
maxspacingtransverse = 1000/x(7);
totalsteel = transversesteel+providedsteel; % is mm2 per m of breadth
```

```
% Spacing
if x(1)<=200
    maxspacing = 3*x(1);
    if maxspacing>400
        maxspacing = 400;
    else
        maxspacing = 3*x(1);
    end
else
    fs = (x(3)/1.15)*((totaldeadload+0.3*liveload)/(ultimateload))*(requiredsteel/providedsteel);
    if fs<=160
        maxspacing = 300;
    elseif 160<fs <= 200
        maxspacing = 250;
    elseif 200<fs<=240
        maxspacing = 200;
    elseif 240<fs<=280
        maxspacing = 150;
    elseif 280<fs<=320
        maxspacing = 100;
    elseif fs>320
        maxspacing = 50;
    end
end

% Constraints
Cineq = [80-x(1),20-axisdistance,k-kdash,requiredsteel-providedsteel,minimumsteel-requiredsteel,
    providedsteel-maximumsteel,row-0.02,shearforce-resistanceshear,f3-1.5,1-f3,
    actualspantodepthratio-basicspantodepthratio,transverserequiredsteel-transversesteel,k1-2,
    170-effectivedepth,maxspacingtransverse-maxspacing,maxspacingmain-maxspacing];
Ceq = [];
end
```

Code 9: MATLAB code for one way slab constraints (Mathworks, 1984)

Ribbed slab problem formulation codes:

```
function x = ribbedslabmapvariables(x)

allX1 = [125,150,175,200,225,250,275,300,325,350,375,400];  %Depth of the slab
allX2 = [20,25,35,40,45,50]; %concrete strength
allX3 = [500,550,600];  %steel strength
allX4 = [12,14,16,20,25,28,32]; %provided diameter of bottom/tensile reinforcment
allX5 = [12,14,16,20,25,28,32]; %provided diameter of transverse reinforcment
allX6 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; %number of bottom/tensile reinforcement bars
allX7 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; %%number of transverse reinforcement bars
allX8 = [100,125,150,175,200,225,250]; % Rib breadth
allX9 = [300,350,400,450,500,550,600,650,700,750,800,850,900,950,1000,1050,1100,1150,1200,1250,1300,1350,1400,1450,1500];
allX10 = [50,60,70,80,90,100]; % top depth
allX11 = [98,142,193,252,393]; % top steel mesh area

x(1) = allX1(x(1));
x(2) = allX2(x(2));
x(3) = allX3(x(3));
x(4) = allX4(x(4));
x(5) = allX5(x(5));
x(6) = allX6(x(6));
x(7) = allX7(x(7));
x(8) = allX8(x(8));
x(9) = allX9(x(9));
x(10) = allX10(x(10));
x(11) = allX11(x(11));

end
```

Code 10: Variable code for ribbed slab (Mathworks, 1984)

```
function y = ribbedslab(x)

x = ribbedslabmapvariables(x)

% Cost and Carbon data
concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=0.79; % embodied carbon emission of formwork per mm3

% Loading
density = 24; % concrete in KN/m2
finishingload = (1.2*x(9))/1000; %load in KN/m
ribweight = (x(8)*x(1)*25)/1000000; %load in KN/m
topweight = (x(10)*density*x(9))/1000000; %load in KN/m
claypot = 0.7; %load in KN/m
partitionload = (1.5*x(9))/1000; %load in KN/m
totaldeadload = finishingload+ribweight+topweight+claypot+partitionload; %load in KN/m
liveload = (2.5*x(9))/1000; % in KN/m
ultimateload = (1.35*totaldeadload)+(1.5*liveload); % in KN/m

% Moment and Shear Force
lengthinx = 5; % in m
lengthiny = 7; % in m
moment = (ultimateload*lengthinx*lengthinx)/8; %in KN-m
shearforce = (ultimateload*lengthinx)/2; % in KN
```

```
% Effective breadth
b1 = (x(9)-x(8))/2;
lo = 0.85*lengthinx*1000;
effectivebreadth1 = (0.2*b1)+(0.1*lo);
effectivebreadth2 = 0.2*lo;
effectivebreadth3 = b1;
bmin = [effectivebreadth1,effectivebreadth2,effectivebreadth3];
effectivebreadth = min(bmin);
b = x(8)+effectivebreadth+effectivebreadth;

% Effective Depth
assumedbardia = 10; %in mm
assumedbardialink = 8; %in mm
cover = 25; % in mm
effectivedepth = x(1)-cover-(assumedbardia/2)-assumedbardialink; % in mm

% Flexure
k = (moment*1000000)/(b*effectivedepth*effectivedepth*x(2));
kdash = 0.168;
leverarm = 0.5*effectivedepth*(1+(1-(3.53*k))^(1/2)); % in mm
maxleverarm = 0.95*effectivedepth;
 if leverarm <= maxleverarm
     z = leverarm;
 else
     z = maxleverarm;
 end
neutralaxisdepth = 2.5*(effectivedepth-z);
requiredsteel = (moment*1000000)/(0.87*x(3)*z);
minimumsteel = (0.26*0.3*x(2)^(2/3)*b*effectivedepth)/x(3);
providedsteel = x(6)*3.14*x(4)*x(4)/4,
maximumsteel = 0.04*(b*x(1)-providedsteel);
```

```
% Top Mesh
toparea = (0.13*1000*x(10))/100, % mm2/m
topprovidedarea = x(11),

% Shear Check
row = requiredsteel/(x(8)*effectivedepth);
k1 = 1+((200/effectivedepth)^(1/2));
 if k1>2
    k1=2;
 else
    k1= 1+ ((200/effectivedepth)^(1/2));
 end
resistanceshear = ((0.12*k1*((100*row*x(2))^(1/3))/1000)*x(8)*effectivedepth),


% Deflection
k2 = 1; % for one way solid slab
rowzero = (x(2)^(1/2))/1000;
rowone = requiredsteel/(x(9)*effectivedepth);
rowtwo = 0;
 if rowone<=rowzero
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/rowone)))+(3.2*(x(2)^(1/2))
    *(((rowzero/rowone)-1)^(3/2))));
 else
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/(rowone-rowtwo))))+((1/12)*
    (x(2)^(1/2))*((rowtwo/rowzero)^(1/2))));
 end
f1 = 1;
f2 = 1; % span less than7m
f3 = (500*providedsteel)/(x(3)*requiredsteel);
basicspantodepthratio = spantodepthratio*f1*f2*f3;
actualspantodepthratio = (lengthinx*1000)/effectivedepth;
```

```
% Areas
mesharea = (topprovidedarea*lengthinx)+(topprovidedarea*lengthiny);
reinforcementarea = providedsteel*lengthinx*(lengthiny*1000/x(9));
totalsteel = mesharea+reinforcementarea;
concreteareatop = ((lengthinx*lengthiny*1000000)-mesharea)*x(10);
concretearearib = ((x(8)*lengthinx*1000*(lengthiny*1000/x(9)))-reinforcementarea)*(x(1)-x(10));
totalconcrete = concreteareatop+concretearearib;
formworkarea = ((x(8)+(2*(x(1)-x(10))))*(lengthiny*1000/x(9)))+(((lengthiny*1000)-
((lengthiny*1000/x(9))*x(8)))*(lengthinx*1000));
formworkareacarbon = (((x(8)+(2*(x(1)-x(10))))*(lengthiny*1000/x(9)))*(x(1)-x(10)))
+((((lengthiny*1000)-((lengthiny*1000/x(9))*x(8)))*(lengthinx*1000))*x(8));

% Objective Function

y(1)=(totalconcrete*concretecost)+(totalsteel*(7850/1000000)*steelcost)+
(formworkarea*formworkcost); %objective function for minimizing cost
y(1)=(totalconcrete*concretecarbon)+(totalsteel*(7850/1000000)*steelcarbon)+
((((formworkareacarbon/1000000000)*2710*formworkcarbon)/300)); %objectie function for minimizing carbon

end
```

Code 11: Main code for ribbed slab (Mathworks, 1984)

```
function [x,fval,exitflag,output,population,score] = untitled(nvars,lb,ub,intcon)
%% This is an auto generated MATLAB file from Optimization Tool.

%% Start with the default options
options = optimoptions('ga');
%% Modify options setting
options = optimoptions(options,'Display', 'off');
[x,fval,exitflag,output,population,score] = ...
ga(@ribbedslab,nvars,[],[],[],[],lb,ub,@ribbedslabcon,intcon,options);
```

Code 12: Genetic algorithm function (Mathworks, 1984)

```
function [Cineq,Ceq] = ribbedslabcon(x)

x = ribbedslabmapvariables(x);

% Cost and Carbon data
concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=0.79; % embodied carbon emission of formwork per mm3

% Loading
density = 24; % concrete in KN/m2
finishingload = (1.2*x(9))/1000; %load in KN/m
ribweight = (x(8)*x(1)*25)/1000000; %load in KN/m
topweight = (x(10)*density*x(9))/1000000; %load in KN/m
claypot = 0.7; %load in KN/m
partitionload = (1.5*x(9))/1000; %load in KN/m
totaldeadload = finishingload+ribweight+topweight+claypot+partitionload; %load in KN/m
liveload = (2.5*x(9))/1000; % in KN/m
ultimateload = (1.35*totaldeadload)+(1.5*liveload); % in KN/m

% Moment and Shear Force
lengthinx = 5; % in m
lengthiny = 7; % in m
moment = (ultimateload*lengthinx*lengthinx)/8; %in KN-m
shearforce = (ultimateload*lengthinx)/2; % in KN
```

```
% Effective breadth
b1 = (x(9)-x(8))/2;
lo = 0.85*lengthinx*1000;
effectivebreadth1 = (0.2*b1)+(0.1*lo);
effectivebreadth2 = 0.2*lo;
effectivebreadth3 = b1;
bmin = [effectivebreadth1,effectivebreadth2,effectivebreadth3];
effectivebreadth = min(bmin);
b = x(8)+effectivebreadth+effectivebreadth;

% Effective Depth
assumedbardia = 10; %in mm
assumedbardialink = 8; %in mm
cover = 25; % in mm
effectivedepth = x(1)-cover-(assumedbardia/2)-assumedbardialink; % in mm
axisdistance = cover+assumedbardia/2;

% Flexure
k = (moment*1000000)/(b*effectivedepth*effectivedepth*x(2));
kdash = 0.168;
leverarm = 0.5*effectivedepth*(1+(1-(3.53*k))^(1/2)); % in mm
maxleverarm = 0.95*effectivedepth;
 if leverarm <= maxleverarm
    z = leverarm;
 else
    z = maxleverarm;
 end
neutralaxisdepth = 2.5*(effectivedepth-z);
requiredneutralaxis = 1.25*x(10);
requiredsteel = (moment*1000000)/(0.87*x(3)*z); % Tensile steel
minimumsteel = (0.26*0.3*x(2)^(2/3)*b*effectivedepth)/x(3);
providedsteel = x(6)*3.14*x(4)*x(4)/4;
maximumsteel = 0.04*(b*x(1)-providedsteel);
```

```
% Top Mesh
toparea = (0.13*1000*x(10))/100; % mm2/m
topprovidedarea = x(11);
totaltoparea = (topprovidedarea*lengthinx)+(topprovidedarea*lengthiny);

% Shear Check
row = requiredsteel/(x(8)*effectivedepth);
k1 = 1+((200/effectivedepth)^(1/2));
 if k1>2
    k1=2;
 else
    k1= 1+ ((200/effectivedepth)^(1/2));
 end
resistanceshear = ((0.12*k1*((100*row*x(2))^(1/3))/1000)*x(8)*effectivedepth);
```

```
% Deflection
k2 = 1; % for one way solid slab
rowzero = (x(2)^(1/2))/1000;
rowone = requiredsteel/(x(9)*effectivedepth);
rowtwo = 0;
 if rowone<=rowzero
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/rowone)))+(3.2*(x(2)^(1/2))
    *(((rowzero/rowone)-1)^(3/2))));
 else
    spantodepthratio = k2*(11+(1.5*((x(2)^(1/2))*(rowzero/(rowone-rowtwo))))+((1/12)
    *(x(2)^(1/2))*((rowtwo/rowzero)^(1/2))));
 end
f1 = 1;
f2 = 1; % span less than7m
f3 = (500*providedsteel)/(x(3)*requiredsteel);
basicspantodepthratio = spantodepthratio*f1*f2*f3;
actualspantodepthratio = (lengthinx*1000)/effectivedepth;
ribdepth = x(1)-x(10);
allowableribdepth = 4*x(8);
flangedepth1 = 50;
flangedepth2 = x(9)/10;
flangedepth = [flangedepth1,flangedepth2];
allowableflangedepth = max(flangedepth);

% Constraints
Cineq = [100-x(8),ribdepth-allowableribdepth,allowableflangedepth-x(10),25-axisdistance
    ,k-kdash,neutralaxisdepth-requiredneutralaxis,requiredsteel-providedsteel,
    minimumsteel-requiredsteel,providedsteel-maximumsteel,toparea-topprovidedarea,
    row-0.02,shearforce-resistanceshear,rowone-rowzero,f3-1.5,1-f3,
    actualspantodepthratio-basicspantodepthratio,k1-2];
Ceq = [];

end
```

Code 14: Constraints code for ribbed slab (Mathworks, 1984)

Flat slab problem formulation codes:

```
function x = flatslabmapvariables(x)

allX1 = [125,150,175,200,225,250,275,300,325,350,375,400];  %Depth of the slab
allX2 = [80,85,90,95,100,105,110,115,120,125,130,135,140,145,150];  %Depth of the drop panel
allX3 = [2200,2300,2400,2500,2600,2700,2800,2900,3000,3100,3200];  % dimension of drop panel
allX4 = [1000,1100,1200,1300,1400,1500];  % dimension of column head
allX5 = [20,25,35,40,45,50]; %concrete strength
allX6 = [500,550,600];  %steel strength
allX7 = [12,14,16,20,25,28,32]; %provided diameter at middle strip of center
allX8 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % Number of bars at middle strip of center
allX9 = [12,14,16,20,25,28,32]; %provided diameter at column strip of center
allX10 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % Number of bars at column strip of center
allX11 = [12,14,16,20,25,28,32]; %provided diameter at middle strip of interior span
allX12 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % Number of bars at middle strip of interior span
allX13 = [12,14,16,20,25,28,32]; %provided diameter at column strip of interior span
allX14 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % Number of bars at column strip of interior span


x(1) = allX1(x(1));
x(2) = allX2(x(2));
x(3) = allX3(x(3));
x(4) = allX4(x(4));
x(5) = allX5(x(5));
x(6) = allX6(x(6));
x(7) = allX7(x(7));
x(8) = allX8(x(8));
x(9) = allX9(x(9));
x(10) = allX10(x(10));
x(11) = allX11(x(11));
x(12) = allX12(x(12));
x(13) = allX13(x(13));
x(14) = allX14(x(14));


end
```

Code 15: Variable code for flat slab (Mathworks, 1984)

```
function y = flatslab(x)

x = flatslabmapvariables(x)

% Cost and Carbon data
concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=0.79; % embodied carbon emission of formwork per mm3

% Loading
density = 25; % concrete in KN/m2
slabweight = (x(1)*density)/1000;
droppanelweight = (x(2)*density)/1000;
finishingload = 1; %load in KN/m2
totaldeadload = slabweight+droppanelweight+finishingload; %load in KN/m
liveload = 5; % in KN/m
lx = 6.5;
ly = 6.5;
bayarea = lx*ly;
loadratio = totaldeadload/liveload;
ultimateload = (1.35*totaldeadload)+(1.5*liveload); % in KN/m
f = ultimateload*lx*lx;

% Effective depth and Span
cover = 25;
assumedbardia = 10; %in mm
spaneffectivedepth = x(1)-cover-(assumedbardia/2); % in mm
supporteffectivedepth = spaneffectivedepth+x(2)
effectivespan = (lx-(x(4)/1000))+(supporteffectivedepth/1000);
columnstrip = x(3)/1000;
middlestrip = lx-columnstrip
```

```
% Flexure reinforcement in X-direction
% At center of interior span
momentatcenter = 0.0063*effectivespan*f;
cmmoment = (0.45*middlestrip/(lx/2))*momentatcenter;
ccmoment = momentatcenter-cmmoment;

% At middle strip in the center of interior span
kcm = (cmmoment*1000000)/(middlestrip*1000*spaneffectivedepth
*spaneffectivedepth*x(5));
kdashcm = 0.168;
leverarmcm = 0.5*spaneffectivedepth*(1+(1-(3.53*kcm))^(1/2)); % in mm
maxleverarmcm = 0.95*spaneffectivedepth;
 if leverarmcm<= maxleverarmcm
    zcm = leverarmcm;
 else
    zcm = maxleverarmcm;
 end

cmrequiredsteel = (cmmoment*1000000)/(0.87*x(6)*zcm);
cmminimumsteel = (0.26*0.3*x(5)^(2/3)*middlestrip*1000*spaneffectivedepth)/x(6);
cmprovidedsteel = x(8)*3.14*x(7)*x(7)/4;
cmmaximumsteel = 0.04*(middlestrip*1000*x(1)-cmprovidedsteel);

% At column strip in the center of interior span
kcc = (ccmoment*1000000)/(columnstrip*1000*supporteffectivedepth*supporteffectivedepth*x(5));
kdashcc = 0.168;
leverarmcc = 0.5*supporteffectivedepth*(1+(1-(3.53*kcc))^(1/2)); % in mm
maxleverarmcc = 0.95*supporteffectivedepth;
 if leverarmcc<= maxleverarmcc
    zcc = leverarmcc;
 else
    zcc = maxleverarmcc;
 end

ccrequiredsteel = (ccmoment*1000000)/(0.87*x(6)*zcc);
ccminimumsteel = (0.26*0.3*x(5)^(2/3)*columnstrip*1000*supporteffectivedepth)/x(6);
ccprovidedsteel = x(10)*3.14*x(9)*x(9)/4;
ccmaximumsteel = 0.04*(columnstrip*1000*x(1)-ccprovidedsteel);
```

```
% At interior span
momentatinterior = 0.0063*effectivespan*f;
immoment = (0.25*middlestrip/(lx/2))*momentatcenter;
icmoment = momentatinterior-immoment;

% At middle strip in the interior span
kim = (immoment*1000000)/(middlestrip*1000*spaneffectivedepth*spaneffectivedepth*x(5));
kdashim = 0.168;
leverarmim = 0.5*spaneffectivedepth*(1+(1-(3.53*kim))^(1/2)); % in mm
maxleverarmim = 0.95*spaneffectivedepth;
 if leverarmim<= maxleverarmim
    zim = leverarmim;
 else
    zim = maxleverarmim;
 end

imrequiredsteel = (immoment*1000000)/(0.87*x(6)*zim);
imminimumsteel = (0.26*0.3*x(5)^(2/3)*middlestrip*1000*spaneffectivedepth)/x(6);
improvidedsteel = x(12)*3.14*x(11)*x(11)/4;
immaximumsteel = 0.04*(middlestrip*1000*x(1)-improvidedsteel);

% At middle strip in the interior span
kic = (icmoment*1000000)/(columnstrip*1000*supporteffectivedepth*supporteffectivedepth*x(5));
kdashic = 0.168;
leverarmic = 0.5*supporteffectivedepth*(1+(1-(3.53*kic))^(1/2)); % in mm
maxleverarmic = 0.95*supporteffectivedepth;
 if leverarmic<= maxleverarmic
    zic = leverarmic;
 else
    zic = maxleverarmic;
 end

icrequiredsteel = (icmoment*1000000)/(0.87*x(6)*zic);
icminimumsteel = (0.26*0.3*x(5)^(2/3)*columnstrip*1000*supporteffectivedepth)/x(6);
icprovidedsteel = x(14)*3.14*x(13)*x(13)/4;
icmaximumsteel = 0.04*(columnstrip*1000*x(1)-icprovidedsteel);
```

```
% Punching Shear
% At Column Head
beta = 1.15;
uo = 3.14*x(4);
chshear = f-(3.14*(x(4)/1000)*(x(4)/1000)*ultimateload);
cheffectiveshear = 1.15*chshear;
chmaxresistanceshear = 0.5*uo*supporteffectivedepth*(0.6*(1-(x(5)/250)))*(x(5)/1.5)*(1/1000);

% At Control Perimeter
cpsectiondia = (x(4)/1000)+(4*supporteffectivedepth/1000);
u1 = 3.14*cpsectiondia*1000;
cpshear = f-(3.14*cpsectiondia*cpsectiondia*ultimateload/4);
cpeffectiveshear = 1.15*cpshear;
cprow = (icprovidedsteel/(supporteffectivedepth*x(3)))
cpk = 1+((200/supporteffectivedepth)^0.5)
cpresistanceshear = (0.12*cpk*((100*cprow*x(5))^(1/3)))*u1*supporteffectivedepth/1000

% At Drop Panel
dpperimeter = 2*spaneffectivedepth
dpu = (2*x(3))+(2*x(3))+(2*3.14*2*dpperimeter)
dparea = (((x(3)/1000)+(4*spaneffectivedepth/1000))^(2))-((4-3.14)*((2*spaneffectivedepth/1000)^(2)))
dpshear = f-(dparea*ultimateload)
dpeffectiveshear = 1.15*dpshear
dprow = (icprovidedsteel/(spaneffectivedepth*x(3)))
dpk = 1+((200/spaneffectivedepth)^0.5)
dpresistanceshear = (0.12*dpk*((100*dprow*x(5))^(1/3)))*dpu*spaneffectivedepth/1000
```

```
% Deflection Check
% At Middle Strip

msk = 1.2; % for one way solid slab
msrowzero = (x(5)^(1/2))/1000;
msrowone = cmrequiredsteel/(middlestrip*spaneffectivedepth);
msrowtwo = imrequiredsteel/(middlestrip*spaneffectivedepth);
 if msrowone<=msrowzero
    msspantodepthratio = msk*(11+(1.5*((x(5)^(1/2))
    *(msrowzero/msrowone))+(3.2*(x(5)^(1/2))*(((msrowzero/msrowone)-1)^(3/2))));
     csf3 = (500*cmprovidedsteel)/(x(6)*cmrequiredsteel);
 else
    msspantodepthratio = msk*(11+(1.5*((x(5)^(1/2))
    *(msrowzero/(msrowone-msrowtwo))))+((1/12)*(x(5)^(1/2))*((msrowtwo/msrowzero)^(1/2))));
    csf3 = (500*improvidedsteel)/(x(6)*imrequiredsteel);
 end
msf1 = 1;
msf2 = 1; % span less than7m
msf3 = (500*cmprovidedsteel)/(x(6)*cmrequiredsteel);
msbasicspantodepthratio = msspantodepthratio*msf1*msf2*msf3;
msactualspantodepthratio = (lx*1000)/spaneffectivedepth;
```

```
% At Column Strip

csk = 1.2; % for one way solid slab
csrowzero = (x(5)^(1/2))/1000;
csrowone = ccrequiredsteel/(columnstrip*supporteffectivedepth);
csrowtwo = icrequiredsteel/(columnstrip*supporteffectivedepth);
 if csrowone<=csrowzero
    csspantodepthratio = csk*(11+(1.5*((x(5)^(1/2))*(csrowzero/csrowone)))+(3.2*(x(5)^(1/2))
    *(((csrowzero/csrowone)-1)^(3/2))));
    csf3 = (500*ccprovidedsteel)/(x(6)*ccrequiredsteel);
 else
    csspantodepthratio = csk*(11+(1.5*((x(5)^(1/2))*(csrowzero/(csrowone-csrowtwo))))
    +((1/12)*(x(5)^(1/2))*((csrowtwo/csrowzero)^(1/2))));
    csf3 = (500*icprovidedsteel)/(x(6)*icrequiredsteel);
 end
csf1 = 1;
csf2 = 1; % span less than7m
csbasicspantodepthratio = csspantodepthratio*csf1*csf2*csf3;
csactualspantodepthratio = (lx*1000)/supporteffectivedepth;

% Total Area
steelarea = cmprovidedsteel+ccprovidedsteel+improvidedsteel+icprovidedsteel
concrete = (((lx*1000)*(ly*1000))-steelarea)*x(1)
formwork = (lx*1000)*(ly*1000)
steel = ((cmprovidedsteel+improvidedsteel)*middlestrip)+((ccprovidedsteel+icprovidedsteel)*columnstrip)

% Objective Function
y(1)= 2*((concrete*concretecost)+(steel*(7850/1000000)*steelcost)
+(formwork*formworkcost)) %objective function for minimizing cost
y(1)= 2*((concrete*concretecarbon)+(steel*(7850/1000000)*steelcarbon)
+((((formwork/1000000000)*2710*x(1)*formworkcarbon)/300))) %objectie function for minimizing carbon

end
```

Code 16: Main code for flat slab (Mathworks, 1984)

```
function [Cineq,Ceq] = flatslabcon(x)

x = flatslabmapvariables(x)

% Cost and Carbon data
concretecost=(110/1000000000); % cost of concrete per mm3
steelcost=(0.8); % cost of steel per Kgs
formworkcost=(6/1000000); % cost of formwork per m2
concretecarbon=(338/1000000000); % embodied carbon emission of concrete per mm3
steelcarbon=(0.87); % embodied carbon emission of steel per mm2
formworkcarbon=0.79; % embodied carbon emission of formwork per mm3

% Loading
density = 25; % concrete in KN/m2
slabweight = (x(1)*density)/1000;
droppanelweight = (x(2)*density)/1000;
finishingload = 1; %load in KN/m2
totaldeadload = slabweight+droppanelweight+finishingload; %load in KN/m
liveload = 5; % in KN/m
lx = 6.5;
ly = 6.5;
bayarea = lx*ly;
loadratio = totaldeadload/liveload;
ultimateload = (1.35*totaldeadload)+(1.5*liveload); % in KN/m
f = ultimateload*lx*lx;

% Effective depth and Span
cover = 25;
assumedbardia = 10; %in mm
linkdia = 8;
axisdistance = cover+(assumedbardia/2)+linkdia;
spaneffectivedepth = x(1)-cover-(assumedbardia/2); % in mm
supporteffectivedepth = spaneffectivedepth+x(2)
effectivespan = (lx-(x(4)/1000))+(supporteffectivedepth/1000);
requiredcolumnstrip = lx/3;
columnstrip = x(3)/1000;
requiredmiddlestrip = lx/2;
middlestrip = lx-columnstrip;
% At center of interior span
momentatcenter = 0.0063*effectivespan*f;
cmmoment = (0.45*middlestrip/(lx/2))*momentatcenter;
ccmoment = momentatcenter-cmmoment;

% At middle strip in the center of interior span
kcm = (cmmoment*1000000)/(middlestrip*1000*spaneffectivedepth
*spaneffectivedepth*x(5));
kdashcm = 0.168;
leverarmcm = 0.5*spaneffectivedepth*(1+(1-(3.53*kcm))^(1/2)); % in mm
maxleverarmcm = 0.95*spaneffectivedepth;
 if leverarmcm<= maxleverarmcm
    zcm = leverarmcm;
 else
    zcm = maxleverarmcm;
 end

cmrequiredsteel = (cmmoment*1000000)/(0.87*x(6)*zcm);
cmminimumsteel = (0.26*0.3*x(5)^(2/3)*middlestrip*1000*spaneffectivedepth)/x(6);
cmprovidedsteel = x(8)*3.14*x(7)*x(7)/4;
cmmaximumsteel = 0.04*(middlestrip*1000*x(1)-cmprovidedsteel);

% At column strip in the center of interior span
kcc = (ccmoment*1000000)/(columnstrip*1000*supporteffectivedepth
*supporteffectivedepth*x(5));
kdashcc = 0.168;
leverarmcc = 0.5*supporteffectivedepth*(1+(1-(3.53*kcc))^(1/2)); % in mm
maxleverarmcc = 0.95*supporteffectivedepth;
 if leverarmcc<= maxleverarmcc
    zcc = leverarmcc;
 else
    zcc = maxleverarmcc;
 end

ccrequiredsteel = (ccmoment*1000000)/(0.87*x(6)*zcc);
ccminimumsteel = (0.26*0.3*x(5)^(2/3)*columnstrip*1000
*supporteffectivedepth)/x(6);
ccprovidedsteel = x(10)*3.14*x(9)*x(9)/4;
ccmaximumsteel = 0.04*(columnstrip*1000*x(1)-ccprovidedsteel);
```

```
% At interior span
momentatinterior = 0.0063*effectivespan*f;
immoment = (0.25*middlestrip/(lx/2))*momentatcenter;
icmoment = momentatinterior-immoment;

% At middle strip in the interior span
kim = (immoment*1000000)/(middlestrip*1000*spaneffectivedepth
*spaneffectivedepth*x(5));
kdashim = 0.168;
leverarmim = 0.5*spaneffectivedepth*(1+(1-(3.53*kim))^(1/2)); % in mm
maxleverarmim = 0.95*spaneffectivedepth;
 if leverarmim<= maxleverarmim
    zim = leverarmim;
 else
    zim = maxleverarmim;
 end

imrequiredsteel = (immoment*1000000)/(0.87*x(6)*zim);
imminimumsteel = (0.26*0.3*x(5)^(2/3)*middlestrip*1000
*spaneffectivedepth)/x(6);
improvidedsteel = x(12)*3.14*x(11)*x(11)/4;
immaximumsteel = 0.04*(middlestrip*1000*x(1)-improvidedsteel);
```

```
% At middle strip in the interior span
kic = (icmoment*1000000)/(columnstrip*1000*supporteffectivedepth
*supporteffectivedepth*x(5));
kdashic = 0.168;
leverarmic = 0.5*supporteffectivedepth*(1+(1-(3.53*kic))^(1/2)); % in mm
maxleverarmic = 0.95*supporteffectivedepth;
 if leverarmic<= maxleverarmic
    zic = leverarmic;
 else
    zic = maxleverarmic;
 end

icrequiredsteel = (icmoment*1000000)/(0.87*x(6)*zic);
icminimumsteel = (0.26*0.3*x(5)^(2/3)*columnstrip*1000
*supporteffectivedepth)/x(6);
icprovidedsteel = x(14)*3.14*x(13)*x(13)/4;
icmaximumsteel = 0.04*(columnstrip*1000*x(1)-icprovidedsteel);

% Punching Shear
% At Column Head
uo = 3.14*x(4);
chshear = f-(3.14*(x(4)/1000)*(x(4)/1000)*ultimateload);
cheffectiveshear = 1.15*chshear;
chmaxresistanceshear = 0.5*uo*supporteffectivedepth
*(0.6*(1-(x(5)/250)))*(x(5)/1.5)*(1/1000);

% At Control Perimeter
cpsectiondia = (x(4)/1000)+(4*supporteffectivedepth/1000);
u1 = 3.14*cpsectiondia*1000;
cpshear = f-(3.14*cpsectiondia*cpsectiondia*ultimateload/4);
cpeffectiveshear = 1.15*cpshear;
cprow = (icprovidedsteel/(supporteffectivedepth*x(3)))
cpk = 1+((200/supporteffectivedepth)^0.5)
cpresistanceshear = (0.12*cpk*((100*cprow*x(5))^(1/3)))
*u1*supporteffectivedepth/1000
```

```
% At Drop Panel
dpperimeter = 2*spaneffectivedepth
dpu = (2*x(3))+(2*x(3))+(2*3.14*2*dpperimeter)
dparea = (((x(3)/1000)+(4*spaneffectivedepth/1000))^(2))-((4-3.14)
*((2*spaneffectivedepth/1000)^(2)))
dpshear = f-(dparea*ultimateload)
dpeffectiveshear = 1.15*dpshear
dprow = (icprovidedsteel/(spaneffectivedepth*x(3)))
dpk = 1+((200/spaneffectivedepth)^0.5)
dpresistanceshear = (0.12*dpk*((100*dprow*x(5))^(1/3)))
*dpu*spaneffectivedepth/1000

% Deflection Check
% At Middle Strip

msk = 1.2; % for one way solid slab
msrowzero = (x(5)^(1/2))/1000;
msrowone = cmrequiredsteel/(middlestrip*spaneffectivedepth);
msrowtwo = imrequiredsteel/(middlestrip*spaneffectivedepth);
 if msrowone<=msrowzero
    msspantodepthratio = msk*(11+(1.5*((x(5)^(1/2))
    *(msrowzero/msrowone)))+(3.2*(x(5)^(1/2))
    *(((msrowzero/msrowone)-1)^(3/2))));
     msf3 = (500*cmprovidedsteel)/(x(6)*cmrequiredsteel);
 else
    msspantodepthratio = msk*(11+(1.5*((x(5)^(1/2))
    *(msrowzero/(msrowone-msrowtwo))))+((1/12)*(x(5)^(1/2))
    *((msrowtwo/msrowzero)^(1/2))));
    msf3 = (500*improvidedsteel)/(x(6)*imrequiredsteel);
 end
msf1 = 1;
msf2 = 1; % span less than7m
msbasicspantodepthratio = msspantodepthratio*msf1*msf2*msf3;
msactualspantodepthratio = (lx*1000)/spaneffectivedepth;
```

```
% At Column Strip

csk = 1.2; % for one way solid slab
csrowzero = (x(5)^(1/2))/1000;
csrowone = ccrequiredsteel/(columnstrip*supporteffectivedepth);
csrowtwo = icrequiredsteel/(columnstrip*supporteffectivedepth);
 if csrowone<=csrowzero
    csspantodepthratio = csk*(11+(1.5*((x(5)^(1/2))
    *(csrowzero/csrowone)))+(3.2*(x(5)^(1/2))
    *(((csrowzero/csrowone)-1)^(3/2))));
    csf3 = (500*ccprovidedsteel)/(x(6)*ccrequiredsteel);
 else
    csspantodepthratio = csk*(11+(1.5*((x(5)^(1/2))
    *(csrowzero/(csrowone-csrowtwo))))+((1/12)*(x(5)^(1/2))
    *((csrowtwo/csrowzero)^(1/2))));
    csf3 = (500*icprovidedsteel)/(x(6)*icrequiredsteel);
 end
csf1 = 1;
csf2 = 1; % span less than7m
csbasicspantodepthratio = csspantodepthratio*csf1*csf2*csf3;
csactualspantodepthratio = (lx*1000)/supporteffectivedepth;

% Total Area
steelarea = cmprovidedsteel+ccprovidedsteel+improvidedsteel+icprovidedsteel
concrete = (((lx*1000)*(ly*1000))-steelarea)*x(1)
formwork = (lx*1000)*(ly*1000)
steel = ((cmprovidedsteel+improvidedsteel)*middlestrip)
+((ccprovidedsteel+icprovidedsteel)*columnstrip)
```

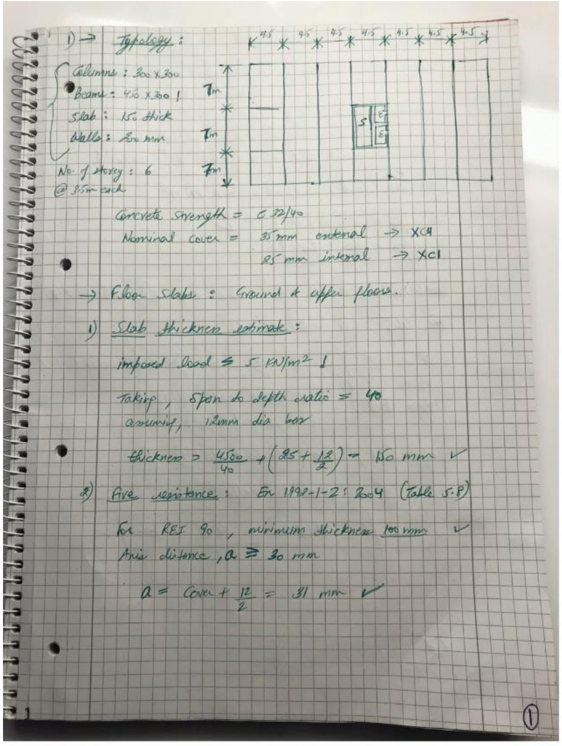
```
% Total Area
steelarea = cmprovidedsteel+ccprovidedsteel+improvidedsteel+icprovidedsteel
concrete = (((lx*1000)*(ly*1000))-steelarea)*x(1)
formwork = (lx*1000)*(ly*1000)
steel = ((cmprovidedsteel+improvidedsteel)*middlestrip)
+((ccprovidedsteel+icprovidedsteel)*columnstrip)

% Constraints function

Cineq = [30-bayarea,1.25-loadratio,180-x(1),15-axisdistance,
    requiredcolumnstrip-columnstrip,requiredmiddlestrip-middlestrip,
    kcm-kdashcm,cmrequiredsteel-cmprovidedsteel,
    cmminimumsteel-cmprovidedsteel,cmprovidedsteel-cmmaximumsteel,
    kcc-kdashcc,ccrequiredsteel-ccprovidedsteel,
    ccminimumsteel-ccprovidedsteel,ccprovidedsteel-ccmaximumsteel,
    kim-kdashim,imrequiredsteel-improvidedsteel,
    imminimumsteel-improvidedsteel,improvidedsteel-immaximumsteel,
    kic-kdashic,icrequiredsteel-icprovidedsteel,
    icminimumsteel-icprovidedsteel,icprovidedsteel-icmaximumsteel,
    cheffectiveshear-chmaxresistanceshear,
    cpeffectiveshear-cpresistanceshear,dpeffectiveshear-dpresistanceshear,
    msactualspantodepthratio-msbasicspantodepthratio
    ,csactualspantodepthratio-csbasicspantodepthratio,];
Ceq = [];


end
```

Code 17: Constraints code for flat slab (Mathworks, 1984)

Frame calculations:

The manual calculations are done as per the Eurocodes. The attached pages contain 4 pages each of the work notebook and hence might be a bit small to see. The thesis is submitted online therefore, it can be maximized and seen well. The pictures are good quality to see after maximizing.

The MATLAB codes for main program, constraint functions and the variables are all attached after the manual calculations.



1) → Typology :

Columns : 300 x 300
Beams : 450 x 300
Slab : 150 thick
Walls : 200 mm

No. of storey : 6
@ 3.5m each

4.5 4.5 4.5 4.5 4.5 4.5 4.5

7m 7m 7m

Concrete strength = C 32/40
Nominal cover = 35 mm external → XC4
25 mm internal → XC1

→ Floor slabs : Ground & upper floors.

1) Slab thickness estimate :

imposed load ≤ 5 KN/m² !

Taking, span to depth ratio = 40
assuming, 12mm dia bar

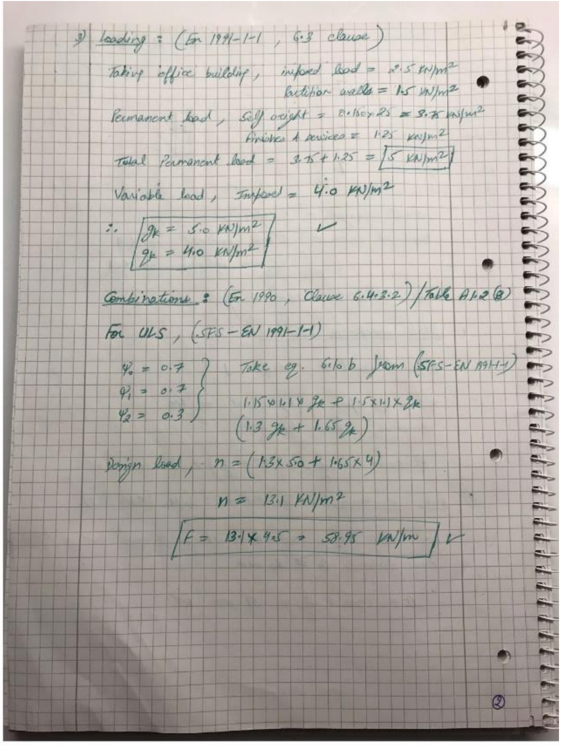
$$\text{thickness} = \frac{4500}{40} + \left(25 + \frac{12}{2}\right) = 150 \text{ mm}$$

2) Fire resistance : EN 1992-1-2 : 2004 (Table 5.8)

For REI 90, minimum thickness 100 mm
Axis distance, $a \geq 30$ mm

$$a = \text{Cover} + \frac{12}{2} = 31 \text{ mm}$$

1



3) Loading : (EN 1991-1-1, 6.3 clause)

Taking office building, imposed load = 2.5 KN/m²
Partition walls = 1.5 KN/m²

Permanent load, Self weight = 0.150 x 25 = 3.75 KN/m²
Finishes & services = 1.25 KN/m²

Total Permanent load = 3.75 + 1.25 = $\boxed{5 \text{ KN/m}^2}$

Variable load, Imposed = 4.0 KN/m²

$$\therefore \boxed{\begin{aligned} g_k &= 5.0 \text{ KN/m}^2 \\ q_k &= 4.0 \text{ KN/m}^2 \end{aligned}}$$

Combinations : (EN 1990, Clause 6.4.3.2) / Table A1.2 (B)

For ULS, (SFS – EN 1991-1-1)

$\psi_0 = 0.7$
$\psi_1 = 0.7$
$\psi_2 = 0.3$

Take eq. 6.10 b from (SFS-EN 1991-1-1)

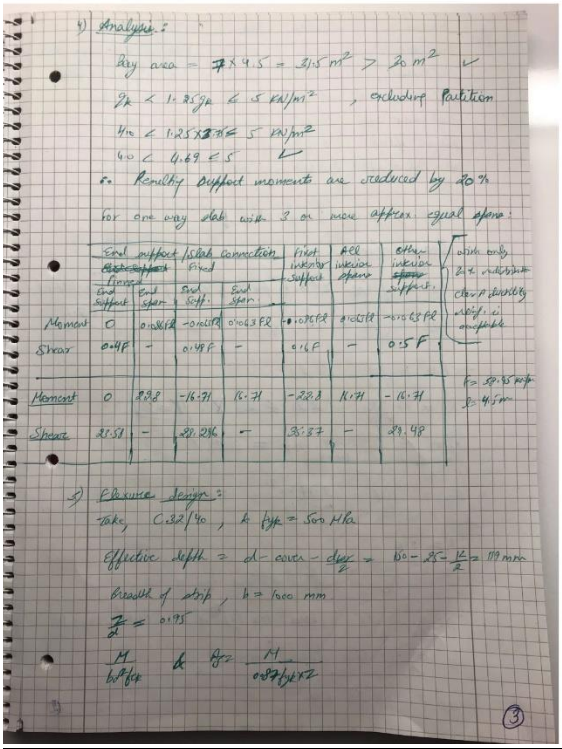
$1.15 \times 1.1 \times g_k + 1.5 \times 1.1 \times q_k$
$(1.3\, g_k + 1.65\, q_k)$

Design load, $n = (1.3 \times 5.0 + 1.65 \times 4)$

$n = 13.1$ KN/m²

$$\boxed{F = 13.1 \times 4.5 = 58.95 \text{ KN/m}}$$

2



4) Analysis :

Bay area = 7 x 4.5 = 31.5 m² > 20 m²

$q_k < 1.25 g_k \leq 5$ KN/m², excluding Partition

$4.0 < 1.25 \times 3.75 \leq 5$ KN/m²
$4.0 < 4.69 \leq 5$

∴ Resulting support moments are reduced by 20 %

For one way slab with 3 or more approx. equal spans:

	End support / slab connection				First interior support	All interior spans	Other interior supports	
	Pinned		Fixed					
	End support	End span	End supp.	End span				
Moment	0	0.086 Fl	-0.063 Fl	0.063 Fl	-0.086 Fl	0.063 Fl	-0.063 Fl	with only 20 % redistribution, class A ductility reinf. is acceptable
Shear	0.4F	–	0.48 F	–	0.6F	–	0.5 F	
Moment	0	22.8	-16.71	16.71	-22.8	16.71	-16.71	F = 58.95 KN/m, l = 4.5m
Shear	23.58	–	28.296	–	35.37	–	29.48	

5) Flexure design :
Take, C 32/40, & f_{yk} = 500 MPa

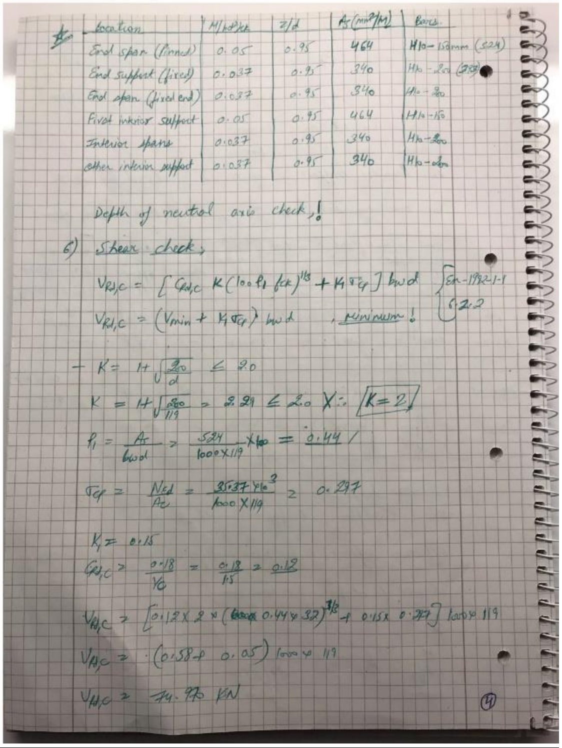
Effective depth = d – cover – $\frac{d_{bar}}{2}$ = 150 – 25 – $\frac{12}{2}$ = 119 mm

Breadth of strip, b = 1000 mm

$\frac{z}{d} = 0.95$

$$\frac{M}{bd^2 f_{ck}} \quad \& \quad A_s = \frac{M}{0.87 f_{yk} \times z}$$

3



Location	$M/bd^2 f_{ck}$	z/d	A_s (mm²/m)	Bars
End span (Pinned)	0.05	0.95	464	H10 – 150mm (524)
End support (fixed)	0.037	0.95	340	H10 – 200 (393)
End span (fixed end)	0.037	0.95	340	H10 – 200
First interior support	0.05	0.95	464	H10 – 150
Interior spans	0.037	0.95	340	H10 – 200
other interior support	0.037	0.95	340	H10 – 200

Depth of neutral axis check !

6) Shear check ;

$V_{Rd,c} = [C_{Rd,c}\, K (100 \rho_1 f_{ck})^{1/3} + K_1 \sigma_{cp}]\, b_w d$
$V_{Rd,c} = (V_{min} + K_1 \sigma_{cp})\, b_w d$, minimum !

EN-1992-1-1 6.2.2

$K = 1 + \sqrt{\frac{200}{d}} \leq 2.0$

$K = 1 + \sqrt{\frac{200}{119}} = 2.29 \leq 2.0$ ✗ ∴ $\boxed{K = 2}$

$\rho_1 = \frac{A_s}{b_w d} = \frac{524}{1000 \times 119} \times 100 = 0.44$

$\sigma_{cp} = \frac{N_{Ed}}{A_c} = \frac{35.37 \times 10^3}{1000 \times 119} = 0.297$

$K_1 = 0.15$

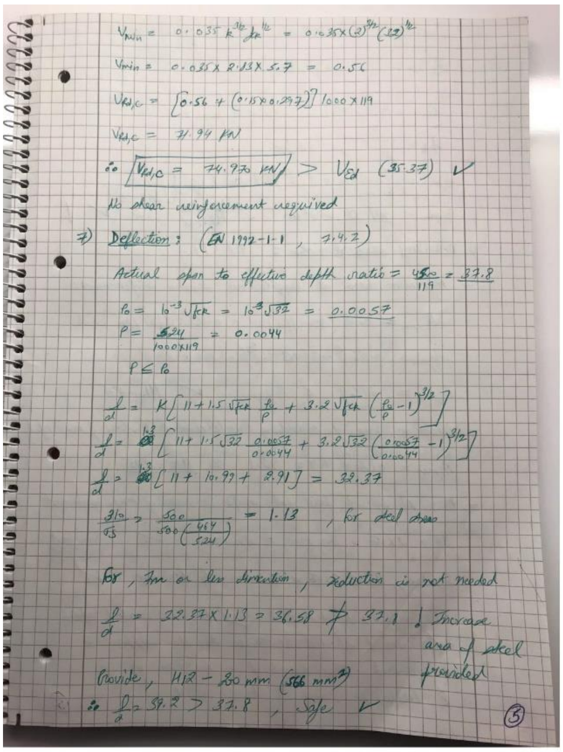
$C_{Rd,c} = \frac{0.18}{\gamma_c} = \frac{0.18}{1.5} = 0.12$

$V_{Rd,c} = [0.12 \times 2 \times (0.44 \times 32)^{1/3} + 0.15 \times 0.297]\, 1000 \times 119$

$V_{Rd,c} = (0.58 + 0.05)\, 1000 \times 119$

$V_{Rd,c} = 74.97$ KN

4



$V_{min} = 0.035\, k^{3/2} f_{ck}^{1/2} = 0.035 \times (2)^{3/2} (32)^{1/2}$

$V_{min} = 0.035 \times 2.83 \times 5.7 = 0.56$

$V_{Rd,c} = [0.56 + (0.15 \times 0.297)]\, 1000 \times 119$

$V_{Rd,c} = 71.94$ kN

$\therefore \boxed{V_{Rd,c} = 74.970 \text{ kN}} > V_{Ed}\ (35.37)$ ✓

No shear reinforcement required

7) **Deflection:** (EN 1992-1-1, 7.4.2)

Actual span to effective depth ratio $= \frac{4500}{119} = \underline{37.8}$

$\rho_0 = 10^{-3}\sqrt{f_{ck}} = 10^{-3}\sqrt{32} = \underline{0.0057}$

$\rho = \frac{524}{1000 \times 119} = 0.0044$

$\rho \le \rho_0$

$\frac{l}{d} = K\left[11 + 1.5\sqrt{f_{ck}}\,\frac{\rho_0}{\rho} + 3.2\sqrt{f_{ck}}\left(\frac{\rho_0}{\rho} - 1\right)^{3/2}\right]$

$\frac{l}{d} = 1.3\left[11 + 1.5\sqrt{32}\,\frac{0.0057}{0.0044} + 3.2\sqrt{32}\left(\frac{0.0057}{0.0044} - 1\right)^{3/2}\right]$

$\frac{l}{d} = 1.3\,[11 + 10.99 + 2.91] = 32.37$

$\frac{310}{\sigma_s} = \frac{500}{500\left(\frac{464}{524}\right)} = 1.13$, for steel stress

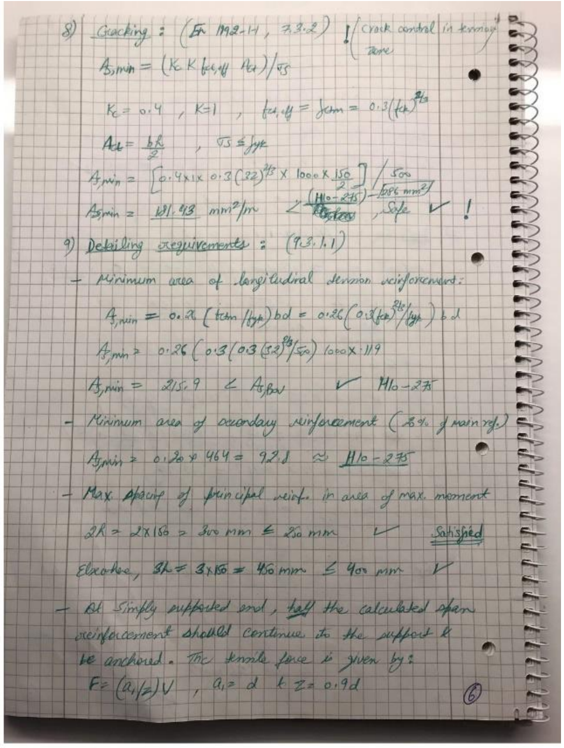
For, 7m or less dimension, reduction is not needed

$\frac{l}{d} = 32.37 \times 1.13 = 36.58 \not> 37.8$! Increase area of steel provided

Provide, H12 – 200 mm (566 mm²)

$\therefore \frac{l}{d} = 39.2 > 37.8$, Safe ✓

(5)



8) Cracking: (EN 1992-1-1, 7.3.2) ! (crack control in tension zone)

$A_{s,min} = (k_c\, k\, f_{ct,eff}\, A_{ct})/\sigma_s$

$k_c = 0.4$, $k = 1$, $f_{ct,eff} = f_{ctm} = 0.3 (f_{ck})^{2/3}$

$A_{ct} = \frac{bh}{2}$, $\sigma_s \le f_{yk}$

$A_{s,min} = \left[0.4 \times 1 \times 0.3 (32)^{2/3} \times 1000 \times \frac{150}{2}\right] / 500$

$A_{s,min} = \underline{181.43}$ mm²/m $<$ (H10 – 275) – 286 mm², Safe ✓ !

9) Detailing requirements: (9.3.1.1)

– Minimum area of longitudinal tension reinforcement:

$A_{s,min} = 0.26\,(f_{ctm}/f_{yk})\,bd = 0.26\left(0.3 (f_{ck})^{2/3}/f_{yk}\right) b\,d$

$A_{s,min} = 0.26\,(0.3\,(0.3\,(32)^{2/3}/500)\,1000 \times 119$

$A_{s,min} = 215.9 < A_{s,prov}$ ✓ H10 – 275

– Minimum area of secondary reinforcement (20% of main ref.)

$A_{s,min} = 0.20 \times 464 = 92.8 \approx$ H10 – 275

– Max. spacing of principal reinf. in area of max. moment

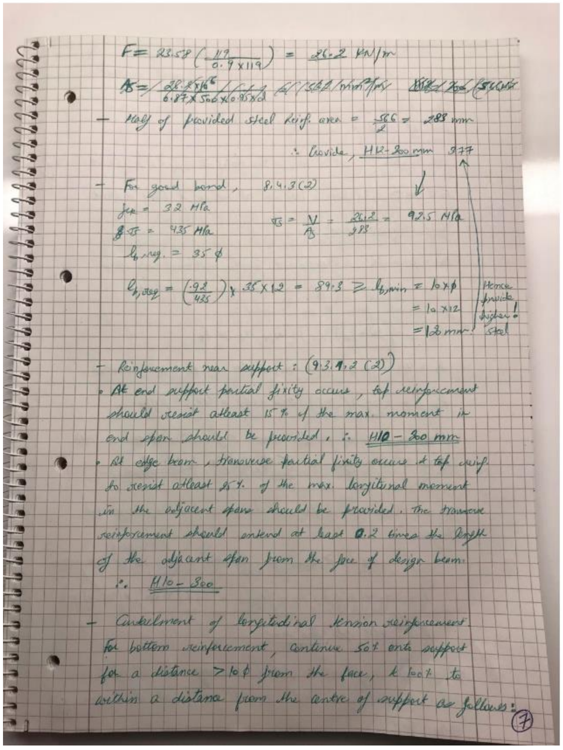
$2h = 2 \times 150 = 300$ mm ≤ 250 mm ✓ Satisfied

Elsewhere, $3h = 3 \times 150 = 450$ mm ≤ 400 mm ✓

– At simply supported end, half the calculated span reinforcement should continue to the support & be anchored. The tensile force is given by:

$F = (a_1/z)V$, $a_1 = d$ & $z = 0.9d$

(6)



$F = 23.58\left(\frac{119}{0.9 \times 119}\right) = \underline{26.2}$ kN/m

$A_s = \frac{28.2 \times 10^6}{6.87 \times 500 \times 0.95 \times d}$

– Half of provided steel reinf. area $= \frac{566}{2} = 283$ mm

$\therefore$ Provide, H12 – 300 mm 377

– For good bond, 8.4.3(2)

$f_{ck} = 3.2$ MPa

$\sigma_s = 435$ MPa

$\sigma_s = \frac{V}{A_s} = \frac{26.2}{283} = 92.5$ MPa

$l_{b,req} = 35\phi$

$l_{b,req} = \left(\frac{92}{435}\right) \times 35 \times 12 = 89.3 \ge l_{b,min} = 10 \times \phi = 10 \times 12 = 120$ mm ! Hence provide higher steel

– Reinforcement near support: (9.3.1.2 (2))

• At end support partial fixity occurs, top reinforcement should resist atleast 15% of the max. moment in end span should be provided. $\therefore$ H10 – 300 mm

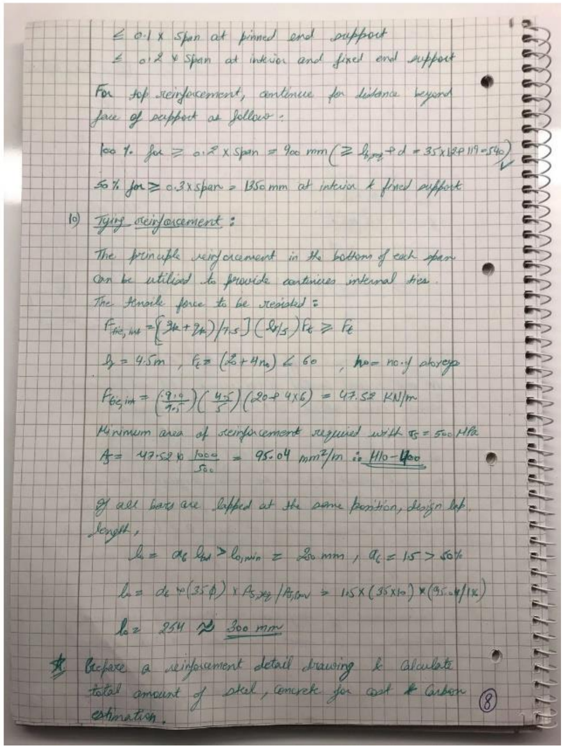
• At edge beam, transverse partial fixity occurs & top reinf. to resist atleast 25% of the max. longitudinal moment in the adjacent span should be provided. The transverse reinforcement should extend at least 0.2 times the length of the adjacent span from the face of design beam.

$\therefore$ H10 – 300

– Curtailment of longitudinal tension reinforcement

For bottom reinforcement, continue 50% onto support for a distance $> 10\phi$ from the face, & 100% to within a distance from the centre of support as follows:

(7)



$\le 0.1 \times$ span at pinned end support

$\le 0.2 \times$ span at interior and fixed end support

For top reinforcement, continue for distance beyond face of support as follow:

100% for $\ge 0.2 \times$ span $= 900$ mm ($\ge l_{b,req} + d = 35 \times 12 + 119 = 540$) ✓

50% for $\ge 0.3 \times$ span $= 1350$ mm at interior & fixed support

10) Tying reinforcement:

The principle reinforcement in the bottom of each span can be utilised to provide continuous internal ties.

The tensile force to be resisted:

$F_{tie,int} = [(g_k + q_k)/7.5]\,(l_r/5)\,F_t \ge F_t$

$l_r = 4.5$ m, $F_t = (20 + 4n_0) \le 60$, $n_0 =$ no. of storeys

$F_{tie,int} = \left(\frac{9.0}{7.5}\right)\left(\frac{4.5}{5}\right)(20 + 4 \times 6) = 47.52$ kN/m

Minimum area of reinforcement required with $\sigma_s = 500$ MPa

$A_s = 47.52 \times \frac{1000}{500} = 95.04$ mm²/m $\therefore$ H10 – 400

If all bars are lapped at the same position, design lap length,

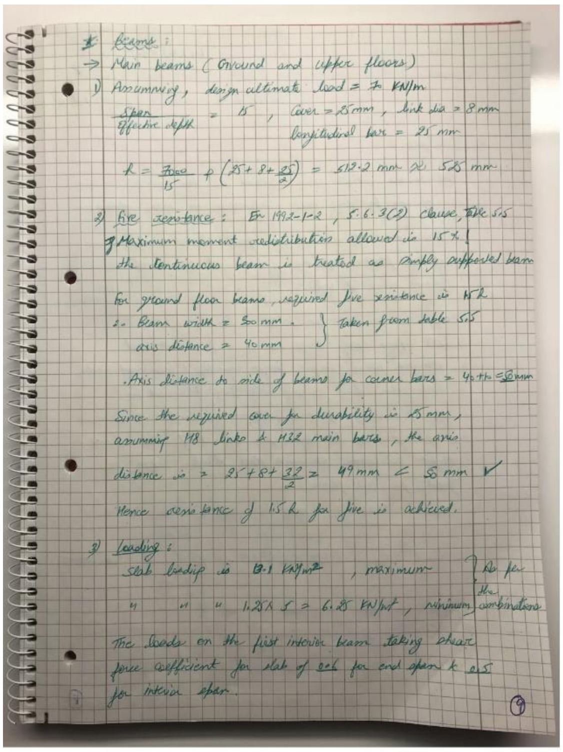
$l_0 = \alpha_6\, l_{bd} > l_{0,min} = 200$ mm, $\alpha_6 = 1.5 > 50\%$

$l_0 = \alpha_6 \times (35\phi) \times A_{s,req}/A_{s,prov} = 1.5 \times (35 \times 10) \times (95.04/196)$

$l_0 = 254 \approx \underline{300 \text{ mm}}$

★ Prepare a reinforcement detail drawing & calculate total amount of steel, concrete for cost & carbon estimation.

(8)



* Beams:

→ Main beams (Ground and upper floors)

1) Assuming, design ultimate load = 70 kN/m

$\frac{\text{Span}}{\text{Effective depth}} = 15$, Cover = 25 mm, link dia = 8 mm, longitudinal bar = 25 mm

$h = \frac{7000}{15} + \left(25 + 8 + \frac{25}{2}\right) = 512.2 \text{ mm} \approx 525 \text{ mm}$

2) Fire resistance: EN 1992-1-2, 5.6.3(2) clause, Table 5.5

Maximum moment redistribution allowed is 15%!

The continuous beam is treated as simply supported beam

For ground floor beams, required fire resistance is 1.5 h

∴ Beam width = 300 mm, axis distance = 40 mm } Taken from table 5.5

∴ Axis distance to side of beams for corner bars = 40 + 10 = 50 mm

Since the required cover for durability is 25 mm, assuming H8 links & H32 main bars, the axis distance is $= 25 + 8 + \frac{32}{2} = 49 \text{ mm} < 50 \text{ mm}$ ✓

Hence resistance of 1.5 h for fire is achieved.

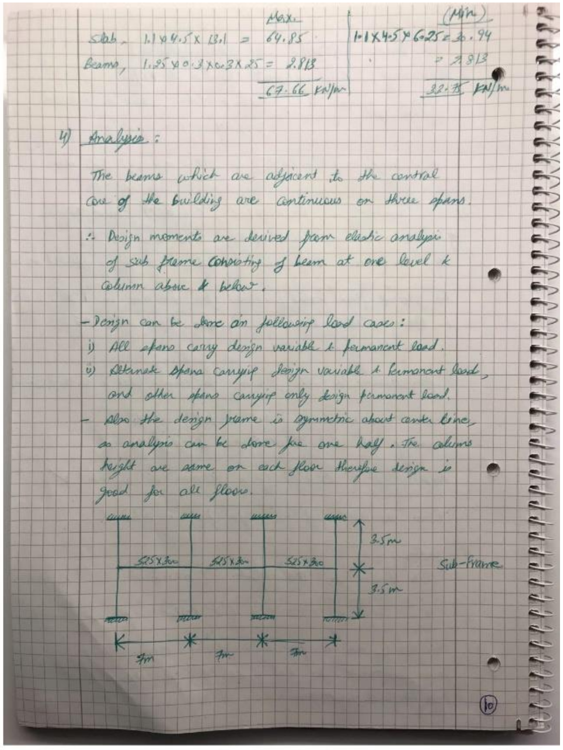
3) Loading:

Slab loading is 13.1 kN/m², maximum } As per the combinations

" " " $1.25 \times 5 = 6.25$ kN/m², minimum

The loads on the first interior beam taking shear force coefficient for slab of 0.6 for end span & 0.5 for interior span.

(9)



	Max.	(Min)
Slab,	$1.1 \times 4.5 \times 13.1 = 64.85$	$1.1 \times 4.5 \times 6.25 = 30.94$
Beams,	$1.25 \times 0.3 \times 0.3 \times 25 = 2.813$	$= 2.813$
	67.66 kN/m	33.75 kN/m

4) Analysis:

The beams which are adjacent to the central core of the building are continuous on three spans.

∴ Design moments are derived from elastic analysis of sub frame consisting of beam at one level & column above & below.

- Design can be done on following load cases:
  i) All spans carry design variable & permanent load.
  ii) Alternate spans carrying design variable & permanent load, and other spans carrying only design permanent load.
- Also the design frame is symmetric about center line, so analysis can be done for one half. The column height are same on each floor therefore design is good for all floors.

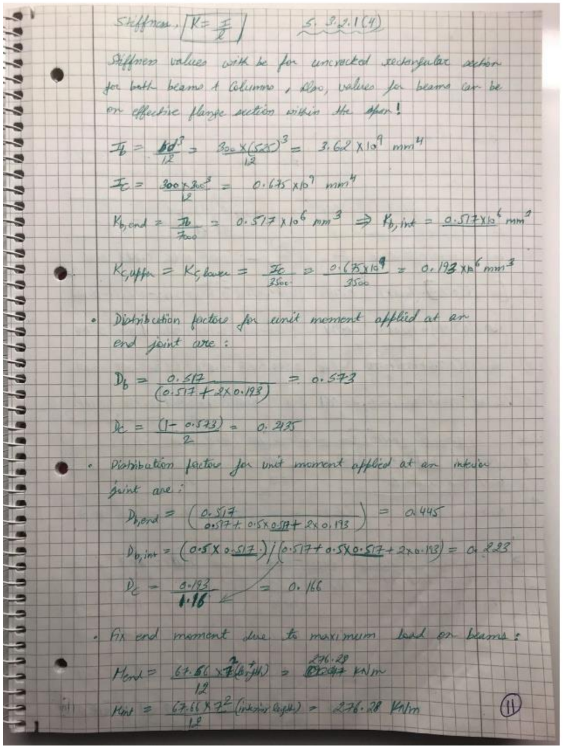
525×300 525×300 525×300

3.5m 3.5m

Sub-Frame

7m 7m 7m

(10)



Stiffness, $\boxed{K = \frac{I}{l}}$ 5.3.2.1(4)

Stiffness values will be for uncracked rectangular section for both beams & columns, Also, values for beams can be on effective flange section within the span!

$I_b = \frac{bd^3}{12} = \frac{300 \times (525)^3}{12} = 3.62 \times 10^9 \text{ mm}^4$

$I_c = \frac{300 \times 300^3}{12} = 0.675 \times 10^9 \text{ mm}^4$

$K_{b,end} = \frac{I_b}{7000} = 0.517 \times 10^6 \text{ mm}^3 \Rightarrow K_{b,int} = 0.517 \times 10^6 \text{ mm}^3$

$K_{c,upper} = K_{c,lower} = \frac{I_c}{3500} = \frac{0.675 \times 10^9}{3500} = 0.193 \times 10^6 \text{ mm}^3$

- Distribution factors for unit moment applied at an end joint are:

$D_b = \frac{0.517}{(0.517 + 2 \times 0.193)} = 0.573$

$D_c = \frac{(1 - 0.573)}{2} = 0.235$

- Distribution factors for unit moment applied at an interior joint are:

$D_{b,end} = \left(\frac{0.517}{0.517 + 0.5 \times 0.517 + 2 \times 0.193}\right) = 0.445$

$D_{b,int} = (0.5 \times 0.517) / (0.517 + 0.5 \times 0.517 + 2 \times 0.193) = 0.223$

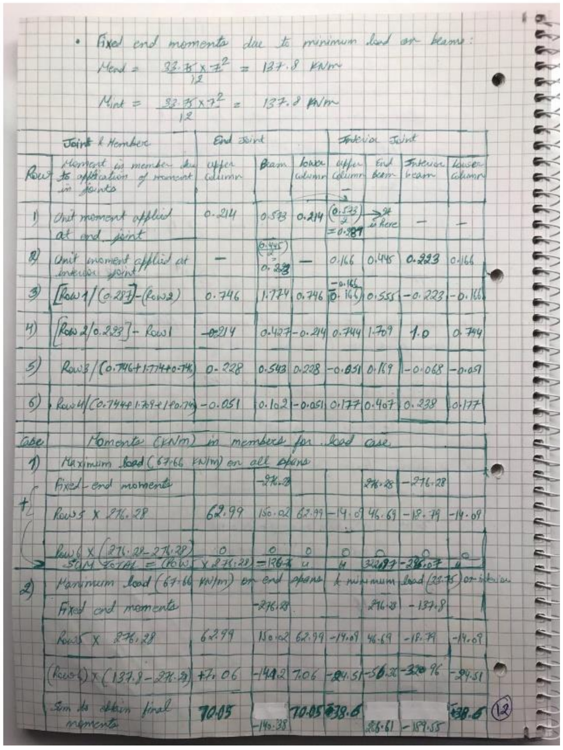
$D_c = \frac{0.193}{1.16} = 0.166$

- Fix end moment due to maximum load on beams:

$M_{end} = \frac{67.66 \times 7^2}{12}$ (end span) $= 276.28$ kN/m

$M_{int} = \frac{67.66 \times 7^2}{12}$ (interior length) $= 276.28$ kN/m

(11)



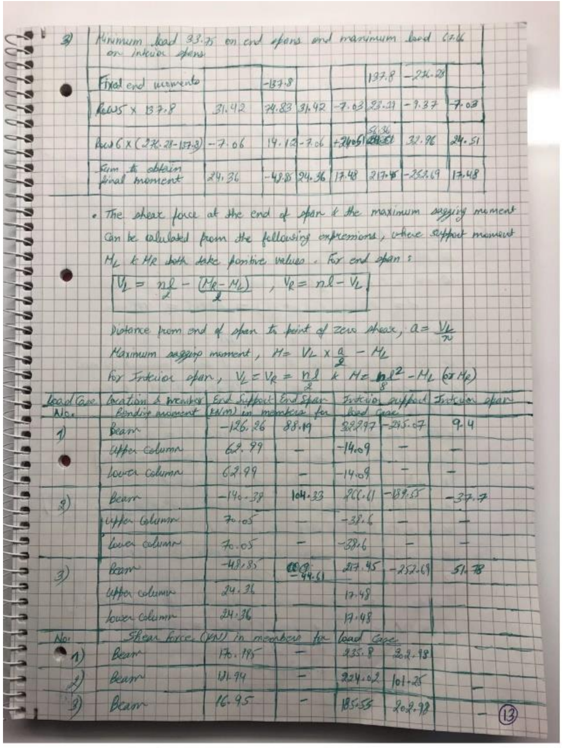
- Fixed end moments due to minimum load on beams:

$M_{end} = \frac{33.75 \times 7^2}{12} = 137.8$ kNm

$M_{int} = \frac{33.75 \times 7^2}{12} = 137.8$ kN/m

Row	Joint & Member / Moment in member due to application of moment in joints	End Joint: Upper column	End Joint: Beam	End Joint: Lower column	Interior Joint: Upper column	Interior Joint: End beam	Interior Joint: Interior beam	Interior Joint: Lower column
1)	Unit moment applied at end joint	0.214	0.573	0.214	$\frac{0.573}{2} = 0.287$	94 is here	–	–
2)	Unit moment applied at interior joint	–	$\frac{0.445}{2} = 0.223$	–	0.166	0.445	0.223	0.166
3)	[Row 1/(0.287)] − (Row 2)	0.746	1.774	0.746	−0.166	0.555	−0.223	−0.166
4)	[Row 2/0.223] − Row 1	−0.214	0.427	−0.214	0.744	1.709	1.0	0.744
5)	Row 3/(0.746+1.774+0.746)	0.228	0.543	0.228	−0.051	0.169	−0.068	−0.051
6)	Row 4/(0.744+1.709+1+0.744)	−0.051	0.102	−0.051	0.177	0.407	0.238	0.177
Case	Moments (kNm) in members for load case							
1)	Maximum load (67.66 kN/m) on all spans							
	Fixed-end moments		−276.28			276.28	−276.28	
	Row 5 × 276.28	62.99	150.02	62.99	−14.09	46.69	−18.79	−14.09
	Row 6 × (276.28 − 276.28)	0	0	0	0	0	0	0
	SUM TOTAL = (Row 5 × 276.28)		−126.2	"	"	322.97	−286.07	"
2)	Maximum load (67.66 kN/m) on end spans & minimum load (33.75) on interior							
	Fixed end moments		−276.28			276.28	−137.8	
	Row 5 × 276.28	62.99	150.02	62.99	−14.09	46.69	−18.79	−14.09
	(Row 6) × (137.8 − 276.28)	+7.06	−14.12	7.06	−24.51	−56.36	−32.96	−24.51
	Sum to obtain final moments	70.05	−140.38	70.05	−38.6	266.61	−189.55	−38.6

(12)



3) Minimum load 33.75 on end spans and maximum load 67.66 on interior spans

Fixed end moments		-137.8			137.8	-276.28	
0.035 × 137.8	31.42	74.83	31.42	-7.03	28.21	-9.37	-7.03
0.06 × (276.28 − 137.8)	-7.06	14.12	-7.06	+24.51	56.36	32.96	24.51
Sum to obtain final moment	24.36	-48.85	24.36	17.48	217.45	-252.69	17.48

- The shear force at the end of span & the maximum sagging moment can be calculated from the following expressions, where support moment M_L & M_R both take positive values. For end span:

$$\boxed{V_L = \frac{nl}{2} - \frac{(M_R - M_L)}{l}, \quad V_R = nl - V_L}$$

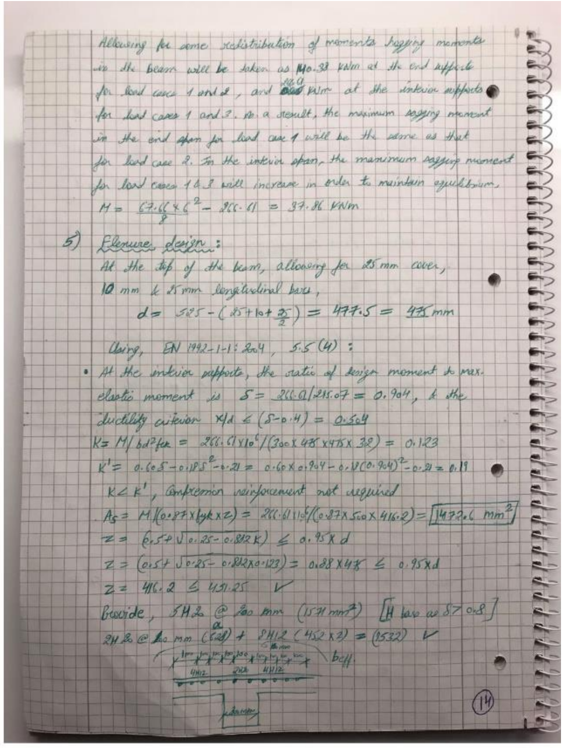
Distance from end of span to point of zero shear, $a = \frac{V_L}{n}$

Maximum sagging moment, $M = V_L \times \frac{a}{2} - M_L$

For Interior span, $V_L = V_R = \frac{nl}{2}$ & $M = \frac{nl^2}{8} - M_L$ (or M_R)

Load Case No.	Location & member	End Support	End Span	Interior support		Interior span
	Bending moment (kNm) in members for load case					
1)	Beam	−126.26	88.19	322.97	−295.07	9.4
	Upper column	62.99	−	−14.09	−	−
	Lower column	62.99	−	−14.09	−	−
2)	Beam	−140.38	104.33	266.61	−189.55	−37.7
	Upper column	70.05	−	−38.6	−	−
	Lower column	70.05	−	−38.6	−	−
3)	Beam	−48.85	−44.61	217.45	−252.69	51.78
	Upper column	24.36		17.48		
	Lower column	24.36		17.48		
No.	Shear force (kN) in members for load case					
1)	Beam	170.195	−	235.8	202.98	
2)	Beam	121.94	−	224.02	101.25	−
3)	Beam	16.95	−	185.55	202.98	−

(13)



Allowing for some redistribution of moments, hogging moments in the beam will be taken as 140.38 kNm at the end supports for load case 1 and 2, and 266.61 kNm at the interior supports for load cases 1 and 3. As a result, the maximum sagging moment in the end span for load case 1 will be the same as that for load case 2. In the interior span, the maximum sagging moment for load cases 1 & 3 will increase in order to maintain equilibrium.

$M = \frac{67.66 \times 6^2}{8} - 266.61 = 37.86$ kNm

5) Flexure design:

At the top of the beam, allowing for 25 mm cover, 10 mm & 25 mm longitudinal bars,

$$d = 525 - (25 + 10 + \frac{25}{2}) = 477.5 = 475 \text{ mm}$$

Using, EN 1992-1-1:2004, 5.5 (4):

- At the interior supports, the ratio of design moment to max. elastic moment is $\delta = 266.61/295.07 = 0.904$, & the ductility criterion $x/d \le (\delta - 0.4) = 0.504$

$K = M/bd^2 f_{ck} = 266.61 \times 10^6/(300 \times 475 \times 475 \times 32) = 0.123$

$K' = 0.60\delta - 0.18\delta^2 - 0.21 = 0.60 \times 0.904 - 0.18(0.904)^2 - 0.21 = 0.19$

$K < K'$, Compression reinforcement not required

$A_s = M/(0.87 \times f_{yk} \times z) = 266.61 \times 10^6/(0.87 \times 500 \times 416.2) = \boxed{1472.6 \text{ mm}^2}$

$z = (0.5 + \sqrt{0.25 - 0.882K}) \le 0.95 \times d$

$z = (0.5 + \sqrt{0.25 - 0.882 \times 0.123}) = 0.88 \times 475 \le 0.95 \times d$

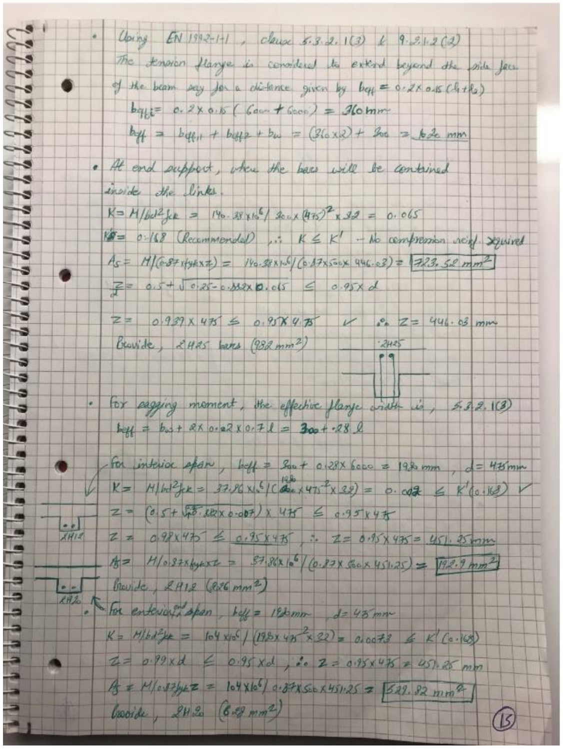
$z = 416.2 \le 451.25$ ✓

Provide, 5H20 @ 200 mm (1571 mm²) [H bars as $\delta > 0.8$]

2H20 @ 200 mm (628) + 8H12 (452 × 2) = (1532) ✓

4H12 2H20 4H12 b_{eff}

(14)



- Using EN 1992-1-1, clause 5.3.2.1(3) & 9.2.1.2(2)

The tension flange is considered to extend beyond the side face of the beam say for a distance given by $b_{eff} = 0.2 \times 0.15 (l_1 + l_2)$

$b_{eff,i} = 0.2 \times 0.15 (6000 + 6000) = 360$ mm

$b_{eff} = b_{eff,1} + b_{eff,2} + b_w = (360 \times 2) + 300 = 1020$ mm

- At end support, where the bars will be contained inside the links.

$K = M/bd^2 f_{ck} = 140.38 \times 10^6/300 \times (475)^2 \times 32 = 0.065$

$K' = 0.168$ (Recommended) ∴ $K \le K'$ – No compression reinf. required

$A_s = M/(0.87 \times f_{yk} \times z) = 140.38 \times 10^6/(0.87 \times 500 \times 446.03) = \boxed{723.52 \text{ mm}^2}$

$\frac{z}{d} = 0.5 + \sqrt{0.25 - 0.882 \times 0.065} \le 0.95 \times d$

$z = 0.937 \times 475 \le 0.95 \times 4.75$ ✓ ∴ $z = 446.03$ mm

Provide, 2H25 bars (982 mm²)

2H25

- For sagging moment, the effective flange width is, 5.3.2.1(3)

$b_{eff} = b_w + 2 \times 0.22 \times 0.7 l = 300 + .28 l$

For interior span, $b_{eff} = 300 + 0.28 \times 6000 = 1980$ mm, $d = 475$ mm

$K = M/bd^2 f_{ck} = 37.86 \times 10^6/(1980 \times 475^2 \times 32) = 0.002 \le K'(0.168)$ ✓

$z = (0.5 + \sqrt{0.25 - 0.882 \times 0.007}) \times 475 \le 0.95 \times 475$

$z = 0.98 \times 475 \le 0.95 \times 475$, ∴ $z = 0.95 \times 475 = 451.25$ mm

$A_s = M/0.87 \times f_{yk} \times z = 37.86 \times 10^6/(0.87 \times 500 \times 451.25) = \boxed{192.9 \text{ mm}^2}$

Provide, 2H12 (226 mm²)

2H12

2H20

- For exterior/end span, $b_{eff} = 1980$ mm, $d = 475$ mm

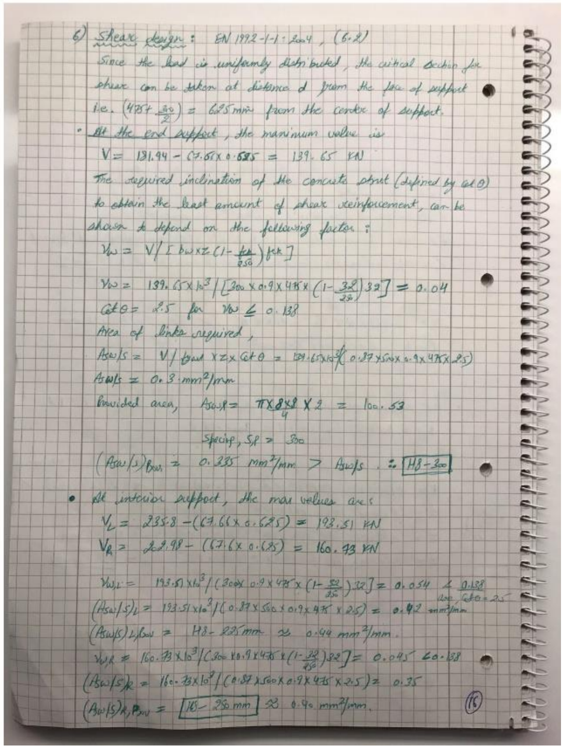
$K = M/bd^2 f_{ck} = 104 \times 10^6/(1980 \times 475^2 \times 32) = 0.0073 \le K'(0.168)$

$z = 0.99 \times d \le 0.95 \times d$, ∴ $z = 0.95 \times 475 = 451.25$ mm

$A_s = M/0.87 f_{yk} z = 104 \times 10^6/0.87 \times 500 \times 451.25 = \boxed{529.82 \text{ mm}^2}$

Provide, 2H20 (628 mm²)

(15)



6) Shear design: EN 1992-1-1:2004, (6.2)

Since the load is uniformly distributed, the critical section for shear can be taken at distance d from the face of support i.e. $(475 + \frac{300}{2}) = 625$ mm from the centre of support.

- At the end support, the maximum value is

$V = 181.94 - (67.66 \times 0.625 = 139.65$ kN

The required inclination of the concrete strut (defined by $\cot\theta$) to obtain the least amount of shear reinforcement, can be shown to depend on the following factor:

$v_w = V/[b_w \times z (1 - \frac{f_{ck}}{250}) f_{ck}]$

$v_w = 139.65 \times 10^3/[300 \times 0.9 \times 475 \times (1 - \frac{32}{250}) 32] = 0.04$

$\cot\theta = 2.5$ for $v_w \le 0.138$

Area of links required,

$A_{sw}/s = V/f_{ywd} \times z \times \cot\theta = 139.65 \times 10^3/(0.87 \times 500 \times 0.9 \times 475 \times 2.5)$

$A_{sw}/s = 0.3$ mm²/mm

Provided area, $A_{sw,p} = \pi \times \frac{8 \times 8}{4} \times 2 = 100.53$

Spacing, $S_p = 300$

$(A_{sw}/s)_{Prov} = 0.335$ mm²/mm $> A_{sw}/s$ ∴ $\boxed{H8-300}$

- At interior support, the max values are:

$V_L = 235.8 - (67.66 \times 0.625) = 193.51$ kN

$V_R = 202.98 - (67.6 \times 0.625) = 160.73$ kN

$v_{w,L} = 193.51 \times 10^3/(300 \times 0.9 \times 475 \times (1 - \frac{32}{250}) 32] = 0.054 \le 0.138$ use $\cot\theta = 2.5$

$(A_{sw}/s)_L = 193.51 \times 10^3/(0.87 \times 500 \times 0.9 \times 475 \times 2.5) = 0.42$ mm²/mm

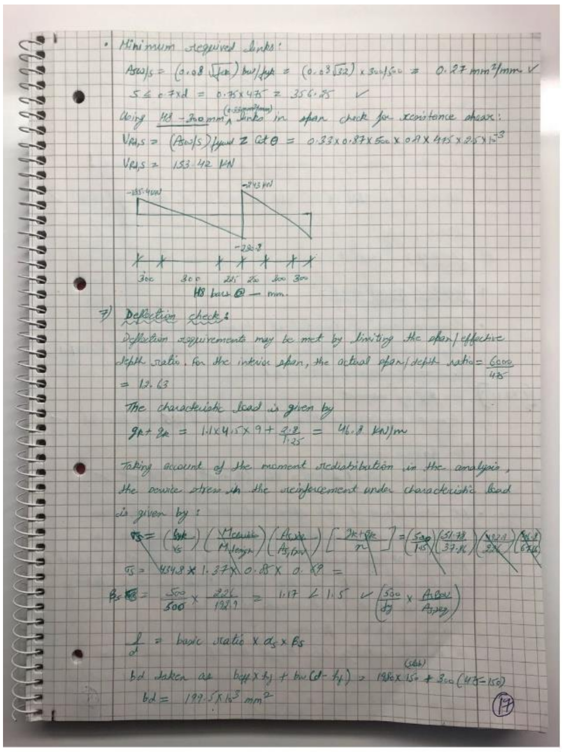
$(A_{sw}/s)_{L,Prov} = $ H8 – 225 mm ≈ 0.44 mm²/mm.

$v_{w,R} = 160.73 \times 10^3/(300 \times 0.9 \times 475 \times (1 - \frac{32}{250}) 32] = 0.045 < 0.138$

$(A_{sw}/s)_R = 160.73 \times 10^3/(0.87 \times 500 \times 0.9 \times 475 \times 2.5) = 0.35$

$(A_{sw}/s)_{R,Prov} = \boxed{\text{H8} - 250 \text{ mm}}$ ≈ 0.40 mm²/mm.

(16)



- Minimum required links:

$A_{sw}/s = (0.08\sqrt{f_{ck}})\, b_w/f_{yk} = (0.08\sqrt{32}) \times 300/500 = 0.27\ mm^2/mm$ ✓

$s \le 0.75 \times d = 0.75 \times 475 = 356.25$ ✓

Using H8 – 300 mm (0.33 mm²/mm) links in span check for resistance shear:

$V_{Rd,s} = (A_{sw}/s) f_{ywd}\, z \cot\theta = 0.33 \times 0.87 \times 500 \times 0.9 \times 475 \times 2.5 \times 10^{-3}$

$V_{Rd,s} = 153.42\ kN$

-255.4 kN, -243 kN, -220.8

300, 800, 225, 250, 800, 300

H8 bars @ — mm.

7) Deflection check:

Deflection requirements may be met by limiting the span/effective depth ratio. For the interior span, the actual span/depth ratio $= \frac{6000}{475} = 12.63$

The characteristic load is given by

$g_k + q_k = 1.1 \times 4.5 \times 9 + \frac{2.8}{1.25} = 46.8\ kN/m$

Taking account of the moment redistribution in the analysis, the service stress in the reinforcement under characteristic load is given by:

$\sigma_s = \left(\frac{f_{yk}}{\gamma_s}\right)\left(\frac{M_{elastic}}{M_{design}}\right)\left(\frac{A_{s,req}}{A_{s,prov}}\right)\left[\frac{g_k + q_k}{n}\right] = \left(\frac{500}{1.15}\right)\left(\frac{51.78}{37.86}\right)\left(\frac{192.9}{226}\right)\left(\frac{46.8}{67.16}\right)$

$\sigma_s = 434.8 \times 1.37 \times 0.85 \times 0.69 =$

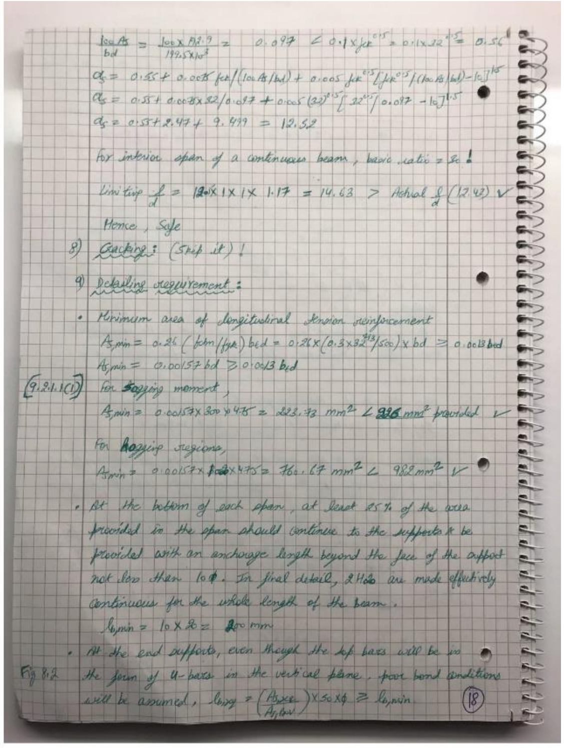
$\beta_s = \frac{500}{500} \times \frac{226}{192.9} = 1.17 < 1.5$ ✓ $\left(\frac{500}{f_y} \times \frac{A_{s,prov}}{A_{s,req}}\right)$

$\frac{l}{d} = \text{basic ratio} \times \alpha_s \times \beta_s$

bd taken as $b_{eff} \times h_f + b_w(d - h_f) = 1980 \times 150 + 300(475 - 150)$ (slab)

$bd = 199.5 \times 10^3\ mm^2$

(17)



$\frac{100 A_s}{bd} = \frac{100 \times 192.9}{199.5 \times 10^3} = 0.097 < 0.1 \times f_{ck}^{0.5} = 0.1 \times 32^{0.5} = 0.56$

$\alpha_s = 0.55 + 0.0075 f_{ck}/(100 A_s/bd) + 0.005 f_{ck}^{0.5}[f_{ck}^{0.5}/(100 A_s/bd) - 10]^{1.5}$

$\alpha_s = 0.55 + 0.0075 \times 32/0.097 + 0.005(32)^{0.5}[32^{0.5}/0.097 - 10]^{1.5}$

$\alpha_s = 0.55 + 2.47 + 9.499 = 12.52$

For interior span of a continuous beam, basic ratio = 20!

Limiting $\frac{l}{d} = 12.5 \times 1 \times 1 \times 1.17 = 14.63 >$ Actual $\frac{l}{d}$ (12.43) ✓

Hence, Safe

8) Cracking: (Skip it)!

9) Detailing requirement:

- Minimum area of longitudinal tension reinforcement

$A_{s,min} = 0.26\,(f_{ctm}/f_{yk})\, b_t d = 0.26 \times (0.3 \times 32^{2/3}/500) \times bd \ge 0.0013\, bd$

$A_{s,min} = 0.00157\, bd \ge 0.0013\, b_t d$

[9.2.1.1(1)] For sagging moment,

$A_{s,min} = 0.00157 \times 300 \times 475 = 223.73\ mm^2 < 226\ mm^2$ provided ✓

For hogging regions,

$A_{s,min} = 0.00157 \times 1020 \times 475 = 760.67\ mm^2 < 982\ mm^2$ ✓

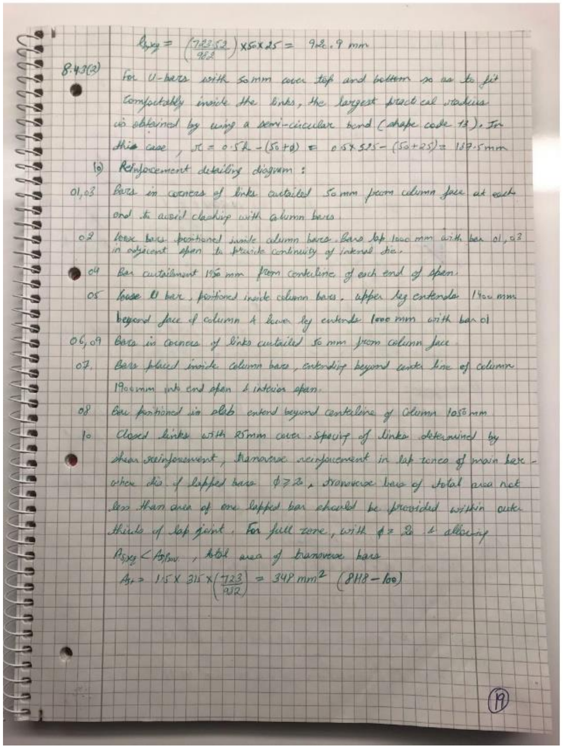
- At the bottom of each span, at least 25% of the area provided in the span should continue to the supports & be provided with an anchorage length beyond the face of the support not less than 10ϕ. In final detail, 2H20 are made effectively continuous for the whole length of the beam.

$l_{b,min} = 10 \times 20 = 200\ mm$

- At the end supports, even though the top bars will be in

Fig 8.2 the form of U-bars in the vertical plane, poor bond conditions will be assumed, $l_{bxy} = \left(\frac{A_{s,req}}{A_{s,prov}}\right) \times 50 \times \phi \ge l_{b,min}$

(18)



$l_{bxy} = \left(\frac{723.52}{982}\right) \times 50 \times 25 = 920.9\ mm$

8.4.3(2) For U-bars with 50 mm cover top and bottom so as to fit comfortably inside the links, the largest practical radius is obtained by using a semi-circular bend (shape code 13). In this case, $r = 0.5h - (50 + \phi) = 0.5 \times 525 - (50 + 25) = 187.5\ mm$

10) Reinforcement detailing diagram:

01, 03 Bars in corners of links curtailed 50 mm from column face at each end to avoid clashing with column bars.

02 Loose bars positioned inside column bars. Bars lap 1000 mm with bar 01, 03 in adjacent span to provide continuity of internal tie.

04 Bar curtailment 150 mm from centreline of each end of span.

05 Loose U bar, positioned inside column bars. Upper leg extends 1400 mm beyond face of column & lower leg extends 1000 mm with bar 01

06, 09 Bars in corners of links curtailed 50 mm from column face

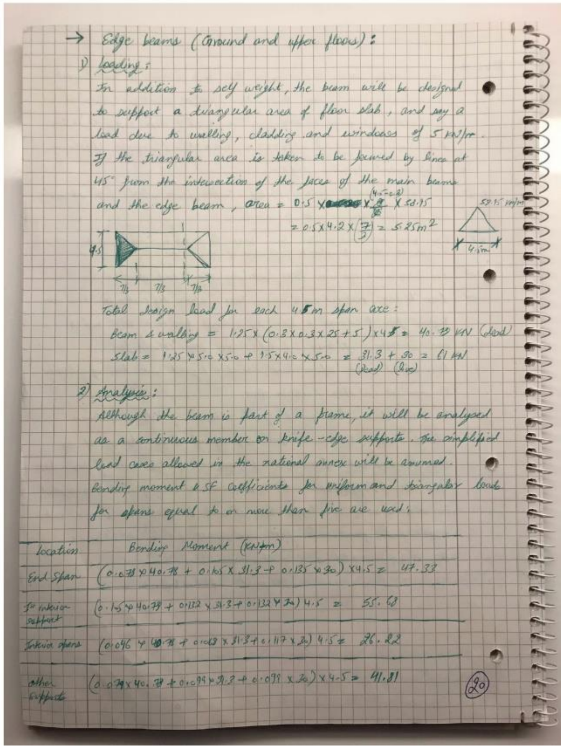
07. Bars placed inside column bars, extending beyond centre line of column 1900 mm into end span & interior span.

08 Bar positioned in slab extend beyond centreline of column 1050 mm

10 Closed links with 25 mm cover. Spacing of links determined by shear reinforcement, transverse reinforcement in lap zones of main bars – when dia of lapped bars $\phi \ge 20$, transverse bars of total area not less than area of one lapped bar should be provided within outer thirds of lap joint. For full zone, with $\phi = 20$ & allowing $A_{s,req} < A_{s,prov}$, total area of transverse bars

$A_{st} = 1.5 \times 315 \times \left(\frac{723}{982}\right) = 348\ mm^2$ (8H8 – 100)

(19)



→ Edge beams (Ground and upper floors):

1) Loading:

In addition to self weight, the beam will be designed to support a triangular area of floor slab, and say a load due to walling, cladding and windows of 5 kN/m. If the triangular area is taken to be formed by lines at 45° from the intersection of the faces of the main beams and the edge beam, area $= 0.5 \times \frac{(4.5 - 0.3)}{3} \times 58.75$

$= 0.5 \times 4.2 \times \left(\frac{7}{3}\right) = 5.25\ m^2$

4.5, 7/3, 7/3, 7/3, 58.75 kN/m, 4.5 m

Total design load for each 4.5 m span are:

Beam & walling $= 1.25 \times (0.3 \times 0.3 \times 25 + 5) \times 4.5 = 40.78$ kN (dead)

Slab $= 1.25 \times 5.0 \times 5.0 + 1.5 \times 4.0 \times 5.0 = 31.3 + 30 = 61$ kN
(dead) (live)

2) Analysis:

Although the beam is part of a frame, it will be analysed as a continuous member on knife-edge supports. The simplified load cases allowed in the national annex will be assumed. Bending moment & SF coefficients for uniform and triangular loads for spans equal to or more than five are used:

Location	Bending Moment (kNm)
End Span	$(0.078 \times 40.78 + 0.105 \times 31.3 + 0.135 \times 30) \times 4.5 = 47.33$
1st interior support	$(0.105 \times 40.78 + 0.132 \times 31.3 + 0.132 \times 30)\, 4.5 = 55.68$
Interior spans	$(0.046 \times 40.78 + 0.068 \times 31.3 + 0.117 \times 30)\, 4.5 = 26.22$
other supports	$(0.079 \times 40.78 + 0.094 \times 31.3 + 0.099 \times 30) \times 4.5 = 41.81$

(20)

Location	Shear force (kN)
End support	$(0.395 \times 40.78 + 0.359 \times 31.3 + 0.434 \times 30) = 40.56$
1st interior support	$(0.605 \times 40.78 + 0.621 \times 31.3 + 0.649 \times 34) = 63.84$
ditto (interior span)	$(0.626 \times 40.78 + 0.532 \times 31.3 + 0.622 \times 30) = 56.76$
Other supports	$(0.500 \times 40.78 + 0.500 \times 31.3 + 0.614 \times 30) = 54.46$

3) Flexure design:

At the supports, allowing for 16mm bars with 62 mm cover, so that the bar at the inside face of the edge beam passes below the U-bars in the main beam.

$d = 525 - (62 + \frac{16}{2}) = 455\text{mm} \approx 450\text{mm}$, $b = 300\text{mm}$

In the spans, allowing for 35mm cover with 8mm links & 16mm bars,

Say, $d = 475\text{mm}$, $b_{eff} = b_w + 0.2 \times 0.7l = 300 + (0.2 \times 0.7 \times 4500)$

$b_{eff} = 930$ mm

The required reinforcement is as follows:

*(5.3.2.1(3))

Location	$M/bd^2 f_{ck} = K$	z/d	Required A_s mm² $(M/0.87 f_{yk} \times z)$	Provided Bars	$z = 0.5 + \sqrt{0.25 - 0.882K}$, $z \le 0.95d$
End span (Pinned end)	$(47.33 \times 10^6 / 930 \times 475^2 \times 32) = 7.05 \times 10^{-3} < K(0.167)$	0.95	$(47.33 \times 10^6 / 0.87 \times 500 \times 451.3) = 241.9\ (mm^2)$	2H16 (402mm²)	$z = 0.5 + \sqrt{0.25 - 0.882 \times 7.05 \times 10^{-3}}$, $z = 0.99 \times d \le 0.95d$, ∴ $z = 0.95 \times 475 = \boxed{451.3}$
1st interior support	$58.68 \times 10^6 / 300 \times 475^2 \times 32 = 0.026$	0.95	$58.68 \times 10^6 / 0.87 \times 500 \times 451.3$ 273.63mm²	(2H20) 2H16 – 402mm² (Revised later)!	$0.5 + \sqrt{0.25 - 0.882 \times 0.026} = 0.97d \nleq 0.95d$, $z = 451.3$
Interior span	$26.22 \times 10^6 / 475^2 \times 930 \times 32 = 3.91 \times 10^{-3}$	0.95 (451.3)	$26.22 \times 10^6 / 0.87 \times 500 \times 451.3 = 133.56\ mm^2$	2H10 (226mm²)	$0.5 + \sqrt{0.25 - 0.882 \times 3.91 \times 10^{-3}} = 0.99d \nleq 0.95d$
Other interior supports	$\frac{41.91 \times 10^6}{475^2 \times 300 \times 32} = 0.0193$	0.95 (451.3)	$\frac{41.91 \times 10^6}{0.87 \times 500 \times 451.3} = 212.97\ mm^2$	2H10 (226 mm²) (Revised later)! (2H12)	$0.5 + \sqrt{0.25 - 0.882 \times 0.026} = 0.97d \nleq 0.95d$

(21)

5.3.2.1(3) At the interior supports, the tension flange is considered to extend beyond the side face of the beam for a distance given by $b_{eff} = 0.2 \times 0.2 \times l = 0.04 \times 4500 = 180$ mm. The 2H16 @ first interior support & 2H10 @ other interior support will be positioned inside the links with an additional H10 in the flange.

4) Shear Design:

Minimum requirements for vertical links are given by:

9.2.2(5) $A_{sw}/s = (0.08\sqrt{f_{ck}})\, b_w/f_{yk} = (0.08\sqrt{32}) \times 300/500 = 0.27\ mm^2/mm$

9.2.2(6) $s \le 0.75d = 0.75 \times 380 = 285\ mm$

The H8 links @ 275mm provides 0.36 mm²/mm. The design shear resistance from links would be:

$V_{Rd,s} = \left(\frac{A_{sw}}{s}\right) f_{ywd}\, z \cot\theta = 0.36 \times 0.87 \times 500 \times 0.9 \times 475 \times 2.5 \times 10^{-3}$

$V_{Rd,s} = 167.4$ kN > All shear forces on beam

Hence, links are appropriate & safe. ✓

5) Deflection:

Since the loading and the span are both less than those for the main beams, there is no need to check for deflection requirements!

6) Cracking (Skip it)!

7) Detailing requirements:

- Minimum area of longitudinal tension reinforcement:

$A_{s,min} = 0.26 \left(\frac{f_{ctm}}{f_{yk}}\right) b_t d = 0.26 \left(\frac{0.3 \times 32^{2/3}}{500}\right) b_t d \ge 0.0013\, b_t d$ $(0.00156\, b_t d)$

For interior support regions:

$A_{s,min} = 0.00156 \times 475 \times (263 + 390) = 484.06\ mm^2 > 457.55\ mm^2$

$A_{s,min} > A_{s,prov}$ (402 for 1st interior span & 226 for others)

(22)

∴ Revise the provided reinforcement to 2H12 (628 mm²) for all interior supports!

For span region,

$A_{s,min} = 0.00156 \times 300 \times 475 = 222.3\ mm^2 > 185.25\ mm^2$

$A_{s,min} < A_{s,prov}$ (402 & 226 mm²), Hence, Safe ✓

- At the bottom of the beam, 2H16 in end span and 2H10 in
9.2.1.5(1) interior span will be provided for the length of span. At the support, where at least 25% of the area in the span is needed, 2H10 will be provided with a lap length of 800 mm.
- At the top of the beam, 2H10 will be provided to support the links for the length of each span. At the end support where the columns provide partial fixity, 2H10 will be provided. At each support, the bars should extend beyond the face of the support far enough to provide a lap length. Assuming poor bond conditions, Projection from face of support $= (50 + 1.5 \times 50 \times \underset{(\phi)}{10}) = 800$ mm
∴ At all support, extend top bars for 800 mm from face of support.

- Tying requirements:

The longitudinal reinforcement can be used to resist the
9.10.2.3 Table NA.1 peripheral tie force,
9.10.1(4) $F_{tie,per} = \cancel{\ldots}\ l_i \times q_1 \ge Q_2$

$q_1 = 10$ kN/m $\quad F_{tie,per} = 4.5 \times 10 = 45$ kN > 70 kN

$Q_2 = 70$ kN $\quad$ Hence, take $F_{tie,per} = 70$ kN

Minimum area of reinforcement required with $\sigma_s = 500$ MPa

$A_{s,min} = 70 \times \frac{1000}{500} = 140\ mm^2$, Use 1H16 bar

The bar on the inside face at the top of the beam will form the peripheral tie.

(23)

→ Column design:

1) Actions on column:

For the column on line B, the sub frame analysis results shown on calculation sheet (5-7) previously gives beam shears & column moments for three load cases, and apply at 2nd, 3rd and 4th floor levels and also for lower floor levels. At roof level, the sub-frame and the loading are significantly different, and another analysis is required. Loading details are below:

Characteristic loading for roof slab,

Slab & finishes $= (3.75 + 1.5) = 5.25$ kN/m² $\quad$ Self weight of slab $\} = 3.75$ kN/m²

Imposed load $= 0.6$ kN/m²

Design ultimate load $= n = 1.25 \times 5.25 + 1.5 \times 0.6 = 7.5$ kN/m²

Design UL for first interior roof beam:

$1.1 \times 4.5 \times 7.5 + 2.8 = 39.93$ kN/m (max.) &

$1.1 \times 4.5 \times (1.25 \times 5 (d.L)) + 2.2 = 33.74$ kN/m (minimum)

Load per storey due to self weight of column:

Columns $= 1.25 \times 0.3 \times 0.3 \times 25 \times \underset{(\text{Storey height} - h \text{ of beam})}{(3.5 - 0.525)} = 8.4$ kN

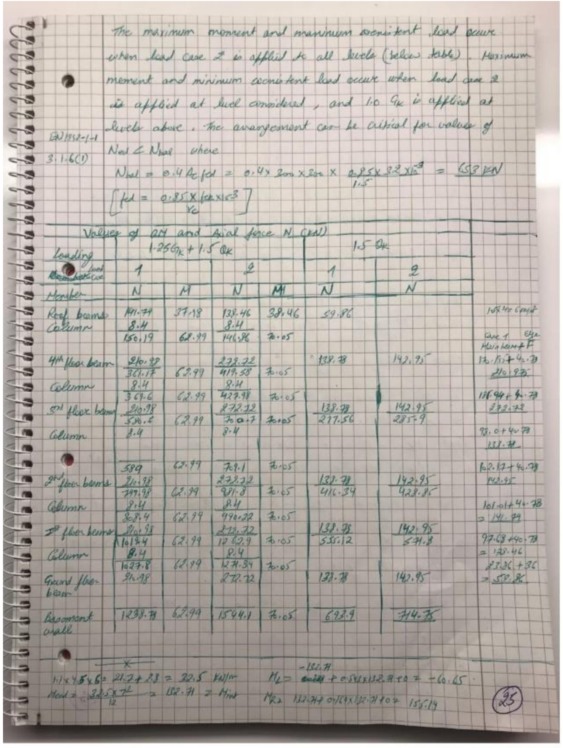
1991-1-1 6.3.1.2 A reduction may be made in the total imposed floor load, according to the number of storeys being supported at the level considered. For up to five storeys, this load may be multiplied by $\alpha_n = 1.1 - n/10$, n = number of storey.

→ External Column B1:

At each level, the load applied is the sum of the shear force for the main beam at line 1, and the uniform load only on the edge beam. Thus, at each floor, the load from the edge beam is $F = 40.78$ kN (Edge beam & walling). At roof, the load due to the self weight of the beam & parapet is

$F = 1.25 \times (0.3 \times 0.3 + 0.15 \times 1.0) \times 25 \times 4.5 = 33.75$ kN

(24)



The maximum moment and maximum consistent load occur when load case 2 is applied to all levels (below table). Maximum moment and minimum consistent load occur when load case 2 is applied at level considered, and 1.0 G_k is applied at levels above. The arrangement can be critical for values of

EN1992-1-1 3.1.6(1) $N_{ed} < N_{bal}$ where

$N_{bal} = 0.4 A_c f_{cd} = 0.4 \times 300 \times 300 \times \frac{0.85 \times 32 \times 10^{-3}}{1.5} = 653$ kN

$\left[f_{cd} = \frac{0.85 \times f_{ck} \times 10^{-3}}{\gamma_c} \right]$

Values of BM and Axial force N (kN)

Loading	$1.25G_k + 1.5Q_k$				$1.5Q_k$		
Load case	1		2		1	2	
Member	N	M	N	M	N	N	
Roof beams	141.79	37.18	138.46	38.46	59.86		1.5×4×6.6
Column	8.4		8.4				
	150.19	62.99	146.86	70.05			Case 1 Max beam F
4th floor beam	210.98		272.72		138.78	142.95	170.197+40.78
	361.17	62.99	419.58	70.05			210.975
Column	8.4		8.4				
	369.6	62.99	427.98	70.05			186.94+40.78
3rd floor beams	210.98		272.72		138.78	142.95	272.72
	580.6	62.99	700.7	70.05	277.56	285.9	
Column	8.4		8.4				98.0+40.78
							138.78
	589	62.99	709.1	70.05			102.17+40.78
2nd floor beams	210.98		272.72		138.78	142.95	142.95
	799.98	62.99	981.8	70.05	416.34	428.85	
Column	8.4		8.4				101.01+40.78
	808.4	62.99	990.22	70.05			= 141.79
1st floor beams	210.98		272.72		138.78	142.95	
	1019.4	62.99	1262.9	70.05	555.12	571.8	97.68+40.78
Column	8.4		8.4				= 138.46
	1027.8	62.99	1271.34	70.05			23.86+36
Ground floor beam	210.98		272.72		138.78	142.95	= 59.86
Basement wall	1238.78	62.99	1544.1	70.05	693.9	714.75	

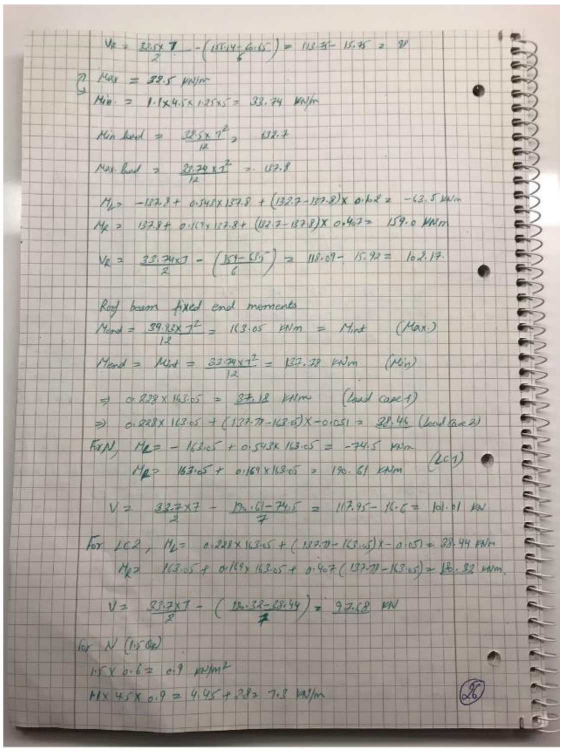
$1.1 \times 4.5 \times 6 = 21.7 + 28 = 32.5$ kN/m

$M_{end} = \frac{32.5 \times 7^2}{12} = 132.71 = M_{int}$

$M_L = -132.71 + 0.543 \times 132.71 + 0 = -60.65$

$M_R = 132.71 + 0.169 \times 132.71 + 0 = 155.14$

(25)



$V_R = \frac{32.5 \times 7}{2} - \left(\frac{155.14 + 60.65}{6} \right) = 113.75 - 15.75 = 98$

Max $= 32.5$ kN/m

Min $= 1.1 \times 4.5 \times 1.25 \times 5 = 33.74$ kN/m

Min load $= \frac{32.5 \times 7^2}{12} = 132.7$

Max load $= \frac{33.74 \times 7^2}{12} = 137.8$

$M_L = -137.8 + 0.543 \times 137.8 + (132.7 - 137.8) \times 0.102 = -63.5$ kNm

$M_R = 137.8 + 0.169 \times 137.8 + (132.7 - 137.8) \times 0.407 = 159.0$ kNm

$V_R = \frac{33.74 \times 7}{2} - \left(\frac{159 - 63.5}{6} \right) = 118.09 - 15.92 = 102.17$

Roof beam fixed end moments

$M_{end} = \frac{39.93 \times 7^2}{12} = 163.05$ kNm $= M_{int}$ (Max)

$M_{end} = M_{int} = \frac{33.74 \times 7^2}{12} = 137.78$ kNm (Min)

$\Rightarrow 0.228 \times 163.05 = \underline{37.18}$ kNm (load case 1)

$\Rightarrow 0.228 \times 163.05 + (137.78 - 163.05) \times -0.051 = \underline{38.46}$ (load case 2)

For N, $M_L = -163.05 + 0.543 \times 163.05 = -74.5$ kNm

$M_R = 163.05 + 0.169 \times 163.05 = 190.61$ kNm (LC1)

$V = \frac{32.7 \times 7}{2} - \frac{190.61 - 74.5}{7} = 117.95 - 16.6 = 101.01$ kN

For LC2, $M_L = 0.228 \times 163.05 + (137.78 - 163.05) \times -0.051 = 38.44$ kNm

$M_R = 163.05 + 0.169 \times 163.05 + 0.407(137.78 - 163.05) = 180.32$ kNm

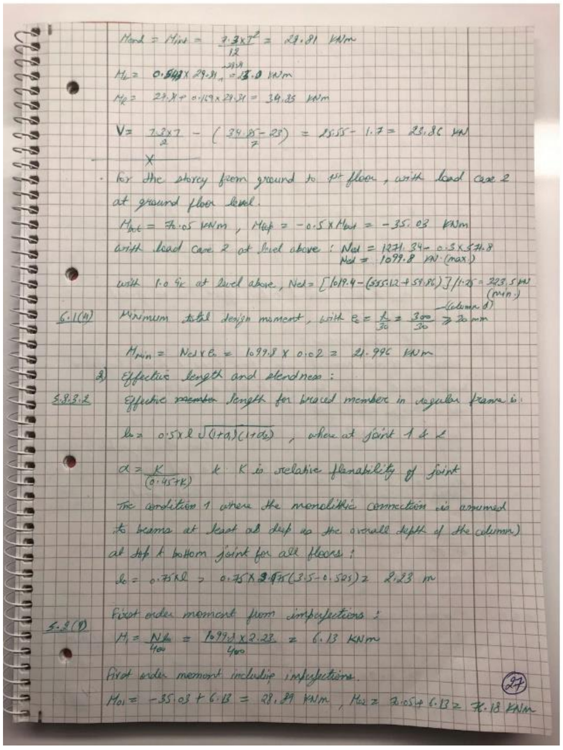
$V = \frac{33.7 \times 7}{2} - \left(\frac{180.32 - 38.44}{7} \right) = \underline{97.68}$ kN

for N ($1.5 Q_k$)

$1.5 \times 0.6 = 0.9$ kN/m^2

$1.1 \times 4.5 \times 0.9 = 4.45 + 2.8 = 7.3$ kN/m

(26)



$M_{end} = M_{int} = \frac{7.3 \times 7^2}{12} = 29.81$ kNm

$M_L = 0.543 \times 29.81 = 13.0$ kNm

$M_R = 29.81 + 0.169 \times 29.81 = 34.85$ kNm

$V = \frac{7.3 \times 7}{2} - \left(\frac{34.85 - 23}{7} \right) = 25.55 - 1.7 = 23.86$ kN

For the storey from ground to 1st floor, with load case 2 at ground floor level.

$M_{bot} = 70.05$ kNm, $M_{top} = -0.5 \times M_{bot} = -35.03$ kNm

with load case 2 at level above: $N_{ed} = 1271.34 - 0.3 \times 571.8$

$N_{ed} = 1099.8$ kN (max)

with 1.0 G_k at level above, $N_{ed} = [1019.4 - (555.12 + 59.86)]/1.25 = 323.5$ kN (min)

6.1(4) Minimum total design moment, with $e_0 = \frac{h}{30} = \frac{300}{30} \geq 20$ mm (column d)

$M_{min} = N_{ed} \times e_0 = 1099.8 \times 0.02 = 21.996$ kNm

2) Effective length and slendness:

5.8.3.2 Effective ~~member~~ length for braced member in regular frame is:

$l_0 = 0.5 \times l \sqrt{(1+\alpha_1)(1+\alpha_2)}$, where at joint 1 & 2

$\alpha = \frac{K}{(0.45+K)}$ & K is relative flexibility of joint

The condition 1 where the monolithic connection is assumed to beams at least as deep as the overall depth of the column) at top & bottom joint for all floors:

$l_0 = 0.75 \times l = 0.75 \times 2.975(3.5 - 0.525) = 2.23$ m

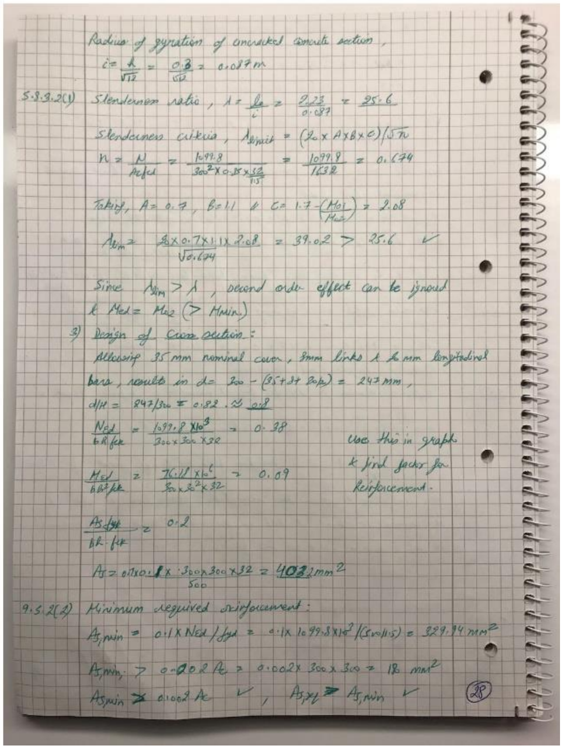
First order moment from imperfections:

5.2(7) $M_i = \frac{N l_0}{400} = \frac{1099.8 \times 2.23}{400} = 6.13$ kNm

First order moment including imperfections.

$M_{01} = -35.03 + 6.13 = 28.89$ kNm, $M_{02} = 70.05 + 6.13 = 76.18$ kNm

(27)



Radius of gyration of uncracked concrete section,

$i = \frac{h}{\sqrt{12}} = \frac{0.3}{\sqrt{12}} = 0.087$ m

5.8.3.2(1) Slenderness ratio, $\lambda = \frac{l_0}{i} = \frac{2.23}{0.087} = \underline{25.6}$

Slenderness criteria, $\lambda_{limit} = (20 \times A \times B \times C)/\sqrt{n}$

$n = \frac{N}{A_c f_{cd}} = \frac{1099.8}{300^2 \times 0.85 \times \frac{32}{1.5}} = \frac{1099.8}{1632} = 0.674$

Taking, $A = 0.7$, $B = 1.1$ & $C = 1.7 - \left(\frac{M_{01}}{M_{02}} \right) = 2.08$

$\lambda_{lim} = \frac{20 \times 0.7 \times 1.1 \times 2.08}{\sqrt{0.674}} = 39.02 > 25.6$ ✓

Since $\lambda_{lim} > \lambda$, second order effect can be ignored & $M_{ed} = M_{02}$ ($> M_{min}$)

3) Design of Cross section:

Allowing 35 mm nominal cover, 8mm links & 20 mm longitudinal bars, results in $d = 300 - (35 + 8 + 20/2) = 247$ mm,

$d/H = 247/300 = 0.82 \simeq \underline{0.8}$

$\frac{N_{Ed}}{bh f_{ck}} = \frac{1099.8 \times 10^3}{300 \times 300 \times 32} = 0.38$

$\frac{M_{Ed}}{bh^2 f_{ck}} = \frac{76.18 \times 10^6}{300 \times 300^2 \times 32} = 0.09$

Use this in graph & find factor for Reinforcement.

$\frac{A_s f_{yk}}{bh \cdot f_{ck}} = 0.2$

$A_s = 0.2 \times 0.8 \times \frac{300 \times 300 \times 32}{500} = \underline{403.2}$ mm^2

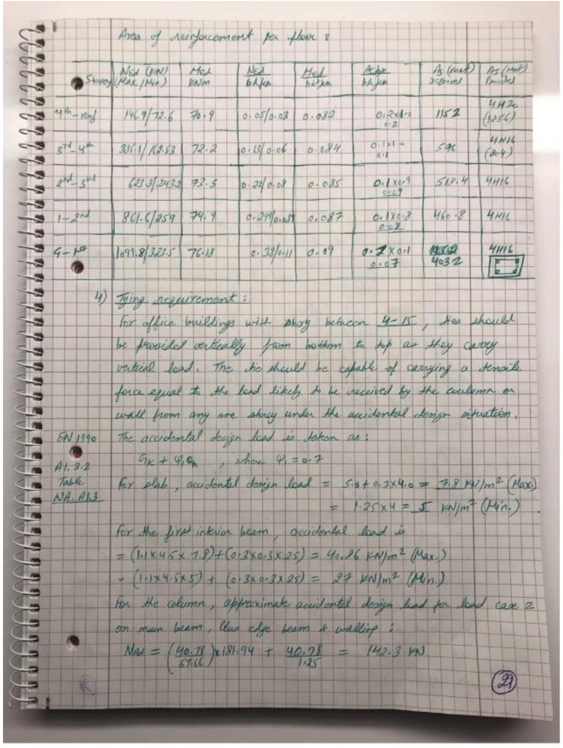
9.5.2(2) Minimum required reinforcement:

$A_{s,min} = 0.1 \times N_{Ed}/f_{yd} = 0.1 \times 1099.8 \times 10^3/(500/1.15) = 329.94$ mm^2

$A_{s,min} > 0.002 A_c = 0.002 \times 300 \times 300 = 180$ mm^2

$A_{s,min} > 0.002 A_c$ ✓, $A_{s,req} > A_{s,min}$ ✓

(28)



Area of reinforcement per floor:

Storey	N_{Ed} (kN) (Max./Min.)	M_{Ed} kNm	$\frac{N_{Ed}}{bhf_{ck}}$	$\frac{M_{Ed}}{bh^2f_{ck}}$	$\frac{A_sf_{yk}}{bhf_{ck}}$	A_s (mm²) required	A_s (mm²) Provided
4th – roof	146.9/72.6	70.9	0.05/0.03	0.082	0.2×1.0 = 0.2	1152	4H20 (1256)
3rd – 4th	305.1/162.53	72.2	0.13/0.06	0.084	0.1×1 = 0.1	576	4H16 (804)
2nd – 3rd	623.3/243.2	73.5	0.22/0.08	0.085	0.1×0.9 = 0.09	518.4	4H16
1 – 2nd	861.6/259	74.9	0.29/0.087	0.087	0.1×0.8 = 0.08	460.8	4H16
G – 1st	1099.8/323.5	76.18	0.38/0.11	0.09	0.7×0.1 = 0.07	403.2	4H16

4) Tying requirement:

For office buildings with story between 4–15, ties should be provided vertically from bottom to top as they carry vertical load. The tie should be capable of carrying a tensile force equal to the load likely to be received by the column or wall from any one storey under the accidental design situation.

EN 1990 A1.3.2 Table NA.A1.3 The accidental design load is taken as:

$G_k + \psi_1 Q_k$, where $\psi_1 = 0.7$

For slab, accidental design load $= 5.0 + 0.7\times4.0 = 7.8\ kN/m^2$ (Max.)

$= 1.25\times4 = 5\ kN/m^2$ (Min.)

For the first interior beam, accidental load is

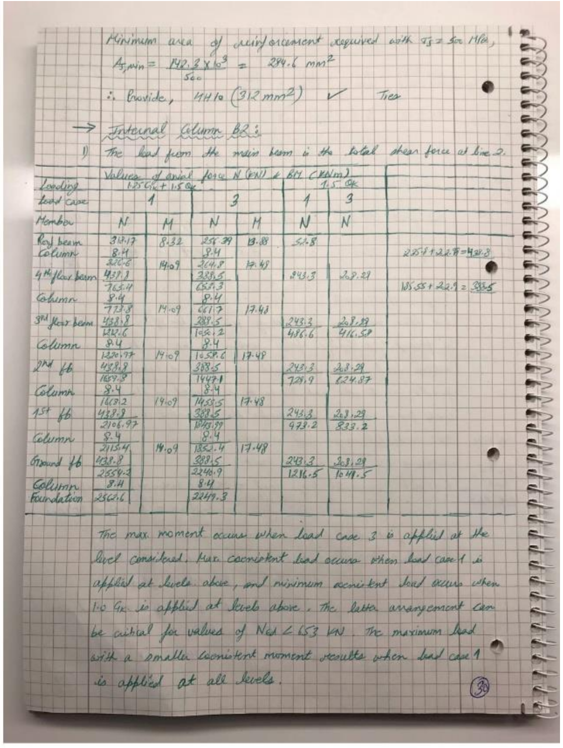
$= (1.1\times4.5\times7.8)+(0.3\times0.3\times25) = 40.86\ kN/m^2$ (Max.)

$= (1.1\times4.5\times5)+(0.3\times0.3\times25) = 27\ kN/m^2$ (Min.)

For the column, approximate accidental design load for load case 2 on main beam, plus edge beam & walling:

$N_{Ad} = \left(\frac{40.78}{67.66}\right)\times181.94 + \frac{40.78}{1.25} = 142.3\ kN$

29



Minimum area of reinforcement required with $f_y = 500$ MPa,

$A_{s,min} = \frac{142.3\times10^3}{500} = 284.6\ mm^2$

∴ Provide, 4H10 (312 mm²) ✓ Ties

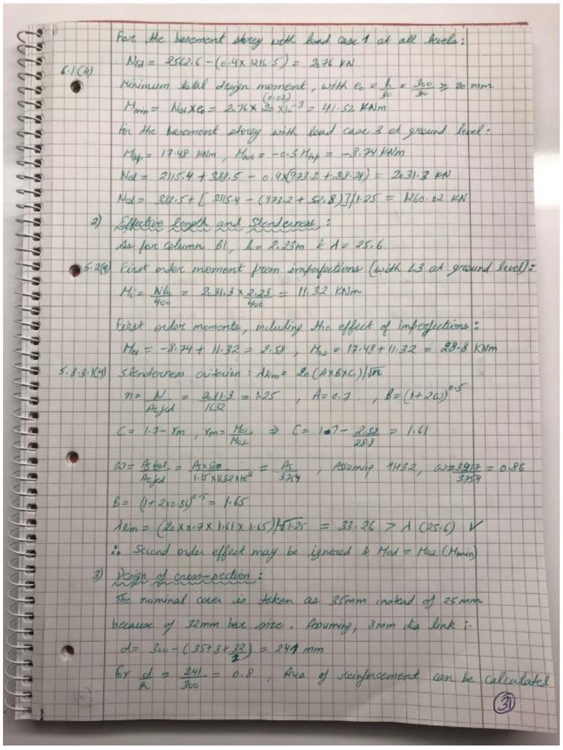
→ Internal Column B2:

1) The load from the main beam is the total shear force at line 2.

Loading	Values of axial force N (kN) & BM (kNm) 1.25 G_k + 1.5 Q_k				1.5 Q_k		
Load case	1		3		1	3	
Member	N	M	N	M	N	N	
Roof beam	318.17	8.32	256.39	13.88	52.8		
Column	8.4		8.4				225.8 + 22.9 = 438.8
	326.6	14.09	264.8	17.48			
4th floor beam	438.8		388.5		243.3	208.29	
	765.4		653.3				185.55 + 22.9 = 388.5
Column	8.4		8.4				
	773.8	14.09	661.7	17.48			
3rd floor beam	438.8		388.5		243.3	208.29	
	1212.6		1050.2		486.6	416.58	
Column	8.4		8.4				
	1220.97	14.09	1058.6	17.48			
2nd fb	438.8		388.5		243.3	208.29	
	1659.8		1447.1		729.9	624.87	
Column	8.4		8.4				
	1668.2	14.09	1455.5	17.48			
1st fb	438.8		388.5		243.3	208.29	
	2106.97		1843.99		973.2	833.2	
Column	8.4		8.4				
	2115.4	14.09	1852.4	17.48			
Ground fb	438.8		388.5		243.3	208.29	
	2554.2		2240.9		1216.5	1041.5	
Column	8.4		8.4				
Foundation	2562.6		2249.3				

The max. moment occurs when load case 3 is applied at the level considered. Max. coexistent load occurs when load case 1 is applied at levels above, and minimum coexistent load occurs when 1.0 G_k is applied at levels above. The latter arrangement can be critical for values of $N_{Ed} < 653$ kN. The maximum load with a smaller coexistent moment results when load case 1 is applied at all levels.

30



For the basement storey with load case 1 at all levels:

6.1(4) $N_{Ed} = 2562.6 - (0.4\times1216.5) = 2076\ kN$

Minimum total design moment, with $e_0 = \frac{h}{30} = \frac{300}{30} \geq 20$ mm

$M_{min} = N_{Ed}\times e_0 = 2076\times\underset{(0.02)}{20}\times10^{-3} = 41.52\ kNm$

For the basement storey with load case 3 at ground level:

$M_{top} = 17.48\ kNm$, $M_{bot} = -0.5M_{top} = -8.74\ kNm$

$N_{Ed} = 2115.4 + 388.5 - 0.4\times(973.2 + 208.29) = 2031.3\ kN$

$N_{Ed} = 388.5 + [2115.4 - (973.2 + 52.8)]/1.25 = 1260.02\ kN$

2) Effective length and Slenderness:

As for column B1, $l_0 = 2.23$ m & $\lambda = 25.6$

5.2(9) First order moment from imperfections (with L3 at ground level):

$M_i = \frac{Nl_0}{400} = 2031.3\times\frac{2.23}{400} = 11.32\ kNm$

First order moments, including the effect of imperfections:

$M_{01} = -8.74 + 11.32 = 2.58$, $M_{02} = 17.48 + 11.32 = 28.8\ kNm$

5.8.3.1(4) Slenderness criterion: $\lambda_{lim} = 20(A\times B\times C)/\sqrt{n}$

$n = \frac{N}{A_cf_{cd}} = \frac{2031.3}{1632} = 1.25$, $A = 0.7$, $B = (1+2\omega)^{0.5}$

$C = 1.7 - r_m$, $r_m = \frac{M_{01}}{M_{02}} \Rightarrow C = 1.7 - \frac{2.58}{28.8} = 1.61$

$\omega = \frac{A_sf_{yd}}{A_cf_{cd}} = \frac{A_s\times500}{1.15\times1632\times10^3} = \frac{A_s}{3754}$, Assuming 4H32, $\omega = \frac{3217}{3754} = 0.86$

$B = (1+2\times0.86)^{0.5} = 1.65$

$\lambda_{lim} = (20\times0.7\times1.61\times1.65)/\sqrt{1.25} = 33.26 > \lambda\ (25.6)$ ✓

∴ Second order effect may be ignored & $M_{Ed} = M_{02}$ (M_{min})

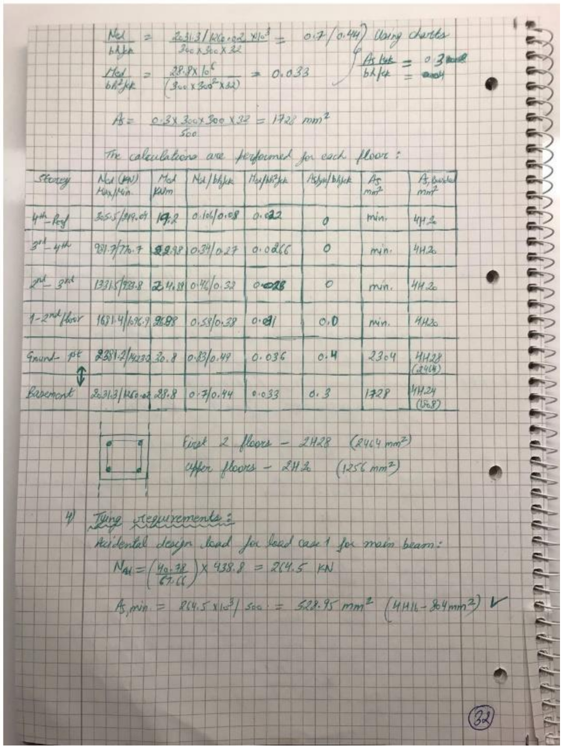
3) Design of cross-section:

The nominal cover is taken as 35mm instead of 25mm because of 32mm bar size. Assuming, 8mm dia link:

$d = 300 - (35 + 8 + \frac{32}{2}) = 241$ mm

For $\frac{d}{h} = \frac{241}{300} = 0.8$, Area of reinforcement can be calculated

31



$\frac{N_{Ed}}{bhf_{ck}} = \frac{2031.3/1260.02\times10^3}{300\times300\times32} = 0.7/0.44$ Using charts $\frac{A_sf_{yk}}{bhf_{ck}} = 0.3$

$\frac{M_{Ed}}{bh^2f_{ck}} = \frac{28.8\times10^6}{(300\times300^2\times32)} = 0.033$

$A_s = \frac{0.3\times300\times300\times32}{500} = 1728\ mm^2$

The calculations are performed for each floor:

Storey	N_{Ed} (kN) Max/Min	M_{Ed} kNm	N_{Ed}/bhf_{ck}	M_{Ed}/bh^2f_{ck}	A_sf_{yk}/bhf_{ck}	A_s mm²	A_s, provided mm²
4th – Roof	305.5/219.04	19.2	0.106/0.08	0.022	0	min.	4H20
3rd – 4th	981.7/776.7	22.98	0.34/0.27	0.0266	0	min.	4H20
2nd – 3rd	1331.5/923.8	24.88	0.46/0.32	0.028	0	min.	4H20
1 – 2nd floor	1681.4/1096.9	26.88	0.58/0.38	0.031	0.0	min.	4H20
Ground – 1st	2381.2/1423.2	30.8	0.83/0.49	0.036	0.4	2304	4H28 (2464)
Basement	2031.3/1260.02	28.8	0.7/0.44	0.033	0.3	1728	4H24 (1808)

First 2 floors – 2H28 (2464 mm²)

Upper floors – 2H20 (1256 mm²)

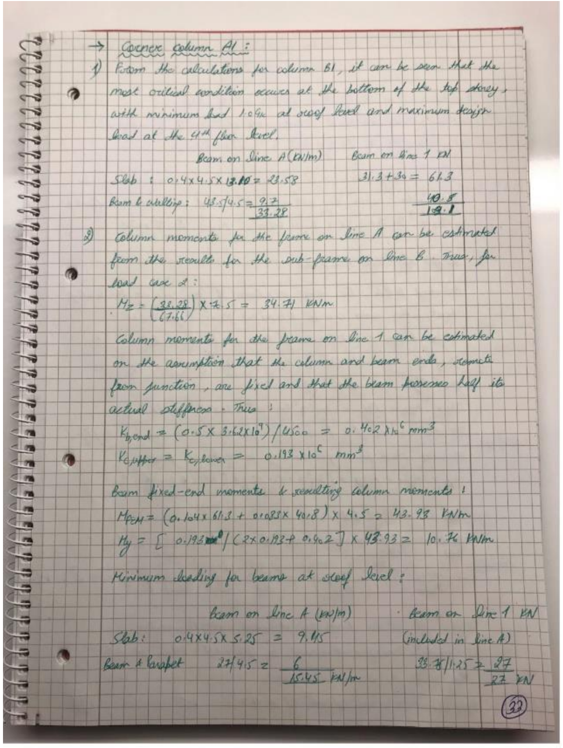
4) Tying requirements:

Accidental design load for load case 1 for main beam:

$N_{Ad} = \left(\frac{40.78}{67.66}\right)\times438.8 = 264.5\ kN$

$A_{s,min} = 264.5\times10^3/500 = 528.95\ mm^2$ (4H16 – 804 mm²) ✓

32



→ Corner column A1 :

1) From the calculations for column B1, it can be seen that the most critical condition occurs at the bottom of the top storey, with minimum load 1.0g_k at roof level and maximum design load at the 4th floor level.

	Beam on line A (KN/m)	Beam on line 1 KN
Slab :	0.4x4.5x13.10 = 23.58	31.3+30 = 61.3
Beam & cladding :	43.5/4.5 = 9.7	40.8
	33.28	102.1

2) Column moments for the frame on line A can be estimated from the results for the sub-frame on line B. Thus, for load case 2 :

$M_z = \left(\frac{33.28}{67.66}\right) \times 70.5 = 34.71$ KNm

Column moments for the frame on line 1 can be estimated on the assumption that the column and beam ends, remote from junction, are fixed and that the beam possesses half its actual stiffness. Thus :

$K_{b,end} = (0.5 \times 3.62 \times 10^9)/4500 = 0.402 \times 10^6$ mm³

$K_{c,upper} = K_{c,lower} = 0.193 \times 10^6$ mm³

Beam fixed-end moments & resulting column moments :

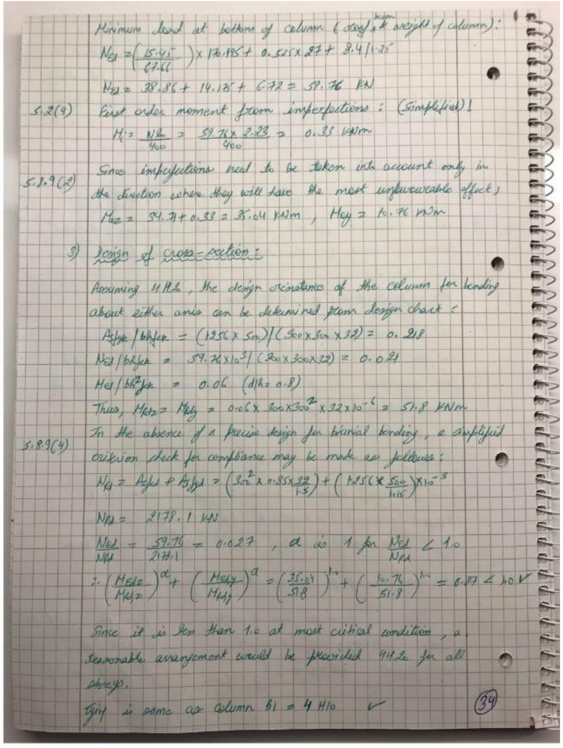
$M_{FEM} = (0.104 \times 61.3 + 0.083 \times 40.8) \times 4.5 = 43.93$ KNm

$M_y = [\,0.193 / (2 \times 0.193 + 0.402\,] \times 43.93 = 10.76$ KNm

Minimum loading for beams at roof level :

	Beam on line A (KN/m)	Beam on line 1 KN
Slab :	0.4x4.5x5.25 = 9.45	(included in line A)
Beam & Parapet	27/4.5 = 6	33.75/1.25 = 27
	15.45 KN/m	27 KN

(33)



Minimum load at bottom of column (roof + self weight of column):

$N_{Ed} = \left(\frac{15.45}{67.66}\right) \times 170.195 + 0.525 \times 27 + 8.4/1.25$

$N_{Ed} = 38.86 + 14.175 + 6.72 = 59.76$ KN

5.2(9) First order moment from imperfections : (Simplified) !

$M_i = \frac{N l_0}{400} = \frac{59.76 \times 2.23}{400} = 0.33$ KNm

5.8.9(2) Since imperfections need to be taken into account only in the direction where they will have the most unfavourable effect ;

$M_{Ez} = 34.71 + 0.33 = 35.04$ KNm , $M_{Ey} = 10.76$ KNm

3) Design of cross-section :

Assuming 4H20, the design resistance of the column for bending about either axis can be determined from design chart :

$A_s f_{yk} / bh f_{ck} = (1256 \times 500)/(300 \times 300 \times 32) = 0.218$

$N_{Ed} / bh f_{ck} = 59.76 \times 10^3 / (300 \times 300 \times 32) = 0.021$

$M_{Ed} / bh^2 f_{ck} = 0.06 \quad (d/h = 0.8)$

Thus, $M_{Rdz} = M_{Rdy} = 0.06 \times 300 \times 300^2 \times 32 \times 10^{-6} = 51.8$ KNm

5.8.9(4) In the absence of a precise design for biaxial bending, a simplified criterion check for compliance may be made as follows :

$N_{Rd} = A_c f_{cd} + A_s f_{yd} = \left(300^2 \times 0.85 \times \frac{32}{1.5}\right) + \left(1256 \times \frac{500}{1.15}\right) \times 10^{-3}$

$N_{Rd} = 2178.1$ KN

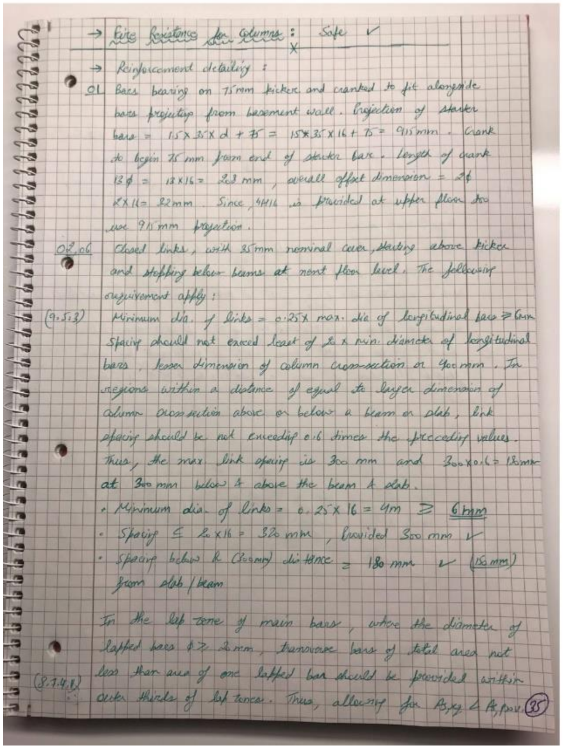
$\frac{N_{Ed}}{N_{Rd}} = \frac{59.76}{2178.1} = 0.027$, a is 1 for $\frac{N_{Ed}}{N_{Rd}} < 1.0$

$\therefore \left(\frac{M_{Edz}}{M_{Rdz}}\right)^a + \left(\frac{M_{Edy}}{M_{Rdy}}\right)^a = \left(\frac{35.04}{51.8}\right)^{1.0} + \left(\frac{10.76}{51.8}\right)^{1.0} = 0.87 < 1.0$ ✓

Since it is less than 1.0 at most critical condition, a reasonable arrangement would be provided 4H20 for all storeys.

Tying is same as column B1 = 4 H10 ✓

(34)



→ Fire Resistance for Columns : Safe ✓

→ Reinforcement detailing :

01 Bars bearing on 75mm kicker and cranked to fit alongside bars projecting from basement wall. Projection of starter bars = 1.5 x 35 x d + 75 = 1.5 x 35 x 16 + 75 = 915 mm. Crank to begin 75 mm from end of starter bar. Length of crank 13ϕ = 13 x 16 = 208 mm, overall offset dimension = 2ϕ 2 x 16 = 32 mm. Since, 4H16 is provided at upper floor too use 915 mm projection.

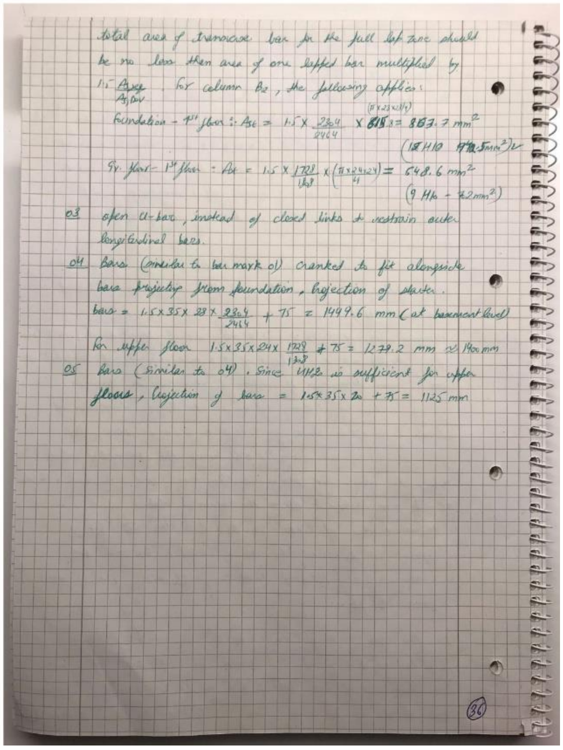
02, 06 Closed links, with 35mm nominal cover, starting above kicker and stopping below beams at next floor level. The following requirement apply :

(9.5.3) Minimum Dia. of links = 0.25 x max. dia of longitudinal bars ≥ 6mm. Spacing should not exceed least of 20 x min. diameter of longitudinal bars, lesser dimension of column cross-section or 400 mm. In regions within a distance of equal to larger dimension of column cross-section above or below a beam or slab, link spacing should be not exceeding 0.6 times the preceding values. Thus, the max. link spacing is 300 mm and 300 x 0.6 = 180 mm at 300 mm below & above the beam & slab.

- Minimum dia. of links = 0.25 x 16 = 4m ≥ 6 mm
- Spacing ≤ 20 x 16 = 320 mm, Provided 300 mm ✓
- Spacing below & (300mm) distance = 180 mm ✓ (150 mm) from slab / beam

(8.7.4.1) In the lap zone of main bars, where the diameter of lapped bars ϕ ≥ 20 mm, transverse bars of total area not less than area of one lapped bar should be provided within outer thirds of lap zones. Thus, allowing for $A_{s,req}$ & $A_{s,prov}$

(35)



total area of transverse bar for the full lap zone should be no less than area of one lapped bar multiplied by $1.5 \frac{A_{s,req}}{A_{s,prov}}$. For column B2, the following applies :

Foundation - 4th floor : $A_{st} = 1.5 \times \frac{2304}{2464} \times 615.8 = 863.7$ mm² $(\pi \times 28 \times 28/4)$

(12 H10 ~~942.5 mm²~~) ✓

Gr. floor - 1st floor : $A_{st} = 1.5 \times \frac{1728}{1808} \times \left(\frac{\pi \times 24 \times 24}{4}\right) = 648.6$ mm²

(9 H10 - 702 mm²)

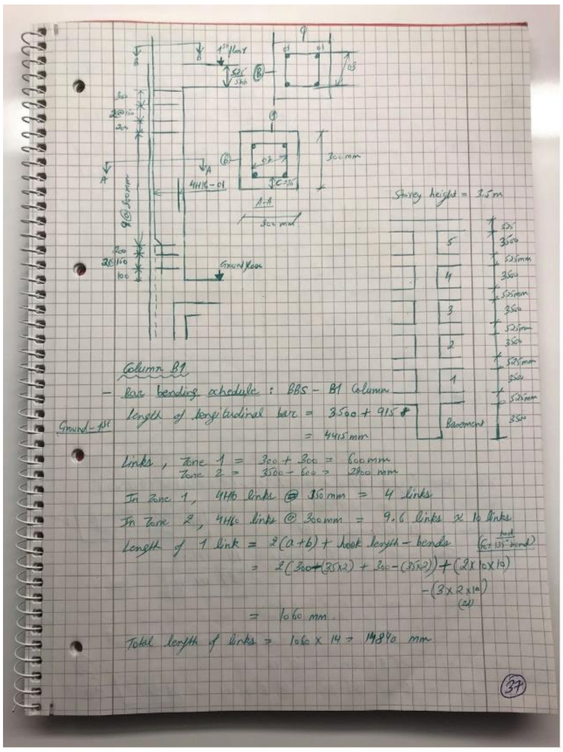
03 open U-bar, instead of closed links to restrain outer longitudinal bars.

04 Bars (similar to bar mark 01) cranked to fit alongside bars projecting from foundation, projection of starter bars = 1.5 x 35 x 28 x $\frac{2304}{2464}$ + 75 = 1449.6 mm (at basement level)

for upper floor 1.5 x 35 x 24 x $\frac{1728}{1808}$ + 75 = 1279.2 mm ≈ 1400 mm

05 Bars (similar to 04). Since 4H20 is sufficient for upper floors, projection of bars = 1.5 x 35 x 20 + 75 = 1125 mm

(36)



Storey height = 3.5m

Column B1

- Bar bending schedule : BBS - B1 Column

Ground - 1st

Length of longitudinal bar = 3500 + 915 = 4415 mm

Links , Zone 1 = 300 + 300 = 600 mm
Zone 2 = 3500 - 600 = 2900 mm

In Zone 1, 4H10 links @ 150 mm = 4 links

In Zone 2, 4H10 links @ 300 mm = 9.6 links ≈ 10 links

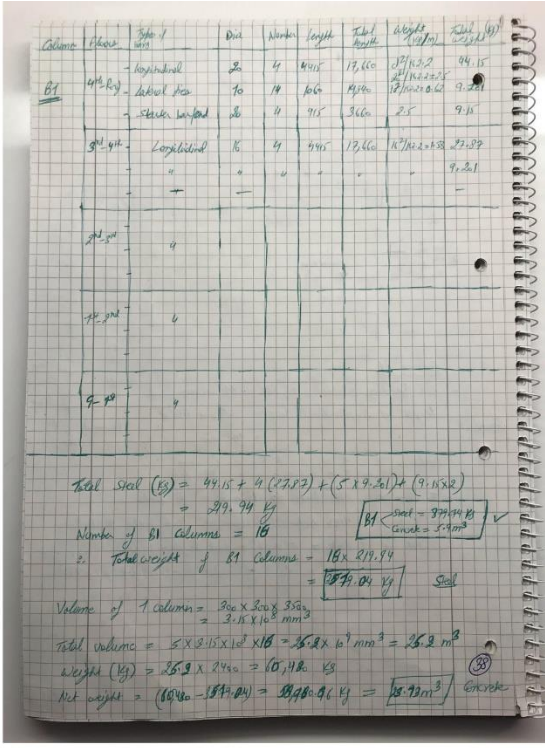
Length of 1 link = 2(a+b) + hook length − bends (6d+135° bend)

= 2(300−(35x2) + 300−(35x2)) + (2x10x10) − (3x2x10)

= 1060 mm.

Total length of links = 1060 x 14 = 14840 mm

37



Column	Floors	Type of bars	Dia	Number	Length	Total length	Weight (kg/m)	Total weight (kg)
B1	4th-Roof	longitudinal	20	4	4415	17,660	$d^2/162.2$ $20^2/162.2=2.5$	44.15
		lateral ties	10	14	1060	14,840	$10^2/162.2=0.62$	9.201
		Starter bar/end	20	4	915	3660	2.5	9.15
	3rd-4th	Longitudinal	16	4	4415	17,660	$16^2/162.2=1.58$	27.87
		"	"	"	"	"	"	9.201
		−	−					−
	2nd-3rd	"						
	1st-2nd	"						
	G-1st	"						

Total Steel (Kg) = 44.15 + 4 (27.87) + (5 x 9.201) + (9.15x2)
= 219.94 Kg

B1 Steel = 3579.04 Kg, Concrete = 5.9 m³

Number of B1 Columns = 16

∴ Total weight of B1 Columns = 16 x 219.94
= 3519.04 Kg Steel

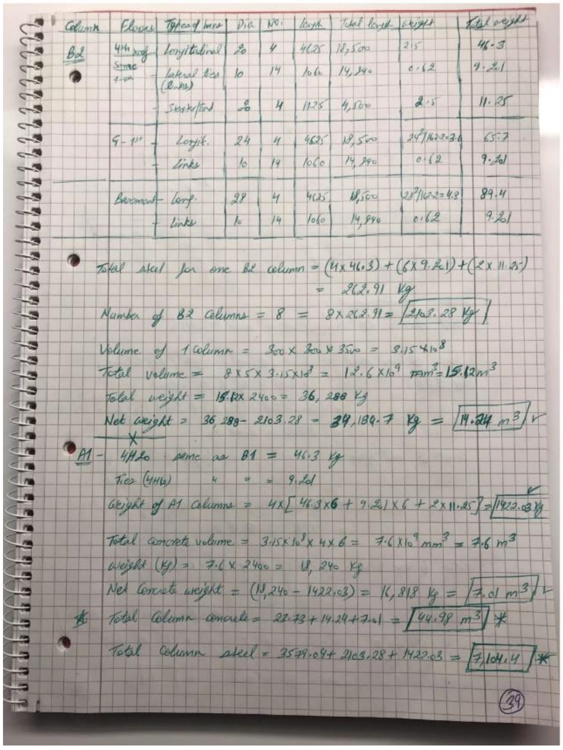
Volume of 1 column = 300 x 300 x 3500
= 3.15 x 10⁸ mm³

Total volume = 5 x 3.15 x 10⁸ x 16 = 25.2 x 10⁹ mm³ = 25.2 m³

Weight (Kg) = 25.2 x 2400 = 60,480 Kg

Net weight = (60,480 − 3519.04) = 39,180.96 Kg = 22.73 m³ Concrete

38



Column	Floors	Type of bars	Dia	No.	Length	Total length	Weight	Total weight
B2	4th-roof Same 1-4th	longitudinal	20	4	4625	18,500	2.5	46.3
		lateral ties (links)	10	14	1060	14,840	0.62	9.21
		Starter/End	20	4	1125	4,500	2.5	11.25
	G-1st	Longit.	24	4	4625	18,500	$24^2/162.2=3.6$	65.7
		links	10	14	1060	14,840	0.62	9.201
	Basement	Long.	28	4	4625	18,500	$28^2/162.2=4.8$	89.4
		links	10	14	1060	14,840	0.62	9.201

Total steel for one B2 column = (4 x 46.3) + (6 x 9.201) + (2 x 11.25)
= 262.91 Kg

Number of B2 Columns = 8 = 8 x 262.91 = 2103.28 Kg

Volume of 1 Column = 300 x 300 x 3500 = 3.15 x 10⁸

Total volume = 8 x 5 x 3.15 x 10⁸ = 12.6 x 10⁹ mm³ = 15.12 m³

Total weight = 15.12 x 2400 = 36, 288 Kg

Net weight = 36,288 − 2103.28 = 34,184.7 Kg = 14.24 m³

A1 - 4H20 same as B1 = 46.3 Kg
Ties (4H10) " " = 9.201

Weight of A1 Columns = 4 x [46.3 x 6 + 9.21 x 6 + 2 x 11.25] = 1422.03 Kg

Total concrete volume = 3.15 x 10⁸ x 4 x 6 = 7.6 x 10⁹ mm³ = 7.6 m³

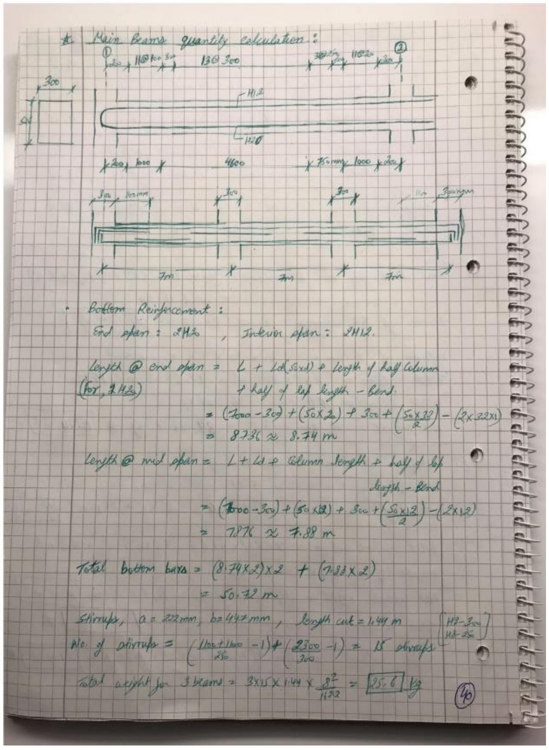
Weight (Kg) = 7.6 x 2400 = 18, 240 Kg

Net Concrete weight = (18,240 − 1422.03) = 16,818 Kg = 7.01 m³

* Total Column concrete = 22.73 + 14.24 + 7.01 = 44.98 m³ *

Total Column steel = 3579.04 + 2103.28 + 1422.03 = 7,104.4 *

39



* Main Beams quantity calculation :

- Bottom Reinforcement :
End span : 2H20 , Interior span : 2H12.

Length @ end span = L + Ld (50xd) + length of half column + half of lap length − Bend.
(For, 2H20)
= (7000 − 300) + (50 x 20) + 300 + (50 x 32 / 2) − (2 x 22 x 1)
= 8736 ≈ 8.74 m

Length @ mid span = L + Ld + Column length + half of lap length − Bend
= (7000 − 300) + (50 x 12) + 300 + (50 x 12 / 2) − (2 x 12)
= 7876 ≈ 7.88 m

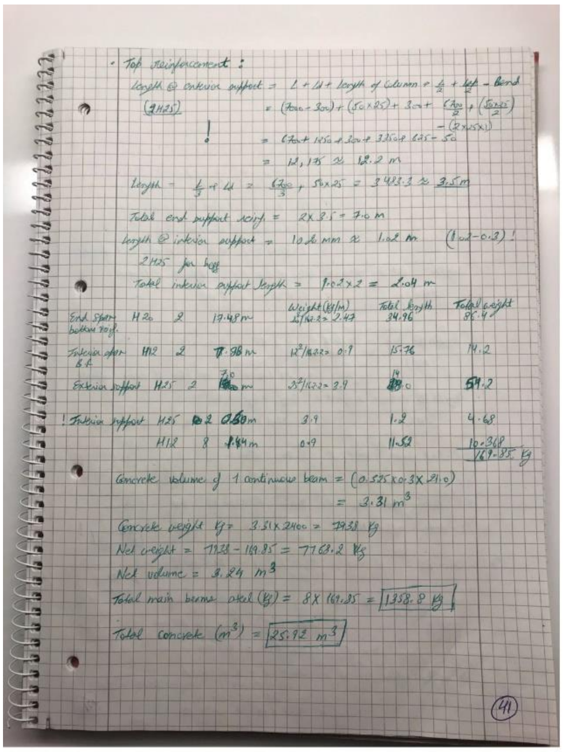
Total bottom bars = (8.74 x 2) x 2 + (7.88 x 2)
= 50.72 m

Stirrups, a = 222 mm, b = 447 mm , length cut = 1.44 m [H8−300, H8−250]

No. of stirrups = ((1100 + 1100)/250 − 1) + (2300/300 − 1) = 15 stirrups

Total weight for 3 beams = 3 x 15 x 1.44 x 8²/162.2 = 25.6 Kg

40



• Top reinforcement :

Length @ exterior support = L + Ld + length of column + $\frac{L}{2}$ + $\frac{Ld}{2}$ − Bend

(2H25) = (7000 − 300) + (50×25) + 300 + $\frac{6700}{2}$ + $\left(\frac{50\times25}{2}\right)$ − (2×25×1)

= 6700 + 1250 + 300 + 3350 + 625 − 50

= 12,175 ≈ 12.2 m

Length = $\frac{L}{3}$ + Ld = $\frac{6700}{3}$ + 50×25 = 3483.3 ≈ 3.5 m

Total end support reinf. = 2×3.5 = 7.0 m

Length @ interior support = 1020 mm ≈ 1.02 m (1.02 − 0.3)!

2H25 for legs

Total interior support length = 1.02×2 = 2.04 m

				Weight (kg/m)	Total length	Total weight
End span bottom reinf.	H20	2	17.48 m	20^2/162.2 = 2.47	34.96	86.4
Interior span B.R	H12	2	7.88 m	12^2/162.2 = 0.9	15.76	14.2
Exterior support	H25	2	7.0 m	25^2/162.2 = 3.9	14.0	54.2
! Interior support	H25	2	0.60 m	3.9	1.2	4.68
	H12	8	1.44 m	0.9	11.52	10.368
						169.85 kg

Concrete volume of 1 continuous beam = (0.525 × 0.3 × 21.0) = 3.31 m^3

Concrete weight kg = 3.31 × 2400 = 7938 kg

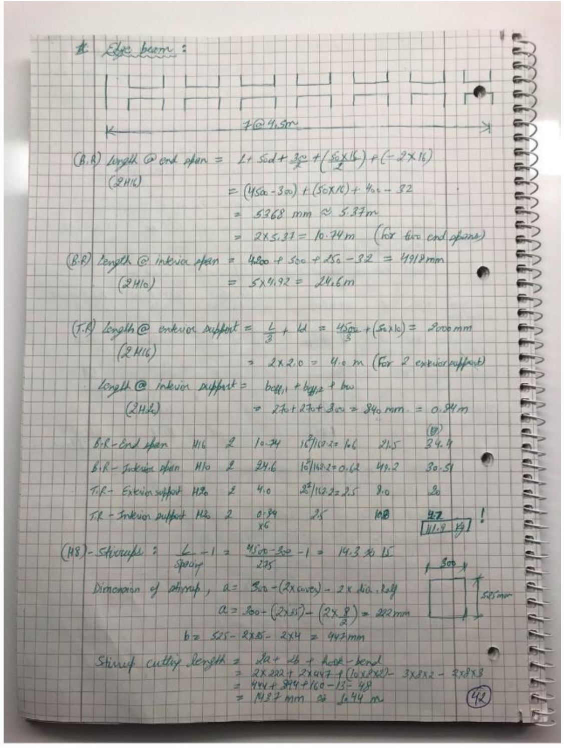
Net weight = 7938 − 169.85 = 7768.2 kg

Net volume = 3.24 m^3

Total main beams steel (kg) = 8 × 169.85 = $\boxed{1358.8 \text{ kg}}$

Total concrete (m^3) = $\boxed{25.92\ m^3}$

(41)



Edge beam :

7@4.5m

(B.R) Length @ end span = L + 50d + $\frac{300}{2}$ + $\left(\frac{50\times16}{2}\right)$ + (−2×16)

(2H16) = (4500 − 300) + (50×16) + 400 − 32

= 5368 mm ≈ 5.37 m

= 2×5.37 = 10.74 m (For two end spans)

(B.R) Length @ interior span = 4200 + 500 + 250 − 32 = 4918 mm

(2H10) = 5×4.92 = 24.6 m

(T.R) Length @ exterior support = $\frac{L}{3}$ + ld = $\frac{4500}{3}$ + (50×10) = 2000 mm

(2H16) = 2×2.0 = 4.0 m (For 2 exterior support)

Length @ interior support = $b_{eff,1}$ + $b_{eff,2}$ + bw

(2H20) = 270 + 270 + 300 = 840 mm = 0.84 m

						(kg)
B.R − End span	H16	2	10.74	16^2/162.2 = 1.6	21.5	34.4
B.R − Interior span	H10	2	24.6	10^2/162.2 = 0.62	49.2	30.51
T.R − Exterior support	H20	2	4.0	20^2/162.2 = 2.5	8.0	20
T.R − Interior support	H20	2	0.84 ×6	2.5	10.8	27
						$\boxed{111.9 \text{ kg}}$!

(H8) − Stirrups : $\frac{L}{\text{Spacing}}$ − 1 = $\frac{4500-300}{275}$ − 1 = 14.3 ≈ 15

Dimension of stirrup, a = 300 − (2×cover) − 2×dia./2

a = 300 − (2×35) − (2×$\frac{8}{2}$) = 222 mm

b = 525 − 2×35 − 2×4 = 447 mm

300, 525 mm

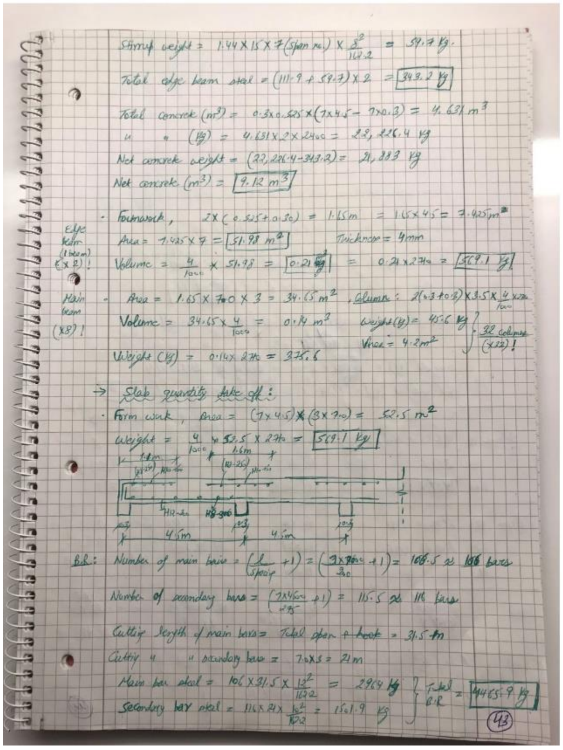
Stirrup cutting length = 2a + 2b + hook − bend

= 2×222 + 2×447 + (10×8×2) − 3×8×2 − 2×8×3

= 444 + 894 + 160 − 13 − 48

= 1437 mm ≈ 1.44 m

(42)



Stirrup weight = 1.44 × 15 × 7 (span no.) × $\frac{8^2}{162.2}$ = 59.7 kg.

Total edge beam steel = (111.9 + 59.7) × 2 = $\boxed{343.2 \text{ kg}}$

Total concrete (m^3) = 0.3 × 0.525 × (7×4.5 − 7×0.3) = 4.631 m^3

" " (kg) = 4.631 × 2 × 2400 = 22,226.4 kg

Net concrete weight = (22,226.4 − 343.2) = 21,883 kg

Net concrete (m^3) = $\boxed{9.12\ m^3}$

Edge beam (1 beam) (×2)! • Formwork, 2×(0.525 + 0.30) = 1.65 m = 1.65 × 4.5 = 7.425 m^2

Area = 7.425 × 7 = $\boxed{51.98\ m^2}$ Thickness = 4 mm

Volume = $\frac{4}{1000}$ × 51.98 = $\boxed{0.21}$ = 0.21 × 2710 = $\boxed{569.1 \text{ kg}}$

Main beam (×8)! • Area = 1.65 × 7.0 × 3 = 34.65 m^2, Column : 2(0.3 + 0.3) × 3.5 × $\frac{4}{1000}$ × 2710

Volume = 34.65 × $\frac{4}{1000}$ = 0.14 m^3 Weight (kg) = 45.6 kg

Weight (kg) = 0.14 × 2710 = 375.6 Area = 4.2 m^2 32 columns (×32)!

→ Slab quantity take off :

• Form work, Area = (7×4.5) × (3×7.0) = 52.5 m^2

Weight = $\frac{4}{1000}$ × 52.5 × 2710 = $\boxed{569.1 \text{ kg}}$

1.6m, 1.6m, H12-200, H8-300, 4.5m, 4.5m

B.R : Number of main bars = $\left(\frac{L}{\text{Spacing}} + 1\right)$ = $\left(\frac{3\times7000}{200} + 1\right)$ = 106.5 ≈ 106 bars

Number of secondary bars = $\left(\frac{7\times4500}{275} + 1\right)$ = 115.5 ≈ 116 bars

Cutting length of main bars = Total span + hook = 31.5 m

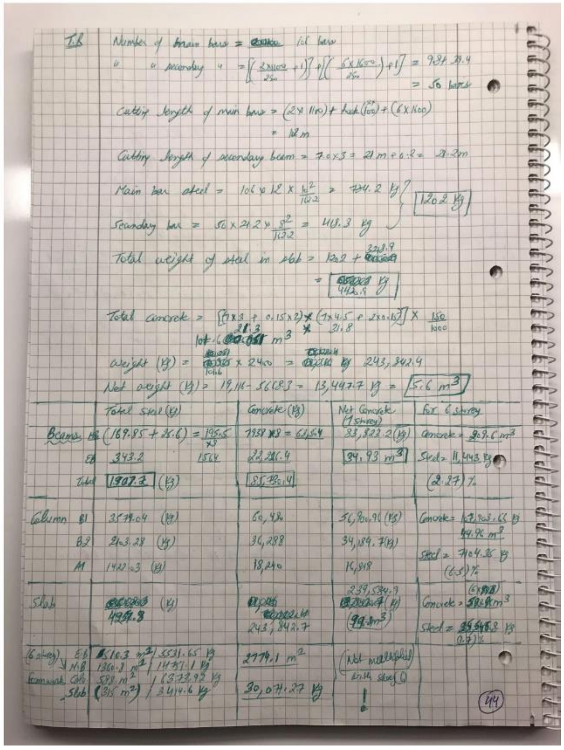
Cutting " " secondary bars = 7.0×3 = 21 m

Main bar steel = 106 × 31.5 × $\frac{12^2}{162.2}$ = 2964 kg

Secondary bar steel = 116 × 21 × $\frac{10^2}{162.2}$ = 1501.9 kg

Total B.R = $\boxed{4465.9 \text{ kg}}$

(43)



T.R Number of main bars = 106 bars

" " secondary " = $\left[\left(\frac{2\times1100}{250} + 1\right)\right] + \left[\left(\frac{6\times1600}{250}\right) + 1\right]$ = 9.8 + 39.4 = 50 bars

Cutting length of main bars = (2×1100) + hook (100) + (6×1600) = 12 m

Cutting length of secondary beam = 7.0×3 = 21 m + 0.2 = 21.2 m

Main bar steel = 106 × 12 × $\frac{12^2}{162.2}$ = 784.2 kg

Secondary bar = 50 × 21.2 × $\frac{8^2}{162.2}$ = 418.3 kg

$\boxed{1202 \text{ kg}}$

Total weight of steel in slab = 1202 + 3228.9

= $\boxed{4426.9 \text{ kg}}$

Total concrete = [(7×3 + 0.15×2) × (7×4.5 + 2×0.15)] × $\frac{150}{1000}$

21.3 × 31.8

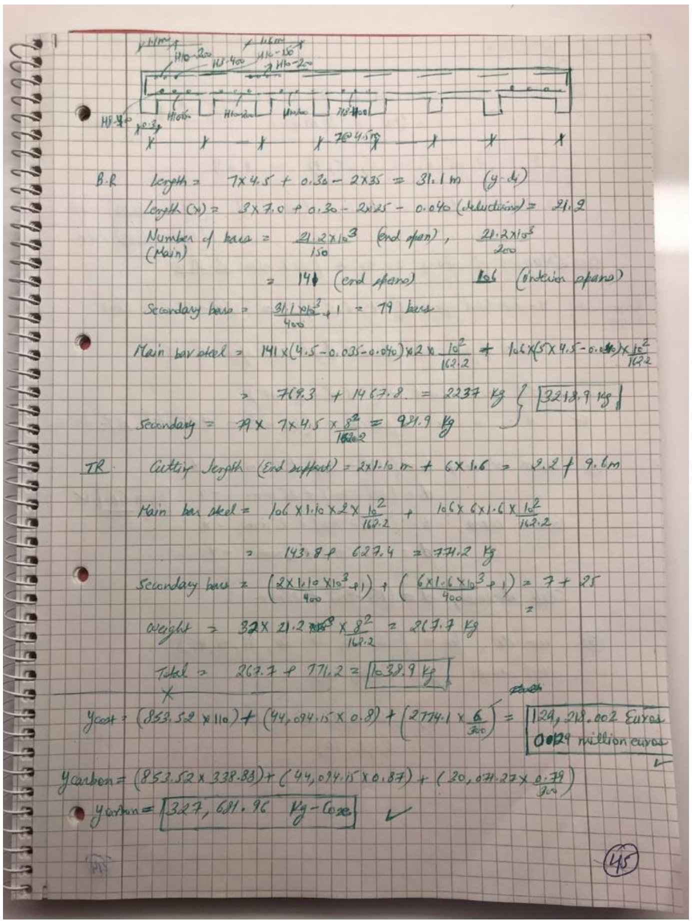
101.6 m^3

Weight (kg) = 101.6 × 2400 = 243,842.4

Net weight (kg) = 19,116 − 5668.3 = 13,447.7 kg = $\boxed{5.6\ m^3}$

		Total Steel (kg)	Concrete (kg)	Net Concrete (1 storey)	For 6 storey
Beams	MB	(169.85 + 25.6) = 195.5 ×8	7938 ×8 = 63,504	33,822.2 (kg)	Concrete = 209.6 m^3
	EB	343.2	1564	22,226.4	$\boxed{34.93\ m^3}$	Steel = 11,443 kg
	Total	$\boxed{1907.2}$ (kg)	$\boxed{85,730.4}$		(2.27)%
Column	B1	3579.04 (kg)	60,480	56,900.96 (kg)	Concrete = 107,903.66 kg 44.96 m^3
	B2	2103.28 (kg)	36,288	34,184.7 (kg)	Steel = 7104.35 kg
	M	1422.3 (kg)	18,240	16,818	(6.5)%
Slab		4426.8 (kg)	243,842.7	239,584.9 (kg) (99.8 m^3)	Concrete = (6×99.8) 598.8 m^3 Steel = 26,548.8 kg (0.7)%
(6 storey) Formwork	E.B / M.B / Cols / Slab	2510.3 m^2 / 5531.65 kg; 1360.8 m^2 / 14,371.1 kg; 588 m^2 / 63,733.92 kg; (315 m^2) / 2414.6 kg	2774.1 m^2 30,071.27 kg	(Not multiplied with storey 6) !	

(44)



B.R

Length = $7 \times 4.5 + 0.30 - 2 \times 35 = 31.1$ m (y-dir)

Length (x) = $3 \times 7.0 + 0.30 - 2 \times 25 - 0.040$ (deductions) = 21.2

Number of bars (Main) = $\frac{21.2 \times 10^3}{150}$ (end span), $\frac{21.2 \times 10^3}{200}$

= 141 (end spans) 106 (interior spans)

Secondary bars = $\frac{31.1 \times 10^3}{400} + 1 = 79$ bars

Main bar steel = $141 \times (4.5 - 0.035 - 0.040) \times 2 \times \frac{10^2}{162.2} + 106 \times (5 \times 4.5 - 0.040) \times \frac{10^2}{162.2}$

= $769.3 + 1467.8 = 2237$ Kg

Secondary = $79 \times 7 \times 4.5 \times \frac{8^2}{162.2} = 981.9$ Kg

$\boxed{3218.9 \text{ Kg}}$

TR

Cutting length (End support) = 2×1.10 m + 6×1.6 = 2.2 + 9.6 m

Main bar steel = $106 \times 1.10 \times 2 \times \frac{10^2}{162.2} + 106 \times 6 \times 1.6 \times \frac{10^2}{162.2}$

= $143.8 + 627.4 = 771.2$ Kg

Secondary bars = $\left(\frac{2 \times 1.10 \times 10^3}{400} + 1\right) + \left(\frac{6 \times 1.6 \times 10^3}{400} + 1\right) = 7 + 25$

Weight = $32 \times 21.2 \times \frac{8^2}{162.2} = 267.7$ Kg

Total = $267.7 + 771.2 = \boxed{1038.9 \text{ Kg}}$

$y_{cost} = (853.52 \times 110) + (44{,}094.15 \times 0.8) + \left(2774.1 \times \frac{6}{300}\right) = \boxed{129{,}218.002 \text{ Euros}}$ 0.129 million euros

$y_{carbon} = (853.52 \times 338.88) + (44{,}094.15 \times 0.87) + \left(30{,}071.27 \times \frac{0.79}{300}\right)$

$y_{carbon} = \boxed{327{,}681.96 \text{ Kg-Co}_2\text{e}}$

45

```
function y = frame(x)

x = framemapvariables(x);

% 1. SLAB DESIGN
% 1.1 LOADING CALCULATIONS
bxdirection = x(2)*(17);
bydirection = x(19)*(20);
bbayarea = bxdirection*bydirection; % in m2
snominalcover = 25; % in mm
sa = snominalcover+(12/2); % in mm
simposedload = 2.5; % in KN/m2 for office building
spartitionwall = 1.5; % in KN/m2 for office building
sselfweight = x(1)*25/1000; % in KN/m2
sfinisheweight = 1.25; % in KN/m2
stotalpermanentload = sselfweight+sfinisheweight; % in KN/m2
stotalvariableload = simposedload+spartitionwall; % in KN/m2
sdesignload = (1.35*stotalpermanentload)+(1.65*stotalvariableload);% in KN/m2
sf = sdesignload*x(2); % in KN/m

% 1.2 BENDING MOMENT AND SHEAR FORCE CALCULATIONS

smpespan = 0.086*sf*x(2); % slab moment pinned end span
smfesup = -0.063*sf*x(2); % slab moment fixed end support
smfespan = 0.063*sf*x(2); % slab moment fixed end span
smfisupport = -0.086*sf*x(2); % slab moment first interior support
smaispan = 0.063*sf*x(2); % slab moment all interior span
smoisupport = -0.063*sf*x(2); % slab moment other interior support

sspesupport = 0.4*sf; % slab shear pinned end support
ssfesupport = 0.48*sf; % slab shear fixed end support
ssfisupport = 0.6*sf; % slab shear first interior support
ssoisupport = 0.5*sf; % slab shear other interior support

% 1.3 FLEXURE DESIGN/CHECK

seffectivedepth = x(1)-snominalcover-(12/2); % in mm
sbreadth = 1000; % in mm

smpespank = (smpespan*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x(3));
smpespanleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smpespank))^(1/2)); % in mm
smpespanmaxleverarm = 0.95*seffectivedepth;
 if smpespanleverarm <= smpespanmaxleverarm
    smpespanz = smpespanleverarm;
 else
    smpespanz = smpespanmaxleverarm;
 end
smpespanrequiredsteel = (smpespan*1000000)/(0.87*x(4)*smpespanz);
smpespanminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smpespanprovidedsteel = x(6)*3.14*x(5)*x(5)/4;
smpespanmaximumsteel = 0.04*(sbreadth*x(1)-smpespanprovidedsteel);

smfesupk = (abs(smfesup)*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x(3));
```

```
smfesupleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smfesupk))^(1/2)); % in mm
smfesupmaxleverarm = 0.95*seffectivedepth;
 if smfesupleverarm <= smfesupmaxleverarm
    smfesupz = smfesupleverarm;
 else
    smfesupz = smfesupmaxleverarm;
 end
smfesuprequiredsteel = (smfesup*1000000)/(0.87*x(4)*smfesupz);
smfesupminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smfesupprovidedsteel = x(8)*3.14*x(7)*x(7)/4;
smfesupmaximumsteel = 0.04*(sbreadth*x(1)-smfesupprovidedsteel);

smfespank = (smfespan*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x(3));
smfespanleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smfespank))^(1/2)); % in mm
smfespanmaxleverarm = 0.95*seffectivedepth;
 if smfespanleverarm <= smfespanmaxleverarm
    smfespanz = smfespanleverarm;
 else
    smfespanz = smfespanmaxleverarm;
 end
smfespanrequiredsteel = (smfespan*1000000)/(0.87*x(4)*smfespanz);
smfespanminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smfespanprovidedsteel = x(10)*3.14*x(9)*x(9)/4;
smfespanmaximumsteel = 0.04*(sbreadth*x(1)-smfespanprovidedsteel);

smfisupportk = (abs(smfisupport)*1000000)/↙
(sbreadth*seffectivedepth*seffectivedepth*x(3));
smfisupportleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smfisupportk))^(1/2)); % in↙
mm
smfisupportmaxleverarm = 0.95*seffectivedepth;
 if smfisupportleverarm <= smfisupportmaxleverarm
    smfisupportz = smfisupportleverarm;
 else
    smfisupportz = smfisupportmaxleverarm;
 end
smfisupportrequiredsteel = (abs(smfisupport)*1000000)/(0.87*x(4)*smfisupportz);
smfisupportminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smfisupportprovidedsteel = x(12)*3.14*x(11)*x(11)/4;
smfisupportmaximumsteel = 0.04*(sbreadth*x(1)-smfisupportprovidedsteel);

smaispank = (smaispan*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x(3));
smaispanleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smaispank))^(1/2)); % in mm
smaispanmaxleverarm = 0.95*seffectivedepth;
 if smaispanleverarm <= smaispanmaxleverarm
    smaispanz = smaispanleverarm;
 else
    smaispanz = smaispanmaxleverarm;
 end
smaispanrequiredsteel = (smaispan*1000000)/(0.87*x(4)*smaispanz);
smaispanminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smaispanprovidedsteel = x(14)*3.14*x(13)*x(13)/4;
smaispanmaximumsteel = 0.04*(sbreadth*x(1)-smaispanprovidedsteel);
```

```
smoisupportk = (abs(smoisupport)*1000000)/
(sbreadth*seffectivedepth*seffectivedepth*x(3));
smoisupportleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smoisupportk))^(1/2)); % in
mm
smoisupportmaxleverarm = 0.95*seffectivedepth;
 if smoisupportleverarm <= smoisupportmaxleverarm
    smoisupportz = smoisupportleverarm;
 else
    smoisupportz = smoisupportmaxleverarm;
 end
smoisupportrequiredsteel = (abs(smoisupport)*1000000)/(0.87*x(4)*smoisupportz);
smoisupportminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smoisupportprovidedsteel = x(16)*3.14*x(15)*x(15)/4;
smoisupportmaximumsteel = 0.04*(sbreadth*x(1)-smoisupportprovidedsteel);

sbrsrequiredarea = 0.2*smpespanprovidedsteel;
sbrsprovidedarea = x(22)*3.14*x(21)*x(21)/4;

% 1.4 SHEAR DESIGN/CHECK

srow = smpespanprovidedsteel/(sbreadth*seffectivedepth);
sk1 = 1+((200/seffectivedepth)^(1/2));
 if sk1>2
    sk1=2;
 else
    sk1= 1+ ((200/seffectivedepth)^(1/2));
 end
sresistanceshear = (0.12*sk1*((100*srow*x(3))^(1/3))/1000)
*sbreadth*seffectivedepth;
sminshear = 0.035*((sk1)^(3/2))*(x(3)^(1/2));
sminresistanceshear = (sminshear+(sk1*(ssfisupport*1000)/sbreadth*seffectivedepth))
*sbreadth*seffectivedepth;


% 1.5 DEFLECTION DESIGN/CHECK

sk2 = 1.3 ; % for one way solid slab
srowzero = (x(3)^(1/2))/1000;
srowone = smpespanprovidedsteel/(sbreadth*seffectivedepth);
srowtwo = 0;
 if srowone<=srowzero
    sspantodepthratio = sk2*(11+(1.5*((x(3)^(1/2))*(srowzero/srowone)))+(3.2*(x(3)^
(1/2))*(((srowzero/srowone)-1)^(3/2))));
 else
    sspantodepthratio = sk2*(11+(1.5*((x(3)^(1/2))*(srowzero/(srowone-srowtwo))))+
((1/12)*(x(3)^(1/2))*((srowtwo/srowzero)^(1/2))));
 end

sf1 = (500*smpespanprovidedsteel)/(x(4)*smpespanrequiredsteel);
sbasicspantodepthratio = sspantodepthratio*sf1; % in mm
sactualspantodepthratio = x(2)*1000/seffectivedepth; % in mm
```

```
% 1.6 CRACKING DESIGN/CHECK

skc = 0.4;
smincrackingarea = (skc*0.3*(x(3)^(3/2))*sbreadth*x(1))/x(4);

% 1.7 QUANTITY CALCULATION
% 1.7.1 STEEL CALCULATION

sbrlx = (x(2)*x(17))+(x(18)/1000)-(2*snominalcover/1000); % SLAB BOTTOM
REINFORCEMENT IN X DIRECTION IN M
sbrly = (x(19)*x(20))+(x(18)/1000)-(2*snominalcover/1000)-0.040; % SLAB BOTTOM
REINFORCEMENT IN Y DIRECTION IN M
sbrnmespan = sbrly*1000/(1000/x(6)); % SLAB BOTTOM REINFORCEMENT NUMBER OF main
BARS IN END SPAN
sbrnmispan = sbrly*1000/(1000/x(14)); % SLAB BOTTOM REINFORCEMENT NUMBER OF main
BARS IN INTERIOR SPAN
sbrnsispan = sbrlx*1000/(1000/x(14)); % SLAB BOTTOM REINFORCEMENT NUMBER OF
secondary BARS
sbrmweight = ((sbrnmespan*(x(2)-snominalcover/1000-0.040)*2*x(5)*x(5))/162.2)+
(sbrnmispan*(x(2)-snominalcover/1000-0.040)*(x(17)-2)*x(14)*x(14)/162.2);% SLAB
BOTTOM REINFORCEMENT MAIN BAR WEIGHT
sbrsweight = sbrnsispan*x(14)*x(2)*x(21)*x(21)/162.2; % WEIGHT IN KG
sbrtweight = sbrmweight+sbrsweight; % SLAB BOTTOM REINFORCEMENT TOTAL STEEL WEIGHT
IN KG

strelx = 0.2*x(2)*2; % SLAB TOP REINFORCEMENT AT END SUPPORT IN X DIRECTION IN M
strilx = 0.3*x(2)*(x(17)-1); % SLAB TOP REINFORCEMENT AT INTERIOR SUPPORT IN X
DIRECTION IN M
strly = (x(19)*x(20))+(x(18)/1000)-(2*snominalcover/1000)-0.040; % SLAB TOP
REINFORCEMENT IN Y DIRECTION IN M
strmweight = ((strly*strelx*x(7)*x(7))/162.2)+(strly*strilx*x(15)*x(15)/162.2); %
SLAB TOP REINFORCEMENT MAIN BAR WEIGHT IN KG
strsweight = (strelx*1000/(1000/x(14))*x(21)*x(21)/162.2)+(strilx*1000/(1000/x(14))
*x(21)*x(21)/162.2); % WEIGHT IN KG
strtweight = strmweight+strsweight; % TOTAL TOP REINFORCEMENT WEIGHT

slabtotalsteel = sbrtweight+strtweight; % TOTAL SLAB STEEL WEIGHT IN KG
slabtotalbuildingsteel = slabtotalsteel*6; % TOTAL BUILDING SLAB STEEL WEIGHT IN KG


% 1.7.2 CONCRETE CALCULATION

stc = ((x(2)*x(17))+(x(19)*x(20)))*2*x(18)*x(1)/1000; % SLAB TOTAL CONCRETE in M3
stcweight = stc*2400; % SLAB TOTAL CONCRETE in KG
sncweight = stcweight-slabtotalsteel; % NET AMOUNT OF CONCRETE IN KG
snc = sncweight/2400; % NET AMOUNT OF CONCRETE IN M3
sncbuilding = snc*6; % NET AMOUNT OF BUILDING SLAB CONCRETE IN M3

% 1.7.3 FORMWORK CALCULATION

sfa = (((x(2)-(x(18)/1000))*x(17))*((x(19)-(x(18)/1000)*x(20))))*6;  % AREA OF
FORMWORK FOR BUILDING
sfweight = sfa*4/1000*2710; %WEIGHT OF FORMWORK FOR BUILDING
```

```
% 2. BEAM DESIGN
% 2.1 MAIN BEAM DESIGN
% 2.1.1 FIRE RESISTANCE/COVER DETERMINATION
 bnominalcover = 25; % IN MM
 baxisdistance = bnominalcover+8+(32/2); % AXIS DISTANCE FOR 1.5 HR OF FIRE

% 2.1.2 LOADING CALULATIONS
mbmaxdesignload = sdesignload; %MAIN BEAM MAXIMUM DESIGN LOAD
mbmindesignload = 1.25*stotalpermanentload; %MAIN BEAM MINIMUM DESIGN LOAD
mbmaxslab = 1.1*mbmaxdesignload*x(2); %MAIN BEAM MAXIMUM DESIGN LOAD DUE TO SLAB
mbminslab = 1.1*x(2)*mbmindesignload; %MAIN BEAM MINIMUM DESIGN LOAD DUE TO SLAB
mbmax = 1.25*x(23)*x(18)*25/1000000; %MAIN BEAM MAXIMUM DESIGN LOAD DUE TO MAIN↙
BEAMS
mbmin = mbmax;
mbtoalmaxload = mbmaxslab+mbmax; %IN KN/M
mbtotalminload = mbminslab+mbmin; % IN KN/M

% 2.1.3 BENDING MOMENT AND SHEAR FORCE ANALYSIS/ SUB FRAME ANALYSIS

mbibeam = x(18)*(x(23)^3)/12; %MOMENT OF INERTIA OF BEAM
mbicolumn = x(24)*(x(25)^3)/12; %MOMENT OF INERTIA OF COLUMN
mbendk = mbibeam/x(19)*1000; %STIFFNESS OF END BEAM
mbintk = mbendk; %STIFFNESS OF INTERIOR BEAM
storeyheight = 3.5;
mbuppercolumnk = mbicolumn/storeyheight*1000; %STIFFNESS OF UPPER COLUMN
mblowercolumnk = mbicolumn/storeyheight*1000; %STIFFNESS OF LOWER COLUMN
mbdfendjointb = mbendk/(mbendk+(2*mbuppercolumnk)); % MAIN BEAM DISTRIBUTION FACTOR↙
AT END JOINT FOR BEAM
mbdfendjointc = (1-mbdfendjointb)/2; % COLUMN DISTRIBUTION FACTOR AT END JOINT FOR↙
COLUMN
mbdfinteriorjointendb = mbendk/(mbendk+(0.5*mbintk)+(2*mblowercolumnk)); % MAIN↙
BEAM DISTRIBUTION FACTOR AT INTERIOR JOINT FOR END BEAM
mbdfinteriorjointintb = (0.5*mbendk)/(mbendk+(0.5*mbintk)+(2*mblowercolumnk)); %↙
MAIN BEAM DISTRIBUTION FACTOR AT INTERIOR JOINT FOR INTERIOR BEAM
mbdfinteriorjointc = (mbuppercolumnk)/(mbendk+(0.5*mbintk)+(2*mblowercolumnk)); %↙
column DISTRIBUTION FACTOR AT END JOINT FOR COLUMN
mbendmomentmax = (mbtoalmaxload*x(19)*x(19))/12;
mbintmomentmax = (mbtoalmaxload*x(19)*x(19))/12;
mbendmomentmin = (mbtotalminload*x(19)*x(19))/12;
mbintmomentmin = (mbtotalminload*x(19)*x(19))/12;
r1ejuc = mbdfendjointc; % Unit moment applied at end joint - upper column of end↙
joint (row1,column1)
r1ejb = mbdfendjointb; % Unit moment applied at end joint - beam of end joint↙
(row1,column2)
r1ejlc = r1ejuc; % Unit moment applied at end joint - lower column of end joint↙
(row1,column3)
r1ijuc = 0; % Unit moment applied at interior joint - upper column of interior↙
joint (row1,column4)
r1ijeb = r1ejb/2; % Unit moment applied at interior joint - end beam of interior↙
joint (row1,column5)
r1ijib = 0; % Unit moment applied at interior joint - interior beam of interior↙
joint (row1,column6)
```

```
r1ijlc = 0; % Unit moment applied at interior joint - lower column of interior ↙
joint (row1,column7)
r2ejuc = 0; % Unit moment applied at end joint - upper column of end joint (row2, ↙
column1)
r2ejb = mbdfinteriorjointendb/2; % Unit moment applied at end joint - beam of end ↙
joint (row2,column2)
r2ejlc = 0; % Unit moment applied at end joint - lower column of end joint (row2, ↙
column3)
r2ijuc = mbdfinteriorjointc; % Unit moment applied at interior joint - upper column ↙
of interior joint (row2,column4)
r2ijeb = mbdfinteriorjointendb; % Unit moment applied at interior joint - end beam ↙
of interior joint (row2,column5)
r2ijib = mbdfinteriorjointintb; % Unit moment applied at interior joint - interior ↙
beam of interior joint (row2,column6)
r2ijlc = r2ijuc; % Unit moment applied at interior joint - lower column of interior ↙
joint (row2,column7)
r3ejuc = (r1ejuc/r1ijeb)-r2ejuc; % Unit moment applied at end joint - upper column ↙
of end joint (row3,column1)
r3ejb = (r1ejb/r1ijeb)-r2ejb; % Unit moment applied at end joint - beam of end ↙
joint (row3,column2)
r3ejlc = (r1ejlc/r1ijeb)-r2ejlc; % Unit moment applied at end joint - lower column ↙
of end joint (row3,column3)
r3ijuc = (r1ijuc/r1ijeb)-r2ijuc; % Unit moment applied at interior joint - upper ↙
column of interior joint (row3,column4)
r3ijeb = (r1ijeb/r1ijeb)-r2ijeb; % Unit moment applied at interior joint - end beam ↙
of interior joint (row3,column5)
r3ijib = (r1ijib/r1ijeb)-r2ijib; % Unit moment applied at interior joint - interior ↙
beam of interior joint (row3,column6)
r3ijlc = (r1ijlc/r1ijeb)-r2ijlc; % Unit moment applied at interior joint - lower ↙
column of interior joint (row3,column7)
r4ejuc = (r2ejuc/r2ejb)-r1ejuc; % Unit moment applied at end joint - upper column ↙
of end joint (row4,column1)
r4ejb = (r2ejb/r2ejb)-r1ejb; % Unit moment applied at end joint - beam of end joint ↙
(row4,column2)
r4ejlc = (r2ejlc/r2ejb)-r1ejlc; % Unit moment applied at end joint - lower column ↙
of end joint (row4,column3)
r4ijuc = (r2ijuc/r2ejb)-r1ijuc; % Unit moment applied at interior joint - upper ↙
column of interior joint (row4,column4)
r4ijeb = (r2ijeb/r2ejb)-r1ijeb; % Unit moment applied at interior joint - end beam ↙
of interior joint (row4,column5)
r4ijib = (r2ijib/r2ejb)-r1ijib; % Unit moment applied at interior joint - interior ↙
beam of interior joint (row4,column6)
r4ijlc = (r2ijlc/r2ejb)-r1ijlc; % Unit moment applied at interior joint - lower ↙
column of interior joint (row4,column7)
r5ejuc = (r3ejuc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at end joint - upper ↙
column of end joint (row5,column1)
r5ejb = (r3ejb/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at end joint - beam of ↙
end joint (row5,column2)
r5ejlc = (r3ejlc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at end joint - lower ↙
column of end joint (row5,column3)
r5ijuc = (r3ijuc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint - ↙
upper column of interior joint (row5,column4)
r5ijeb = (r3ijeb/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint - ↙
```

```
end beam of interior joint (row5,column5)
r5ijib = (r3ijib/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint -↙
interior beam of interior joint (row5,column6)
r5ijlc = (r3ijlc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint -↙
lower column of interior joint (row5,column7)
r6ejuc = (r4ejuc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at end joint↙
- upper column of end joint (row6,column1)
r6ejb = (r4ejb/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at end joint -↙
beam of end joint (row6,column2)
r6ejlc = (r4ejlc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at end joint↙
- lower column of end joint (row6,column3)
r6ijuc = (r4ijuc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - upper column of interior joint (row6,column4)
r6ijeb = (r4ijeb/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - end beam of interior joint (row6,column5)
r6ijib = (r4ijib/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - interior beam of interior joint (row6,column6)
r6ijlc = (r4ijlc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - lower column of interior joint (row6,column7)
c1r1ejuc = 0; % Moment in members for case 1 - upper column of end joint (case1, ↙
row1,column1)
c1r1ejb = -mbendmomentmax; % Moment in members for case 1 - beam of end joint↙
(case1,row1,column2)
c1r1ejlc = 0; % Moment in members for case 1 - beam of end joint (case1,row1, ↙
column3)
c1r1ijuc = 0; % Moment in members for case 1 - upper column of interior joint↙
(case1,row1,column4)
c1r1ijeb = mbendmomentmax; % Moment in members for case 1 - end beam of interior↙
joint (case1,row1,column5)
c1r1ijib = -mbendmomentmax; % Moment in members for case 1 - interior beam of↙
interior joint (case1,row1,column6)
c1r1ijlc = 0; % Moment in members for case 1 - lower column of interior joint↙
(case1,row1,column7)
c1r2ejuc = r5ejuc*mbendmomentmax; % Moment in members for case 1 - upper column of↙
end joint (case1,row2,column1)
c1r2ejb = r5ejb*mbendmomentmax; % Moment in members for case 1 - beam of end joint↙
(case2,row1,column2)
c1r2ejlc = r5ejlc*mbendmomentmax; % Moment in members for case 1 - beam of end↙
joint (case1,row2,column3)
c1r2ijuc = r5ijuc*mbendmomentmax; % Moment in members for case 1 - upper column of↙
interior joint (case1,row2,column4)
c1r2ijeb = r5ijeb*mbendmomentmax; % Moment in members for case 1 - end beam of↙
interior joint (case1,row2,column5)
c1r2ijib = r5ijib*mbendmomentmax; % Moment in members for case 1 - interior beam of↙
interior joint (case1,row2,column6)
c1r2ijlc = r5ijlc*mbendmomentmax; % Moment in members for case 1 - lower column of↙
interior joint (case1,row2,column7)
c1r3ejuc = r6ejuc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
upper column of end joint (case1,row3,column1)
c1r3ejb = r6ejb*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
beam of end joint (case2,row3,column2)
c1r3ejlc = r6ejlc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
beam of end joint (case1,row3,column3)
```

```
c1r3ijuc = r6ijuc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
upper column of interior joint (case1,row3,column4)
c1r3ijeb = r6ijeb*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
end beam of interior joint (case1,row3,column5)
c1r3ijib = r6ijib*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
interior beam of interior joint (case1,row3,column6)
c1r3ijlc = r6ijlc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
lower column of interior joint (case1,row3,column7)
c1r4ejuc = c1r1ejuc+c1r2ejuc+c1r3ejuc; % Sum of Moments in members for case 1 -↙
upper column of end joint (case1,row4,column1)
c1r4ejb = c1r1ejb+c1r2ejb+c1r3ejb; % Sum of Moments in members for case 1 - beam of↙
end joint (case2,row4,column2)
c1r4ejlc = c1r1ejlc+c1r2ejlc+c1r3ejlc; % Sum of Moments in members for case 1 -↙
beam of end joint (case1,row4,column3)
c1r4ijuc = c1r1ijuc+c1r2ijuc+c1r3ijuc; % Sum of Moments in members for case 1 -↙
upper column of interior joint (case1,row4,column4)
c1r4ijeb = c1r1ijeb+c1r2ijeb+c1r3ijeb; % Sum of Moments in members for case 1 - end↙
beam of interior joint (case1,row4,column5)
c1r4ijib = c1r1ijib+c1r2ijib+c1r3ijib; % Sum of Moments in members for case 1 -↙
interior beam of interior joint (case1,row4,column6)
c1r4ijlc = c1r1ijlc+c1r2ijlc+c1r3ijlc; % Sum of Moments in members for case 1 -↙
lower column of interior joint (case1,row4,column7)
c2r1ejuc = 0; % Moment in members for case 2 - upper column of end joint (case1,↙
row1,column1)
c2r1ejb = -mbendmomentmax; % Moment in members for case 2 - beam of end joint↙
(case1,row1,column2)
c2r1ejlc = 0; % Moment in members for case 2 - beam of end joint (case1,row1,↙
column3)
c2r1ijuc = 0; % Moment in members for case 2 - upper column of interior joint↙
(case1,row1,column4)
c2r1ijeb = mbendmomentmax; % Moment in members for case 2 - end beam of interior↙
joint (case1,row1,column5)
c2r1ijib = -mbintmomentmin; % Moment in members for case 2 - interior beam of↙
interior joint (case1,row1,column6)
c2r1ijlc = 0; % Moment in members for case 2 - lower column of interior joint↙
(case1,row1,column7)
c2r2ejuc = r5ejuc*mbendmomentmax; % Moment in members for case 2 - upper column of↙
end joint (case1,row2,column1)
c2r2ejb = r5ejb*mbendmomentmax; % Moment in members for case 2 - beam of end joint↙
(case2,row1,column2)
c2r2ejlc = r5ejlc*mbendmomentmax; % Moment in members for case 2 - beam of end↙
joint (case1,row2,column3)
c2r2ijuc = r5ijuc*mbendmomentmax; % Moment in members for case 2 - upper column of↙
interior joint (case1,row2,column4)
c2r2ijeb = r5ijeb*mbendmomentmax; % Moment in members for case 2 - end beam of↙
interior joint (case1,row2,column5)
c2r2ijib = r5ijib*mbendmomentmax; % Moment in members for case 2 - interior beam of↙
interior joint (case1,row2,column6)
c2r2ijlc = r5ijlc*mbendmomentmax; % Moment in members for case 2 - lower column of↙
interior joint (case1,row2,column7)
c2r3ejuc = r6ejuc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
upper column of end joint (case1,row3,column1)
c2r3ejb = r6ejb*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
```

```
beam of end joint (case2,row3,column2)
c2r3ejlc = r6ejlc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -
beam of end joint (case1,row3,column3)
c2r3ijuc = r6ijuc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -
upper column of interior joint (case1,row3,column4)
c2r3ijeb = r6ijeb*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -
end beam of interior joint (case1,row3,column5)
c2r3ijib = r6ijib*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -
interior beam of interior joint (case1,row3,column6)
c2r3ijlc = r6ijlc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -
lower column of interior joint (case1,row3,column7)
c2r4ejuc = c2r1ejuc+c2r2ejuc+c2r3ejuc; % Sum of Moments in members for case 2 -
upper column of end joint (case1,row4,column1)
c2r4ejb = c2r1ejb+c2r2ejb+c2r3ejb; % Sum of Moments in members for case 2 - beam of
end joint (case2,row4,column2)
c2r4ejlc = c2r1ejlc+c2r2ejlc+c2r3ejlc; % Sum of Moments in members for case 2 -
beam of end joint (case1,row4,column3)
c2r4ijuc = c2r1ijuc+c2r2ijuc+c2r3ijuc; % Sum of Moments in members for case 2 -
upper column of interior joint (case1,row4,column4)
c2r4ijeb = c2r1ijeb+c2r2ijeb+c2r3ijeb; % Sum of Moments in members for case 2 - end
beam of interior joint (case1,row4,column5)
c2r4ijib = c2r1ijib+c2r2ijib+c2r3ijib; % Sum of Moments in members for case 2 -
interior beam of interior joint (case1,row4,column6)
c2r4ijlc = c2r1ijlc+c2r2ijlc+c2r3ijlc; % Sum of Moments in members for case 2 -
lower column of interior joint (case1,row4,column7)
c3r1ejuc = 0; % Moment in members for case 3 - upper column of end joint (case1,
row1,column1)
c3r1ejb = -mbintmomentmin; % Moment in members for case 3 - beam of end joint
(case1,row1,column2)
c3r1ejlc = 0; % Moment in members for case 3 - beam of end joint (case1,row1,
column3)
c3r1ijuc = 0; % Moment in members for case 3 - upper column of interior joint
(case1,row1,column4)
c3r1ijeb = mbintmomentmin; % Moment in members for case 3 - end beam of interior
joint (case1,row1,column5)
c3r1ijib = -mbendmomentmax; % Moment in members for case 3 - interior beam of
interior joint (case1,row1,column6)
c3r1ijlc = 0; % Moment in members for case 3 - lower column of interior joint
(case1,row1,column7)
c3r2ejuc = r5ejuc*mbintmomentmin; % Moment in members for case 3 - upper column of
end joint (case1,row2,column1)
c3r2ejb = r5ejb*mbintmomentmin; % Moment in members for case 3 - beam of end joint
(case2,row1,column2)
c3r2ejlc = r5ejlc*mbintmomentmin; % Moment in members for case 3 - beam of end
joint (case1,row2,column3)
c3r2ijuc = r5ijuc*mbintmomentmin; % Moment in members for case 3 - upper column of
interior joint (case1,row2,column4)
c3r2ijeb = r5ijeb*mbintmomentmin; % Moment in members for case 3 - end beam of
interior joint (case1,row2,column5)
c3r2ijib = r5ijib*mbintmomentmin; % Moment in members for case 3 - interior beam of
interior joint (case1,row2,column6)
c3r2ijlc = r5ijlc*mbintmomentmin; % Moment in members for case 3 - lower column of
interior joint (case1,row2,column7)
```

```
c3r3ejuc = r6ejuc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
upper column of end joint (case1,row3,column1)
c3r3ejb = r6ejb*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
beam of end joint (case2,row3,column2)
c3r3ejlc = r6ejlc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
beam of end joint (case1,row3,column3)
c3r3ijuc = r6ijuc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
upper column of interior joint (case1,row3,column4)
c3r3ijeb = r6ijeb*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
end beam of interior joint (case1,row3,column5)
c3r3ijib = r6ijib*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
interior beam of interior joint (case1,row3,column6)
c3r3ijlc = r6ijlc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
lower column of interior joint (case1,row3,column7)
c3r4ejuc = c3r1ejuc+c3r2ejuc+c3r3ejuc; % Sum of Moments in members for case 3 -
upper column of end joint (case1,row4,column1)
c3r4ejb = c3r1ejb+c3r2ejb+c3r3ejb; % Sum of Moments in members for case 3 - beam of
end joint (case2,row4,column2)
c3r4ejlc = c3r1ejlc+c3r2ejlc+c3r3ejlc; % Sum of Moments in members for case 3 -
beam of end joint (case1,row4,column3)
c3r4ijuc = c3r1ijuc+c3r2ijuc+c3r3ijuc; % Sum of Moments in members for case 3 -
upper column of interior joint (case1,row4,column4)
c3r4ijeb = c3r1ijeb+c3r2ijeb+c3r3ijeb; % Sum of Moments in members for case 3 - end
beam of interior joint (case1,row4,column5)
c3r4ijib = c3r1ijib+c3r2ijib+c3r3ijib; % Sum of Moments in members for case 3 -
interior beam of interior joint (case1,row4,column6)
c3r4ijlc = c3r1ijlc+c3r2ijlc+c3r3ijlc; % Sum of Moments in members for case 3 -
lower column of interior joint (case1,row4,column7)

% 2.1.3.1 FINAL BENDING MOMENT AND SHEAR FORCE ANALYSIS/ SUB FRAME ANALYSIS
c1r1bendsupport = c1r4ejb; % Load case 1 - Beam moment end support
mbvl = (mbtoalmaxload*x(19)/2)-((c1r4ijeb-abs(c1r4ejb))/x(19)); % Main beam - Left
support moment for load case 1
mbvr = (mbtoalmaxload*x(19))-(mbvl); % Main beam - Right support moment for load
case 1
mba = mbvl/mbtoalmaxload; % Main beam load case 1 - distance to zero shear
c1mbmaxsagging = (mbvl*mba/2)- abs(c1r4ejb); % Load case 1 main beam - maximum
sagging
c1r1bendspan = c1mbmaxsagging; % Load case 1 - Beam moment end span
c1r1bisleft = c1r4ijeb; % Load case 1 - Beam moment interior left support
c1r1bisright = c1r4ijib; % Load case 1 - Beam moment interior right support
c1r1binteriorspan = ((mbtoalmaxload*x(19)*x(19))/8)-abs(c1r1bisright); % Load case
1 - Beam moment interior span
c1r2ucendsupport = c1r4ejuc; % Load case 1 - Upper column moment end support
c1r2ucendspan = 0; % Load case 1 - Upper column moment end span
c1r2ucisleft = c1r4ijuc; % Load case 1 - Upper column moment interior left support
c1r2ucisright = 0; % Load case 1 - Upper column moment interior right support
c1r2ucinteriorspan = 0; % Load case 1 - Upper column moment interior span
c1r3lcendsupport = c1r4ejlc; % Load case 1 - Upper column moment end support
c1r3lcendspan = 0; % Load case 1 - Upper column moment end span
c1r3lcisleft = c1r4ijlc; % Load case 1 - Upper column moment interior left support
c1r3lcisright = 0; % Load case 1 - Upper column moment interior right support
c1r3lcinteriorspan = 0; % Load case 1 - Upper column moment interior span
```

```
c2r1bendsupport = c2r4ejb; % Load case 2 - Beam moment end support
c2mbvl = (mbtoalmaxload*x(19)/2)-((c2r4ijeb-abs(c2r4ejb))/x(19)); % Main beam -
Left support moment for load case 2
c2mbvr = (mbtoalmaxload*x(19))-(c2mbvl); % Main beam - Right support moment for
load case 2
c2mba = c2mbvl/mbtoalmaxload; % Main beam load case 2 - distance to zero shear
c2mbmaxsagging = (c2mbvl*mba/2)- abs(c2r4ejb); % Load case 2 main beam - maximum
sagging
c2r1bendspan = c2mbmaxsagging; % Load case 2 - Beam moment end span
c2r1bisleft = c2r4ijeb; % Load case 2 - Beam moment interior left support
c2r1bisright = c2r4ijib; % Load case 2 - Beam moment interior right support
c2r1binteriorspan = ((mbtoalmaxload*x(19)*x(19))/8)-abs(c2r1bisright); % Load case
2 - Beam moment interior span
c2r2ucendsupport = c2r4ejuc; % Load case 2 - Upper column moment end support
c2r2ucendspan = 0; % Load case 2 - Upper column moment end span
c2r2ucisleft = c2r4ijuc; % Load case 2 - Upper column moment interior left support
c2r2ucisright = 0; % Load case 2 - Upper column moment interior right support
c2r2ucinteriorspan = 0; % Load case 2 - Upper column moment interior span
c2r3lcendsupport = c2r4ejlc; % Load case 2 - Upper column moment end support
c2r3lcendspan = 0; % Load case 2 - Upper column moment end span
c2r3lcisleft = c2r4ijlc; % Load case 2 - Upper column moment interior left support
c2r3lcisright = 0; % Load case 2 - Upper column moment interior right support
c2r3lcinteriorspan = 0; % Load case 2 - Upper column moment interior span

c3r1bendsupport = c3r4ejb; % Load case 3 - Beam moment end support
c3mbvl = (mbtoalmaxload*x(19)/2)-((c3r4ijeb-abs(c3r4ejb))/x(19)); % Main beam -
Left support moment for load case 3
c3mbvr = (mbtoalmaxload*x(19))-(c3mbvl); % Main beam - Right support moment for
load case 3
c3mba = c3mbvl/mbtoalmaxload; % Main beam load case 3 - distance to zero shear
c3mbmaxsagging = (c3mbvl*mba/2)- abs(c3r4ejb); % Load case 3 main beam - maximum
sagging
c3r1bendspan = c3mbmaxsagging; % Load case 3 - Beam moment end span
c3r1bisleft = c3r4ijeb; % Load case 3 - Beam moment interior left support
c3r1bisright = c3r4ijib; % Load case 3 - Beam moment interior right support
c3r1binteriorspan = ((mbtoalmaxload*x(19)*x(19))/8)-abs(c3r1bisright); % Load case
3 - Beam moment interior span
c3ucendsupport = c3r4ejuc; % Load case 3 - Upper column moment end support
c3r2ucendspan = 0; % Load case 3 - Upper column moment end span
c3r2ucisleft = c3r4ijuc; % Load case 3 - Upper column moment interior left support
c3r2ucisright = 0; % Load case 3 - Upper column moment interior right support
c3r2ucinteriorspan = 0; % Load case 3 - Upper column moment interior span
c3r3lcendsupport = c3r4ejlc; % Load case 3 - Upper column moment end support
c3r3lcendspan = 0; % Load case 3 - Upper column moment end span
c3r3lcisleft = c3r4ijlc; % Load case 3 - Upper column moment interior left support
c3r3lcisright = 0; % Load case 3 - Upper column moment interior right support
c3r3lcinteriorspan = 0; % Load case 3 - Upper column moment interior span

c1r1sfbendsupport = mbvl; % Load case 1 - Beam shear force end support
c1r1sfbendendspan = 0; % Load case 1 - Beam shear force end span
c1r1sfbisleft = mbvr; % Load case 1 - Beam shear force interior left support
c1r1sfbisright = (mbtoalmaxload*x(19))/2; % Load case 1 - Beam shear force interior
```

```
right support
c1r1sfbinteriorspan = 0; % Load case 1 - Beam shear force interior span

c2r2sfbendsupport = abs(c2mbvl); % Load case 2 - Beam shear force end support
c2r2sfbendendspan = 0; % Load case 2 - Beam shear force end span
c2r2sfbisleft = c2mbvr; % Load case 2 - Beam shear force interior left support
c2r2sfbisright = (mbtotalminload*x(19))/2; % Load case 2 - Beam shear force↙
interior right support
c2r2sfbinteriorspan = 0; % Load case 2 - Beam shear force interior span

c3r3sfbendsupport = c3mbvl; % Load case 3 - Beam shear force end support
c3r3sfbendendspan = 0; % Load case 3 - Beam shear force end span
c3r3sfbisleft = c3mbvr; % Load case 3 - Beam shear force interior left support
c3r3sfbisright = (mbtoalmaxload*x(19))/2; % Load case 3 - Beam shear force interior↙
right support
c3r3sfbinteriorspan = 0; % Load case 3 - Beam shear force interior span

if abs(c2r1bendsupport)>abs(c1r1bendsupport)
   mbendsupportmoment = abs(c2r1bendsupport);% At end support
else
   mbendsupportmoment = abs(c2r1bendsupport); % At end support
end
mbinteriorsupportmoment = abs(c2r1bisleft); % At interior support
mbinteriorspanmoment = ((mbtoalmaxload*x(19)*x(19))/8)-mbinteriorsupportmoment; %↙
At interior span

% 2.1.4 FLEXURE DESIGN/CHECK
% 2.1.4.1 At interior support

mbeffectivedepth = x(23)-bnominalcover-10-(18/2); %Effective depth of the main beam
mbk = (mbinteriorsupportmoment*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x↙
(3));
mbkdash = 0.168;
mbleverarm = 0.5*mbeffectivedepth*(1+(1-(3.53*mbk))^(1/2)); % in mm
mbmaxleverarm = 0.95*mbeffectivedepth;
 if mbleverarm <= mbmaxleverarm
    mbz = mbleverarm;
 else
    mbz = mbmaxleverarm;
 end
mbrequiredsteel = (mbinteriorsupportmoment*1000000)/(0.87*x(4)*mbz); % Main beam↙
tensile steel at interior support
mbminimumsteel = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main beam↙
minimum tensile steel at interior support
mbprovidedsteel = x(26)*3.14*x(27)*x(27)/4; % Main beam provided tensile steel at↙
interior support

% 2.1.4.2 At end support

mbkes = (mbendsupportmoment*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x↙
(3));
mbkdashes = 0.168;
mbleverarmes = 0.5*mbeffectivedepth*(1+(1-(3.53*mbkes))^(1/2)); % in mm
```

```
mbmaxleverarmes = 0.95*mbeffectivedepth;
 if mbleverarmes <= mbmaxleverarmes
    mbzes = mbleverarmes;
 else
    mbzes = mbmaxleverarmes;
 end
mbrequiredsteeles = (mbendsupportmoment*1000000)/(0.87*x(4)*mbzes); % Main beam
tensile steel at interior support
mbminimumsteeles = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main beam
minimum tensile steel at interior support
mbprovidedsteeles = x(28)*3.14*x(29)*x(29)/4; % Main beam provided tensile steel at
interior support

% 2.1.4.3 At interior span

mbkis = (mbinteriorspanmoment*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x
(3));
mbkdashis = 0.168;
mbleverarmis = 0.5*mbeffectivedepth*(1+(1-(3.53*mbkis))^(1/2)); % in mm
mbmaxleverarmis = 0.95*mbeffectivedepth;
 if mbleverarmis <= mbmaxleverarmis
    mbzis = mbleverarmis;
 else
    mbzis = mbmaxleverarmis;
 end
mbrequiredsteelis = (mbinteriorspanmoment*1000000)/(0.87*x(4)*mbzis); % Main beam
tensile steel at interior span
mbminimumsteelis = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main beam
minimum tensile steel at interior span
mbprovidedsteelis = x(30)*3.14*x(31)*x(31)/4; % Main beam provided tensile steel at
interior span

% 2.1.4.4 At exterior span

mbkespan = (abs(c2r1bendspan)*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x
(3));
mbkdashespan = 0.168;
mbleverarmespan = 0.5*mbeffectivedepth*(1+(1-(3.53*mbkespan))^(1/2)); % in mm
mbmaxleverarmespan = 0.95*mbeffectivedepth;
 if mbleverarmespan <= mbmaxleverarmespan
    mbzespan = mbleverarmespan;
 else
    mbzespan = mbmaxleverarmespan;
 end
mbrequiredsteelespan = (abs(c2r1bendspan)*1000000)/(0.87*x(4)*mbzespan); % Main
beam tensile steel at exterior span
mbminimumsteelespan = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main
beam minimum tensile steel at exterior span
mbprovidedsteelespan = x(32)*3.14*x(33)*x(33)/4; % Main beam provided tensile steel
at exterior span

% 2.1.5 SHEAR DESIGN/CHECK
```

```
mbcriticaldistance = mbeffectivedepth+(x(24)/2);

% 2.1.5.1 At End Support

mbsdendsupport = c2r2sfbendsupport-(mbtoalmaxload*mbcriticaldistance/1000); % Main
beam shear design end support
mbsdesroww = mbsdendsupport/(x(18)*0.9*mbeffectivedepth*(1-(x(3))/250)*x(3)); %
Main beam shear design factor(row)
mbsdesrowwlimit = 0.138;
mbcottheta = 2.5;
mbsdesrequiredsteel = (mbsdendsupport*1000*1000)/(0.87*x(4)*0.
9*mbeffectivedepth*mbcottheta); % Main beam shear design end support required steel
mbsdesprovidedsteel = x(34)*3.14*x(35)*x(35)/4; % % Main beam shear design end
support provided steel

% 2.1.5.2 At Interior Support

mbsdisleft = abs(c1r1sfbisleft)-(mbtoalmaxload*mbcriticaldistance/1000); % Main
beam shear design interior support
mbsdisleftroww = mbsdisleft/(x(18)*0.9*mbeffectivedepth*(1-(x(3))/250)*x(3)); %
Main beam shear design factor(row)
mbsdisleftrowwlimit = 0.138;
mbsdisleftcottheta = 2.5;
mbsdisleftrequiredsteel = (mbsdisleft*1000*1000)/(0.87*x(4)*0.
9*mbeffectivedepth*mbsdisleftcottheta); % Main beam shear design end support
required steel
mbsdisleftprovidedsteel = x(36)*3.14*x(37)*x(37)/4; % % Main beam shear design end
support provided steel

mbsdisright = abs(c1r1sfbisright)-(mbtoalmaxload*mbcriticaldistance/1000); % Main
beam shear design interior support
mbsdisrightroww = mbsdisright/(x(18)*0.9*mbeffectivedepth*(1-(x(3))/250)*x(3)); %
Main beam shear design factor(row)
mbsdisrightrowwlimit = 0.138;
mbsdisrightcottheta = 2.5;
mbsdisrightrequiredsteel = (mbsdisright*1000*1000)/(0.87*x(4)*0.
9*mbeffectivedepth*mbsdisrightcottheta); % Main beam shear design end support
required steel
mbsdisrightprovidedsteel = x(38)*3.14*x(39)*x(39)/4; % % Main beam shear design end
support provided steel

% 2.1.6 DEFLECTION DESIGN/CHECK

mbactualspantodepth = x(19)/mbeffectivedepth;
mbdeflectionload = sf;
mbbeta = (500/x(4))/(mbprovidedsteelis/mbrequiredsteelis);
mbdeflectioneffectivebreadth = ((x(18)+(0.28*x(19)*1000))*x(1))+(x(18)*
(mbeffectivedepth-x(1)));
mbalpha = (0.55+(0.0075*x(3)/(100*mbrequiredsteelis/mbdeflectioneffectivebreadth)))
+(0.005*(x(3)^0.5)*(((x(3)^0.5)/
(100*mbrequiredsteelis/mbdeflectioneffectivebreadth))-10)^1.5);
mblimitingratio = 30*0.8*(x(19)/(mbdeflectioneffectivebreadth/x(18)))
*mbbeta*mbalpha;
```

```
% 2.1.7 REINFOREMENT REQUIREMETNS/DETAILING
% 2.1.7.1 STEEL CALCULATION

mbbrespanl = ((x(19)*1000-x(24))+(50*x(33))+(x(24)/2)+(50*x(33)/2)-(2*x(33))) ↙
/1000;% Main beam bottom reinforcement end span cut length
mbbrespanw = mbbrespanl*2*x(33)*x(33)/162.2; % Weight of end span bottom↙
reinforcement in Kg
mbbrispanl = ((x(19)*1000-x(24))+(50*x(31))+(x(24)/2)+(50*x(31)/2)-(2*x(31))) ↙
/1000;% Main beam bottom reinforcement interior span cut length
mbbrispanw = mbbrispanl*(x(20)-2)*x(31)*x(31)/162.2; % Weight of interior span↙
bottom reinforcement in Kg
mbtresupportl = ((((x(19)*1000)-x(24))/3)+(50*x(29)))/1000;% Main beam top↙
reinforcement exterior support cut length
mbbresupportw = mbtresupportl*2*x(29)*x(29)/162.2; % Weight of exterior support↙
bottom reinforcement in Kg
mbtrisupportl = ((((0.2*0.15*(x(19)*1000*2))*2)+x(24)))/1000;% Main beam top↙
interior support cut length
mbbrisupportw = mbtrisupportl*(x(20)-1)*x(27)*x(27)/162.2; % Weight of interior↙
support bottom reinforcement in Kg
mbstirrupa = x(18)-(2*bnominalcover)-(2*x(37)/2); % Main beam stirrup breadth
mbstirrupb = x(23)-(2*bnominalcover)-(2*x(37)/2); % Main beam stirrup breadth
mbstirrup = ((2*(mbstirrupa+mbstirrupb))+(10*2*x(37))-(3*4*x(37)))/1000; % Main↙
beam stirrup length
mbstirrupnumber = ((x(19)*1000-x(18))/(1000/x(36)))-1; % Number of stirrups
mbstirrupweight = mbstirrup*mbstirrupnumber*x(20)*x(37)/162.2; %Total stirrup↙
weight for entire spans
mbtotalsteelonebeam =↙
mbstirrupweight+mbbrespanw+mbbrispanw+mbbresupportw+mbbrisupportw; % Total steel↙
weight for one main beam
mbtotalsteel = mbtotalsteelonebeam*(x(17)+1)*6; % Total main beam steel for entire↙
building

% 2.1.7.2 Concrete CALCULATION

mbcvolume = x(18)*x(23)*x(20)*x(19)/1000000;% Main beam volue of one total beam in↙
m3
mbcfloorvolume = mbcvolume*(x(17)+1); % Main beam volume on one floor in m3
mbcbuildingvolume = mbcfloorvolume*6; % Main beam volume on one floor in m3
mbcbuildingnetweight = (mbcbuildingvolume*2400)-(mbtotalsteel); % Main beam net↙
weight for whole building in m3
mbcbuildingnetvolume = mbcbuildingnetweight/2400;  % Main beam net volume on whole↙
builsing in m3

% 2.1.7.3 Formwork CALCULATION

mbfarea = (2*x(23)/1000+x(18)/1000)*x(19)*x(20)*(x(17)+1)*6; % Main beam formwork↙
area of whole building in m2
mbfweight = mbfarea*(4/1000)*2710; % Main beam formwork weight of whole building in↙
Kg

% 2.2 EDGE BEAM DESIGN
% 2.2.1 LOADING CALULATIONS
```

```
ebtarea = 0.5*(x(2)-x(24)/1000)*(1/3); % Edge beam triangular area m2
ebwcw = 5; % Edge beam loading due to walling, cladding, windows in Kn/m
ebwb = 1.25*(ebwcw+(25*x(18)*x(23))/1000000)*x(2); % Edge beam load plus walling in KN
ebsdload = (1.25*ebtarea*stotalpermanentload); % Edge beam dead load due to slab in KN
ebslload = (1.25*ebtarea*stotalvariableload); % Edge beam live load due to slab in KN

% 2.2.2 BENDING MOMENT AND SHEAR FORCE ANALYSIS/ SUB FRAME ANALYSIS

ebr1bmes = ((0.078*ebwb)+(0.105*ebsdload)+(0.135*ebslload))*x(2); % Edge beam bending moment in end span Knm
ebr2bmisupport = ((0.105*ebwb)+(0.132*ebsdload)+(0.132*ebslload))*x(2); % Edge beam bending moment in first interior support KNm
ebr3bmispan = ((0.046*ebwb)+(0.068*ebsdload)+(0.117*ebslload))*x(2); % Edge beam bending moment in interior span KNm
ebr4bmosupport = ((0.079*ebwb)+(0.099*ebsdload)+(0.099*ebslload))*x(2); % Edge beam bending moment in other support KNm

ebr1sfes = ((0.395*ebwb)+(0.369*ebsdload)+(0.434*ebslload)); % Edge beam shear force in end span Knm
ebr2sfisupport = ((0.605*ebwb)+(0.631*ebsdload)+(0.649*ebslload)); % Edge beam shear force in first interior support KNm
ebr3sfispan = ((0.526*ebwb)+(0.532*ebsdload)+(0.622*ebslload)); % Edge beam shear force in interior span KNm
ebr4sfosupport = ((0.5*ebwb)+(0.5*ebsdload)+(0.614*ebslload)); % Edge beam shear force in other support KNm

% 2.2.3 FLEXURE DESIGN/CHECK

ebeffectivedepth = mbeffectivedepth; % Edge beam effective depth
ebeffectivebreadth = x(18)+(0.2*0.7*x(2)*1000);

% 2.2.3.1 At END SPAN

ebkes = (ebr1bmes*1000000)/(ebeffectivebreadth*ebeffectivedepth*ebeffectivedepth*x(3)); % Edge beam end span
ebkdashes = 0.168;
ebleverarmes = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkes))^(1/2)); % in mm
ebmaxleverarmes = 0.95*ebeffectivedepth;
 if ebleverarmes <= ebmaxleverarmes
    ebzes = ebleverarmes;
 else
    ebzes = ebmaxleverarmes;
 end
ebrequiredsteeles = (ebr1bmes*1000000)/(0.87*x(4)*ebzes); % Main beam tensile steel at interior support
ebminimumsteeles = (0.26*0.3*x(3)^(2/3)*ebeffectivedepth*ebeffectivebreadth)/x(4); % Main beam minimum tensile steel at interior support
ebprovidedsteeles = x(40)*3.14*x(41)*x(41)/4; % Main beam provided tensile steel at interior support
```

```
% 2.2.3.2 At 1st INTERIOR SUPPORT

ebkis = (ebr2bmisupport*1000000)/(x(18)*ebeffectivedepth*ebeffectivedepth*x(3)); %
Edge beam end span
ebkdashis = 0.168;
ebleverarmis = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkis))^(1/2)); % in mm
ebmaxleverarmis = 0.95*ebeffectivedepth;
 if ebleverarmis <= ebmaxleverarmis
    ebzis = ebleverarmis;
 else
    ebzis = ebmaxleverarmis;
 end
ebrequiredsteelis = (ebr2bmisupport*1000000)/(0.87*x(4)*ebzis); % Main beam tensile
steel at interior support
ebminimumsteelis = (0.26*0.3*x(3)^(2/3)*x(18)*ebeffectivedepth)/x(4); % Main beam
minimum tensile steel at interior support
ebprovidedsteelis = x(42)*3.14*x(43)*x(43)/4; % Main beam provided tensile steel at
interior support

% 2.2.3.3 At Interior SPAN

ebkispan = (ebr3bmispan*1000000)/
(ebeffectivebreadth*ebeffectivedepth*ebeffectivedepth*x(3)); % Edge beam interior
span
ebkdashispan = 0.168;
ebleverarmispan = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkispan))^(1/2)); % in mm
ebmaxleverarmispan = 0.95*ebeffectivedepth;
 if ebleverarmispan <= ebmaxleverarmispan
    ebzispan = ebleverarmispan;
 else
    ebzispan = ebmaxleverarmispan;
 end
ebrequiredsteelispan = (ebr3bmispan*1000000)/(0.87*x(4)*ebzispan); % Main beam
tensile steel at interior span
ebminimumsteelispan = (0.26*0.3*x(3)^(2/3)*ebeffectivedepth*ebeffectivebreadth)/x
(4); % Main beam minimum tensile steel at interior span
ebprovidedsteelispan = x(44)*3.14*x(45)*x(45)/4; % Main beam provided tensile steel
at interior span

% 2.2.3.4 At OTHER INTERIOR SUPPORT

ebkos = (ebr4bmosupport*1000000)/(x(18)*ebeffectivedepth*ebeffectivedepth*x(3)); %
Edge beam OTHER INTERIOR SUPPORT
ebkdashos = 0.168;
ebleverarmos = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkos))^(1/2)); % in mm
ebmaxleverarmos = 0.95*ebeffectivedepth;
 if ebleverarmos <= ebmaxleverarmos
    ebzos = ebleverarmos;
 else
    ebzos = ebmaxleverarmos;
 end
ebrequiredsteelos = (ebr4bmosupport*1000000)/(0.87*x(4)*ebzos); % Main beam tensile
```

```
steel at interior support
ebminimumsteelos = (0.26*0.3*x(3)^(2/3)*x(18)*ebeffectivedepth)/x(4); % Main beam
minimum tensile steel at interior support
ebprovidedsteelos = x(46)*3.14*x(47)*x(47)/4; % Main beam provided tensile steel at
interior support

% 2.2.4 SHEAR DESIGN/CHECK

ebsdrow = ebr2sfisupport/(x(18)*0.87*0.9*ebeffectivedepth/1000*(1-(x(3))/250)*x
(3)); % Main beam shear design factor(row)
ebsdrowlimit = 0.138;
ebcottheta = 2.5;
ebsdrequiredsteel = (ebr2sfisupport*1000)/(0.87*x(4)*0.
9*ebeffectivedepth/1000*mbcottheta); % Main beam shear design end support required
steel
ebsdprovidedsteel = x(48)*3.14*x(49)*x(49)/4; % % Main beam shear design end
support provided steel

% 2.2.5 REINFOREMENT REQUIREMETNS/DETAILING
% 2.2.5.1 STEEL CALCULATION

ebbrespanl = ((x(2)*1000-x(24))+(50*x(41))+(x(24)/2)+(50*x(41)/2)-(2*x(41)))/1000;%
Edge beam bottom reinforcement end span cut length
ebbrespanw = ebbrespanl*2*x(41)*x(41)/162.2; % Weight of end span bottom
reinforcement in Kg
ebbrispanl = ((x(2)*1000-x(24))+(50*x(45))+(x(24)/2)+(50*x(45)/2)-(2*x(45)))/1000;%
Edge beam bottom reinforcement interior span cut length
ebbrispanw = ebbrispanl*(x(17)-2)*x(45)*x(45)/162.2; % Weight of interior span
bottom reinforcement in Kg
ebtrisupportl = (((x(2)*1000)/3)+(50*x(43)))/1000;% Edge beam top reinforcement
first interior support cut length
ebtrisupportw = ebtrisupportl*2*x(43)*x(43)/162.2; % Weight of first interior
support top reinforcement in Kg
ebtrosupportl = (((0.9*x(24))*2)+x(24))/1000;% Edge beam top other interior support
cut length
ebtrosupportw = ebtrosupportl*(x(17)-1)*x(47)*x(47)/162.2; % Weight of other
interior support top reinforcement in Kg
ebstirrupa = x(18)-(2*bnominalcover)-(2*x(49)/2); % Edge beam stirrup breadth
ebstirrupb = x(23)-(2*bnominalcover)-(2*x(49)/2); % Edge beam stirrup breadth
ebstirrup = ((2*(ebstirrupa+ebstirrupb))+(10*2*x(49))-(6*2*x(49)))/1000; % Edge
beam stirrup length
ebstirrupnumber = ((x(2)*1000-x(18))/(1000/x(48)))-1; % Number of stirrups
ebstirrupweight = ebstirrup*ebstirrupnumber*x(17)*x(49)/162.2; %Total stirrup
weight for entire spans
ebtotalsteelonebeam =
ebstirrupweight+ebbrespanw+ebbrispanw+ebtrisupportw+ebtrosupportw; % Total steel
weight for one main beam
ebtotalsteel = ebtotalsteelonebeam*2*6; % Total edge beam steel for entire building

% 2.2.5.2 Concrete CALCULATION

ebcvolume = x(18)*x(23)*x(17)*x(2)/1000000; % Main beam volue of one total beam in
m3
```

```
ebcfloorvolume = ebcvolume*2; % Main beam volume on one floor in m3
ebcbuildingvolume = ebcfloorvolume*6; % Main beam volume for whole building in m3
ebcbuildingnetweight = (ebcbuildingvolume*2400)-(ebtotalsteel); % Main beam net↙
weight on one floor in m3
ebcbuildingnetvolume = ebcbuildingnetweight/2400 ; % Edge beam net volume on whole↙
builsing in m3

% 2.5.5.3 Formwork CALCULATION

ebfarea = (2*x(23)/1000+x(18)/1000)*x(17)*x(2)*2*6; % Edge beam formwork area of↙
whole building in m2
ebfweight = ebfarea*(4/1000)*2710; % Edge beam formwork weight of whole building in↙
Kg


% 3. COLUMN DESIGN
% 3.1 LOADING

cdf = (1.25*stotalpermanentload)+(1.5*0.6); % Column design load in KN/m2
croofbeammax = (1.1*x(2)*cdf)+mbmax; % Column max interior roof KN/m
croofbeammin = (1.1*x(2)*1.25*stotalpermanentload)+mbmax; % Column min interior↙
roof KN/m
cswperfloor = 1.25*(x(24)/1000)*(x(25)/1000)*25*(3.5-(x(23)/1000)); % Column self↙
weight per floor

% 3.2 EXTERNAL COLUMN
% 3.2.1 BENDING MOMENT AND AXIAL FORCE ANALYSIS

ecfeb = ebwb; % External column load due to edge beam KN
exroof = (1.25*((x(24)*x(25))+0.15*1)*25*x(2)); % External column roof load due to↙
self weight of beam and parapet

ecfminload = (1.1*x(2)*1.5*stotalvariableload)+(mbmax/1000000); % External column↙
minimum load
ecfmaxload = 1.1*x(2)*1.25*stotalpermanentload; % External column maximum load
ecmend = (ecfminload*x(19)*x(19))/12; % External column end moment minimum
ecmendmax = (ecfmaxload*x(19)*x(19))/12; % External column end moment maximum
ecmleft = -ecmend+(0.543*ecmend);
ecmright = ecmend+(0.169*ecmend);
ecsright = (ecfminload*x(19)/2)-((ecmright+ecmleft)/x(19)); % External column shear↙
right
ecmleft1 = -ecmendmax+(0.543*ecmendmax)+((ecmend-ecmendmax)*0.102);
ecmright1 = ecmendmax+(0.169*ecmendmax)+((ecmend-ecmendmax)*0.407);
ecsright1 = (ecmendmax*x(19)/2)-((ecmright1+ecmleft1)/x(19)); % External column↙
shear right

ecmendminr = (croofbeammin*x(19)*x(19))/12; % External column end moment minimum↙
roof beam
ecmendmaxr = (croofbeammax*x(19)*x(19))/12; % External column end moment maximum↙
roof beam
ecmrlc1 = 0.228*ecmendmaxr; % For load case 1 moment
ecmrlc2 = (0.228*ecmendmaxr)+((ecmendminr-ecmendmaxr)*(-0.051)); % For load case 2↙
moment
```

```
ecmrlc1left = -ecmendmaxr+(0.543*ecmendmaxr); % Left moment load case 1
ecmrlc1right = ecmendmaxr+(0.169*ecmendmaxr); % Right moment load case 1
ecsrlc1 = (croofbeammin*x(19)/2)-((ecmrlc1right+ecmrlc1left)/x(19)); % Shear load↙
case 1
ecmrlc2left = 0.228*ecmendmaxr+((ecmendminr-ecmendmaxr)*(-0.051)); % Left moment↙
load case 2
ecmrlc2right = ecmendmaxr+(0.169*ecmendmaxr)+(0.407*(ecmendminr-ecmendmaxr)); %↙
Right moment load case 2
ecsrlc2 = (croofbeammin*x(19)/2)-((ecmrlc2right+ecmrlc2left)/x(19)); % Shear load↙
case 2

ecroofbeam = (1.1*x(2)*1.5*0.6)+mbmax; % Column max interior roof imposed load KN/m
ecmroofbeam = (ecroofbeam*x(19)*x(19))/12; % External column end moment due to↙
imposed load
ecmroofbeamleft = 0.543*ecmroofbeam;
ecmroofbeamright = ecmroofbeam+(0.169*ecmroofbeam);
ecsroofbeam = (ecroofbeam*x(19)/2)-((ecmroofbeamright-ecmroofbeamleft)/x(19));

ec11r1n = ecsrlc1+ecfeb; % Roof beam
ec11r1m = ecmrlc1;
ec12r1n = ecsrlc2+ecfeb;
ec12r1m = ecmrlc2;
ec21r1n = ecsroofbeam+exroof;

ec11r2n = cswperfloor; % Column
ec12r2n = cswperfloor;

ec11r3n = ec11r2n+ec11r1n;
ec11r3m = c1r2ucendsupport;
ec12r3n = ec12r2n+ec12r1n;
ec12r3m = c2r2ucendsupport;

ec11r4n = c1r1sfbendsupport+ecfeb; % 4th floor beam
ec12r4n = c2r2sfbendsupport+ecfeb;
ec21r4n = ecsright+ecfeb;
ec22r4n = ecsright1+ecfeb;

ec11r5n = ec11r4n+ec11r3n;
ec11r5m = c1r2ucendsupport;
ec12r5n = ec12r4n+ec12r3n;
ec12r5m = c2r2ucendsupport;

ec11r6n = cswperfloor;
ec12r6n = cswperfloor;

ec11r7n = ec11r6n+ec11r5n;
ec11r7m = ec11r5m;
ec12r7n = cswperfloor+ec12r5n;
ec12r7m = ec12r5m;

ec11r8n = c1r1sfbendsupport+ecfeb; % 3th floor beam
ec12r8n = c2r2sfbendsupport+ecfeb;
ec21r8n = ecsright+ecfeb;
```

```
ec22r8n = ecsright1+ecfeb;

ec11r9n = ec11r8n+ec11r7n;
ec11r9m = ec11r5m;
ec12r9n = ec12r7n+ec12r8n;
ec12r9m = ec12r5m;
ec21r9n = ec21r8n+ec21r4n;
ec22r9n = ec22r8n+ec22r4n;

ec11r10n = cswperfloor;
ec12r10n = cswperfloor;

ec11r11n = ec11r10n+ec11r9n;
ec11r11m = ec11r5m;
ec12r11n = ec12r10n+ec12r9n;
ec12r11m = ec12r5m;

ec11r12n = c1r1sfbendsupport+ecfeb; % 2th floor beam
ec12r12n = c2r2sfbendsupport+ecfeb;
ec21r12n = ecsright+ecfeb;
ec22r12n = ecsright1+ecfeb;

ec11r13n = ec11r12n+ec11r11n;
ec11r13m = ec11r5m;
ec12r13n = ec12r12n+ec12r11n;
ec12r13m = ec12r5m;
ec21r13n = ec21r12n+ec21r9n;
ec22r13n = ec22r12n+ec22r9n;

ec11r14n = ec11r5m;
ec12r14n = ec12r5m;

ec11r15n = ec11r14n+ec11r13n; % 1th floor beam
ec11r15m = ec11r5m;
ec12r15n = ec12r14n+ec12r13n;
ec12r15m = ec12r5m;

ec11r16n = c1r1sfbendsupport+ecfeb;
ec12r16n = c2r2sfbendsupport+ecfeb;
ec21r16n = ecsright+ecfeb;
ec22r16n = ecsright1+ecfeb;

ec11r17n = ec11r16n+ec11r15n;
ec11r17m = ec11r5m;
ec12r17n = ec12r16n+ec12r15n;
ec12r17m = ec12r5m;
ec21r17n = ec21r16n+ec21r13n;
ec22r17n = ec22r16n+ec22r13n;

ec11r18n = ec11r5m;
ec12r18n = ec12r5m;

ec11r19n = ec11r18n+ec11r17n; % Ground floor beam
```

```
ec11r19m = ec11r5m;
ec12r19n = ec12r18n+ec12r17n;
ec12r19m = ec12r5m;

ec11r20n = c1r1sfbendsupport+ecfeb;
ec12r20n = c2r2sfbendsupport+ecfeb;
ec21r20n = ecsright+ecfeb;
ec22r20n = ecsright1+ecfeb;

ec11r21n = ec11r20n+ec11r19n; % Basement wall
ec11r21m = ec11r5m;
ec12r21n = ec12r20n+ec12r19n;
ec12r21m = ec12r5m;
ec21r21n = ec21r20n+ec21r17n;
ec22r21n = ec22r20n+ec22r17n;

ecmbotlc2 = ec12r21m; % Moment at bottom from ground to first floor
ecmtoplc2 = -0.5*ecmbotlc2; % Moment at bottom from ground to first floor
ecnedmax = ec12r19n-0.3*ec22r17n; % Ned maximum
ecnedmin = (ec11r17n-(ec21r17n+ec21r1n))/1.25; % Ned minimum
ecnedmintotal = ecnedmax*x(25)/30 ; % Ned minimum total in KNm

% 3.2.2 EFFECTIVE LENGTH AND SLENDERNESS

eclo = 0.75*(3.5-x(23)/1000);% Effective legth
ecm1 = (ecnedmax*eclo)/400; % First order moment
ecm01i = ecmtoplc2+ecm1; % First order moment with imperfections
ecm02i = ecmbotlc2+ecm1; % First order moment with imperfections
eci = x(25)/(12^(1/2));% Radius of gyration
ecslenderness = eclo/eci; % Slenderness ratio
ecn = (ecnedmax)/(x(24)*x(25)*0.85*x(3)/1.5);
ecc = 1.7-(ecm01i/ecm02i);
ecslendernesslimit = (20*0.7*1.1*ecc)/(ecn^(1/2)); % Slenderness ratio limit

% 3.2.3 DESIGN OF CROSS-SECTION

eced = x(25)-(35+8+20/2); % External column effective depth
ecedratio = eced/x(24); % External column effective depth to height
ecg1nedratio = (ecnedmax*1000)/(x(24)*x(25)*x(3)); % ground-first floor axial force↙
ratio
ecg1medratio = (ecm02i*1000000)/(x(24)*x(25)*x(25)*x(3)); % ground-first floor↙
moment ratio
ecg1rs = ((ecg1nedratio*ecg1medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel↙
for ground floor to first floor
ecg1ps = x(50)*3.14*x(51)*x(51)/4; % provided steel for ground floor to first floor

ecmbot12 = ec12r21m; % Moment at bottom from first to second floor
ecnedmax12 = ec12r15n-0.3*ec22r13n; % Ned maximum
ecm12 = (ecnedmax12*eclo)/400; % First order moment
ecm02i12 = ecmbot12+ecm12; % First order moment with imperfections
ecg1nedratio12 = (ecnedmax12*1000)/(x(24)*x(25)*x(3)); % first to second floor↙
axial force ratio
ecg1medratio12 = (ecm02i12*1000000)/(x(24)*x(25)*x(25)*x(3)); % first to second↙
```

```
floor moment ratio
ec12rs = ((ecg1nedratio12*ecg1medratio12*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for first to second floor
ec12ps = x(52)*3.14*x(53)*x(53)/4; % provided steel for first to second floor

ecmbot23 = ec12r21m; % Moment at bottom from second to third floor
ecnedmax23 = ec12r11n-0.3*ec22r9n; % Ned maximum
ecm23 = (ecnedmax23*eclo)/400; % First order moment
ecm02i23 = ecmbot23+ecm23; % First order moment with imperfections
ecg1nedratio23 = (ecnedmax23*1000)/(x(24)*x(25)*x(3)); % second to third floor
floor axial force ratio
ecg1medratio23 = (ecm02i23*1000000)/(x(24)*x(25)*x(25)*x(3)); % second to third
floor moment ratio
ec23rs = ((ecg1nedratio23*ecg1medratio23*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for second to third floor
ec23ps = x(54)*3.14*x(55)*x(55)/4; % provided steel for second to third floor

ecmbot34 = ec12r21m; % Moment at bottom from third to fourth floor
ecnedmax34 = ec12r7n-0.3*ec22r4n; % Ned maximum
ecm34 = (ecnedmax34*eclo)/400; % First order moment
ecm02i34 = ecmbot34+ecm34; % First order moment with imperfections
ecg1nedratio34 = (ecnedmax34*1000)/(x(24)*x(25)*x(3)); % third to fourth floor
floor axial force ratio
ecg1medratio34 = (ecm02i34*1000000)/(x(24)*x(25)*x(25)*x(3)); % third to fourth
floor moment ratio
ec34rs = ((ecg1nedratio34*ecg1medratio34*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for third to fourth floor
ec34ps = x(56)*3.14*x(57)*x(57)/4; % provided steel for third to fourth floor

ecmbot4r = ec12r21m; % Moment at bottom from fourth to roof floor
ecnedmax4r = ec12r7n-0.3*ec22r4n; % Ned maximum
ecm4r = (ecnedmax4r*eclo)/400; % First order moment
ecm02i4r = ecmbot4r+ecm4r; % First order moment with imperfections
ecg1nedratio4r = (ecnedmax4r*1000)/(x(24)*x(25)*x(3)); % fourth to roof floor axial
force ratio
ecg1medratio4r = (ecm02i4r*1000000)/(x(24)*x(25)*x(25)*x(3)); % fourth to roof
floor moment ratio
ec4rrs = ((ecg1nedratio4r*ecg1medratio4r*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for fourth to roof floor
ec4rps = x(58)*3.14*x(59)*x(59)/4; % provided steel for fourth to roof floor

% 3.2.3.1 DESIGN OF TIES

ectalsmax = stotalpermanentload+(0.7*stotalvariableload); %External column ties
accidental load slab max
ectalsmin = 1.25*stotalvariableload; %External column ties accidental load slab min
ectalbmax = (1.1*x(2)*ectalsmax)+(x(24)*x(25)*25); %External column ties accidental
load beam max
ectalbmin = (1.1*x(2)*ectalsmin)+(x(24)*x(25)*25); %External column ties accidental
load beam min
ectaltotal = ((ecfeb/(ectalbmax+ectalbmin))*c2r2sfbendsupport)+(ecfeb/1.25);
ectrr = (ectaltotal*1000)/x(4); % External column ties required reinforcement
ectpr = x(60)*3.14*x(61)*x(61)/4; % External column ties provided reinforcement
```

```
% 3.2.4 REINFORCEMENT REQUIREMENTS/DETAILING

ec4rsbl = ((1.5*35*x(59))+75)/1000;% External column fourth-roof floor starter bar
length
ec4rl = (3500+ec4rsbl)/1000;% External column fourth-roof floor bar length
ec4rz1links = 600/150; % Links in zone 1
ec4rz2links = (3500-600)/(1000/x(60)); % Links in zone 2
ec4rlinkl = ((2*((x(24)-35*2)+(x(25)-35*2)))+(2*10*x(61))+(3*2*x(61)))/1000; % Link
Length
ec4rlinkltotal = ec4rlinkl*(ec4rz1links+ec4rz2links);% Link Length total
ec4rsteelw = ec4rl*x(58)*(x(59)*x(59)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2)+
(ec4rsbl*x(58)*x(58)/162.2); %Total weight 4-roof floor one column
ec34steelw = ec4rl*x(56)*(x(57)*x(57)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2); %
Total weight 3-4 floor one column
ec23steelw = ec4rl*x(54)*(x(55)*x(55)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2); %
Total weight 2-3 floor one column
ec12steelw = ec4rl*x(52)*(x(53)*x(53)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2); %
Total weight 1-2 floor one column
ecg1sbl = ((1.5*35*x(51))+75)/1000;% External column fourth-roof floor starter bar
length
ecg1steelw = ec4rl*x(50)*(x(51)*x(51)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2)+
(ecg1sbl*x(50)); %Total weight g-1 floor one column
ecsteeltotalone = ec4rsteelw+ec34steelw+ec23steelw+ec12steelw+ecg1steelw; %Total
steel in kg for one column whole building
ecsteeltotal = ecsteeltotalone*(((x(17)-1)*2)+((x(20)-1)*2)); %Total steel in kg
for external columns in whole building
ecconcretetotal = (x(24)*x(25)*3.5*5/1000000)*(((x(17)-1)*2)+((x(20)-1)*2));%Total
concrete in m3 for external columns in whole building
ecconcretenet = ((ecconcretetotal*2400)-ecsteeltotal)/2400; % Net concrete in m3
for external columns in whole building
ecformwork = (2*((x(24)/1000)+(x(25)/1000)))*(3.5*5)*(((x(17)-1)*2)+((x(20)-1)
*2));% External column formwork m2
ecformworkw = ecformwork*(4/1000)*2710;% External column formwork weight kg

% 3.3 INTERNAL COLUMN
% 3.3.1 BENDING MOMENT AND AXIAL FORCE ANALYSIS

icfminload = 1.1*x(2)*stotalpermanentload; % Internal column minimum load
icmendmin = (icfminload*x(19)*x(19))/12; % Internal column end moment minimum
iclc1left = ecsrlc1; % Internal column load case 1 left
iclc1right = (croofbeammax*x(19))-iclc1left; % Internal column load case 1 right
iclc1rightis = (croofbeammax*x(19))/2; % Internal column load case 1 left internal
support
iclc1rn = iclc1rightis+iclc1right; % Internal column load case 1 axial force

iclc3mendmax = ecmendmaxr; % Load case 3 maximum moment
iclc3mendmin = ecmendminr; % Load case 3 minimum moment
iclc3mleft = (0.228*iclc3mendmin)+(-0.051*(iclc3mendmax-iclc3mendmin)); % Internal
column load case 3 moment left
iclc3mright = iclc3mendmin+(0.169*iclc3mendmin)+((iclc3mendmax-iclc3mendmin)*0.
407); % Internal column load case 3 moment right
iclc3sleft = ((croofbeammin*x(19))/2)-((iclc3mright-iclc3mleft)/7); % Internal
```

```
column load case 3 shear left
iclc3sright = (croofbeammin*x(19))-iclc3sleft; % Internal column load case 3 shear↙
right
iclc3rn = (croofbeammin*x(19)/2)+iclc3sright; % Internal column load case 1 axial↙
force roof

iclc1sleftl2r = ecsroofbeam; % Internal column load case 1 shear left load setting↙
2
iclc1srightl2r = (ecroofbeam*x(19))-iclc1sleftl2r; % Internal column load case 1↙
shear right load setting 2
iclc1l2rn = ((ecroofbeam*x(19))/2)+iclc1srightl2r;% Internal column load case 1↙
axial force roof load setting 2

iclc1sleftl2 = ecsright; % Internal column load case 1 shear left load setting 2
iclc1srightl2 = (ecfminload*x(19))-iclc1sleftl2; % Internal column load case 1↙
shear right load setting 2
iclc1l2n = (ecfminload*x(19)/2)+iclc1srightl2;% Internal column load case 1 axial↙
force roof load setting 2

icmendmax = ecmendmaxr; % Internal column end moment maximum
icmendmin = (1.1*x(2)*1.25*stotalvariableload*x(19)*x(19))/12; % Internal column↙
end moment minimum
iclc3mleftl2 = -icmendmin+(0.543*icmendmin)+((icmendmax-icmendmin)*0.102);
iclc3mrightl2 = icmendmin+(0.169*icmendmin)+((icmendmax-icmendmin)*0.407);
iclc3sleftl2 = (((1.1*x(2)*1.25*stotalvariableload)*x(19))/2)-((iclc3mrightl2-↙
iclc3mleftl2)/x(19)); % Internal column load case 1 shear left load setting 2
iclc3srightl2 = ((1.1*x(2)*1.25*stotalvariableload)*x(19))-iclc3sleftl2; % Internal↙
column load case 1 shear right load setting 2
iclc3l2n = ((1.1*x(2)*1.25*stotalvariableload)*x(19)/2)+iclc3srightl2;% Internal↙
column load case 1 axial force roof load setting 2

ic11r1n = iclc1rn;
ic11r1m = abs(-0.051*ecmendmaxr);
ic13r1n = iclc3rn;
ic13r1m = ic11r1m+((ecmend-icmendmin)*0.177);
ic21r1n = iclc1l2rn;

ic11r2n = cswperfloor;
ic13r2n = cswperfloor;

ic11r3n = ic11r2n+ic11r1n;
ic11r3m = abs(c1r2ucisleft);
ic13r3n = ic13r2n+ic13r1n;
ic13r3m = abs(c3r2ucisleft);

ic11r4n = c1r1sfbisleft+c1r1sfbisright;
ic13r4n = c3r3sfbisleft+c3r3sfbisright;
ic21r4n = iclc1l2n;
ic23r4n = iclc3l2n;

ic11r5n = ic11r4n+ic11r3n;
ic13r5n = ic13r4n+ic13r3n;
```

```
ic11r6n = cswperfloor;
ic13r6n = cswperfloor;

ic11r7n = ic11r6n+ic11r5n;
ic11r7m = abs(c1r2ucisleft);
ic13r7n = ic13r6n+ic13r5n;
ic13r7m = abs(c3r2ucisleft);

ic11r8n = c1r1sfbisleft+c1r1sfbisright;
ic13r8n = c3r3sfbisleft+c3r3sfbisright;
ic21r8n = iclc1l2n;
ic23r8n = iclc3l2n;

ic11r9n = ic11r8n+ic11r7n;
ic13r9n = ic13r8n+ic13r7n;
ic21r9n = ic21r8n+ic21r4n;
ic23r9n = ic23r8n+ic23r4n;

ic11r10n = cswperfloor;
ic13r10n = cswperfloor;

ic11r11n = ic11r10n+ic11r9n;
ic11r11m = abs(c1r2ucisleft);
ic13r11n = ic13r10n+ic13r9n;
ic13r11m = abs(c3r2ucisleft);

ic11r12n = c1r1sfbisleft+c1r1sfbisright;
ic13r12n = c3r3sfbisleft+c3r3sfbisright;
ic21r12n = iclc1l2n;
ic23r12n = iclc3l2n;

ic11r13n = ic11r12n+ic11r11n;
ic13r13n = ic13r12n+ic13r11n;
ic21r13n = ic21r12n+ic21r9n;
ic23r13n = ic23r12n+ic23r9n;

ic11r14n = cswperfloor;
ic13r14n = cswperfloor;

ic11r15n = ic11r14n+ic11r13n;
ic11r15m = abs(c1r2ucisleft);
ic13r15n = ic13r14n+ic13r13n;
ic13r15m = abs(c3r2ucisleft);

ic11r16n = c1r1sfbisleft+c1r1sfbisright;
ic13r16n = c3r3sfbisleft+c3r3sfbisright;
ic21r16n = iclc1l2n;
ic23r16n = iclc3l2n;

ic11r17n = ic11r16n+ic11r15n;
ic13r17n = ic13r16n+ic13r15n;
ic21r17n = ic21r16n+ic21r13n;
ic23r17n = ic23r16n+ic23r13n;
```

```
ic11r18n = cswperfloor;
ic13r18n = cswperfloor;

ic11r19n = ic11r18n+ic11r17n;
ic11r19m = abs(c1r2ucisleft);
ic13r19n = ic13r18n+ic13r17n;
ic13r19m = abs(c3r2ucisleft);

ic11r20n = c1r1sfbisleft+c1r1sfbisright;
ic13r20n = c3r3sfbisleft+c3r3sfbisright;
ic21r20n = iclc1l2n;
ic23r20n = iclc3l2n;

ic11r21n = ic11r20n+ic11r19n;
ic13r21n = ic13r20n+ic13r19n;
ic21r21n = ic21r20n+ic21r17n;
ic23r21n = ic23r20n+ic23r17n;

ic11r22n = cswperfloor;
ic13r22n = cswperfloor;

ic11r23n = ic11r22n+ic11r21n;
ic13r23n = ic13r22n+ic13r21n;

% 3.3.2 EFFECTIVE LENGTH AND SLENDERNESS

icmtoplc3 = ic13r19m; % Moment at top from basement to ground floor load case 3
icmbotlc3 = -0.5*icmtoplc3; % Moment at bottom from basement to ground floor load
case 3
icnedmax = ic11r19n+ic13r20n-(0.4*(ic21r17n+ic23r16n)); % Ned maximum
icnedmin = ic13r20n+((ic11r19n-(ic21r17n+ic21r1n))/1.25); % Ned minimum

iclo = 0.75*(3.5-x(23)/1000);% Effective legth
icslenderness = ecslenderness;% Slenderness ratio
icm1 = (icnedmax*iclo)/400; % First order moment
icm01i = icmbotlc3+icm1; % First order moment with imperfections
icm02i = icmtoplc3+icm1; % First order moment with imperfections
icc = 1.7-(icm01i/icm02i);
icn = (icnedmax)/(x(24)*x(25)*0.85*x(3)/1.5);
icslendernesslimit = (20*0.7*1.1*icc)/(icn^(1/2)); % Slenderness ratio limit

% 3.3.3 DESIGN OF CROSS-SECTION

icbgnedmax = ic11r23n+ic13r20n-(0.4*(ic21r21n+ic23r16n)); % Ned maximum basement-
ground floor
icbgmedmax = icmtoplc3+((icbgnedmax*iclo)/400);
icbgnedratio = (icbgnedmax*1000)/(x(24)*x(25)*x(3)); % basement-ground floor axial
force ratio
icbgmedratio = (icbgmedmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % basement-ground
floor moment ratio
icbgrs = ((icbgnedratio*icbgmedratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel
for basement-ground floor
```

```
icbgps = x(62)*3.14*x(63)*x(63)/4; % provided steel for basement-ground floor

icg1nedmax = ic11r19n+ic13r20n-(0.4*(ic21r17n+ic23r16n)); % Ned maximum ground-
first floor
icg1medmax = icmtoplc3+((icg1nedmax*iclo)/400);
icg1nedratio = (icg1nedmax*1000)/(x(24)*x(25)*x(3)); % ground-first floor axial
force ratio
icg1medratio = (icg1medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % ground-first floor
moment ratio
icg1rs = ((icg1nedratio*icg1medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel
for ground-first floor
icg1ps = x(64)*3.14*x(65)*x(65)/4; % provided steel for ground-first floor

ic12nedmax = ic11r15n+ic13r20n-(0.4*(ic21r13n+ic23r16n)); % Ned maximum first-
second floor
ic12medmax = icmtoplc3+((ic12nedmax*iclo)/400);
ic12nedratio = (ic12nedmax*1000)/(x(24)*x(25)*x(3)); % first-second floor axial
force ratio
ic12medratio = (ic12medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % first-second floor
moment ratio
ic12rs = ((ic12nedratio*ic12medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel
for first-second floor
ic12ps = x(66)*3.14*x(67)*x(67)/4; % provided steel for first-second floor

ic23nedmax = ic11r11n+ic13r20n-(0.4*(ic21r9n+ic23r16n)); % Ned maximum second-third
floor
ic23medmax = icmtoplc3+((ic23nedmax*iclo)/400);
ic23nedratio = (ic23nedmax*1000)/(x(24)*x(25)*x(3)); % second-third floor axial
force ratio
ic23medratio = (ic23medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % second-third floor
moment ratio
ic23rs = ((ic23nedratio*ic23medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel
for second-third floor
ic23ps = x(68)*3.14*x(69)*x(69)/4; % provided steel for second-third floor

ic34nedmax = ic11r7n+ic13r20n-(0.4*(ic21r4n+ic23r16n)); % Ned maximum third-fourth
floor
ic34medmax = icmtoplc3+((ic34nedmax*iclo)/400);
ic34nedratio = (ic34nedmax*1000)/(x(24)*x(25)*x(3)); % third-fourth floor axial
force ratio
ic34medratio = (ic34medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % third-fourth floor
moment ratio
ic34rs = ((ic34nedratio*ic34medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel
for third-fourth floor
ic34ps = x(70)*3.14*x(71)*x(71)/4; % provided steel for third-fourth floor

ic4rnedmax = ic11r3n+ic13r20n-(0.4*(ic21r1n)); % Ned maximum fourth-roof floor
ic4rmedmax = icmtoplc3+((ic4rnedmax*iclo)/400);
ic4rnedratio = (ic4rnedmax*1000)/(x(24)*x(25)*x(3)); % fourth-roof floor axial
force ratio
ic4rmedratio = (ic4rmedmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % fourth-roof floor
moment ratio
ic4rrs = ((ic4rnedratio*ic4rmedratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel
```

```
for fourth-roof floor
ic4rps = x(72)*3.14*x(73)*x(73)/4; % provided steel for fourth-roof floor

% 3.3.3.1 DESIGN OF TIES - INTERNAL COLUMN

ictaltotal = (ecfeb/(ectalbmax+ectalbmin))*ic11r4n; % Internal column ties↙
accidental load beam min
ictrr = (ictaltotal*1000)/x(4); % Internal column ties required reinforcement
ictpr = x(74)*3.14*x(75)*x(75)/4; % Internal column ties provided reinforcement

% 3.3.4 REINFORCEMENT REQUIREMENTS/DETAILING

ic4rsbl = ((1.5*35*x(73))+75)/1000;% Internal column fourth-roof floor starter bar↙
length
ic4rl = (3500+ic4rsbl)/1000;% Internal column fourth-roof floor bar length
ic4rz1links = 600/150; % Links in zone 1
ic4rz2links = (3500-600)/(1000/x(74)); % Links in zone 2
ic4rlinkl = ((2*((x(24)-35*2)+(x(25)-35*2)))+(2*10*x(75))+(3*2*x(75)))/1000; % Link↙
Length
ic4rlinkltotal = ic4rlinkl*(ic4rz1links+ic4rz2links);% Link Length total
ic4rsteelw = ic4rl*x(72)*(x(73)*x(73)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2)+ ↙
(ic4rsbl*x(72)*x(72)/162.2); %Total weight 4-roof floor one column
ic34steelw = ic4rl*x(70)*(x(71)*x(71)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); %↙
Total weight 3-4 floor one column
ic23steelw = ic4rl*x(68)*(x(69)*x(69)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); %↙
Total weight 2-3 floor one column
ic12steelw = ic4rl*x(66)*(x(67)*x(67)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); %↙
Total weight 1-2 floor one column
icg1steelw = ic4rl*x(64)*(x(65)*x(65)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); %↙
Total weight g-1 floor one column
icbgsbl = ((1.5*35*x(63))+75)/1000;% Internal column basement-ground floor starter↙
bar length
icbgsteelw = ic4rl*x(62)*(x(63)*x(63)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2)+ ↙
(icbgsbl*x(62)); %Total weight basement-ground floor one column
icsteeltotalone =↙
ic4rsteelw+ic34steelw+ic23steelw+ic12steelw+icg1steelw+icbgsteelw; %Total steel in↙
kg for one column whole building
icsteeltotal = icsteeltotalone*((x(17)-1)*(x(20)-1)); %Total steel in kg for↙
internal columns in whole building

icconcretetotal = (x(24)*x(25)*3.5*6/1000000)*((x(17)-1)*(x(20)-1));%Total concrete↙
in m3 for internal columns in whole building
icconcretenet = ((icconcretetotal*2400)-icsteeltotal)/2400; % Net concrete in m3↙
for Internal columns in whole building

icformwork = (2*((x(24)/1000)+(x(25)/1000)))*(3.5*6)*((x(17)-1)*(x(20)-1)); %↙
Internal column formwork m2
icformworkw = icformwork*(4/1000)*2710; % Internal column formwork weight kg

% 3.4 CORNER COLUMN
% 3.4.1 BENDING MOMENT AND AXIAL FORCE ANALYSIS

ccslinea = 0.4*x(2)*sdesignload; % Load due to slab on line A
```

```
ccblinea = ebwb/x(2); % Load due to beam and waling on line A
ccsline1 = ebsdload+ebslload; % Load due to slab on line 1
ccbline1 = ecfeb; % Load due to beam and waling on line 1
cctotallinea = ccslinea+ccblinea; % Total load on line A
cctotalline1 = ccsline1+ccbline1; % Total load on line 1
ccmz = (cctotallinea/mbtoalmaxload)*c2r2ucendsupport; % Column moment of frame on
line A

ccbendk = (0.5*mbendk)/(x(2)*1000); % Stiffness of end beam
ccuck = mbicolumn/storeyheight*1000; %STIFFNESS OF UPPER COLUMN
cclck = mbicolumn/storeyheight*1000; %STIFFNESS OF LOWER COLUMN

ccbfem = ((0.104*ccsline1)+(0.083*ccbline1))/x(2); % Beam fix end moment
ccmy = (ccuck/((2*cclck)+ccbendk))*ccbfem; % Column moment

ccslinear = 0.4*x(2)*ebtarea; % Load due to slab on roof level at line A
ccblinea1r = mbtotalminload/1.25; % Load due to beam and waling on roof level at
line 1
ccblinear = ccblinea1r/x(2); % Load due to beam and waling on roof level at line A
cctotallinear = ccslinear+ccblinear; % Total load on line A
cctotalline1r = ccblinea1r; % Total load on line 1

ccned = (((cctotallinear/mbtoalmaxload)*abs(c1r1sfbendsupport))+(0.525
*cctotalline1r)+(abs(cswperfloor)/1.25))/1000;
ccmi = (ccned*iclo)/(400); % First order moments from imperfections
ccm0z = ccmi+ccmz; % First order moments from imperfections
ccm0y = ccmy; % First order moments from imperfections

% 3.4.2 DESIGN OF CROSS-SECTION

ccnedratio = (ccned*1000)/(x(24)*x(25)*x(3)); % fourth-roof floor axial force ratio
ccmedratio = (ccm0z*1000000)/(x(24)*x(25)*x(25)*x(3)); % fourth-roof floor moment
ratio
ccrs = ((ccnedratio*ccmedratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel for
fourth-roof floor
ccps = x(76)*3.14*x(77)*x(77)/4; % provided steel for fourth-roof floor


% 3.4.3 REINFORCEMENT REQUIREMENTS/DETAILING

cc4rsbl = ((1.5*35*x(77))+75)/1000; % Corner column fourth-roof floor starter bar
length
cc4rl = (3500+cc4rsbl)/1000;% Corner column fourth-roof floor bar length
cc4rz1links = 600/150; % Links in zone 1
cc4rz2links = (3500-600)/(1000/x(74)); % Links in zone 2
cc4rlinkl = ((2*((x(24)-35*2)+(x(25)-35*2)))+(2*10*x(75))+(3*2*x(75)))/1000; % Link
Length
cc4rlinkltotal = cc4rlinkl*(cc4rz1links+cc4rz2links);% Link Length total
cctotalsteelonew = 5*(cc4rl*x(76)*(x(77)*x(77)/162.2))+5*(cc4rlinkltotal*x(75)*x
(75)/162.2)+2*(cc4rsbl*x(77)*x(77)/162.2); %Total weight 4-roof floor o
cctotalsteelbuildingw = 4*cctotalsteelonew; % Total building steel in corner
columns
```

```
ccconcretetotal = (x(24)*x(25)*3.5*5/1000000)*4;%Total concrete in m3 for internal
columns in whole building
ccconcretenet = ((ccconcretetotal*2400)-cctotalsteelbuildingw)/2400; % Net concrete
in m3 for Internal columns in whole building

ccformwork = (2*((x(24)/1000)+(x(25)/1000)))*(3.5*5)*4; % Internal column formwork
m2
ccformworkw = ccformwork*(4/1000)*2710; % Internal column formwork weight kg

% 4 QUANTITY CALCULATION

bsteel= slabtotalbuildingsteel;  % Building slab steel in kg
bbsteel = mbtotalsteel+ebtotalsteel; % Building beams steel in kg
bcsteel = ecsteeltotal+icsteeltotal+cctotalsteelbuildingw; % Building column steel
in kg
btotalsteel = bsteel+bbsteel+bcsteel; %Building total steel in kg

bsconcrete= sncbuilding; % Building slab concrete in m3
bbconcrete = mbcbuildingnetvolume+ebcbuildingnetvolume;  % Building beams concrete
in m3
bcconcrete = ecconcretenet+icconcretenet+ccconcretenet; % Building column concrete
in m3
btotalconcrete = bsconcrete+bbconcrete+bcconcrete; %Building total concrete in m3

bsfa = sfa; % Building slab formwork in m2
bbfa = mbfarea+ebfarea; % Building beams formwork in m2
bcfa = ecformwork+icformwork+ccformwork; % Building column formwork in m2
btotalfa = bsfa+bbfa+bcfa; %Building total formwork in m2


bsfw = sfweight; % Building slab formwork in kg
bbfw = mbfweight+ebfweight+ebfweight;  % Building beams formwork in kg
bcfw = ecformworkw+icformworkw+ccformworkw; % Building column formwork in kg
btotalfw = bsfw+bbfw+bcfw; %Building total formwork in kg


% OUTPUT FUNCTION
y(1) = (btotalconcrete*110)+(btotalsteel*0.8)+(btotalfa*6/300); % Total cost
function
%y(1) = (btotalconcrete*338)+(btotalsteel*0.87)+(btotalfw*0.79/300); % Total carbon
function

end
```

```
function x = framemapvariables(x)

allX1 = [150,175,200,225,250,275,300,325,350,375,400];  %DEPTH OF SLAB
allX2 = [3,3.5,4,4.5,5,5.5,6,6.5,7]; % CLEAR SPANS IN X-DIRECTION
allX3 = [20,25,28,30,35,40,45]; %CONCRETE STRENGTH
allX4 = [500,550];  %STEEL STRENGTH
allX5 = [12,14,16,20,25,28,32]; % DIAMETER for smpespan
allX6 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
SMPESPAN
allX7 = [12,14,16,20,25,28,32]; % DIAMETER for smfesup
allX8 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
smfesup
allX9 = [12,14,16,20,25,28,32]; % DIAMETER for smfespan
allX10 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
smfespan
allX11 = [12,14,16,20,25,28,32]; % DIAMETER for smfisupport
allX12 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
smfisupport
allX13 = [12,14,16,20,25,28,32]; % DIAMETER for smaispan
allX14 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
smaispan
allX15 = [12,14,16,20,25,28,32]; % DIAMETER for smoisupport
allX16 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
smoisupport
allX17 = [5,6,7,8,9,10,11]; % NUMBER OF SPANS IN X-DIRECTION
allX18 = [200,225,250,275,300,325,350,375,400,425,450,475,500,525,550,575,600];  %↙
BREADTH OF MAIN BEAM
allX19 = [3,3.5,4,4.5,5,5.5,6,6.5,7]; % CLEAR SPANS IN Y-DIRECTION
allX20 = [3,4,5,6,7]; % NUMBER OF SPANS IN Y-DIRECTION
allX21 = [8,10,12]; % SECONDARY BAR PROVIDED DIAMETER
allX22 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF↙
SECONDARY REINFORCEMENT BARS
allX23 = [250,275,300,325,350,375,400,425,450,475,500,525,550,575,600];  %DEPTH OF↙
MAIN BEAM
allX24 =↙
[125,150,175,200,225,250,275,300,325,350,375,400,425,450,475,500,525,550,575,600];↙
%BREADTH OF COLUMN
allX25 = [200,225,250,275,300,325,350,375,400,425,450,475,500,525,550,575,600];  %↙
DEPTH OF COLUMN
allX26 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
main beam interior support
allX27 = [12,14,16,20,25,28,32]; % DIAMETER for main beam interior support
allX28 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
main beam end support
allX29 = [12,14,16,20,25,28,32]; % DIAMETER for main beam end support
allX30 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
main beam interior span
allX31 = [12,14,16,20,25,28,32]; % DIAMETER for main beam interior span
allX32 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for↙
main beam exterior span
allX33 = [12,14,16,20,25,28,32]; % DIAMETER for main beam exterior span
allX34 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS↙
(shear design at end support)
```

```
allX35 = [8,10,12]; % LINK DIAMETER for main beam end support (shear design/links dia)
allX36 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS (shear design at interior left support)
allX37 = [8,10,12]; % LINK DIAMETER for main beam interior left support (shear design/links dia)
allX38 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS (shear design at interior right support)
allX39 = [8,10,12]; % LINK DIAMETER for main beam interior right support (shear design/links dia)
allX40 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for edge beam end span
allX41 = [12,14,16,20,25,28,32]; % DIAMETER for edge beam end span
allX42 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for edge beam first interior support
allX43 = [12,14,16,20,25,28,32]; % DIAMETER for edge beam first interior support
allX44 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for edge beam interior span
allX45 = [12,14,16,20,25,28,32]; % DIAMETER for edge beam interior span
allX46 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for edge beam other interior support
allX47 = [12,14,16,20,25,28,32]; % DIAMETER for edge beam other interior support
allX48 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for edge beam shear force
allX49 = [12,14,16,20,25,28,32]; % DIAMETER for edge beam shear force
allX50 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for external column g-1
allX51 = [12,14,16,20,25,28,32]; % DIAMETER for external column g-1
allX52 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for external column 1-2 floor
allX53 = [12,14,16,20,25,28,32]; % DIAMETER for external column 1-2 floor
allX54 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for external column 2-3 floor
allX55 = [12,14,16,20,25,28,32]; % DIAMETER for external column 2-3 floor
allX56 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for external column 3-4 floor
allX57 = [12,14,16,20,25,28,32]; % DIAMETER for external column 3-4 floor
allX58 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for external column 4-roof floor
allX59 = [12,14,16,20,25,28,32]; % DIAMETER for external column 4-roof floor
allX60 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for external column ties
allX61 = [8,10,12]; % DIAMETER for external column ties
allX62 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for internal column basement-ground floor
allX63 = [12,14,16,20,25,28,32]; % DIAMETER for internal column basement-ground floor
allX64 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for internal column ground-1 floor
allX65 = [12,14,16,20,25,28,32]; % DIAMETER for internal column ground-1 floor
allX66 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for internal column 1-2 floor
allX67 = [12,14,16,20,25,28,32]; % DIAMETER for internal column 1-2 floor
```

```
allX68 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for
internal column 2-3 floor
allX69 = [12,14,16,20,25,28,32]; % DIAMETER for internal column 2-3 floor
allX70 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for
internal column 3-4 floor
allX71 = [12,14,16,20,25,28,32]; % DIAMETER for internal column 3-4 floor
allX72 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for
internal column 4-r floor
allX73 = [12,14,16,20,25,28,32]; % DIAMETER for internal column 4-r floor
allX74 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for
internal column ties
allX75 = [8,10,12]; % DIAMETER for internal column ties
allX76 = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; % NUMBER OF BARS for
corner column
allX77 = [12,14,16,20,25,28,32]; % DIAMETER for corner column



x(1) = allX1(x(1));
x(2) = allX2(x(2));
x(3) = allX3(x(3));
x(4) = allX4(x(4));
x(5) = allX5(x(5));
x(6) = allX6(x(6));
x(7) = allX7(x(7));
x(8) = allX8(x(8));
x(9) = allX9(x(9));
x(10) = allX10(x(10));
x(11) = allX11(x(11));
x(12) = allX12(x(12));
x(13) = allX13(x(13));
x(14) = allX14(x(14));
x(15) = allX15(x(15));
x(16) = allX16(x(16));
x(17) = allX17(x(17));
x(18) = allX18(x(18));
x(19) = allX19(x(19));
x(20) = allX20(x(20));
x(21) = allX21(x(21));
x(22) = allX22(x(22));
x(23) = allX23(x(23));
x(24) = allX24(x(24));
x(25) = allX25(x(25));
x(26) = allX26(x(26));
x(27) = allX27(x(27));
x(28) = allX28(x(28));
x(29) = allX29(x(29));
x(30) = allX30(x(30));
x(31) = allX31(x(31));
x(32) = allX32(x(32));
x(33) = allX33(x(33));
x(34) = allX34(x(34));
x(35) = allX35(x(35));
```

```
x(36) = allX36(x(36));
x(37) = allX37(x(37));
x(38) = allX38(x(38));
x(39) = allX39(x(39));
x(40) = allX40(x(40));
x(41) = allX41(x(41));
x(42) = allX42(x(42));
x(43) = allX43(x(43));
x(44) = allX44(x(44));
x(45) = allX45(x(45));
x(46) = allX46(x(46));
x(47) = allX47(x(47));
x(48) = allX48(x(48));
x(49) = allX49(x(49));
x(50) = allX50(x(50));
x(51) = allX51(x(51));
x(52) = allX52(x(52));
x(53) = allX53(x(53));
x(54) = allX54(x(54));
x(55) = allX55(x(55));
x(56) = allX56(x(56));
x(57) = allX57(x(57));
x(58) = allX58(x(58));
x(59) = allX59(x(59));
x(60) = allX60(x(60));
x(61) = allX61(x(61));
x(62) = allX62(x(62));
x(63) = allX63(x(63));
x(64) = allX64(x(64));
x(65) = allX65(x(65));
x(66) = allX66(x(66));
x(67) = allX67(x(67));
x(68) = allX68(x(68));
x(69) = allX69(x(69));
x(70) = allX70(x(70));
x(71) = allX71(x(71));
x(72) = allX72(x(72));
x(73) = allX73(x(73));
x(74) = allX74(x(74));
x(75) = allX75(x(75));
x(76) = allX76(x(76));
x(77) = allX77(x(77));



end
```

```
function [Cineq,Ceq] = framenonlcon(x)

x = framemapvariables(x);

% 1. SLAB DESIGN
% 1.1 LOADING CALCULATIONS
bxdirection = x(2)*(17);
bydirection = x(19)*(20);
bbayarea = bxdirection*bydirection; % in m2
snominalcover = 25; % in mm
sa = snominalcover+(12/2); % in mm
simposedload = 2.5; % in KN/m2 for office building
spartitionwall = 1.5; % in KN/m2 for office building
sselfweight = x(1)*25/1000; % in KN/m2
sfinisheweight = 1.25; % in KN/m2
stotalpermanentload = sselfweight+sfinisheweight; % in KN/m2
stotalvariableload = simposedload+spartitionwall; % in KN/m2
sdesignload = (1.35*stotalpermanentload)+(1.65*stotalvariableload);% in KN/m2
sf = sdesignload*x(2); % in KN/m

% 1.2 BENDING MOMENT AND SHEAR FORCE CALCULATIONS

smpespan = 0.086*sf*x(2); % slab moment pinned end span
smfesup = -0.063*sf*x(2); % slab moment fixed end support
smfespan = 0.063*sf*x(2); % slab moment fixed end span
smfisupport = -0.086*sf*x(2); % slab moment first interior support
smaispan = 0.063*sf*x(2); % slab moment all interior span
smoisupport = -0.063*sf*x(2); % slab moment other interior support

sspesupport = 0.4*sf; % slab shear pinned end support
ssfesupport = 0.48*sf; % slab shear fixed end support
ssfisupport = 0.6*sf; % slab shear first interior support
ssoisupport = 0.5*sf; % slab shear other interior support

% 1.3 FLEXURE DESIGN/CHECK

seffectivedepth = x(1)-snominalcover-(12/2); % in mm
sbreadth = 1000; % in mm

smpespank = (abs(smpespan)*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x↙
(3));
smpespanleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smpespank))^(1/2)); % in mm
smpespanmaxleverarm = 0.95*seffectivedepth;
 if smpespanleverarm <= smpespanmaxleverarm
    smpespanz = smpespanleverarm;
 else
    smpespanz = smpespanmaxleverarm;
 end
smpespanrequiredsteel = (abs(smpespan)*1000000)/(0.87*x(4)*smpespanz);
smpespanminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smpespanprovidedsteel = x(6)*3.14*x(5)*x(5)/4;
smpespanmaximumsteel = 0.04*(sbreadth*x(1)-smpespanprovidedsteel);
```

```
smfesupk = (abs(smfesup)*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x(3));
smfesupleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smfesupk))^(1/2)); % in mm
smfesupmaxleverarm = 0.95*seffectivedepth;
 if smfesupleverarm <= smfesupmaxleverarm
    smfesupz = smfesupleverarm;
 else
    smfesupz = smfesupmaxleverarm;
 end
smfesuprequiredsteel = (smfesup*1000000)/(0.87*x(4)*smfesupz);
smfesupminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smfesupprovidedsteel = x(8)*3.14*x(7)*x(7)/4;
smfesupmaximumsteel = 0.04*(sbreadth*x(1)-smfesupprovidedsteel);

smfespank = (smfespan*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x(3));
smfespanleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smfespank))^(1/2)); % in mm
smfespanmaxleverarm = 0.95*seffectivedepth;
 if smfespanleverarm <= smfespanmaxleverarm
    smfespanz = smfespanleverarm;
 else
    smfespanz = smfespanmaxleverarm;
 end
smfespanrequiredsteel = (smfespan*1000000)/(0.87*x(4)*smfespanz);
smfespanminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smfespanprovidedsteel = x(10)*3.14*x(9)*x(9)/4;
smfespanmaximumsteel = 0.04*(sbreadth*x(1)-smfespanprovidedsteel);

smfisupportk = (abs(smfisupport)*1000000)/↙
(sbreadth*seffectivedepth*seffectivedepth*x(3));
smfisupportleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smfisupportk))^(1/2)); % in↙
mm
smfisupportmaxleverarm = 0.95*seffectivedepth;
 if smfisupportleverarm <= smfisupportmaxleverarm
    smfisupportz = smfisupportleverarm;
 else
    smfisupportz = smfisupportmaxleverarm;
 end
smfisupportrequiredsteel = (abs(smfisupport)*1000000)/(0.87*x(4)*smfisupportz);
smfisupportminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smfisupportprovidedsteel = x(12)*3.14*x(11)*x(11)/4;
smfisupportmaximumsteel = 0.04*(sbreadth*x(1)-smfisupportprovidedsteel);

smaispank = (smaispan*1000000)/(sbreadth*seffectivedepth*seffectivedepth*x(3));
smaispanleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smaispank))^(1/2)); % in mm
smaispanmaxleverarm = 0.95*seffectivedepth;
 if smaispanleverarm <= smaispanmaxleverarm
    smaispanz = smaispanleverarm;
 else
    smaispanz = smaispanmaxleverarm;
 end
smaispanrequiredsteel = (smaispan*1000000)/(0.87*x(4)*smaispanz);
smaispanminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smaispanprovidedsteel = x(14)*3.14*x(13)*x(13)/4;
smaispanmaximumsteel = 0.04*(sbreadth*x(1)-smaispanprovidedsteel);
```

```
smoisupportk = (abs(smoisupport)*1000000)/
(sbreadth*seffectivedepth*seffectivedepth*x(3));
smoisupportleverarm = 0.5*seffectivedepth*(1+(1-(3.53*smoisupportk))^(1/2)); % in
mm
smoisupportmaxleverarm = 0.95*seffectivedepth;
 if smoisupportleverarm <= smoisupportmaxleverarm
    smoisupportz = smoisupportleverarm;
 else
    smoisupportz = smoisupportmaxleverarm;
 end
smoisupportrequiredsteel = (abs(smoisupport)*1000000)/(0.87*x(4)*smoisupportz);
smoisupportminimumsteel = (0.26*0.3*x(3)^(2/3)*sbreadth*seffectivedepth)/x(4);
smoisupportprovidedsteel = x(16)*3.14*x(15)*x(15)/4;
smoisupportmaximumsteel = 0.04*(sbreadth*x(1)-smoisupportprovidedsteel);

sbrsrequiredarea = 0.2*smpespanprovidedsteel;
sbrsprovidedarea = x(22)*3.14*x(21)*x(21)/4;

% 1.4 SHEAR DESIGN/CHECK

srow = smpespanprovidedsteel/(sbreadth*seffectivedepth);
sk1 = 1+((200/seffectivedepth)^(1/2));
 if sk1>2
    sk1=2;
 else
    sk1= 1+ ((200/seffectivedepth)^(1/2));
 end
sresistanceshear = (0.12*sk1*((100*srow*x(3))^(1/3))/1000)
*sbreadth*seffectivedepth;
sminshear = 0.035*((sk1)^(3/2))*(x(3)^(1/2));
sminresistanceshear = (sminshear+(sk1*(ssfisupport*1000)/sbreadth*seffectivedepth))
*sbreadth*seffectivedepth;


% 1.5 DEFLECTION DESIGN/CHECK

sk2 = 1.3 ; % for one way solid slab
srowzero = (x(3)^(1/2))/1000;
srowone = smpespanprovidedsteel/(sbreadth*seffectivedepth);
srowtwo = 0;
 if srowone<=srowzero
    sspantodepthratio = sk2*(11+(1.5*((x(3)^(1/2))*(srowzero/srowone)))+(3.2*(x(3)^
(1/2))*(((srowzero/srowone)-1)^(3/2))));
 else
    sspantodepthratio = sk2*(11+(1.5*((x(3)^(1/2))*(srowzero/(srowone-srowtwo))))+
((1/12)*(x(3)^(1/2))*((srowtwo/srowzero)^(1/2))));
 end

sf1 = (500*smpespanprovidedsteel)/(x(4)*smpespanrequiredsteel);
sbasicspantodepthratio = sspantodepthratio*sf1; % in mm
sactualspantodepthratio = x(2)*1000/seffectivedepth; % in mm
```

```
% 1.6 CRACKING DESIGN/CHECK

skc = 0.4;
smincrackingarea = (skc*0.3*(x(3)^(3/2))*sbreadth*x(1))/x(4);

% 1.7 QUANTITY CALCULATION
% 1.7.1 STEEL CALCULATION

sbrlx = (x(2)*x(17))+(x(18)/1000)-(2*snominalcover/1000); % SLAB BOTTOM ↙
REINFORCEMENT IN X DIRECTION IN M
sbrly = (x(19)*x(20))+(x(18)/1000)-(2*snominalcover/1000)-0.040; % SLAB BOTTOM ↙
REINFORCEMENT IN Y DIRECTION IN M
sbrnmespan = sbrly*1000/(1000/x(6)); % SLAB BOTTOM REINFORCEMENT NUMBER OF main ↙
BARS IN END SPAN
sbrnmispan = sbrly*1000/(1000/x(14)); % SLAB BOTTOM REINFORCEMENT NUMBER OF main ↙
BARS IN INTERIOR SPAN
sbrnsispan = sbrlx*1000/(1000/x(14)); % SLAB BOTTOM REINFORCEMENT NUMBER OF ↙
secondary BARS
sbrmweight = ((sbrnmespan*(x(2)-snominalcover/1000-0.040)*2*x(5)*x(5))/162.2)+ ↙
(sbrnmispan*(x(2)-snominalcover/1000-0.040)*(x(17)-2)*x(14)*x(14)/162.2); % SLAB↙
BOTTOM REINFORCEMENT MAIN BAR WEIGHT
sbrsweight = sbrnsispan*x(14)*x(2)*x(21)*x(21)/162.2; % WEIGHT IN KG
sbrtweight = sbrmweight+sbrsweight; % SLAB BOTTOM REINFORCEMENT TOTAL STEEL WEIGHT↙
IN KG

strelx = 0.2*x(2)*2; % SLAB TOP REINFORCEMENT AT END SUPPORT IN X DIRECTION IN M
strilx = 0.3*x(2)*(x(17)-1); % SLAB TOP REINFORCEMENT AT INTERIOR SUPPORT IN X ↙
DIRECTION IN M
strly = (x(19)*x(20))+(x(18)/1000)-(2*snominalcover/1000)-0.040; % SLAB TOP↙
REINFORCEMENT IN Y DIRECTION IN M
strmweight = ((strly*strelx)*(x(7)*x(7)/162.2))+((strly*strilx)*(x(15)*x(15)/162. ↙
2)); % SLAB TOP REINFORCEMENT MAIN BAR WEIGHT IN KG
strsweight = (strelx*1000/(1000/x(14))*x(21)*x(21)/162.2)+(strilx*1000/(1000/x(14)) ↙
*x(21)*x(21)/162.2); % WEIGHT IN KG
strtweight = strmweight+strsweight; % TOTAL TOP REINFORCEMENT WEIGHT

slabtotalsteel = sbrtweight+strtweight; % TOTAL SLAB STEEL WEIGHT IN KG
slabtotalbuildingsteel = slabtotalsteel*6; % TOTAL BUILDING SLAB STEEL WEIGHT IN KG


% 1.7.2 CONCRETE CALCULATION

stc = ((x(2)*x(17))*(x(19)*x(20)))*x(1)/1000; % SLAB TOTAL CONCRETE in M3
stcweight = stc*2400; % SLAB TOTAL CONCRETE in KG
sncweight = stcweight-slabtotalsteel; % NET AMOUNT OF CONCRETE IN KG
snc = sncweight/2400; % NET AMOUNT OF CONCRETE IN M3
sncbuilding = snc*6; % NET AMOUNT OF BUILDING SLAB CONCRETE IN M3

% 1.7.3 FORMWORK CALCULATION

sfa = (((x(2)-(x(18)/1000))*x(17))*((x(19)-(x(18)/1000)*x(20))))*6;  % AREA OF↙
FORMWORK FOR BUILDING
```

```
sfweight = sfa*4/1000*2710; %WEIGHT OF FORMWORK FOR BUILDING

% 2. BEAM DESIGN
% 2.1 MAIN BEAM DESIGN
% 2.1.1 FIRE RESISTANCE/COVER DETERMINATION
 bnominalcover = 25; % IN MM
 baxisdistance = bnominalcover+8+(32/2); % AXIS DISTANCE FOR 1.5 HR OF FIRE

% 2.1.2 LOADING CALULATIONS
mbmaxdesignload = sdesignload; %MAIN BEAM MAXIMUM DESIGN LOAD
mbmindesignload = 1.25*stotalpermanentload; %MAIN BEAM MINIMUM DESIGN LOAD
mbmaxslab = 1.1*mbmaxdesignload*x(2); %MAIN BEAM MAXIMUM DESIGN LOAD DUE TO SLAB
mbminslab = 1.1*x(2)*mbmindesignload; %MAIN BEAM MINIMUM DESIGN LOAD DUE TO SLAB
mbmax = 1.25*x(23)*x(18)*25/1000000; %MAIN BEAM MAXIMUM DESIGN LOAD DUE TO MAIN↙
BEAMS
mbmin = mbmax;
mbtoalmaxload = mbmaxslab+mbmax; %IN KN/M
mbtotalminload = mbminslab+mbmin; % IN KN/M

% 2.1.3 BENDING MOMENT AND SHEAR FORCE ANALYSIS/ SUB FRAME ANALYSIS

mbibeam = x(18)*(x(23)^3)/12; %MOMENT OF INERTIA OF BEAM
mbicolumn = x(24)*(x(25)^3)/12; %MOMENT OF INERTIA OF COLUMN
mbendk = mbibeam/x(19)*1000; %STIFFNESS OF END BEAM
mbintk = mbendk; %STIFFNESS OF INTERIOR BEAM
storeyheight = 3.5;
mbuppercolumnk = mbicolumn/storeyheight*1000; %STIFFNESS OF UPPER COLUMN
mblowercolumnk = mbicolumn/storeyheight*1000; %STIFFNESS OF LOWER COLUMN
mbdfendjointb = mbendk/(mbendk+(2*mbuppercolumnk)); % MAIN BEAM DISTRIBUTION FACTOR↙
AT END JOINT FOR BEAM
mbdfendjointc = (1-mbdfendjointb)/2; % COLUMN DISTRIBUTION FACTOR AT END JOINT FOR↙
COLUMN
mbdfinteriorjointendb = mbendk/(mbendk+(0.5*mbintk)+(2*mblowercolumnk)); % MAIN↙
BEAM DISTRIBUTION FACTOR AT INTERIOR JOINT FOR END BEAM
mbdfinteriorjointintb = (0.5*mbendk)/(mbendk+(0.5*mbintk)+(2*mblowercolumnk)); %↙
MAIN BEAM DISTRIBUTION FACTOR AT INTERIOR JOINT FOR INTERIOR BEAM
mbdfinteriorjointc = (mbuppercolumnk)/(mbendk+(0.5*mbintk)+(2*mblowercolumnk)); %↙
column DISTRIBUTION FACTOR AT END JOINT FOR COLUMN
mbendmomentmax = (mbtoalmaxload*x(19)*x(19))/12;
mbintmomentmax = (mbtoalmaxload*x(19)*x(19))/12;
mbendmomentmin = (mbtotalminload*x(19)*x(19))/12;
mbintmomentmin = (mbtotalminload*x(19)*x(19))/12;
r1ejuc = mbdfendjointc; % Unit moment applied at end joint - upper column of end↙
joint (row1,column1)
r1ejb = mbdfendjointb; % Unit moment applied at end joint - beam of end joint↙
(row1,column2)
r1ejlc = r1ejuc; % Unit moment applied at end joint - lower column of end joint↙
(row1,column3)
r1ijuc = 0; % Unit moment applied at interior joint - upper column of interior↙
joint (row1,column4)
r1ijeb = r1ejb/2; % Unit moment applied at interior joint - end beam of interior↙
joint (row1,column5)
r1ijib = 0; % Unit moment applied at interior joint - interior beam of interior↙
```

```
joint (row1,column6)
r1ijlc = 0; % Unit moment applied at interior joint - lower column of interior ↙
joint (row1,column7)
r2ejuc = 0; % Unit moment applied at end joint - upper column of end joint (row2, ↙
column1)
r2ejb = mbdfinteriorjointendb/2; % Unit moment applied at end joint - beam of end ↙
joint (row2,column2)
r2ejlc = 0; % Unit moment applied at end joint - lower column of end joint (row2, ↙
column3)
r2ijuc = mbdfinteriorjointc; % Unit moment applied at interior joint - upper column ↙
of interior joint (row2,column4)
r2ijeb = mbdfinteriorjointendb; % Unit moment applied at interior joint - end beam ↙
of interior joint (row2,column5)
r2ijib = mbdfinteriorjointintb; % Unit moment applied at interior joint - interior ↙
beam of interior joint (row2,column6)
r2ijlc = r2ijuc; % Unit moment applied at interior joint - lower column of interior ↙
joint (row2,column7)
r3ejuc = (r1ejuc/r1ijeb)-r2ejuc; % Unit moment applied at end joint - upper column ↙
of end joint (row3,column1)
r3ejb = (r1ejb/r1ijeb)-r2ejb; % Unit moment applied at end joint - beam of end ↙
joint (row3,column2)
r3ejlc = (r1ejlc/r1ijeb)-r2ejlc; % Unit moment applied at end joint - lower column ↙
of end joint (row3,column3)
r3ijuc = (r1ijuc/r1ijeb)-r2ijuc; % Unit moment applied at interior joint - upper ↙
column of interior joint (row3,column4)
r3ijeb = (r1ijeb/r1ijeb)-r2ijeb; % Unit moment applied at interior joint - end beam ↙
of interior joint (row3,column5)
r3ijib = (r1ijib/r1ijeb)-r2ijib; % Unit moment applied at interior joint - interior ↙
beam of interior joint (row3,column6)
r3ijlc = (r1ijlc/r1ijeb)-r2ijlc; % Unit moment applied at interior joint - lower ↙
column of interior joint (row3,column7)
r4ejuc = (r2ejuc/r2ejb)-r1ejuc; % Unit moment applied at end joint - upper column ↙
of end joint (row4,column1)
r4ejb = (r2ejb/r2ejb)-r1ejb; % Unit moment applied at end joint - beam of end joint ↙
(row4,column2)
r4ejlc = (r2ejlc/r2ejb)-r1ejlc; % Unit moment applied at end joint - lower column ↙
of end joint (row4,column3)
r4ijuc = (r2ijuc/r2ejb)-r1ijuc; % Unit moment applied at interior joint - upper ↙
column of interior joint (row4,column4)
r4ijeb = (r2ijeb/r2ejb)-r1ijeb; % Unit moment applied at interior joint - end beam ↙
of interior joint (row4,column5)
r4ijib = (r2ijib/r2ejb)-r1ijib; % Unit moment applied at interior joint - interior ↙
beam of interior joint (row4,column6)
r4ijlc = (r2ijlc/r2ejb)-r1ijlc; % Unit moment applied at interior joint - lower ↙
column of interior joint (row4,column7)
r5ejuc = (r3ejuc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at end joint - upper ↙
column of end joint (row5,column1)
r5ejb = (r3ejb/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at end joint - beam of ↙
end joint (row5,column2)
r5ejlc = (r3ejlc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at end joint - lower ↙
column of end joint (row5,column3)
r5ijuc = (r3ijuc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint - ↙
upper column of interior joint (row5,column4)
```

```
r5ijeb = (r3ijeb/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint -↙
end beam of interior joint (row5,column5)
r5ijib = (r3ijib/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint -↙
interior beam of interior joint (row5,column6)
r5ijlc = (r3ijlc/(r3ejuc+r3ejb+r3ejlc)); % Unit moment applied at interior joint -↙
lower column of interior joint (row5,column7)
r6ejuc = (r4ejuc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at end joint↙
- upper column of end joint (row6,column1)
r6ejb = (r4ejb/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at end joint -↙
beam of end joint (row6,column2)
r6ejlc = (r4ejlc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at end joint↙
- lower column of end joint (row6,column3)
r6ijuc = (r4ijuc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - upper column of interior joint (row6,column4)
r6ijeb = (r4ijeb/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - end beam of interior joint (row6,column5)
r6ijib = (r4ijib/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - interior beam of interior joint (row6,column6)
r6ijlc = (r4ijlc/(r4ijlc+r4ijib+r4ijeb+r4ijuc)); % Unit moment applied at interior↙
joint - lower column of interior joint (row6,column7)
c1r1ejuc = 0; % Moment in members for case 1 - upper column of end joint (case1, ↙
row1,column1)
c1r1ejb = -mbendmomentmax; % Moment in members for case 1 - beam of end joint↙
(case1,row1,column2)
c1r1ejlc = 0; % Moment in members for case 1 - beam of end joint (case1,row1, ↙
column3)
c1r1ijuc = 0; % Moment in members for case 1 - upper column of interior joint↙
(case1,row1,column4)
c1r1ijeb = mbendmomentmax; % Moment in members for case 1 - end beam of interior↙
joint (case1,row1,column5)
c1r1ijib = -mbendmomentmax; % Moment in members for case 1 - interior beam of↙
interior joint (case1,row1,column6)
c1r1ijlc = 0; % Moment in members for case 1 - lower column of interior joint↙
(case1,row1,column7)
c1r2ejuc = r5ejuc*mbendmomentmax; % Moment in members for case 1 - upper column of↙
end joint (case1,row2,column1)
c1r2ejb = r5ejb*mbendmomentmax; % Moment in members for case 1 - beam of end joint↙
(case2,row1,column2)
c1r2ejlc = r5ejlc*mbendmomentmax; % Moment in members for case 1 - beam of end↙
joint (case1,row2,column3)
c1r2ijuc = r5ijuc*mbendmomentmax; % Moment in members for case 1 - upper column of↙
interior joint (case1,row2,column4)
c1r2ijeb = r5ijeb*mbendmomentmax; % Moment in members for case 1 - end beam of↙
interior joint (case1,row2,column5)
c1r2ijib = r5ijib*mbendmomentmax; % Moment in members for case 1 - interior beam of↙
interior joint (case1,row2,column6)
c1r2ijlc = r5ijlc*mbendmomentmax; % Moment in members for case 1 - lower column of↙
interior joint (case1,row2,column7)
c1r3ejuc = r6ejuc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
upper column of end joint (case1,row3,column1)
c1r3ejb = r6ejb*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
beam of end joint (case2,row3,column2)
c1r3ejlc = r6ejlc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 -↙
```

```
beam of end joint (case1,row3,column3)
c1r3ijuc = r6ijuc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 - upper column of interior joint (case1,row3,column4)
c1r3ijeb = r6ijeb*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 - end beam of interior joint (case1,row3,column5)
c1r3ijib = r6ijib*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 - interior beam of interior joint (case1,row3,column6)
c1r3ijlc = r6ijlc*(mbendmomentmax-mbintmomentmax); % Moment in members for case 1 - lower column of interior joint (case1,row3,column7)
c1r4ejuc = c1r1ejuc+c1r2ejuc+c1r3ejuc; % Sum of Moments in members for case 1 - upper column of end joint (case1,row4,column1)
c1r4ejb = c1r1ejb+c1r2ejb+c1r3ejb; % Sum of Moments in members for case 1 - beam of end joint (case2,row4,column2)
c1r4ejlc = c1r1ejlc+c1r2ejlc+c1r3ejlc; % Sum of Moments in members for case 1 - beam of end joint (case1,row4,column3)
c1r4ijuc = c1r1ijuc+c1r2ijuc+c1r3ijuc; % Sum of Moments in members for case 1 - upper column of interior joint (case1,row4,column4)
c1r4ijeb = c1r1ijeb+c1r2ijeb+c1r3ijeb; % Sum of Moments in members for case 1 - end beam of interior joint (case1,row4,column5)
c1r4ijib = c1r1ijib+c1r2ijib+c1r3ijib; % Sum of Moments in members for case 1 - interior beam of interior joint (case1,row4,column6)
c1r4ijlc = c1r1ijlc+c1r2ijlc+c1r3ijlc; % Sum of Moments in members for case 1 - lower column of interior joint (case1,row4,column7)
c2r1ejuc = 0; % Moment in members for case 2 - upper column of end joint (case1, row1,column1)
c2r1ejb = -mbendmomentmax; % Moment in members for case 2 - beam of end joint (case1,row1,column2)
c2r1ejlc = 0; % Moment in members for case 2 - beam of end joint (case1,row1, column3)
c2r1ijuc = 0; % Moment in members for case 2 - upper column of interior joint (case1,row1,column4)
c2r1ijeb = mbendmomentmax; % Moment in members for case 2 - end beam of interior joint (case1,row1,column5)
c2r1ijib = -mbintmomentmin; % Moment in members for case 2 - interior beam of interior joint (case1,row1,column6)
c2r1ijlc = 0; % Moment in members for case 2 - lower column of interior joint (case1,row1,column7)
c2r2ejuc = r5ejuc*mbendmomentmax; % Moment in members for case 2 - upper column of end joint (case1,row2,column1)
c2r2ejb = r5ejb*mbendmomentmax; % Moment in members for case 2 - beam of end joint (case2,row1,column2)
c2r2ejlc = r5ejlc*mbendmomentmax; % Moment in members for case 2 - beam of end joint (case1,row2,column3)
c2r2ijuc = r5ijuc*mbendmomentmax; % Moment in members for case 2 - upper column of interior joint (case1,row2,column4)
c2r2ijeb = r5ijeb*mbendmomentmax; % Moment in members for case 2 - end beam of interior joint (case1,row2,column5)
c2r2ijib = r5ijib*mbendmomentmax; % Moment in members for case 2 - interior beam of interior joint (case1,row2,column6)
c2r2ijlc = r5ijlc*mbendmomentmax; % Moment in members for case 2 - lower column of interior joint (case1,row2,column7)
c2r3ejuc = r6ejuc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 - upper column of end joint (case1,row3,column1)
```

```
c2r3ejb = r6ejb*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
beam of end joint (case2,row3,column2)
c2r3ejlc = r6ejlc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
beam of end joint (case1,row3,column3)
c2r3ijuc = r6ijuc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
upper column of interior joint (case1,row3,column4)
c2r3ijeb = r6ijeb*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
end beam of interior joint (case1,row3,column5)
c2r3ijib = r6ijib*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
interior beam of interior joint (case1,row3,column6)
c2r3ijlc = r6ijlc*(mbintmomentmin-mbintmomentmax); % Moment in members for case 2 -↙
lower column of interior joint (case1,row3,column7)
c2r4ejuc = c2r1ejuc+c2r2ejuc+c2r3ejuc; % Sum of Moments in members for case 2 -↙
upper column of end joint (case1,row4,column1)
c2r4ejb = c2r1ejb+c2r2ejb+c2r3ejb; % Sum of Moments in members for case 2 - beam of↙
end joint (case2,row4,column2)
c2r4ejlc = c2r1ejlc+c2r2ejlc+c2r3ejlc; % Sum of Moments in members for case 2 -↙
beam of end joint (case1,row4,column3)
c2r4ijuc = c2r1ijuc+c2r2ijuc+c2r3ijuc; % Sum of Moments in members for case 2 -↙
upper column of interior joint (case1,row4,column4)
c2r4ijeb = c2r1ijeb+c2r2ijeb+c2r3ijeb; % Sum of Moments in members for case 2 - end↙
beam of interior joint (case1,row4,column5)
c2r4ijib = c2r1ijib+c2r2ijib+c2r3ijib; % Sum of Moments in members for case 2 -↙
interior beam of interior joint (case1,row4,column6)
c2r4ijlc = c2r1ijlc+c2r2ijlc+c2r3ijlc; % Sum of Moments in members for case 2 -↙
lower column of interior joint (case1,row4,column7)
c3r1ejuc = 0; % Moment in members for case 3 - upper column of end joint (case1,↙
row1,column1)
c3r1ejb = -mbintmomentmin; % Moment in members for case 3 - beam of end joint↙
(case1,row1,column2)
c3r1ejlc = 0; % Moment in members for case 3 - beam of end joint (case1,row1,↙
column3)
c3r1ijuc = 0; % Moment in members for case 3 - upper column of interior joint↙
(case1,row1,column4)
c3r1ijeb = mbintmomentmin; % Moment in members for case 3 - end beam of interior↙
joint (case1,row1,column5)
c3r1ijib = -mbendmomentmax; % Moment in members for case 3 - interior beam of↙
interior joint (case1,row1,column6)
c3r1ijlc = 0; % Moment in members for case 3 - lower column of interior joint↙
(case1,row1,column7)
c3r2ejuc = r5ejuc*mbintmomentmin; % Moment in members for case 3 - upper column of↙
end joint (case1,row2,column1)
c3r2ejb = r5ejb*mbintmomentmin; % Moment in members for case 3 - beam of end joint↙
(case2,row1,column2)
c3r2ejlc = r5ejlc*mbintmomentmin; % Moment in members for case 3 - beam of end↙
joint (case1,row2,column3)
c3r2ijuc = r5ijuc*mbintmomentmin; % Moment in members for case 3 - upper column of↙
interior joint (case1,row2,column4)
c3r2ijeb = r5ijeb*mbintmomentmin; % Moment in members for case 3 - end beam of↙
interior joint (case1,row2,column5)
c3r2ijib = r5ijib*mbintmomentmin; % Moment in members for case 3 - interior beam of↙
interior joint (case1,row2,column6)
c3r2ijlc = r5ijlc*mbintmomentmin; % Moment in members for case 3 - lower column of↙
```

```
interior joint (case1,row2,column7)
c3r3ejuc = r6ejuc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
upper column of end joint (case1,row3,column1)
c3r3ejb = r6ejb*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
beam of end joint (case2,row3,column2)
c3r3ejlc = r6ejlc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
beam of end joint (case1,row3,column3)
c3r3ijuc = r6ijuc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
upper column of interior joint (case1,row3,column4)
c3r3ijeb = r6ijeb*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
end beam of interior joint (case1,row3,column5)
c3r3ijib = r6ijib*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
interior beam of interior joint (case1,row3,column6)
c3r3ijlc = r6ijlc*(mbintmomentmax-mbintmomentmin); % Moment in members for case 3 -
lower column of interior joint (case1,row3,column7)
c3r4ejuc = c3r1ejuc+c3r2ejuc+c3r3ejuc; % Sum of Moments in members for case 3 -
upper column of end joint (case1,row4,column1)
c3r4ejb = c3r1ejb+c3r2ejb+c3r3ejb; % Sum of Moments in members for case 3 - beam of
end joint (case2,row4,column2)
c3r4ejlc = c3r1ejlc+c3r2ejlc+c3r3ejlc; % Sum of Moments in members for case 3 -
beam of end joint (case1,row4,column3)
c3r4ijuc = c3r1ijuc+c3r2ijuc+c3r3ijuc; % Sum of Moments in members for case 3 -
upper column of interior joint (case1,row4,column4)
c3r4ijeb = c3r1ijeb+c3r2ijeb+c3r3ijeb; % Sum of Moments in members for case 3 - end
beam of interior joint (case1,row4,column5)
c3r4ijib = c3r1ijib+c3r2ijib+c3r3ijib; % Sum of Moments in members for case 3 -
interior beam of interior joint (case1,row4,column6)
c3r4ijlc = c3r1ijlc+c3r2ijlc+c3r3ijlc; % Sum of Moments in members for case 3 -
lower column of interior joint (case1,row4,column7)

% 2.1.3.1 FINAL BENDING MOMENT AND SHEAR FORCE ANALYSIS/ SUB FRAME ANALYSIS
c1r1bendsupport = c1r4ejb; % Load case 1 - Beam moment end support
mbvl = (mbtoalmaxload*x(19)/2)-((c1r4ijeb-abs(c1r4ejb))/x(19)); % Main beam - Left
support moment for load case 1
mbvr = (mbtoalmaxload*x(19))-(mbvl); % Main beam - Right support moment for load
case 1
mba = mbvl/mbtoalmaxload; % Main beam load case 1 - distance to zero shear
c1mbmaxsagging = (mbvl*mba/2)- abs(c1r4ejb); % Load case 1 main beam - maximum
sagging
c1r1bendspan = c1mbmaxsagging; % Load case 1 - Beam moment end span
c1r1bisleft = c1r4ijeb; % Load case 1 - Beam moment interior left support
c1r1bisright = c1r4ijib; % Load case 1 - Beam moment interior right support
c1r1binteriorspan = ((mbtoalmaxload*x(19)*x(19))/8)-abs(c1r1bisright); % Load case
1 - Beam moment interior span
c1r2ucendsupport = c1r4ejuc; % Load case 1 - Upper column moment end support
c1r2ucendspan = 0; % Load case 1 - Upper column moment end span
c1r2ucisleft = c1r4ijuc; % Load case 1 - Upper column moment interior left support
c1r2ucisright = 0; % Load case 1 - Upper column moment interior right support
c1r2ucinteriorspan = 0; % Load case 1 - Upper column moment interior span
c1r3lcendsupport = c1r4ejlc; % Load case 1 - Upper column moment end support
c1r3lcendspan = 0; % Load case 1 - Upper column moment end span
c1r3lcisleft = c1r4ijlc; % Load case 1 - Upper column moment interior left support
c1r3lcisright = 0; % Load case 1 - Upper column moment interior right support
```

```
c1r3lcinteriorspan = 0; % Load case 1 - Upper column moment interior span

c2r1bendsupport = c2r4ejb; % Load case 2 - Beam moment end support
c2mbvl = (mbtoalmaxload*x(19)/2)-((c2r4ijeb-abs(c2r4ejb))/x(19)); % Main beam -
Left support moment for load case 2
c2mbvr = (mbtoalmaxload*x(19))-(c2mbvl); % Main beam - Right support moment for
load case 2
c2mba = c2mbvl/mbtoalmaxload; % Main beam load case 2 - distance to zero shear
c2mbmaxsagging = (c2mbvl*mba/2)- abs(c2r4ejb); % Load case 2 main beam - maximum
sagging
c2r1bendspan = c2mbmaxsagging; % Load case 2 - Beam moment end span
c2r1bisleft = c2r4ijeb; % Load case 2 - Beam moment interior left support
c2r1bisright = c2r4ijib; % Load case 2 - Beam moment interior right support
c2r1binteriorspan = ((mbtoalmaxload*x(19)*x(19))/8)-abs(c2r1bisright); % Load case
2 - Beam moment interior span
c2r2ucendsupport = c2r4ejuc; % Load case 2 - Upper column moment end support
c2r2ucendspan = 0; % Load case 2 - Upper column moment end span
c2r2ucisleft = c2r4ijuc; % Load case 2 - Upper column moment interior left support
c2r2ucisright = 0; % Load case 2 - Upper column moment interior right support
c2r2ucinteriorspan = 0; % Load case 2 - Upper column moment interior span
c2r3lcendsupport = c2r4ejlc; % Load case 2 - Upper column moment end support
c2r3lcendspan = 0; % Load case 2 - Upper column moment end span
c2r3lcisleft = c2r4ijlc; % Load case 2 - Upper column moment interior left support
c2r3lcisright = 0; % Load case 2 - Upper column moment interior right support
c2r3lcinteriorspan = 0; % Load case 2 - Upper column moment interior span

c3r1bendsupport = c3r4ejb; % Load case 3 - Beam moment end support
c3mbvl = (mbtoalmaxload*x(19)/2)-((c3r4ijeb-abs(c3r4ejb))/x(19)); % Main beam -
Left support moment for load case 3
c3mbvr = (mbtoalmaxload*x(19))-(c3mbvl); % Main beam - Right support moment for
load case 3
c3mba = c3mbvl/mbtoalmaxload; % Main beam load case 3 - distance to zero shear
c3mbmaxsagging = (c3mbvl*mba/2)- abs(c3r4ejb); % Load case 3 main beam - maximum
sagging
c3r1bendspan = c3mbmaxsagging; % Load case 3 - Beam moment end span
c3r1bisleft = c3r4ijeb; % Load case 3 - Beam moment interior left support
c3r1bisright = c3r4ijib; % Load case 3 - Beam moment interior right support
c3r1binteriorspan = ((mbtoalmaxload*x(19)*x(19))/8)-abs(c3r1bisright); % Load case
3 - Beam moment interior span
c3ucendsupport = c3r4ejuc; % Load case 3 - Upper column moment end support
c3r2ucendspan = 0; % Load case 3 - Upper column moment end span
c3r2ucisleft = c3r4ijuc; % Load case 3 - Upper column moment interior left support
c3r2ucisright = 0; % Load case 3 - Upper column moment interior right support
c3r2ucinteriorspan = 0; % Load case 3 - Upper column moment interior span
c3r3lcendsupport = c3r4ejlc; % Load case 3 - Upper column moment end support
c3r3lcendspan = 0; % Load case 3 - Upper column moment end span
c3r3lcisleft = c3r4ijlc; % Load case 3 - Upper column moment interior left support
c3r3lcisright = 0; % Load case 3 - Upper column moment interior right support
c3r3lcinteriorspan = 0; % Load case 3 - Upper column moment interior span

c1r1sfbendsupport = mbvl; % Load case 1 - Beam shear force end support
c1r1sfbendendspan = 0; % Load case 1 - Beam shear force end span
c1r1sfbisleft = mbvr; % Load case 1 - Beam shear force interior left support
```

```
c1r1sfbisright = (mbtoalmaxload*x(19))/2; % Load case 1 - Beam shear force interior
right support
c1r1sfbinteriorspan = 0; % Load case 1 - Beam shear force interior span

c2r2sfbendsupport = abs(c2mbvl); % Load case 2 - Beam shear force end support
c2r2sfbendendspan = 0; % Load case 2 - Beam shear force end span
c2r2sfbisleft = c2mbvr; % Load case 2 - Beam shear force interior left support
c2r2sfbisright = (mbtotalminload*x(19))/2; % Load case 2 - Beam shear force
interior right support
c2r2sfbinteriorspan = 0; % Load case 2 - Beam shear force interior span

c3r3sfbendsupport = c3mbvl; % Load case 3 - Beam shear force end support
c3r3sfbendendspan = 0; % Load case 3 - Beam shear force end span
c3r3sfbisleft = c3mbvr; % Load case 3 - Beam shear force interior left support
c3r3sfbisright = (mbtoalmaxload*x(19))/2; % Load case 3 - Beam shear force interior
right support
c3r3sfbinteriorspan = 0; % Load case 3 - Beam shear force interior span

if abs(c2r1bendsupport)>abs(c1r1bendsupport)
   mbendsupportmoment = abs(c2r1bendsupport);% At end support
else
   mbendsupportmoment = abs(c2r1bendsupport); % At end support
end
mbinteriorsupportmoment = abs(c2r1bisleft); % At interior support
mbinteriorspanmoment = ((mbtoalmaxload*x(19)*x(19))/8)-mbinteriorsupportmoment; %
At interior span

% 2.1.4 FLEXURE DESIGN/CHECK
% 2.1.4.1 At interior support

mbeffectivedepth = x(23)-bnominalcover-10-(18/2); %Effective depth of the main beam
mbk = (mbinteriorsupportmoment*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x
(3));
mbkdash = 0.168;
mbleverarm = 0.5*mbeffectivedepth*(1+(1-(3.53*mbk))^(1/2)); % in mm
mbmaxleverarm = 0.95*mbeffectivedepth;
 if mbleverarm <= mbmaxleverarm
    mbz = mbleverarm;
 else
    mbz = mbmaxleverarm;
 end
mbrequiredsteel = (mbinteriorsupportmoment*1000000)/(0.87*x(4)*mbz); % Main beam
tensile steel at interior support
mbminimumsteel = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main beam
minimum tensile steel at interior support
mbprovidedsteel = x(26)*3.14*x(27)*x(27)/4; % Main beam provided tensile steel at
interior support

% 2.1.4.2 At end support

mbkes = (mbendsupportmoment*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x
(3));
mbkdashes = 0.168;
```

```
mbleverarmes = 0.5*mbeffectivedepth*(1+(1-(3.53*mbkes))^(1/2)); % in mm
mbmaxleverarmes = 0.95*mbeffectivedepth;
 if mbleverarmes <= mbmaxleverarmes
    mbzes = mbleverarmes;
 else
    mbzes = mbmaxleverarmes;
 end
mbrequiredsteeles = (mbendsupportmoment*1000000)/(0.87*x(4)*mbzes); % Main beam tensile steel at interior support
mbminimumsteeles = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main beam minimum tensile steel at interior support
mbprovidedsteeles = x(28)*3.14*x(29)*x(29)/4; % Main beam provided tensile steel at interior support

% 2.1.4.3 At interior span

mbkis = (mbinteriorspanmoment*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x(3));
mbkdashis = 0.168;
mbleverarmis = 0.5*mbeffectivedepth*(1+(1-(3.53*mbkis))^(1/2)); % in mm
mbmaxleverarmis = 0.95*mbeffectivedepth;
 if mbleverarmis <= mbmaxleverarmis
    mbzis = mbleverarmis;
 else
    mbzis = mbmaxleverarmis;
 end
mbrequiredsteelis = (mbinteriorspanmoment*1000000)/(0.87*x(4)*mbzis); % Main beam tensile steel at interior span
mbminimumsteelis = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main beam minimum tensile steel at interior span
mbprovidedsteelis = x(30)*3.14*x(31)*x(31)/4; % Main beam provided tensile steel at interior span

% 2.1.4.4 At exterior span

mbkespan = (abs(c2r1bendspan)*1000000)/(x(18)*mbeffectivedepth*mbeffectivedepth*x(3));
mbkdashespan = 0.168;
mbleverarmespan = 0.5*mbeffectivedepth*(1+(1-(3.53*mbkespan))^(1/2)); % in mm
mbmaxleverarmespan = 0.95*mbeffectivedepth;
 if mbleverarmespan <= mbmaxleverarmespan
    mbzespan = mbleverarmespan;
 else
    mbzespan = mbmaxleverarmespan;
 end
mbrequiredsteelespan = (abs(c2r1bendspan)*1000000)/(0.87*x(4)*mbzespan); % Main beam tensile steel at exterior span
mbminimumsteelespan = (0.26*0.3*x(3)^(2/3)*x(18)*mbeffectivedepth)/x(4); % Main beam minimum tensile steel at exterior span
mbprovidedsteelespan = x(32)*3.14*x(33)*x(33)/4; % Main beam provided tensile steel at exterior span

% 2.1.5 SHEAR DESIGN/CHECK
```

```
mbcriticaldistance = mbeffectivedepth+(x(24)/2);

% 2.1.5.1 At End Support

mbsdendsupport = c2r2sfbendsupport-(mbtoalmaxload*mbcriticaldistance/1000); % Main↙
beam shear design end support
mbsdesroww = mbsdendsupport/(x(18)*0.9*mbeffectivedepth*(1-(x(3))/250)*x(3)); %↙
Main beam shear design factor(row)
mbsdesrowwlimit = 0.138;
mbcottheta = 2.5;
mbsdesrequiredsteel = (mbsdendsupport*1000*1000)/(0.87*x(4)*0.↙
9*mbeffectivedepth*mbcottheta); % Main beam shear design end support required steel
mbsdesprovidedsteel = x(34)*3.14*x(35)*x(35)/4; % % Main beam shear design end↙
support provided steel

% 2.1.5.2 At Interior Support

mbsdisleft = abs(c1r1sfbisleft)-(mbtoalmaxload*mbcriticaldistance/1000); % Main↙
beam shear design interior support
mbsdisleftroww = mbsdisleft/(x(18)*0.9*mbeffectivedepth*(1-(x(3))/250)*x(3)); %↙
Main beam shear design factor(row)
mbsdisleftrowwlimit = 0.138;
mbsdisleftcottheta = 2.5;
mbsdisleftrequiredsteel = (mbsdisleft*1000*1000)/(0.87*x(4)*0.↙
9*mbeffectivedepth*mbsdisleftcottheta); % Main beam shear design end support↙
required steel
mbsdisleftprovidedsteel = x(36)*3.14*x(37)*x(37)/4; % % Main beam shear design end↙
support provided steel

mbsdisright = abs(c1r1sfbisright)-(mbtoalmaxload*mbcriticaldistance/1000); % Main↙
beam shear design interior support
mbsdisrightroww = mbsdisright/(x(18)*0.9*mbeffectivedepth*(1-(x(3))/250)*x(3)); %↙
Main beam shear design factor(row)
mbsdisrightrowwlimit = 0.138;
mbsdisrightcottheta = 2.5;
mbsdisrightrequiredsteel = (mbsdisright*1000*1000)/(0.87*x(4)*0.↙
9*mbeffectivedepth*mbsdisrightcottheta); % Main beam shear design end support↙
required steel
mbsdisrightprovidedsteel = x(38)*3.14*x(39)*x(39)/4; % % Main beam shear design end↙
support provided steel

% 2.1.6 DEFLECTION DESIGN/CHECK

mbactualspantodepth = x(19)/mbeffectivedepth;
mbdeflectionload = sf;
mbbeta = (500/x(4))/(mbprovidedsteelis/mbrequiredsteelis);
mbdeflectioneffectivebreadth = ((x(18)+(0.28*x(19)*1000))*x(1))+(x(18)*↙
(mbeffectivedepth-x(1)));
mbalpha = (0.55+(0.0075*x(3)/(100*mbrequiredsteelis/mbdeflectioneffectivebreadth)))↙
+(0.005*(x(3)^0.5)*(((x(3)^0.5)/↙
(100*mbrequiredsteelis/mbdeflectioneffectivebreadth))-10)^1.5);
mblimitingratio = 30*0.8*(x(19)/(mbdeflectioneffectivebreadth/x(18)))↙
```

```
*mbbeta*mbalpha;

% 2.1.7 REINFOREMENT REQUIREMETNS/DETAILING
% 2.1.7.1 STEEL CALCULATION

mbbrespanl = ((x(19)*1000-x(24))+(50*x(33))+(x(24)/2)+(50*x(33)/2)-(2*x(33))) ↙
/1000;% Main beam bottom reinforcement end span cut length
mbbrespanw = mbbrespanl*2*x(33)*x(33)/162.2; % Weight of end span bottom↙
reinforcement in Kg
mbbrispanl = ((x(19)*1000-x(24))+(50*x(31))+(x(24)/2)+(50*x(31)/2)-(2*x(31))) ↙
/1000;% Main beam bottom reinforcement interior span cut length
mbbrispanw = mbbrispanl*(x(20)-2)*x(31)*x(31)/162.2; % Weight of interior span↙
bottom reinforcement in Kg
mbtresupportl = ((((x(19)*1000)-x(24))/3)+(50*x(29)))/1000;% Main beam top↙
reinforcement exterior support cut length
mbbresupportw = mbtresupportl*2*x(29)*x(29)/162.2; % Weight of exterior support↙
bottom reinforcement in Kg
mbtrisupportl = ((((0.2*0.15*(x(19)*1000*2))*2)+x(24)))/1000;% Main beam top↙
interior support cut length
mbbrisupportw = mbtrisupportl*(x(20)-1)*x(27)*x(27)/162.2; % Weight of interior↙
support bottom reinforcement in Kg
mbstirrupa = x(18)-(2*bnominalcover)-(2*x(37)/2); % Main beam stirrup breadth
mbstirrupb = x(23)-(2*bnominalcover)-(2*x(37)/2); % Main beam stirrup breadth
mbstirrup = ((2*(mbstirrupa+mbstirrupb))+(10*2*x(37))-(3*4*x(37)))/1000; % Main↙
beam stirrup length
mbstirrupnumber = ((x(19)*1000-x(18))/(1000/x(36)))-1; % Number of stirrups
mbstirrupweight = mbstirrup*mbstirrupnumber*x(20)*x(37)/162.2; %Total stirrup↙
weight for entire spans
mbtotalsteelonebeam =↙
mbstirrupweight+mbbrespanw+mbbrispanw+mbbresupportw+mbbrisupportw; % Total steel↙
weight for one main beam
mbtotalsteel = mbtotalsteelonebeam*(x(17)+1)*6; % Total main beam steel for entire↙
building

% 2.1.7.2 Concrete CALCULATION

mbcvolume = x(18)*x(23)*x(20)*x(19)/1000000;% Main beam volue of one total beam in↙
m3
mbcfloorvolume = mbcvolume*(x(17)+1); % Main beam volume on one floor in m3
mbcbuildingvolume = mbcfloorvolume*6; % Main beam volume on one floor in m3
mbcbuildingnetweight = (mbcbuildingvolume*2400)-(mbtotalsteel); % Main beam net↙
weight for whole building in m3
mbcbuildingnetvolume = mbcbuildingnetweight/2400;  % Main beam net volume on whole↙
builsing in m3

% 2.1.7.3 Formwork CALCULATION

mbfarea = (2*x(23)/1000+x(18)/1000)*x(19)*x(20)*(x(17)+1)*6 % Main beam formwork↙
area of whole building in m2
mbfweight = mbfarea*(4/1000)*2710; % Main beam formwork weight of whole building in↙
Kg

% 2.2 EDGE BEAM DESIGN
```

```
% 2.2.1 LOADING CALULATIONS

ebtarea = 0.5*(x(2)-x(24)/1000)*(1/3); % Edge beam triangular area m2
ebwcw = 5; % Edge beam loading due to walling, cladding, windows in Kn/m
ebwb = 1.25*(ebwcw+(25*x(18)*x(23))/1000000)*x(2); % Edge beam load plus walling in
KN
ebsdload = (1.25*ebtarea*stotalpermanentload); % Edge beam dead load due to slab in
KN
ebslload = (1.25*ebtarea*stotalvariableload); % Edge beam live load due to slab in
KN

% 2.2.2 BENDING MOMENT AND SHEAR FORCE ANALYSIS/ SUB FRAME ANALYSIS

ebr1bmes = ((0.078*ebwb)+(0.105*ebsdload)+(0.135*ebslload))*x(2); % Edge beam
bending moment in end span Knm
ebr2bmisupport = ((0.105*ebwb)+(0.132*ebsdload)+(0.132*ebslload))*x(2); % Edge beam
bending moment in first interior support KNm
ebr3bmispan = ((0.046*ebwb)+(0.068*ebsdload)+(0.117*ebslload))*x(2); % Edge beam
bending moment in interior span KNm
ebr4bmosupport = ((0.079*ebwb)+(0.099*ebsdload)+(0.099*ebslload))*x(2); % Edge beam
bending moment in other support KNm

ebr1sfes = ((0.395*ebwb)+(0.369*ebsdload)+(0.434*ebslload)); % Edge beam shear
force in end span Knm
ebr2sfisupport = ((0.605*ebwb)+(0.631*ebsdload)+(0.649*ebslload)); % Edge beam
shear force in first interior support KNm
ebr3sfispan = ((0.526*ebwb)+(0.532*ebsdload)+(0.622*ebslload)); % Edge beam shear
force in interior span KNm
ebr4sfosupport = ((0.5*ebwb)+(0.5*ebsdload)+(0.614*ebslload)); % Edge beam shear
force in other support KNm

% 2.2.3 FLEXURE DESIGN/CHECK

ebeffectivedepth = mbeffectivedepth; % Edge beam effective depth
ebeffectivebreadth = x(18)+(0.2*0.7*x(2)*1000);

% 2.2.3.1 At END SPAN

ebkes = (ebr1bmes*1000000)/(ebeffectivebreadth*ebeffectivedepth*ebeffectivedepth*x
(3)); % Edge beam end span
ebkdashes = 0.168;
ebleverarmes = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkes))^(1/2)); % in mm
ebmaxleverarmes = 0.95*ebeffectivedepth;
 if ebleverarmes <= ebmaxleverarmes
    ebzes = ebleverarmes;
 else
    ebzes = ebmaxleverarmes;
 end
ebrequiredsteeles = (ebr1bmes*1000000)/(0.87*x(4)*ebzes); % Main beam tensile steel
at interior support
ebminimumsteeles = (0.26*0.3*x(3)^(2/3)*ebeffectivedepth*ebeffectivebreadth)/x(4);
% Main beam minimum tensile steel at interior support
ebprovidedsteeles = x(40)*3.14*x(41)*x(41)/4; % Main beam provided tensile steel at
```

```
interior support

% 2.2.3.2 At 1st INTERIOR SUPPORT

ebkis = (ebr2bmisupport*1000000)/(x(18)*ebeffectivedepth*ebeffectivedepth*x(3)); %
Edge beam end span
ebkdashis = 0.168;
ebleverarmis = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkis))^(1/2)); % in mm
ebmaxleverarmis = 0.95*ebeffectivedepth;
 if ebleverarmis <= ebmaxleverarmis
    ebzis = ebleverarmis;
 else
    ebzis = ebmaxleverarmis;
 end
ebrequiredsteelis = (ebr2bmisupport*1000000)/(0.87*x(4)*ebzis); % Main beam tensile
steel at interior support
ebminimumsteelis = (0.26*0.3*x(3)^(2/3)*x(18)*ebeffectivedepth)/x(4); % Main beam
minimum tensile steel at interior support
ebprovidedsteelis = x(42)*3.14*x(43)*x(43)/4; % Main beam provided tensile steel at
interior support

% 2.2.3.3 At Interior SPAN

ebkispan = (ebr3bmispan*1000000)/
(ebeffectivebreadth*ebeffectivedepth*ebeffectivedepth*x(3)); % Edge beam interior
span
ebkdashispan = 0.168;
ebleverarmispan = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkispan))^(1/2)); % in mm
ebmaxleverarmispan = 0.95*ebeffectivedepth;
 if ebleverarmispan <= ebmaxleverarmispan
    ebzispan = ebleverarmispan;
 else
    ebzispan = ebmaxleverarmispan;
 end
ebrequiredsteelispan = (ebr3bmispan*1000000)/(0.87*x(4)*ebzispan); % Main beam
tensile steel at interior span
ebminimumsteelispan = (0.26*0.3*x(3)^(2/3)*ebeffectivedepth*ebeffectivebreadth)/x
(4); % Main beam minimum tensile steel at interior span
ebprovidedsteelispan = x(44)*3.14*x(45)*x(45)/4; % Main beam provided tensile steel
at interior span

% 2.2.3.4 At OTHER INTERIOR SUPPORT

ebkos = (ebr4bmosupport*1000000)/(x(18)*ebeffectivedepth*ebeffectivedepth*x(3)); %
Edge beam OTHER INTERIOR SUPPORT
ebkdashos = 0.168;
ebleverarmos = 0.5*ebeffectivedepth*(1+(1-(3.53*ebkos))^(1/2)); % in mm
ebmaxleverarmos = 0.95*ebeffectivedepth;
 if ebleverarmos <= ebmaxleverarmos
    ebzos = ebleverarmos;
 else
    ebzos = ebmaxleverarmos;
 end
```

```
ebrequiredsteelos = (ebr4bmosupport*1000000)/(0.87*x(4)*ebzos); % Main beam tensile↙
steel at interior support
ebminimumsteelos = (0.26*0.3*x(3)^(2/3)*x(18)*ebeffectivedepth)/x(4); % Main beam↙
minimum tensile steel at interior support
ebprovidedsteelos = x(46)*3.14*x(47)*x(47)/4; % Main beam provided tensile steel at↙
interior support

% 2.2.4 SHEAR DESIGN/CHECK

ebsdrow = ebr2sfisupport/(x(18)*0.87*0.9*ebeffectivedepth/1000*(1-(x(3))/250)*x ↙
(3)); % Main beam shear design factor(row)
ebsdrowlimit = 0.138;
ebcottheta = 2.5;
ebsdrequiredsteel = (ebr2sfisupport*1000)/(0.87*x(4)*0.↙
9*ebeffectivedepth/1000*mbcottheta); % Main beam shear design end support required↙
steel
ebsdprovidedsteel = x(48)*3.14*x(49)*x(49)/4; % % Main beam shear design end↙
support provided steel

% 2.2.5 REINFOREMENT REQUIREMETNS/DETAILING
% 2.2.5.1 STEEL CALCULATION

ebbrespanl = ((x(2)*1000-x(24))+(50*x(41))+(x(24)/2)+(50*x(41)/2)-(2*x(41)))/1000;%↙
Edge beam bottom reinforcement end span cut length
ebbrespanw = ebbrespanl*2*x(41)*x(41)/162.2; % Weight of end span bottom↙
reinforcement in Kg
ebbrispanl = ((x(2)*1000-x(24))+(50*x(45))+(x(24)/2)+(50*x(45)/2)-(2*x(45)))/1000;%↙
Edge beam bottom reinforcement interior span cut length
ebbrispanw = ebbrispanl*(x(17)-2)*x(45)*x(45)/162.2; % Weight of interior span↙
bottom reinforcement in Kg
ebtrisupportl = (((x(2)*1000)/3)+(50*x(43)))/1000;% Edge beam top reinforcement↙
first interior support cut length
ebtrisupportw = ebtrisupportl*2*x(43)*x(43)/162.2; % Weight of first interior↙
support top reinforcement in Kg
ebtrosupportl = (((0.9*x(24))*2)+x(24))/1000;% Edge beam top other interior support↙
cut length
ebtrosupportw = ebtrosupportl*(x(17)-1)*x(47)*x(47)/162.2; % Weight of other↙
interior support top reinforcement in Kg
ebstirrupa = x(18)-(2*bnominalcover)-(2*x(49)/2); % Edge beam stirrup breadth
ebstirrupb = x(23)-(2*bnominalcover)-(2*x(49)/2); % Edge beam stirrup breadth
ebstirrup = ((2*(ebstirrupa+ebstirrupb))+(10*2*x(49))-(6*2*x(49)))/1000; % Edge↙
beam stirrup length
ebstirrupnumber = ((x(2)*1000-x(18))/(1000/x(48)))-1; % Number of stirrups
ebstirrupweight = ebstirrup*ebstirrupnumber*x(17)*x(49)/162.2; %Total stirrup↙
weight for entire spans
ebtotalsteelonebeam =↙
ebstirrupweight+ebbrespanw+ebbrispanw+ebtrisupportw+ebtrosupportw; % Total steel↙
weight for one main beam
ebtotalsteel = ebtotalsteelonebeam*2*6; % Total edge beam steel for entire building

% 2.2.5.2 Concrete CALCULATION

ebcvolume = x(18)*x(23)*x(17)*x(2)/1000000; % Main beam volue of one total beam in↙
```

```
m3
ebcfloorvolume = ebcvolume*2; % Main beam volume on one floor in m3
ebcbuildingvolume = ebcfloorvolume*6; % Main beam volume for whole building in m3
ebcbuildingnetweight = (ebcbuildingvolume*2400)-(ebtotalsteel); % Main beam net↙
weight on one floor in m3
ebcbuildingnetvolume = ebcbuildingnetweight/2400;  % Edge beam net volume on whole↙
builsing in m3

% 2.5.5.3 Formwork CALCULATION

ebfarea = (2*x(23)/1000+x(18)/1000)*x(17)*x(2)*2*6; % Edge beam formwork area of↙
whole building in m2
ebfweight = ebfarea*(4/1000)*2710; % Edge beam formwork weight of whole building in↙
Kg


% 3. COLUMN DESIGN
% 3.1 LOADING

cdf = (1.25*stotalpermanentload)+(1.5*0.6); % Column design load in KN/m2
croofbeammax = (1.1*x(2)*cdf)+mbmax; % Column max interior roof KN/m
croofbeammin = (1.1*x(2)*1.25*stotalpermanentload)+mbmax; % Column min interior↙
roof KN/m
cswperfloor = 1.25*(x(24)/1000)*(x(25)/1000)*25*(3.5-(x(23)/1000)); % Column self↙
weight per floor

% 3.2 EXTERNAL COLUMN
% 3.2.1 BENDING MOMENT AND AXIAL FORCE ANALYSIS

ecfeb = ebwb; % External column load due to edge beam KN
exroof = (1.25*((x(24)*x(25))+0.15*1)*25*x(2)); % External column roof load due to↙
self weight of beam and parapet

ecfminload = (1.1*x(2)*1.5*stotalvariableload)+(mbmax/1000000); % External column↙
minimum load
ecfmaxload = 1.1*x(2)*1.25*stotalpermanentload; % External column maximum load
ecmend = (ecfminload*x(19)*x(19))/12; % External column end moment minimum
ecmendmax = (ecfmaxload*x(19)*x(19))/12; % External column end moment maximum
ecmleft = -ecmend+(0.543*ecmend);
ecmright = ecmend+(0.169*ecmend);
ecsright = (ecfminload*x(19)/2)-((ecmright+ecmleft)/x(19)); % External column shear↙
right
ecmleft1 = -ecmendmax+(0.543*ecmendmax)+((ecmend-ecmendmax)*0.102);
ecmright1 = ecmendmax+(0.169*ecmendmax)+((ecmend-ecmendmax)*0.407);
ecsright1 = (ecmendmax*x(19)/2)-((ecmright1+ecmleft1)/x(19)); % External column↙
shear right

ecmendminr = (croofbeammin*x(19)*x(19))/12; % External column end moment minimum↙
roof beam
ecmendmaxr = (croofbeammax*x(19)*x(19))/12; % External column end moment maximum↙
roof beam
ecmrlc1 = 0.228*ecmendmaxr; % For load case 1 moment
ecmrlc2 = (0.228*ecmendmaxr)+((ecmendminr-ecmendmaxr)*(-0.051)); % For load case 2↙
```

```
moment
ecmrlc1left = -ecmendmaxr+(0.543*ecmendmaxr); % Left moment load case 1
ecmrlc1right = ecmendmaxr+(0.169*ecmendmaxr); % Right moment load case 1
ecsrlc1 = (croofbeammin*x(19)/2)-((ecmrlc1right+ecmrlc1left)/x(19)); % Shear load
case 1
ecmrlc2left = 0.228*ecmendmaxr+((ecmendminr-ecmendmaxr)*(-0.051)); % Left moment
load case 2
ecmrlc2right = ecmendmaxr+(0.169*ecmendmaxr)+(0.407*(ecmendminr-ecmendmaxr)); %
Right moment load case 2
ecsrlc2 = (croofbeammin*x(19)/2)-((ecmrlc2right+ecmrlc2left)/x(19)); % Shear load
case 2

ecroofbeam = (1.1*x(2)*1.5*0.6)+mbmax; % Column max interior roof imposed load KN/m
ecmroofbeam = (ecroofbeam*x(19)*x(19))/12; % External column end moment due to
imposed load
ecmroofbeamleft = 0.543*ecmroofbeam;
ecmroofbeamright = ecmroofbeam+(0.169*ecmroofbeam);
ecsroofbeam = (ecroofbeam*x(19)/2)-((ecmroofbeamright-ecmroofbeamleft)/x(19));

ec11r1n = ecsrlc1+ecfeb; % Roof beam
ec11r1m = ecmrlc1;
ec12r1n = ecsrlc2+ecfeb;
ec12r1m = ecmrlc2;
ec21r1n = ecsroofbeam+exroof;

ec11r2n = cswperfloor; % Column
ec12r2n = cswperfloor;

ec11r3n = ec11r2n+ec11r1n;
ec11r3m = c1r2ucendsupport;
ec12r3n = ec12r2n+ec12r1n;
ec12r3m = c2r2ucendsupport;

ec11r4n = c1r1sfbendsupport+ecfeb; % 4th floor beam
ec12r4n = c2r2sfbendsupport+ecfeb;
ec21r4n = ecsright+ecfeb;
ec22r4n = ecsright1+ecfeb;

ec11r5n = ec11r4n+ec11r3n;
ec11r5m = c1r2ucendsupport;
ec12r5n = ec12r4n+ec12r3n;
ec12r5m = c2r2ucendsupport;

ec11r6n = cswperfloor;
ec12r6n = cswperfloor;

ec11r7n = ec11r6n+ec11r5n;
ec11r7m = ec11r5m;
ec12r7n = cswperfloor+ec12r5n;
ec12r7m = ec12r5m;

ec11r8n = c1r1sfbendsupport+ecfeb; % 3th floor beam
ec12r8n = c2r2sfbendsupport+ecfeb;
```

```
ec21r8n = ecsright+ecfeb;
ec22r8n = ecsright1+ecfeb;

ec11r9n = ec11r8n+ec11r7n;
ec11r9m = ec11r5m;
ec12r9n = ec12r7n+ec12r8n;
ec12r9m = ec12r5m;
ec21r9n = ec21r8n+ec21r4n;
ec22r9n = ec22r8n+ec22r4n;

ec11r10n = cswperfloor;
ec12r10n = cswperfloor;

ec11r11n = ec11r10n+ec11r9n;
ec11r11m = ec11r5m;
ec12r11n = ec12r10n+ec12r9n;
ec12r11m = ec12r5m;

ec11r12n = c1r1sfbendsupport+ecfeb; % 2th floor beam
ec12r12n = c2r2sfbendsupport+ecfeb;
ec21r12n = ecsright+ecfeb;
ec22r12n = ecsright1+ecfeb;

ec11r13n = ec11r12n+ec11r11n;
ec11r13m = ec11r5m;
ec12r13n = ec12r12n+ec12r11n;
ec12r13m = ec12r5m;
ec21r13n = ec21r12n+ec21r9n;
ec22r13n = ec22r12n+ec22r9n;

ec11r14n = ec11r5m;
ec12r14n = ec12r5m;

ec11r15n = ec11r14n+ec11r13n; % 1th floor beam
ec11r15m = ec11r5m;
ec12r15n = ec12r14n+ec12r13n;
ec12r15m = ec12r5m;

ec11r16n = c1r1sfbendsupport+ecfeb;
ec12r16n = c2r2sfbendsupport+ecfeb;
ec21r16n = ecsright+ecfeb;
ec22r16n = ecsright1+ecfeb;

ec11r17n = ec11r16n+ec11r15n;
ec11r17m = ec11r5m;
ec12r17n = ec12r16n+ec12r15n;
ec12r17m = ec12r5m;
ec21r17n = ec21r16n+ec21r13n;
ec22r17n = ec22r16n+ec22r13n;

ec11r18n = ec11r5m;
ec12r18n = ec12r5m;
```

```
ec11r19n = ec11r18n+ec11r17n; % Ground floor beam
ec11r19m = ec11r5m;
ec12r19n = ec12r18n+ec12r17n;
ec12r19m = ec12r5m;

ec11r20n = c1r1sfbendsupport+ecfeb;
ec12r20n = c2r2sfbendsupport+ecfeb;
ec21r20n = ecsright+ecfeb;
ec22r20n = ecsright1+ecfeb;

ec11r21n = ec11r20n+ec11r19n; % Basement wall
ec11r21m = ec11r5m;
ec12r21n = ec12r20n+ec12r19n;
ec12r21m = ec12r5m;
ec21r21n = ec21r20n+ec21r17n;
ec22r21n = ec22r20n+ec22r17n;

ecmbotlc2 = ec12r21m; % Moment at bottom from ground to first floor
ecmtoplc2 = -0.5*ecmbotlc2; % Moment at bottom from ground to first floor
ecnedmax = ec12r19n-0.3*ec22r17n; % Ned maximum
ecnedmin = (ec11r17n-(ec21r17n+ec21r1n))/1.25; % Ned minimum
ecnedmintotal = ecnedmax*x(25)/30 ; % Ned minimum total in KNm

% 3.2.2 EFFECTIVE LENGTH AND SLENDERNESS

eclo = 0.75*(3.5-x(23)/1000);% Effective legth
ecm1 = (ecnedmax*eclo)/400; % First order moment
ecm01i = ecmtoplc2+ecm1; % First order moment with imperfections
ecm02i = ecmbotlc2+ecm1; % First order moment with imperfections
eci = x(25)/(12^(1/2));% Radius of gyration
ecslenderness = eclo/eci; % Slenderness ratio
ecn = (ecnedmax)/(x(24)*x(25)*0.85*x(3)/1.5);
ecc = 1.7-(ecm01i/ecm02i);
ecslendernesslimit = (20*0.7*1.1*ecc)/(ecn^(1/2)); % Slenderness ratio limit

% 3.2.3 DESIGN OF CROSS-SECTION

eced = x(25)-(35+8+20/2); % External column effective depth
ecedratio = eced/x(24); % External column effective depth to height
ecg1nedratio = (ecnedmax*1000)/(x(24)*x(25)*x(3)); % ground-first floor axial force↙
ratio
ecg1medratio = (ecm02i*1000000)/(x(24)*x(25)*x(25)*x(3)); % ground-first floor↙
moment ratio
ecg1rs = ((ecg1nedratio*ecg1medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel↙
for ground floor to first floor
ecg1ps = x(50)*3.14*x(51)*x(51)/4; % provided steel for ground floor to first floor

ecmbot12 = ec12r21m; % Moment at bottom from first to second floor
ecnedmax12 = ec12r15n-0.3*ec22r13n; % Ned maximum
ecm12 = (ecnedmax12*eclo)/400; % First order moment
ecm02i12 = ecmbot12+ecm12; % First order moment with imperfections
ecg1nedratio12 = (ecnedmax12*1000)/(x(24)*x(25)*x(3)); % first to second floor↙
axial force ratio
```

```
ecg1medratio12 = (ecm02i12*1000000)/(x(24)*x(25)*x(25)*x(3)); % first to second
floor moment ratio
ec12rs = ((ecg1nedratio12*ecg1medratio12*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for first to second floor
ec12ps = x(52)*3.14*x(53)*x(53)/4; % provided steel for first to second floor

ecmbot23 = ec12r21m; % Moment at bottom from second to third floor
ecnedmax23 = ec12r11n-0.3*ec22r9n; % Ned maximum
ecm23 = (ecnedmax23*eclo)/400; % First order moment
ecm02i23 = ecmbot23+ecm23; % First order moment with imperfections
ecg1nedratio23 = (ecnedmax23*1000)/(x(24)*x(25)*x(3)); % second to third floor
floor axial force ratio
ecg1medratio23 = (ecm02i23*1000000)/(x(24)*x(25)*x(25)*x(3)); % second to third
floor moment ratio
ec23rs = ((ecg1nedratio23*ecg1medratio23*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for second to third floor
ec23ps = x(54)*3.14*x(55)*x(55)/4; % provided steel for second to third floor

ecmbot34 = ec12r21m; % Moment at bottom from third to fourth floor
ecnedmax34 = ec12r7n-0.3*ec22r4n; % Ned maximum
ecm34 = (ecnedmax34*eclo)/400; % First order moment
ecm02i34 = ecmbot34+ecm34; % First order moment with imperfections
ecg1nedratio34 = (ecnedmax34*1000)/(x(24)*x(25)*x(3)); % third to fourth floor
floor axial force ratio
ecg1medratio34 = (ecm02i34*1000000)/(x(24)*x(25)*x(25)*x(3)); % third to fourth
floor moment ratio
ec34rs = ((ecg1nedratio34*ecg1medratio34*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for third to fourth floor
ec34ps = x(56)*3.14*x(57)*x(57)/4; % provided steel for third to fourth floor

ecmbot4r = ec12r21m; % Moment at bottom from fourth to roof floor
ecnedmax4r = ec12r7n-0.3*ec22r4n; % Ned maximum
ecm4r = (ecnedmax4r*eclo)/400; % First order moment
ecm02i4r = ecmbot4r+ecm4r; % First order moment with imperfections
ecg1nedratio4r = (ecnedmax4r*1000)/(x(24)*x(25)*x(3)); % fourth to roof floor axial
force ratio
ecg1medratio4r = (ecm02i4r*1000000)/(x(24)*x(25)*x(25)*x(3)); % fourth to roof
floor moment ratio
ec4rrs = ((ecg1nedratio4r*ecg1medratio4r*10)*(x(24)*x(25)*x(3)))/x(4); % Required
steel for fourth to roof floor
ec4rps = x(58)*3.14*x(59)*x(59)/4; % provided steel for fourth to roof floor

% 3.2.3.1 DESIGN OF TIES

ectalsmax = stotalpermanentload+(0.7*stotalvariableload); %External column ties
accidental load slab max
ectalsmin = 1.25*stotalvariableload; %External column ties accidental load slab min
ectalbmax = (1.1*x(2)*ectalsmax)+(x(24)*x(25)*25); %External column ties accidental
load beam max
ectalbmin = (1.1*x(2)*ectalsmin)+(x(24)*x(25)*25); %External column ties accidental
load beam min
ectaltotal = ((ecfeb/(ectalbmax+ectalbmin))*c2r2sfbendsupport)+(ecfeb/1.25);
ectrr = (ectaltotal*1000)/x(4); % External column ties required reinforcement
```

```
ectpr = x(60)*3.14*x(61)*x(61)/4; % External column ties provided reinforcement

% 3.2.4 REINFORCEMENT REQUIREMENTS/DETAILING

ec4rsbl = ((1.5*35*x(59))+75)/1000;% External column fourth-roof floor starter bar
length
ec4rl = (3500+ec4rsbl)/1000;% External column fourth-roof floor bar length
ec4rz1links = 600/150; % Links in zone 1
ec4rz2links = (3500-600)/(1000/x(60)); % Links in zone 2
ec4rlinkl = ((2*((x(24)-35*2)+(x(25)-35*2)))+(2*10*x(61))+(3*2*x(61)))/1000; % Link
Length
ec4rlinkltotal = ec4rlinkl*(ec4rz1links+ec4rz2links);% Link Length total
ec4rsteelw = ec4rl*x(58)*(x(59)*x(59)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2)+
(ec4rsbl*x(58)*x(58)/162.2); %Total weight 4-roof floor one column
ec34steelw = ec4rl*x(56)*(x(57)*x(57)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2); %
Total weight 3-4 floor one column
ec23steelw = ec4rl*x(54)*(x(55)*x(55)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2); %
Total weight 2-3 floor one column
ec12steelw = ec4rl*x(52)*(x(53)*x(53)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2); %
Total weight 1-2 floor one column
ecg1sbl = ((1.5*35*x(51))+75)/1000;% External column fourth-roof floor starter bar
length
ecg1steelw = ec4rl*x(50)*(x(51)*x(51)/162.2)+(ec4rlinkltotal*x(61)*x(61)/162.2)+
(ecg1sbl*x(50)); %Total weight g-1 floor one column
ecsteeltotalone = ec4rsteelw+ec34steelw+ec23steelw+ec12steelw+ecg1steelw; %Total
steel in kg for one column whole building
ecsteeltotal = ecsteeltotalone*(((x(17)-1)*2)+((x(20)-1)*2)); %Total steel in kg
for external columns in whole building
ecconcretetotal = (x(24)*x(25)*3.5*5/1000000)*(((x(17)-1)*2)+((x(20)-1)*2));%Total
concrete in m3 for external columns in whole building
ecconcretenet = ((ecconcretetotal*2400)-ecsteeltotal)/2400; % Net concrete in m3
for external columns in whole building
ecformwork = (2*((x(24)/1000)+(x(25)/1000)))*(3.5*5)*(((x(17)-1)*2)+((x(20)-1)
*2));% External column formwork m2
ecformworkw = ecformwork*(4/1000)*2710; % External column formwork weight kg

% 3.3 INTERNAL COLUMN
% 3.3.1 BENDING MOMENT AND AXIAL FORCE ANALYSIS

icfminload = 1.1*x(2)*stotalpermanentload; % Internal column minimum load
icmendmin = (icfminload*x(19)*x(19))/12; % Internal column end moment minimum
iclc1left = ecsrlc1; % Internal column load case 1 left
iclc1right = (croofbeammax*x(19))-iclc1left; % Internal column load case 1 right
iclc1rightis = (croofbeammax*x(19))/2; % Internal column load case 1 left internal
support
iclc1rn = iclc1rightis+iclc1right; % Internal column load case 1 axial force

iclc3mendmax = ecmendmaxr; % Load case 3 maximum moment
iclc3mendmin = ecmendminr; % Load case 3 minimum moment
iclc3mleft = (0.228*iclc3mendmin)+(-0.051*(iclc3mendmax-iclc3mendmin)); % Internal
column load case 3 moment left
iclc3mright = iclc3mendmin+(0.169*iclc3mendmin)+((iclc3mendmax-iclc3mendmin)*0.
407); % Internal column load case 3 moment right
```

```
iclc3sleft = ((croofbeammin*x(19))/2)-((iclc3mright-iclc3mleft)/7); % Internal
column load case 3 shear left
iclc3sright = (croofbeammin*x(19))-iclc3sleft; % Internal column load case 3 shear
right
iclc3rn = (croofbeammin*x(19)/2)+iclc3sright; % Internal column load case 1 axial
force roof

iclc1sleftl2r = ecsroofbeam; % Internal column load case 1 shear left load setting
2
iclc1srightl2r = (ecroofbeam*x(19))-iclc1sleftl2r; % Internal column load case 1
shear right load setting 2
iclc1l2rn = ((ecroofbeam*x(19))/2)+iclc1srightl2r;% Internal column load case 1
axial force roof load setting 2

iclc1sleftl2 = ecsright; % Internal column load case 1 shear left load setting 2
iclc1srightl2 = (ecfminload*x(19))-iclc1sleftl2; % Internal column load case 1
shear right load setting 2
iclc1l2n = (ecfminload*x(19)/2)+iclc1srightl2;% Internal column load case 1 axial
force roof load setting 2

icmendmax = ecmendmaxr; % Internal column end moment maximum
icmendmin = (1.1*x(2)*1.25*stotalvariableload*x(19)*x(19))/12; % Internal column
end moment minimum
iclc3mleftl2 = -icmendmin+(0.543*icmendmin)+((icmendmax-icmendmin)*0.102);
iclc3mrightl2 = icmendmin+(0.169*icmendmin)+((icmendmax-icmendmin)*0.407);
iclc3sleftl2 = (((1.1*x(2)*1.25*stotalvariableload)*x(19))/2)-((iclc3mrightl2-
iclc3mleftl2)/x(19)); % Internal column load case 1 shear left load setting 2
iclc3srightl2 = ((1.1*x(2)*1.25*stotalvariableload)*x(19))-iclc3sleftl2; % Internal
column load case 1 shear right load setting 2
iclc3l2n = ((1.1*x(2)*1.25*stotalvariableload)*x(19)/2)+iclc3srightl2;% Internal
column load case 1 axial force roof load setting 2

ic11r1n = iclc1rn;
ic11r1m = abs(-0.051*ecmendmaxr);
ic13r1n = iclc3rn;
ic13r1m = ic11r1m+((ecmend-icmendmin)*0.177);
ic21r1n = iclc1l2rn;

ic11r2n = cswperfloor;
ic13r2n = cswperfloor;

ic11r3n = ic11r2n+ic11r1n;
ic11r3m = abs(c1r2ucisleft);
ic13r3n = ic13r2n+ic13r1n;
ic13r3m = abs(c3r2ucisleft);

ic11r4n = c1r1sfbisleft+c1r1sfbisright;
ic13r4n = c3r3sfbisleft+c3r3sfbisright;
ic21r4n = iclc1l2n;
ic23r4n = iclc3l2n;

ic11r5n = ic11r4n+ic11r3n;
ic13r5n = ic13r4n+ic13r3n;
```

```
ic11r6n = cswperfloor;
ic13r6n = cswperfloor;

ic11r7n = ic11r6n+ic11r5n;
ic11r7m = abs(c1r2ucisleft);
ic13r7n = ic13r6n+ic13r5n;
ic13r7m = abs(c3r2ucisleft);

ic11r8n = c1r1sfbisleft+c1r1sfbisright;
ic13r8n = c3r3sfbisleft+c3r3sfbisright;
ic21r8n = iclc1l2n;
ic23r8n = iclc3l2n;

ic11r9n = ic11r8n+ic11r7n;
ic13r9n = ic13r8n+ic13r7n;
ic21r9n = ic21r8n+ic21r4n;
ic23r9n = ic23r8n+ic23r4n;

ic11r10n = cswperfloor;
ic13r10n = cswperfloor;

ic11r11n = ic11r10n+ic11r9n;
ic11r11m = abs(c1r2ucisleft);
ic13r11n = ic13r10n+ic13r9n;
ic13r11m = abs(c3r2ucisleft);

ic11r12n = c1r1sfbisleft+c1r1sfbisright;
ic13r12n = c3r3sfbisleft+c3r3sfbisright;
ic21r12n = iclc1l2n;
ic23r12n = iclc3l2n;

ic11r13n = ic11r12n+ic11r11n;
ic13r13n = ic13r12n+ic13r11n;
ic21r13n = ic21r12n+ic21r9n;
ic23r13n = ic23r12n+ic23r9n;

ic11r14n = cswperfloor;
ic13r14n = cswperfloor;

ic11r15n = ic11r14n+ic11r13n;
ic11r15m = abs(c1r2ucisleft);
ic13r15n = ic13r14n+ic13r13n;
ic13r15m = abs(c3r2ucisleft);

ic11r16n = c1r1sfbisleft+c1r1sfbisright;
ic13r16n = c3r3sfbisleft+c3r3sfbisright;
ic21r16n = iclc1l2n;
ic23r16n = iclc3l2n;

ic11r17n = ic11r16n+ic11r15n;
ic13r17n = ic13r16n+ic13r15n;
ic21r17n = ic21r16n+ic21r13n;
```

```
ic23r17n = ic23r16n+ic23r13n;

ic11r18n = cswperfloor;
ic13r18n = cswperfloor;

ic11r19n = ic11r18n+ic11r17n;
ic11r19m = abs(c1r2ucisleft);
ic13r19n = ic13r18n+ic13r17n;
ic13r19m = abs(c3r2ucisleft);

ic11r20n = c1r1sfbisleft+c1r1sfbisright;
ic13r20n = c3r3sfbisleft+c3r3sfbisright;
ic21r20n = iclc1l2n;
ic23r20n = iclc3l2n;

ic11r21n = ic11r20n+ic11r19n;
ic13r21n = ic13r20n+ic13r19n;
ic21r21n = ic21r20n+ic21r17n;
ic23r21n = ic23r20n+ic23r17n;

ic11r22n = cswperfloor;
ic13r22n = cswperfloor;

ic11r23n = ic11r22n+ic11r21n;
ic13r23n = ic13r22n+ic13r21n;

% 3.3.2 EFFECTIVE LENGTH AND SLENDERNESS

icmtoplc3 = ic13r19m; % Moment at top from basement to ground floor load case 3
icmbotlc3 = -0.5*icmtoplc3; % Moment at bottom from basement to ground floor load
case 3
icnedmax = ic11r19n+ic13r20n-(0.4*(ic21r17n+ic23r16n)); % Ned maximum
icnedmin = ic13r20n+((ic11r19n-(ic21r17n+ic21r1n))/1.25); % Ned minimum

iclo = 0.75*(3.5-x(23)/1000);% Effective legth
icslenderness = ecslenderness;% Slenderness ratio
icm1 = (icnedmax*iclo)/400; % First order moment
icm01i = icmbotlc3+icm1; % First order moment with imperfections
icm02i = icmtoplc3+icm1; % First order moment with imperfections
icc = 1.7-(icm01i/icm02i);
icn = (icnedmax)/(x(24)*x(25)*0.85*x(3)/1.5);
icslendernesslimit = (20*0.7*1.1*icc)/(icn^(1/2)); % Slenderness ratio limit

% 3.3.3 DESIGN OF CROSS-SECTION

icbgnedmax = ic11r23n+ic13r20n-(0.4*(ic21r21n+ic23r16n)); % Ned maximum basement-
ground floor
icbgmedmax = icmtoplc3+((icbgnedmax*iclo)/400);
icbgnedratio = (icbgnedmax*1000)/(x(24)*x(25)*x(3)); % basement-ground floor axial
force ratio
icbgmedratio = (icbgmedmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % basement-ground
floor moment ratio
icbgrs = ((icbgnedratio*icbgmedratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel
```

```
for basement-ground floor
icbgps = x(62)*3.14*x(63)*x(63)/4; % provided steel for basement-ground floor

icg1nedmax = ic11r19n+ic13r20n-(0.4*(ic21r17n+ic23r16n)); % Ned maximum ground-↙
first floor
icg1medmax = icmtoplc3+((icg1nedmax*iclo)/400);
icg1nedratio = (icg1nedmax*1000)/(x(24)*x(25)*x(3)); % ground-first floor axial↙
force ratio
icg1medratio = (icg1medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % ground-first floor↙
moment ratio
icg1rs = ((icg1nedratio*icg1medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel↙
for ground-first floor
icg1ps = x(64)*3.14*x(65)*x(65)/4; % provided steel for ground-first floor

ic12nedmax = ic11r15n+ic13r20n-(0.4*(ic21r13n+ic23r16n)); % Ned maximum first-↙
second floor
ic12medmax = icmtoplc3+((ic12nedmax*iclo)/400);
ic12nedratio = (ic12nedmax*1000)/(x(24)*x(25)*x(3)); % first-second floor axial↙
force ratio
ic12medratio = (ic12medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % first-second floor↙
moment ratio
ic12rs = ((ic12nedratio*ic12medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel↙
for first-second floor
ic12ps = x(66)*3.14*x(67)*x(67)/4; % provided steel for first-second floor

ic23nedmax = ic11r11n+ic13r20n-(0.4*(ic21r9n+ic23r16n)); % Ned maximum second-third↙
floor
ic23medmax = icmtoplc3+((ic23nedmax*iclo)/400);
ic23nedratio = (ic23nedmax*1000)/(x(24)*x(25)*x(3)); % second-third floor axial↙
force ratio
ic23medratio = (ic23medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % second-third floor↙
moment ratio
ic23rs = ((ic23nedratio*ic23medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel↙
for second-third floor
ic23ps = x(68)*3.14*x(69)*x(69)/4; % provided steel for second-third floor

ic34nedmax = ic11r7n+ic13r20n-(0.4*(ic21r4n+ic23r16n)); % Ned maximum third-fourth↙
floor
ic34medmax = icmtoplc3+((ic34nedmax*iclo)/400);
ic34nedratio = (ic34nedmax*1000)/(x(24)*x(25)*x(3)); % third-fourth floor axial↙
force ratio
ic34medratio = (ic34medmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % third-fourth floor↙
moment ratio
ic34rs = ((ic34nedratio*ic34medratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel↙
for third-fourth floor
ic34ps = x(70)*3.14*x(71)*x(71)/4; % provided steel for third-fourth floor

ic4rnedmax = ic11r3n+ic13r20n-(0.4*(ic21r1n)); % Ned maximum fourth-roof floor
ic4rmedmax = icmtoplc3+((ic4rnedmax*iclo)/400);
ic4rnedratio = (ic4rnedmax*1000)/(x(24)*x(25)*x(3)); % fourth-roof floor axial↙
force ratio
ic4rmedratio = (ic4rmedmax*1000000)/(x(24)*x(25)*x(25)*x(3)); % fourth-roof floor↙
moment ratio
```

```
ic4rrs = ((ic4rnedratio*ic4rmedratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel for fourth-roof floor
ic4rps = x(72)*3.14*x(73)*x(73)/4; % provided steel for fourth-roof floor

% 3.3.3.1 DESIGN OF TIES - INTERNAL COLUMN

ictaltotal = (ecfeb/(ectalbmax+ectalbmin))*ic11r4n; % Internal column ties accidental load beam min
ictrr = (ictaltotal*1000)/x(4); % Internal column ties required reinforcement
ictpr = x(74)*3.14*x(75)*x(75)/4; % Internal column ties provided reinforcement

% 3.3.4 REINFORCEMENT REQUIREMENTS/DETAILING

ic4rsbl = ((1.5*35*x(73))+75)/1000;% Internal column fourth-roof floor starter bar length
ic4rl = (3500+ic4rsbl)/1000;% Internal column fourth-roof floor bar length
ic4rz1links = 600/150; % Links in zone 1
ic4rz2links = (3500-600)/(1000/x(74)); % Links in zone 2
ic4rlinkl = ((2*((x(24)-35*2)+(x(25)-35*2)))+(2*10*x(75))+(3*2*x(75)))/1000; % Link Length
ic4rlinkltotal = ic4rlinkl*(ic4rz1links+ic4rz2links);% Link Length total
ic4rsteelw = ic4rl*x(72)*(x(73)*x(73)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2)+(ic4rsbl*x(72)*x(72)/162.2); %Total weight 4-roof floor one column
ic34steelw = ic4rl*x(70)*(x(71)*x(71)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); % Total weight 3-4 floor one column
ic23steelw = ic4rl*x(68)*(x(69)*x(69)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); % Total weight 2-3 floor one column
ic12steelw = ic4rl*x(66)*(x(67)*x(67)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); % Total weight 1-2 floor one column
icg1steelw = ic4rl*x(64)*(x(65)*x(65)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2); % Total weight g-1 floor one column
icbgsbl = ((1.5*35*x(63))+75)/1000;% Internal column basement-ground floor starter bar length
icbgsteelw = ic4rl*x(62)*(x(63)*x(63)/162.2)+(ic4rlinkltotal*x(75)*x(75)/162.2)+(icbgsbl*x(62)); %Total weight basement-ground floor one column
icsteeltotalone = ic4rsteelw+ic34steelw+ic23steelw+ic12steelw+icg1steelw+icbgsteelw; %Total steel in kg for one column whole building
icsteeltotal = icsteeltotalone*((x(17)-1)*(x(20)-1)); %Total steel in kg for internal columns in whole building

icconcretetotal = (x(24)*x(25)*3.5*6/1000000)*((x(17)-1)*(x(20)-1));%Total concrete in m3 for internal columns in whole building
icconcretenet = ((icconcretetotal*2400)-icsteeltotal)/2400; % Net concrete in m3 for Internal columns in whole building

icformwork = (2*((x(24)/1000)+(x(25)/1000)))*(3.5*6)*((x(17)-1)*(x(20)-1)); % Internal column formwork m2
icformworkw = icformwork*(4/1000)*2710; % Internal column formwork weight kg

% 3.4 CORNER COLUMN
% 3.4.1 BENDING MOMENT AND AXIAL FORCE ANALYSIS
```

```
ccslinea = 0.4*x(2)*sdesignload; % Load due to slab on line A
ccblinea = ebwb/x(2); % Load due to beam and waling on line A
ccsline1 = ebsdload+ebslload; % Load due to slab on line 1
ccbline1 = ecfeb; % Load due to beam and waling on line 1
cctotallinea = ccslinea+ccblinea; % Total load on line A
cctotalline1 = ccsline1+ccbline1; % Total load on line 1
ccmz = (cctotallinea/mbtoalmaxload)*c2r2ucendsupport; % Column moment of frame on↙
line A

ccbendk = (0.5*mbendk)/(x(2)*1000); % Stiffness of end beam
ccuck = mbicolumn/storeyheight*1000; %STIFFNESS OF UPPER COLUMN
cclck = mbicolumn/storeyheight*1000; %STIFFNESS OF LOWER COLUMN

ccbfem = ((0.104*ccsline1)+(0.083*ccbline1))/x(2); % Beam fix end moment
ccmy = (ccuck/((2*cclck)+ccbendk))*ccbfem; % Column moment

ccslinear = 0.4*x(2)*ebtarea; % Load due to slab on roof level at line A
ccblinea1r = mbtotalminload/1.25; % Load due to beam and waling on roof level at↙
line 1
ccblinear = ccblinea1r/x(2); % Load due to beam and waling on roof level at line A
cctotallinear = ccslinear+ccblinear; % Total load on line A
cctotalline1r = ccblinea1r; % Total load on line 1

ccned = (((cctotallinear/mbtoalmaxload)*abs(c1r1sfbendsupport))+(0.525 ↙
*cctotalline1r)+(abs(cswperfloor)/1.25))/1000;
ccmi = (ccned*iclo)/(400); % First order moments from imperfections
ccm0z = ccmi+ccmz; % First order moments from imperfections
ccm0y = ccmy; % First order moments from imperfections

% 3.4.2 DESIGN OF CROSS-SECTION

ccnedratio = (ccned*1000)/(x(24)*x(25)*x(3)); % fourth-roof floor axial force ratio
ccmedratio = (ccm0z*1000000)/(x(24)*x(25)*x(25)*x(3)); % fourth-roof floor moment↙
ratio
ccrs = ((ccnedratio*ccmedratio*10)*(x(24)*x(25)*x(3)))/x(4); % Required steel for↙
fourth-roof floor
ccps = x(76)*3.14*x(77)*x(77)/4; % provided steel for fourth-roof floor


% 3.4.3 REINFORCEMENT REQUIREMENTS/DETAILING

cc4rsbl = ((1.5*35*x(77))+75)/1000; % Corner column fourth-roof floor starter bar↙
length
cc4rl = (3500+cc4rsbl)/1000;% Corner column fourth-roof floor bar length
cc4rz1links = 600/150; % Links in zone 1
cc4rz2links = (3500-600)/(1000/x(74)); % Links in zone 2
cc4rlinkl = ((2*((x(24)-35*2)+(x(25)-35*2)))+(2*10*x(75))+(3*2*x(75)))/1000; % Link↙
Length
cc4rlinkltotal = cc4rlinkl*(cc4rz1links+cc4rz2links);% Link Length total
cctotalsteelonew = 5*(cc4rl*x(76)*(x(77)*x(77)/162.2))+5*(cc4rlinkltotal*x(75)*x ↙
(75)/162.2)+2*(cc4rsbl*x(77)*x(77)/162.2); %Total weight 4-roof floor o
cctotalsteelbuildingw = 4*cctotalsteelonew; % Total building steel in corner↙
columns
```

```
ccconcretetotal = (x(24)*x(25)*3.5*5/1000000)*4;%Total concrete in m3 for internal↙
columns in whole building
ccconcretenet = ((ccconcretetotal*2400)-cctotalsteelbuildingw)/2400; % Net concrete↙
in m3 for Internal columns in whole building

ccformwork = (2*((x(24)/1000)+(x(25)/1000)))*(3.5*5)*4; % Internal column formwork↙
m2
ccformworkw = ccformwork*(4/1000)*2710; % Internal column formwork weight kg

% 4 QUANTITY CALCULATION

bsteel= slabtotalbuildingsteel;  % Building slab steel in kg
bbsteel = mbtotalsteel+ebtotalsteel; % Building beams steel in kg
bcsteel = ecsteeltotal+icsteeltotal+cctotalsteelbuildingw; % Building column steel↙
in kg
btotalsteel = bsteel+bbsteel+bcsteel; %Building total steel in kg

bsconcrete= sncbuilding; % Building slab concrete in m3
bbconcrete = mbcbuildingnetvolume+ebcbuildingnetvolume;  % Building beams concrete↙
in m3
bcconcrete = ecconcretenet+icconcretenet+ccconcretenet; % Building column concrete↙
in m3
btotalconcrete = bsconcrete+bbconcrete+bcconcrete; %Building total concrete in m3

bsfa = sfa; % Building slab formwork in m2
bbfa = mbfarea+ebfarea; % Building beams formwork in m2
bcfa = ecformwork+icformwork+ccformwork; % Building column formwork in m2
btotalfa = bsfa+bbfa+bcfa; %Building total formwork in m2


bsfw = sfweight; % Building slab formwork in kg
bbfw = mbfweight+ebfweight+ebfweight;  % Building beams formwork in kg
bcfw = ecformworkw+icformworkw+ccformworkw; % Building column formwork in kg
btotalfw = bsfw+bbfw+bcfw; %Building total formwork in kg

Cineq = [30-sa,smpespanrequiredsteel-smpespanprovidedsteel,smpespanminimumsteel-↙
smpespanrequiredsteel,...
smfesuprequiredsteel-smfesupprovidedsteel,smfesupminimumsteel-smfesuprequiredsteel,↙
smfesuprequiredsteel-smfesupmaximumsteel,smfespanrequiredsteel-↙
smfespanprovidedsteel,smfespanminimumsteel-smfespanrequiredsteel,↙
smfespanrequiredsteel-smfespanmaximumsteel,...
smfisupportrequiredsteel-smfisupportprovidedsteel,smfisupportminimumsteel-↙
smfisupportrequiredsteel,smfisupportrequiredsteel-smfisupportmaximumsteel,↙
smaispanrequiredsteel-smaispanprovidedsteel,smaispanminimumsteel-↙
smaispanrequiredsteel,...
smaispanrequiredsteel-smaispanmaximumsteel,smoisupportrequiredsteel-↙
smoisupportprovidedsteel,smoisupportminimumsteel-smoisupportrequiredsteel,↙
smoisupportrequiredsteel-smoisupportmaximumsteel,sminresistanceshear-↙
sresistanceshear,ssfisupport-sresistanceshear,...
sactualspantodepthratio-sbasicspantodepthratio,baxisdistance-50,mbk-mbkdash,↙
mbrequiredsteel-mbprovidedsteel,mbminimumsteel-mbrequiredsteel,mbkes-mbkdashes,↙
mbrequiredsteeles-mbprovidedsteeles,mbminimumsteeles-mbrequiredsteeles,mbkis-↙
```

```
mbkdashis,...
mbminimumsteelis-mbrequiredsteelis,mbrequiredsteelis-mbprovidedsteelis,mbkespan-
mbkdashespan,mbminimumsteelespan-mbrequiredsteelespan,mbrequiredsteelespan-
mbprovidedsteelespan,mbsdesroww-mbsdesrowwlimit,mbsdesrequiredsteel-
mbsdesprovidedsteel,...
mbsdisleftroww-mbsdisleftrowwlimit,mbsdisleftrequiredsteel-mbsdisleftprovidedsteel,
mbsdisrightroww-mbsdisrightrowwlimit,mbsdisrightrequiredsteel-
mbsdisrightprovidedsteel,mbactualspantodepth-mblimitingratio,ebkes-ebkdashes,
ebminimumsteeles-ebrequiredsteeles,...
ebrequiredsteeles-ebprovidedsteeles,ebkis-ebkdashis,ebminimumsteelis-
ebrequiredsteelis,ebrequiredsteelis-ebprovidedsteelis,ebkispan-ebkdashispan,
ebminimumsteelispan-ebrequiredsteelispan,ebrequiredsteelispan-ebprovidedsteelispan,
ebkos-ebkdashos,...
ebminimumsteelos-ebrequiredsteelos,ebrequiredsteelos-ebprovidedsteelos,ebsdrow-
ebsdrowlimit,ebsdrequiredsteel-ebsdprovidedsteel,ecslenderness-ecslendernesslimit,
ectrr-ectpr,ec4rrs-ec4rps,ec34rs-ec34ps,ec23rs-ec23ps,ec12rs-ec12ps,ecg1rs-
ecg1ps,...
icslenderness-icslendernesslimit,icbgrs-icbgps,icg1rs-icg1ps,ic12rs-ic12ps,ic23rs-
ic23ps,ic34rs-ic34ps,ic4rrs-ic4rps,ictrr-ictpr,ccrs-ccps,175-x(23),10-
(100*mbrequiredsteelis/mbdeflectioneffectivebreadth),bxdirection-35,28-bxdirection,
bydirection-23,18-bydirection,...
ccnedratio-1,ccmedratio-1,sbrnmespan-140,sbrnsispan-250];

Ceq = [];


end
```