RESEARCH ARTICLE

Development of preliminary structural analysis software

Ö.H. Bettemir*[ID], K. Türk[ID]

İnönü University, Department of Civil Engineering, Malatya, Turkey

Abstract

In this study, structural analysis software is developed to analyze buildings by slope-deflection method with minimum data entry requirement. It is adequate to define the dimensions of the columns and beams for one floor and the developed software replicates the assigned dimensions for the remaining floors. Similarly, vertical loads on the beams are also replicated for each floor. Aforementioned design approach expedites the data entry process without affecting the reliability of the analysis for preliminary design stage. Structural analysis software forms the slope-deflection equations and computes the rotations of the nodes and horizontal displacements of floors by matrix inversion. The software gives feedback on the suitability of the dimensions of the beams and columns by considering the obtained moments, shear forces, and normal forces. The designer can update the dimensions of the columns as well as beams and repeat the analysis until the dimensions are optimized. Henceforth the structure can be analyzed on robust structural analysis software with a few modifications. This approach would save important amount of time and work hours at design offices. Because state-of-the art building information modeling software require many attribute data about each structural elements and materials. Therefore, updating any section of the structure may cause revision of the attribute data as well. Developed software is tested on 20 floor structure with four spans. Utilization of a simple design approach decreases the allocation of robust design software and reduces the required number of structural analysis software for the design offices. In addition to this, the structural analysis software is freeware and can be used by civil engineering students for validation of their solutions.

Keywords

1. Introduction

In the beginning of the previous century Maney developed slope-deflection equations for the analysis of frame structures [1]. The proposed method contains many assumptions and simplifications in order to reduce the computational demand. The simplifications enabled analysis of a structure by hand computations. The method was implemented for the analysis of structure until the invention of the computer, although it requires matrix inversion. Ignorance of strain of structural elements under normal and shear stress prevent obtaining robust solutions However, introducing the aforementioned effects into the equilibrium equations increases both the number of unknowns, and the size of the matrix to be inverted. Effects of deflections caused by shear and normal forces are relatively low when compared with the deflections of the beams caused by bending moments. However, invention of computers annihilated the

---

* Corresponding author
  E-mail: onder.bettemir@inonu.edu.tr

necessity of simplifications and the popularity of slope-deflection equation decreased.

State-of-the art structural analysis software can perform the structural analysis of any type of structure. However, as the software become sophisticated, the users should have knowledge on the implemented solution algorithms. Main problem of the computerized structural analysis is the differences between the real structure and the abstracted structure which is defined to the structural analysis software. Another problem is that robust structural analysis ends up with very large matrixes to be inverted. The computational capabilities of the computers are high enough to invert the matrixes but inversion of large matrixes increases the probability of ill-conditioned equation system. If the user does not have knowledge on the numerical methods, adverse consequences of ill-conditioned problem may not be recognized. On the other hand, utilization of structural analysis software is mandatory because design and analysis of a structure require significant computations.

Off the shelf structural analysis software are very detailed and complex as their capabilities are improved step by step as new versions are released. Building Information Modeling properties are added to structural analysis software at each version update. Therefore, in addition to definition of nodes, structural elements, support conditions, and details of the structural utilities are also defined simultaneously. Defining nodes, structural elements, mechanical properties of the structural elements and the loads take significant amount of time. During the preliminary design stage many changes are applied to the design and each change is updated on the abstracted design. Revision of the position and dimensions of the structural elements can be difficult and time consuming. This is because after changing the span of the axes, collision of pipes or ventilation system may occur. Therefore, small changes may end up with important design revisions.

Robust structural analysis is necessary when the architectural and mechanical design of the structure is finalized. In this study, preliminary structural design software is developed in order to obtain approximate dimensions of the structural frame by entering very few data. Structural analysis software utilizes slope-deflection equations which can be written in matrix form. In addition to this, the abstracted structure is simplified so that dimensions of the beams, columns, and slabs are defined for one floor and the remaining floors are replicated. As a result, properties of the structure are defined very easily. Data entry process involves only entering a few numbers on a text file. In addition to this computation time is less than one second even for high structures. Implementation of slope-deflection method reduces the analysis time compared by finite-element methods. On the other hand, slope-deflection method is not as accurate as finite element methods but the accuracy of slope-deflection is adequate for preliminary structural analysis.

Fazio and Gowri investigated many structural analysis software [2]. Ha (1990) also examined the interchangeability of C and Fortran languages for computer programming of finite element methods [3]. Sanal compared C and Fortran languages based on memory allocation and mentioned that C programming language is superior because of its dynamic memory allocation capability [4]. Gu developed a finite element analysis software with C programming language [5]. Development of a structural analysis computer program is not an achievement because there are abundant of off the shelf commercial structural analysis programs. Some of the notable commercial programs can be given as 1•2•Build which is a structural analysis software for the predesign of two-dimensional frame structures. EngiSSol is and structural analysis software which can perform 3D analysis. ADINA Structures is a finite element analysis program. ADAPT-Floor-Pro is a reinforced and post-tensioned concrete slab design software, which can analyze buildings in 3D. Advance Design software utilizes finite element method for structural analysis with Building Information Modeling (BIM). ArchiCAD is also 3D structural analysis program with BIM implementation. ETABS, SAP2000, ProBina, and Sta4CAD are also

well known commercial structural analysis programs.

In the second section theory of the slope-deflection equations are briefly introduced. Moreover, implementation of slope-deflection equations on C++ programming language is illustrated. Case study problems are defined and solutions are provided in the third section. Finally, the results are discussed and concluded.

2. Theory

Slope-deflection equations are derived from virtual load method by Maney [1]. In this method, deflections of beams and columns are expressed in terms of member end rotations. Member-end moments are computed according to the member-end rotations. Member-end rotations of the structural element *AB* are represented by θ_A and θ_B at points A and B respectively (Fig. 1). Ψ_{AB} represents the element chord rotations caused by side-sway or settlement. In Fig. 1 relative vertical displacement between point A and B is represented by Δ_{AB} so that the chord rotation of the structural element is equal to the relative displacement per length of the structural element. If the definition is generalized, the chord rotation can be written as in Eq. 1.

$$\Psi_{ij} = \frac{\Delta_{ij}}{L_{ij}} \tag{1}$$

where, *i*, *j* denotes the start and end nodes of the member *ij*, L_{ij} represents the length of the member and Δ_{ij} represents the relative displacement

between the end nodes of the member. Slope deflection equations contain many simplifications to reduce the computational demand. Similarly, the chord rotation formula is the simplified case of rotation angle of bended element. Therefore, Eq. 1 is valid for very small displacements.

Generalized member-end moments for the situation given in Figure 1 can be written as in Eq. 2.

$$M_{ij} = 2E\frac{I_{ij}}{L_{ij}}\left\{2\theta_i + \theta_j - 3\Psi_{ij}\right\} + FEM_{ij}$$

$$M_{ji} = 2E\frac{I_{ij}}{L_{ij}}\left\{\theta_i + 2\theta_j - 3\Psi_{ij}\right\} + FEM_{ji} \tag{2}$$

where, *E* is the elastic modulus, I_{ij} is the moment of inertia, L_{ij} is the length of the structural element. FEM_{ij} represents the fixed end moments caused by the loads acting on the member *ij*. Derivation of Eq. 2 and FEM can be obtained from a textbook [6, 7]. Sign convention of member-end moments given in Eq. 2 is represented in Fig. 2. Sign conventions of member-end moments are not compatible with the conventional sign convention. The reason of this discrepancy is to form equilibrium equations with dominant coefficient of the rotation angle of the corresponding node.

If sign conventions of member-end moments shown in Fig. 3 were compatible with conventional sign convention, at point C the moment equation would be written as $M_{CB} + M_{CE} - M_{CD} = 0$. However, in Fig. 3 when the member-end moments are considered it is seen that all of the member-end moments at the joint are counter clockwise. This property makes construction of equilibrium equations straightforward.
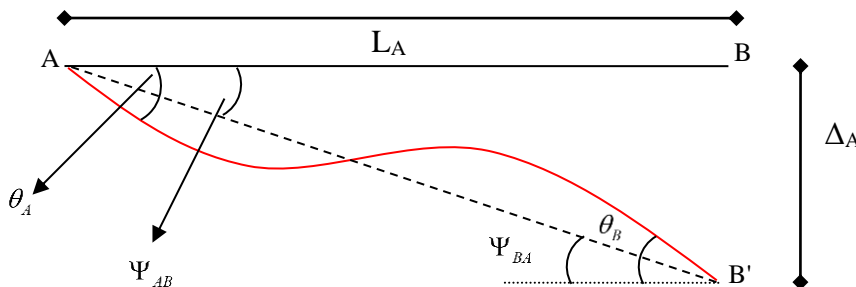

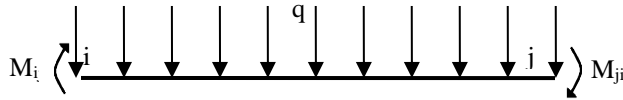**Fig. 1.** Member-end rotations according to slope-deflection method

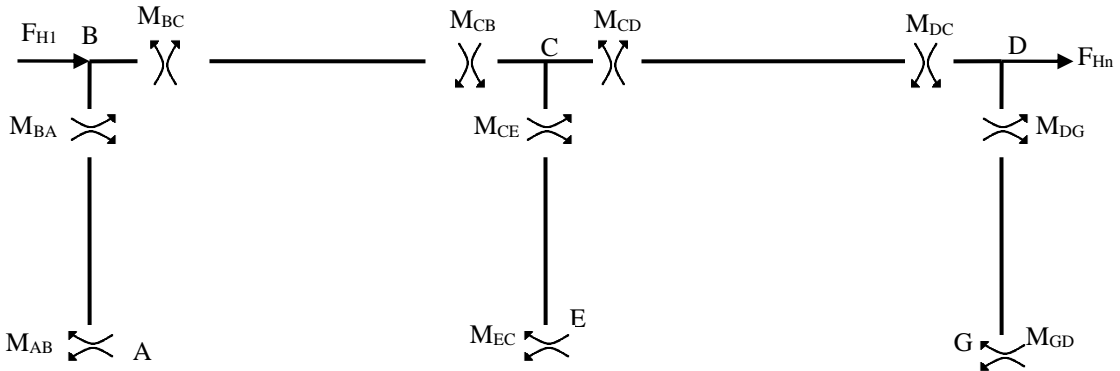**Fig. 2.** Sign convention of member-end moments

Fig. 3. Member-end moments at a frame structure

Summation of the member-end moments have to be zero and $M_{CB} + M_{CE} + M_{CD} = 0$ is written for node C, since the joint is under equilibrium. Alternative sign convention makes the automated generation of equilibrium equations by a computer simple. In addition to its simplicity, the diagonals of the coefficient matrix of the slope deflection equation become dominant. Dominance of diagonals accelerates the computation of matrix inverse by numerical methods. In addition to this, occurrence of an ill-conditioned system is prevented.

Fig. 3 represents a typical joint of a frame structure where beams and columns join. Equilibrium conditions are written for each joint and the joint rotations. Number of equilibrium equations becomes equal to the number of unknown joint rotations. If side-sway is not prevented there would be additional unknowns related with the side-sway of the structure. Each floor might sway different amount; therefore side-sway of each floor becomes an unknown to be solved. Additional equilibrium conditions should be written in order to obtain enough number of equations. Side-sway is caused by the shear forces at the columns of the structure. Therefore, additional equations are obtained by equilibrium conditions of the columns. Fig. 4 represents the free body diagram of the columns of the structure given in Fig. 3.

Moment equilibrium of column CE at point C is written as (Fig. 4);

$$\circlearrowleft + \sum M_C = 0 \quad M_{CE} + M_{EC} + hH_E = 0$$

$$H_E = -\frac{M_{CE} + M_{EC}}{h}$$

Moment equilibrium of column BA at point B is written as (Figure 4);

$$\circlearrowleft + \sum M_B = 0 \quad M_{AB} + M_{BA} + hH_A = 0$$

$$H_A = -\frac{M_{AB} + M_{BA}}{h}$$

Moment equilibrium of column DG at point D is written as (Figure 4);

$$\circlearrowleft + \sum M_D = 0 \quad M_{DG} + M_{GD} + hH_G = 0$$

$$H_G = -\frac{M_{DG} + M_{GD}}{h}$$

H   horizontal force equilibrium is written as;

$$\overset{+}{\rightarrow}\sum F_X = 0 \qquad H_E + H_A + H_G + \sum F_H = 0 \qquad (3)$$

where $\sum F_H$ is the total horizontal force acting at the corresponding floor and its upper floors.
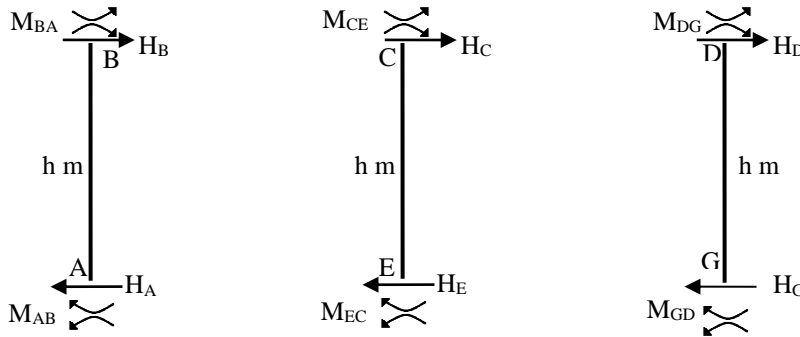
Fig. 4. Free body diagram of the columns

If there are n columns at one floor, for the jth floor the horizontal force equilibrium condition can be generalized as given in Eq. 4.

$$\overset{+}{\rightarrow}\sum F_X = 0 \qquad \sum_{i=1}^{n} H_i + \sum F_{Hj} = 0 \qquad (4)$$

where, $\sum F_{Hj}$ is the total horizontal force at the jth and the upper floors, and H_i is the shear force at the bottom section of the ith column. H_i is substituted by moment equilibrium written for the ith column. If the end nodes of the ith column is represented as i and i', then shear force at the bottom of the ith column, H_i can be written as given in Eq. 5.

$$H_i = -\frac{M_{ii'} + M_{i'i}}{h} \qquad (5)$$

Eq. 4 can be written as shown below, when eq. 5 is substituted.

$$\overset{+}{\rightarrow}\sum F_X = 0 \quad \sum_{i=1}^{n}\left(-\frac{M_{ii'} + M_{i'i}}{L_i}\right) + \sum F_{Hi} = 0 \quad (6)$$

Eq.6 is implemented for each floor of the frame structure and adequate equations are formed to solve the unknown side-sways of each floor. When the equations are written in systematic order the matrix equation given in Eq. 7 is obtained.

XC = B   (7)

where X represents the coefficients of the rotation angles of the joints and side-sway of the floors, C represents the unknown parameters, B represents the constant terms which are obtained by the summation of FEM at the joints of the beams and

horizontal forces at the joints of the columns. In order to compute the unknown parameters, X matrix should be inverted and multiplied by the constant terms. Matrix inversion is computed by Gauss-Jordan method [8]. The matrix inversion can be computed as long as the structure is stable [9].

3. Development of software

Computer codes are generated by C++ which can form slope-deflection equations without human intervention. The flowchart of the codes is given in Fig. 5. Initially member-end moments of the columns and beams are computed by Eq. 2. If the structural element is a beam then Ψ angle becomes the relative rotation caused by the settlement of the foundations. On the other hand, if the structural element is a column, Ψ angle becomes the rotation angle caused by the side-sway of the corresponding floor.

Rotations at end of the beams are caused by relative settlement of the foundations. Rotations of the beams are computed by Eq. 8 which is particular form of Eq. 1 in which the displacements are caused by settlements of the footings of the foundations.

$$\Psi_{ij} = \frac{\delta_j - \delta_i}{L_{ij}} \qquad (8)$$

In Eq. 8 δ_i and δ_j represent the settlement of the foundation i and j respectively. L_{ij} is the span of the beam between the nodes i and j. C++ code for the implementation of equilibrium equation for the beams is given below;

```
for (i=0; i < floorNumber; i++) {
        for (j=0; j < axeNumber-1; j++) {
        beamMoment[i*(axeNumber-1)*2+j*2][i*axeNumber+j] = 2.0*EConcrete*IBeam[j]/ axSpan [j]*2.0;
        beamMoment[i*(axeNumber-1)*2+j*2][i*axeNumber+j+1] = 2.0*EConcrete*IBeam[j]/ axSpan [j];
        beamFEM[i*(axeNumber-1)*2+j*2]=-FEM[j][0]+6.0*EConcrete*IBeam[j]*rot[j]/axSpan[j]/axSpan[j];

        beamMoment[i*(axeNumber-1)*2+j*2+1][i*axeNumber+j+1] = 2.0*EConcrete*IBeam[j]/axSpan[j]*2.0;
        beamMoment[i*(axeNumber-1)*2+j*2+1][i*axeNumber+j] = 2.0*EConcrete*IBeam[j]/ axSpan [j];
        beamFEM[i*(axeNumber-1)*2+j*2+1]=-FEM[j][1]+6.0*EConcrete*IBeam[j]*rot[j]/axSpan[j]/axSpan[j];
                }
        }
```
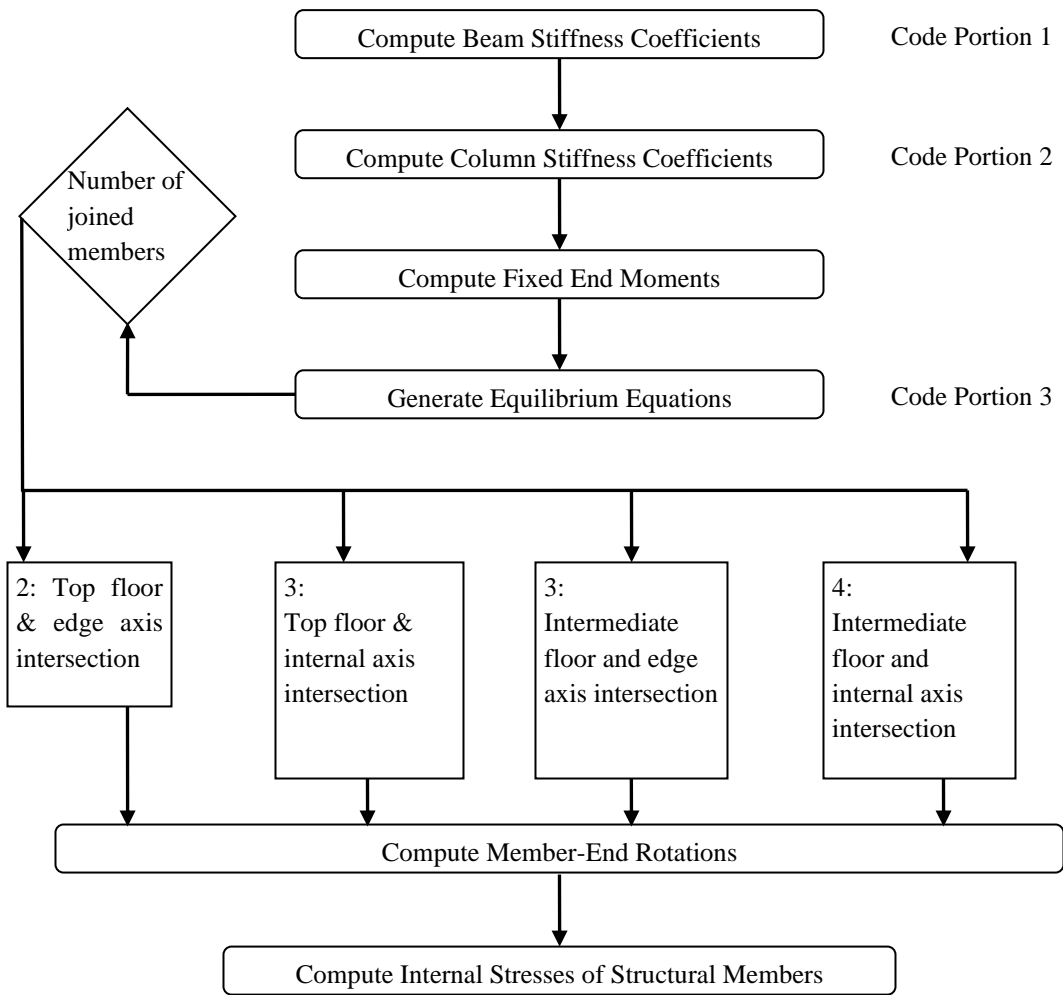

Figure 5. Flowchart of developed software

The computer code computes the coefficients of θ_i and θ_j angles for the beams. Similar code is written for the columns as well. Then the moment equilibrium conditions for the joints are programmed by the following C++ code. Computer code, given above, implements Eq. 2 to the beams.

```
for (i=0; i < floorNumber; i++) {
        for (j=0; j < axNumber; j++) {
                if(i==floorNumber-1) { if (j==0) {
for(k=0;k<floorNumber*(axNumber+1);k++) Xmat[i*axNumber+j][k] = beamMoment[i*(axNumber-1)*2][k] +
colMoment[i*axNumber*2][k];
const[i*axNumber+j] = beamFEM[i*(axNumber-1)*2]; }
else if(j==axNumber-1) {
for(k=0;k<floorNumber*(axNumber+1);k++) Xmat[i*axNumber+j][k] = beamMoment[i*(axNumber-1)*2+2*j-
1][k] + colMoment[i*axNumber*2+2*j][k];
const [i*axNumber+j] = beamFEM[i*(axNumber-1)*2+2*j-1];
                                } else {
for(k=0; k<floorNumber*(axNumber+1); k++) Xmat[i*axNumber+j][k] = beamMoment[i*(axNumber-1)*2+2*j-
1][k] + beamMoment[i*(axNumber-1)*2+2*j][k] + colMoment[i*axNumber*2+2*j][k];
const [i*axNumber+j] = beamFEM[i*(axNumber-1)*2+2*j-1] + beamFEM[i*(axNumber-1)*2+2*j]; } } else {
        if (j==0) {
for(k=0;k<floorNumber*(axNumber+1);k++) Xmat[i*axNumber+j][k] = beamMoment[i*(axNumber-1)*2][k] +
colMoment[i*axNumber*2][k] + colMoment[(i+1)*axNumber*2+1][k];
const [i*axNumber+j] = beamFEM[i*(axNumber-1)*2];
                } else if(j==axNumber-1) {
for(k=0;k<floorNumber*(axNumber+1);k++) Xmat[i*axNumber+j][k] = beamMoment[i*(axNumber-1)*2+2*j-
1][k] + colMoment[i*axNumber*2+2*j][k] + colMoment[(i+1)*axNumber*2+2*j+1][k];
const [i*axNumber+j] = beamFEM[i*(axNumber-1)*2+2*j-1];
                                } else {
for(k=0; k<floorNumber*(axNumber+1); k++) Xmat[i*axNumber+j][k] = beamMoment[i*(axNumber-1)*2+2*j-
1][k]      +      beamMoment[i*(axNumber-1)*2+2*j][k]      +      colMoment[i*axNumber*2+2*j][k]      +
colMoment[(i+1)*axNumber*2 + 2*j+1][k];
const [i*axNumber+j] = beamFEM[i*(axNumber-1)*2+2*j-1] + beamFEM[i*(axNumber-1)*2+2*j];
                                        }        }        }        }
```

Computer code, given above, implements moment equilibrium equation to all of the joints. The code scans all of the floors and axes of the building. Number of floors and axes is specially treated since at the top floor there will not be columns joining from the upper floor. Similarly, at the first axis and the last axis there will not be any beams spanning from neighboring axis. Shear force equilibrium is formed for each floor by the following C++ code.

```
for (i=0; i < floorNumber; i++) {
for(k=0; k< floorNumber *( axNumber +1); k++) Xmat[floorNumber *axNumber + i][k] = 0.0;
for (j=0; j < axNumber; j++) {

for(k=0; k< floorNumber *( axNumber +1); k++) Xmat[floorNumber * axNumber + i][k] += colMoment[i*
axNumber *2+2*j][k] + colMoment [i* axNumber *2+2*j+1][k];
const[floorNumber * axNumber + i] = -floorHght * cumHorForce[i];
                }
    }
```

Obtained matrix is inverted and member-end moments, shear forces and normal forces are computed.

4. Case studies

Developed software is tested on five test problems. Test problems are sorted from the simplest frame structure to more complicated structure. Test problems are defined in order to represent the easiness of the definition of the frame structure. In addition to this, fast solution of the test problem is another advantage of the algorithm. Test problems are solved on one core of 1.6 GHz i5-4200U CPU.

4.1. Case problem 1

First case study problem consists of one storey, one span frame structure. Loads and support conditions are given in Fig. 6. In addition to this foundation A and D are settled 1 and 2 cm respectively. The frame structure is reinforced concrete with 25 MPa compressive strength. Modulus of elasticity of the concrete is taken as 30.0 GPa.

Data defining the structure and loading conditions are entered as text file. Required data for the definition of the first case problem illustrated in Fig. 6 is given below.

```
2 1 25.0 420.0 30.0 3.0
1 2
60.0
5.0
25.0 60.0 25.0 60.0
60.0 25.0
30.0
```

In the first row of data file, numbers of axis and floor, characteristic compressive strength of concrete in MPa, characteristic strength of steel in MPa, elastic modulus of concrete in GPa and height of floor in meter are represented. Settlements of foundations are defined in the second row in centimeters. Magnitude of uniformly distributed load is entered at the third row. Span between the axes is defined at the fourth row, while dimensions of the columns and beams are defined at the fifth and sixth rows respectively. Horizontal forces are defined at the seventh row. In order to define the dimensions and material properties of the structure, only 17 numbers are entered.
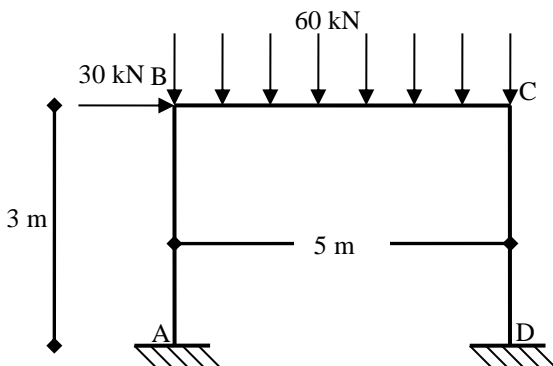
Fig. 6. Frame structure and loading conditions of the first case study problem

This represents the simplicity and practicality of the structural analysis software. Member-end rotations and side-sway is computed as 0.0022081, 0.0011397 radians and 0.0027609 meter, respectively. When the parameters are substituted into equation set, normal forces of the columns are computed as 111.54 kN, and 188.46 kN for columns BA and CD respectively. The analysis duration is reported as 1 millisecond.

4.2. Case problem 2

Second case study problem consists of one storey, two span frame structure. Loads and support conditions are given in Fig. 7. In addition to this, foundations A, D, and F are settled 1, 3, and 1 cm respectively.

Data related with the structure is entered as text file. Required data for the structural analysis is given below.

```
3 1 25.0 420.0 30.0 3.0
1 3 1
60.0 72.0
5.0 6.0
25.0 60.0 25.0 60.0 25.0 60.0
60.0 25.0 60.0 25.0
100.0
```

Data entrance involves entering twenty-four numbers since there are one more beam and column when compared with the previous test problem. However, data entrance is still quick and easy. Three rotation angles at the nodes and side-sway of the structure are computed as 0.0029545, 0.0009382, -0.0021469 radians, and 0.0014285 meter respectively. Normal forces on the columns are obtained as 121.78, 301.67, and 308.55 kN for columns AB, DC, and FE respectively. If the foundation settlements were equal, which means there is no differential settlement, normal forces on the columns would be 101.97, 383.21, and 246.82 kN respectively. Second analysis with no differential settlement is conducted by replacing the number 3 by 1 which is in the second line. The convenience of data entrance process makes implementation of what if scenarios very easy. Computation time is 1 millisecond.
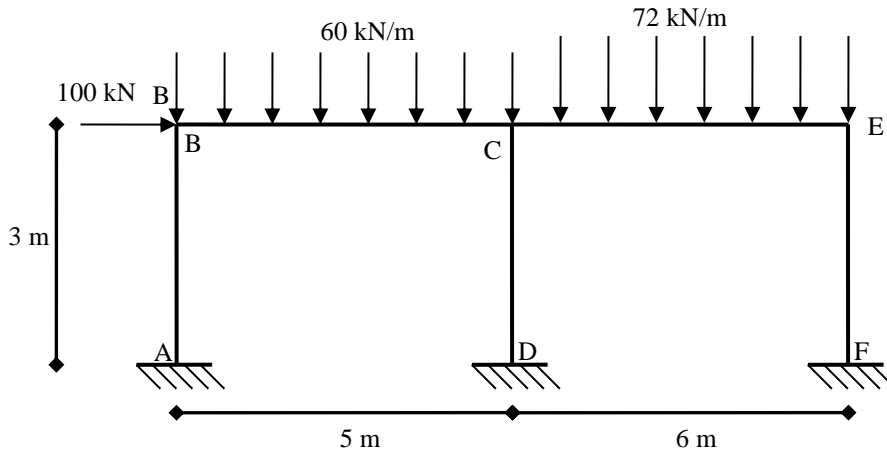
Fig. 7. Frame structure and loading conditions of the second case study problem

4.3. Case problem 3

Third case study problem consists of a building with two floors, and two spans. Loads and support conditions are given in Fig. 8. In order to define the structure and loading conditions, the following data is entered.

```
3 2 25.0 420.0 30.0 3.0
1 3 1
60.0 72.0
5.0 6.0
25.0 60.0 25.0 80.0 25.0 60.0
60.0 25.0 60.0 25.0
100.0 150.0
```

Only one additional data is entered to define the horizontal load acting on the second floor. The dimensions of the structural elements and uniformly distributed loads defined for the first floor are duplicated for the second floor. Therefore, data entry process becomes fast and easy even though the number of floors is increased.

Normal forces on the columns are computed as 246.05, 106.15, and 379.80 kN for the second floor and 419.46, 335.94, and 708.60 kN for the first floor. If the settlements of the foundations were equal to each other normal forces on the columns would be computed as 126.67, 351.08, and 254.25 kN for the second floor, and 230.89, 731.82, and 501.29 kN for the first floor. In this case problem 8x8 matrix is inverted. Computation time is 6 milliseconds.

4.4. Case problem 4

Fourth case study problem consists of a building with four floors and three spans. Loads and support conditions are given in Fig. 9. In order to define the structure and loading conditions, the following data is entered.

```
4 4 25.0 420.0 30.0 3.0
1 4 4 1
60.0 72.0 60.0
5.5 6.0 5.5
25.0 60.0 25.0 80.0 25.0 80.0 25.0 60.0
60.0 25.0 60.0 25.0 60.0 25.0
100.0 150.0 200.0 250.0
```

Structure and the loading conditions are defined by entering 34 numbers. Data entrance is still simple and fast. Normal forces on the columns are computed as 205.53, 387.46, 293.76, and 205.25 kN for the fourth floor, 563.53, 618.61, 422.67, and 579.18 kN for the third floor, 869.22, 753.74, 709.24, and 943.80 kN for the second floor, 1053.67, 1034.73, 1082.09, and 1197.52 kN for the first floor.

If there is no differential settlement, normal forces on the columns are computed as 152.31, 384.33, 391.67, and 163.70 kN for the fourth floor, 315.99, 748.45, 759.03, and 360.53 kN for the third floor, 448.43, 1111.09, 1175.73, and 540.74 kN for the second floor, and finally 558.24, 1506.35, 1575.29, and 728.12 kN for the first floor respectively. Duration of the analysis is measured as 15 milliseconds.
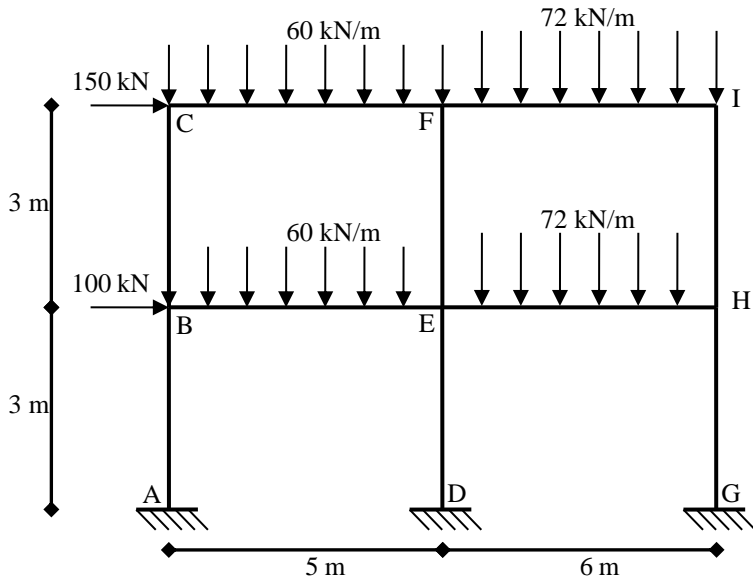
Fig. 8. Frame structure and loading conditions of the third case study problem
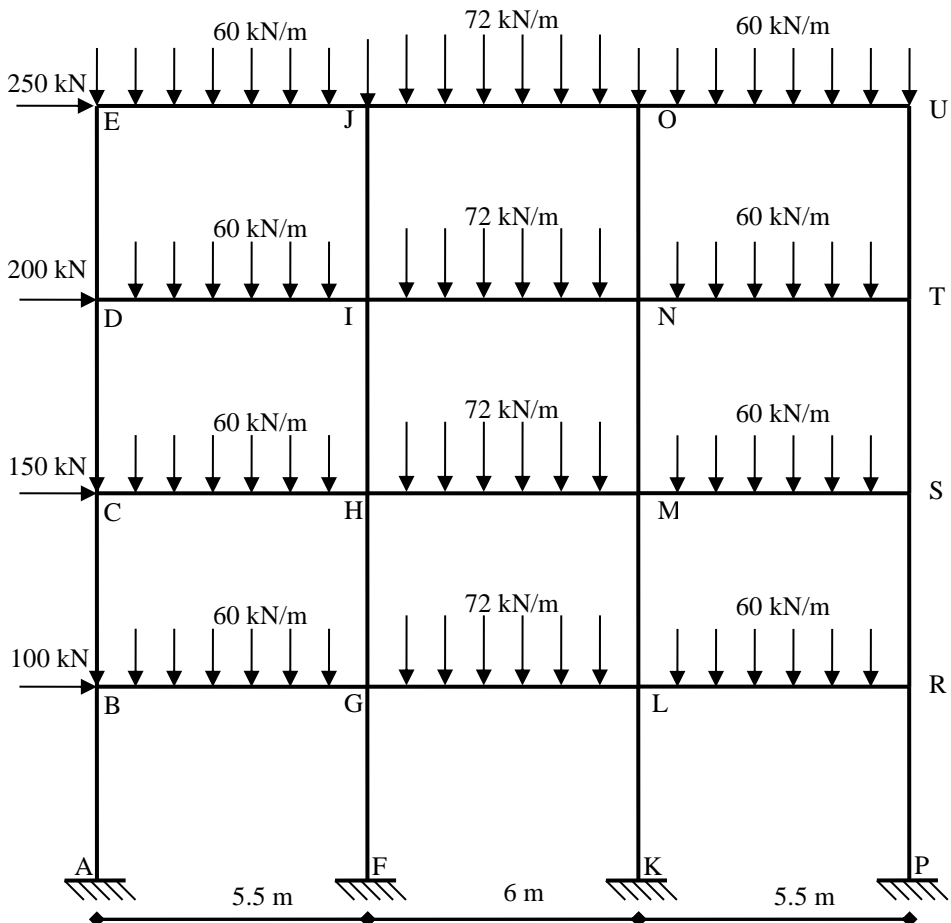

Fig. 9. Frame structure and loading conditions of the fourth case study problem

4.5. Case problem 5

Fifth case study problem consists of a building with twenty floors and four spans. Loads and support conditions are not given as figure because of allocation of large space. Loading conditions of this test problem are similar to the previous case problem. Magnitudes of the distributed loads acting on the beams are defined as 40, 60, 80, and 60 kN/m respectively. The horizontal forces increase with 30 kN gradient at each floor. In order to define the structure and loading conditions, the following data is entered.

```
5 20 25.0 420.0 30.25 3.0
1 4 6 6 1
40.0 60.0 80.0 60.0
4.95 6.5 6.25 5.50
80.0 30.0 30.0 120.0 30.0 120.0 30.0 120.0 100.0 30.0
60.0 30.0 60.0 30.0 60.0 30.0 60.0 30.0
30.0 60.0 90.0 120.0 150.0 180.0 210.0 240.0 270.0
300.0 330.0 360.0 390.0 420.0 450.0 480.0 510.0
540.0 570.0 600.0
```

The structure is analyzed in 735 milliseconds. Computation duration includes the preparation of report file that contains the stiffness matrix and coefficients of member-end moments. In this sample problem number of unknowns is 120. Consequently, the inverse of a 120x120 matrix is computed. Computational demand of the analyzed structure is very large. Data entrance procedure is facilitated by simplifications on the structure where the loads and dimension of the structure is the same at each floor. Therefore, different design combinations can be tried and the suitability of different design alternatives can be evaluated.

Developed software utilizes slope-deflection equations which is 2-D. Therefore, off the shelf commercial programs which utilize 3D finite element analysis methods would not be an equitable comparison. LinPro structural analysis software is used for the comparison of first four case study problems. Developed software models the structure by considering only rotations at the nodes; on the other hand, LinPro models not only joint rotations but also horizontal and vertical deflections. Consequently, LinPro forms larger stiffness matrixes than the developed software.

First four test problems are solved on LinPro 2.7 and Frame3DD but the installed versions do not report solution time. Some of the commercial structural analysis software report solution time but these programs enforce 3D structural analysis. Therefore, it will not be fair to compare with the computation durations of 2D structural analysis by considering only joint rotations, and of 3D structural analysis by considering both joint rotations and displacements. In the last example, 20 floor building with 4 spans is analyzed. This building has 120 unknowns since the structure is analyzed by slope-deflection method. If a 3D structural analysis is executed depending on the number of perpendicular axes, it is expected to have more than 50 unknowns at each floor which makes more than 1000 unknowns to be solved. Therefore, solution time is expected to be less for the developed software. Besides computation time, definition of the loads and structural elements are significantly faster than the 3D structural analysis when the developed software is used.

5. Conclusion

In this study structural analysis software for the preliminary design of frame structures is developed and tested on five case study problems. Analysis times of the structures are less than one second for all of the cases. Moreover, defining the dimensions and properties of the structure and loading conditions are very simple. Data entrance procedure of a frame structure with twenty floors, four spans (five axes) requires entering 57 numbers. Therefore, the preliminary design of the structure can be performed in a short time. Many design alternatives can be tested and what if scenarios can be executed. The dimensions of the structural elements and span of the beams can be determined approximately before the final design.

The developed software is a freeware and can be used for educational purposed as well. The students can evaluate their solutions by using the developed preliminary structural analysis software. Structural analysis by using slope-deflection equations is an outdated operation. However, programming the process on C++ is relatively easy.

In addition to this, size of the matrix to be inverted is smaller than matrix or finite element methods. The large sized matrixes allocate very high memory and they should be handled specifically, which causes difficulties in programming.

Most of the computer codes are provided in the text. Correctness of the algorithm is tested on case problems. The moment equilibrium, horizontal and vertical force equilibrium conditions are satisfied at each test problems. A researcher can use these codes and implement slope-deflection algorithm as well. Civil engineers can develop their own specific structural analysis software for any type of frame structures. Investment costs for the structural analysis software can be reduced by utilizing the proposed approach. In addition to this, the preliminary design step can be completed in shorter time compared to the design process by using BIM based structural analysis software.

References

[1]   Maney GA (1915) "Studies in Engineering", Minneapolis: University of Minnesota.

[2]   Fazio P, and Gowri K (1987) Structural analysis software and the C programming language. Computers & structures, 25(3): 463-465.

[3]   Ha KH (1990) C language for finite element programming. Computers & Structures, 37(5): 873-880.

[4]   Sanal Z (1994) Finite element programming and C. Computers & structures, 51(6): 671-686.

[5]   Gu Y, Zhang H, Guan Z, Kan Z, Li Y, & Zhong W (1999) New generation software of structural analysis and design optimization--JIFEX. Structural Engineering and Mechanics, 7(6): 589-599.

[6]   Leet K, and Uang CM, "Fundamentals of Structural Analysis (second ed)", McGraw-Hill Education, 2002.

[7]   West HH, and Geshwindner LF, "Fundamentals of Structural Analysis", John Wiley & Sons, United States of America, 2002.

[8]   Golub GH, Van L, and Charles F "Matrix Computations (third ed.)", Baltimore: Johns Hopkins, 1996.

[9]   Bettemir ÖH (2017) Akıllı Telefonlarda Yapı Analizi için Hızlı Yakınsayan Moment Dağıtma Algoritması. Teknik Dergi, 28(1): 7765-7774., doi: 10.18400/tekderg.299141 (Technical Note).