ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
INSTITUTE OF TECHNOLOGY
DEPARTMENT OF CIVIL ENGINEERING

DEVELOPMENT OF A PROGRAM FOR DYNAMIC ANALYSIS OF BUILDING FRAMES

By

Dessie Ayelign

A Thesis Submitted to the Graduate School of the Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Civil Engineering

June 2011

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
INSTITUTE OF TECHNOLOGY
DEPARTMENT OF CIVIL ENGINEERING

DEVELOPMENT OF A PROGRAM FOR DYNAMIC ANALYSIS OF BUILDING FRAMES

By

Dessie Ayelign

Approved by Board of Examiners

Dr. -Ing Bedilu Habte Advisor	_______________ Signature	______________ Date
Dr. Adil Zekaria External Examiner	_______________ Signature	______________ Date
Dr. Shifferaw Taye Internal Examiner	_______________ Signature	______________ Date
Ato Belete berhanu Chairman	_______________ Signature	______________ Date

Declaration

I declare that this thesis is my original work. This thesis has not been presented for any other university and is not concurrently submitted in candidature of any other degree, and that all sources of material used for the thesis have been duly acknowledged.

Name: Dessie Ayelign

Signature: __________

Place: Addis Ababa University
Institute of Technology

Acknowledgements

It goes without saying that my first indebtedness is to the developers of Scilab and the consortium that continues to develop it. I must also thank my advisor Dr.-IngBediluHabte, who first introduced me to Scilab. I appreciate his unreserved assistance, constructive and timely comments at all stages of my work. I should strongly appreciate his patience full guidance in many discussions we made on various problems I faced during the course of the work and for lending me the necessary books and materials in the field of study, and for his comments and excellent advice throughout the preparation of this thesis. All my postgraduate instructors deserve great thanks since their inputs are valuable.

The research was financially supported by Addis Ababa University, School of Graduate Studies for which they should deserve great credits.

I would also like to use this opportunity to convey my gratitude to my friends. Without their support and encouragement, I could not have this opportunity to complete my study.

Finally, I have no words to express my warm feeling of appreciation and thanks to my family for their lovely encouragement.

Dessie Ayelign

Abstract

Earthquake analysis is normally a two-stage process: we first estimate the dynamic properties of the structure (natural frequencies and mode shapes) by analyzing it in the absence of external loads, and then use these properties in the determination of earthquake response. Earthquakes often induce non-linear response in structures. However, most practical seismic design continues to be based on linear analysis.

The objective of this research is to develop a software program for Earthquake analysis of planar frames according to the provisions given by Ethiopian Building Code Standard (EBCS-8, 1995) and compare the results from the developed program with other sources.

The most commonly used methods for dynamic analysis of structures subjected to earthquake loads are modal and spectral analysis. Response spectral analysis given by the provision of EBCS-8, 1995 is used for developing this computer program.

In this thesis, an overview of various dynamic analysis procedures is presented with a detailed description for response-spectrum analysis, which is used to develop the computer program. Conclusions are drawn concerning the reliability of the responses given by the developed program. Recommendations for the use of the program are also provided.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\ddot{U}g(t)$	*Ground acceleration*
ϕ_n	*Mode shapes of vibration*
$[\ddot{U}(t), \dot{U}(t), U(t)]$	*The time dependent nodal acceleration, velocity and displacement*
[Cx, Cz]	*X and Z direction cosines of a member*
$[F_I, F_D, F_R, F_E]$	*Vector of inertia force, damping force, restoring force, earthquake load force*
$[K_D, K_R, K_W]$	*Factor reflecting ductility class, regularity and prevailing failure*
A	*Cross sectional area*
An	*Modal acceleration*
c	*Damping coefficient*
Cn	*Generalized damping matrix*
E	*Modulus of elasticity*
E_E	*Seismic action effect*
fn	*Equivalent static lateral force*
I	*Importance factor*
K	*Stiffness matrix*
Kn	*Generalized stiffness matrix*
L	*Length of a member*
Ln	*Modal mass participation*
M	*Mass matrix*
Mn	*Generalized mass matrix*
P	*Vector of nodal loads*
P(t)	*Time dependent force*
Pn(t)	*Generalized time dependent force*
r	*Rotation matrix*
R	*Support reaction*
S	*Site coefficient for soil characteristics*
$S_d(t)$	*Design spectrum*

S_n *Modal inertia force distributions*

T_n *Natural periods of vibration*

U *Displacement vector*

U_d *Known nodal displacements*

U_f *Unknown nodal displacements*

αo *Bedrock acceleration*

β *Design response factor*

γ *Behavior factor*

γd *Basic value of behavior factor*

ι *Vector of influence coefficient*

ξ *Damping ratio*

ρ_{in} *Correlation coefficient*

ω_n *Natural frequencies of vibration*

Γ_n *Modal mass participation factor*

1. Introduction

Structural analysis (i.e., modeling and calculation of responses such as stresses and displacements due to applied loading) is one of the major components of the structural design process. The quality of the final design depends directly on the quality of the analysis method employed. Thus, engineers have always strived for developing accurate analysis methods, which provide realistic predictions of structural response. However, an accurate analysis method is usually achieved at the expense of increased computational time and cost. This requires that designers find an option to develop a program for doing the analysis w+
ith short time and minimum cost.

Seismic design is the most frequent dynamic problem that structural engineers encounter in practice.Dynamic analysis is the most natural approach towards the assessment of earthquake response, but is significantly more demanding than static analysis in terms of computational effort and interpretation of results.

Earthquakes often induce non-linear response in structures. However, most practical seismic design continues to be based on linear analysis. The effect of non-linearity is generally to reduce the seismic demands on the structure, and this is normally accounted for by a simple modification to the linear analysis procedure.The most commonly used methods for dynamic analysis of structures subjected to earthquake loads are modal and spectral analysis.In this thesis, a computer program has been developed for the dynamic analysis of structures according to the Ethiopian Building Code Standards.

This thesis consists of five chapters of which chapter two deals with literature review of thesubject matter. This chapter briefly summarizes structural analysis of a system including modeling and methods of analysis of a structure.

The third chapter contains the core of the research work. This chapter explains how the input data will be prepared; how the program is developed; how the program is run and how to use the outputs of the program. It also contains the flowchart of the steps used to develop the program and briefly introduces the software Scilab.

In the forth chapter, frames are selected for the purpose of comparison. Each frame will be discussed part by part in thischapter with their output results found from different analysis procedures.

The last chapter is made to contain the conclusions drawn from the outputs based on the developed program and the recommendations based on the findings.Finally, the flowchart and the coding of the developed program are provided in the appendix.

1.1. Problem Statement

- An accurate analysis method is usually achieved at the expense of increased computational time and cost. This requires that designers find an option for doing the analysis with short time and minimum cost. Therefore, use of software is essential.
- Most programs developed for structural analysis using response spectrum analysis until now are not based on our code, Ethiopian Building Code Standard (EBCS-8, 1995). As a result, a program, which is developed, based on our local code is necessary.
- EBCS-8, 1995 gives an option to combine modal responses using Complete Quadratic Combination(CQC) rule without giving the equation for this method. This program will help to do Response-spectrum analysis and combine the responses according to the relevant modal combination rules.

1.2. Objective of the Research

The main objective of this research is to develop a computer program for Earthquake analysis of planar frames according to the provisions given in EBCS-8, 1995.

The specific objective of the research is:

- To develop a computer program for response spectrum analysis of planar frames;
- To verify the analysis results of the developed program in this research by comparing with results from other sources.

1.3. Methodology

The methodology of the research includes:

- Literature survey;
- Preparing the analysis equations used for programming;
- Preparing the flowchart of the steps used in static as well as dynamic analysis of frames;
- Developing the program;
- Making a hand calculation and using the program to analyze selected frames;
- Comparing the results.

1.4. Application of the Research

- ✓ The developed computer program will be applicable for analysis of planar frames for static and/or seismicloads. A seismic load due to ground acceleration is performed according to the design spectrum provided by EBCS 8, 1995.
- ✓ The program is flexible to insert the design spectrum required by the user and modify for any other design spectrum.
- ✓ Students may use this for future research in line with upgrading the program to handlemany other types of structural analysis such as non-linear analysis.

2. Literature Review

2.1. Structural Analysis Procedures

The analysis of a structural system to determine the deformations and forces induced byapplied loads or ground excitation is an essential step in the design of a structure to resistearthquakes.

The use of seismic analysis both in research and practice has increased substantially in recent yearsdue to the proliferation of verified and user - friendly software and the availability of fast computers (Bozorgnia & Bertero, 2006).

A structural analysis procedure requires:

i) A model of the structure,

ii) A representation of the earthquake ground motion or the effects of the ground motion and

iii) A method of analysis for forming and solving the governing equation.

2.1.1. Modeling of structures

Structural analysis is performed on a model of the structure—not on the real structure—so the analysis can be no more accurate due to the assumptions in the model. The model must represent the distribution and possible time variation of stiffness, strength, deformation capacity and mass of the structure with accuracy sufficient for the purpose of the analysis in the design process(Bozorgnia & Bertero, 2006).

All real structures potentially have an infinite number of displacements. Therefore, the most critical phase of a structural analysis is to create a computer model with a finite number of members connected at nodes (joints) that will simulate the behavior of the real structure. The mass of a structural system, which can be accurately estimated, is lumped at the nodes. In addition, for linear elastic structures, the stiffness properties of the members can be approximated with a high degree of confidence with the aid of experimental data.

All structures are three dimensional, but it is important to decide whether to use a three dimensional model or simpler two-dimensional models. The analysis methods are the same whether the model is two-dimensional or three-dimensional. Generally, two-dimensional models are acceptable for buildings with regular configuration and minimal torsion; otherwise, a three-dimensional model is necessary with a representation of the floor diaphragms as rigid or flexible components(Bozorgnia & Bertero, 2006).

The geometry of the structural model is described by the position of the nodes in a global coordinate system, denoted by X, Y and Z.Two nodes define a frame element, which may be either straight or curved. This research is limited to straight elements because a curved element can always be approximated by several straight elements at the expense of increased modeling effort and computational cost. The element geometry is established in a local coordinate system *x*, *y*, *z*.

The element response can be completely described by the relation between the force vector **p** and the displacement vector **u**. For three-dimensional (3d) elements, the force vector has 12 components: at each node, there are three forces in the local *x, y, z* coordinate system and three moments about the axes of the local coordinate system. In the two-dimensional (2d) case, there are two forces and one moment at each node, providing six components of the force vector. The displacement vector is defined in an analogous manner; and includes translations in the direction of the local axes and rotations about the local axes at each node (Bozorgnia & Bertero, 2006).

Modeling Considerations

The location of the joints and members is critical in determining the accuracy of the structural model. Some of the factors that are needed to consider when defining themembers (and hence joints) for the structure are:

- The number of members should be sufficient to describe the geometry of the structure.
- Member boundaries, and hence joints, should be located at points of discontinuity:
    - Structural boundaries, e.g., corners
    - Changes in material properties
    - Changes in thickness and other geometric properties
    - Support points (restraints)
    - Points of application of concentrated loads
    - Points of abrupt change in member loads
- More than one member should be used to model the length of any span forwhich dynamic behavior is important. This is required because the mass is always lumped at the joints, even if it is contributed by the members.

2.1.2. Loads and Boundary Conditions

Loads are specified forces applied to members (elements) or nodes. Gravity loads may be applied toelements or considered as nodal loads depending on the gravity load path. The vector ofnodal loads for a structure is denoted by **P,** with six components of force at each node for3d problems and three components for 2d problems. In contrast with nodal loads, elementloads are included in the element force-deformation relationship as distributed loads **w***(x)*defined in the local coordinate system for the element(Bozorgnia & Bertero, 2006).

The displacements of all nodes are collected into a single displacement vector **u** for theentire model in which each component is a degree of freedom. The set of allglobal degree of freedoms (DOFs)are separated into two subsets: the DOFs with unknown displacement values and theDOFs with specified displacement value. Each DOF in the model must be included inone of the two sets. The unknown displacements are called the free DOFs and aredenoted by $\mathbf{u_f}$. The second sets of displacements correspond to the restrained DOFs; and are denoted by $\mathbf{u_d}$. The restrained DOFs are generally assigned a value of zero to indicate afixed displacement. Theselection of restrained displacements at the supports is an important step in the structuralmodeling, and the supports of a model are commonly identified with the symbols shownin Figure 2.1 for typical two-dimensional cases. The arrows in Figure 2.1 indicate the restrained DOFs,and thus the corresponding support reactions of each support type.

Since the displacements are partitioned into two sets, so is the nodal force vector, **P**. Thenodal forces at the free DOFs of the model are specified as nodal loads, and are denotedby $\mathbf{P}_f$. The forces at the restrained DOFs are the supportreactions and are denoted by $\mathbf{P}_d$. These can be evaluated once the equations for the free DOFs are solved.

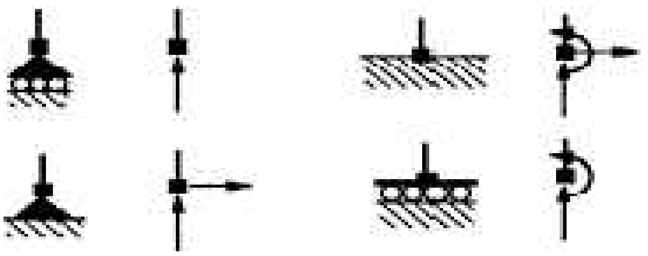


Figure 2.1: Typical support symbols for the two-dimensional case(Bozorgnia & Bertero, 2006).

2.1.3. Equilibrium and Compatibility

Equilibrium equations set the externally applied loads equal to the sum of theinternal element forces at all joints or node points of a structural system; they arethe most fundamental equations in structural analysis and design. The exactsolution for a problem in mechanics requires that the equationsof equilibrium for all elements within the structure must be satisfied.

Equilibrium is a fundamental law of physics and cannot be violated within a"real" structural system. Therefore, it is critical that the mathematical model,which is used to simulate the behavior of a real structure, also satisfies thosebasic equilibrium equations (Wilson, 2002).

In the analysis of a structural system of discrete elements, all elements connectedto a joint or node point must have the same absolute displacement. If the nodedisplacements are given, all element deformations can be calculated from thebasic equations of geometry. Compatibility equations are mathematical equations that determine whether a particular deformation will leave a body in a compatible state.Compatibility requirements should be satisfied. However, if one has a choice of satisfying equilibrium or compatibility, one should use the equilibrium basedsolution. For real nonlinear structures, equilibrium is always satisfied in the deformed position. Many real structures do not satisfy compatibility causedby creep, joint slippage, incremental construction and directional yielding (Wilson, 2002).

2.1.4. Methods of Analysis

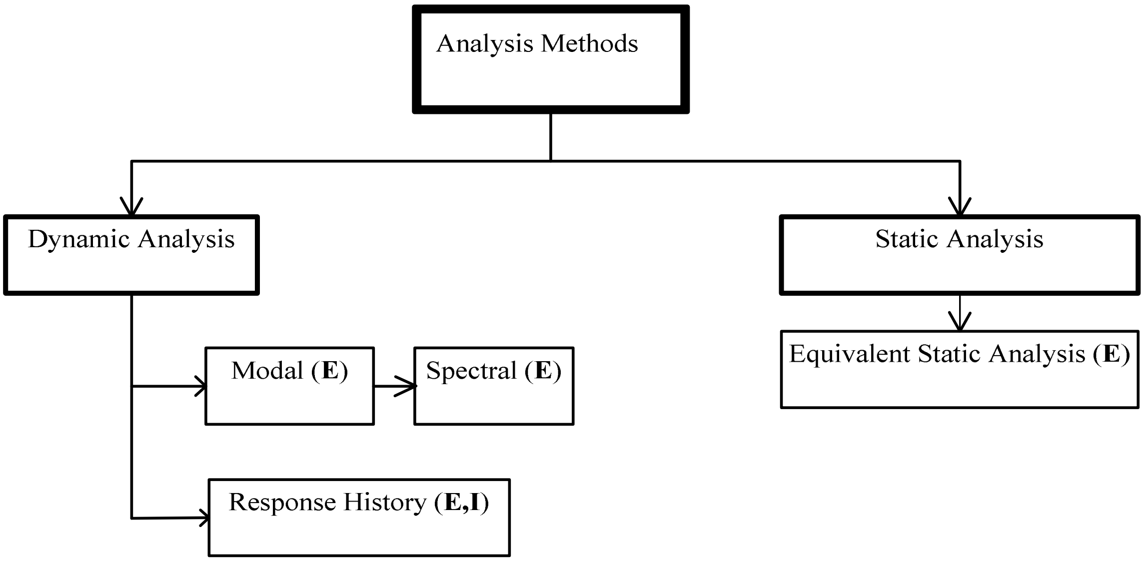
An overview of the main methods of structural analysis used in earthquake engineering issummarized in Figure 2.2.

There is arange of methods from a plastic analysis to a sophisticated nonlinear, dynamic analysis ofa detailed structural model that can be used, depending on the purpose of the analysis inthe design process. An important decision in a structural analysis is to assume whether the relationshipbetween forces and displacements is linear or nonlinear.

Dynamic analysis is the most naturalapproach towards the assessment of earthquake response, but is significantly more demanding thanstatic analysis in terms of computational effort and interpretation of results(Elnashai and Sarno, 2008).

An earthquake analysis generally includes gravity loads and a representation of theground motion at the site of the structure. Earthquake ground motion induces the mass ina structure to accelerate, and the resulting response history can be computed by dynamicanalysis methods. In many design procedures, it is common to perform a dynamic analysiswith a response spectrum representation of the ground motion expected at the site(Chopra, 1995). For response history analysis, several analyses with different groundmotion histories of the earthquake hazard are generally required.

After a structural model and earthquake loading are defined, an analysis method isneeded to compute the response. The governing equations are formed using equilibrium,compatibility and force-deformation relationships for the elements and the structure, andare expressed in terms of unknown displacements (or degrees of freedom, referred to asDOFs). To elucidate the theory and provide a compact mathematicalrepresentation, the fundamental relationships are expressed using matrix algebra. Sincethe governing equations may have a large number of degrees of freedom, they must besolved numerically using a computer-based analysis method.



Key: **E** = elastic analysis; **I** = inelastic analysis

Figure 2.2: Common Methods of Structural Analysis used in Earthquake(Elnashai and Sarno, 2008).

It is clear that staticapproaches are less time - consuming but often reliable only for a limited class of structures, e.g. regular systems under normal strong ground motion.

Inelastic large displacement response history is the most powerful tool of analysis. However, its potentialaccuracy and reliability are balanced by its complexity due to the selection of the seismic input, the structural modeling and the time – consuming computational schemes (Elnashai and Sarno, 2008).

2.2. Dynamic Analysis

2.2.1. Dynamic Properties of Structures

For linear dynamic analysis, a structure can be defined by three key properties: its stiffness, mass and damping.

All structures gradually dissipate energy as they move, through a variety of internal mechanisms that are normally grouped together and known as damping. Without damping, a structure, once set in motion, would continue to vibrate indefinitely. There are many different mechanisms of damping in structures. However, analysis methods are based on the assumption of linear viscous damping, in which a viscous dashpot generates a retarding force proportional to the velocity difference across it. The damping coefficient, *c*, is the constant of proportionality between force and velocity, measured in Newton-second per meter (Ns/m). Whereas it should be possible to calculate values of mass 'm'and stiffness 'k'with some confidence, *c* is a rather nebulous quantity that is difficult to estimate. It is more convenient to convert the damping coefficient, *c*,to a dimensionless parameter ξ, called the damping ratio(Elghazouli, 2009):

$$\xi = \frac{c}{2\sqrt{km}} \qquad (2.1)$$

ξ can be estimated based on experience of similar structures.

In reality, all structures have distributed mass, stiffness and damping. It is possible to obtain reasonably accurate estimates of the dynamic behavior using *lumped parameter* models, in which the structure is modeled as a number of discrete masses connected by light spring elements representing the structural stiffness and dashpots representing damping.

Each possible displacement of the structure is known as a *degree of freedom*. Obviously, a real structure with distributed mass and stiffness has an infinite number of degrees of freedom, but in lumped-parameter idealizations,this is limited to the possible displacements of the lumped masses(Elghazouli, 2009).

2.2.2. Dynamic Equilibrium

All real physical structures behave dynamically when subjected to loads ordisplacements. The additional inertia forces, ***from Newton's second law***, areequal to the mass times the acceleration. If the loads or displacements areapplied very slowly, the inertia forces can be neglected and a static load analysiscan be justified. Hence, dynamic analysis is a simple extension of static analysis.

The equation of equilibrium for a multi - degree of freedom (MDOF) system subjected to earthquakeaction is as follows:

$$\mathbf{F}_I + \mathbf{F}_D + \mathbf{F}_R = \mathbf{F}_E \quad (2.2)$$

where $\mathbf{F}_I$ is the inertia force vector;$\mathbf{F}_D$ is the damping force vector;$\mathbf{F}_R$is the vector of restoring force; and$\mathbf{F}_E$is the vector of earthquake load.Equation (2.2) is based on physical laws and is valid for both linear andnonlinear systems if equilibrium is formulated with respect to the deformedgeometry of the structure.

For many structural systems, the approximation of linear structural behavior is made to convert the physical equilibrium statement, Equation (2.2), to the following set of second-order, linear, differential equations (Wilson, 2002):

$$\mathbf{m\ddot{u}(t)_a + c\acute{u}(t)_a + ku(t)_a = F_E} \quad (2.3)$$

in which **m** is the mass matrix (lumped or consistent), **c** is a viscous damping matrix (which is normally selected to approximate energy dissipation in the real structure) and **k** is the static stiffness matrix for the system of structural elements.

The time-dependent vectors $\mathbf{u(t)_a}$, $\mathbf{\acute{u}(t)_a}$and $\mathbf{\ddot{u}(t)_a}$are the absolute node displacements, velocities and accelerations, respectively.

The external load$\mathbf{F_E}$for earthquake excitation is:

$$\mathbf{F_E = -m\iota\, \ddot{u}_g(t)} \quad (2.4)$$

ι is a vector of influence coefficients. For simple structural modelswith degrees of freedom corresponding to the horizontal displacements at storey level, ι is a unit vector.In this case, it represents the rigid body acceleration of the structure due to a unit base acceleration. Theuse of MDOF lamped systems for dynamic analyses results in a diagonal mass matrix **m,** in whichtranslational and rotational masses are located along the main diagonal.

2.2.3. Methods of Dynamic Analysis

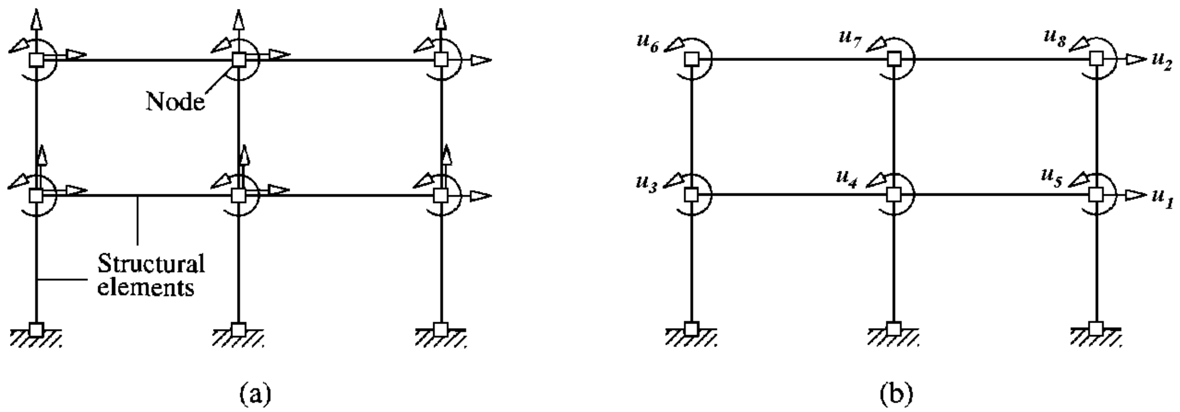
To design economically a structure subjected to severe seismic actions, post elastic behavior is allowed. The default method of analysis uses linear procedures, and post elastic behavior is accounted for by simplified methods. More detailed analysis methods are normally only utilized in important or irregular structures.

Dynamic analysis is a two-stage process: first the dynamic properties of the structure (natural frequencies and mode shapes) are estimated by analyzing it in the absence of external loads, and then use these properties in the determination of earthquake response.

Some of the methods of dynamic analysis of structures exist as shown in Figures 2.2. These methods can be employed either in the time or in the frequency domain. The most commonly used methods for dynamic analysis of structures subjected to earthquake loads are modal, spectral and response history(Elnashai and Sarno, 2008).

Discretization of Frames for Dynamic Analysis

A frame structure can be idealized as an assemblage of elements- beams, columns, walls-interconnected at nodal points or nodes. The displacements of the nodes are the degrees of freedom. In general, a node in a planar two-dimensional frame has three DOFs-two translational and one rotation. A node in a three-dimensional frame has six DOFs –three translations (the x, y, and z components) and three rotations (about the x, y, and z-axes). For example, a two-story and two-bay planar frame has 6 nodes and 18 DOFs (Figure 2.3a). Axial deformations of beams can be neglected in analyzing most buildings, and axial deformations of columns need not be considered for low-rise buildings.With these assumptions the two-storey, two-bay frame has eight DOFs (Figure 2.3b).



(a) Axial deformation included, 18 DOFs; (b) Axial deformation neglected, 8 DOFs.

Figure 2.3: Degrees of Freedom

This structural idealizationis used for illustration. The external dynamic forces are applied at the nodes (Figure 2.4). The moments p_3(t) to p_8(t) are zero in most (if not all) practical cases (Chopra, 1995).

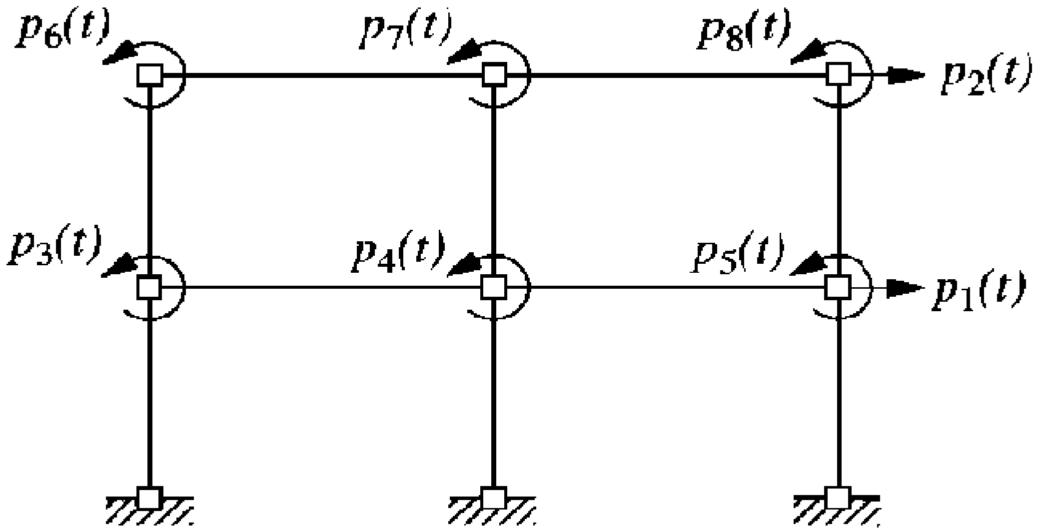


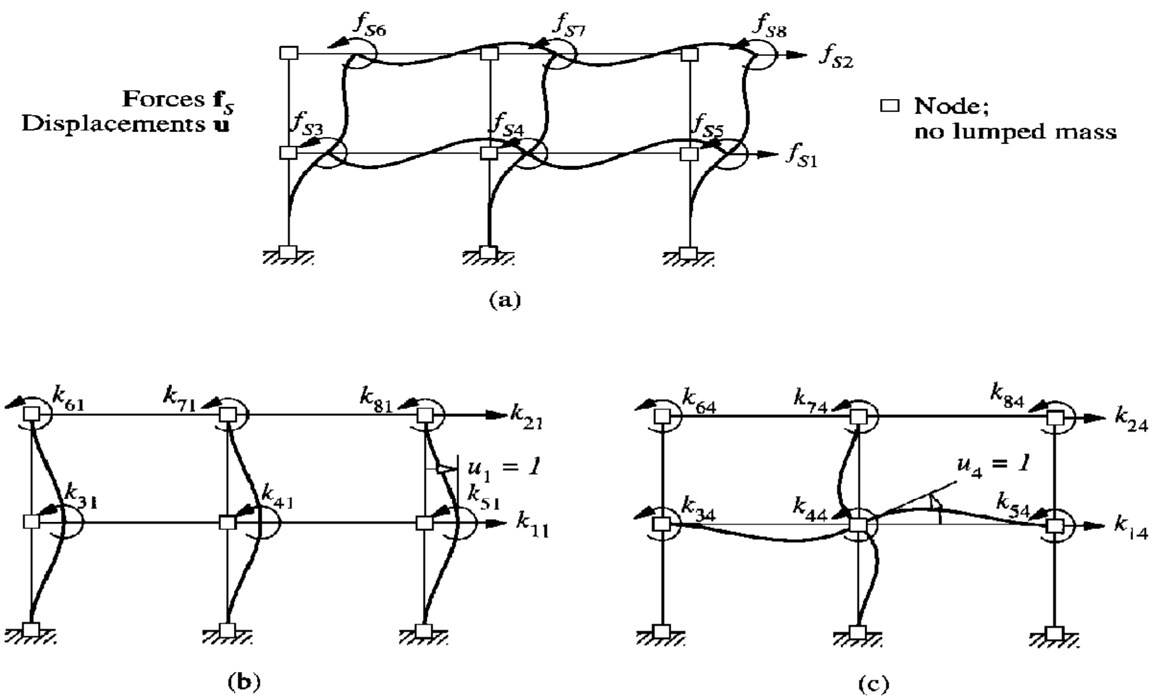
Figure 2.4: External dynamic Forces, p(t).

Elastic Forces

The external forces $\mathbf{f}_{sj}$on the stiffness component of the structure are related to the resulting displacements $\mathbf{u}_j$ (Figure 2.5a). For linear systems, this relationship can be obtained by the method of superposition and the concept of stiffness influence coefficients (Chopra, 1995).

A unit displacement along DOF j is applied, holding all other displacements to zero as shown in Figure 2.5; to maintain these displacements, forces must be applied along all DOFs.

The stiffness influence coefficient,$\mathbf{k}_{ij}$,is the force required along DOF i to cause a unit displacement at DOF j. In particular, the forces $\mathbf{k}_{i1}$ (i = 1,2,. . . , 8) shown in Figure 2.5b are required to maintain the deflected shape associated with $\mathbf{u}_1$ =1 and all other $\mathbf{u}_j$= 0. Similarly, the forces $\mathbf{k}_{i4}$ (i =1, 2, . . ., 8) shown in Figure 2.5c are required to maintain the deflected shape associated with $\mathbf{u}_4$= 1 and all other $\mathbf{u}_j$ = 0. All forces in Figure 2.5 are shown with their positive signs. However, some of them may be negative to be consistent with the imposed deformations.



(a) Stiffness components of frame; (b) stiffness influence coefficients for $\mathbf{u}_1$ = 1; (c) stiffness influence coefficients for $\mathbf{u}_4$ = 1.

Figure 2.5: Elastic Forces

The force $\mathbf{f_{si}}$ at DOF i associated with displacements $\mathbf{u_j}$, j=1 to N (Figure 2.5a), is obtained by superposition:

$$\mathbf{f_{si} = k_{i1}u_1 + k_{i2}u_2 + \ldots + k_{iN}u_N} \quad \text{……………………………………………...(2.5)}$$

One such equation exists for each i = 1 to N. The set of N equations can be written in matrix form:

$$\begin{bmatrix} f_{S1} \\ f_{S2} \\ \vdots \\ f_{SN} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1j} & \cdots & k_{1N} \\ k_{21} & k_{22} & \cdots & k_{2j} & \cdots & k_{2N} \\ \vdots & \vdots & & \vdots & & \vdots \\ k_{N1} & k_{N2} & \cdots & k_{Nj} & \cdots & k_{NN} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{Bmatrix}$$

Or

$$\mathbf{f_s} = \mathbf{ku} \quad \text{……………………………………………………………….. (2.6)}$$

where **k** is the stiffness matrix of the structure; it is a symmetric matrix (i.e. $\mathbf{k_{ij}} = \mathbf{k_{ji}}$).

The stiffness matrix **k** for a discretized system can be determined by any one of several methods. The j^{th} column of **k** can be obtained by calculating the forces $\mathbf{k_{ij}}$ (i= 1, 2, …, N) required to produce $\mathbf{u_j}$ =1 (with all other $\mathbf{u_i}$ =0). The direct equilibrium method is feasible to implement such calculations for simple structures with a few DOFs; it is not practical, however, for complex structures or for computer implementation. The most commonly used method is the direct stiffness method where in the stiffness matrices of individual elements are assembled to obtain the structural stiffness matrix (Chopra, 1995).

Inertia Forces

The external forces f_{Ij} acting on the mass component of the structure are related to the accelerations $\mathbf{\ddot{u}_j}$ (Figure 2.6a). Apply unit acceleration along DOF j, while the accelerations in all other DOFs are kept zero. According to D'Alembert's principle, the fictitious inertia forces oppose these accelerations; therefore, external forces will be necessary to equilibrate these inertia forces. The mass influence coefficient m_{ij} is the external force in DOF i due to unit acceleration along DOF j. In particular, the forces $\mathbf{m_{i1}}$ (i= 1, 2, …, 8) shown in Figure 2.6b are required in the various DOF to equilibrate the inertia forces associated with $\mathbf{\ddot{u}_1}$=1 and all other $\mathbf{\ddot{u}_j}$=0. Similarly, the forces $\mathbf{m}_{i4}$(i = 1, 2, …, 8) shown in Figure 2.6c are necessary to cause acceleration $\ddot{u}_4$ =1 and all other $\ddot{u}_j$ = 0. The force f_{Ii} at DOF i associated with accelerations $\ddot{u}_j$, j= 1 to N (Figure 2.6a), is obtained by superposition (Chopra, 1995):

$$\mathbf{f}_{Ii} = \mathbf{m}_{i1}\ddot{\mathbf{u}}_1 + \mathbf{m}_{i2}\ddot{\mathbf{u}}_2 + \ldots + \mathbf{m}_{ij}\ddot{\mathbf{u}}_j + \ldots + \mathbf{m}_{iN}\ddot{\mathbf{u}}_N \qquad (2.7)$$

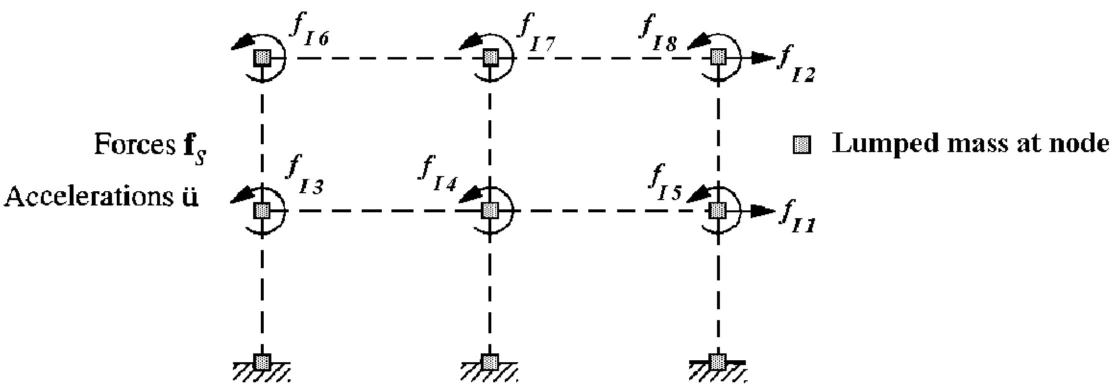
One such equation exists for i= 1 to N. The set of N equations can be written in matrix form:

$$\begin{bmatrix} f_{I1} \\ f_{I2} \\ \vdots \\ f_{IN} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1j} & \cdots & m_{1N} \\ m_{21} & m_{22} & \cdots & m_{2j} & \cdots & m_{2N} \\ \vdots & \vdots & & \vdots & & \vdots \\ m_{N1} & m_{N2} & \cdots & m_{Nj} & \cdots & m_{NN} \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \vdots \\ \ddot{u}_N \end{Bmatrix}$$

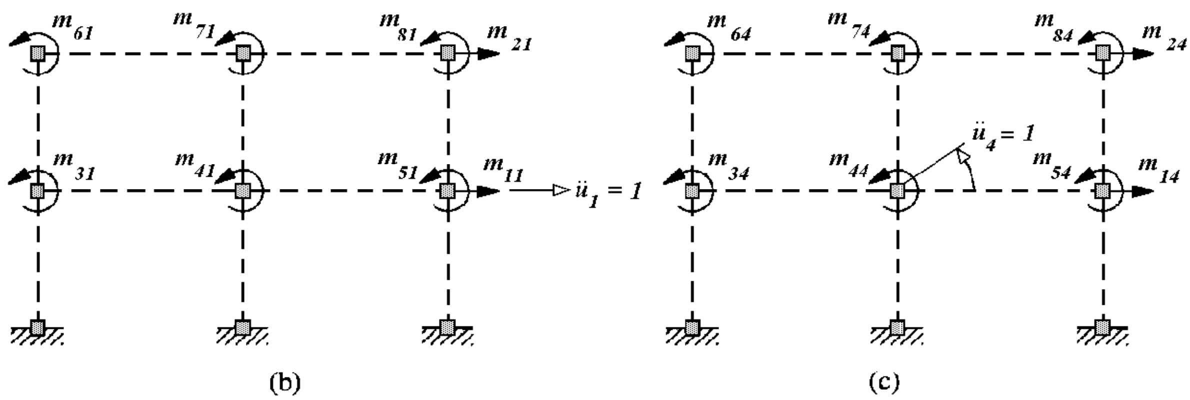
Or

$$\mathbf{f}_I = \mathbf{m}\ddot{\mathbf{u}} \qquad (2.8)$$

where **m** is the mass matrix of the structure; just like the stiffness matrix, the mass matrix is symmetric (i.e. $\mathbf{m}_{ij}=\mathbf{m}_{ji}$).



(a)



(a) Mass component of frame; (b) mass influence coefficients for $\ddot{u}_1$ **= 1; (c) mass influence coefficients for** $\ddot{u}_4$**= 1.**

Figure 2.6: Inertia Forces

The mass is distributed throughout an actual structure, but it can be idealized as lumped or concentrated at the nodes of the discretized structure; usually, such a lumped mass idealization is satisfactory. The lumped mass at a node is determined from the portion of the weight that can reasonably be assigned to the node. Each structural element is replaced by point masses at its two nodes, with the distribution of the two masses being determined by statics. The lumped mass at a node of the structure is the sum of the mass contributions of all the structural elements connected to the node. This procedure is illustrated schematically in Figure 2.7 for a two-story, two-bay frame where the beam mass includes the floor-slab mass it supports.

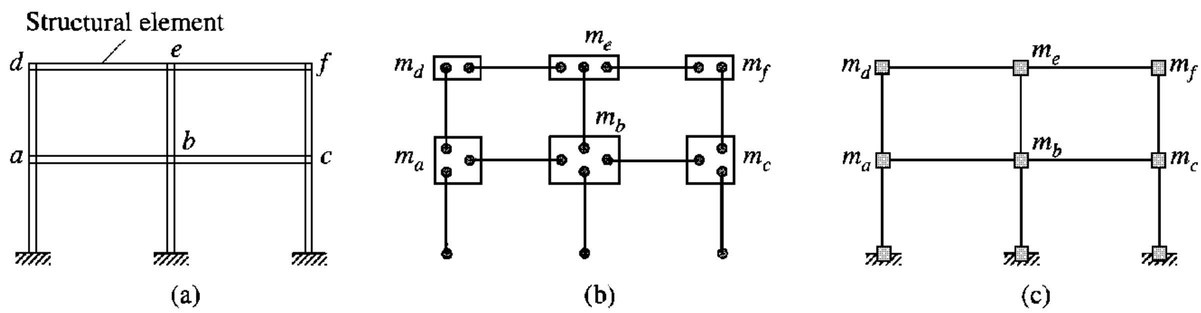


Figure 2.7:Lumping of mass at structural nodes

The lumped masses $\mathbf{m_a}$, $\mathbf{m_b}$, and so on, at the various nodes are identified. Once the lumped masses at the nodes have been calculated, the mass matrix for the structure can readily be formulated. Consider again the two-story, two-bay frame of Figure 2.3b (Chopra, 1995). The external forces associated with acceleration $\mathbf{\ddot{u}_1}$ =1 (Figure 2.6b) are $\mathbf{m_{11}}$=$\mathbf{m_1}$, where $\mathbf{m_1}$ = $\mathbf{m_a}$ +

$\mathbf{m_b}$+ $\mathbf{m_c}$ (Figure 2.7c), and $\mathbf{m_{i1}}$ = 0 for i =2, 3, …, 8. Similarly, the external forces $\mathbf{m_{i4}}$associated with $\mathbf{\ddot{u}_4}$ = 1 (Figure 2.6c) are zero for all i, except possibly for i = 4. The coefficient $\mathbf{m_{44}}$ is equal to the rotational inertia of the mass lumped at the middle node at the floor. This rotational inertia has negligible influence on the dynamics of practical structures; thus $\mathbf{m_{44}}$= 0.

In general, then, for a lumped mass idealization, the mass matrix is diagonal:

$$\mathbf{m_{ij} = 0 for i \neq j m_{jj} = m_j or 0}$$

Where$\mathbf{m_j}$ is the lumped mass associated with the j^{th} translational DOF and $\mathbf{m_{jj}}$= 0 for a rotational DOF (Chopra, 1995).

Natural Vibration Frequencies and Modes:

Free vibration of linear multi-degree-of-freedom systems is governed by equation (2.3) with F_E = 0, which for systems without damping is

$$\mathbf{m\ddot{u} + ku = 0} \quad \text{(2.9)}$$

Equation (2.9) represents N homogenous differential equations that are coupled through the mass matrix, the stiffness matrix, or both matrices; N is the number of DOFs.

The free vibration of an undamped system in one of its natural vibration modes can be described mathematically by:

$$\mathbf{u(t) = q_n(t)\phi_n} \quad \text{(2.10)}$$

The deflected shape ϕ_n does not vary with time. The time variation of the displacements is described by the simple harmonic function:

$$\mathbf{q_n(t) = A_n cos\omega_n t + B_n Sin\omega_n t} \quad \text{(2.11)}$$

where $\mathbf{A_n}$and $\mathbf{B_n}$ are constants of integration that can be determined from the initial conditions that initiate the motion. Combining Equations (2.10) and (2.11) gives:

$$\mathbf{u(t) = \phi_n(A_n cos\ \omega_n t + B_n\ Sin\ \omega_n t)} \quad \text{(2.12)}$$

ω_n and ϕ_n are unknowns.

Substituting equation (2.12) into equation (2.9) gives:

$$\mathbf{[-\omega_n^2 m\phi_n + k\phi_n]q_n(t) = 0} \quad \text{(2.13)}$$

This equation can be satisfied in one of two ways. Either $\mathbf{q_n(t) = 0}$, which implies that $\mathbf{u(t) = 0}$ and there is no motion of the system (this is so-called trivial solution), or the natural frequencies $\boldsymbol{\omega_n}$ and modes $\boldsymbol{\phi_n}$ must satisfy the following algebraic equation:

$$\mathbf{k}\boldsymbol{\phi_n} = \boldsymbol{\omega_n}^2 \mathbf{m}\boldsymbol{\phi_n} \quad \text{(2.14)}$$

This algebraic problem is called the matrix eigenvalue problem. When necessary it is called the real eigenvalue problem to distinguish it from the complex eigenvalue problem for systems with damping. The stiffness and mass matrix k and m are known; the problem is to determine the scalar ω_n^2 and ϕ_n.

To indicate the formal solution to equation (2.14), it is rewritten as

$$[\mathbf{k} - \boldsymbol{\omega_n}^2 \mathbf{m}]\boldsymbol{\phi_n} = \mathbf{0} \quad \text{(2.15)}$$

which can be interpreted as a set of N homogeneous algebraic equations for the N elements ϕ_{jn} (j=1, 2,…, N). This set always has the trivial solution $\phi_n = 0$, which is not useful because it implies no motion. It has nontrivial solutions of

$$\mathbf{det}[\mathbf{k} - \boldsymbol{\omega_n}^2 \mathbf{m}] = \mathbf{0} \quad \text{(2.16)}$$

Whenthe determinant is expanded, a polynomial of order N in $\boldsymbol{\omega_n}^2$is obtained. Equation (2.16) is known as the characteristic equation or frequency equation. This equation has N real and positive roots for $\boldsymbol{\omega_n}^2$because **m** and **k,** the structural mass and stiffness matrices, are symmetric and positive definite. The positive definite property of k is assured for all structures supported in a way that prevents rigid-body motion. Such is the case for civil engineering structures of interest to us, but not for unrestrained structures such as aircraft in flight (Chopra, 1995).

The N roots of equation (2.16) determine the N natural frequencies ω_n (n =1,2, …, N) of vibration. These roots of the characteristic equation are also known as eigenvalues, characteristic values, or normal values. When a natural frequency ω_n is known, equation (2.15) can be solved for the corresponding vector ϕ_n to within a multiplicative constant. The eigenvalue problem does not fix the absolute amplitude of the vectors ϕ_n, only the shape of the vector given by the relative values of the N displacements ϕjn (j=1, 2, …, N). Corresponding to the N natural vibration frequencies ωn of an N-DOF system, there are N independent vectors ϕn, which are known as natural modes of vibration, or natural mode shapes of vibration. These vectors are also known as eigenvectors, characteristic vectors, or normal modes.

In summary, a vibrating system with N DOFs has N natural vibration frequencies ωn (n=1, 2, …, N), arranging in sequence from smallest to largest (ω1<ω2<…<ωN); corresponding natural periods Tn; and natural modes ϕ_n. The subscript n denotes the mode number and the first mode (n=1) is also known as the fundamental mode (Chopra, 1995).

2.2.4. Mode Superposition Method

The most common and effective approach of seismic analysis for linear structural systems is the mode superposition method. After a set of orthogonalvectors have been evaluated, this method reduces the large set of global equilibrium equations to a relatively small number of uncoupled second order differential equations. The numerical solution of those equations involvesgreatly reduced computational time (Wilson, 2002).

The equation of equilibrium for a multi - degree of freedom (MDOF) system subjected to dynamic action is as follows:

$$\mathbf{m\ddot{u} + c\acute{u} + ku = p(t)} \quad (2.17)$$

The damping matrix **c** would not be needed in modal analysis of earthquake response; instead, modal damping ratios suffice and their numerical values can be estimated(Chopra, 1995).In civil engineering, the damping ratio generally takes a value in the range 0.01 to 0.1, and an assumed value of 0.05 is widely used in earthquake engineering.

The displacement **u** of an N-DOF system can be expressed as the superposition of the modal contributions(Chopra, 1995):

$$\mathbf{u(t)} = \sum_{n=1}^{N} \boldsymbol{\phi}_n \mathbf{q}_n(\mathbf{t}) \quad (2.18)$$

Substituting equation (2.18) into equation (2.17) gives:

$$\sum_{r=1}^{N} \boldsymbol{m\phi_r \ddot{q}_r}(\mathbf{t}) + \sum_{r=1}^{N} \boldsymbol{c\phi_r \dot{q}_r}(\mathbf{t}) + \sum_{r=1}^{N} \boldsymbol{k\phi_r q_r}(\mathbf{t}) = \mathbf{p(t)} \quad (2.19)$$

Premultiplying each term in this equation by ϕ_n^T gives

$$\sum_{r=1}^{N} \boldsymbol{\phi_n^T m\phi_r \ddot{q}_r}(\mathbf{t}) + \sum_{r=1}^{N} \boldsymbol{\phi_n^T c\phi_r \dot{q}_r}(\mathbf{t}) + \sum_{r=1}^{N} \boldsymbol{\phi_n^T k\phi_r q_r}(\mathbf{t}) = \boldsymbol{\phi_n^T}\mathbf{p(t)} \quad (2.20)$$

If the system has classical damping, the modal equations will be uncoupled. So, the above equation can be rewritten as:

$$\mathbf{M_n \ddot{q}_n + C_n \dot{q}_n + K_n q_n = P_n(t)} \quad (2.21)$$

where, $\mathbf{M_n = \phi_n^T m\, \phi_n}$ $\mathbf{K_n = \phi_n^T k\, \phi_n}$ $\mathbf{C_n = \phi_n^T c\, \phi_n}$ $\mathbf{P_n(t) = \phi_n^T P(t)}$

Equation (2.21) may be interpreted as the equation governing the response $q_n(t)$ of system with mass M_n, damping C_n, stiffness K_n, and exciting force $P_n(t)$. Therefore, M_n is called the generalized *mass* for the n^{th} natural mode, C_n is called the *generalized damping* for the n^{th} natural mode, K_n is the *generalized stiffness* for the n^{th} mode, and $P_n(t)$ is the *generalized force* for the n^{th} mode.

Dividing equation (2.21) by M_n gives:

$$\ddot{q}_n(t) + 2\zeta_n\omega_n\dot{q}_n(t) + \omega_n^2 q_n(t) = \frac{P_n(t)}{M_n} \quad \text{……………………………… (2.22)}$$

ζ_n is the damping ratio for the n^{th} mode. The damping ratio is usually not computed but is estimated based on experimental data for structures similar to the one being analyzed.

In summary, the set of N coupled differential equations in nodal displacements $u_j(t)$ has transformed to the set of N uncoupled equations in modal coordinates $q_n(t)$.

Generally, the following steps are used for modal analysis (Chopra, 1995):

1. Define the structural properties.
   a. Determine the mass matrix m and stiffness matrix k.
   b. Estimate the modal damping ratios ζ_n.
2. Determine the natural frequencies ω_n and modes ϕ_n.
3. Compute the response in each mode by the following steps:
   a. Set up equation (2.21) or equation (2.22) and solve for $q_n(t)$.
   b. Compute the nodal displacements $u_n(t)$ from equation (2.18).
   c. Compute the element forces associated with the nodal displacements $u_n(t)$.
4. Compute the contributions of all the modes to determine the total response. In particular, the nodal displacements u(t) and element forces.

Modal Expansion of Displacements and Forces for Earthquake Loading:

The spatial distribution of the effective earthquake forces is defined by **s**= **mι.** This force distribution can be expanded as a summation of modal inertia force distributions **Sn**(Chopra, 1995).

$$\mathbf{m\iota} = \sum_{n=1}^{N} \Gamma_n \mathbf{m} \phi_n \quad \text{…………………………………………………(2.23)}$$

where;

$$\Gamma_n = \frac{L_n}{M_n} \quad L_n = \phi_n^T \mathbf{m\iota} \qquad M_n = \phi_n^T m \phi_n \quad \text{……...………….. (2.24)}$$

Equation (2.24) for the coefficient Γ_n can be derived by pre multiplying both sides of equation (2.23) by ϕ_r^T and using the orthogonality property of modes. The contribution of the n^{th} mode to the excitation vector **mι** is

$$\mathbf{S_n = \Gamma_n m \phi_n} \text{..(2.25)}$$

which is independent of how the modes are normalized.

Equation (2.22) is specialized for earthquake excitation by replacing p(t) by $p_{eff}(t) = -m\iota\, \ddot{u}_g(t)$ to obtain:

$$\ddot{q}_n(t) + 2\zeta_n\omega_n\dot{q}_n(t) + \omega_n^2 q_n(t) = -\Gamma_n\ddot{u}_g(t) \text{..............................(2.26)}$$

The solution $q_n(t)$ can readily be obtained by solving this differential equation using appropriate methods.

2.2.5. Response Spectra Analysis

Response spectra analysis is the basic mode superposition method, which is restricted to linearly elasticanalysis, produces the complete time history response of joint displacements andmember forces.

There are computational advantages in using the response spectrum method ofseismic analysis for the prediction of displacements and member forces in structuralsystems. The method involves the calculation of only the maximum values of thedisplacements and member forces in each mode using smooth design spectra thatare the average of several earthquake motions (Wilson, 2002).

The response of a wide range of structures to a particular earthquake can be summarized using a response spectrum. The time-domain response of numerous single-degree of freedom (SDOF) systems having different natural periods is computed, and the maximum absolute displacement (or acceleration, or velocity) achieved is plotted as a function of the SDOF system period.

Therefore, the response spectrum shows the peak response of a SDOF structure to a particular earthquake, as a function of the natural period and damping ratios of the structure.

For seismic motion, the typical modal equation Eq. (2.26) is rewritten as

$$\ddot{q}_n(t) + 2\zeta_n\omega_n\dot{q}_n(t) + \omega_n^2 q_n(t) = -\Gamma_n\ddot{u}_g(t) \text{ .... (2.27)}$$

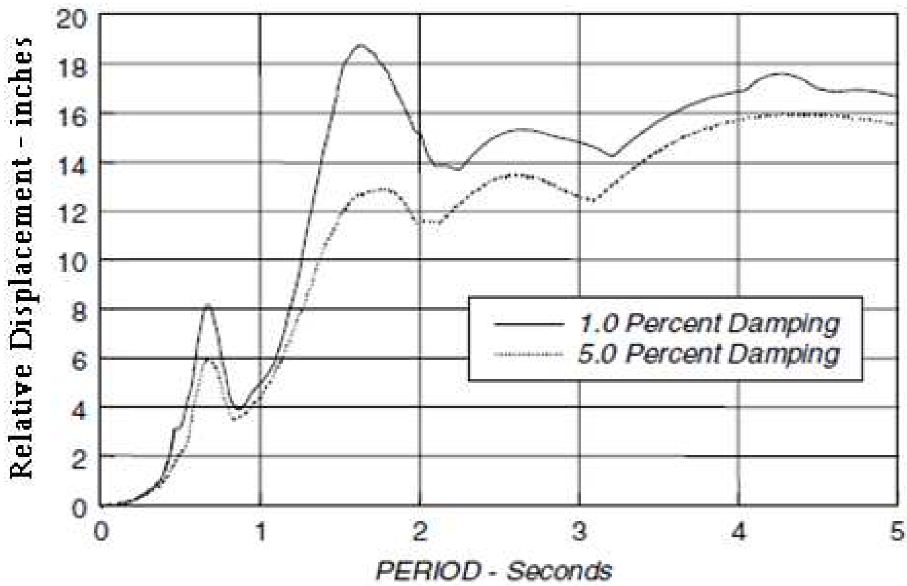
Two major problems must be solved to obtain anapproximate response spectrum solution to this equation. First, maximum peak forces and displacements must be estimated.Second, after the response has been solved, itis necessary to estimate the maximum response from earthquake motion acting at the same time.

It is possible to solve Equation (2.27) at various values of ω and plot a curve of the maximum peak response $q(\omega)_{max}$. For this accelerationinput, the curve is by definition the ***displacement response spectrum*** for theearthquake motion. A different curve will exist for each different value ofdamping (Wilson, 2002).

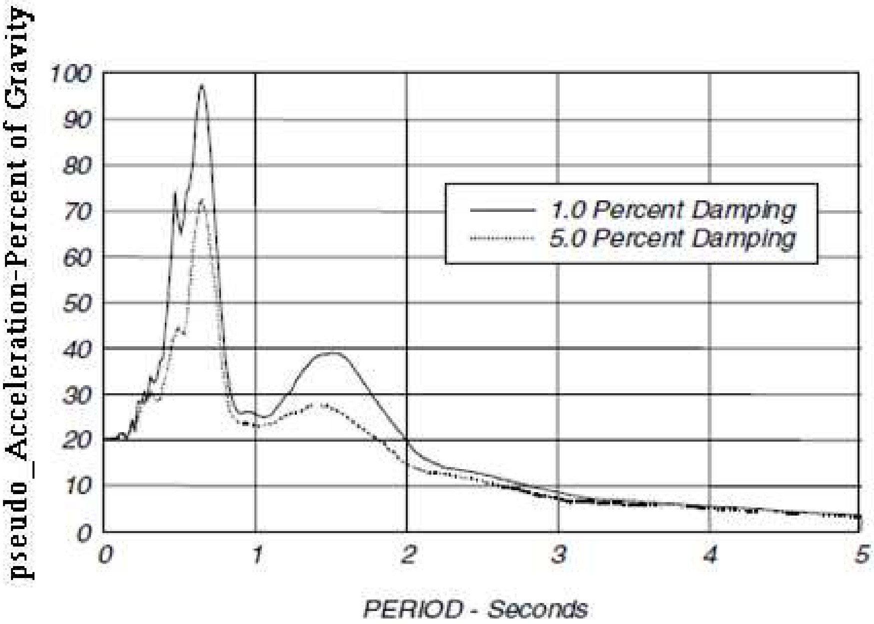
A plot of $\omega q(\omega)_{MAX}$is defined as the ***pseudo-velocity spectrum*** and a plot of$\omega^2 q(\omega)_{MAX}$is defined as the ***pseudo-acceleration spectrum***.

The three curves- displacement response spectrum, pseudo-velocity spectrum,and pseudo-acceleration spectrum- are normally plotted as one curve on speciallog paper. However, the pseudo-values have minimum physical significance andare not an essential part of a response spectrum analysis. The true values formaximum velocity and acceleration must be calculated from the solution of Equation (2.27).

It is apparent that all response spectrum curves represent the properties of the earthquake at a specific site and are not a function of the properties of the structural system. After estimation is made for the linear viscous damping properties of the structure, a specific response spectrum curve is selected. Figure 2.8 shows typical response spectrum given by Wilson, 2002.



a) Relative Displacement Spectrum $q(\omega)_{MAX}$-Inches



b) Pseudo-Acceleration Spectrum, $S(\omega)a = \omega^2 q(\omega)_{MAX}$– Percent of Gravity

Figure 2.8: Response Spectrum

To create a design spectrum, it is normal to compute spectra for several different earthquakes, then envelope and smooth them, resulting in a single curve that encapsulates the dynamic characteristics of a large number of possible earthquake accelerograms.

Design spectra are not even curves as shown in Figure 2.8 because they areintended to be the average of many earthquakes. Now, manybuilding codes specify design spectra in the form shown in Figure 2.9.

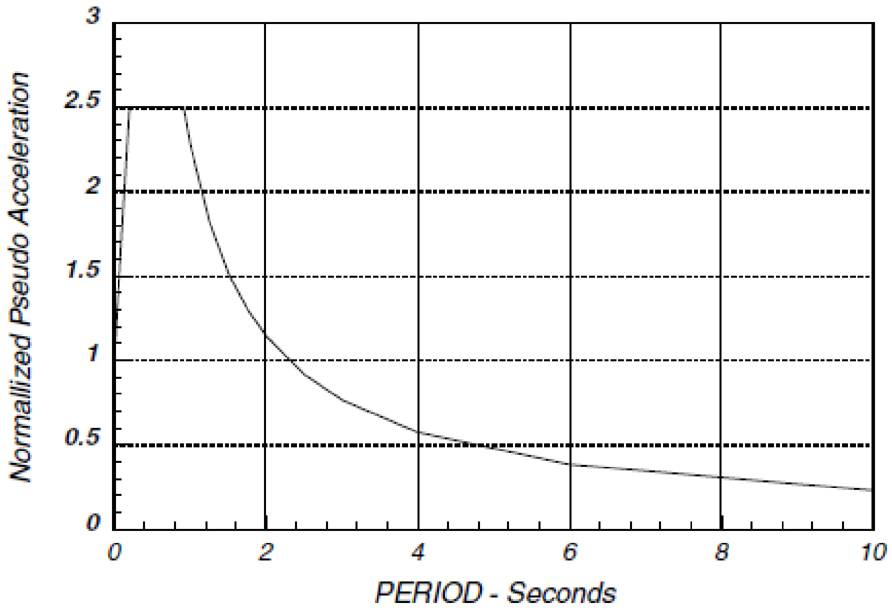


Figure 2.9: Typical Design Spectrum

In the design spectrum, there are three regions of response.Very stiff, short period structures simply move with the ground. At intermediate periods, there is dynamic amplification of the ground motion,and at long periods, the structure moves less than the ground beneath it.

It is also noticeable that the different soil types give rise to varying levels of amplification of the bedrock motions, and affect the period range over which amplification occurs (Wilson, 2002).

Calculation of Modal Response

The maximum modal displacement for a structural model can now be calculatedfor a typical mode nwith period $T_n$and corresponding spectrum response value$S(\omega_n)$. The maximum modal response associated with period $T_n$is given by Wilson, 2002 as:

$$\mathbf{q(T_n)_{max} = S(\omega_n)/\ \omega_n^2} \quad (2.28)$$

The maximum modal displacement response of the structural model is calculated from the following equation:

$$\mathbf{u_n = \phi_n\, q(T_n)_{max}} \quad (2.29)$$

Modal Combination Rules

Approximations must be introduced in combining the peak modal responses determined from the earthquake response spectrum because no information is available when these peak modal values occur.The assumption that all modal peaks occur at the same time and their algebraic sign is ignored provides an upper bound to the peak value of the total response (Chopra, 1995):

$$\boldsymbol{r_o} \leq \sum_{n=1}^{N} |\boldsymbol{r_{no}}| \qquad (2.30)$$

Where $\mathbf{r_o}$ is the peak response, and r_{no} (n=1, 2,…, N) is the peak modal responses.

This upper-bound value is usually too conservative. Therefore, this *absolute sum* (ABSUM) modal combination rule is not popular in structural design applications.

The *square-root-of-sum-of-squares* (SRSS) rule for modal combination is

$$\boldsymbol{r_o} \approx \left(\sum_{n=1}^{N} \boldsymbol{r_{no}^2}\right)^{1/2} \qquad (2.31)$$

The peak response in each mode is squared, the squared modal peaks are summed, and the square root of the sum provides an estimate of the peak total response. This modal combination rule provides excellent response estimates for only structures with well-separated natural frequencies.

The *complete quadratic combination* (CQC) rule for modal combination is applicable goes a wider class of structures as it overcomes the limitations of the SRSS rule. According to the CQC rule

$$\boldsymbol{r_o} \approx \left(\sum_{i=1}^{N}\sum_{n=1}^{N} \boldsymbol{\rho_{in}\, r_{io} r_{no}}\right)^{1/2} \qquad (2.32)$$

Each of the N^2 terms on the right side of this equation is the product of the peak responses in the i^{th} and n^{th} modes and the correction coefficient ρ_{in} for these two modes; ρ_{in} varies between 0 and 1 and ρ_{in} =1 for i = n. thus the above equation can be rewritten as

$$r_o \simeq \left(\sum_{n=1}^{N} r_{no}^2 + \underbrace{\sum_{i=1}^{N}\sum_{n=1}^{N}}_{i \neq n} \rho_{in} r_{io} r_{no}\right)^{1/2} \qquad (2.33)$$

It isinstructive to specialize for systems with the same damping ratio in all modes subjected to earthquake excitation with durations long enough to replace by $\zeta'_n = \zeta_n$. We substitute $\zeta_i = \zeta_n = \zeta$, introduce $\beta_{in} = \omega_i/\omega_n$.

Theequation for the correlation coefficient due to Der Kiureghian is

$$\rho_{in} = \frac{8\sqrt{\zeta_i \zeta_n}(\zeta_i + \beta_{in}\zeta_n)\beta_{in}^{3/2}}{(1-\beta_{in}^2)^2 + 4\zeta_i\zeta_n\beta_{in}(1+\beta_{in}^2) + 4(\zeta_i^2+\zeta_n^2)\beta_{in}^2} \qquad (2.34)$$

This equation also implies that $\rho_{in} = \rho_{ni}$, $\rho_{in} = 1$ for i =n or for two modes with equal frequencies and equal damping ratios. For equal damping $\zeta_i = \zeta_n = \zeta$, this equation simplifies to (Chopra, 1995)

$$\rho_{in} = \frac{8\zeta^2(1+\beta_{in})\beta_{in}^{3/2}}{(1-\beta_{in}^2)^2 + 4\zeta^2\beta_{in}(1+\beta_{in})^2} \tag{2.35}$$

Summary for Response Spectrum Analysis

The following steps are used for response spectrum analysis (Chopra, 1995):

1. Define the structural properties.
   a. Determine the mass matrix **m** and lateral stiffness matrix **k**.
   b. Estimate the modal damping ratios ζ_n.
2. Determine the natural frequencies ω_n (natural periods $T_n = 2\pi/\omega_n$) and natural modes ϕ_n of vibration.
3. Compute the peak response in the n^{th} mode by the following steps to be repeated for all modes, n = 1, 2, …, N:
   a. Corresponding to natural period T_n and damping ratio ζ_n, read D_n and A_n, the deformation and pseudo-acceleration, from the earthquake response spectrum or the design spectrum.
   b. Compute the floor displacements and story drifts.
   c. Compute the equivalent static lateral forces f_n.
   d. Compute the story forces-shear and overturning moment-and element forces-bending moments and shears- by static analysis of the structure subjected to lateral forces f_n.
4. Determine an estimate for the peak value r of any response quantity by combining the peak modal values r_n according to the SRSS rule, if the natural frequencies are well separated. The CQC rule should be used if the natural frequencies are closely spaced.

2.3. EBCS 8, 1995 Provisions for Dynamic Analysis

EBCS 8, 1995 applies to the design and construction of buildings in seismic regions. Its purpose is to insure that, in the event of earthquake,

- Human lives are protected
- Damage is limited
- Structures important for civil protection remain operational.

2.3.1. Seismic action

EBCS 8, 1995 provides that the country has subdivided in to four seismic zones, depending on the local hazard. Seismic zones with a design ground acceleration $\mathbf{a_g}$ not greater than 0.05g(zone 1 and zone 2) are low seismicity zones, for which reduced or simplified seismic design procedures for certain types or categories of structures may be used.

Clause 1.4.2 of EBCS 8, 1995 provides that the earthquake motion at a given point of the surface is generally represented by an elastic ground acceleration response spectrum, henceforth called 'elastic response spectrum'. Normalized elastic response spectra are shown in Figure 2.11, representing the free field ground-acceleration response.

The seismic action is described by two orthogonal components considered as independent and represented by the same response spectrum.

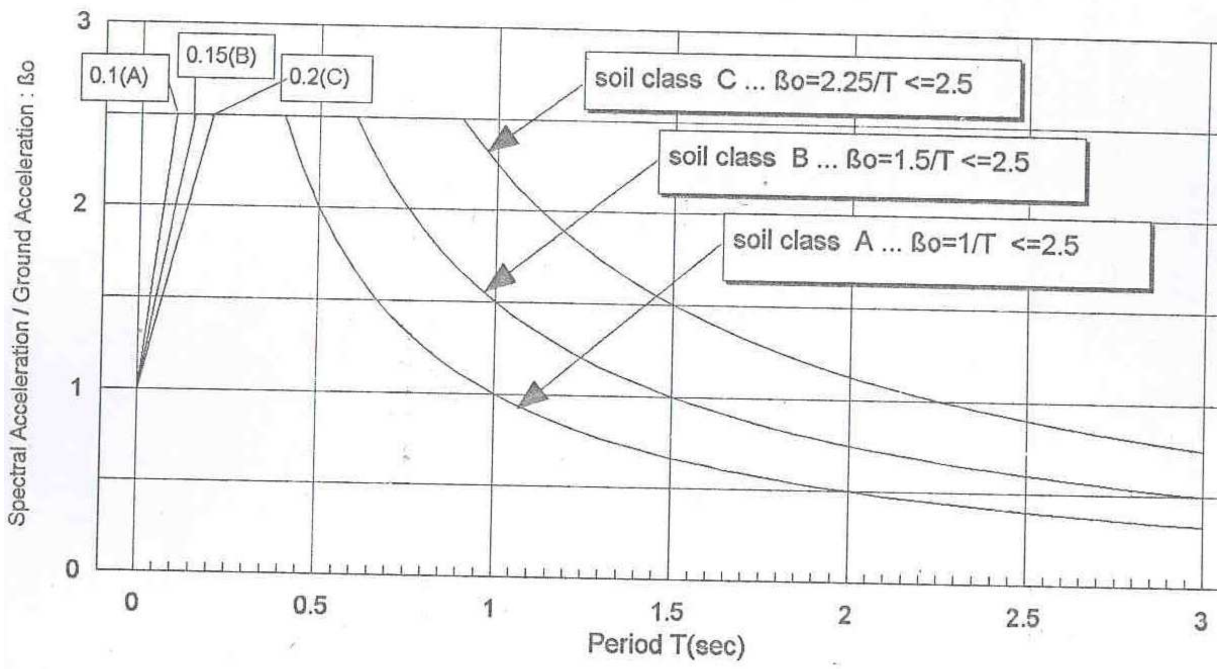


Figure 2.10: Normalized Elastic Response Spectra

2.3.2. Design spectrum

The capacity of structural systems to resist seismic actions in the nonlinear range generally permits their design for forces smaller than those corresponding to a linear elastic response.

To avoid explicit nonlinear structural analysis in design, the energy dissipation capacity of the structure, through mainly ductile behavior of its elements and/or other mechanisms, is taken into account by performing a linear analysis based on the building's fundamental period and a response spectrum, reduced with respect to elastic one, henceforth called "design spectrum". This reduction is accomplished by introducing the behavior factor γ.This behavior factor γ approximates the ratio of the minimum seismic forces that may be use in design with a conventional linear model, still ensuring a satisfactory response of the structure, to the seismic forces that the structure would experience if its response were completely elastic with 5% viscous damping. The values of the behavior factor γ forconcrete structures, which also accounts for the influence of viscous damping being different from 5%, are given for the various structural systems and according to various ductility levels in the clause 3.3.2 of the standard as:

$$\gamma = \gamma_o k_D k_R k_W \leq 0.70 \quad (2.36)$$

where γ_o is thebasic value of the behaviorfactor dependent on the structural type. γ_0 = 0.20 for frame systems.

K_D is the factor reflecting the ductility class.

K_R is the factor reflecting the structural regularity in elevation.

K_w is the factor reflecting the prevailing failure made in structural systems with walls.

For linear analysis, the design spectrum $S_d(T)$normalized by the acceleration of gravity g is defined by the following expression:

$$S_d(T) = \alpha\beta\gamma \quad (2.37)$$

The parameter α in equation (2.37) is the ratio of the design bedrock acceleration to the acceleration of gravity g and is given by:

$$\alpha = \alpha_o I \quad (2.38)$$

where α_o is the bedrock acceleration ratio for the site and depends on the seismic zone as given in Table 1.

zone	4	3	2	1
α_o	0.10	0.07	0.05	0.03

Table 1: Bedrock Acceleration Ratio α_o

I is the importance factor given by Table 2.

Importance category	Buildings	Importance factors I
I	Buildings, whose integrity during earthquakes are of vital importance for civil protection, e.g. hospitals, fire stations, power plants, etc.	1.4
II	Buildings whose seismic resistance is of importance in view of the consequences associated with a collapse, e.g. schools, assembly halls, cultural institutions, etc.	1.2
III	Ordinary buildings, not belonging to the other categories	1.0
IV	Buildings of minor importance for public safety, e.g. agricultural buildings, etc.	0.8

Table 2: Importance categories and importance factors for buildings

The parameter β is the design response factor for the site and is given by Equation (2.39)

$$\beta = \frac{1.2S}{T^{\frac{2}{3}}} \leq 2.5 \quad \ldots\ldots (2.39)$$

The parameter S in Equation (2.39) is the site coefficient for soil characteristics given in Table 3.

Subsoil class	A	B	C
S	1.0	1.2	1.5

Table 3: Site coefficient *S*

2.3.3. Dynamic Response Spectrum Analysis

EBCS 8, 1995 restricts the use of planar model to buildings complying with the criteria for regularity in plan. In this case, the analysis can be performed using two planar models, one for each main direction.

Buildings not complying with the criteria for regularity in plan and elevation shall be analyzed using spatial model.

This code provides that the response of all modes of vibration contributing significantly to the global response shall be taken into account. This provision may be satisfied by either of the following:

- By demonstrating that the sum of the effective modal masses for the modes considered amounts at least 90 % of the total mass of the structure
- By demonstrating thatall modes with effective modal masses greater than 5% of the total mass are considered

The response in two vibration modes i and j may be considered as independent of each other when their periods Ti and Tj satisfy the following condition:

$$Tj \leq 0.9Ti \quad \text{(2.39)}$$

Whenever all relevant modal responses can be regarded as independent of each other, the maximum value of the seismic action effect E_E may be taken as:

$$E_E = \sqrt{\sum E_{Ei}{}^2} \quad \text{(2.40)}$$

where E_E is the seismic action effect under consideration (force, displacement, etc.), E_{Ei} is the value of this seismic action effect due to the vibration mode i.

If the above condition is not satisfied, more accurate procedures for combining modal maxima (e.g. “Complete Quadratic Combination”) shall be adopted.

3. Development of the Program and Flowchart

3.1. Programming language

A **programming language** is an artificial language designed to express computations that can be performed by a machine, particularly a computer. Programming languages can be used to create programs that control the behavior of a machine, to express algorithms precisely, or as a mode of human communication.A programming language is a notation for writing programs, which are specifications of a computation or algorithm.

A programming language's surface form is known as its syntax. Most programming languages are purely textual; they use sequences of text including words, numbers, and punctuation, much like written natural languages. On the other hand, there are some programming languages which are more graphical in nature, using visual relationships between symbols to specify a program.

The syntax of a language describes the possible combinations of symbols that form a syntactically correct program. Most programming languages have an associated core library (sometimes known as the 'standard library', especially if it is included as part of the published language standard), which is conventionally made available by all implementations of the language. Core libraries typically include definitions for commonly used algorithms, data structures, and mechanisms for input and output.

Many languages have been designed from scratch, altered to meet new needs, combined with other languages, and eventually fallen into disuse. Although there have been attempts to design one "universal" programming language that serves all purposes, all of them have failed to be.

Computer programming (often shortened to **programming** or **coding**) is the process of designing, writing, testing, debugging/troubleshooting, and maintaining the source code of computer programs. This source code is written in a programming language. The purpose of programming is to create a program that exhibits a certain desired behavior. The process of writing source code often requires expertise in many different subjects, including knowledge of the application domain, specialized algorithms and formal logic.

In addition to the programming languages, there are also **developed softwares** that can be used to develop programs. These are widely used to develop a program because these softwares have built-in functions and reduce additional effort to define some functions. For example, Scilab or Matlab has functions that can transpose andinverse a matrix. This reduces additional effort to define the transpose and inverse of a matrix.

3.2. What is Scilab

Scilab is a computational and programming environment chosen for this thesis work. It is developed since 1990 by researchers from INRIA (French National Institute for Research in Computer Science andControl)and ENPC (National School of Bridges and Roads);it is now maintained and developed byScilab Consortium.

Scilab is open source anddistributed freely through the Internet since 1994, Scilab is currently being used in educational and industrial environments around the world (Satish Annigeri,2008).

Scilab is software for numerical mathematics and scientific visualization. It is capable ofinteractive calculations as well as automation of computations through programming. It providesall basic operations on matrices through built-in functions so that the trouble of developing andtesting code for basic operations are completely avoided. Its ability to plot 2D and 3D graphshelps in visualizing the data we work with. All these make Scilab an excellent tool for teaching,especially those subjects that involve matrix operations. Further, the numerous toolboxes that areavailable for various specialized applications make it an important tool for research.

Scilab has sophisticated data structures (includinglists, polynomials, rational functions, linear systems...), an interpreter and a high-levelprogramming language.

Scilab has been designed to be an open system where the user can define new data types andoperations on these data types.

A key feature of the Scilab syntax is its ability to handle matrices: basic matrix manipulationssuch as inverse, extraction or transpose are immediately performed as well as basic operationssuch as addition or multiplication.

Scilab has an open programming environment where the creation of functions and librariesof functions is completely in the hands of the user. Functions are recognized asdata objects in

Scilab and, thus, can be manipulated or created as other data objects. For example,functions can be defined inside Scilab and passed as input or output arguments of other functions.

3.3. Preparation of Input data for the Program

After this, the program developed is called '**SAF**' to mean Structural Analysis of Frames.

The first task in developing a program is to clearly define the way data is organizedwithin the program (Satish Annigeri, 2008). The input data foranalysis of a plane frame for earthquake and static loading shall be organized as follows:

Nodal points (joints) play a fundamental role in the analysis of any structure. Joints are the points of connection between the elements, and they are the primary locations in the structure at which the displacements are known or are to be determined. These points are prepared according to the modeling considerations discussed in section 2.1.1 of this document. The coordinate matrix has to be inserted for the program (SAF) as an input with Variable name '**xz'**. It has size of nx2, where '**n**' is the number of nodes in the frame. It isassumed that row number corresponds to the node number. Thus row '**i**' stores thecoordinates of node number '**i**'. Columns 1 and 2 correspond to the**x**and **z**coordinates of the node, respectively.

The member property matrix contains the material property and section property of a member. If different kinds of materials and section properties are used for the members of the frame, each property has to be inserted as an input with variable name '**mprop**'.To model axial members that do not transmit moments at the ends, the geometric section property,second moment of area with referenceto the neutral axis of the section (I), is set to zero. The member property matrix has size of npx3, where **'np'** is the number of memberproperties requiredone for each type. Columns 1, 2 and 3 represent Modulus ofelasticity (E), Area of cross section (A) and Second moment of area with referenceto the neutral axis of the section (I) respectively.

The frame element (member) is used to model beam, column and brace behavior in planar frame. Each element has its own local coordinate system for defining section properties and loads, and for interpreting output. A frame element is represented by a straight line connecting two joints and inserted as an input with variable name '**conn**'. The connectivity matrix has size of mx3, where '**m**' is the number of members in the frame.Row number corresponds to the member number. Columns 1 and 2 correspond tothe start and end node numbers of the member, respectively. Column 3 correspondsto Member Property ID, referring to the row number of the material property storedin the matrix **'mprop'**.

Supports (nodes with constrained degrees of freedom) are nodes with known displacements and are inserted as an input with variable name '**bc**'. It has sizeof nsx3, where **'ns'** is the number of supports. Columns 1, 2 and 3 correspond to theconstraint code for displacement along x-axis (dx), displacement along z-axis (dz)and rotation about y-axis (ry) respectively. Constraint code 1 indicates that thedegree of freedom (dof) is constrained and 0 indicates unconstrained.

Joint loads are loads applied directly on the frame joints, and are inserted as an input with variable name '**jtloads**'. It has size of njlx4, where **'njl'** is the number of joint loads applied on the frame joints. Column1 corresponds to the node number where the joint load is applied. Columns 2, 3and 4 correspond to component of load along x-axis (Fx), component of load along z-axis (Fz) and Moment about y-axis (My) respectively. Joint load components must be defined in global axes.

The sign convention is:

- Positive loads are oriented in the positive direction of global coordinate axes.
- Positive bending moments are counter clockwise moments.

Member loads, are loads applied on members, has to be prepared as input with variable name '**loads**'. It has sizeof nmlx4, where 'nml' is the number of member loads. Column 1 corresponds to the member number on which member loads are applied. Column 2 corresponds to the member load intensity (w) at start node of the member. Column 3 corresponds to the member load intensity (w) at end node of the member. Column 4 corresponds to the code number indicating whether the load intensity is in the direction of the member axes or global axes. 1 at column 4 indicates the load is in member axes, 2 at column 4 indicates the load is in the direction of global z-axes and 3 at column 4 indicates the load is in the direction of global x-axes. Sign convention follows the right-hand rule.

In a dynamic analysis, the mass of the structure is used to compute inertial forces. The mass contributed by the Frame element is lumped at the two joints of the element. In addition to this mass, the mass of the partitions and floor finishes are calculated and lumped at the relevant joints.The total mass is apportioned to the two joints in the same way as a similarlydistributed transverse load would cause reactions at the ends of a simply- supported beam. The effects of end releases are ignored when apportioning mass.Thelumped masshas to be prepared in the above form and inserted as an input with variable name ‘**mass**’. It has size of nmx2, where ‘nm’ is the number of lumped masses.Column 1 corresponds to the node number and column 2 corresponds to the lumped mass of the node.

The location of the site according to the zoning provided by EBCS 8, 1995 has to be inserted as input with variable name ‘**zone**’.The influence of local ground conditions on the seismic action shall generally be accounted for by considering the three subsoil classes A, B, C. it has to be inserted as an input with symbol ‘**sclass**’.This is the subsoil condition of the site according to EBCS 8. Code 1 corresponds to soil class A, 2 corresponds to soil class B and 3 correspond to soil class C.

Buildingsare generally classified into four importance categories, which depend on the size of the building, on its value and importance for the public safety and on the possibility of human losses in case of a collapse. The importance category has to be inserted as an input with variable name ‘**importance**’ based on the classifications of Table 2.

The behavior factor γ approximates the ratio of the minimum seismic forces that may be use in design with a conventional linear model, still ensuring a satisfactory response of the structure, to the seismic forces that the structure would experience if its response were completely elastic with 5% viscous damping. The values of the behavior factor γ, which also accounts for the influence of viscous damping being different from 5%, has to be calculated using Equation (2.36) and inserted as input with variable name ‘**behavior**’.

Data for static analysis:

If only the responses ofa frame for static loads are desired, the following input data must be prepared as explained above.

i. Coordinates matrix (xz),
ii. Member property matrix (mprop),
iii. Connectivity matrix (conn),
iv. Boundary constraint matrix (bc),
v. Joint loads matrix (jtloads) and
vi. Member load matrix (loads).

Data for dynamic analysis:

If only the responses of a frame due to ground motion are desired, the following input data must be prepared as explained above.

i. Coordinates matrix (xz),
ii. Member property matrix (mprop),
iii. Connectivity matrix (conn),
iv. Boundary constraint matrix (bc),
v. Lumped masses at nodes (mass)
vi. Location of the site (zone)
vii. Subsoil condition (sclass)
viii. Importance category of the structure (importance) and
ix. Behavior factor of the system (behavior)

3.4. Program Development

3.4.1. Location Matrix for a Plane Frame Analysis

In a plane frame, the number of degrees of freedom (dof) per node is 3, namely dx, dz and ry,and therefore the total number of degrees of freedom is 3 times the number of nodes (**n**).

However, every structure has supports with certain degrees of freedom constrained (otherwisethe structure would be a free body capable of undergoing rigid-body motion). These constraineddof have zero displacements. As a result, the number of unknown degrees of freedom (**ndof**) is smaller than 3x**n**.

In the direct stiffness matrix method, the portion of the stiffness matrix corresponding tozero displacements is not assembled, as it does not affect the calculation of the nodaldisplacements.

A generalized representation of the dof numbering is called the location matrix. It has a size **n**x3, where '**n**' is the number of nodes in the plane frame.

Columns 1 to 3 store the constraintcode for displacements along the 3 possible dof, namely, dx, dz and ry for each node. The code is0 for unconstrained dof and 1 for constrained dof.

The location matrix '**lm**' is compiled in 2 stages. In the first stage, '**lm**' is initialized to a zeromatrix, implying that all dof are unconstrained. The number of nodes with constrained dof isavailable in column 1 of matrix '**bc**'. For each of these nodes, the zeros in the corresponding rowsof the location matrix are replaced by the constraint codes from '**bc**'.

In the second stage, the variable representing the number of dof of the plane frameis initialized to 0 and '**lm**' is processed row by row, For each constraint code 0 (unconstraineddof), '**ndof**' is incremented by 1 and stored in the place of the 0. If the constraint code is 1(constrained dof), it is set to 0 indicating the corresponding displacement to be 0.

At the end of this stage, '**ndof**' will be the total number of dof of the plane frame and '**lm**'will have zeros for a dof with zero displacement or a unique dof number for an unconstraineddof. The size of structure stiffness matrix must therefore be **ndof** x **ndof**.

3.4.2. Stiffness Matrix of a Plane Frame Element

The stiffness matrix of a plane frame member with reference to its local axes is of size 6x6 and is given as shown below.

$$K=\begin{bmatrix} EA/L & 0 & 0 & -EA/L & 0 & 0 \\ 0 & 12EI/L^3 & 6EI/L^2 & 0 & -12EI/L^3 & 6EI/L^2 \\ 0 & 6EI/L^2 & 4EI/L & 0 & -6EI/L2 & 2EI/L \\ -EA/L & 0 & 0 & EA/L & 0 & 0 \\ 0 & -12EI/L^3 & -6EI/L^2 & 0 & 12EI/L^3 & -6EI/L^2 \\ 0 & 6EI/L^2 & 2EI/L & 0 & -6EI/L^2 & 4EI/L \end{bmatrix}$$

This can be generated by first defining a zero matrix of size 6x6, then defining the elements of the upper triangular matrix and finally copying the upper triangle into the lower triangle.This function takes the length of the member as one of the input. Therefore, a function to calculate the length of a member is needed. Later on,the direction cosines of the member are needed to do arelated calculation. A single function to do both these tasks will be written.With this function available, the local stiffness matrix of any member can be computed by first calculating its length, extracting its material properties from '**mprop**' and then computing its local stiffness matrix.

The local stiffness matrix is computed with reference to the local axis of a member, which may or may not be parallel to the global axes. Therefore, it is necessary to transform the member stiffness matrix from local axes to global axes before it is superposed with stiffness matrices of other members in order to assemble the structure stiffness matrix. This transformation requires the rotation matrix of the member and is expressed in terms of the direction cosines of the member. The rotation matrix for a plane frame member is as given below:

$$r = \begin{bmatrix} Cx & Cz & 0 & 0 & 0 & 0 \\ -Cz & Cx & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & Cx & Cz & 0 \\ 0 & 0 & 0 & -Cz & Cx & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where Cx and Cz are the direction cosines of the member and are calculated as $Cx = dx/L = (X_2 - X_1)/L$ and $Cz = dz/L = (Z_2 - Z_1)/L$ where $L = \sqrt{dx^2 + dz^2}$.

The global stiffness matrix of a plane frame member is calculated as ***K* = r^T^*k*r**,where **k**is local stiffness matrix and **r**is the rotation matrix of the member. Thus writing a function to calculate the global stiffness matrix of a plane frame member is straightforward once the functions to calculate the local stiffness matrix and rotation matrix are available.

Assembling the Plane Frame Structure Stiffness Matrix:

The structure stiffness matrix of a plane frame is obtained by computing the global stiffness matrix for an individual member of the plane frame and superposing it with the structure stiffness matrix. To be able to so, one needs to determine a mapping between the rows and columns of the global stiffness matrix of a member and those of the structure stiffness matrix. This is done by writing the dof numbers for the rows and columns of the member global stiffness matrix and superposing the elements with the corresponding elements of the structure stiffness matrix. Thus, knowing the numbers of the start and end nodes of a member and the corresponding dof numbers from the location matrix '**lm**' is necessary.The global stiffness matrix of one member is superposed onto the existing structure stiffness matrix, which is initialized to zero to start with.To compute the structure stiffness matrix considering all members, we need to call the function, which assembles the structure stiffness matrix, inside aloop.

3.4.3. Assembling the Plane Frame Mass Matrix and Static Condensation

One of the tasks to perform dynamic analysis is assembling the mass matrix of the system. The mass matrix of a plane frame is obtained by calling the lumped mass for an individual node of the plane frame. This matrix has the same size as the structure stiffness matrix.

The mass matrix is compiled in the following stages. First, the mass matrix '**m**' is initialized to zero, implying all the translational and rotational masses are zero. Then, the node number, at which each mass is lumped, is extracted from column 1 of the input data '**mass**'. Thirdly, translational dof number of the node in global x-axes has been extracted from column 1 of the location matrix. Lastly, the mass of the node is assembled in their respective translational dof in x-axes. At the end of this stage, the translational masses in z-axes and rotational masses remain zero.

Static Condensation:

After assembling the structure stiffness matrix and mass matrix, the structure stiffness matrix will be condensed and the mass matrix will be assembled by extracting only the translational masses in global x-axes.

The static condensation method is used to eliminate from dynamic analysis those DOFs of a structure to which zero mass is assigned; however, all the DOFs are included in the static analysis. Typically, the mass of the structureis idealized as concentrated in point lumps at the nodes, and the mass matrix contains zero diagonal elements in the rotational DOFs. These are the DOFs that can be eliminated from the dynamic analysis of the structure if the dynamic excitation does not include any external forces in the rotational DOFs, as in the case of earthquake excitation. Even if included in formulating the stiffness matrix, the vertical DOFs of the building can also be eliminated from dynamic analysis-because the inertial effects associated with the vertical DOFs of building frames are usually small-provided that the dynamic excitation does not include vertical forces at the nodes, as in the case of horizontal ground motion (Chopra,1995).

The equation of motion for a system excluding damping from Equation(2.3) is written in partitioned form:

$$\begin{bmatrix} mtt & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{u}t \\ \ddot{u}o \end{Bmatrix} + \begin{bmatrix} ktt & kto \\ kot & koo \end{bmatrix} \begin{Bmatrix} ut \\ uo \end{Bmatrix} = \begin{Bmatrix} Pt(t) \\ 0 \end{Bmatrix} \quad \text{(3.1)}$$

Where, uo denotes the DOFs with zero mass and ut the remaining DOFs. The two partitioned equations are:

$m_{tt}\ \ddot{u}_t + k_{tt}u_t + k_{to}\ u_o = P_t(t)$ …………………………………………….……… (3.2a)

$k_{ot}u_t + k_{oo}\ u_o = 0$ ……………………………………………….………………. (3.2b)

Because no inertia terms or external forces are associated with u_o, Equation (3.2b) permits a static relationship between u_o and u_t:

$u_o = -k_{oo}^{-1}k_{ot}u_t$…………………………………………………………………... (3.3)

Substituting Equation(3.3) into Equation (3.2a) gives

$m_{tt}\ \ddot{u}_t + \acute{K}_{tt}u_t = P_t(t)$……………………………………………………………….. (3.4)

Where, $\acute{K}_{tt}$ is the condensed stiffness matrix given by:

$\acute{K}_{tt} = k_{tt} - k_{ot}^{T}k_{oo}^{-1}k_{ot}$…………………………………..………………………….. (3.5)

This task is compiled in four stages. In the first stage, the number of translational degrees of freedom is calculated by counting the non-zero values of column 1 of the location matrix. Second, the diagonal mass matrix '**mtt**' is calculated by ignoring the zero values of the mass matrix (translational masses inz-axes and rotational masses). Third, the location matrix is arranged by bringing the translational dof in x-axes to the top and next the other degrees of freedoms are arranged from lower to higher as one column matrix (vector) '**lmd**'. Following this, the structure stiffness matrix is arranged in the order of previously defined vector '**lmd**' (i.e. bringing the horizontal translational DOFs to the top rows and left columns of structure stiffness matrix). Lastly, the structure stiffness matrix is condensed using Equation (3.5).

3.4.4. Eigenvector Analysis

Eigenvector analysis determines the undamped free vibration mode shapes and frequencies of the system. These natural Modes provide an excellent insight into the behavior of the structure. They can also be used as the basis for response-spectrum analyses.

Eigenvector analysis involves the solution of the generalized eigenvalue problem given by:

$$[\mathbf{k}-\omega_n^2\mathbf{m}]\phi_n=\mathbf{0}$$ ………………………………….………..…………… (3.6)

where ***K*** is the stiffness matrix, ***m*** is the diagonal mass matrix, ω_n^2 is the diagonal matrix of eigenvalues, and ϕ_n is the matrix of corresponding eigenvectors (modeshapes).Each eigenvalue-eigenvector pair is called a natural Vibration Mode of the structure. The eigenvalue is the square of the circular frequency, ω_n, for that Mode.

The number of Modes actually found, ***n***, is limited by the number of mass degrees of freedom in the model. A mass degree of freedom is anydegree of freedom that possesses translational mass.All the Modes that are actually found will be available for any subsequent response-spectrum analysis processing.

3.4.5. Response-spectrum Analysis

The dynamic equilibrium equations associated with the response of a structure to ground motion are given by:

$$m\ddot{u}(t) + c\acute{u}(t) + ku(t) = -m\imath\, \ddot{u}_g(t) \quad (3.7)$$

where ***k***is the stiffness matrix; ***c***is the proportional damping matrix; ***m***is the diagonal mass matrix; **u**, **ú** and **ü** are the relative displacements, velocities, and accelerations with respect to the ground;and $\ddot{u}_g$ is the uniform ground acceleration.

Response-spectrum analysis seeks the likely maximum response to these equations rather than the full time history. The earthquake ground acceleration is given as a digitized response-spectrum curve of pseudo-spectral accelerationresponse versus period of the structure. No correspondence between two different response quantities is available and no information is available as to when this extreme value occurs during the seismic loading, or as to what the values of other response quantities are at that time.

Response-spectrum analysis is performed using mode superposition (Wilson, 2002). Modes will have been computed using eigenvector analysis.

Theresponse-spectrum analysis is done using the design spectrum given by EBCS 8, 1995. The equation of the three curves for each soil class is programmed to get the ratio of spectral acceleration to ground acceleration (β_0) for each mode as a function of period and soil class. The modal acceleration for each mode is calculated by relating the bedrock acceleration ratiofor the site (α_0), importance factor (I) and behavior factor (γ) as:

$$An = \alpha*\beta_0*\gamma *9.81 \quad (3.8)$$

where

$$\alpha = \alpha_0*I \quad (3.9)$$

After getting the modal accelerations, the equivalentstatic forces for the ground acceleration will be determined by:

$$f_n = S_nA_n \quad (3.10)$$

$$\mathbf{f_{jn} = \Gamma_n m_j \phi_{jn} A_n} \quad (3.11)$$

where $\mathbf{f_n}$ is the vector of forces $\mathbf{f_{jn}}$ at the various joints, j=1, 2… N; $\mathbf{S_n}$is the modal expansion of earthquake forces defined by:

$$\mathbf{S_n = \Gamma_n m_j \phi_{jn}} \quad (3.12)$$

where $\mathbf{m_j}$ is the lumped mass at joint **j;**$\mathbf{\phi_{jn}}$ is the modal shapes of vibration of mode **n;** and$\mathbf{A_n}$ is the modal acceleration defined by Equation (3.8) (Chopra, 1995).

These equivalent modal loads are combined to get the equivalent nodal loads using CQC method or SRSS method given in section 2.2.6of this document. The CQC method takes into account the statistical coupling between closely spaced Modes caused by modal damping. SRSS method does not take into account any coupling of Modes as do the CQC method. These two methods are programmed and the results are combined based on the closeness of the modes.

3.4.6. Assembling the Load Vector

After finding the joint stiffness matrix, the next step in the static analysis is to consider the loads on the structure. It is convenient initially to handle the loads at the joints and the loads on the members separately. The reason for doing so is that the joint loads and member loads are treated in different ways. The joint loads are ready for immediate placement into a vector of actions to be used in the solution, but the loads on the members are taken into account by calculating the fixed end actions that they produce. These fixed-end actions may then be transformed into equivalent joint loads and combined with the actual joint loads on the structure (Weaver and Gere, 1980).

The load vector has the same number of rows as the number of degrees of freedom of the structure. Each element of the load vector represents the load applied corresponding to a dof number. Loads applied on a plane frame can be either loads applied at the nodes of the frame or loads applied on the members. The two different types of loads have to be processed differently before assembling the load vector.

Joint loads are an input data to the program and are specified as components in global coordinate system (components along x, z-axes and moment about y-axis). After knowing the node number at which the load is applied, it is easy to identify the dof numbers corresponding to the node from the location matrix. The components of the load at a node are then superposed with the corresponding element in the load vector.

Since fixed end actions from member loads are in member axes, they are transformed to global axes using therotation matrix. Then, the load vector due to one of the member loadis calculated andsuperposed onto the previous load vector.To assemble the load vector due to all member loads, the function, which performs this task, has to be called once foreach member load, within a loop. The load vector **P** has to be initialized to zero at the start. The contribution to the loadvector due to joint loads will first be assembled and after that, the contribution due to themember loads will be assembled.

3.4.7. Nodal Displacements

The nodal (joint) displacements for unrestrained nodes are determined by solving the stiffness equation:

$$\mathbf{U} = \mathbf{k}^{-1} * \mathbf{P} \quad \text{(3.13)}$$

where **P** is the load vector;**k** is the structure stiffness matrix; and **U** is the nodal displacements.

For static analysis of a frame, theload vector **P**is determined by superposing the actual joint loads with equivalent joint loads gained from member loads as shown in Section 3.3.6. For dynamic analysis, the load vector **P**is determined from response spectrum analysis given in section 3.3.5. This task needs to write a single function that calls the functions, which assembles the structure stiffness matrix and the load vectorof a plane frame, in the right sequence and solves the stiffness equation for the nodal displacements.

This function is the super function that performs the complete task of assembling the structure stiffness matrix, assembling the load vector and solving the stiffness equation to obtain the nodal displacements.After this function is called, the nodal displacements are known and the task that remains is to extract the member end forces and support reactions from the calculated nodal displacements.

3.4.8. Extracting Member End Forces and Support Reactions

The **Member End forces** are the forces and moments of a member endsin member axes that are gained by the equation:

$$\mathbf{f} = \boldsymbol{k} * \boldsymbol{u} - \boldsymbol{Peq} \quad (3.14)$$

where $\boldsymbol{k}$ is the local stiffness matrix of a member;$\boldsymbol{Peq}$is the equivalent joint load at the member ends (for static member loads); and $\boldsymbol{u}$ is the displacement vector of the member ends in local axes given by:

$$\boldsymbol{u} = \mathbf{r}^*\mathbf{U} \quad (3.15)$$

where **r** is the rotation matrix of the member; and **U** is the displacement of the member ends in global axes.

These member end forces are:

- P, the axial force
- V, the shear force
- M, the bending moment in the x-z plane (about the y-axis)

The sign convention is:

- Positive forces are oriented in the positive direction of the element local coordinate axes.
- Positive bending moments are counterclockwise moments.

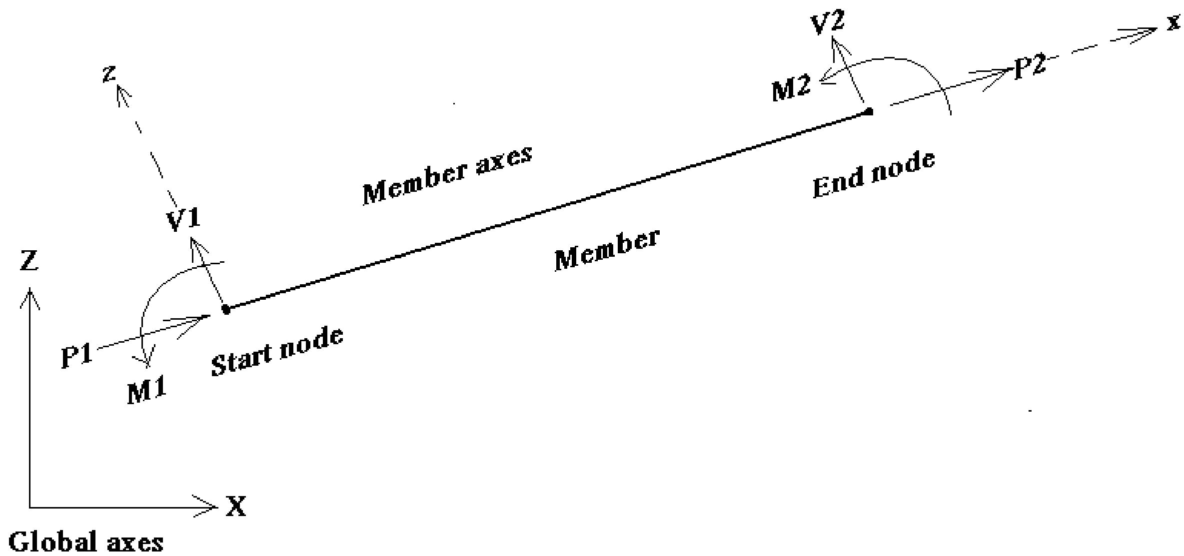


Figure 3.1: Sign Convention for Member End Actions

Once the nodal displacements are available, calculating the member end forces in local axes is possible if we know the displacements pertaining to the nodes of the member concerned. This information is available in the location matrix and can easily be obtained. To express the member end forces in local axes, the displacements of the member ends are transformed from global to local axes.

Support Reactions:

If the displacement of a joint along any of its degrees of freedom has a known value,zero (e.g., at support points), a**restraint** must be applied to that degree of freedom.

The force or moment along the degree of freedom that is required to enforce the restraint is called the **reaction**, and it is determined by the analysis. The reaction includes the forces (or moments) from all elements connected to the restrained degree of freedom, as well as all loads applied to the degree of freedom.

$$\mathbf{R = F - P_{actual}} \quad \text{.......... (3.16)}$$

where $\mathbf{P_{actual}}$is the actual applied joint loads on the support; **F** is the forces (or moments) from all elements connected to the restrained degree of freedom in global axes given by:

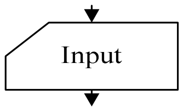
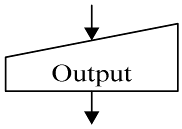
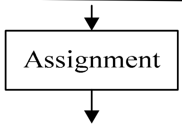
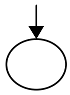
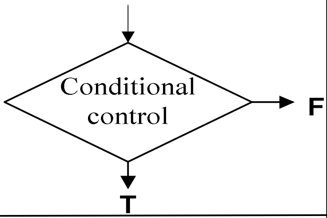
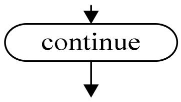
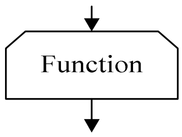
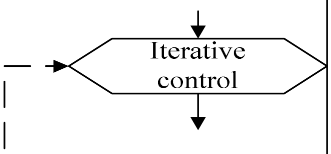
$$\mathbf{F = r * f} \quad \text{.......... (3.17)}$$

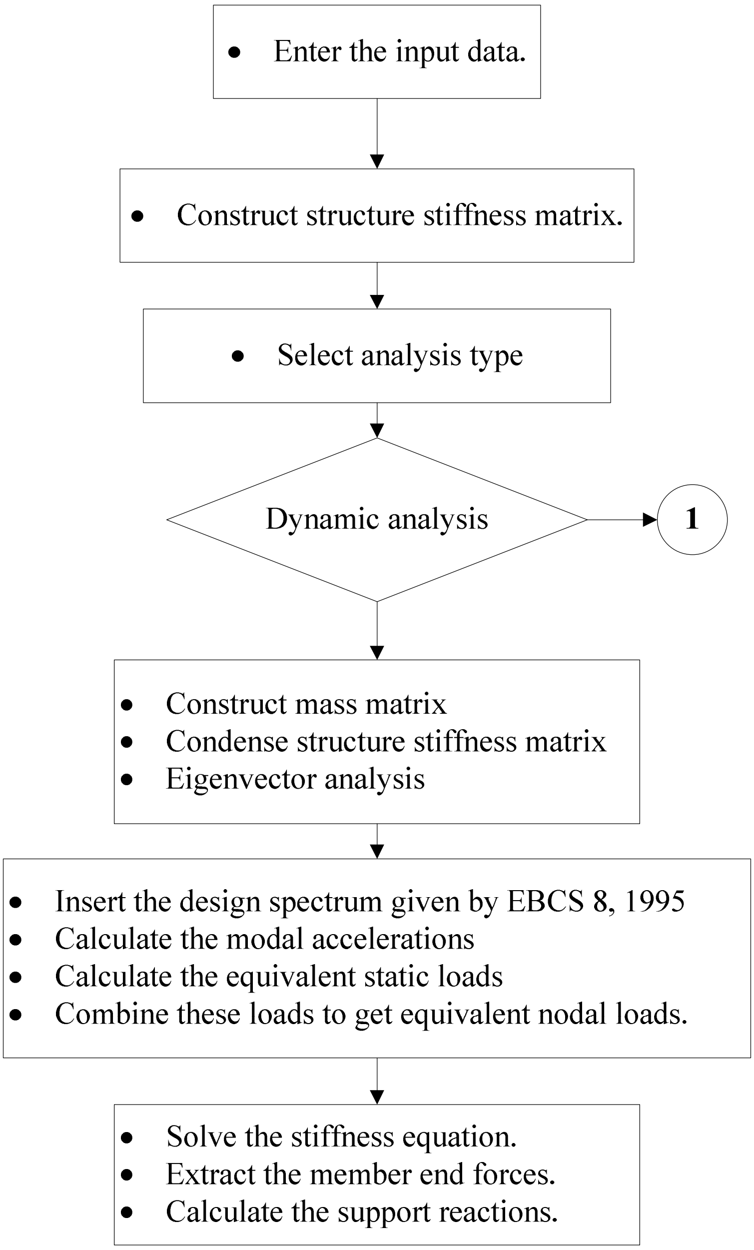
where **r** is the rotation matrix of the member; and **f** is the member end forces from all elements connected to the restrained degree of freedom in local axes calculated above in Equation (3.14).

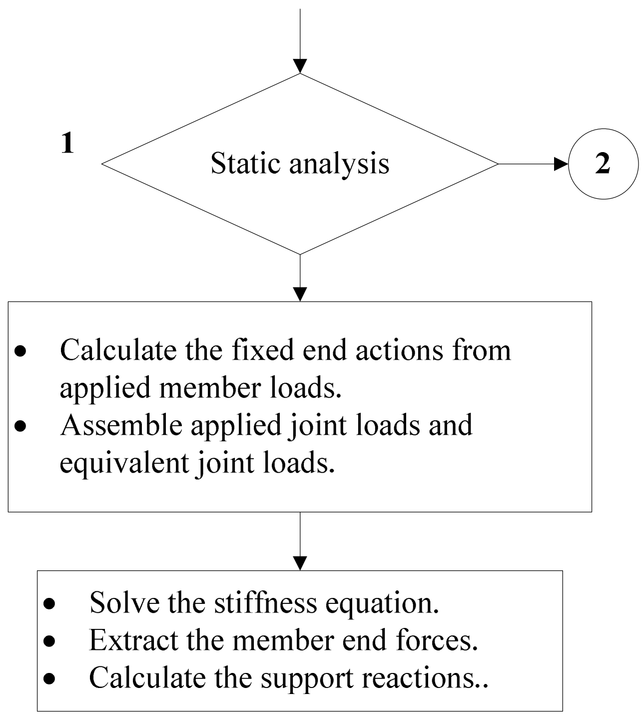
3.5. Flowchart of the Program

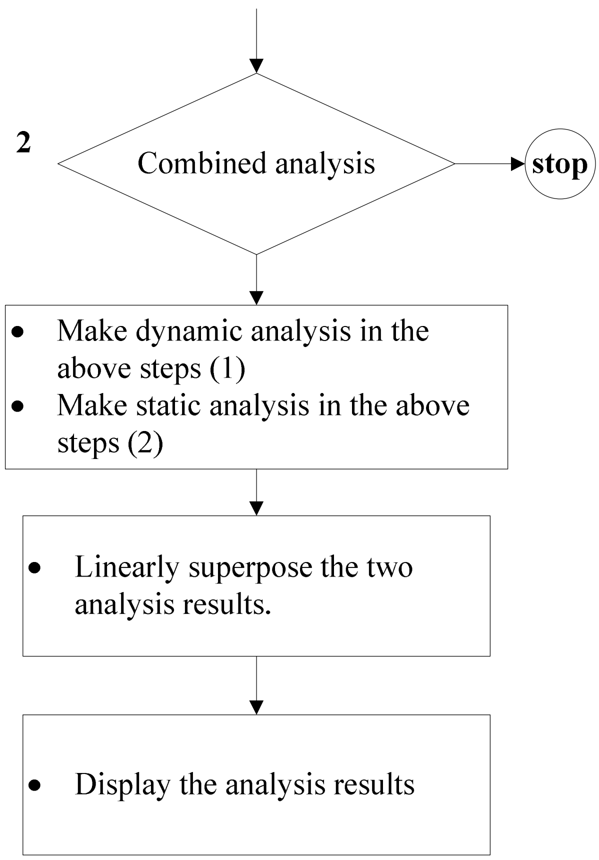
The Flowchart for the analysis of frames for dynamic and/or static loads are presented in this section. Only flowchart to show the steps used to develop the program is presented here. Detailed flowchart for the program is presented in the Appendix I.

Symbols and their meanings as used in the Flowchart

Types of statement	Flow chart symbol
a. Input	
b. output	
c. Assignment	
d. Unconditional control	
e. Conditional control	
f. Continue	
g. Function	
h. Iterative control	


Enter the input data.
Construct structure stiffness matrix.
Select analysis type
Dynamic analysis
1
Construct mass matrix
Condense structure stiffness matrix
Eigenvector analysis
Insert the design spectrum given by EBCS 8, 1995
Calculate the modal accelerations
Calculate the equivalent static loads
Combine these loads to get equivalent nodal loads.
Solve the stiffness equation.
Extract the member end forces.
Calculate the support reactions.


1
Static analysis
2
Calculate the fixed end actions from applied member loads.
Assemble applied joint loads and equivalent joint loads.
Solve the stiffness equation.
Extract the member end forces.
Calculate the support reactions..


2
Combined analysis
stop
Make dynamic analysis in the above steps (1)
Make static analysis in the above steps (2)
Linearly superpose the two analysis results.
Display the analysis results

3.6. Running the Program

Before running SAF, the user has to prepare the input data required for the analysis in the form defined by Section 3.2 and save it into a file. Then, the usercan run SAF by saying ‘Run with scilab-5.3.0’ or open the program and ‘execute’ the file on scilab window. When he/she runs the program, dialog boxes to communicate with him/her will be displayed. The first dialog box seems as shown in Figure 3.2, which is nothing but used to say ‘welcome’.

Figure 3.2: Dialog Box

The second dialog box seems as shown in Figure 3.3, which is used to select the analysis type that the user want to do.Therefore, he/she has to select the analysis type by writing ‘1’ on the desired analysis type.

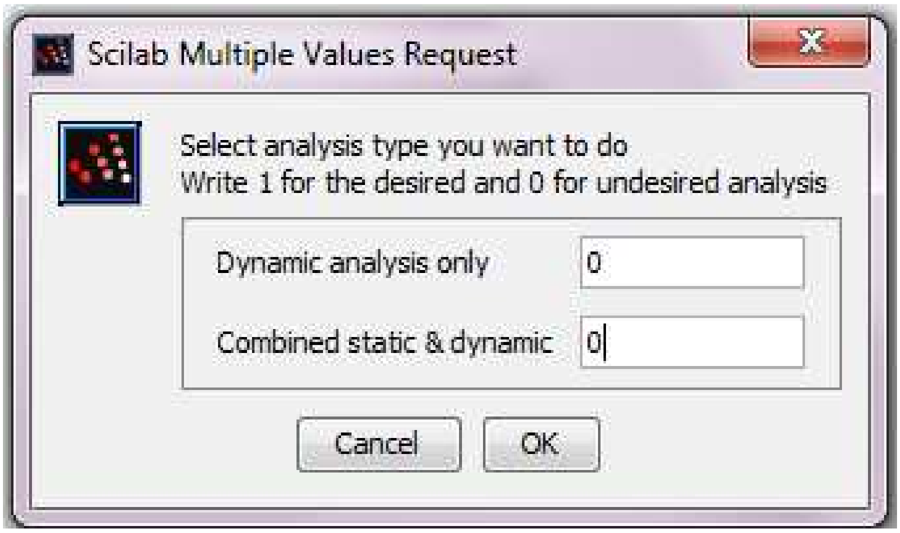


Figure 3.3: Selection of Analysis Type

After choosing the analysis type, another dialog will display to insert the input file (Figure 3.4). The user has to select the file name of the input file.

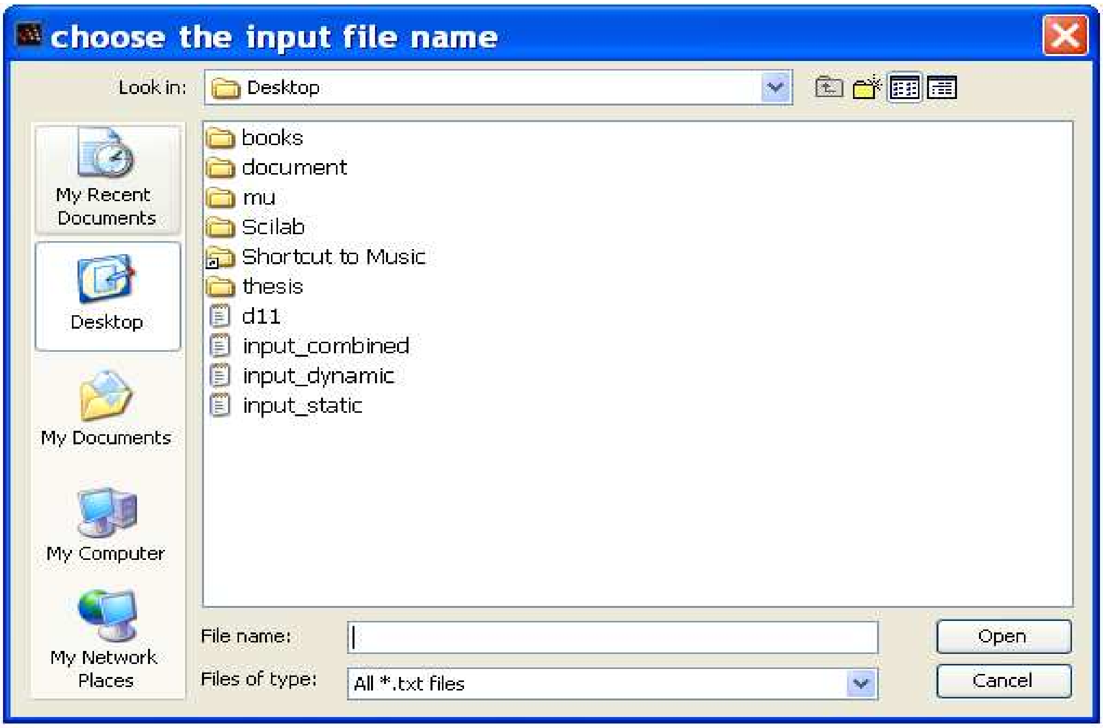


Figure 3.4: Choosing the Input File

Finally, the output from the analysis will display on scilab window. The user can copy this output to any other format and save it to use in another time.

4. Numerical Examples

4.1 Frame to Demonstrate Dynamic Analysis

Figure 4.1 shows a diagonally braced plane frame consisting of prismatic members. The two diagonal members, member 4 and 5, are axial members (resist axial load only). Joint number 1 and 2 are fully restrained. All the members are made from identical material with modulus of elasticity (E) = 3GPa and have cross-section of 300mmx400mm. The lumped masses at Node 3 and 4 are 1000kg and 2000kg respectively. The location of the site is in Zone 1 and the subsoil condition is soil class A. The Importance category of the system is category 1 and the behavior factor of the system is 0.7.

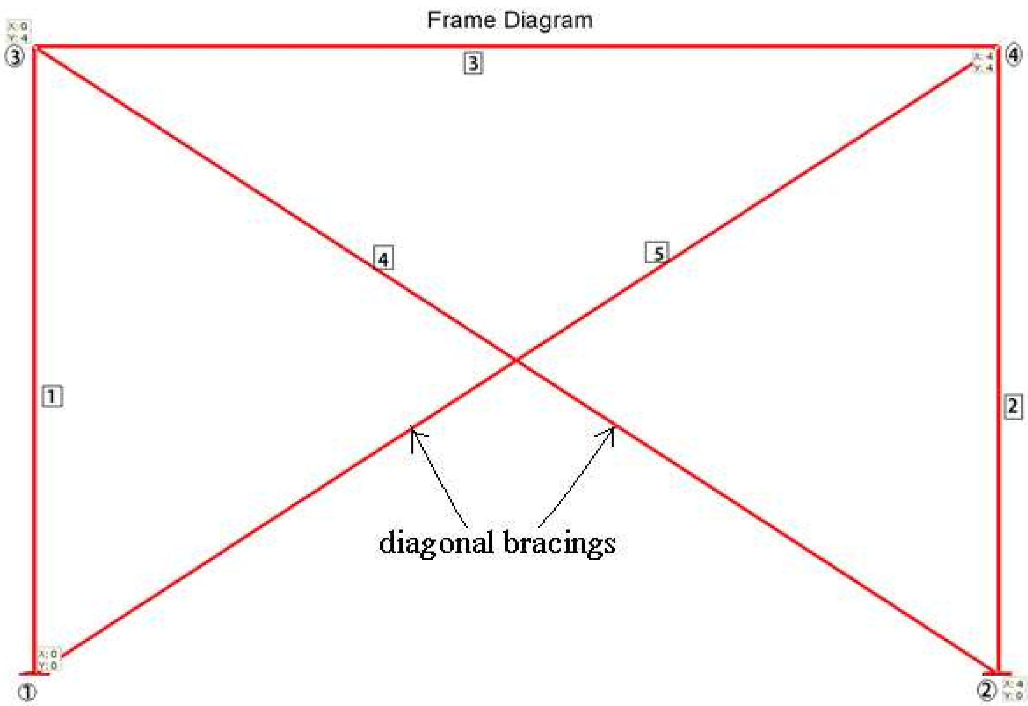


Figure 4.1: Diagonally Braced Plane Frame

Hand Calculations for the Frame:

Cross-sectional properties

Modulus of elasticity E = $30*10^8 N/m^2$

Area of the cross-section (A) = 0.3m*0.4m = $0.12m^2$

Moment of inertia for flexural members (member 1, 2 and 3) I= $0.3m*(0.4m)^3/12 = 1.6*10^{-3}\ m^4$

Step 1: Construction of stiffness matrix.

The frame has six degrees of freedoms (DOFs) at the two unrestrained nodes, node 3 and 4, three at each node as shown and numbered in Figure 4.2.

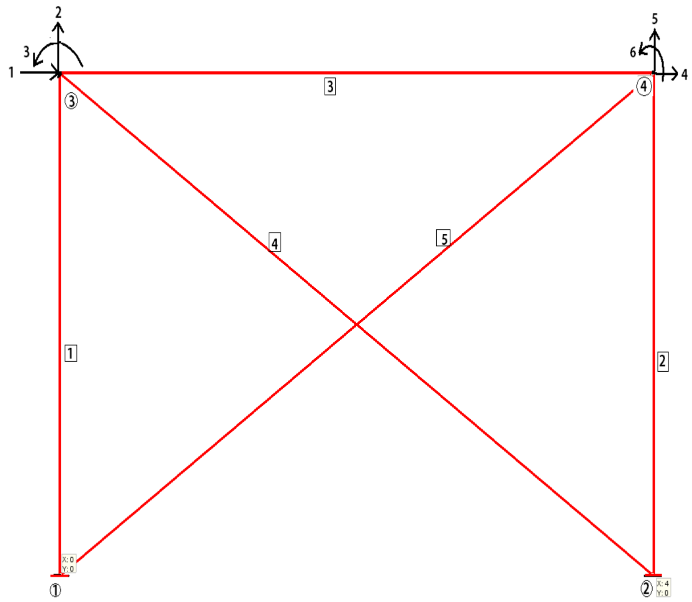


Figure 4.2: Degrees of freedom for the Frame

A unit displacement is applied at each DOF to get the stiffness coefficients. Only stiffness coefficients for DOF number 1 is shown in Figure 4.3. Similar approach is applied for all other stiffness coefficients.

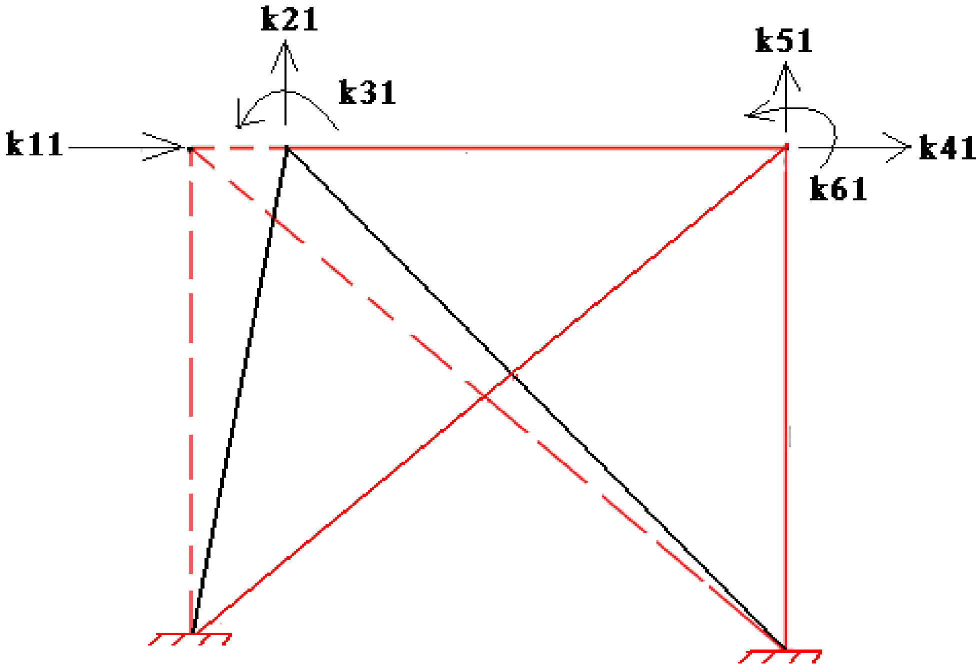


Figure 4.3: Stiffness Coefficients

These coefficients are calculated as follows:

$k11 = 12EI/L_1^3 + EA/L_3 + EA/L_4 * (\cos^2\theta)$
$= 12*30*10^8*0.0016/4^3 + 30*10^8*0.12/4 + 30*10^8*0.12/4\sqrt{2}*(\cos^2 45^0)$ $=\mathbf{1.227*10^8}$

$k21 = 0+0 +-EA/L_4*(\cos\theta*\sin\theta)$
$= -30*10^8*0.12/4\sqrt{2}*(\cos45^0*\sin45^0)$ = $\mathbf{-31819805}$

$k31=-6EI/L_1^2 + 0+0$
$=- 6*30*10^8*0.0016/4^2$ = $\mathbf{-1.8*10^6}$

$k41=-EA/L_3$
$= -30*10^8*0.12/4$ = $\mathbf{-9*10^7}$

k51=0

k61=0

where; E is the modulus of elasticity; I is the moment of inertia ; L1, L3 and L4 are the length of member 1, 3 and 4 respectively; θ is the inclination angle of member 4 about horizontal axes.

Finally,by computing other stiffness coefficients, the stiffness matrix will be:

$$K = \begin{bmatrix} 1.227*10^8 & -31819805 & 1800000 & -90000000 & 0 & 0 \\ -31819805 & 1.227*10^8 & 1800000 & 0 & -900000 & 1800000 \\ 1800000 & 1800000 & 9600000 & 0 & -1800000 & 2400000 \\ -90000000 & 0 & 0 & 1.227*10^8 & 31819805 & 1800000 \\ 0 & -900000 & -1800000 & 31819805 & 1.227*10^8 & -1800000 \\ 0 & 1800000 & 2400000 & 1800000 & -1800000 & 9600000 \end{bmatrix}$$

Step 2: Construction of mass matrix

The mass matrix has a non-zero value only for horizontal DOFs (DOF number 1 and 4).

$$m = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Taking only the horizontal DOFs by avoiding the zero values of the mass matrix:

$$mtt = \begin{bmatrix} 1000 & 0 \\ 0 & 2000 \end{bmatrix}$$

Step3: Condensation of stiffness matrix

i. Rearrange stiffness matrix by bringing horizontal translations to top band of the stiffness matrix.

$$
K_t = \left[\begin{array}{cc|cccc}
1.227*10^8 & -90000000 & -31819805 & 1800000 & 0 & 0 \\
-90000000 & 1.227*10^8 & 0 & 0 & 31819805 & 1800000 \\
\hline
-31819805 & 0 & 1.227*10^8 & 1800000 & -90000000 & 1800000 \\
1800000 & 0 & 1800000 & 9600000 & -1800000 & 2400000 \\
0 & 31819805 & -90000000 & -1800000 & 1.227*10^8 & -1800000 \\
0 & 1800000 & 1800000 & 2400000 & -1800000 & 9600000
\end{array}\right]
$$

ii. Condense the stiffness matrix

$$
K_{tt} = \begin{vmatrix} 113900000 & -90026994 \\ -90026994 & 113900000 \end{vmatrix}
$$

Step 4: Solving the characteristic equation

$\det [k_{tt} - m_{tt}\omega_n^2] = 0$

$$
\begin{vmatrix} 113900000 - 1000\omega n2 & -90026994 \\ -90026994 & 113900000 - 2000\omega n2 \end{vmatrix} = 0
$$

$(1.139*10^8 - 1000\omega_n^2) * (1.139*10^8 - 2000\omega_n^2) - (-90026994) * (-90026994) = 0$

$4.875*10^{15} - 3.418*10^{11}\omega_n^2 + 2000000\omega_n^4 = 0$

$\omega_n^2 = 155188.76$ or 15708.27

The natural frequencies of vibration

Mode 1: $\omega_n = 125.33$ rad/sec

Mode 2: $\omega_n = 393.94$ rad/sec

The natural periods of vibration

Mode 1: $T_n = 2\pi/\omega_1 = 0.05$ sec

Mode 2: $T_n = 2\pi/\omega_2 = 0.02$ sec

The natural mode shapes of vibration

The natural mode shapes of vibration are gained by solving the equation $[k_{tt} - m_{tt}\omega_n^2]\ \phi_n = 0$ for each mode.

Mode 1: ω_n= 125.33 rad/sec

$$\begin{bmatrix} 98223085 - 90026994 \\ - 9002699482514814 \end{bmatrix} * \begin{Bmatrix} \phi 11 \\ \phi 21 \end{Bmatrix} = 0$$

Give ϕ_{11} = 1 and solve for ϕ_{21}

Φ_{21} = 1.0910

Mode 2: ω_n= 393.94 rad/sec

$$\begin{bmatrix} - 41257407 & - 90026994 \\ -90026994 & - 196400000 \end{bmatrix} * \begin{Bmatrix} \phi 12 \\ \phi 22 \end{Bmatrix} = 0$$

Give ϕ_{12} = 1 and solve for ϕ_{22}

Φ_{22} = - 0.46

$$\Phi n = \begin{Bmatrix} \phi 11 & \phi 12 \\ \phi 21 & \phi 22 \end{Bmatrix} = \begin{Bmatrix} 1.00 & 1.00 \\ 1.09 & - 0.46 \end{Bmatrix}$$

Modal mass property:

$$Mn(1)=\phi n1^T*mtt*\ \phi n1=\{1.00 \quad 1.09\} * \begin{bmatrix} 1000 & 0 \\ 0 & 2000 \end{bmatrix} * \begin{Bmatrix} 1.00 \\ 1.09 \end{Bmatrix} =$$

$$3380.56 Mn(2)=\phi n2^T*mtt*\ \phi n2 = \{1.00 \quad - 0.46\} * \begin{bmatrix} 1000 & 0 \\ 0 & 2000 \end{bmatrix} * \begin{Bmatrix} 1 \\ -0.46 \end{Bmatrix} = 1419.89$$

Modal mass participation:

$$Ln(1)= \phi n1^T*mtt* \begin{Bmatrix} 1.00 \\ 1.00 \end{Bmatrix} = \{1.00 \quad 1.09\} * \begin{bmatrix} 1000 & 0 \\ 0 & 2000 \end{bmatrix} * \begin{Bmatrix} 1.00 \\ 1.00 \end{Bmatrix} = 3182.00$$

$$Ln(2)= \phi n2^T*mtt* \begin{Bmatrix} 1.00 \\ 1.00 \end{Bmatrix} = \{1.00 \quad - 0.46\} * \begin{bmatrix} 1000 & 0 \\ 0 & 2000 \end{bmatrix} * \begin{Bmatrix} 1.00 \\ 1.00 \end{Bmatrix} = 83.60$$

Modal mass participation factor:

$\Gamma_n(1)=Ln(1)/Mn(1)$ = 3182.00/3380.56 = 0.94

$\Gamma_n(2)=Ln(2)/Mn(2)$ = 83.60/1419.89 = 0.06

Note: check summation of modal mass participation factors is equal to one.

0.06+0.94 = 1 Ok!

Modal accelerations

Mode 1: $An=\alpha*\beta_o*\gamma*9.81$, where $\alpha=\alpha_o*I$; β_o is the ordinate of the design spectrum; γ is the behavior factor.

α_o= 0.03 for zone 1, I = 1.4 for importance category 1; α = 0.03*1.4 =0.04

From design spectrum given by EBCS 8, 1995, β_o = 1.75 for T_n = 0.05 and soil class A.

$\gamma = 0.07$

So, $A_{n1} = 0.042*1.75*0.7*9.81 = 0.50$ m/sec^2

Mode 2: $\beta_o = 1.24$ for $T_n = 0.02$ and soil class A.

So, $A_{n2} = 0.04*1.24*0.7*9.81 = 0.36$m/sec^2

Equivalent nodal loads

$fn(i,j)=\Gamma_n(j)*mtt(i,i)*\phi(i,j)*An(j)$

$fn(1,1) = \Gamma_n(1)*mtt(1,1)*\phi(1,1)*An(1) = 0.94*1000*1.00*0.50 = 475.02$ N

$fn(1,2) = \Gamma_n(2)*mtt(1,1)*\phi(1,2)*An(2) = 0.06*1000*1.00*0.36 = 21.03$ N

$fn(2,1) = \Gamma_n(1)*mtt(2,2)*\phi(2,1)*An(1) = 0.94*2000*1.09*0.50 = 1036.50$ N

$fn(2,2) = \Gamma_n(2)*mtt(2,2)*\phi(2,2)*An(2) = 0.06*2000*-0.46*0.36 = -19.27$ N

Combination rule

EBCS 8, 1995 recommends that if the natural periods for two modes are independent of each other (i.e. $T(i) \leq 0.9T(j)$), use SRSS combination rule. Other wise use CQC method.

$T(1) = 0.318T(2)$ $\rightarrow$ use SRSS rule to combine the two modes.

$$fn(i) = \sqrt{\sum_{j=1}^{n} fn(i,j)^2}$$

$$fn(1)=\sqrt{fn(1,1)^2 + fn(1,2)^2}$$

$$= \sqrt{21.03 * 21.03 + 475.02 * 475.02} = 475.49 \text{ N}$$

$$fn(2)=\sqrt{fn(2,1)^2 + fn(2,2)^2}$$

$$= \sqrt{19.27 * 19.27 + 1036.50 * 1036.50} = 1036.68 \text{ N}$$

Load vector:

The load vector have a non zero value at DOF numbers 1 and 4 (i.e. fn(1) = 475.49 N and fn(2) = 1036.68 N respectively).

$$p = \begin{Bmatrix} 475.49 \\ 0.00 \\ 0.00 \\ 1036.68 \\ 0.00 \\ 0.00 \end{Bmatrix} \text{N}$$

Nodal displacements:

The nodal displacements can be gained by solving Equation (3.13):K*U = K^{-1}P; where, K is the stiffness matrix; U is the nodal displacements in global axes; and P is the load vector.

U = K^{-1}P

$$u = \begin{bmatrix} 2.34*10^{-8} & 6.16*10^{-9} & -5.41*10^{-9} & 1.85*10^{-8} & -4.88*10^{-9} & -4.18*10^{-9} \\ 6.16*10^{-9} & 9.81*10^{-9} & -2.65*10^{-9} & 4.88*10^{-9} & -1.27*10^{-9} & --2.33*10^{-9} \\ -5.41*10^{-9} & -2.65*10^{-9} & 1.00*10^{-7} & -4.18*10^{-9} & 2.33*10^{-9} & -2.65*10^{-8} \\ 1.85*10^{-8} & 4.88*10^{-9} & -4.18*10^{-9} & 2.34*10^{-8} & -6.16*10^{-9} & -5.41*10^{-9} \\ -4.88*10^{-9} & -1.27*10^{-9} & 2.33*10^{-9} & -6.16*10^{-9} & 9.81*10^{-9} & 2.65*10^{-9} \\ -4.18*10^{-9} & -2.33*10^{-9} & -2.65*10^{-8} & -5.41*10^{-9} & 2.65*10^{-9} & 1.00*10^{-7} \end{bmatrix} * \begin{bmatrix} 475.49 \\ 0.00 \\ 0.00 \\ 1036.68 \\ 0.00 \\ 0.00 \end{bmatrix}$$

$$U = \begin{Bmatrix} 3.03 * 10^{-5} m \\ 0.80 * 10^{-5} \text{ m} \\ -0.69 * 10^{-5} \text{ rad} \\ 3.30 * 10^{-5} \text{ m} \\ -0.87 * 10^{-5} \text{ m} \\ -0.76 * 10^{-5} \text{rad} \end{Bmatrix}$$

Member end forces:

Only member end forces for member number 3 will be calculated here to check the reliability of the written program.

Since the local axes for member number 3 coincides with the global axes, rotation of global displacements at the member ends is not desired.

There is no directly applied member load on the member. Therefore, the member end forces can be gained by computing Equation (3.14):

f=k*u-$\mathbf{P_{eq}}$; where, **f** is the member end force;**k** is the local stiffness matrix of the member,**u** is the local displacement vector for the member ends; and $\mathbf{P_{eq}}$ is the equivalent loads on member ends, which is zero for dynamic loads only.

$$k = \begin{bmatrix} 90000000 & 0 & 0 & -90000000 & 0 & 0 \\ 0 & 900000 & 1800000 & 0 & -900000 & 1800000 \\ 0 & 1800000 & 4800000 & 0 & -1800000 & 2400000 \\ -90000000 & 0 & 0 & 90000000 & 0 & 0 \\ 0 & -900000 & -1800000 & 0 & 900000 & -1800000 \\ 0 & 1800000 & 2400000 & 0 & -1800000 & 4800000 \end{bmatrix}$$

$$u = \begin{Bmatrix} 3.03 * 10^{-5} m \\ 0.80 * 10^{-5}\ \text{m} \\ -0.69 * 10^{-5}\ \text{rad} \\ 3.30 * 10^{-5}\ \text{m} \\ -0.87 * 10^{-5}\ \text{m} \\ -0.76 * 10^{-5}\text{rad} \end{Bmatrix}$$

$$f = k * u = \begin{Bmatrix} -247.64\ N \\ -11.07\ \text{N} \\ -21.31\ \text{Nm} \\ 247.64\ \text{N} \\ 11.07\ \text{N} \\ -22.96\ \text{Nm} \end{Bmatrix}$$

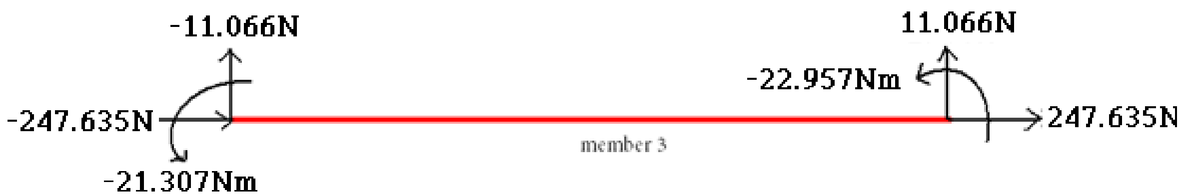


Figure 4.4: Member End Forces

Preparation of Input data for the Program

The input data required for dynamic analysis of a plane frame are coordinates matrix ‘xz’, member property matrix ‘mprop’, connectivity matrix ‘conn’ ,boundary constraint matrix ‘bc’, lumped mass at the nodes ‘mass’,location of the site ‘zone’, subsoil condition of the site ‘sclass’, importance category of the structure ‘importance’ and behavior factor of the system ‘behavior’.

Therefore, for this frame the data is organized as follows:

//Insert Coordinates matrix.xz=[0,0;4,0;0,4;4,4];

//Insert Member Property matrix.mprop=[30*10^8,0.12,0.0016;30*10^8,0.12,0];

//Insert Connectivity matrix.conn=[1,3,1;2,4,1;3,4,1;2,3,2;1,4,2];

//Insert Boundary Constraint matrix.bc=[1,1,1,1;2,1,1,1];

//Insert mass of the system lumped at nodes.mass=[3,1000;4,2000];

//Insert location of the site according to the zoning provided by EBCS 8.zone=1;

//Insert the subsoil condition of the site according to EBCS 8.sclass=1;

//Insert importance category of the building according to the classification given by EBCS 8.importance=1;

//Insert the behavior factor of the structure.behavior=0.7;

Analysis Outputs from SAF:

Number of degrees of freedom = 6

Number of horizontal translational degrees of freedom = 2

The natural frequencies of vibration

Mode ωn

1. 125.33
2. 393.94

The natural periods of vibration

Mode Tn

1. 0.05
2. 0.02

The natural mode shapes of vibration

1.001.00

1.09-0.46

The modal mass participation factor

Mode	Γ_n
1.	0.94
2.	0.06

The modal accelerations

Mode	An
1.	0.50
2.	0.36

The nodal displacement from dynamic analysis in global axes

Node	Ux	Uz	ry
1.	$0.00*10^0$	$0.00*10^0$	$0.00*10^0$
2.	$0.00*10^0$	$0.00*10^0$	$0.00*10^0$
3.	$3.03*10^{-5}$	$0.80*10^{-5}$	$-0.69*10^{-5}$
4.	$3.30*10^{-5}$	$-0.87*10^{-5}$	$-0.76*10^{-5}$

The member end forces of the system in local axes

Member	P1	V1	M1	P2	V2	M2
1.	-720.29	14.81	37.93	720.29	-14.81	21.33
2.	785.05	16.06	41.23	-785.05	-16.06	22.99
3.	-247.96	-11.08	-21.33	247.96	11.08	-22.99
4.	1002.97	0.00	0.00	-1002.97	0.00	0.00
5.	-1094.56	0.00	0.00	1094.56	0.00	0.00

Support reactions in global axes

Node	Rx	Rz	My
1.	-788.79	-1494.26	37.93
2.	-725.26	1494.26	41.23

Comparison of the Results:

Mode		ω_n (rad/sec)	T_n (sec)
1	Hand calculation	0.05	125.33
	SAF	0.05	125.33
2	Hand calculation	0.02	393.94
	SAF	0.02	393.94

Table 4: Comparison of natural frequencies and periods of vibration

Mode		Γ_n	A_n (m/sec^2)
1	Hand calculation	0.50	0.94
	SAF	0.51	0.94
2	Hand calculation	0.36	0.06
	SAF	0.36	0.06

Table 5: Comparison of modal mass participation factor and modal acceleration

Node		Ux (10^{-2}mm)	Uz (10^{-2}mm)	ry(10^{-6} rad)
3	Hand calculation	3.03	0.80	-6.90
	SAF	3.03	0.80	-6.90
4	Hand calculation	3.30	-0.87	-7.60
	SAF	3.30	-0.87	-7.60

Table 6: Comparison of nodal displacements

Member		P1 (N)	V1 (N)	M1(Nm)	P2 (N)	V2 (N)	M2(Nm)
3	Hand calculation	-247.64	-11.07	-21.31	247.64	11.07	-22.96
	SAF	-247.96	-11.08	-21.33	247.96	11.08	-22.99

Table 4: Comparison of member end forces

The difference between the above values comes from the rounding of values when the frame is analyzed manually.In general, the analysis results of the developed program are similar with the results gained by hand calculation. Therefore, the program can accurately analyze frames for dynamic loading.

4.2. A G+6 Building Frame Analysis

The frame given in Figure 4.5 is analyzed with the developed program, SAF,and SAP 2000 Version-14 to compare the results of the program with results of SAP 2000 Version-14.The frame is analyzed for both static and dynamic loading. The static loads and support conditions are as shown in Figure 4.5 and the lumped masses at the nodes aregiven in Figure 4.6. The assignment of node number and member number is as shown in Figure 4.7.

Cross section property of:

- beams is 350 mm X 350 mm
- exterior columns is 400 mm X 400 mm
- interior columns is 400 mm X 300 mm

Modulus of elasticity of:

- beams is 30GPa
- columns is 29GPa

The location of the site is in Zone 2 and the subsoil condition is soil class A. The Importance category of the system is category 1 and the Behavior factor of the system is 0.7.

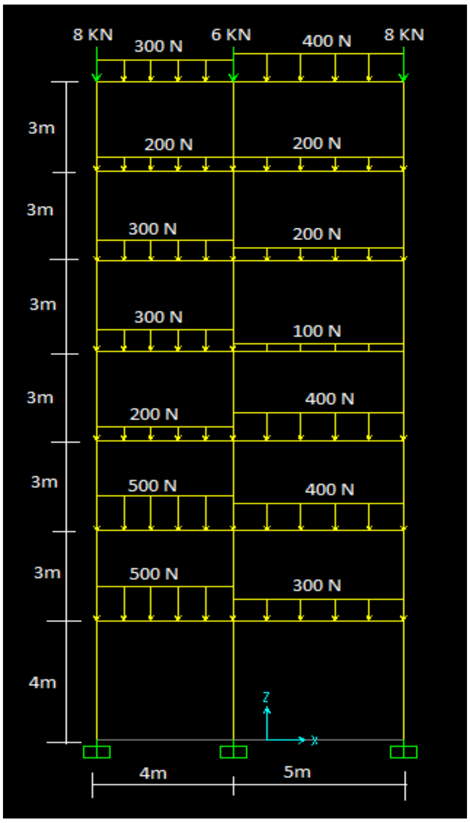


Figure 4.5: AG+6 Building Frame

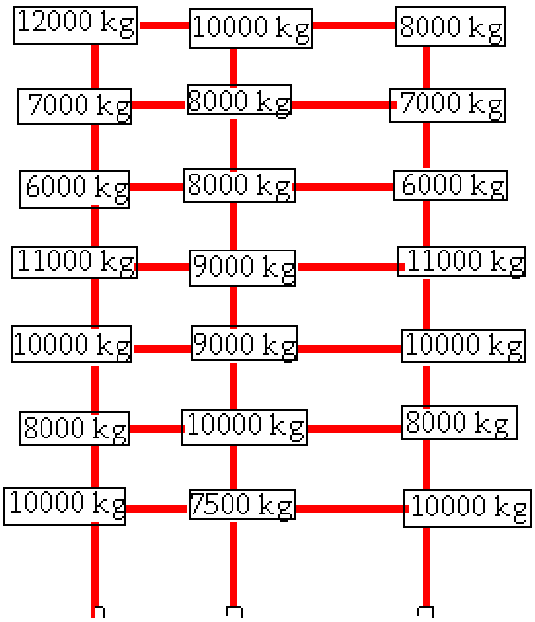


Figure 4.6: The Lumped Mass at the Nodes

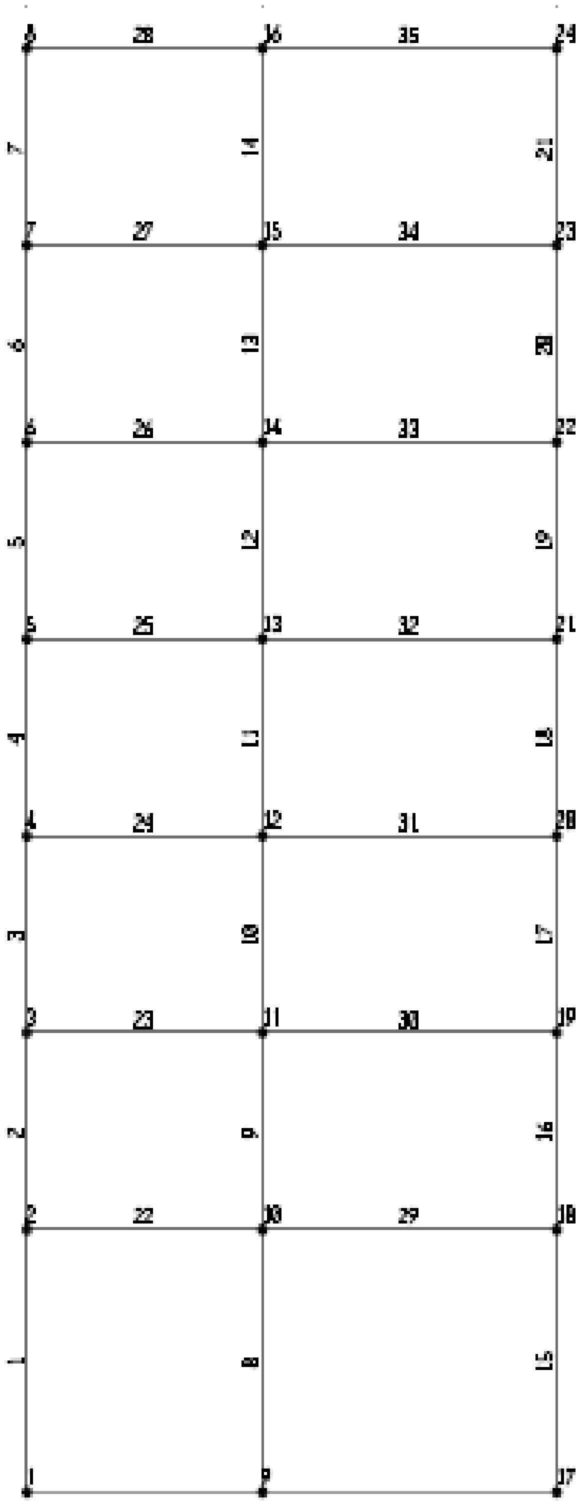


Figure 4.7: Assignment of Numbers for Nodes and Members

Preparation of input data for SAF

//coordinates matrix

xz= [0,0;0,4;0,7;0,10;0,13;0,16;0,19;0,22;4,0;4,4;4,7;4,10;4,13;4,16;4,19;4,22;9,0;9,4;9,7; 9,10; 9,13; 9,16; 9,19; 9,22];

//member property matrix

mprop=[29*10^9,0.3*0.4,(0.3*0.4^3)/12; 29*10^9,0.4*0.4,(0.4*0.4^3)/12; 30*10^9,0.35*0.35,(0.35*0.35^3)/12];

//connectivity matrix

conn = [1,2,2; 2,3,2; 3,4,2; 4,5,2; 5,6,2; 6,7,2; 7,8,2; 9,10,1; 10,11,1; 11,12,1; 12,13,1; 13,14,1; 14,15,1; 15,16,1; 17,18,2; 18,19,2; 19,20,2; 20,21,2; 21,22,2; 22,23,2; 23,24,2; 2,10,3; 3,11,3; 4,12,3; 5,13,3; 6,14,3; 7,15,3; 8,16,3; 10,18,3; 11,19,3; 12,20,3; 13,21,3; 14,22,3; 15,23,3; 16,24,3];

//boundary constraint matrix

bc=[1,1,1,1; 9,1,1,1; 17,1,1,1];

//joint loads matrix

jtloads = [8,0,-8000,0; 16,0,-6000,0; 24,0,-8000,0];

//member loads matrix

loads = [22,-500,-500,1; 29,-300,-300,1; 23,-500,-500,1; 30,-400,-400,1; 24,-200,-200,1; 31,-300,-400,1; 25,-300,-300,1; 32,-100,-100,1; 26,-300,-300,1; 33,-200,-200,1; 27,-200,-200,1; 34,-200,-200,1; 28,-300,-300,1; 35,-400,-400,1];

//lumped mass

mass = [2,10000; 10,7500; 18,10000; 3,8000; 11,10000; 19,8000; 4,10000; 12,9000; 20,10000;5,11000; 13,9000; 21,11000; 6,6000; 14,8000; 22,6000; 7,7000; 15,8000; 23,7000; 8,12000; 16,10000; 24,8000];

zone=2;sclass=1;importance=1;behavior=0.7;

Analysis Results from SAF

Number of degrees of freedom = 63

Number of horizontal translational degrees of freedom = 21

The natural frequencies of vibration

Mode	ωn
1.	5.21
2.	15.79
3.	29.49
4.	46.20
5.	63.82
6.	87.05
7.	102.70
8.	276.41
9.	289.45
10.	292.01
11.	292.46
12.	326.49
13.	345.98
14.	376.80
15.	499.83
16.	512.63
17.	519.91
18.	524.91
19.	554.76
20.	574.64
21.	593.53

The natural periods of vibration

Mode	Tn
1.	1.20
2.	0.40
3.	0.21
4.	0.14
5.	0.10
6.	0.07
7.	0.06
8.	0.02
9.	0.02
10.	0.02
11.	0.02
12.	0.02
13.	0.02
14.	0.02
15.	0.01
16.	0.01
17.	0.01
18.	0.01
19.	0.01
20.	0.01
21.	0.01

The modal mass participation factor

Mode Γn

1. $26.68*10^{-2}$
2. $23.08*10^{-2}$
3. $23.28*10^{-2}$
4. $13.83*10^{-2}$
5. $8.36*10^{-2}$
6. $4.36*10^{-2}$
7. $0.41*10^{-2}$
8. $9.08*10^{-9}$
9. $6.13*10^{-5}$
10. $-1.02*10^{-5}$
11. $-0.02*10^{-5}$
12. $-4.52*10^{-5}$
13. $-7.03 *10^{-10}$
14. $-2.77*10^{-8}$
15. $-1.46*10^{-11}$
16. $- 8.99*10^{-9}$
17. $-0.23*10^{-5}$
18. $-3.14*10^{-5}$
19. $4.44*10^{-5}$
20. $3.23*10^{-10}$
21. $1.78*10^{-8}$

The modal accelerations

Mode An

1. $39.90*10^{-2}$
2. $120.17*10^{-2}$
3. $120.17*10^{-2}$
4. $120.17*10^{-2}$
5. $119.06*10^{-2}$
6. $100.11*10^{-2}$
7. $92.18*10^{-2}$
8. $64.46*10^{-2}$
9. $63.72*10^{-2}$
10. $63.58*10^{-2}$
11. $63.56*10^{-2}$
12. $61.94*10^{-2}$
13. $61.16*10^{-2}$
14. $60.09*10^{-2}$
15. $57.13*10^{-2}$
16. $56.91*10^{-2}$
17. $56.78*10^{-2}$
18. $56.70*10^{-2}$
19. $56.23*10^{-2}$
20. $55.95*10^{-2}$
21. $55.70*10^{-2}$

The nodal displacements from dynamic analysis in global axes

Node	Ux	Uz	ry
1.	$0.00*10^{-2}$	$0.00*10^{-2}$	$0.00*10^{-2}$
2.	$0.65*10^{-2}$	$0.01*10^{-2}$	$-0.17*10^{-2}$
3.	$1.22*10^{-2}$	$0.21*10^{-3}$	$-0.15*10^{-2}$
4.	$1.71*10^{-2}$	$2.65*10^{-4}$	$-0.13*10^{-2}$
5.	$2.12*10^{-2}$	$3.00*10^{-4}$	$-0.10*10^{-2}$
6.	$2.43*10^{-2}$	$3.21*10^{-4}$	$-0.08*10^{-2}$
7.	$2.67*10^{-2}$	$3.32*10^{-4}$	$-0.06*10^{-2}$
8.	$2.83*10^{-2}$	$3.35*10^{-4}$	$-0.04*10^{-2}$
9.	$0.00*10^{-2}$	$0.00*10^{-2}$	$0.00*10^{-2}$
10.	$0.65*10^{-2}$	$-0.52*10^{-4}$	$-0.11*10^{-2}$
11.	$1.22*10^{-2}$	$-0.81*10^{-4}$	$-0.11*10^{-2}$
12.	$1.71*10^{-2}$	$-0.99*10^{-4}$	$-0.09*10^{-2}$
13.	$2.12*10^{-2}$	$-1.11*10^{-4}$	$-0.08*10^{-2}$
14.	$2.43*10^{-2}$	$-1.17*10^{-4}$	$-0.06*10^{-2}$
15.	$2.67*10^{-2}$	$-1.19*10^{-4}$	$-0.04*10^{-2}$
16.	$2.83*10^{-2}$	$-1.20*10^{-4}$	$-0.02*10^{-2}$
17.	$0.00*10^{-2}$	$0.00*10^{-2}$	$0.00*10^{-2}$
18.	$0.64*10^{-2}$	$-0.96*10^{-4}$	$-0.18*10^{-2}$
19.	$1.22*10^{-2}$	$-1.50*10^{-4}$	$-0.15*10^{-2}$
20.	$1.71*10^{-2}$	$-1.90*10^{-4}$	$-0.13*10^{-2}$
21.	$2.12*10^{-2}$	$-2.17*10^{-4}$	$-0.10*10^{-2}$
22.	$2.43*10^{-2}$	$-2.33*10^{-4}$	$-0.08*10^{-2}$
23.	$2.67*10^{-2}$	$-2.42*10^{-4}$	$-0.06*10^{-2}$
24.	$2.83*10^{-2}$	$-2.45*10^{-4}$	$-0.04*10^{-2}$

Support reactions from dynamic loads in global axes

Node	Rx	Rz	My
1.	-36195.71	-156293.27	98392.02
9.	-36776.23	45510.98	86667.76
17.	-34384.53	110782.28	95936.79

The total nodal displacement from combined analysis in global axes

Node	Ux	Uz	ry
1	0.00E+00	0.00E+00	0.00E+00
2	6.48E-03	1.24E-04	-1.68E-03
3	1.22E-02	1.92E-04	-1.49E-03
4	1.71E-02	2.39E-04	-1.26E-03
5	2.12E-02	2.68E-04	-1.01E-03
6	2.43E-02	2.82E-04	-7.78E-04
7	2.67E-02	2.87E-04	-5.64E-04
8	2.83E-02	2.85E-04	-3.75E-04
9	0.00E+00	0.00E+00	0.00E+00
10	6.49E-03	-6.97E-05	-1.13E-03
11	1.22E-02	-1.10E-04	-1.15E-03
12	1.71E-02	-1.39E-04	-9.56E-04
13	2.12E-02	-1.59E-04	-7.66E-04
14	2.43E-02	-1.73E-04	-5.94E-04
15	2.67E-02	-1.82E-04	-4.45E-04
16	2.83E-02	-1.89E-04	-2.42E-04
17	0.00E+00	0.00E+00	0.00E+00
18	6.48E-03	-1.07E-04	-1.75E-03
19	1.22E-02	-1.70E-04	-1.52E-03

20	1.71E-02	-2.17E-04	-1.29E-03
21	2.12E-02	-2.50E-04	-1.02E-03
22	2.43E-02	-2.73E-04	-7.83E-04
23	2.83E-02	-2.88E-04	-5.66E-04
24	2.67E-02	-2.97E-04	-3.43E-04

Support reactions for combined loading in global axes

Node	Rx	Rz	My
1.	-36113.32	-143422.66	98283.75
9.	-36782.31	60625.68	86678.19
17.	-34460.84	123996.97	96043.59

Analysis Outputfrom SAP 2000

Modal Periods And Frequencies				
OutputCase	**StepType**	**StepNum**	**Period**	**CircFreq**
Text	Text	Unitless	Sec	rad/sec
MODAL	Mode	1	1.22	5.15
MODAL	Mode	2	0.40	15.58
MODAL	Mode	3	0.22	29.08
MODAL	Mode	4	0.14	45.49
MODAL	Mode	5	0.10	62.68
MODAL	Mode	6	0.07	85.16
MODAL	Mode	7	0.06	100.18
MODAL	Mode	8	0.02	276.28
MODAL	Mode	9	0.02	289.32
MODAL	Mode	10	0.02	291.72
MODAL	Mode	11	0.02	292.42
MODAL	Mode	12	0.02	326.10
MODAL	Mode	13	0.02	345.82
MODAL	Mode	14	0.02	376.37
MODAL	Mode	15	0.01	499.81
MODAL	Mode	16	0.01	512.56
MODAL	Mode	17	0.01	519.84
MODAL	Mode	18	0.01	524.68
MODAL	Mode	19	0.01	554.65
MODAL	Mode	20	0.01	574.56

Response Spectrum Modal Information						
OutputCase	ModalCase	StepType	StepNum	Period	DampRatio	U1Acc
Text	Text	Text	Unitless	Sec	Unitless	m/sec2
EARTHQUAKE	MODAL	Mode	1	1.22	0.05	0.64
EARTHQUAKE	MODAL	Mode	2	0.40	0.05	1.92
EARTHQUAKE	MODAL	Mode	3	0.22	0.05	1.93
EARTHQUAKE	MODAL	Mode	4	0.14	0.05	1.93
EARTHQUAKE	MODAL	Mode	5	0.10	0.05	1.93
EARTHQUAKE	MODAL	Mode	6	0.07	0.05	1.63
EARTHQUAKE	MODAL	Mode	7	0.06	0.05	1.50
EARTHQUAKE	MODAL	Mode	8	0.02	0.05	1.04
EARTHQUAKE	MODAL	Mode	9	0.02	0.05	1.02
EARTHQUAKE	MODAL	Mode	10	0.02	0.05	1.02
EARTHQUAKE	MODAL	Mode	11	0.02	0.05	1.02
EARTHQUAKE	MODAL	Mode	12	0.02	0.05	1.00
EARTHQUAKE	MODAL	Mode	13	0.02	0.05	0.98
EARTHQUAKE	MODAL	Mode	14	0.02	0.05	0.97
EARTHQUAKE	MODAL	Mode	15	0.01	0.05	0.92
EARTHQUAKE	MODAL	Mode	16	0.01	0.05	0.91
EARTHQUAKE	MODAL	Mode	17	0.01	0.05	0.91
EARTHQUAKE	MODAL	Mode	18	0.01	0.05	0.91
EARTHQUAKE	MODAL	Mode	19	0.01	0.05	0.90
EARTHQUAKE	MODAL	Mode	20	0.01	0.05	0.90
EARTHQUAKE	MODAL	Mode	21	0.01	0.05	0.89

Joint Displacements (Partially)						
Joint	OutputCase	CaseType	StepType	U1	U3	R2
Text	Text	Text	Text	m	m	Radians
1	EARTHQUAKE	LinRespSpec	Max	0.00E+00	0.00E+00	0.00E+00
1	Static	LinStatic		0.00E+00	0.00E+00	0.00E+00
1	COMB1	Combination	Max	0.00E+00	0.00E+00	0.00E+00
1	COMB1	Combination	Min	0.00E+00	0.00E+00	0.00E+00
2	EARTHQUAKE	LinRespSpec	Max	6.69E-03	1.47E-04	1.74E-03
2	Static	LinStatic		2.00E-07	-1.10E-05	3.77E-06
2	COMB1	Combination	Max	6.69E-03	1.36E-04	1.75E-03
2	COMB1	Combination	Min	-6.69E-03	-1.58E-04	-1.74E-03
3	EARTHQUAKE	LinRespSpec	Max	1.27E-02	2.35E-04	1.61E-03
3	Static	LinStatic		9.66E-07	-1.90E-05	3.30E-06
3	COMB1	Combination	Max	1.27E-02	2.16E-04	1.61E-03
3	COMB1	Combination	Min	-1.27E-02	-2.53E-04	-1.61E-03

4	EARTHQUAKE	LinRespSpec	Max	1.81E-02	3.00E-04	1.46E-03
4	Static	LinStatic		8.14E-07	-2.60E-05	8.25E-07
4	COMB1	Combination	Max	1.81E-02	2.74E-04	1.46E-03
4	COMB1	Combination	Min	-1.81E-02	-3.26E-04	-1.46E-03
5	EARTHQUAKE	LinRespSpec	Max	2.27E-02	3.46E-04	1.26E-03
5	Static	LinStatic		1.82E-06	-3.30E-05	3.16E-06
5	COMB1	Combination	Max	2.27E-02	3.13E-04	1.27E-03
5	COMB1	Combination	Min	-2.27E-02	-3.78E-04	-1.26E-03
6	EARTHQUAKE	LinRespSpec	Max	2.64E-02	3.74E-04	1.07E-03
6	Static	LinStatic		4.37E-06	-3.90E-05	3.06E-06
6	COMB1	Combination	Max	2.64E-02	3.35E-04	1.07E-03
6	COMB1	Combination	Min	-2.64E-02	-4.13E-04	-1.07E-03
7	EARTHQUAKE	LinRespSpec	Max	2.92E-02	3.89E-04	8.21E-04
7	Static	LinStatic		6.17E-06	-4.50E-05	1.55E-06
7	COMB1	Combination	Max	2.92E-02	3.44E-04	8.22E-04
7	COMB1	Combination	Min	-2.92E-02	-4.33E-04	-8.19E-04
8	EARTHQUAKE	LinRespSpec	Max	3.12E-02	3.94E-04	5.39E-04
8	Static	LinStatic		6.77E-06	-5.00E-05	4.65E-06
8	COMB1	Combination	Max	3.12E-02	3.43E-04	5.44E-04
8	COMB1	Combination	Min	-3.12E-02	-4.44E-04	-5.34E-04
9	EARTHQUAKE	LinRespSpec	Max	0.00E+00	0.00E+00	0.00E+00
9	Static	LinStatic		0.00E+00	0.00E+00	0.00E+00
9	COMB1	Combination	Max	0.00E+00	0.00E+00	0.00E+00
9	COMB1	Combination	Min	0.00E+00	0.00E+00	0.00E+00

Joint Reactions						
Joint	**OutputCase**	**CaseType**	**StepType**	**F1**	**F3**	**M2**
Text	Text	Text	Text	N	N	N-m
1	EARTHQUAKE	LinRespSpec	Max	36194.30	170656.46	99187.32
1	Static	LinStatic		82.55	12858.43	106.80
1	COMB1	Combination	Max	36276.85	183514.89	99294.13
1	COMB1	Combination	Min	-36111.75	-157798.03	-99080.52
9	EARTHQUAKE	LinRespSpec	Max	36855.64	48683.02	87191.39
9	Static	LinStatic		-6.09	15136.58	-10.24
9	COMB1	Combination	Max	36849.55	63819.60	87181.15
9	COMB1	Combination	Min	-36861.72	-33546.45	-87201.64
17	EARTHQUAKE	LinRespSpec	Max	34427.25	121991.57	96824.76
17	Static	LinStatic		-76.46	13204.99	-105.30
17	COMB1	Combination	Max	34350.78	135196.57	96719.45
17	COMB1	Combination	Min	-34503.71	-108786.58	-96930.06

Comparison of the Results:

Mode	Period		Circular Frequency	
	SAF	SAP 2000	SAF	SAP 2000
1	1.20	1.22	5.21	5.15
2	0.40	0.40	15.29	15.58
3	0.21	0.22	29.49	29.08
4	0.14	0.14	46.20	45.49
5	0.10	0.10	63.82	62.68
6	0.07	0.07	87.05	85.16
7	0.06	0.06	102.70	100.18
8	0.02	0.02	276.41	276.28
9	0.02	0.02	289.45	289.32
10	0.02	0.02	292.01	291.72
11	0.02	0.02	292.46	292.42
12	0.02	0.02	326.49	326.10
13	0.02	0.02	345.98	345.82
14	0.02	0.02	376.80	376.37
15	0.01	0.01	499.83	499.81
16	0.01	0.01	512.63	512.56
17	0.01	0.01	519.91	519.84
18	0.01	0.01	524.91	524.68
19	0.01	0.01	554.76	554.65
20	0.01	0.01	574.64	574.56
21	0.01	0.01	593.53	593.35

Table 8: Comparison of natural frequencies and periods of vibration

Joint	Output Case	SAP 2000			SAF		
		F1	F3	M2	F1	F3	M2
1	EARTHQUAKE	36194.30	170656.46	99187.32	36195.72	156293.27	98392.02
1	COMBINED	36276.85	183514.89	99294.13	36113.32	143422.66	98283.75
9	EARTHQUAKE	36855.64	48683.02	87191.39	36776.23	45510.98	86667.76
9	COMBINED	36849.55	63819.60	87181.15	36782.32	60625.69	86678.19
17	EARTHQUAKE	34427.25	121991.57	96824.76	34384.53	110782.28	95936.79
17	COMBINED	34350.78	135196.57	96719.45	34460.85	123996.97	96043.59

Table 9: Comparison of joint reactions

The analysis results from the developed program are similar to the results from SAP 2000. Therefore, the program can accurately analyze planar frames.

5. Conclusions and Recommendations

5.1. Conclusion

Different commercial computer programs have included features for Response-spectrum analysis but the method they have adopted is not fully disclosed.The computer program developed here can be used for accuracy since it shows what has been done.

In general, the analysis results from the developed program are similar with the results from other sources. Therefore, the program can accurately analyze frames for static and/or seismic loading.

5.2. Recommendations

From the study that has been carried out, the followings are the recommendations drawn from the result:

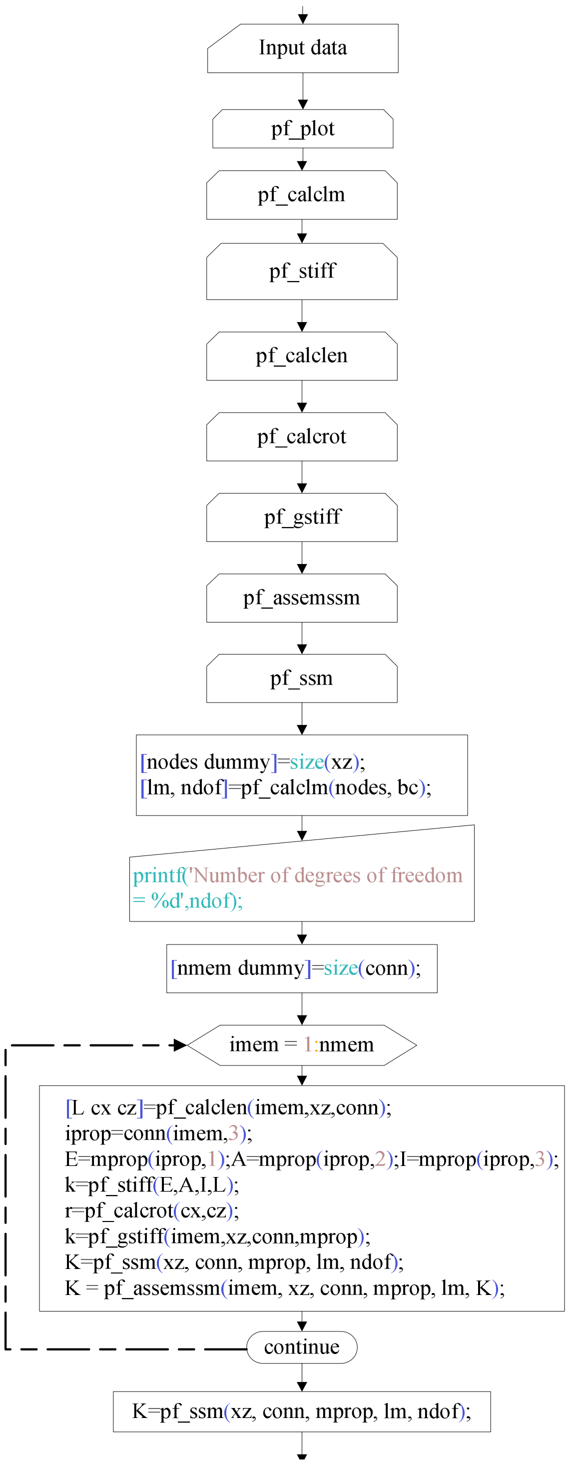
- ✓ As the developed program is easy to apply, consulting firms can make use of the program for analysis of planar frames.
- ✓ Researchers can upgrade this program for three-dimensional frames with some modifications.
- ✓ The measuring unit for different parameters is controlled by the user and the user has to use consistent unit.
- ✓ If one assignment is not properly defined or the input is not properly inserted with a given variable name, Scilab program will stop without computing other assignments written below that assignment.

References

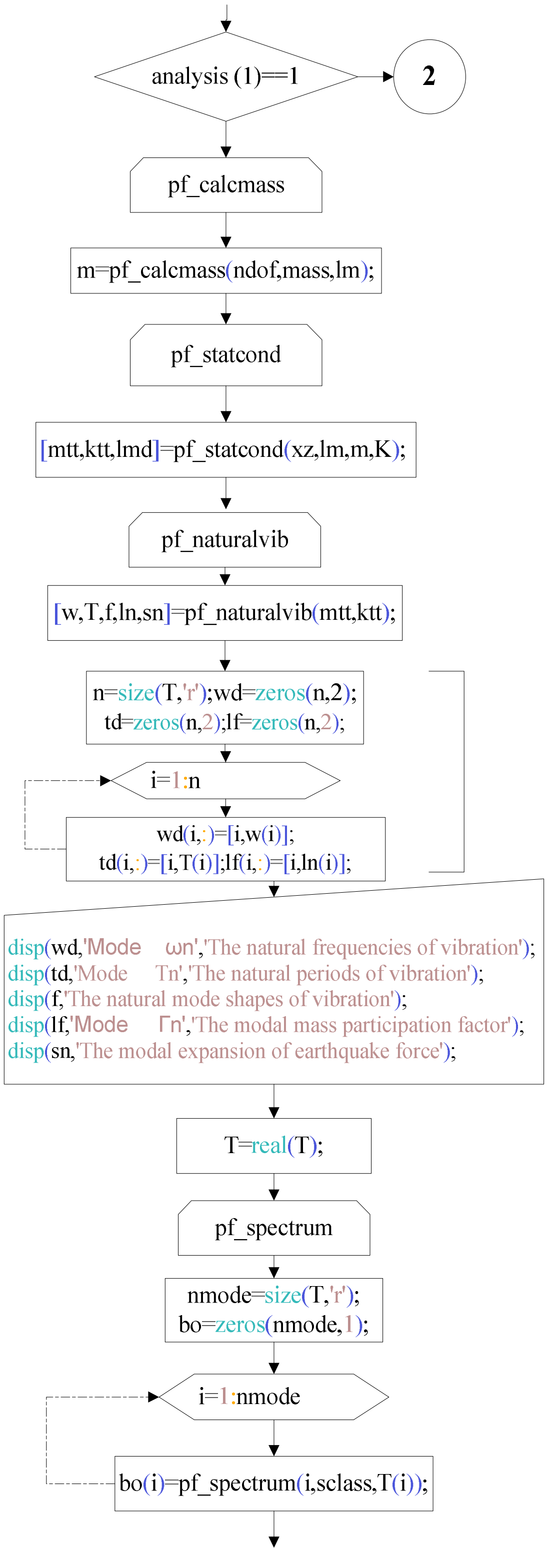
1. Bozorgnia, Yousef andBertero, vitelmo V. (2006). *Earthquake Engineering: Seismology to Performance-Based Engineering.* CRC Press LLC, New York.
2. Chopra, Anil K. (1995). *Dynamics of Structures: Theory and Application to Earthquake Engineering.* Prentice-Hall, Inc., New Jersey.
3. EBCS 8, (1995). *Design of Structures for Earthquake Resistance*. Ministry of Works and Urban Development, Addis Ababa.
4. Elghazouli, Ahmed Y. (2009). *Seismic Design of Buildings to Euro code 8.* Spon Press, London.
5. Elnashai, Amr S. and Sarno, Luigi D. (2008). *Fundamentals of Earthquake Engineering.* John Willey & Sons, Ltd., New York.
6. Gillberto E. Urroz (2001). *Programming with scilab*. www.infoclearinghouse.com, Charlston, S.C.
7. Graeme Chandler and Stephen Roberts (2007). *Australian National UniversityTeaching Modules: Scilab Tutorials.*Australian National University, Australia.
8. Satish Annigeri (2004).*Scilab A Hands on Introduction*, B.V. Bhoomaraddi College of Engineering & Technology, Hubli.
9. Satish Annigeri (2008).*Matrix Structural Analysis of Plane Frames using Scilab.* B.V. Bhoomaraddi College of Engineering & Technology, Hubli.
10. W. Weaver, Jr. and James M. Gere (1980). *Matrix Analysis of Framed Structures.*Van Nostraned Reinhold Company Inc., New York.
11. Wilson, Edward L.(2002). *Three-Dimensional Static and Dynamic Analysis of Structures*. Computers & Structures, Inc., California.

Appendix I: Flowchart for the Program

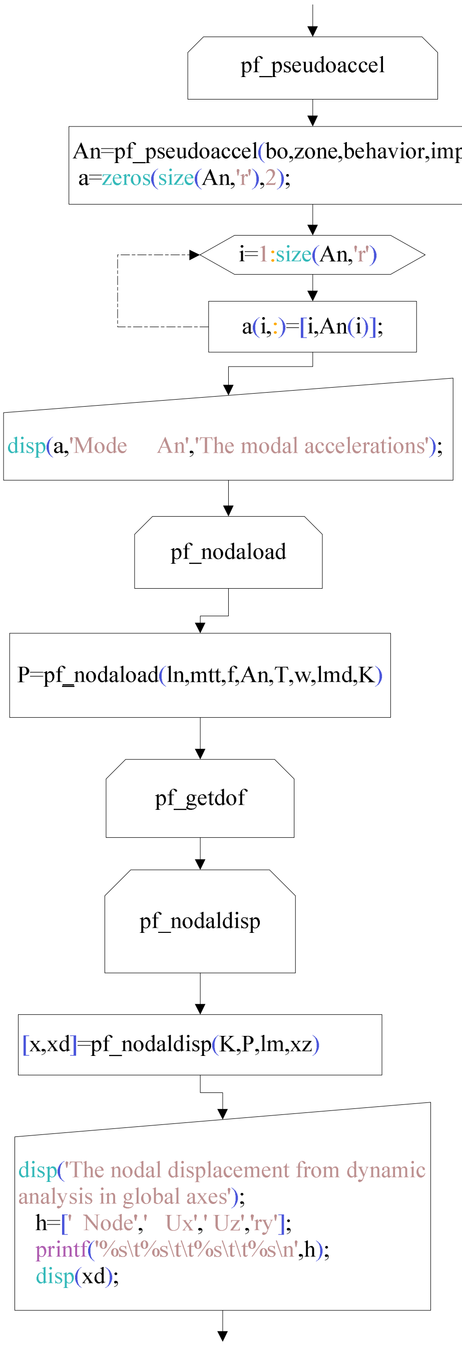
Main Program for Plane Frame Analysis



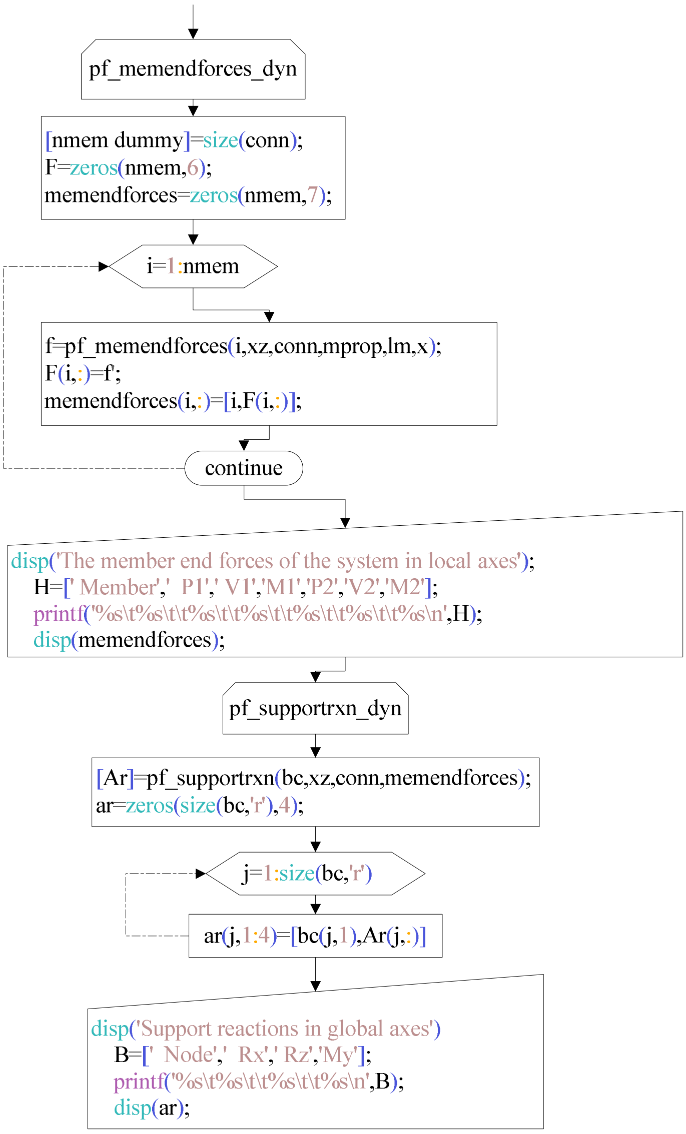
- Insert input data of the frame. (Structural data, Load data and Site data).
- A function to draw the members of the frame.
- A function to calculate the number of degree of freedom(DOF) of a frame and to give DOF number for each DOF.
- A function to calculate stiffness matrix of a member in member axes.
- A function to calculate length and direction cosine of a member.
- A function to calculate transformation matrix of a member.
- A function to calculate stiffness matrix of a member in global axes.
- A function to assemble joint stiffness matrix of a frame.
- A function to extract the upper band of stiffness matrix (free degree of freedom only).
- Calculate number of joints 'nodes'.
- Calculate location matrix 'lm' and number of degrees of freedom 'ndof' from function 'pf_calclm'.
- Display ndof on scilab window with heading 'Number of degrees of freedom'.
- Calculate number of members 'nmem' .
- Length and direction cosines of imemth member.
- Member property of the member.
- Cross-sectional parameters of the member.
- Stiffness matrix of the member.
- Transformation matrix of the member.
- Global stiffness of the member.
- Assemble the member stiffness matrix.
- Assembled structure stiffness matrix.



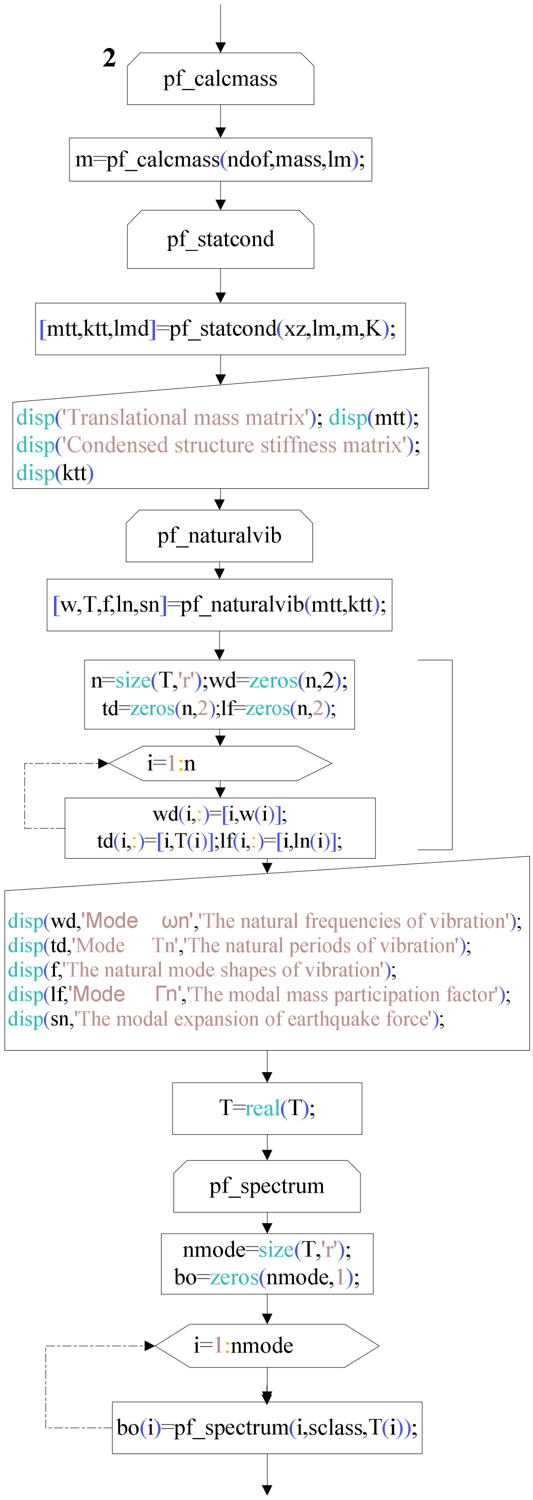
- If analysis(1) is not equal to 1, go to 2 and process another set of programs. i.e. 'dynamic analysis' is not selected.
- A function to calculate the mass matrix of the structure.
- Calculate the mass matrix of the system.
- A function to condense stiffness matrix of a structure.
- Calculate translational mass matrix and condensed stiffness matrix.
- A function to make modal analysis of the system.
- Calculate natural frequencies, periods, modal shapes of vibration, modal mass participation factor and modal expansion of earthquake forces.
- Preparing the modal properties for displaying them with their mode number.
- Display the natural frequencies, natural periods, mode shapes, modal mass participation and expansion of force on scilab window with heading 'The natural frequencies of vibration', 'The natural periods of vibration', 'The natural modes of vibration', The modal mass participation factor', 'The modal expansion of earthquake force' respectively.
- Real part of the T matrix. i.e. ignore zero imaginary part.
- A function to get the value of the bedrock acceleration 'βo' from the design spectrum given by EBCS 8.
- Number of modes.
- Initialize the bedrock acceleration for each mode to zero.
- Bedrock acceleration for each mode.



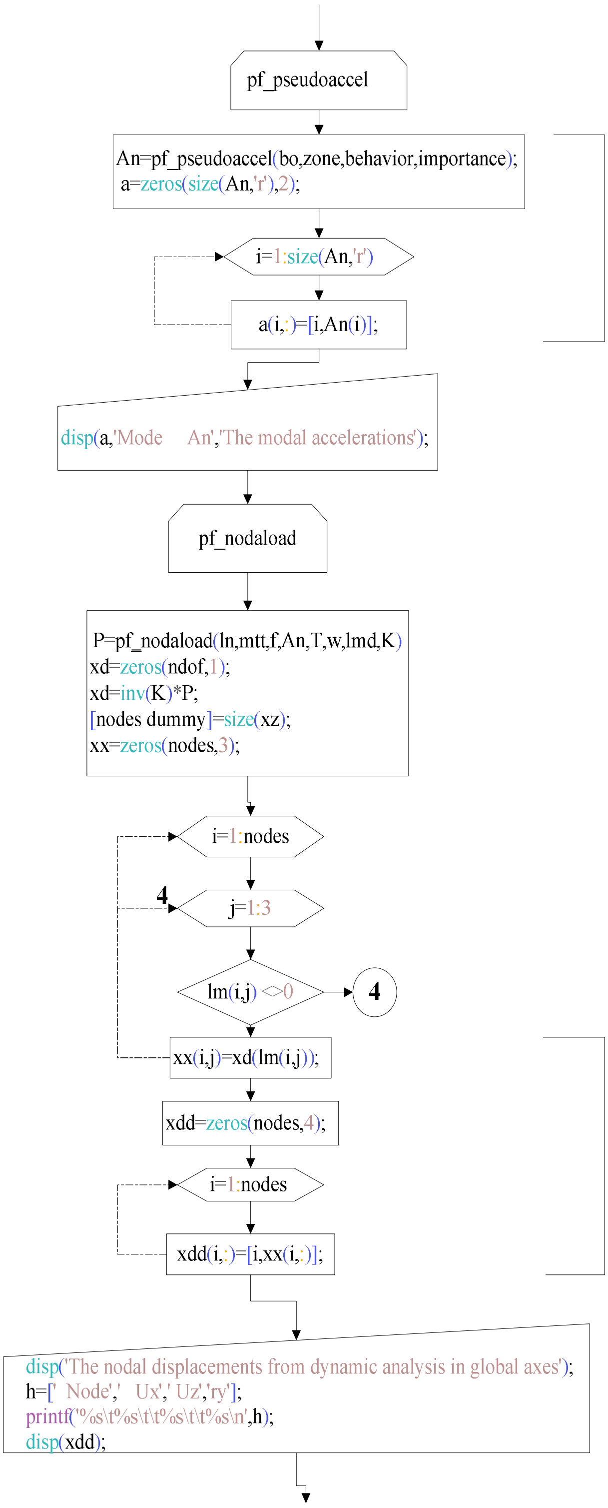
- A function to calculate modal accelerations.
- Preparing modal accelerations to display them with their mode number.
- Display the modal accelerations on scilab window with heading 'The modal accelerations'.
- A function to calculate the equivalent nodal load of the system by combining the modal loads using SRSS or CQC method.
- Calculate the equivalent nodal load of the system.
- A function to get DOF number for a member ends.
- A function to calculate the nodal displacements by solving the stiffness equation.
- Calculate the nodal displacements.
- Display the nodal displacements on scilab window with heading 'The nodal displacement from dynamic analysis in global axes
' 'Node Ux Uz ry'.



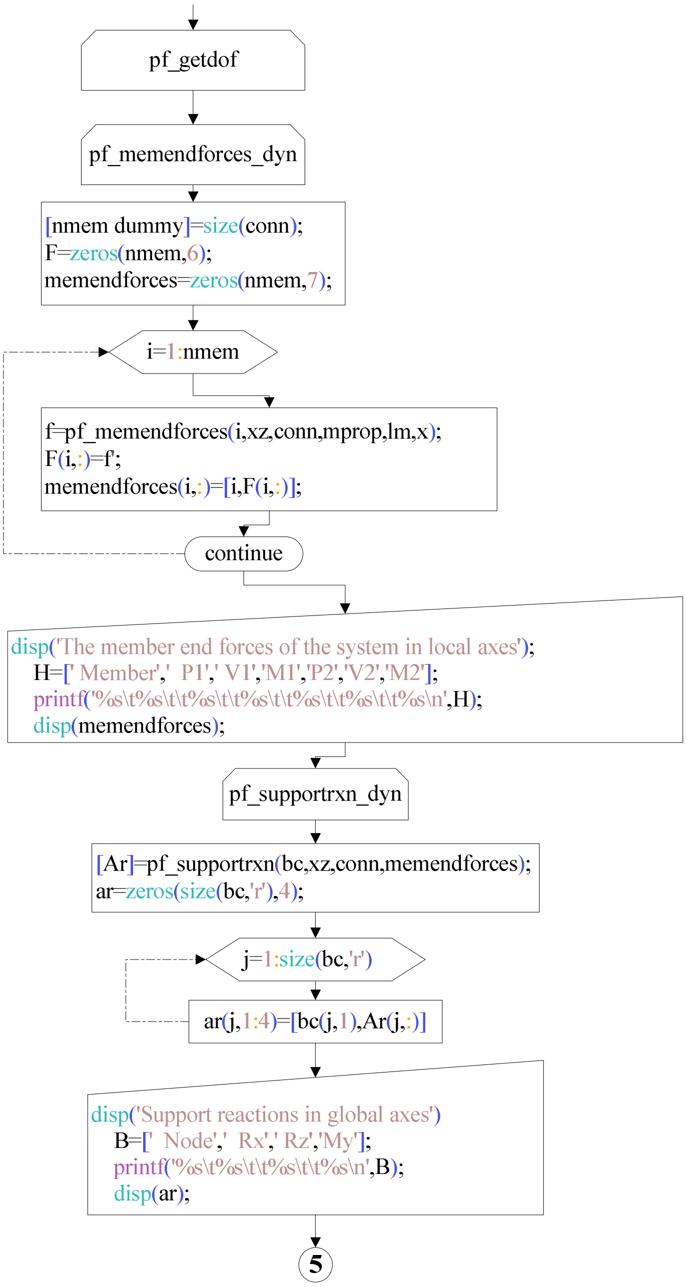
- A function to get member end forces of each member in their axes.
- Calculate number of member.
- Initialize F to zero.
- Initialize member end forces to zero.
- Index on member end forces.
- Calculate the member end forces.
- Display the member end forces on scilab window with heading:
  'The member end forces of the system in local axes');
  'Member P1 V1 M1 P2 V2 M2'
- A function to get the support reactions of restrained nodes.
- Calculate the support reactions.
- Index on support reactions.
- Support reactions with their node.
- Display the support reactions on scilab window with heading:
  'Support reactions in global axes'
  ' Node Rx Rz My'



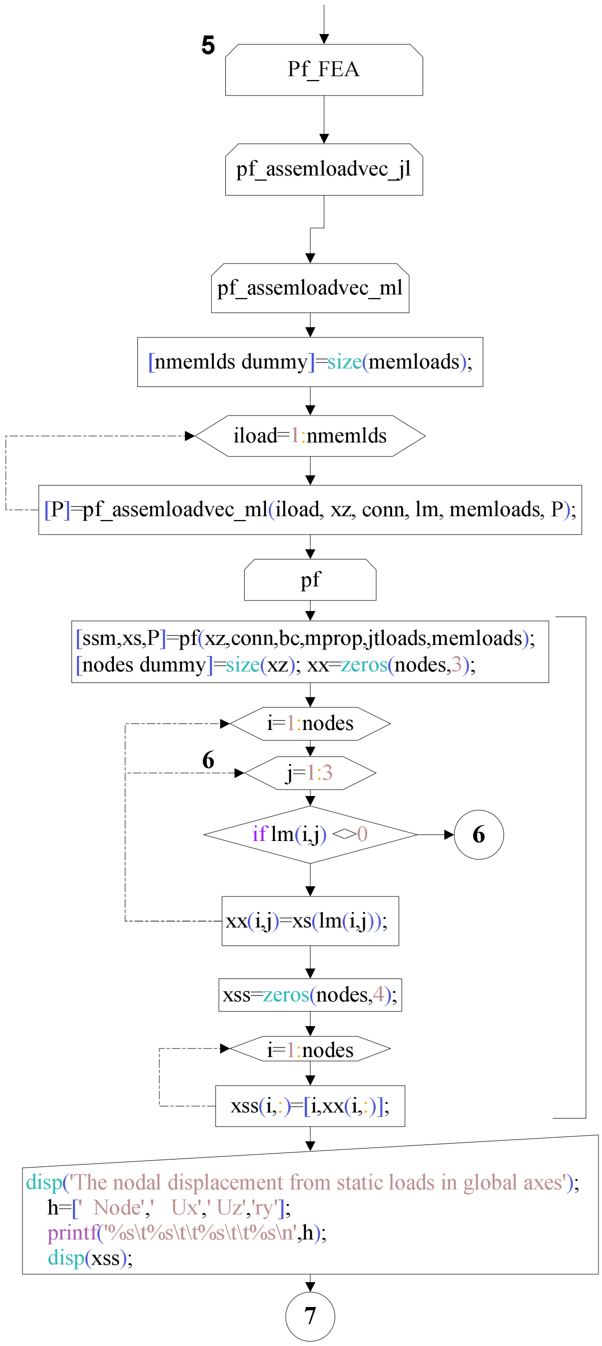
- A function to calculate the mass matrix of the structure.
- Calculate the mass matrix of the system.
- A function to condense stiffness matrix of a structure.
- Calculate translational mass matrix and condensed stiffness matrix.
- Display translational mass matrix on scilab window with heading 'Translational mass matrix'.
- Display condensed stiffness matrix on scilab window with heading 'Condensed structure stiffness matrix'.
- A function to make modal analysis of the system.
- Calculate natural frequencies, periods, modal shapes of vibration, modal mass participation factor and modal expansion of earthquake forces.
- Preparing the modal properties for displaying them with their mode number.
- Display the natural frequencies, natural periods, mode shapes, modal mass participation and expansion of force on scilab window with heading 'The natural frequencies of vibration', 'The natural periods of vibration', 'The natural modes of vibration', The modal mass participation factor', 'The modal expansion of earthquake force' respectively.
- Real part of the T matrix. i.e. ignore zero imaginary part.
- A function to get the value of the bedrock acceleration 'βo' from the design spectrum given by EBCS 8.
- Number of modes.
- Initialize the bedrock acceleration for each mode to zero.
- Bedrock acceleration for each mode.



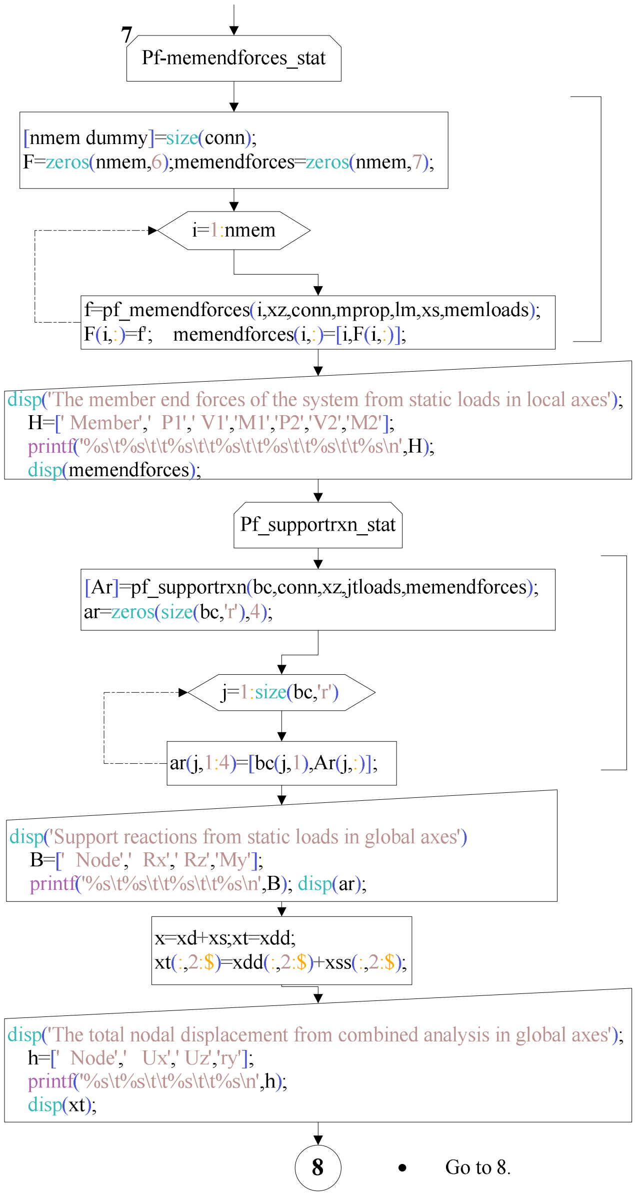
- A function to calculate modal accelerations.
- Preparing modal accelerations to display them with their mode number.
- Display the modal accelerations on scilab window with heading 'The modal accelerations'.
- A function to calculate the equivalent nodal load of the system by combining the modal loads using SRSS or CQC method.
- Calculate the equivalent nodal load of the system.
- Solve the stiffness equation for nodal displacements due to dynamic loading.
- Indexes on nodal displacements.
- If lm(i,j) is equal to zero, go to 4.
- Calculate nodal displacements from dynamic loading only.
- Display the nodal displacements on scilab window with heading: 'The nodal displacements from dynamic analysis in global axes' 'Node Ux Uz ry'



- A function to get dof number of member ends.
- A function to get member end forces of each member in their axes.
- Calculate number of member.
- Initialize F to zero.
- Initialize member end forces to zero.
- Index on member end forces.
- Calculate the member end forces.
- Display the member end forces on scilab window with heading 'The member end forces of the system from dynamic loads in loc axes'); 'Member P1 V1 M1 P2 V2 M2'
- A function to get the support reactions of restrained nodes.
- Calculate the support reactions.
- Index on support reactions.
- Support reactions with their node.
- Display the support reactions on scilab window with heading: 'Support reactions from dynamic loads in global axes' ' Node Rx Rz My'



- A function to calculate the fixed end actions from member loads.
- A function to assemble applied nodal loads.
- A function to assemble applied loads on members.
- Number of applied member loads.
- Assembled applied load vector.
- A function to solve the stiffness equation.
- Calculate nodal displacements due to applied static loads.
- Display the nodal displacements from static loads with heading:
  'The nodal displacement from static loads in global axes'
  'Node Ux Uz ry'
- Go to 7.



- A function to calculate member end forces from static loads.
- Member end forces of the system from static loads.
- Display the member end forces with heading:
  'The member end forces of the system from static loads in local axes'
  'Member P1 V1 M1 P2 V2 M2'
- A function to calculate the support reactions from static loads.
- Support reactions from static loads.
- Display the support reactions from static loads with heaing:
  'Support reactions from static loads in global axes'
  'Node Rx Rz My'
- Linearly superposed nodal displacements from dynamic and static loads.
- Display the nodal displacements with heading:
  'The total nodal displacement from combined analysis in global axes'
  'Node Ux Uz ry'
- Go to 8.

8

```
Pf_memendforces_stat
```

- A function to calculate the member end forces from combined dynamic and static loads.

```
[nmem dummy]=size(conn);
F=zeros(nmem,6);memendforces=zeros(nmem,7);

i=1:nmem

f=pf_memendforces(i,xz,conn,mprop,lm,x,memloads);
F(i,:)=f';
memendforces(i,:)=[i,F(i,:)];
```

- Member end forces of the system by linearly superposing the dynamic and static responses.

```
disp('The combined member end forces of the system in local axes');
  H=[' Member',' P1',' V1','M1','P2','V2','M2'];
  printf('%s\t%s\t\t%s\t\t%s\t\t%s\t\t%s\t\t%s\n',H);
  disp(memendforces);
```

- Display the member end forces with heading: 'The combined member end forces of the system in local axes' 'Member P1 V1 M1 P2 V2 M2

```
Pf_supportrxn_stat
```

- A function to calculate the support reactions from dynamic and static loads.

```
[Ar]=pf_supportrxn(bc,conn,xz,jtloads,memendforces);
ar=zeros(size(bc,'r'),4);

j=1:size(bc,'r')

ar(j,1:4)=[bc(j,1),Ar(j,:)];
```

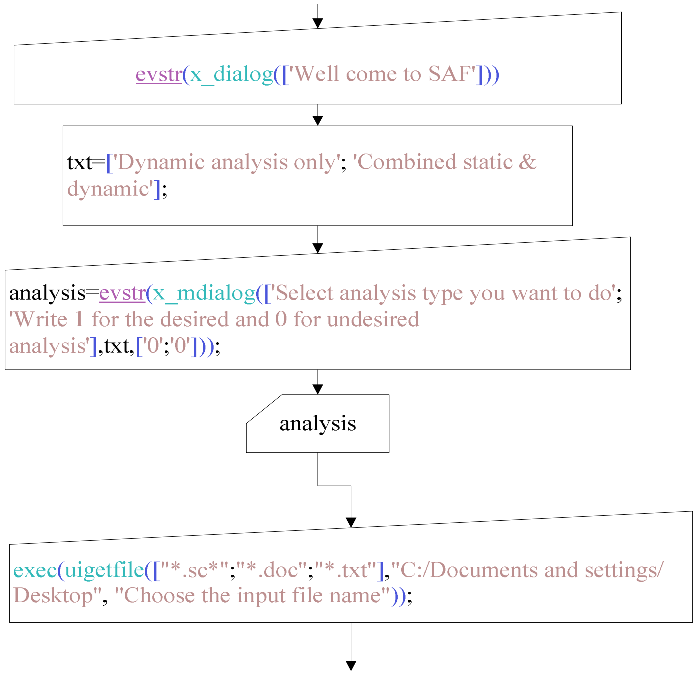
- Support reactions from combined dynamic and static loads.

```
disp('Support reactions for combined loading in global axes')
  B=[' Node',' Rx',' Rz','My'];
  printf('%s\t%s\t\t%s\t\t%s\n',B);
  disp(ar);
```

- Display the support reactions with heading: 'Combined support reactions in global axes' ' Node Rx Rz My'

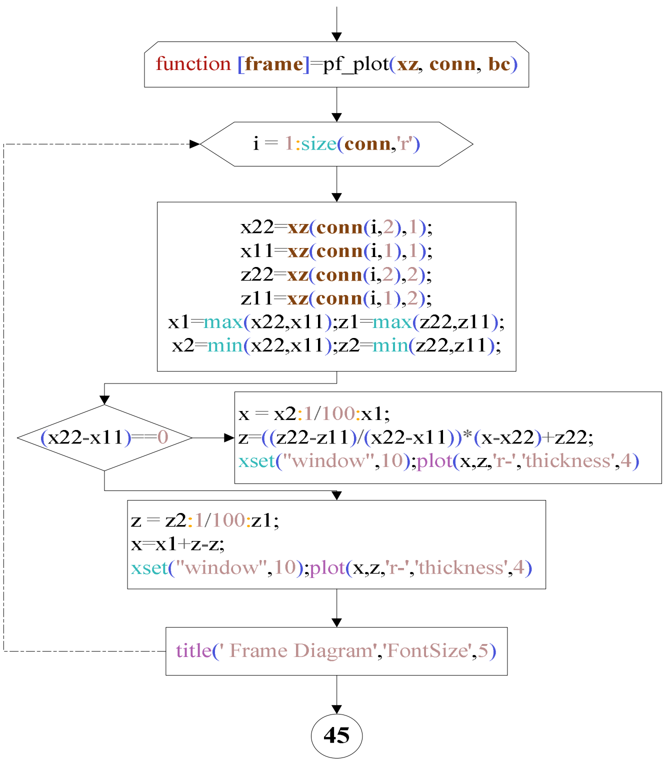
Flowcharts for subprograms

1. Subprogram 'Input data'

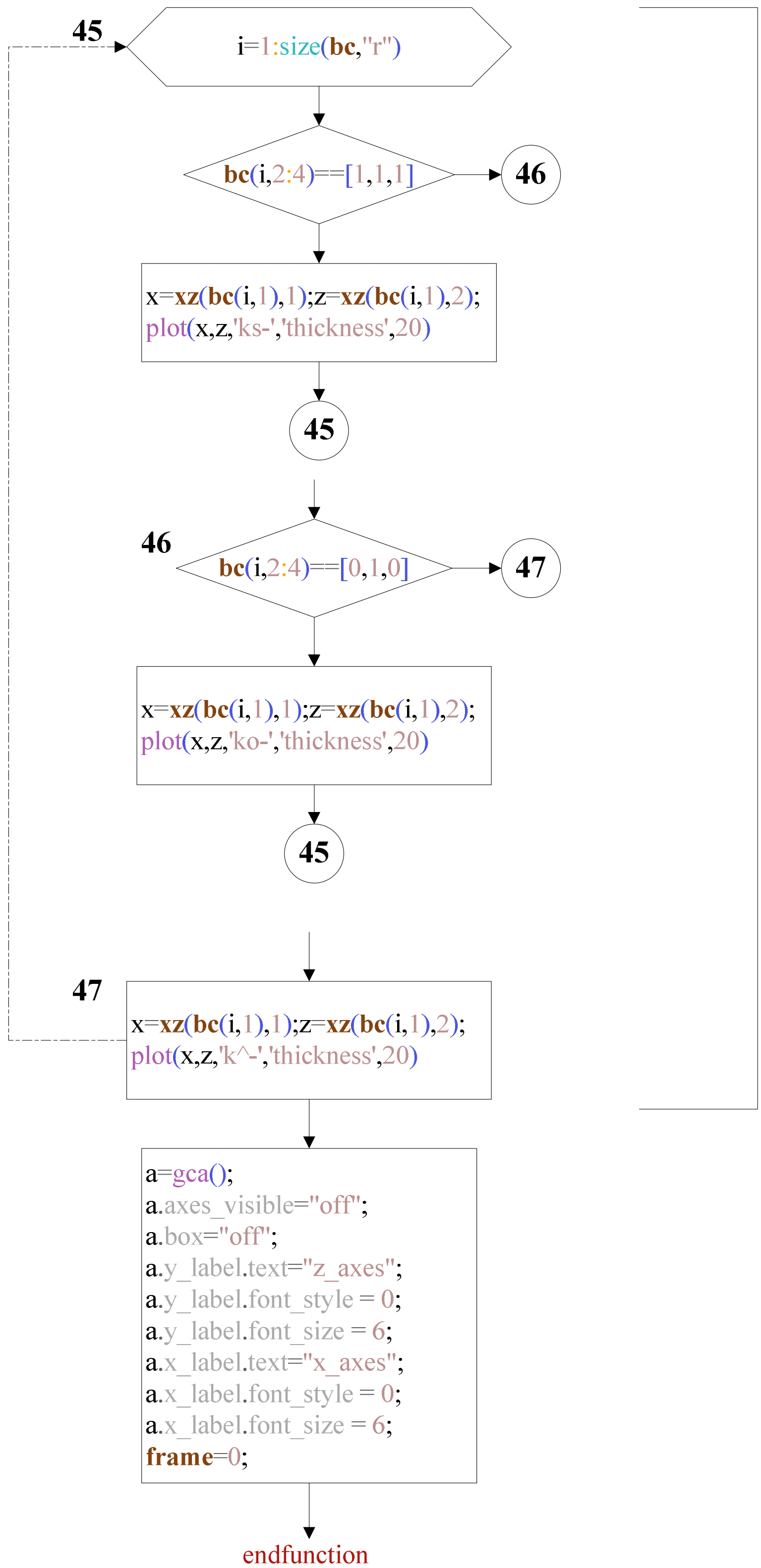


- Display dialog with statements: Well come to SAF
- Types of analysis which can be done by the program.
- Display dialog to select types of analysis.
- Insert type of analysis (Dynamic, Static or Combined dynamic and static analysis) to the displayed dialog.
- Display dialog to insert the path of the input file location and execute the file to the scilab window.

2. Function 'pf_plot'



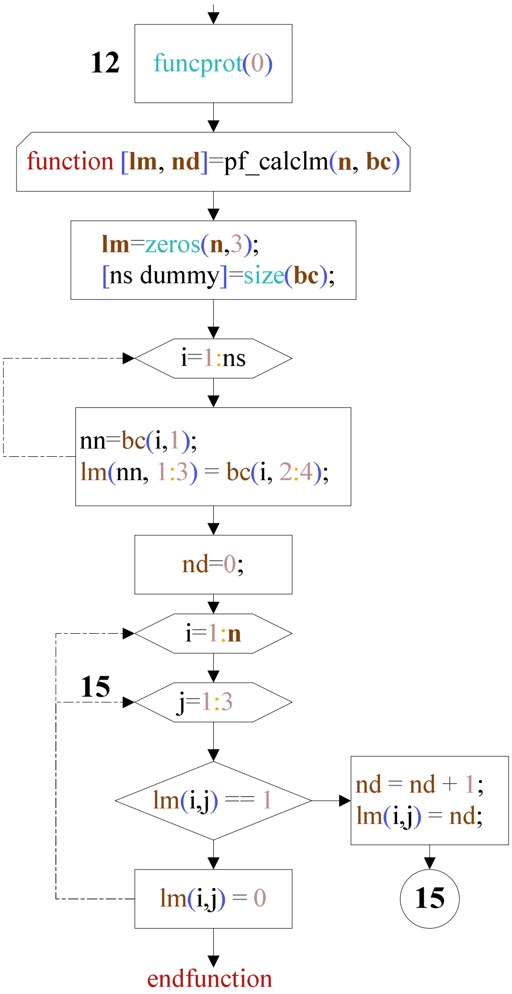
- A function to draw the members of a frame.
- Index on member
- X-Coordinate for end node of the member
- X-Coordinate for start node of the member
- Z-Coordinate for end node of the member
- Z-Coordinate for start node of the member
- Maximum of the coordinates
- Minimum of the coordinates
- Draw the member
- Draw the member in graphic window number 10.
- Give title for the frame.



- Draw the restraint type

- Setting of the coordinate system at which the frame was drawn.

3. Function 'pf_calclm'



- Do not display the message 'Redefining function ...' when using more than one function in the program.
- A function to assign a DOF number for unrestrained nodes.
- Initialize location matrix to zero.
- Calculate number of restrained nodes.
- Index on boundary constraint matrix.
- Node number of i^{th} support.
- Constraint codes of i^{th} support.
- Initialize number of DOF to zero.
- Indexes on location matrix.
- Unconstrained DOF.
- Go to 15.

4. Function 'pf_stiff'

```
function [k]=pf_stiff(E, A, I, L)

k = zeros(6,6);
k(1,1) = E*A/L; k(1,4) = -k(1,1);
k(2,2) = (12*E*I)/L^3; k(2,3) = (6*E*I)/L^2;
k(2,5) = -k(2,2); k(2,6) = k(2,3);
k(3,3) = (4*E*I)/L;
k(3,5) = -k(2,3); k(3,6) = k(3,3)/2;
k(4,4) = k(1,1);
k(5,5) = k(2,2); k(5,6) = -k(2,3);
k(6,6) = k(3,3);

i = 2:6
  j = 1:i-1
    k(i,j) = k(j,i);
  continue

endfunction
```

- Function to get member stiffness matrix in local axes.
- The local stiffness matrix of a member above diagonal.
- Symmetry of the stiffness matrix.

5. Function 'pf_calclen'

```
function [l, cx, cz]=pf_calclen(imem, xz, conn)

n=conn(imem,:);
p1 = xz(n(1),:);
p2 = xz(n(2),:);
dxdz=p2-p1;
dxdz2=dxdz.^2;
l=sqrt(sum(dxdz2));
cx = dxdz(1) / l;
cz = dxdz(2) / l;

endfunction
```

- A function to calculate length and direction cosines of a member.
- Start and end nodes of member number imem
- x,z coordinates of start node
- x,z coordinates of end node
- x,z projections of member
- Square of the projections
- Length of member
- X-direction direction cosine of member
- Z-direction direction cosine of member

6. Function ‘pf_calcrot’

```
function [r]=pf_calcrot(cx, cz)

r = zeros(6,6);
r(1,1) = cx;
r(1,2) = cz;
r(2,1) = -r(1,2);
r(2,2) = r(1,1);
r(3,3) = 1;
r(4:6, 4:6) = r(1:3, 1:3);

endfunction
```

- A function to calculate the transformation matrix of a mem
- Rotation matrix of a member for the start node.
- Copy rotation matrix of start node to end node of a membe

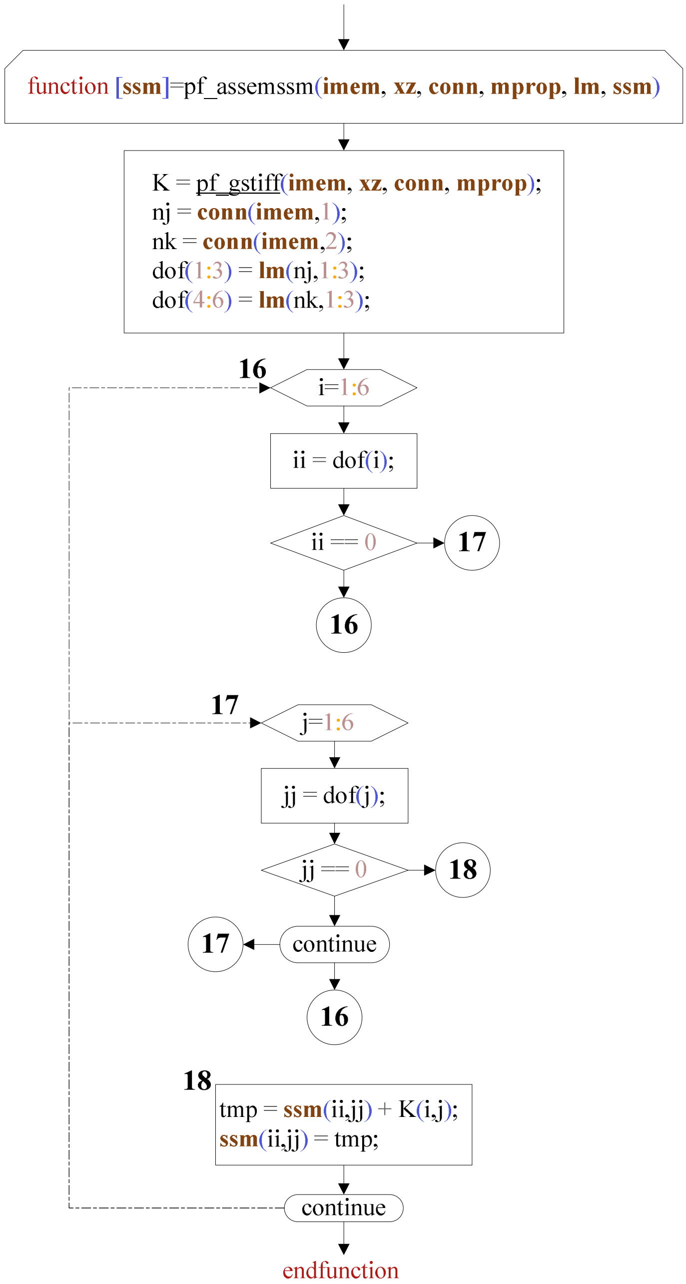
7. Function ‘pf_gstiff’

```
function [K]=pf_gstiff(imem, xz, conn, mprop)

[L cx cz] = pf_calclen(imem, xz, conn);
r = pf_calcrot(cx, cz);
iprop = conn(imem,3);
E = mprop(iprop,1);
A = mprop(iprop,2);
I = mprop(iprop,3);
k = pf_stiff(E, A, I, L);
K = r' * k * r;

endfunction
```

- A function to transform local to global stiffness matrix.

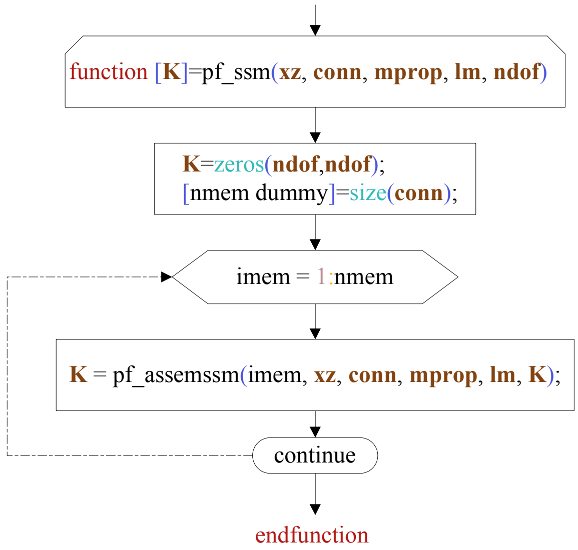
- Calculate length and direction cosines
- Calculate rotation matrix
- Property id of imemth member
- Modulus of elasticity
- Area of cross section
- Second moment of area of cross section about neutral axes
- Local stiffness matrix of ith member
- Global stiffness matrix of imemth member

8. Function ‘pf_assemssm’



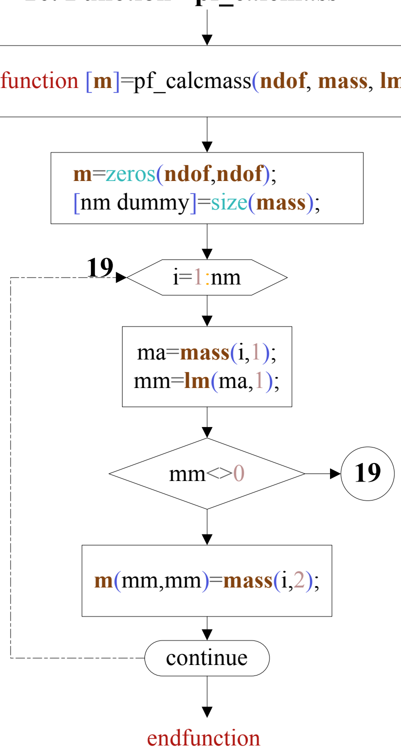
- A function to assemble global stiffness matrix for only unrestrained DOF.
- Global stiffness matrix of imem[th] member
- Start node number of the member
- End node number of the member
- DOF number of the start node.
- DOF number of the end node.
- If ii is not equal to zero, go to 17.
- Go to 16.
- If jj is not equal to zero, go to 18
- Go to 16.
- Assemble structure stiffness matrix for unrestrained DOFs.

9. Function ‘pf_ssm’



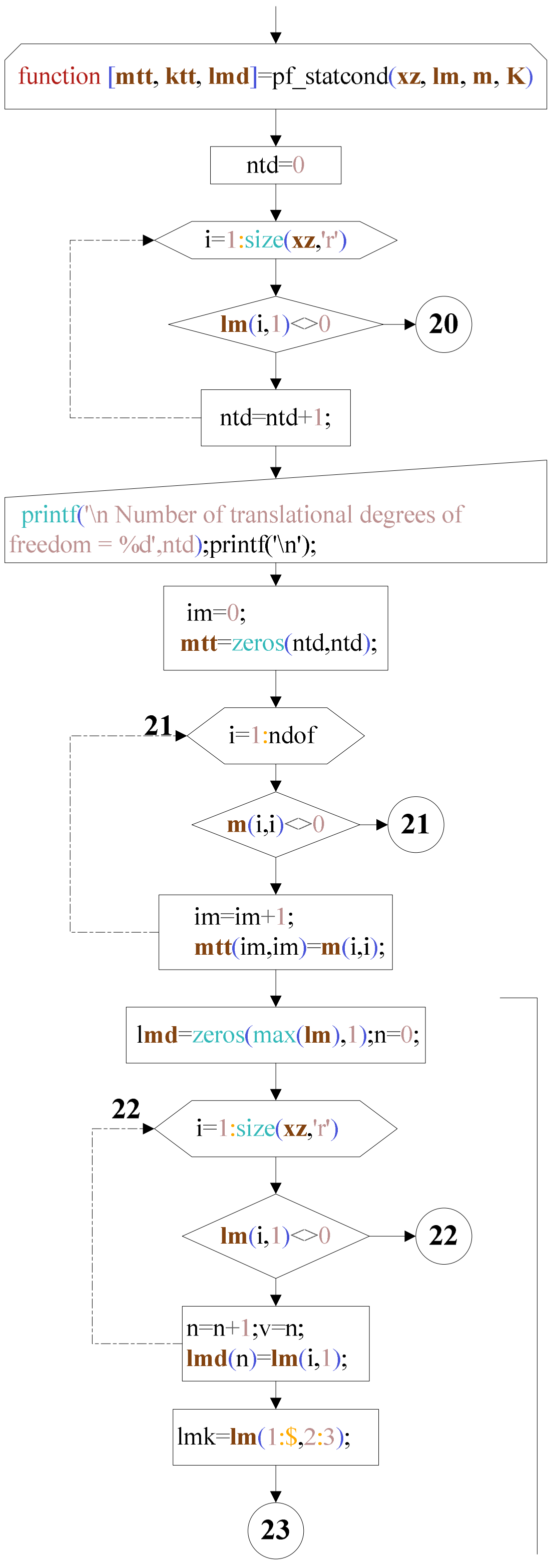
- A function to assemble structure stiffness matrix of all members.
- Initialize structure stiffness matrix to zero.
- Number of members.
- Structure stiffness matrix.

10. Function ‘pf_calcmass’

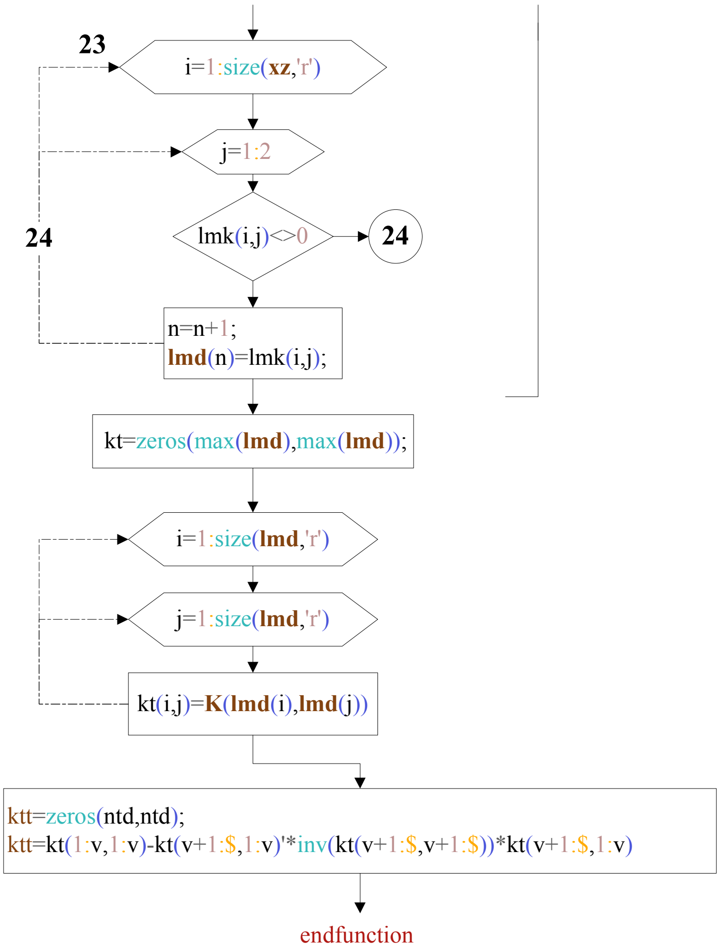


- A function to assemble the lumped mass of each node at respective DOF number..
- Initialize the mass matrix to zero.
- Number of lumped masses.
- Node number of the i^{th} lumped mass.
- DOF number of the node (only translational DOF).
- If mm=0, go to 19.
- Mass matrix.

11. Function ‘pf_statcond’

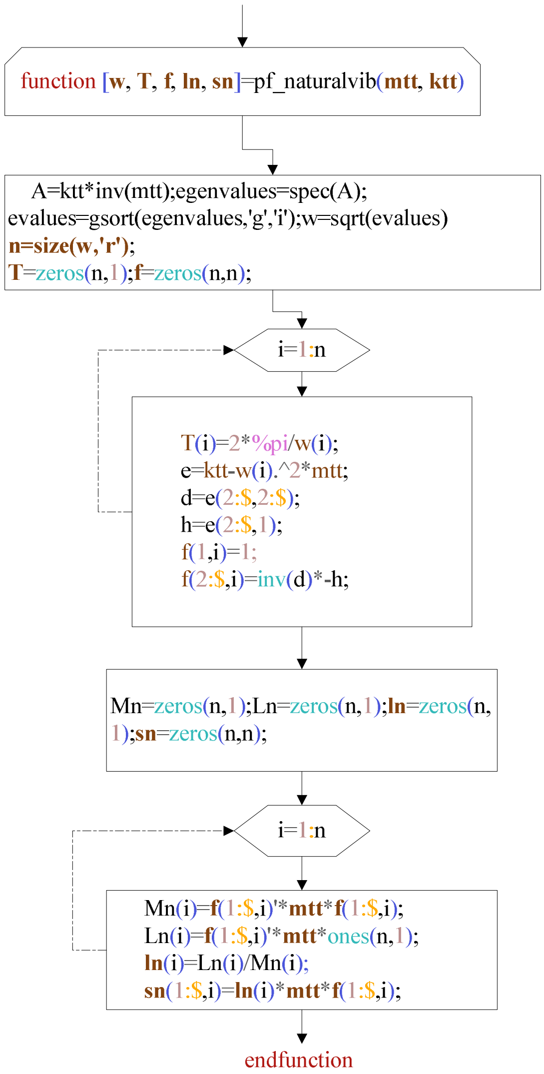


- A function to condense the stiffness matrix.
- Initialize translational DOF to zero.
- Index on nodes.
- If lm(i,1) = 0, go to 20.
- Number of translational DOFs.
- Display number of translational DOF with heading: 'Number of translational degrees of freedom’
- Initialize translational mass matrix to zero.
- Index on translational mass matrix.
- If m(i,i) = 0, go to 21.
- Translational mass matrix.
- Preparing data for static condensation of stiffness matrix.



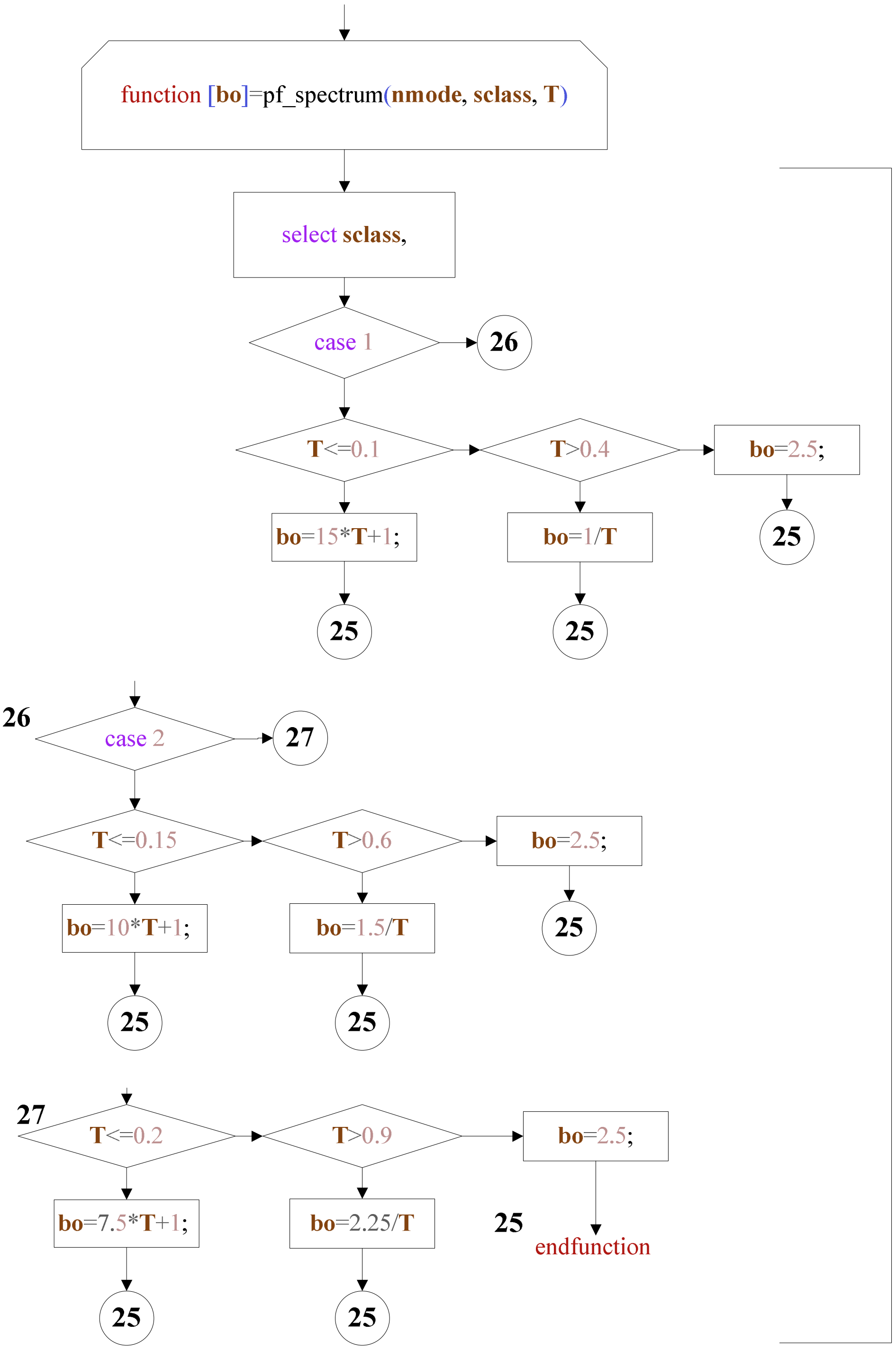
- Preparation of data to bring stiffness matrix for translational DOF to the upper part of stiffness matrix
- Initialize rearranged stiffness matrix to zero.
- Indexes on condensed stiffness matrix.
- Bring stiffness matrix for translational DOF to upper part of stiffness matrix.
- Condensed stiffness matrix.

12. Function 'pf_naturalvib'



- A function to make modal analysis.
- Develop characteristic equation and solve eigenvalue problem.
- Number of modes.
- Initialize natural vibration parameters to zero.
- Index on modes.
- Natural vibration period
- Natural mode shapes
- Initialize modal properties to zero.
- Index on modes.
- Modal mass property
- Modal mass participation
- Modal mass participation factor
- Modal expansion of earthquake force

13. Function 'pf_spectrum'



- A function to get the ordinate of the design spectrum for a mode.
- Ordinate of the design spectrum.

Appendix II: Program Code

In this section, the coding of the developed program is provided.

The concept of code for static analysis is taken from Satish Annigeri, 2008with modifications of the input data and some programming functions.

```
evstr (x_dialog(['Welcome to SAF']));
txt=['Dynamic analysis only'; 'Combined static & dynamic'];
analysis=evstr(x_mdialog(['Select analysis type you want to do'; 'Write 1 for the desired and 0 for
undesired analysis'],txt,['0';'0']));
exec(uigetfile(["*.sc*";"*.doc";"*.txt"],"C:/Documents and settings/Desktop", "Choose the input file
name"));
//Draw the frame members
function[frame]=pf_plot(xz, conn, bc)
fori=1:size(conn,'r')
x22=xz(conn(i,2),1);x11=xz(conn(i,1),1);
z22=xz(conn(i,2),2);z11=xz(conn(i,1),2);
x1=max(x22,x11);z1=max(z22,z11);
x2=min(x22,x11);z2=min(z22,z11);
if(x22-x11)==0then,
z=z2:1/100:z1;x=x1+z-z;
xset("window",10);plot(x,z,'r-','thickness',4)
else
x=x2:1/100:x1;z=((z22-z11)/(x22-x11))*(x-x22)+z22;
xset("window",10);plot(x,z,'r-','thickness',4)
end
title(' Frame Diagram','FontSize',5)
end
fori=1:size(bc,"r")
ifbc(i,2:4)==[1,1,1]then
x=xz(bc(i,1),1);z=xz(bc(i,1),2);plot(x,z,'ks-','thickness',20)
elseifbc(i,2:4)==[0,1,0]then
x=xz(bc(i,1),1);z=xz(bc(i,1),2);plot(x,z,'ko-','thickness',20)
else
x=xz(bc(i,1),1);z=xz(bc(i,1),2);plot(x,z,'k^-','thickness',20)
end
end
a=gca();a.axes_visible="off";a.box="off";
a.y_label.text="z_axes";a.y_label.font_style=0;a.y_label.font_size=6;
a.x_label.text="x_axes";a.x_label.font_style=0;a.x_label.font_size=6;
frame=0;
endfunction
[frame]=pf_plot(xz,conn,bc);
```

```
//Location matrix
funcprot(0)
function[lm, nd]=pf_calclm(n, bc)
lm=zeros(n,3);
[nsdummy]=size(bc);// ns=number of supports
fori=1:ns
nn=bc(i,1);// node number of ith support
lm(nn,1:3)=bc(i,2:4);// constraint codes of ith support
end;
nd=0;// initialize number of dof to zero
fori=1:n
forj=1:3
iflm(i,j)==1then// constrained dof
lm(i,j)=0;
else// unconstrained dof
nd=nd+1;lm(i,j)=nd;
end
end
end
endfunction
//Local stiffness matrix
function[k]=pf_stiff(E, A, I, L)
k=zeros(6,6);
k(1,1)=E*A/L;k(1,4)=-k(1,1);
k(2,2)=(12*E*I)/L^3;k(2,3)=(6*E*I)/L^2;k(2,5)=-k(2,2);k(2,6)=k(2,3);
k(3,3)=(4*E*I)/L;k(3,5)=-k(2,3);k(3,6)=k(3,3)/2;
k(4,4)=k(1,1);
k(5,5)=k(2,2);k(5,6)=-k(2,3);
k(6,6)=k(3,3);
fori=2:6
forj=1:i-1
k(i,j)=k(j,i);
end
end
endfunction
//Length and direction cosine
function[l, cx, cz]=pf_calclen(imem, xz, conn)
n=conn(imem,:);// start and end nodes of member number i
p1=xz(n(1),:);// x,z coordinates of start node
p2=xz(n(2),:);// x,z coordinates of end node
dxdz=p2-p1;// x,z projections of member
dxdz2=dxdz.^2;// square of the projections
l=sqrt(sum(dxdz2));// length of member
```

```
cx=dxdz(1)/l;// x-direction direction cosine of member
cz=dxdz(2)/l;// z-direction direction cosine of member
endfunction
//Rotation matrix
function[r]=pf_calcrot(cx, cz)
r=zeros(6,6);// initialize rotation matrix to zero
r(1,1)=cx;r(1,2)=cz;
r(2,1)=-r(1,2);r(2,2)=r(1,1);
r(3,3)=1;
r(4:6,4:6)=r(1:3,1:3);// copy rows and columns 1:3, 1:3 into 4:6, 4:6
endfunction
//Global stiffness
function[K]=pf_gstiff(imem, xz, conn, mprop)
[Lcxcz]=pf_calclen(imem,xz,conn);// length and direction cosines
r=pf_calcrot(cx,cz);// calculate rotation matrix
iprop=conn(imem,3);// property id of imem th member
E=mprop(iprop,1);// Modulus of elasticity
A=mprop(iprop,2);// Area of cross section
I=mprop(iprop,3);// Second moment of area of cross section about NA
k=pf_stiff(E,A,I,L);// local stiffness matrix of ith member
K=r'*k*r;// global stiffness matrix of imem th member
endfunction
//Assembling stiffness matrix
function[ssm]=pf_assemssm(imem, xz, conn, mprop, lm, ssm)
K=pf_gstiff(imem,xz,conn,mprop);
nj=conn(imem,1);//start node number of the member
nk=conn(imem,2);//end node number of the member
dof(1:3)=lm(nj,1:3);dof(4:6)=lm(nk,1:3);
fori=1:6
ii=dof(i);
ifii==0then
else
forj=1:6
jj=dof(j);
ifjj==0then
else
tmp=ssm(ii,jj)+K(i,j);ssm(ii,jj)=tmp;
end
end
end
end
endfunction
function[K]=pf_ssm(xz, conn, mprop, lm, ndof)
```

```
K=zeros(ndof,ndof);[nmemdummy]=size(conn);
forimem=1:nmem
K=pf_assemssm(imem,xz,conn,mprop,lm,K);
end
endfunction
[nodesdummy]=size(xz);[lm,ndof]=pf_calclm(nodes,bc);
printf (' Number of degrees of freedom = %d',ndof);
[nmemdummy]=size(conn);
forimem=1:nmem
[Lcxcz]=pf_calclen(imem,xz,conn);iprop=conn(imem,3);
E=mprop(iprop,1);A=mprop(iprop,2);I=mprop(iprop,3);
k=pf_stiff(E,A,I,L);r=pf_calcrot(cx,cz);
k=pf_gstiff(imem,xz,conn,mprop);K=pf_ssm(xz,conn,mprop,lm,ndof);
K=pf_assemssm(imem,xz,conn,mprop,lm,K);
end
K=pf_ssm(xz,conn,mprop,lm,ndof);//Free joint stiffness matrix
//Dynamic analysis
ifanalysis(1)==1then
//Mass matrix
function[m]=pf_calcmass(ndof, mass, lm)
m=zeros(ndof,ndof);
[nmdummy]=size(mass);//Number of lumped masses
fori=1:nm
ma=mass(i,1);//node number at which the ith mass exists
mm=lm(ma,1);//degree of freedom number of the node number at which the ith mass exists
ifmm<>0then//For unrestrained node in horizontal translation only.
m(mm,mm)=mass(i,2);
end
end
endfunction
m=pf_calcmass(ndof,mass,lm);
//Static condensation
function[mtt, ktt, lmd]=pf_statcond(xz, lm, m, K)
ntd=0;//initialize horizontal translational DOF to zero.
fori=1:size(xz,'r')
iflm(i,1)<>0then
ntd=ntd+1;//number of horizontal translational DOFs
end
end
printf('\n Number of horizontal translational degrees of freedom =
%d',ntd);printf('\n');im=0;mtt=zeros(ntd,ntd);
fori=1:ndof
ifm(i,i)<>0then
```

```
im=im+1;
mtt(im,im)=m(i,i);//only horizontal translational mass matrix
end
end
lmd=zeros(max(lm),1);n=0;
fori=1:size(xz,'r')
iflm(i,1)<>0then
n=n+1; v=n; lmd(n)=lm(i,1);
end
end
lmk=lm(1:$,2:3);
fori=1:size(xz,'r')
forj=1:2
iflmk(i,j)<>0then
n=n+1;lmd(n)=lmk(i,j);
end
end
end
kt=zeros(max(lmd),max(lmd));
fori=1:size(lmd,'r')
forj=1:size(lmd,'r')
kt(i,j)=K(lmd(i),lmd(j));// Bringing horizontal translational DOF to the top
end
end
ktt=zeros(ntd,ntd);
ktt=kt(1:v,1:v)-kt(v+1:$,1:v)'*inv(kt(v+1:$,v+1:$))*kt(v+1:$,1:v);//condensation of stiffness matrix
endfunction
[mtt,ktt,lmd]=pf_statcond(xz,lm,m,K);
//Free vibration analysis
function[w, T, f, ln, sn]=pf_naturalvib(mtt, ktt)
A=ktt*inv(mtt);egenvalues=spec (A);evalues=gsort (egenvalues,'g','i');
w=sqrt (evalues);//The natural frequencies of vibration of the system
        n=size(w,'r');T=zeros (n,1);f=zeros (n,n);
for i=1:n
T(i)=2*%pi/w(i);//The natural periods of vibration of the system
e=ktt-w(i).^2*mtt;d=e(2:$,2:$);h=e(2:$,1);
f(1,i)=1;f(2:$,i)=inv(d)*-h; //The natural mode shapes of vibration
end
Mn=zeros(n,1);Ln=zeros(n,1);ln=zeros(n,1);sn=zeros(n,n);
fori=1:n
Mn(i)=f(1:$,i)'*mtt*f(1:$,i);//Modal mass property
Ln(i)=f(1:$,i)'*mtt*ones(n,1);//Modal mass participation
ln(i)=Ln(i)/Mn(i);//Modal mass participation factor
```

```
sn(1:$,i)=ln(i)*mtt*f(1:$,i);//Modal expansion of earthquake force
end
endfunction
[w,T,f,ln,sn]=pf_naturalvib(mtt,ktt);
n=size(T,'r');
wd=zeros(n,2);td=zeros(n,2);lf=zeros(n,2);
fori=1:n
wd(i,:)=[i,w(i)];td(i,:)=[i,T(i)];lf(i,:)=[i,ln(i)];
end
disp(wd,'Mode     ωn','The natural frequencies of vibration');
disp(td,'Mode     Tn','The natural periods of vibration');
disp(f,'The natural mode shapes of vibration');
disp(lf,'Mode     Γn','The modal mass participation factor');
disp(sn,'The modal expansion of earthquake force');
//Modal acceleration
T=real(T);
function[bo]=pf_spectrum(nmode, sclass, T)
selectsclass,
case1then
ifT<=0.1thenbo=15*T+1;
elseifT>0.4thenbo=1/T;
elsebo=2.5;
end
case2then
ifT<=0.15thenbo=10*T+1;
elseifT>0.6thenbo=1.5/T;
elsebo=2.5;
end
case3then
ifT<=0.2thenbo=7.5*T+1;
elseifT>0.9thenbo=2.25/T;
elsebo=2.5;
end
end
endfunction
nmode=size(T,'r');bo=zeros(nmode,1);
fori=1:nmode
bo(i)=pf_spectrum(i,sclass,T(i));
end
function[An]=pf_pseudoaccel(bo, zone, behavior, importance)
ao=0;//Bedrock acceleration ratio for the site 'ao'
ifzone==4then
ao=0.1;
```

```
elseifzone==3then
ao=0.07;
elseifzone==2then
ao=0.05;
else
ao=0.03;
end
I=0;//Importance factor for buildings 'I'
ifimportance==1then
I=1.4;
elseifimportance==2then
I=1.2;
elseifimportance==3then
I=1.0;
else
I=0.8;
end
a=ao*I;gama=behavior;An=zeros(nmode,1);An=a*bo*gama*9.81;
endfunction
An=pf_pseudoaccel(bo,zone,behavior,importance);
a=zeros(size(An,'r'),2);
fori=1:size(An,'r')
a(i,:)=[i,An(i)];
end
disp(a,'Mode     An','The modal accelerations');
function[P]=pf_nodaload(ln, mtt, f, An, T, w, lmd, K)
n=size(T,'r');
fn=zeros(n,n);
fori=1:n
forj=1:n
fn(i,j)=ln(j)*mtt(i,i)*f(i,j)*An(j);//equivalent static force
end
end
Fjt=zeros(n,1);//initialize the equivalent nodal static load to zero
t=real(T);
ifmin(t)<=0.9*max(t)then//SRSS combination rule
fori=1:n
g=0;
forj=1:n
g=fn(i,j).^2+g;
end
Fjt(i)=sqrt(g);//equivalent nodal static load from SRSS method
end
```

```
else//CQC combination rule
b=zeros(n,n);p=zeros(n,n);
fork=1:n
g=0;
fori=1:n
forj=1:n
b(i,j)=w(i)/w(j);
p(i,j)=(0.0002*(1+b(i,j))*b(i,j).^1.5)/((1-b(i,j).^2).^2+0.0001*b(i,j)*(1+b(i,j)).^2);
g=p(i,j)*fn(i,k)*fn(j,k)+g;
end
end
Fjt(k)=sqrt(g);//equivalent nodal static load from CQC method
end
end

[ndofdummy]=size(K);
P=zeros(ndof,1);//Initialize the equivalent nodal load to zero.
fori=1:n
P(lmd(i))=Fjt(i);
end
endfunction
[P]=pf_nodaload(ln,mtt,f,An,T,w,lmd,K);
function[dof]=pf_getdof(imem, conn, lm)
dof=zeros(1,6);
n1=conn(imem,1);
dof(1:3)=lm(n1,:);//DOF number of the start node of the member 'imem'
n2=conn(imem,2);
dof(4:6)=lm(n2,:);//DOF number of the end node of the member 'imem'
endfunction
function[x, xd]=pf_nodaldisp(K, P, lm, xz)
ndof=max(lm);
K=pf_ssm(xz,conn,mprop,lm,ndof);
[P]=pf_nodaload(ln,mtt,f,An,T,w,lmd,K);
x=zeros(ndof,1);//initialize the nodal displacement to zero
x=inv(K)*P;//nodal displacement from dynamic analysis in global axes
[nodesdummy]=size(xz);
xx=zeros(nodes,3);
fori=1:nodes
forj=1:3
iflm(i,j)<>0then
xx(i,j)=x(lm(i,j));
end
end
```

```
end
xd=zeros(nodes,4);
fori=1:nodes
xd(i,:)=[i,xx(i,:)];
end
endfunction
[x,xd]=pf_nodaldisp(K,P,lm,xz);
disp('The nodal displacement from dynamic analysis in global axes');
h=['  Node','   Ux',' Uz','ry'];
printf('%s\t%s\t\t%s\t\t%s\n',h);
disp(xd);
function[f]=pf_memendforces(imem, xz, conn, mprop, lm, x)
iprop=conn(imem,3);// Property ID for member imem
E=mprop(iprop,1);A=mprop(iprop,2);I=mprop(iprop,3);
[Lcxcz]=pf_calclen(imem,xz,conn);// Length and direction cosines
r=pf_calcrot(cx,cz);// Rotation matrix
k=pf_stiff(E,A,I,L);// Local stiffness matrix
u=zeros(6,1);// Initialize member end displacements
dof=pf_getdof(imem,conn,lm);// Get DOF numbers for the ends of member
fori=1:6
idof=dof(i);
ifidof~=0then
u(i)=x(idof);// Copy global displacement into u
end
end
uu=r*u;// Displacements in local axes
f=zeros(6,1);
f=k*uu;// Member end forces in local axes
endfunction
[nmemdummy]=size(conn);
F=zeros(nmem,6);memendforces=zeros(nmem,7);
fori=1:nmem
f=pf_memendforces(i,xz,conn,mprop,lm,x);
F(i,:)=f';
memendforces(i,:)=[i,F(i,:)];
end
disp('The member end forces of the system in local axes');
H=[' Member','  P1',' V1','M1','P2','V2','M2'];
printf('%s\t%s\t\t%s\t\t%s\t\t%s\t\t%s\t\t%s\n',H);
disp(memendforces);
function[Ar]=pf_supportrxn(bc, xz, conn, memendforces)
b=size(bc,'r');
Ar=zeros(b,3);Am=zeros(size(conn,'r'),6);
```

```
fori=1:size(conn,'r')
[Lcxcz]=pf_calclen(i,xz,conn);
r=pf_calcrot(cx,cz);
Am(i,:)=(r'*memendforces(i,2:7)')';
nn=conn(i,1);
nm=conn(i,2);
forj=1:b
ifbc(j,1)==nnthen
Ar(j,1:3)=Ar(j,1:3)+Am(i,1:3);
end
ifbc(j,1)==nmthen
Ar(j,1:3)=Ar(j,1:3)+Am(i,4:6);
end
end
end
endfunction
[Ar]=pf_supportrxn(bc,xz,conn,memendforces);
ar=zeros(size(bc,'r'),4);
forj=1:size(bc,'r')
ar(j,1:4)=[bc(j,1),Ar(j,:)];
end
disp('Support reactions in global axes')
B=['  Node','  Rx',' Rz','My'];
printf('%s\t%s\t\t%s\t\t%s\n',B);
disp(ar);
```

***** ============THE END============ *****