Calculate And Plot (Bauinformatik) erdbebeningenieurwesen

02-05-2019

Abdul Mannan

119189

Civil Engineering (NHRE),Msc

Shravani kumari pariseka

119483

Digital Engineering

Submitted To

Nathalie Dittrich

Bauhaus-Universität Weimar

Chair of Media Security

Bauhausstr. 11, R217

99423 Weimar,Germany

Acknowledgement

We are very grateful to our supervisor, **Nathalie Dittrich** whose advice and support led us to successfully accomplish this project.

Contents

Automatic Earthquake force calculation of 2D Frame

Ing.Abdul Mannan

abdul.mannan@uni-weimar.de

Abstract

With increasing development around the globe, the field of Civil engineering demands automation in their work to avoid big mistakes in the process of analysis and design. Software development in civil engineering is applied in the past but due to the fact that it is programmed by the non-engineers often lead to the problems in the results of the solution. To develop a software which will fulfil the exact requirements of the engineer, It is necessary for the engineer to write program of their software by their own .Keeping the same idea in view being an civil engineer ,a Java based software program is developed which calculate the behaviour of a Building after an earthquake .The program gives the forces developed in each story of a building after an earthquake. The main varying parameters of the building are no of stories, heights and loads.

1- Motivation

In civil engineering the problem we face as an engineer in the authencity of a result given by any software. To confirm the results of the software we usually do manual calculation of some of the task to match the results, which are time consuming and the chance of error is also greater. There are a lot of design challenges left to solve but most time of an engineer goes in doing the repetitive manual tasks.

The idea behind this project is to automate the engineer task, which gives an engineer more time to focus on solving real challenges .The result of this software is according to the exact need of an engineer so that they don't have to perform manual calculations

Goals

To apply the knowledge of civil engineering specifically earthquake engineering and develop a software according to our actual needs, what a engineer actually wants

The goal of this project is to

- Calculate the forces developed in a building when the earthquake occurs.
- To plot the curve of these calculated forces

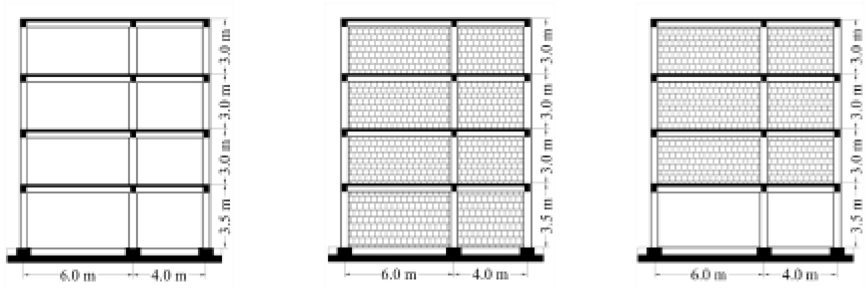
2- Concept

Assumptions of the project

- The masses of each floor is given.
- Horizontal seismic forces and its distribution over the height is calculated according to the UBC-97 (Uniform Building code) formulation
- Transfer of the system into a 2D numerical model means that the building is analysed as a 2D frame .

Table 1: Principle schemes of the system variations

Variations		
bare frame [BF]	with infill walls [FIW]	soft story [SS]



There are three different types of frame in the above Table.1 ,which are

- Bare frame
- Infill frame
- Soft Story

The user based on their matriculation number will get as a result the type of frame they have to use for the project from the software automatically

The above Table 1 gives us the parameter of height for the input of our software.

- Length of Bay 1=6.0m
- Length of Bay 2=4.0m
- Height Top=3.0m

- Height Bottom=3.5m
- Height Middle =3.0 m

Table 2: Applicable variations to the original frame elevation

No. [X]	Variation I: No. of Stories	No. [Y]	Variation II: - Arrangement of infill walls - material parameters	No. [Z]	Variation III: Set of time histories
0	3	0	RC: **lower** bound Infill: **weak** System: **FIW**	0	1
1	4	1	RC: **lower** bound Infill: **weak** System: **SS**	1	2
2	5	2	RC: **lower** bound Infill: **medium** System: **FIW**	2	3
3	6	3	RC: **lower** bound Infill: **strong** System: **FIW**	3	4
4	7	4	RC: **lower** bound Infill: **strong** System: **FIW**	4	5
5	7	5	RC: **upper** bound Infill: **weak** System: **FIW**	5	1
6	6	6	RC: **upper** bound Infill: **weak** System: **SS**	6	2
7	5	7	RC: **upper** bound Infill: **medium** System: **FIW**	7	3
8	4	8	RC: **upper** bound Infill: **strong** System: **FIW**	8	4
9	3	9	RC: **upper** bound Infill: **strong** System: **FIW**	9	5

Table 2 has 3 main parameters which I have programmed in software namely,

- X (The 3rd last digit of matriculation number)
  After entering X no of floors of building comes as a result
- Y (The 2nd last digit of matriculation number)
  After entering Y Frame type of building comes as a result

- Z (The last digit of the matriculation number)
  After entering Z time history data set # comes as a result

For example, if Matriculation number is 1 1 9 **1** **8** **9**

- X=1
- Y =8
- Z =9

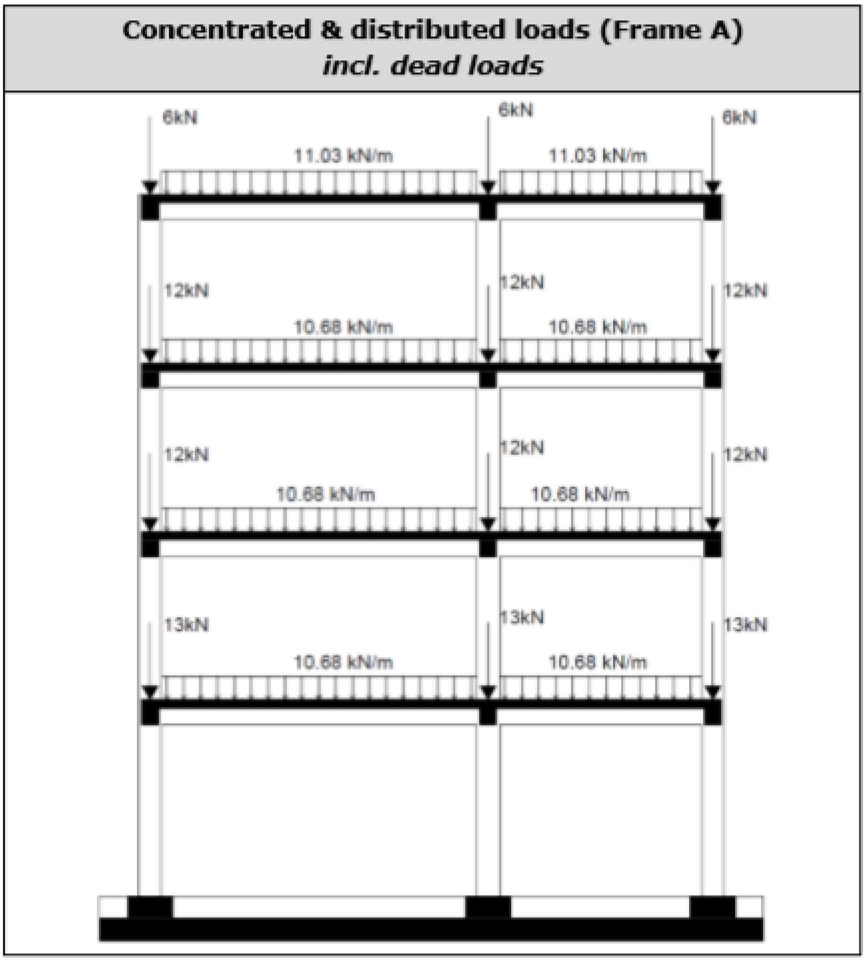


Figure 1 Frame description

Figure 1 gives us the information of loads (Weights) applied to the building as a dead load

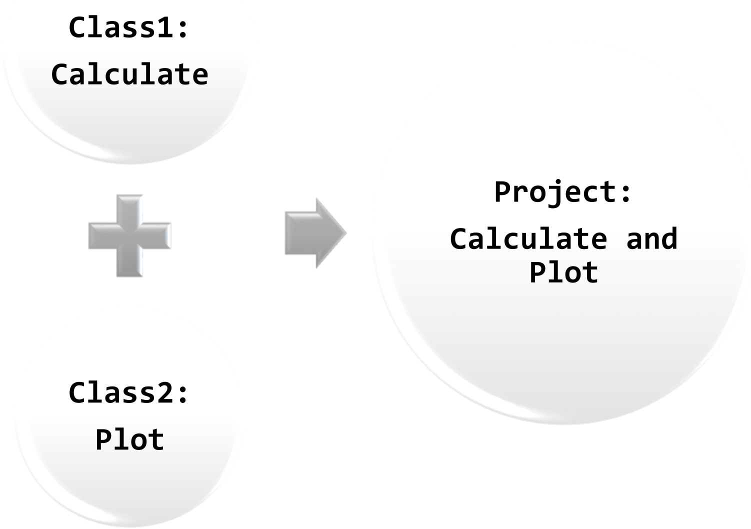
- Weight of top floor beam=11.03 kN/m
- Weight of ground floor beam=10.68 kN/m
- Weight of middle floor beam=10.68 kN/m
- Weight of top floor column=6kN
- Weight of ground floor column=12kN
- Weight of middle floor column=13kN

3-Implementation

The equation and method implementation is based on Uniform building code(UBC-97) Seismic provisions

From Chapter 16 STRUCTURAL DESIGN REQUIREMENTS

Section 1601 :1630.2 Static Force Procedure.



Method 1: propX (int z)

This method is based on X values in Table 2. At first, I use if else condition which is a problem as I have to write a lot then I got the idea of switch case. In this method I have used the concept of switch case instead of if else condition. This method takes X value as integer argument and return the integer value which is the number of floors of the building.

Method 2: propY (int y)

This method is based on Y values in Table 2. At first, I use if else condition which is a problem as I have to write a lot then I got the idea of switch case. In this method I have used the concept of switch case instead of if else condition. This method takes Y value as integer argument and return the String which is the type of building (2D frame)

Method 3: propZ (int z)

This method is based on Z values in Table 2. At first, I use if else condition which is a problem as I have to write a lot then I got the idea of switch case. In this method I have used the concept of switch case instead of if else condition. This method takes Z value as integer argument and return the integer value which is the Time history data set number.

Method 4: weightTotal(int x,double [] inpVal)

This method takes argument as X value as integer and array of input values and return the total weight of whole building after performing some calculations.

Method 5: weightTop(double [] inpVal)

This method takes argument as an array of input values and return the weight of top floor of building after performing some calculations.

Method 6: weightBot(double [] inpVal)

This method takes argument as an array of input values and return the weight of bottom (Ground) floor of building after performing some calculations.

Method 7: weightMid(int x,double [] inpVal)

This method takes argument as an array of input values and return the weight of middle floor of building after performing some calculations.

Method 8: baseShearCoff(int x,double [] inpVal)

This method takes argument as X value as integer and array of input values and return the base Shear coefficient of whole building after performing some calculations

Method 9: timePeriod(int x,double [ ] inpVal)

This method takes argument as X value as integer and array of input values and return the Time period of whole building after performing some calculations

Method 10: heightCal(int x,double [ ] inpVal)

This method takes argument as X value as integer and array of input values and return the height of whole building after performing some calculations

Method 11: baseShear(int x,double [ ] inpVal)

This method takes argument as X value as integer and array of input values and return the base shear of building after performing some calculations. I used if else condition for taking some decision.

Method 12: comFt(int x,double [ ] inpVal)

This method takes argument as X value as integer and array of input values and return the factor Ft after performing some calculations. I used if else condition for taking some decision.

Method 13: sumOfFnWei(int x,double [ ] inpVal)

This method takes argument as X value as integer and array of input values and return the sum of weight which is cumulative after performing some calculations. I used if else condition for taking some decision and by using for loop for iteration.

Method 14: wxhx(int x,double [] inpVal)

This method takes argument as X value as integer and array of input values and return the sum of weight of each middle floor it is also cumulative after performing some calculations. I used if else condition for taking some decision and by using for loop for iteration.

Method 15: wxhxGnT(int x,double [] inpVal)

This method takes argument as X value as integer and array of input values and return the sum of weight of ground and top floor, it is also cumulative after performing some calculations. I used if else condition for taking some decision and by using for loop for iteration.

Method 16 : vMinFt(int x,double [] inpVal)

This method takes argument as X value as integer and array of input values and return the factor vMinFt, after performing some calculations.

Method 17 : fXmid(int x,double [] inpVal)

This method takes argument as X value as integer and array of input values and return Story shear force of each middle floor developed as a result of earthquake after performing some calculations. I used if else condition for taking some decision and by using for loop for iteration. This method returns an array of values stored according to the varying no of floors of building

Method 18 : fXtg(int x,double [ ] inpVal)

This method takes argument as X value as integer and array of input values and return Story shear force of top and ground floor developed as a result of earthquake after performing some calculations. I used if else condition for taking some decision and by using for loop for iteration. This method returns an array of values stored according to the varying no of floors of building.

Method 19: graphPlot(double [] data,int x)

This method takes input the value of X and array of data values (Story shear forces which is the output of calculation) coming after performing calculations. JFreeChart is used in this method for plotting the graph.

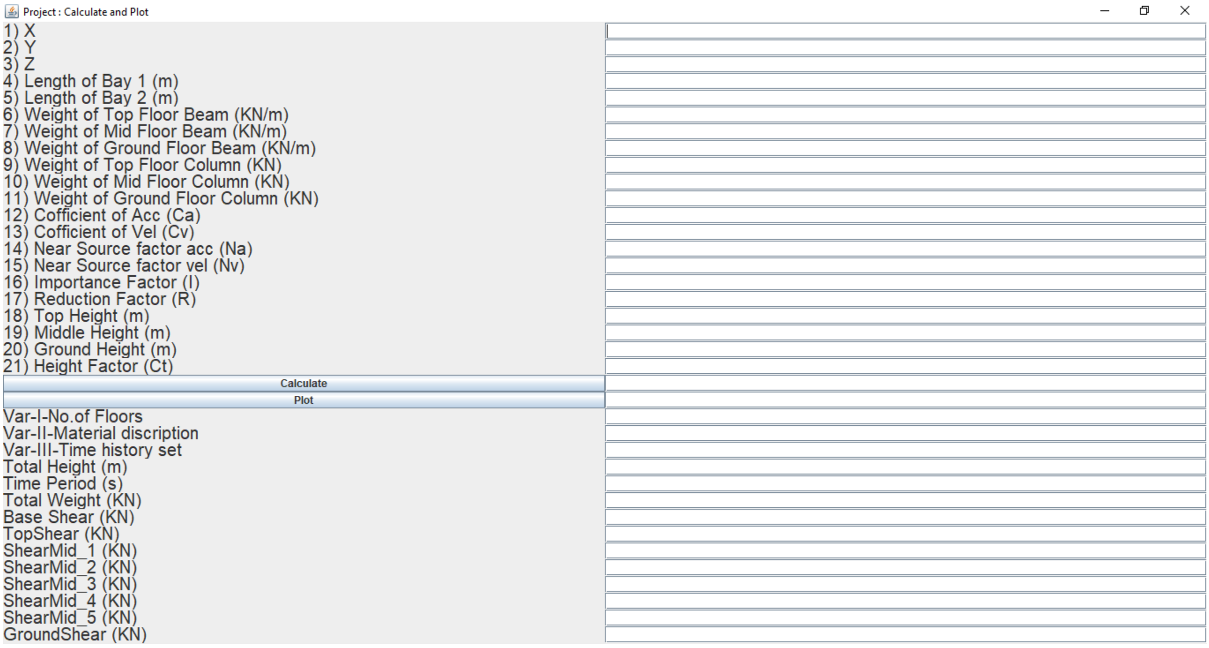
Problems

- As the no of floors of the building keeps on changing by which the calculation for whole of the building is behaving in different way that’s the big challenge I have occurred. For that I have divide my problem into different sub methods which solve my problem.

- As when the no of floors changes the size of return array is varying so I got the index out of bounds error, I solves that error by using the some of the conditions.
- The problem to do test cases as my all methods are taking the arguments from user input in Gui but in test cases I need the values .So I have put all my input values in an array and I have given that array as parameter to my methods .Then I will be able to run and passed the test cases .
- The graph plotting is also a problem as the values are not passing correctly .I created a different class for graph “Plot” and then call it to my “Calculate” program for plotting .

USER GUIDE

It contain 21 input parameter from 1 to 21 which a user have to enter



1)X ,2) Y and 3) Z should be taken from the table below

Table 2: Applicable variations to the original frame elevation

No. [X]	Variation I: No. of Stories	No. [Y]	Variation II: - Arrangement of infill walls - material parameters	No. [Z]	Variation III: Set of time histories
0	3	0	RC: **lower** bound Infill: **weak** System: **FIW**	0	1
1	4	1	RC: **lower** bound Infill: **weak** System: **SS**	1	2
2	5	2	RC: **lower** bound Infill: **medium** System: **FIW**	2	3
3	6	3	RC: **lower** bound Infill: **strong** System: **FIW**	3	4
4	7	4	RC: **lower** bound Infill: **strong** System: **FIW**	4	5
5	7	5	RC: **upper** bound Infill: **weak** System: **FIW**	5	1
6	6	6	RC: **upper** bound Infill: **weak** System: **SS**	6	2
7	5	7	RC: **upper** bound Infill: **medium** System: **FIW**	7	3
8	4	8	RC: **upper** bound Infill: **strong** System: **FIW**	8	4
9	3	9	RC: **upper** bound Infill: **strong** System: **FIW**	9	5

4) Length of Bay 1 (m)=6

5) Length of Bay 2 (m)=4

6) Weight of Top Floor Beam (KN/m)=11.03

7) Weight of Mid Floor Beam (KN/m) =10.68

8) Weight of Ground Floor Beam (KN/m)= 10.68

9) Weight of Top Floor Column (KN) =6

10) Weight of Mid Floor Column (KN) =12

11) Weight of Ground Floor Column (KN) =13

12) Cofficient of Acc (Ca)

TABLE 16-Q—SEISMIC COEFFICIENT C_a

SOIL PROFILE TYPE	SEISMIC ZONE FACTOR, Z				
	Z = 0.075	Z = 0.15	Z = 0.2	Z = 0.3	Z = 0.4
S_A	0.06	0.12	0.16	0.24	$0.32N_a$
S_B	0.08	0.15	0.20	0.30	$0.40N_a$
S_C	0.09	0.18	0.24	0.33	$0.40N_a$
S_D	0.12	0.22	0.28	0.36	$0.44N_a$
S_E	0.19	0.30	0.34	0.36	$0.36N_a$
S_F	See Footnote 1				

[1]Site-specific geotechnical investigation and dynamic site response analysis shall be performed to determine seismic coefficients for Soil Profile Type S_F.

13) Cofficient of Vel (Cv)

TABLE 16-R—SEISMIC COEFFICIENT C_v

SOIL PROFILE TYPE	SEISMIC ZONE FACTOR, Z				
	Z = 0.075	Z = 0.15	Z = 0.2	Z = 0.3	Z = 0.4
S_A	0.06	0.12	0.16	0.24	$0.32N_v$
S_B	0.08	0.15	0.20	0.30	$0.40N_v$
S_C	0.13	0.25	0.32	0.45	$0.56N_v$
S_D	0.18	0.32	0.40	0.54	$0.64N_v$
S_E	0.26	0.50	0.64	0.84	$0.96N_v$
S_F	See Footnote 1				

[1]Site-specific geotechnical investigation and dynamic site response analysis shall be performed to determine seismic coefficients for Soil Profile Type S_F.

14) Near Source factor acc (Na)

TABLE 16-S—NEAR-SOURCE FACTOR N_a[1]

SEISMIC SOURCE TYPE	CLOSEST DISTANCE TO KNOWN SEISMIC SOURCE[2,3]		
	≤ 2 km	5 km	≥ 10 km
A	1.5	1.2	1.0
B	1.3	1.0	1.0
C	1.0	1.0	1.0

[1]The Near-Source Factor may be based on the linear interpolation of values for distances other than those shown in the table.

[2]The location and type of seismic sources to be used for design shall be established based on approved geotechnical data (e.g., most recent mapping of active faults by the United States Geological Survey or the California Division of Mines and Geology).

[3]The closest distance to seismic source shall be taken as the minimum distance between the site and the area described by the vertical projection of the source on the surface (i.e., surface projection of fault plane). The surface projection need not include portions of the source at depths of 10 km or greater. The largest value of the Near-Source Factor considering all sources shall be used for design.

15) Near Source factor acc (Nv)

TABLE 16-T—NEAR-SOURCE FACTOR N_v[1]

SEISMIC SOURCE TYPE	CLOSEST DISTANCE TO KNOWN SEISMIC SOURCE[2,3]			
	≤ 2 km	5 km	10 km	≥ 15 km
A	2.0	1.6	1.2	1.0
B	1.6	1.2	1.0	1.0
C	1.0	1.0	1.0	1.0

[1]The Near-Source Factor may be based on the linear interpolation of values for distances other than those shown in the table.

[2]The location and type of seismic sources to be used for design shall be established based on approved geotechnical data (e.g., most recent mapping of active faults by the United States Geological Survey or the California Division of Mines and Geology).

[3]The closest distance to seismic source shall be taken as the minimum distance between the site and the area described by the vertical projection of the source on the surface (i.e., surface projection of fault plane). The surface projection need not include portions of the source at depths of 10 km or greater. The largest value of the Near-Source Factor considering all sources shall be used for design.

16) Importance Factor (I)

TABLE 16-K—OCCUPANCY CATEGORY

OCCUPANCY CATEGORY	OCCUPANCY OR FUNCTIONS OF STRUCTURE	SEISMIC IMPORTANCE FACTOR, I	SEISMIC IMPORTANCE[1] FACTOR, I_p	WIND IMPORTANCE FACTOR, I_w
1. Essential facilities[2]	Group I, Division 1 Occupancies having surgery and emergency treatment areas Fire and police stations Garages and shelters for emergency vehicles and emergency aircraft Structures and shelters in emergency-preparedness centers Aviation control towers Structures and equipment in government communication centers and other facilities required for emergency response Standby power-generating equipment for Category 1 facilities Tanks or other structures containing housing or supporting water or other fire-suppression material or equipment required for the protection of Category 1, 2 or 3 structures	1.25	1.50	1.15
2. Hazardous facilities	Group H, Divisions 1, 2, 6 and 7 Occupancies and structures therein housing or supporting toxic or explosive chemicals or substances Nonbuilding structures housing, supporting or containing quantities of toxic or explosive substances that, if contained within a building, would cause that building to be classified as a Group H, Division 1, 2 or 7 Occupancy	1.25	1.50	1.15
3. Special occupancy structures[3]	Group A, Divisions 1, 2 and 2.1 Occupancies Buildings housing Group E, Divisions 1 and 3 Occupancies with a capacity greater than 300 students Buildings housing Group B Occupancies used for college or adult education with a capacity greater than 500 students Group I, Divisions 1 and 2 Occupancies with 50 or more resident incapacitated patients, but not included in Category 1 Group I, Division 3 Occupancies All structures with an occupancy greater than 5,000 persons Structures and equipment in power-generating stations, and other public utility facilities not included in Category 1 or Category 2 above, and required for continued operation	1.00	1.00	1.00
4. Standard occupancy structures[3]	All structures housing occupancies or having functions not listed in Category 1, 2 or 3 and Group U Occupancy towers	1.00	1.00	1.00
5. Miscellaneous structures	Group U Occupancies except for towers	1.00	1.00	1.00

[1]The limitation of I_p for panel connections in Section 1633.2.4 shall be 1.0 for the entire connector.
[2]Structural observation requirements are given in Section 1702.
[3]For anchorage of machinery and equipment required for life-safety systems, the value of I_p shall be taken as 1.5.

17) Reduction Factor (R) =1.0

18) Top Height (m)= 3

19) Middle Height (m) = 3

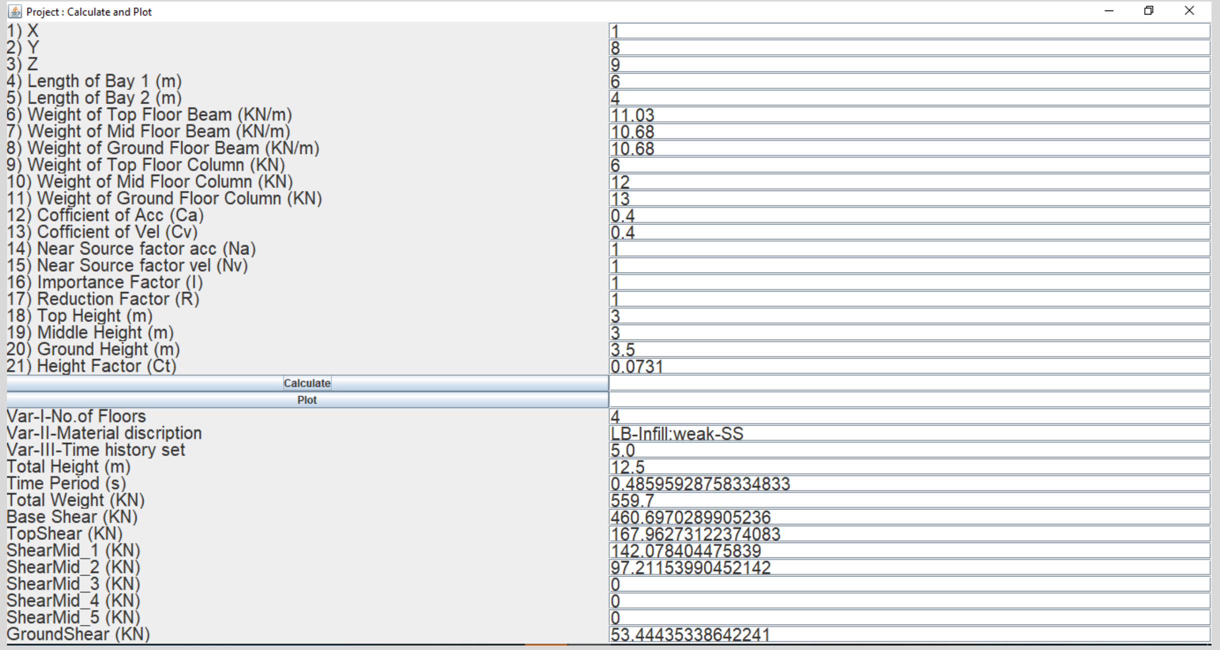
20) Ground Height (m)= 3.5

21) Height Factor (Ct)

C_t = 0.035 (0.0853) for steel moment-resisting frames.
C_t = 0.030 (0.0731) for reinforced concrete moment-resisting frames and eccentrically braced frames.
C_t = 0.020 (0.0488) for all other buildings.

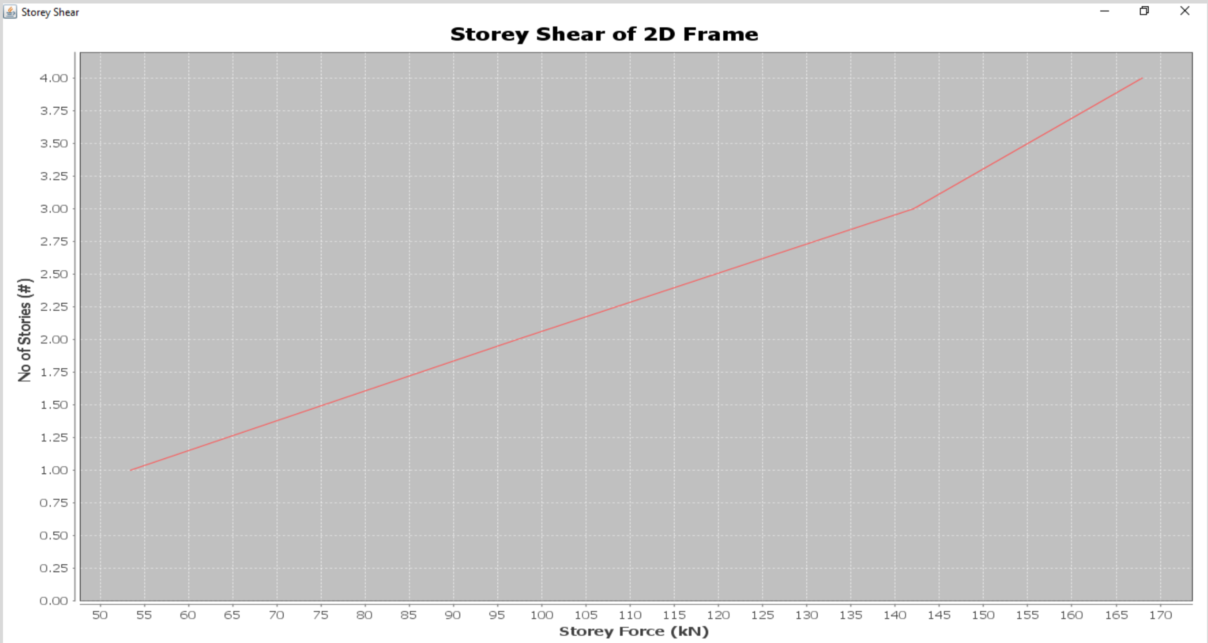
Then click calculate button and then plot button to plot the graph

Example :Fill up form : After filling the form press calculate button



Then click plot button

The graph will be plotted



4-Conclusion

1- Yes, at last We have able to solve all challenges .This project teaches us a lot of things .This project give our path a way to follow .

References

- UBC-97,Uniform building code
- https://docs.oracle.com/javase/8/docs/
- https://www.youtube.com/watch?v=cw31L_OwX3A&list=LLRkVz3oUd1GQ1KNwXBC_NqA&index=3&t=1844s