ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGYSCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

Modeling of Three Dimensional Framed Structures Under Seismic Loading To Investigate Behavior With And Without The Underlying Soil Using Opensees.

By

Belay worku

A Thesis in structural Engineering Stream

April,2016

Addis Ababa

A Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in civil Engineering

The undersigned have examined the thesis entitled '**Modeling of Three Dimensional Framed Structures Under Seismic Loading To Investigate Behavior With And Without The Underlying Soil Using Opensees**' presented by **Belay Worku**, a candidate for the degree of **Master of Science** and hereby certify that it is worthy of acceptance.

Dr.Ing.Bedilu Habte		
Advisor	Signature	Date
Dr.Ing		
Internal Examiner	Signature	Date
Dr.Ing		
External Examiner	Signature	Date
Chair person	Signature	Date

UNDERTAKING

I certify that research work titled "Modeling of Three Dimensional Framed Structures under Seismic Loading to Investigate Behavior with and without the Underlying Soil Using Opensees" is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged / referred.

____

Belay Worku

Abstract

A reinforced concrete building symmetric in plan, having a height of 30 m, located in seismic zone II (a_g=0.05g), has been analyzed and designed using Etabs software and according to EBCS provision. After the design is completed, the reinforced concrete frame is modeled and analyzed, employing geometric nonlinear Finite Element Modeling using Opensees software under two different boundary conditions: (I) fixed-base (no Soil-Structure Interaction), and (ii) considering Soil-Structure Interaction (SSI).The nonlinear behavior of a soil-foundation system alter the seismic response of a structure by providing additional flexibility to the system and dissipating hysteretic energy at the soil-foundation interface. However, the current design practice is still reluctant to consider the nonlinearity of the soil-foundation system, primarily due to lack of reliable modeling techniques. This study is motivated towards evaluating the effect of nonlinear soil-structure interaction (SSI) and frame structure without soil under seismic responses of low-rise reinforced conctremoment resisting frame (RCMRF) structures. In order to achieve SSI, a Winklerbasedapproach is adopted, where the soil beneath the foundation is assumed to be a system of closely-spaced, independent, nonlinear spring elements. Static pushover analysis is performed on a 10-story RCMRF building and the performance of the structure is evaluated through a variety of force and displacement demand parameters. It is observed that incorporation of nonlinear SSI leads to an increase in story displacement demand and a significant reduction in base moment, base shear and inter-story drift demands, indicating the importance of its consideration towards achieving an economic, yet safe seismic design.RCMRF was also modeled using sap 2000 with and without considering soil effect. The pattern of the outputs seems the same. But, the outputs of sap are relatively smaller in value than the opensees outputs.

Acknowledgement

First I would like to thank the Almighty God for everything in my daily activities. Next, I am very joyful to express my deepest respect and thanks to my advisor Dr.Ing.Bedilu Habte for his unreserved assistance, constructive and timely comments at all stages of my work.The title of this thesis work was Dr.Ing.Bedilu’s idea.Next; I would also thank my famlies and friends.

Belay Worku

Table of Contents

Lists of Figures

List of tables

Lists of Abbreviation

#	Comment
3D	three dimension
B	beam width
barAreaSec	Area of longitudinal-reinforcement bars(diam.=20mm)
beamSec	beam section
Bs	strain-hardening ratio
Bsec	Column width
ColSec	column section
coreY	The distance from the section z-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete
coreZ	The distance from the section y-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete
CoverSec	Column covers to reinforcing steel NA.
coverY	The distance from the section z-axis to the edge of the cover concrete -- outer edge of cover concrete
coverZ	The distance from the section y-axis to the edge of the cover concrete -- outer edge of cover concrete
Crad	Radation damping
CR1	control the transition from elastic to plastic branches
CR2	control the transition from elastic to plastic branches
D	beam depth
Disp	ıcement
dispBeamColumn	distributed-plasticity, displacement –based beam column element
du1	first displacement increment (pseudo-time step) in the next invocation of the analysis command
Ec	Concrete Elastic Modulus
Ele	Element
EBCS	Ethiopian building code
eleTag	element tag

EnergyIncr:	energy increment test
eps2C: concrete	strain at ultimate stress for confined
eps2U	stress for unconfined concrete
Es	modulus of steel
Ets	tension softening stiffness
Fc	Concrete compressive+e strength, (+Tension,-Compression)
fc1C	Confined concrete (mander model), maximum stress
fc1U	Unconfined concrete (todeschini parabolic model), maximum stress
fc2C	ultimate stress for confined concrete
fc2U	ultimate stress for unconfined concrete
FE	finite element
ftC	tensile strength +tension for confined concrete
ftU	tensile strength +tension for unconfined concrete
Fy	Steel yield stress
GC	shear modulus
GJ	torsional stiffness
Hsec	column depth
IDbeamTransf	Identity of beam transformation
IDcolTransf	Identity of column transformation
IDconcCore	Material Identity number of confined core concrete
IDconcCover	Material Identity number of confined core concrete
IDreinf	material Identity number of reinforcement
Kfc	ratio of confined to unconfined concrete strength
KN	Kilo newoton
Lambda	ratio between unloading slope at $eps2 and initial slope $Ec
M	Meter
Ndf	Number of degree of freedom
Ndm	Dimension of problem (1, 2, 3)
nfCoreY	number of fibers for concrete in y-direction -- core concrete
nfCoreZ	number of fibers for concrete in z-direction
nfCoverY	number of fibers for concrete in y-direction -- cover concrete

nfCoverZ	number of fibers for concrete in z-direction
Np	number of integration points
numBarsSec	number of longitudinal-reinforcement bars in steel layer. (Symmetric top & bot)
numIntgrPts	number of integration point
PDelta	p-delta
PerpDirn:	perpendicular direction
Quadr	Quadrilateral
RC	Reinforced concrete
RCM	Reverse cuthill-Mckee
Ro	control the transition from elastic to plastic branches
Sec	Second
Sec	Section
SecTag	tag for symmetric section
SecTag3D	ID tag for combined behavior for 3D model
SecTagTorsion	ID tag for torsional section behavior
T	Torsion
Tp	The maximum maginitude of drag force in component 1 of element 1 of the nonlinear spring in fraction
Z50	The displacement at which 50% of Qult is mobilized during monotonic loading

1. Introduction

Structural design in earthquake prone regions is heavily dependent on accurate and reliable analysis procedures prior to construction. The ground accelerations experienced can create large lateral forces on the structures, which often will produce inelastic material behavior. Before the arrival of the finite element method these analyses were done by simple static methods, commonly referred to as equivalent force method, similar to what is provided in building codes as an alternative method of analysis. This method is however, not sufficient for analysis of the stronger earthquakes, as change of structural properties due to inelasticity will affect the overall response of the structure. Because of this fact it is a rather cumbersome task to predict the behavior of a building, especially since there is no known way to the magnitude of the next earthquake .The last decade's improvement of computational power and technology has enabled more sophisticated analysis methods. To obtain more realistic approximations on structural response, nonlinear analysis is used in modeling of structures prone to seismic loading. In this work geometric nonlinear modeling using finite element method under seismic load will be conducted. Comparison is also made on the effect of the underlying soil on the analysis result with frame structure without soil.

1.1. Non-Linear Computational Mechanics

Two sources of nonlinearity exist in the analysis of solid continua, namely, material (occur in model when applied load causes large displacement and / or rotation, large strain, or a combination of both) and geometric (occur when material stress-strain relationship depend on load history (plasticity problems), load duration (creep problems), temperature (thermo plasticity), or combination of all) nonlinearity. The former occurs when, for whatever reason, the stress strain behavior given by the constitutive relation is nonlinear, whereas the latter is important when changes in geometry, however large or small, have a significant effect on the load deformation behavior. Material nonlinearity can be considered to encompass contact friction, whereas geometric nonlinearity includes deformation-dependent boundary conditions and loading. The finite element method is a procedure whereby the continuum behavior described at infinity of points is approximated in terms of a finite number of points, called nodes, located at specific points in the continuum. These nodes are used to define regions, called finite elements, over which both the geometry and the primary variables in the governing equations are approximated.

In general, nonlinear analysis is harder, requires much more thought when setting up the model, and requires more thought when setting up the analysis, takes more computational time, and does not always converge to the correct solution, but most problems require nonlinear analysis. In this work geometric non-linear analysis is considered.

1.1.2. Importance of Finite Element Modeling

The finite element method is a procedure whereby the continuum behavior described at infinity of points is approximated in terms of a finite number of points, called nodes, located at specific points in the continuum. These nodes are used to define regions, called finite elements, over which both the geometry and the primary variables in the governing equations are approximated. To model the complex behavior of reinforced concrete analytically in its non-linear zone is difficult. This has led engineers in the past to rely heavily on empirical formulas which were derived from numerous experiments for the design of reinforced concrete structures.

The Finite Element method makes it possible to take into account non-linear response. The Finite element method is an analytical tool which is able to model Reinforced concrete or retrofitted structure and is able to calculate the non-linear behavior of the structural members is Finite element method. For structural design and assessment of reinforced concrete members, the non-linear finite element (FE) analysis has become an important tool. The method can be used to study the behavior of reinforced and pre-stressed concrete structures including both force and stress redistribution. With the advent of digital computers and powerful methods of analysis, such as the finite element method many efforts to develop analytical solutions which would obviate the need for experiments have been undertaken by investigators. The finite element method has thus become a powerful computational tool, which allows complex analyses of the nonlinear response of RC structures to be carried out in a routine fashion. In this work opensees finite element software is used to model and analyze RC structure with and without considering underlying soil effect.

1.2. Statement of the problem

The dynamic response of structure depends upon nature of soil located under foundation, so neglecting soil-structure interaction in the non linear analysisis unsafe. During an earthquake, the load and deformation characteristic of the structural and geotechnical (soil) components of the foundations of structures can effect, and in some cases dominate, seismic response and overall performance. The modeling of soil and structural parts of foundations inherently accounts the interaction of soil and structure. In soil structure interaction the appropriate modeling of the flux of energy from the soil to the structure, and then back from the structure to the soil is accounted for. In this work the effects of the soil on nonlinear modeling of RC frame structure will be investigated and compared without considering soil effect.

1.3. Objectives

1.3.1. General objectives

The general objective of this work is to investigate the behavior of Reinforced concrete building situated on clay soil of seismic zone II (ground acceleration of 0.05g) with and without underlying soil by non-linear analysis.

1.3.2. Specific objective

A ten story 3D reinforced concrete structure under seismic load is modeled and analyzed using opensees finite element software.

The following are the main objectives of the present study:

- To model the reinforced concrete frame with and without considering underlying soil using finite element software (opensees).
- To study the response of reinforced concrete frame using non-linear finite element analysis under seismic loading.
- To compare the output of the two boundary conditions of reinforced concrete frame with and without soil effect.

1.4. Methodology of the Study

- Review of existing literatures by different researchers on related titles.
- Selection of the types of Reinforced concrete building used for analysis.
- A ten story 2 bay in the x-direction and 1 bay in the y-direction moment resisting reinforced concrete building frame, representing the conventional type of regular low-rise building frames, resting on shallow foundation, is selected in conjunction with a clay soil with presumptive value of 280Kpa, as classified in the EBCS 7, 1995 and analyzed using

appropriate software (Etabs).Then, using EBCS recommendation the framed is designed and detailed.

- After design is completed ,then the frame is modeled and analyzed, employing nonlinear Finite Element Modeling using Opensees software under two different boundary conditions: (I) fixed-base (no Soil-Structure Interaction), and (ii) considering Soil-Structure Interaction (SSI).Then output for the two boundary conditions are compared.

1.5. Application of the Research

- The developed modeling using opensees finite element software will be applicable for analysis of three dimensional reinforced concrete frames for static loads only.
- The object oriented programme is flexible to modify dimension, material strength, geometry, loading, analysis types, etc.
- Students may use this for future research in line with upgrading the object oriented programe to handle many other types of structural analysis for steel, composite frames, etc.

1.6 .Content of the thesis

This thesis consists of five chapters. Chapter one introduces some ideas on the title and the main task of this work. Chapter two deals with literature review i.e. related research done by different researchers work and the theoretical background of the modeling software (opensees).The third chapter deals with the analysis and design of the selected frame structure and the details of the structure modeling in opensees, material modeling, and analytical programming procedure steps involved in modeling of the frame structure. In the fourth chapter, the result of non-linear analysis of frame structure with and without soil is discussed in details. In chapter five, conclusions and recommendations of the present study are given followed by the references.

2. Literature review

2.1. Seismic Behavior of Frame Structure

Not every structural component in a building is designed to resist seismic loads. Rather, some building elements are designed only for gravity or vertical loads.

Framing systems are composed of horizontal girder elements, vertical columns and joint connections that can transmit lateral loads in addition to gravity loads.

There are three response characteristics used as the most important parameters that describe the behaviour of structures and their foundations when subjected to earthquakes. These are stiffness, strength (or capacity) and ductility. Prior to defi ning the three quantities, it is instructive to reiterate the defi nition of two more fundamental quantities, namely ‘action’ and ‘deformation’. The former is used in to indicate stress resultants of all types, while the latter is used to indicate strain resultants.

Stiffness is the ability of a component or an assembly of components to resist deformations when subjected to actions. It is expressed as the ratio between action and deformation at a given level of either of the two quantities and the corresponding value of the other. Therefore, stiffness is not a constant value. K_i is the stiffness at a required deformation δ_i and corresponds to force resistance V_i. If increments or first derivatives of actions and deformations are used, the ensuing stiffness is the tangent value. If total actions and deformations are used, the ensuing stiffness is the secant value.

Strength is the capacity of a component or an assembly of components for load resistance at a given response station. It is also not a constant value. The term ‘strength’ is to represent both action resistance and the ability to endure deformation, or deformation capacity.

Ductility is the ability of a component or an assembly of components to deform beyond the elastic limit, and is expressed as the ratio between a maximum value of a deformation quantity and the same quantity at the yield limit state. The displacement ductility μ is the ratio between the maximum or ultimate displacement δ_u and the yield displacement δ_y .

Demand is the action or deformation imposed on a component or an assembly of components when subjected to earthquake ground motion. This demand is not constant. It continuously varies as the structural characteristics vary during inelastic response. It also varies with the characteristics of the input motion.

Supply is the action or deformation capacity of a component or an assembly of components when subjected to earthquake ground motion. Therefore, the supply represents the response of the structure to the demand. It may continuously vary as the structural characteristics change during inelastic response. It also varies with the characteristics of the input motion

2.2. Earthquake Loading

The response of a building during an earthquake can be classified as a very dynamic event. Ground accelerations at the base of the structure cause the building to sway back and forth like an inverted pendulum. The movement of the ground and the inertia of the structure cause shear forces to develop at the structures base. The shear forces and displacements caused by this inertial movement in turn cause axial and rotational forces to develop within the structural elements of the building. If a structure is designed to be ductile, some energy caused by seismic action will be absorbed by inelastic behavior in structural components. In order to design structures to perform in this manner during a seismic event, engineers must be able to predict the seismic forces associated with a buildings dynamic response for preliminary design.However, in this work the earthquake load is calculated using equivalent lateral force method using EBCS 8 1995 provision and these loads are applied to each node to assumed direction for frame structure modeling in opensees.

2.2.1. Equivalent Lateral Force Method

The equivalent lateral force method (ELFM) centers around the calculation of the base shear force caused by the buildings inertial response to seismic action at the foundation level. As the ground moves in one direction, the inertia of the buildings floors resists the motion, which in turn, causes lateral displacements at each story level and a horizontal reaction or shear force at the base supports of the structure. As the floor level displaces, the connecting columns and ultimately the supports below the story try to overcome the floors inertial resistance to the ground motion, which causes internal member forces. The ELFM idealizes this inertial resistance at each story level by applying an equivalent lateral seismic force as shown in Figure 1 to move each floor laterally from the top down, rather than moving the ground laterally from the bottom. The ELFM ultimately captures the first modal shape of the building without having to conduct a modal analysis and allows a static analysis approach to be used for the determination of internal forces, shears, moments, and displacements for design.

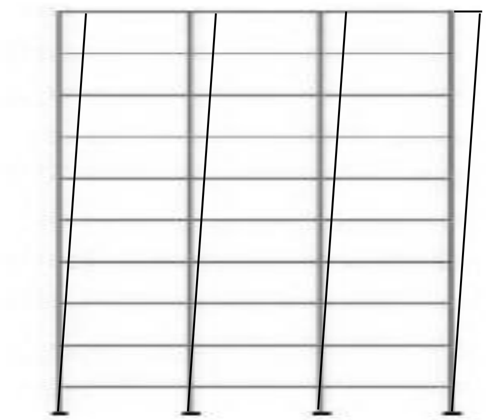


Figure. 1.Movement Of Floor Due To Lateral Loads

The provisions of EBCS 8, 1995 calculate the base shear force for each main direction as the multiplication of the building weight with a seismic coefficient as shown:

$$F_{b=S_{d(T_1)}W} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{2.2.1.1}$$

$$T_1 = C_1 H^{3/4} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{2.2.1.2}$$

C_1= 0.075 for reinforced concrete moment-resisting frames and eccentrically braced steel frames.
Sd (T1), ordinate of the design spectrum at period T1

H is height of the building above the base in meter.
T1 is fundamental period of vibration of the structure for translation motion in the direction considered. 'W' is seismic dead load computed as follows. The effects of the seismic action shall be evaluated using a seismic dead load, W, obtained as the total permanent load plus 25% of the floor variable (live) load, for storage and warehouse occupancies. In other occupancies, no allowance for live loads need be made.

The building weight W meanwhile is used to accounts for the inertia of the building. It refers to the weight of the structure that would be anticipated during a seismic event. This would include the dead weight of the structure, the weight of all floor partitions, and the weight of all tanks and permanent equipment in the building (MacGregor et al 2005, 1000). Additionally, a minimum of 25 percent of the buildings live load must also be applied to account for possible occupants at the time of the event. The seismic coefficient, s, accounts for the soil and site conditions, the design ground acceleration, and the fundamental period and ductility of the building. It seeks to characterize how the weight of the building will respond to a seismic event. The EBCS 8, 1995 specification describes the factor as follows:

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

$$s_d(T) = \alpha\beta\gamma \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 2.2.1.3$$

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

$$\alpha = \alpha_o I \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 2.2.1.4$$

Where α = the bedrock acceleration ratio for the site and depends on the seismic zone as given in the fig.1.

Table 1.Bedrock Acceleration Ratio αo

Zone	4	3	2	1
α_o	0.1	0.07	0.05	0.03

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

$$\beta = \frac{1.2s}{T^{2/3}}, \leq 2.5 \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 2.2.1.5$$

The parameter β) is the site coefficient for soil characteristics given in Table 2.

Table 2.site coefficient S

Subsoil class	A	B	C
S	1	1.2	1.5

2.2.2. Importance Categories and Importance Factors

(1) Buildings are 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 categories are characterized by different importance factors I as described in Clause 1.2.1. in EBCS 8,1995.

(3) The importance factor I = 1.0 is associated with a design seismic event having a reference return period as indicated in Clause 1.4.1(3) in EBCS 8,1995.

(4) The definitions of the importance categories and the related importance factors are given in Table 3.

Table 3.Importance Categories and Importance Factors for Buildings

Importance category	Buildings	Importance factor I
I	Building whose integrity during earthquake is 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, institution etc	1.2
III	Ordinary buildings ,not belonging to the other categories	1
IV	Buildings of minor importance for public safety, e.g. agricultural buildings, etc.	0.8

2.2.3. Distribution of the Horizontal Seismic Forces

The lateral seismic force at each story is calculated as a proportion of the base shear with respect to the weight and height of the floor as defined in the following equation (EBCS 8 1995, 2.3.3.2.3):

(1) The base shear force shall be distributed over the height of the structure in conformance with Equation) in the absence of a more rigorous procedure.

$$F_b = F_t + \sum_{j=1}^{n} F_i \quad \ldots\ldots 2.2.3.1$$

(2) The concentrated force F, at the top, which is in addition to F_n shall be determined from the equation:

$$F_t = 0.07T_1F_b \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.2.2.3.2$$

(3) The remaining portion of the base shear shall be distributed over the height of the structure, including level n according to the following formula:

$$F_i = \frac{(F_b - F_t)W_ih_i}{\sum_{j=1}^{n} W_jh_j} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.2.2.3.3$$

(4) At each level designated as i, the force F; .shall be applied over the area of the building in accordance with the mass distribution at that level. Stresses in each structural element shall be calculated as the effect of force F; and F, combined applied at the appropriate levels above the base.

(5) The horizontal forces F, determined in the above manner shall be distributed to the lateral load resisting system assuming rigid floors.

Based on this equation and the fact that the ELFM is capturing the first modal response of the structure, the top story of the building will most likely have the largest seismic loading because this story will experience the most lateral movement during an event. As a check, the lateral story forces should sum to the value of the base shear, V. These story forces can now be used to calculate forces and deflections within the lateral load resisting system. In addition to the lateral forces acting on a structure, the deflection and stability of the structure must also be calculated for a seismic event. Specifically, EBCS 8 1995 outlines permissible values for the design story drift and the stability factor. The design story drift (Δ) is determined as the difference in lateral deflection (δx) between the top and bottom of a specific story as shown in Figure 2.

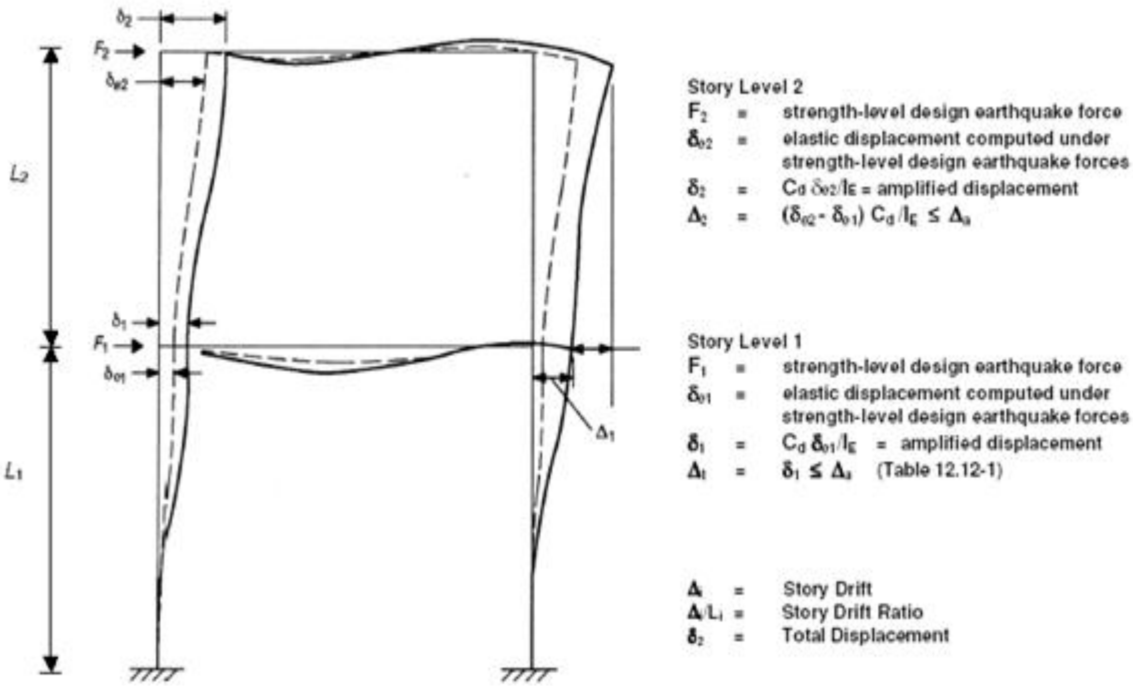


Figure. 2.Deflection of Frame structure due to lateral loads

2.2.4. Displacement Analysis

(1) The displacements induced by the design seismic action shall be calculated on the basis of the elastic deformation of the structural system by means of the following simplified expression:

The allowable deflection is determined as:

$$d_s = \frac{d_e}{\gamma_d}$$

Where d_s is the displacement of a point of the structural system induced by the design seismic action.

γ_d is displacement behavior factor, assumed equal to γ unless otherwise specified.

d_e is displacement of the same point of the structural system, as determined by a linear analysis based on the design response spectrum according to 1.4.2.2(4) EBCS 8 1995.

2.2.5. Limitation of Inter-storey Drift

(1) Unless otherwise specified in Chapters 3 to 6 EBCS 8 1995, the following limits on design inter storey drift shall be observed:

(a) For buildings having non-structural elements of brittle materials attached to the structure;

$d_r \leq 0.01h$

(b) For buildings having non-structural elements fixed in a way as not to interfere with structural deformations:

$d_r \leq 0.015h$

Where d_r design inter-storey drift as defined in 2.4.2.2(2) in EBCS 8 1995.

H is storey height

A more ductile structure will therefore be allowed a larger design deflection. The deflection (δxe) corresponds to the lateral deflection calculated by an elastic analysis, while the factor (I) refers to the importance factor of the building. A more important structure will therefore have a lower allowable deflection. The stability of the structure is represented by a stability factor θ, which considers possible P-Δ effects on the shears and moments in the structure. P-Δ effects occur from the horizontal displacement of vertical loads in the structure. This eccentricity must be accounted for in the shears and moments of the structure and therefore requires a second-order analysis. However, these effects can be ignored if the stability factor outlined in EBCS 8 1995 is less than 0.10 (EBCS 8 1995, 30). The factor θ for a given level x is defined by the following formula:

$$\theta = \frac{P_{tot}}{V_{tot}h} d_r \leq 0.10 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{2.2.5.1}$$

The factor (Ptot) corresponds to the vertical loading above the specified level x while (dr) is the story drift at the level. (Vtot) is the shear value acting between the story and the story below it, while (h) is the story height in meters. The factor of θ must also not be greater than 0.25.

If the value of θ is greater than 0.10 but less than 0.25, displacement and forces within the structure are to be multiplied by a factor of a (EBCS 8 1995, 30). However, if θ is greater than the maximum, the structure "is potentially unstable and must be redesigned" (EBCS 8 1995).

Where, $a = {}^{1}/_{(1-\theta)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{2.2.5.2}$

2.3. Frame Structural Analysis Procedures

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

The use of seismic analysis both in research and practice has increased substantially in recent years due 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.

2.3.1. Modeling of Frame 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 distributed all over the length of the member. 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.

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 12components: 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.

2.3.2. Modeling Considerations in Frames

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 the members (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 for which static behavior is important. This is required because the mass is always distributed at the member's overall length.

2.4. Loads and Boundary Conditions in Frames

Loads are specified forces applied to members (elements) or nodes. Gravity loads may be applied to elements' nodes or considered as nodal loads depending on the gravity load path. The vector of nodal loads for a structure is denoted by **P,** with six components of force at each node for 3D problems. In contrast with nodal loads, element loads 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 the entire model in which each component is a degree of freedom. The set of all global degree of freedoms (DOFs) are separated into two subsets: the DOFs with unknown displacement values and the DOFs with specified displacement value. Each DOF in the model must be included in one of the two sets. The unknown displacements are called the free DOFs and are denoted by $\mathbf{u_f}$. The second sets of displacements correspond to the restrained DOFs; and are denoted by **ud**. The restrained DOFs are generally assigned a value of zero to indicate a fixed displacement.

The selection of restrained displacements at the supports is an important step in the structural modeling and the supports of a model are commonly identified with the symbols Shown in 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**. The nodal forces at the free DOFs of the model are specified as nodal loads, and are denoted by **P**f. The forces at the restrained DOFs are the support reactions and are denoted by **P**d. These can be evaluated once the equations for the free DOFs are solved.

2.5. Equilibrium and Compatibility in Frame Structures

Equilibrium equations set the externally applied loads equal to the sum of the internal element forces at all joints or node points of a structural system; they are the most fundamental equations in structural analysis and design. The exact solution for a problem in mechanics requires that the equations of 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 those basic equilibrium equations (Wilson, 2002).

In the analysis of a structural system of discrete elements, all elements connected to a joint or node point must have the same absolute displacement. If the node displacements are given, all element deformations can be calculated from the basic 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 based solution. For real nonlinear structures, equilibrium is always satisfied in the deformed position. Many real structures do not satisfy compatibility caused by creep, joint slippage, incremental construction and directional yielding (Wilson, 2002).

2.6. Methods of Analysis

2.6.1. Nonlinear Static Analysis

In the nonlinear static procedure, the structural model is subjected to an incremental lateral load whose distribution represents the inertia forces expected during ground shaking. The lateral load is applied until the imposed displacements reach the so-called “target displacement,” which represents the displacement demand that the earthquake ground motions would impose on the structure. Once loaded to the target displacement, the demand parameters for the structural components are compared with the respective acceptance criteria for the desired performance state. System level demand parameters, such as story drifts and base shears, may also be checked. The nonlinear static procedure is applicable to low-rise regular buildings, where the response is dominated by the fundamental sway mode of vibration. It is less suitable for taller, slender, or irregular buildings, where multiple vibration modes affect the behavior for further discussion on the applicability of nonlinear static analysis.

2.6.2 .Nonlinear Static Versus Nonlinear Dynamic Analysis

Nonlinear dynamic analysis methods generally provide more realistic models of structural response to strong ground shaking and, thereby, provide more reliable assessment of earthquake performance than nonlinear static analysis. Nonlinear static analysis is limited in its ability to capture transient dynamic behavior with cyclic loading and degradation. Nevertheless, the nonlinear static procedure provides a convenient and fairly reliable method for structures whose dynamic response is governed by first-mode sway motions. One way to check this is by comparing the deformed geometry from a pushover analysis to the elastic first-mode vibration shape. In general, the nonlinear static procedure works well for low-rise buildings (less than about five stories) with symmetrical regular configurations. FEMA 440, FEMA 440A, and NIST (2010) provide further details on the simplifying assumptions and limitations on nonlinear static analysis. However, even when the nonlinear static procedure is not appropriate for a complete performance evaluation, nonlinear static analysis can be an effective design tool to investigate aspects of the analysis model and the nonlinear response that are difficult to do by nonlinear dynamic analysis. For example, nonlinear static analysis can be useful to (1) check and debug the nonlinear analysis model, (2) augment understanding of the yielding mechanisms and deformation demands, and (3) investigate alternative design parameters and how variations in the component properties may affect response.

2.7. Foundations and Soil-Structure Interaction

For many projects the effect of site soils is dealt with independently of the structure. The structural designer inputs “free field” response spectra or ground motions directly to an analysis of the structure without any consideration of interaction and then designs the foundations for the resulting forces. However, in some cases this approach can give unrealistic results, leading to foundations and superstructure designs that may be overly conservative or unconservative.

In most cases ignoring soil-structure interaction is conservative, provided the design response spectra and ground motions adequately envelope the kinematic effect of the foundation structure and its effect on site response. This may be difficult to do in cases such as the following, where soil-structure interaction analysis is advisable to reduce risk:

The foundation system alters the soil properties (e.g., a pile foundation in soft soils).

Buildings with a deep basement or pile foundation system, where it is difficult to determine the effective ground excitation and where the structural inertia forces are dependent on the foundation reaction with the soil. This issue is compounded for sites where the soil properties vary significantly with depth where the site conditions are susceptible to large ground deformations, e.g., lateral spreading or ground fault rupture, or soil liquefaction.

Soil-structure interaction analysis is also undertaken to realize substantial construction cost savings by reducing the conservatisms in the conventional approach. This is typically worthwhile on sites with relatively soft soils where:

The flexibility of the soil-foundation system significantly elongates the effective natural periods of the structure and increases the damping, leading to reduced earthquake design forces.Where the structure is massive and its inertia forces significantly increase the strain levels in the soil relative to the free field response.Incorporation of foundation effects into structural-response prediction requires some basic understanding of soils and soil-structure interaction. The behavior of soils is significantly nonlinear under strong ground shaking, and soil materials display strain softening, energy dissipation through material hysteresis and radiation damping, and strain rate dependency. Soils generally have no distinct yield point and exhibit gradual reduction of stiffness with increasing strain. Certain types of soil (typically saturated sands and silts) can develop excess pore water pressure during earthquake shaking, resulting in reduced effective stress levels, softening, weakening and in the extreme, complete loss of strength (liquefaction). While pore pressure generation and liquefaction models exist, treatment of liquefiable soils in structural analysis is a significant challenge.Even if a detailed geotechnical investigation is available, a high degree of uncertainty in behavior of the soils will remain. For this reason, it is recommended that analyses are undertaken using upper and lower bounds of soil properties. The upper bound soil stiffness and strength is usually more critical for the demands on the structure itself, and lower bound properties may be critical for the design of the foundation.

2.7.1. Overview of Soil-Structure Analysis Issues

Modeling of soil-structure interaction is dominated by the issues associated with the soil being an infinite medium, making it difficult to model the transmission of earthquake-induced stress and strain waves through the boundaries of the soil model. Whereas various forms of transmitting boundaries have been devised for linear frequency domain analysis, no exact boundary formulations exist for nonlinear (time domain) dynamic analysis. There are two generic approaches to practical nonlinear soil-structure interaction analysis:

The direct approach, in which a volume of soil is modeled explicitly with the structure and a “total” solution, is obtained in a single analysis.

The indirect (substructure) approach in which the analysis is performed in two stages: (1) The effective input motions seen by the structure are derived by consideration of the incoming seismic waves and the geometry of the foundation (kinematic interaction); and (2) The dynamic response of the structure is calculated by applying the motions to the structural model via a simplified representation of the foundation (inertial interaction).

2.7.1.1. Direct Approach in the Time Domain

The direct approach is described first since it is more intuitive, although the computational modeling requirements are more advanced. In this approach the soil is discretized using solid (brick) nonlinear finite elements (or finite difference elements) and the structure and the structural foundation system, which may be flexible and nonlinear, are modeled explicitly. The ground motions (including spatial variability, when significant) are applied at special boundaries at the base and sides of the model, and the kinematic interaction is modeled directly. The medium beneath the lower boundary of the model is modeled as grounded linear viscous dampers, to which the desired ‘bedrock’ ground motions are induced via applied force history. This is done to prevent spurious stress wave reflection at the lower boundary.

The horizontal dimension of the soil block should be sufficiently large such that the motion at the nodes of the lateral boundaries can be considered as that of the free-field. The free-field ground motion histories are applied to the lateral boundaries. It is advisable to test the sensitivity of the model to the finite element mesh density, soil properties, and proximity of boundaries to the structure.

2.7.1.2. Indirect Approach

In the indirect approach, the dynamic compliances of the soil domain are represented as spring-damper pairs for each degree of freedom of the foundation being considered.Note that the number of spring-damper pairs may be larger than depending on (1) whether the foundation is modeled as rigid or flexible, and (2) how rocking is represented, either as a rotational spring or multiple axial springs.The properties of the spring damper pairs may be based on (1) analytical and numerical solutions for rigid bodies sitting on or embedded in an elastic halfspace (Wolf 1985; Werkle and Waas 1986), or (2) alternatively by numerical frequency domain analysis. If the soil resistance is represented by springs having gradual strain softening, the initial stiffness characteristics may be based upon small strain soil moduli. If a simpler (e.g., bilinear) spring is used, then a representative elastic stiffness, based on expected strain level, should be used. In either case a strength cap equal to the expected ultimate resistance should be introduced. Since forces in the soil dampers may be large, the strength limit (say in foundation sliding) should be applied to the combined spring-damper force.

2.8. Opensees Simulation

Opensees (Open system for earthquake engineering simulation) is object-oriented open source software which allows users to implement finite element methods to model the structural and geotechnical systems and simulate the response under earthquake loading. It has been under development by the Pacific Earthquake Engineering Research Center since 1997. Because Opensees is object-oriented framework software, in a finite element application, mainly four types of objects, model builder object, domain object, recorder object and analysis object need to be constructed. In Opensees, the interpretation is accomplished by adding commands into Tcl script for finite element analysis. Each command is associated with a C++ procedure that is built inside and is called by the interpreter to analyze the command (Mazzoni et. al 2006).

Why is Opensees chosen in this study? Several reasons are given as follows. To begin with, it is an open source which is free to be used. Second, both linear and nonlinear structural and geotechnical models can be built in Opensees. Third, various simulations: static push-over analysis, static reversed-cyclic analysis, dynamic time-series analysis, uniform-support excitation, multi-support excitation can be effectively conducted. Last but not the least, Opensees provides a library of various materials, elements and analysis which is powerful for numerical simulation of nonlinear systems.

However, since Opensees is research-developed software which denotes that it is not maturely developed, the simulations conducted using Opensees need to be testified with results obtained from other software. In this work it is compared with sap.

2.10.1. Modeling Limitations

- 3D elements are modeled with 1-D line elements with a 2D cross section and it is assumed that plane sections remain plane.
- No dependable predictive model on flexure & shear interaction due to the reason that it is not yet proved.
- For many existing structures it is unknown exactly what is within the cover
- Quality-control in construction is good enough
- Material properties over time
- Concrete strength with curing
- Creep and Fatigue
- Corrosion in reinforcement

2.10.2. Considerations in Modeling RC Frames in Opensees

- **Material**s (Confined Concrete, Unconfined Concrete, Reinforcing Steel)
- **Sections (**Elastic Section, Uniaxial Section –uncoupled axial & flexure, Fiber Section – coupled P-M-M Interaction)
- **System (**2D/3D,Rigid/Flexible Diaphragm)
- **Elements (**Structural Element, Beams –no axial load, Columns (P-M Interaction)
- **Plastic-Hinge behavior (**Confinement, Hinge length & growth, Yield Penetration, Bond Stress/Strength, Bar Pull-out, Anchorage loss, Bar elongation and buckling)
- **Element Type (**Continuum model, Distributed plasticity, Lumped plasticity, Displacement-based)
- **Beam-Column Connections (**Shear, Moment-Shear Interaction, Shear-Critical Elements)

2.10.3. Concrete Materials in Opensees

There are two types of materials currently available in Opensees, uniaxial materials and nDmaterials. The different types of concrete and steel materials are among the uniaxial materials. There are three types of concrete available:

i. Concrete01: uniaxial Kent-Scott-Park concrete materials object with degraded linear unloading/reloading stiffness according to the work of Karsan-Jirsa and no tensile strength
ii. Concrete02: uniaxial concrete material object with tensile strength and linear tension softening
iii. Concrete03: Uniaxial concrete materials object with tensile strength and nonlinear tension softening.

In this work Concrete02 will be used for the structure under consideration, as the tensile strength of the concrete is of interest in the elastic range, and modeling nonlinear tension softening is considered. The cover and core concrete will be modeled as different materials, using the same material type, but different stress and strain characteristics and different material tags.

2.10.4. Material Model for reinforcing steel bar

Capturing the structural response and associated damage require accurate modeling of localized inelastic deformations occurring at the member end regions as identified by shaded areas in Figure below. These member end deformations consist of two components: 1) the flexural deformation that causes inelastic strains in the longitudinal bars and concrete, and 2) the member end rotation, as indicated by arrows in Figure below, due to reinforcement slip. The slip considered here is the result of strain penetration along a portion of the fully anchored bars into the adjoining concrete members (e.g., footings and joints) during the elastic and inelastic response of a structure.

Ignoring the strain penetration component may appear to produce satisfactory force-displacement response of the structural system by compromising strain penetration effects with greater contribution of the flexural action at a given lateral load. However, this approach will appreciably overestimate the strains and section curvatures in the critical inelastic regions of the member, and thereby overestimate the structural damage.

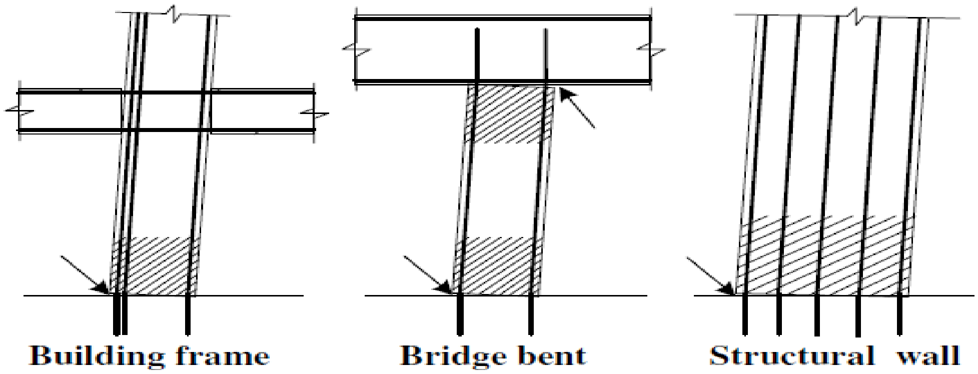


Figure. 3.Expected inelastic regions at the column and wall ends

The zero-length section element available in Opensees may be used to accurately model the strain penetration effects (or the fixed end rotations shown in Figure 4). Zero-length section elements have been generally used for section analyses to calculate the moment corresponding to a given curvature. To model the fixed-end rotation, the zero-length section element should be placed at the intersection between the flexural member and an adjoining member representing a footing or joint as shown in Figure 5. A duplicate node is also required between a fiber-based beam-column element and the adjoining concrete element as shown in Figure 5. The translational degree-of-freedom of this new node (i.e., node j in Figure 5) should be constrained to the other node (i.e., node i in Figure 5) to prevent sliding of the beam-column element under lateral loads because the shear resistance is not included in the zero-length section element.

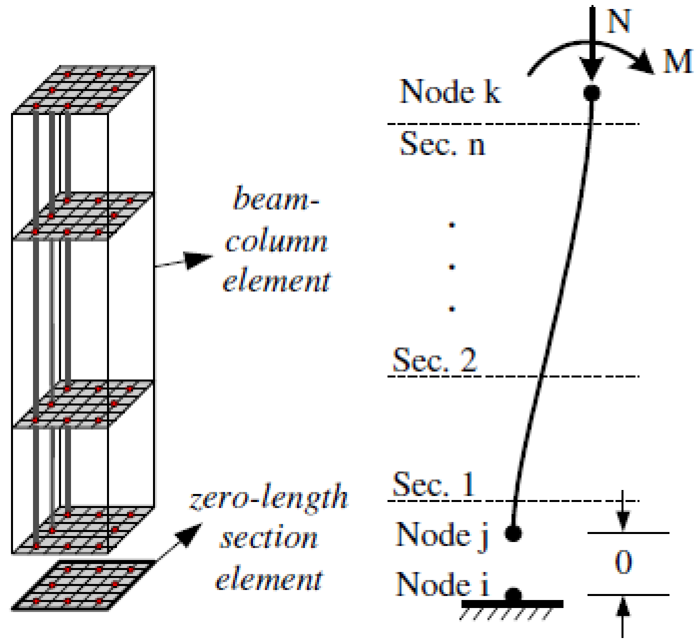


Figure. 4.Adding a zero-length section element to a beam-column element

The zero-length section element in Opensees is assumed to have a unit length such that the element deformations (i.e., elongation and rotation) are equal to the section deformations (i.e., axial strain and curvature). The material model for the steel fibers in the zero-length section element represents the bar slip instead of strain for a given bar stress. The uniaxial material model Bond_SP01 is developed for steel fibers in the zero-length section elements.

2.10.5. Material Model for Concrete Fibers

Similar to the model proposed for the steel fibers, a material model describing the monotonic response and hysteretic rules is also required for the concrete fibers. The combination of using the zero-length section element and enforcing the plane section assumption at the end of a flexural member impose high deformations to the extreme concrete fibers in the zero-length element. These deformations would likely correspond to concrete compressive strains significantly greater than the strain capacity stipulated by typical confined concrete models. Such high compressive strains at the end of flexural members are possible because of additional confinement effects expected from the adjoining members and because of complex localized deformation at the member end. Without further proof, it is suggested that the concrete fibers in the zero-length section element follow a concrete model in Opensees (e.g., Concrete02). To accommodate the large deformations expected to the extreme concrete fibers in the zero-length element, this concrete model may be assumed to follow a perfectly plastic behavior once the concrete strength reduces to 80% of the confined compressive strength. A parametric study has indicated that the simulation results would not be very sensitive to the compressive strain chosen to trigger the perfectly plastic behavior for the concrete fibers in the zero-length section element

2.11. Concrete Material Models

2.11.1. Unconfined and Confined Concrete

Kent and Park (1971) proposed a stress-strain equation for both unconfined and confined concrete. In their model they generalized Hognestad's (1951) equation to more completely describe the post-peak stress-strain behavior. In this model the ascending branch is represented by modifying the Hognestad second degree parabola by replacing by 0.85 f'c by f' and εco by 0.002.

$$\left[\frac{2\varepsilon_c}{\varepsilon_o} - \left(\frac{\varepsilon_c}{\varepsilon_o}\right)^2\right] \ldots\ldots 2.11.1$$

The post-peak branch was assumed to be a straight line whose slope was defined primarily as a function of concrete strength.

$$\left[\frac{2\varepsilon_c}{\varepsilon_o} - \left(\frac{\varepsilon_c}{\varepsilon_o}\right)^2\right] \ldots\ldots 2.11.2$$

$$f_c = f'_c\,[1 - Z(\varepsilon_c - \varepsilon_o)] \ldots\ldots 2.11.3$$

in which $z = \frac{0.5}{\varepsilon_{50u} - \varepsilon_{co}}$ …… 2.11.4

Where ε_{50u} = the strains corresponding to the stress equal to 50% of the maximum concrete strength for unconfined concrete.

$$\varepsilon_{50u} = \frac{3 + 0.29f_c{}'}{145f'_c - 1000} \quad [\ f_c{}' \text{ in Mpa}] \ldots\ldots 2.11.5$$

The Kent and Park model is represented in Figure 5.

Popovics (1973) proposed a single equation to describe unconfined concrete stress-strain behavior. A major appeal of this model is that it only requires three parameters to control the entire pre and post peak behavior, specifically f'c, ε_{co} and E_c.

$$\frac{f_c}{f'_c} = \frac{n\frac{\varepsilon_c}{\varepsilon_{co}}}{(n-1)+(\frac{\varepsilon_c}{\varepsilon_o})} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 2.11.6$$

Popovics equation works well for most normal strength concrete (f'$_c$=55Mpa), but it lacks the necessary control over the slope of the post-peak branch for high strength concrete.

Thorenfeldt et al. (1987) made modifications to the Popovics (1973) relation to adjust the descending branch of the concrete stress-strain relation. The authors proposed the following equation for the unconfined concrete stress-strain relation.

$$\frac{f_c}{f'_c} = \frac{n\frac{\varepsilon_c}{\varepsilon_{co}}}{(n-1)+(\frac{\varepsilon_c}{\varepsilon_o}^{nk})} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots .2.11.7$$

In the above equation 'k' takes a value of 1 for values of ε_c /ε_c < 1and values greater than 1 for of ε_c /ε_c >1. Thus by adjusting the value of 'k' the post-peak branch of the stress-strain relation can be made steeper. This approach can be used for high-strength concrete where the post-peak branch becomes steeper with increase in the concrete compressive strength.

Tsai (1988) proposed a generalized form of the Popovics (1973) equation which has greater control over the post-peak branch of the stress-strain relation. Tsai's equation consists of two additional parameters, one to control the ascending and a second to control the post-peak behavior of the stress-strain curve. The proposed stress-strain relation for unconfined concrete by Tsai is

$$y = \frac{mx}{1 + \left(m - \frac{n}{n-1}\right) + \frac{x^n}{n-1}} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 2.11.8$$

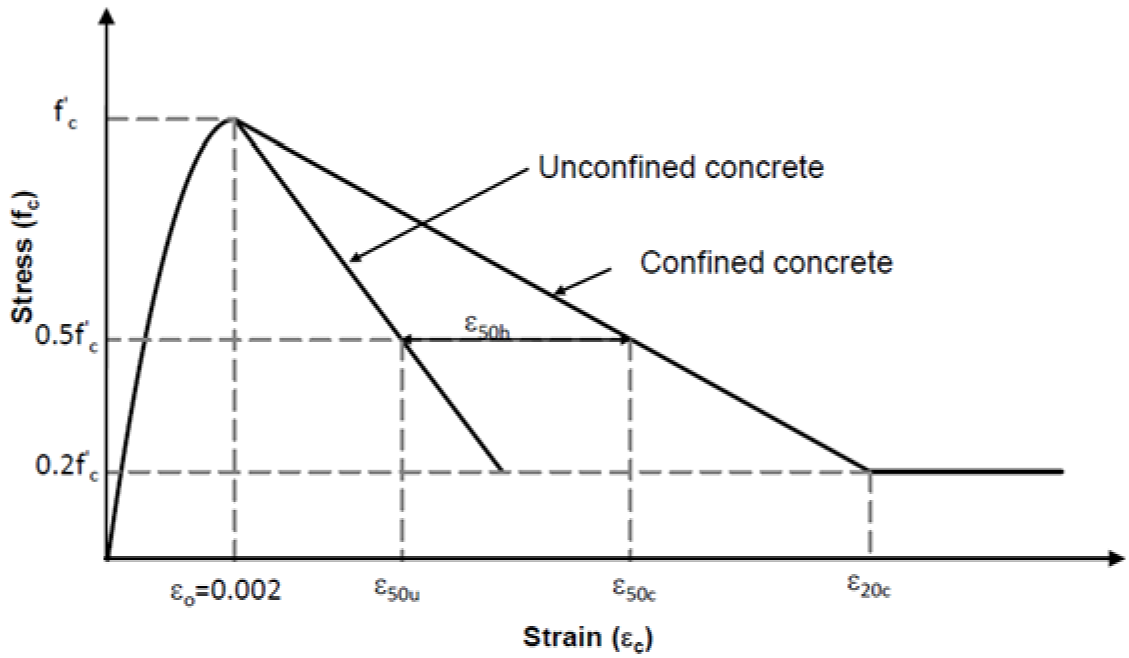


Figure. 5.proposed stress-strain model for confined and unconfined concrete-Kent and Park (1971) modeles

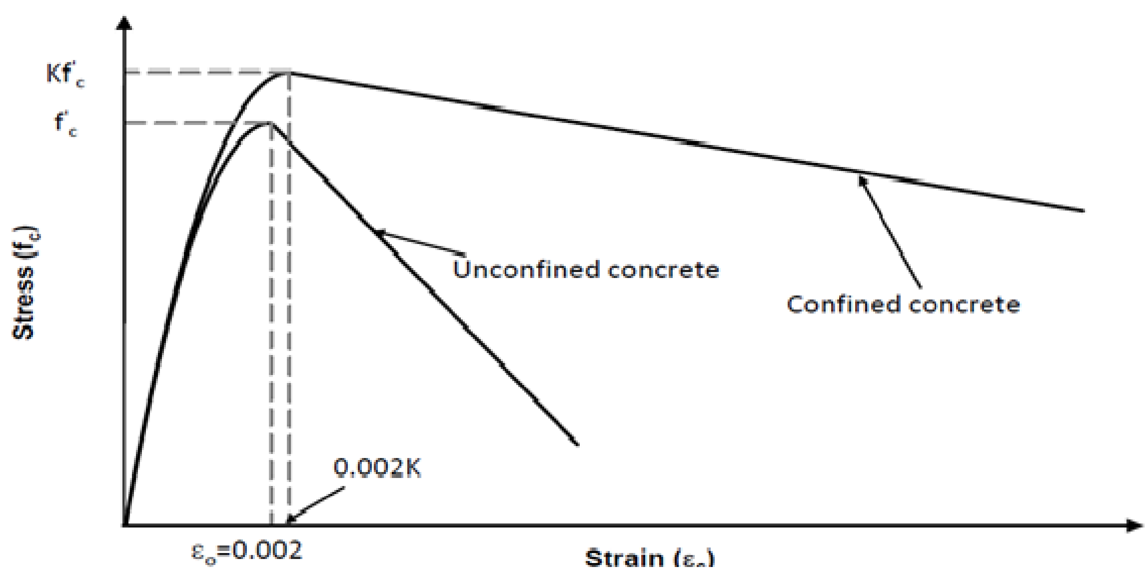


Figure. 6.stress-strain behavior of compressed concrete confined by rectangular steel hoops-Modified Kent and Park (Scott et al.1982) model

2.12. Preparation of Input data for the Modeling

The first task in developing a modeling is to clearly define the way data is organized withinthe Program. The input data for analysis of three dimensional reinforced concrete frames for earthquake (equivalent static) and gravity loading are organized as follows:

Fig.7 shows the basic structure of the model-building process using the scripting language.A series of tcls cripts store data defining the building geometry and material properties. Once the data has been stored into the lists and the arrays, OpenSees commands are called upon to use this data to define the model.Model defining progresses through, definition of the nodes, definition of the material models, definition of the sections, and definition of the elements. Once the whole model has been set up appropriate components sets the various analysis tools in OpenSees and performs the required analysis. Fig.7 shows the model building process, data transfer, and tasks required to accomplish the analysis of the model.

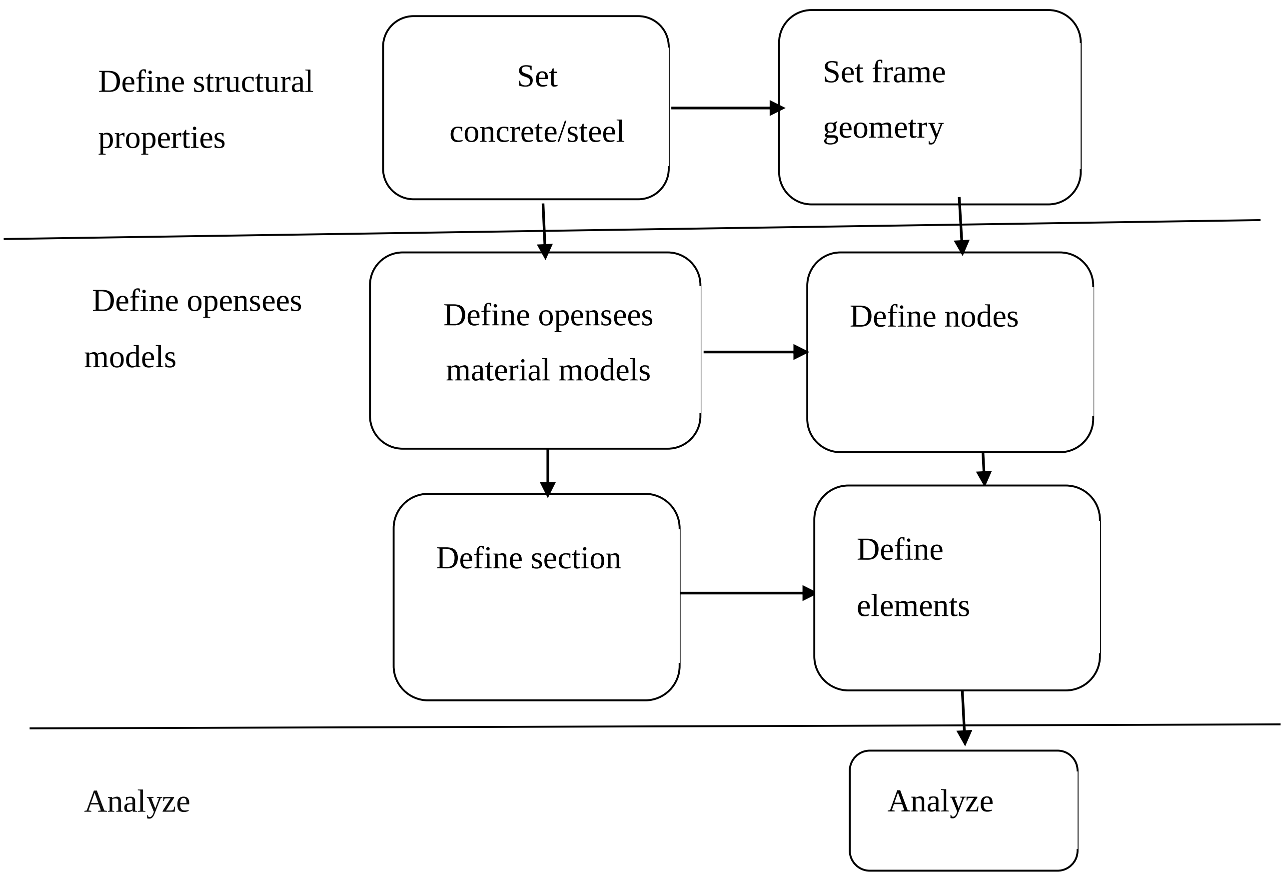


Figure. 7.Component of Analysis in opensees

2.13. Main abstraction in opensees framework

Opensees comprise of a set of modules to perform creation of the finite element model, specification of any analysis procedure, selection of quantities to be monitored during the analysis, and the output of the results. In each finite element analysis, an analysis is used to construct 4 main types of object, as shown in Fig 8.

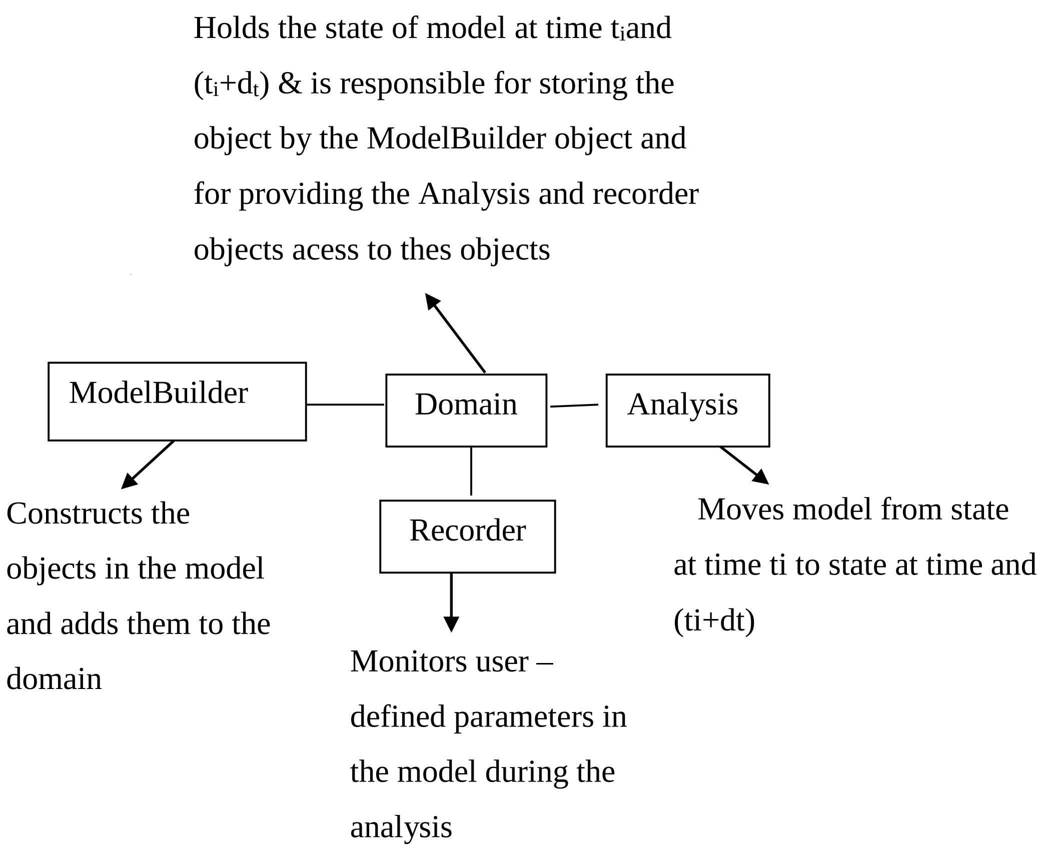


Figure. 8.Main abstraction in opensees framework

2.14. Modeling Development on opensees

In three dimensional frames, the number of degrees of freedom (dof) per node is 6, namely dx, dy, dz, rx,ry and rz (three translations, and three rotations), and therefore, the total number of degrees of freedom is 6 times the number of nodes (**n**).

2.14.1. ModelBuilder

The ModelBuilder is the object in the program responsible for building the following objects in the model and adding them to the domain [Node, Mass, Material, Section, Element, LoadPatternTimeSeries, Transformation, Block, Constraint].In this work three dimensional reinforced concrete frame structure is used.

2.14.2. Nodal Coordinates

Once the dimension of the problem is defined, it is recommended to define the coordinates of the nodes, the mass associated with each node and DOF and the boundary conditions at the nodes. The nodal coordinates are defined using the node command. The numbers of parameters associated with this command are referenced to the model command. In this work the coordinate of mass is not defined due to the analysis case that static analysis is used.

2.14.3. Boundary Conditions

The boundary conditions are defined using the fix command. The tag 0 represents an unconstrained (free) degree of freedom; the tags 1 represents a constrained (fixed) DOF.In this work the foundation for fixed type of frame structure were 6 degree of freedom and for soil-structure ,the first 3 degree of freedoms (displacements) were fixed and the rest threes' are pines (rotation).

2.14.4. Materials

Once the nodes have been defined, the next step towards defining elements is the material definition. Concrete02 and steel02 will be used for the structure under consideration of this work.The cover and core concrete will be modeled as different materials, using the same material type, but different stress and strain characteristics and different material tags.

2.14.5. Element Cross Section

Some element types require that the element cross section be defined a-priori, this is done using the section command. The section is used to represent force-deformation (or resultant stress-strain) relationships at beam-column and plate sample points. While there are many types of sections available; the fiber section will be used to define the cross section of the column in the structure under consideration. A fiber section has a general geometric configuration formed by sub regions of simpler, regular shapes (e.g. quadrilateral, circular, and triangular regions) called patches. In addition, individual or layers of reinforcement bars can be specified. Both beams and columns cross section's core concrete cover and reinforcements are defined using fiber section command.In this work the fiber section was formed by quadrilateral region for both beam and column section and reinforcement bars were arranged in layer.

2.14.6. Elements and Elements Connectivity

Once the element cross section has been defined, additional mechanical properties must be associated (aggregated) to it. Elastic torsion needs to be added to the column under consideration, using an elastic uniaxial material. The geometric transformation is used to relate the local element, and section, coordinates to the global system coordinates. In this work therefore, elastic torsion is used in the modeling of frame structure.

2.14.7. Nonlinear Beam Column Element

This is used to construct a nonlinearBeam Column element object, which is based on the non-iterative (or iterative) force formulation, and considers the spread of plasticity along the element. There are basically two types of Nonlinear Beam-Column Elements

Force based elements

- Distributed plasticity (nonlinearBeamColumn)
- Concentrated plasticity with elastic interior (beamWithHinges)
- Displacement based element
- Distributed plasticity with linear curvature distribution (dispBeamColumn)

In this work Displacement based element is used.

2.14.8. Loads and Analysis in Opensees

2.14.8.1. Gravity and other Constant Loads

Gravity loads are independent of the type of lateral loading and are considered part of the structural model.

In Opensees loads is applied in a three-step process:

- Loads must be defined in a load pattern.
- The analysis must be then defined and its features.
- The loads are then applied when the analysis is executed.

In this work gravity and lateral loads are applied to nodes of the 3D frame structure for openseesmodeling.

2.14.8.2. Gravity loads (live and dead loads)

Using EBCS provision for two ways slab, the gravity load is transferred from slabs to Beams. And these loads are then converted into nodal loads (gravity load and moment) for openseesmodeling and are applied to respective nodes based on the following calculation and sign is changed for moment.

Fixed end moment is given by: FEM=$WL^2/_{12}$ (KNm) and a nodal gravity load that is applied to each respective node is calculated by applying P= (WL/2) (KN) loads to each Node of the beam end.

2.14.8.3. Lateral Loads

Lateral load that is applied to the framed structural nodes in opensees modeling was calculated using EBCS 8 provision. These loads are applied to their respective nodes in the horizontal direction.

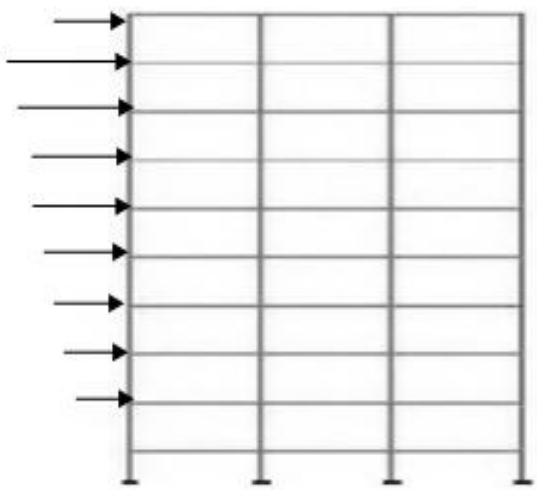


Figure. 9.Framed structure for modeling of opensees

2.14.9. Recorder Object

- Monitors user-defined parameters in the model during the analysis.
- Monitors the state of a domain component (node, element, etc.) during an analysis.
- Writes this state to a file or to a database at selected intervals during the analysis.

- Recorders can also be placed anywhere on a fiber section to measure fiber stresses and strains. When more than one material may occupy the location specified (such as a steel bar at the edge of the confined-concrete core), a preferred material can be specified. The location of the recorder is specified using the local coordinate system. If no fiber is located at that coordinate, a blank file will be output (very common error).In this work Recorder Object is placed before pushover analysis but after the defining lateral loads.

2.14.10. Pushover Analysis in opensees

Pushover analysis is a static, nonlinear procedure using simplified nonlinear technique to estimate seismic structural deformations. It is an incremental static analysis used to determine the force-displacement relationship, or the capacity curve, for a structure or structural element. The analysis involves applying horizontal loads, in a prescribed pattern, to the structure incrementally, i.e. pushing the structure and plotting the total applied shear force and associated lateral displacement at each increment, until the structure collapse. In this technique a computer model of the building is subjected to a lateral load of a certain shape (i.e., inverted triangular or uniform). In this work gravity loads (factored Live and Dead Loads transferred from two way slab to beams are converted to nodal loads) were applied to each nodes of the frame. Next, lateral loads are applied to each node with appropriate direction but, prior to it is an opensees script called "loadConst -time 0.0" which means the opensees program understand that gravity load is constant load applied at each node of the element, then additional loads of lateral load is applied. Finally pushover analysis in a loop is made to run both constant gravity load and lateral load with increasing condition and after the prescribed iteration, the pushover analysis will be successful or fail.

2.15. Foundation Model in opensees

2.15.1. Soil- Structure Interaction in opensees

According to the seismic improvement of current structure provision, the members of
Structure and foundation must be modeled together in Beam-on-Nonlinear-Winkler-Foundation (BNWF) model to consider soil structure interaction.

Soil-structure interaction can lead to modification of building response. Soil flexibility results in period elongation and damping increase. The main relevant impacts are to modify the overall lateral displacement and to provide additional flexibility at the base level that may relieve inelastic deformation demands in the superstructure. In this study two horizontal springs (lateral and sliding) and one vertical spring were used in main direction of structures to simulate soil structure interaction. The stiffness of each spring is estimated using Gazetas equation for shallow foundation stiffness. [Hutchinson2, P. R. (2008)]

2.15.2. Beam-on-Nonlinear Winkler Foundation (BNWF) Models

A spring responds only to loads acting parallel to its axis, so loads acting in a perpendicular direction have no effect on the response of the spring. Nonlinear springs for shallow foundations have been used in conjunction with gapping and damper elements. BNWF approach assumes that soil-foundation interface is closely-spaced, independent, inelastic spring elements. The BNWF model implemented into OpenSees consists of elastic beam-column elements that capture the structural footing behavior with independent zero-length soil elements that model the soil-footing behavior. Currently it is developed for two-dimensional analysis only for opensees modeling. Therefore, the one-dimensional elastic beam-column elements used for the footing have three degrees-of-freedom per node (i.e., horizontal, vertical, and rotation). One-dimensional uniaxial springs are used to simulate the vertical load displacement behavior (q-z), horizontal passive load-displacement behavior against the side of a footing (p-y), and horizontal shear-sliding behavior at the base of a footing (t-y). Moment-rotation behavior is captured by distributing vertical springs along the base of the footing.

The material models are mechanistic, based on an arrangement of various linear and nonlinear springs, gap elements, and dashpots. Radiation damping can be accounted for using a dashpot that is placed in parallel with the far-field elastic component. The backbone curves are thus characterized by a linear-elastic region, followed by an increasingly growing nonlinear region. The QzSimple2 material has an asymmetric hysteretic response, with a backbone curve defined by an ultimate load on the compression side and a reduced strength in tension to account for the low strength of soil in tension. The PxSimple2 material is envisioned to capture the passive resistance, associated stiffness, and potential gapping of embedded shallow footings subjected to lateral loads. This material model is characterized by a pinched hysteretic behavior, which can more suitably account for the phenomena of gapping during unloading on the opposite side of a footing. The TxSimple2material is intended to capture the frictional resistance along the base of a shallow foundation.

This material is characterized by a large initial stiffness and a broad hysteresis, as anticipated for frictional behavior associated with foundation sliding.

The functional forms and parameters describing the p-y, t-y, and q-z springs are similar, so only the q-z model is described here. The backbone curve has linear and nonlinear regions. The linear-elastic portion of the backbone curve is described by the initial stiffness k.$q = SK_z$, whereq represents the spring force, and s represents the spring deflection. The upper limit of the linear-elastic region, defined as q_0, is taken as a fraction of the ultimate loadqultas follows:

$q_0 = C_r q_{ult}$, where, Cr is a parameter specified in OpenSees.

The nonlinear (post-yield) portion of the backbone is described by:

$$q = q_{ult} - (q_{ult} - q_0\left[\left(\frac{cs_{50}}{cs_{50}+s-s_0}\right)^n\right], \text{for } s > s0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.15.1$$

Where S_{50} is the displacement at which 50% of the ultimate load is mobilized, $s_0$is the displacement at load q_0, and both c and n are constitutive parameters controlling the shape of the post-yield portion of the backbone curve. Matlock's (1970) recommended backbone for soft clay is closely approximated using c = 10, n = 5, and Cr = 0.35.And these values were used in the opensees modeling for initial and post yield backbone curve for this work.

It is evident from the above equations that the shape of spring backbone curves, which is basically a controlling factor for the SSI behavior, is mainly dependent on two physical parameters related to soil characteristics; namely, capacity (qult), and initial elastic stiffness (kin).The capacity and elastic stiffness of each spring is obtained by distributing the global footing capacity and stiffness utilizing proper tributary area of each spring.

The footing capacity is derived using the general bearing capacity equation from Terzaghi (1943) with shape, depth and inclination factors after Meyerhof (1963) as shown in the equations below.

$$Q_{Ult} = C\mathrm{N}_c F_{CS} F_{cd} F_{ci} + \Upsilon D_f N_q F_{qs} F_{qd} F_{qi} + 0.5\gamma B N_\gamma F\gamma_y F_{\gamma s} F_{\gamma d} F_{\gamma i} \ldots\ldots 2.15.2$$

Where, qult is the ultimate vertical bearing capacity per unit area of footing, c the cohesion, γ the unit weight of soil, Df is the depth of embedment, and B the width of footing. Bearing capacity factors, Nc, Nqand Nγarecalculated after Meyerhof (1963):

$$\mathrm{Nq} = \tan^2\left(45^0 + \frac{\emptyset}{2}\right) e^{\pi \tan(\emptyset)} \ldots\ldots\ldots\ldots\ldots 2.15.3$$

$$\mathrm{Nc} = (\mathrm{Nq} - 1)\cot\emptyset \ldots\ldots\ldots\ldots\ldots 2.15.4$$

$$\mathrm{N\Upsilon} = (\mathrm{Nq} - 1)\tan(1.4\emptyset) \ldots\ldots\ldots\ldots\ldots 2.15.5$$

For the p-x material, the ultimate lateral load capacity is determined as the total passive resisting force acting on the front side of the embedded footing. For homogeneous backfill against the footing, the passive resisting force can be calculated using a linearly varying pressure distribution resulting in the following expression:

$$\mathrm{Pult} = 0.5\Upsilon \mathrm{Df}^2\,\mathrm{Kp} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.15.6$$

Where, pult = passive earth pressure per unit length of footing, and Kp= passive earth pressure coefficient. For the t-x material, the lateral load capacity is the total sliding (frictional) resistance, which can be defined as the shear strength between the soil and the footing as:

$$t_{ult} = W_g \tan\delta + A_b c \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.15.7$$

Where, tult = frictional resistance per unit area of foundation, Wg = vertical force acting at the base of the foundation, δ = angle of friction between the foundation and soil (typically varying from 1/3φ to 2/3 φ) and Ab =the area of the base of footing in contact with the soil(=L × B).C=cohesion of soil.

The initial elastic stiffness (vertical and lateral) of the footing is derived from Gazetas (1991) as follows:

$$K_V = \frac{GL}{1-V}[0.73 + 1.54\left(\frac{B}{L}\right)^{0.75}] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.15.8$$

$$K_h = \frac{GL}{2-V}[2 + 2.5\left(\frac{B}{L}\right)^{0.85} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.15.9$$

where k_v and k_h are the vertical and lateral initial elastic stiffness of the footing, respectively; G is the shearmodulus of soil; ν is the Poisson’s ratio of soil; and Band L are the footing width and length, respectively. The instantaneous tangent stiffness kp, which describes the load-displacement relation within the post-yield or nonlinear region of the backbone curves, may be expressed as:

$$K_p = n(q_{ult} - q_0)\left[\frac{(cz_{50})^n}{(cz_{50}|-z_0 + z|)^{n+1}}\right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.15.10$$

In this paper the effects of nonlinear SSI on the structural response is studied in terms of base moment, base shear, story displacement, and inter-story drift.

2.15.3. Parameters used in the modeling of soil in opensees

Radiation damping (cz). This dashpot coefficient is considered to be a physicalparameter that is well documented in the literature. Theparameter is sensitive to soil stiffness, footing shape, aspect ratio andembedment.

Tension capacity (TP). The tension capacity parameter, TP, determines the maximum magnitude of the drag force in Component 1 of the nonlinear springs. It is the ratio of tension capacity to bearing capacity with typical selected values of 0 to 0.10 (as suggested in Boulanger et al., 1999), although, more recently some experts (e.g., Kutter) have recommended using a TP value of zero.

Distribution and magnitude of vertical stiffness. Two parameters are necessary to account for the distribution and magnitude of the vertical stiffness along the length of a footing: (1) the stiffness intensity ratio, Rk(where, Rk=Kend/Kmid); and (2) the end length ratio, Re (where, Re=Lend/2L). A variable stiffness distribution along the length is used to force the distributed BNWF spring model to match the overall rotational stiffness. The end region, Lend, is defined as the length of the edge region over which the stiffness is increased. Seismic Evaluation and Retrofit of Concrete Buildings (ATC, 1996) suggests the use of Lend = B/6 from each end of the footing. This expression of end length ratio is independent of the footing aspect ratio. Harden and Hutchinson (2009) suggest an expression that is a function of the footing aspect ratio.

Spring spacing (S). The spring spacing is input by the user as a fraction of the footing half-length L (S = le/L), where le is the non-normalized spring spacing. A maximum element length equal to 8% of the footing half-length (i.e., a minimum number of 25 springs along the full length of the footing) is recommended to provide numerical stability and reasonable accuracy.

Shape parameters (Cr, c, n). These parameters are hard-wired into the OpenSees implementation of the material models, meaning that they are not specified by users. The recommended values are soil-type dependent, and were developed based on comparisons of model prediction test data as described by Raychowdhury and Hutchinson (2008).

Different researcher did similar issues on non linear finite element modeling of soil structure interaction with different finite element software, but with dynamic loading only. This work investigates soil structure interaction like different researchers did on non-linear modeling of three dimensional reinforced concrete structures but, with opensees software and with static loading in low seismic zone (ground acceleration =0.05g).

Opensees is as its name indicates open software that can be modified based on the interest of analysts and a powerful finite element software for analysis of earthquake loading, and it is easy to use .The reason why this software is selected in this work is because it is the preferable software in researches as it can model both linear and non linear modeling of structures, soil- structures, and soil. Other advantage of opensees is that it can model different structures with different dimension, for example in this work the soil is modeled in two dimensional modeling where as the structure is modeled in three dimension.

This work focuses on the development of software program which models and analyzes reinforced concrete building using opensees software with and without considering soil effect under static seismic loads on displacement based beam-column element formulation and with Fiber model of section. The program has incorporated TCL command which is used to detect softening problem when the applied displacement at the top of the column increase. The outputs of the developed program are nodal displacement, storey drift, local and global forces, section deformation, section forces, reaction forces.These outputs are compared with sap software resuluts with similar modeling.

3. Development of modeling and Procedure on Opensees

3. 1.Structural Characteristics of the Models

In this study, moment resisting reinforced concrete building frame resting on a shallow foundation, representing conventional types of buildings in a relatively low risk earthquake prone zone has been chosen. In the selection of the frames' span width, attempt was made to make this width to be conforming to architectural norms and constructional practices of the conventional buildings in Ethiopia.Dimensional characteristics of the structural model are summarized in table 4.

Table 4.Dimensional characteristics of the frame

Number of Stories	Number of Bays	Storey Height (m)	Bay Width (m)	Total Height (m)	Total Width (m)
4	3	3	4	30	12

3.1.2. Analysis and Design of 3D Framed Reinforced Concrete Structure

A ten-story RC building located designed for gravity and earthquake loads (equivalent static loads) is studied. The rectangular plan of building is 4 m by 12 m. The story height is 3 m with a total height of 30 m. The structural system is symmetrical and plan layout is shown in Figure 5 .The frames of building were designed as gravity frames. The thicknesses of floor and roof slab are taken as 0.15 m and live load $2KN/m^2$. The Framed was analyzed using Etabs with gravity and static equivalent method of analysis for seismic load and designed and detailed as per EBCS, 1995 recommendation. The reinforcements and the dimension of column, beam, and foundation are as shown in the table 7. The building is supported on isolated footing with dimension of 3x3x0.6 m. It was designed for clay soil with bearing capacity of presumptive value of 280 K Pa. The cylinder compressive strengths of concrete columns, beams, and Foundation are 25 M Pa. The expected yield strength of steel deformed bars is 400 M Pa.

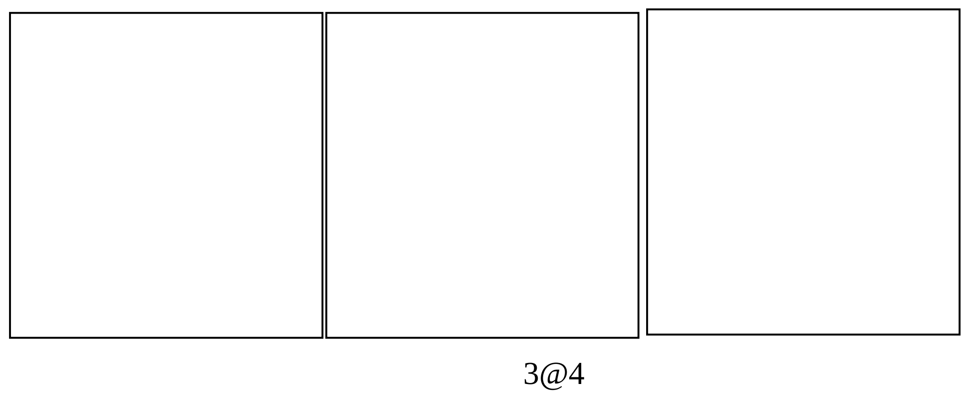


Figure. 10.The plan of the Building

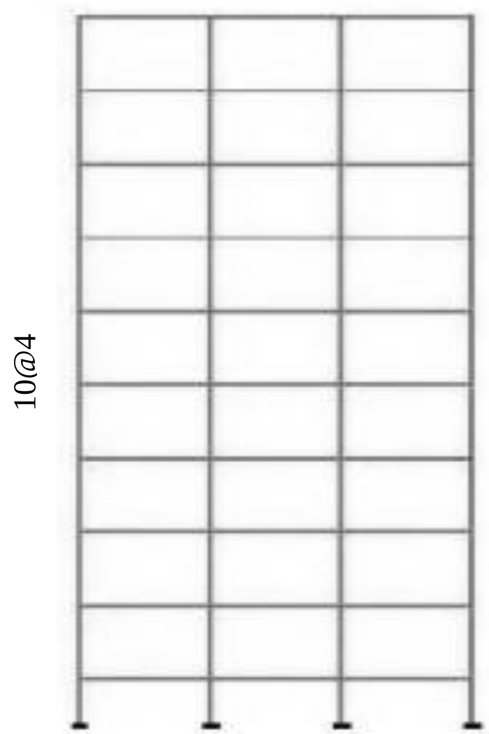


Figure. 11.Elevation View of the frame structure

Table 5.Dimension and Reinforcement of the Beam, column, and Foundation of Fame

Name	Dimension (mm)	Rebar (S400)Mpa	Concrete (C-25)Mpa	Reinforcement			
				Top (mm)	Middle (mm)	Bottom (mm)	Cover (mm)
Beams	250x600	S-400	C-25	2Ø20	2Ø20	2Ø20	25
columns	600x600	S-400	C-25	4Ø20	4Ø20	4Ø20	25
Footings	3000x300x600	S-400	C-25	Ø12 c/c 80 in both direction			30

In the modeling of the frame structure in opensees, the reinforcement, material grades of the each section of frame are inserted in the openseesprogramme .Gravity loads (dead and live loads) transferred from two ways slab (EBCS 2, 1995) to the beams are changed to nodal loads and are applied at each nodes of frame structure in the modeling. Seismic loads in zone II (ground acceleration=0.05g) using equivalent static load analysis is calculated based on the EBCS 8,1995 recommendation and are applied at each node of the frame based on the assumed direction in the modeling of the frame.

3.2. Modeling in finite element software (opensees)

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 soft wares** that can be used to develop programs. These are widely used to develop a program because thissoftwarehas built-in functions and reduce additional effort to define some functions like opensees that is used in modeling in this work. One thing that makes opensees different from other finite element software is that the command is written by the analysts but it has its own command, so it requires the knowledge of the input data and proper sequence of the command.

3.3. What is opensees?

Opensees (Open system for earthquake engineering simulation) is object-oriented open source software which allows users to implement finite element methods to model the structural and geotechnical systems and simulate the response under earthquake loading. It has been under development by the Pacific Earthquake Engineering Research Center since 1997. Because Opensees is object-oriented software framework, in a finite element application, mainly four types of objects, model builder object, domain object, recorder object, and analysis object need to be prepared. In Opensees, the interpretation is accomplished by adding commands into Tcl script for finite element analysis. Each command is associated with a C++ procedure that is built inside and is called by the interpreter to analyze the command (Mazzoni et. al 2006). Opensees is chosen in this study because it is an open source which is free to be used. Second, both linear and nonlinear structural and geotechnical models can be built in Opensees. Third, various simulations: static push-over analysis, static reversed-cyclic analysis, dynamic time-series analysis, uniform-support excitation, multi-support excitation can be effectively conducted. Last but not the least, Opensees provides a library of various materials, elements and analysis which is powerful for numerical simulation of nonlinear systems. In this work nonlinear structural and geotechnical models are built and static pushover simulation is conducted for three dimensional reinforced concrete structure building.

3.4. Preparation of Input data for the Modeling

The first task in developing a modeling is to clearly define the way data is organized withinthe Program. The input data for analysis of three dimensional reinforced concrete frames for earthquake (equivalent static) and gravity loading are organized as follows:

Fig.7 shows the basic structure of the model-building process using the scripting language.A series of tcls cripts store data defining the building geometry and material properties. Once the data has been stored into the lists and the arrays, OpenSees commands are called upon to use this data to define the model.Model defining progresses through, definition of the nodes, definition of the material models, definition of the sections, and definition of the elements. Once the whole model has been set up appropriate components sets the various analysis tools in OpenSees and performs the required analysis. Fig.7 shows the model building process, data transfer, and tasks required to accomplish the analysis of the model.

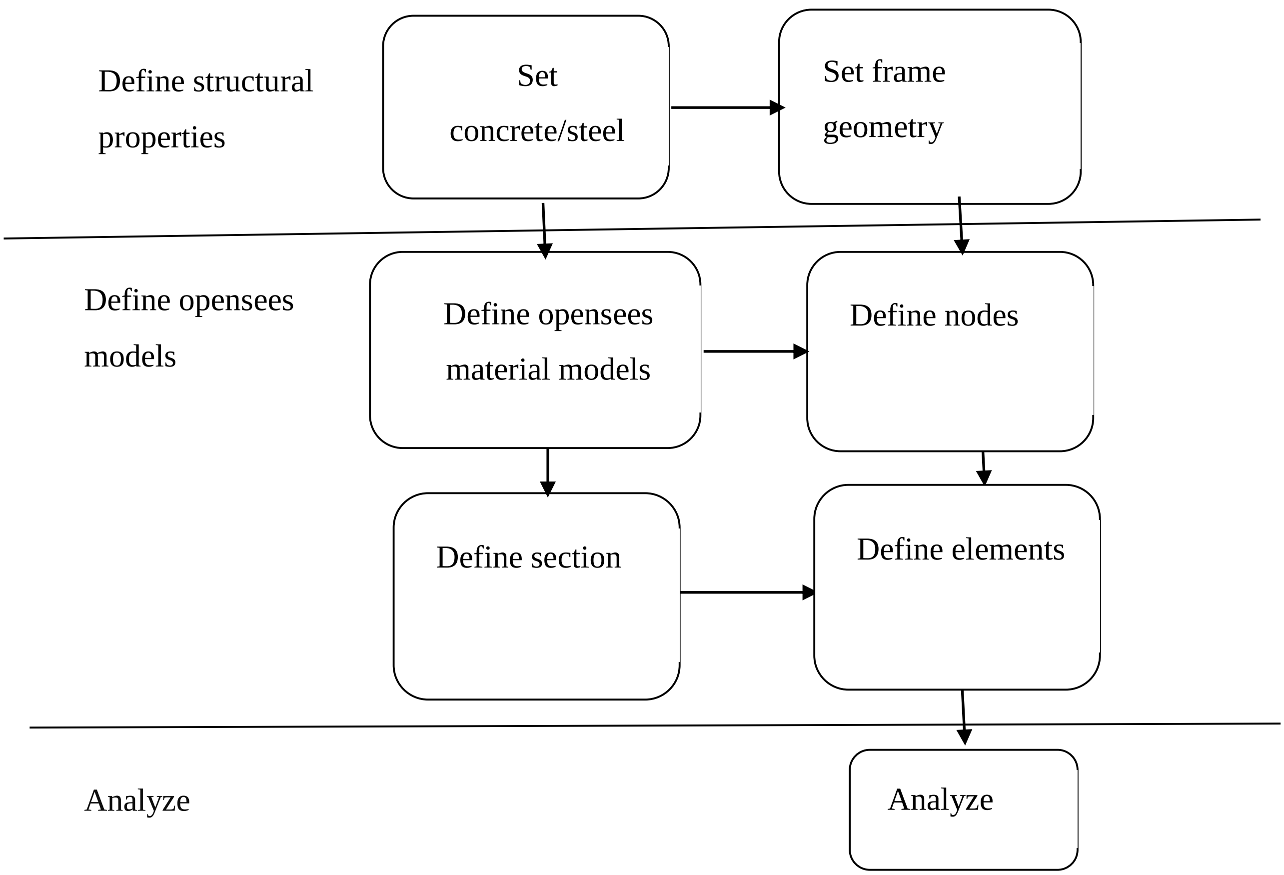


Figure. 12.Component of Analysis in opensees

3.5. Modeling Development on opensees

In three dimensional frames, the number of degrees of freedom (dof) per node is 6, namely dx, dy, dz, rx,ry and rz (three translations, and three rotations), and therefore, the total number of degrees of freedom is 6 times the number of nodes (**n**).

3.5.1. ModelBuilder

The ModelBuilder is the object in the program responsible for building the following objects in the model and adding them to the domain [Node, Mass, Material, Section, Element, LoadPatternTimeSeries, Transformation, Block, Constraint].In this work three dimensional reinforced concrete frame structure is used.

3.5.2. Nodal Coordinates

Once the dimension of the problem is defined, it is recommended to define the coordinates of the nodes, the mass associated with each node and DOF and the boundary conditions at the nodes.The nodal coordinates are defined using the node command. The numbers of parameters associated with this command are referenced to the model command. In this work the coordinate of mass is not defined due to the analysis case that static analysis is used.

3.5.3. Boundary Conditions

The boundary conditions are defined using the fix command. The tag 0 represents an unconstrained (free) degree of freedom; the tags 1 represents a constrained (fixed) DOF.In this work the foundation for fixed type of frame structure were 6 degree of freedom and for soil-structure ,the first 3 degree of freedoms (displacements) were fixed and the rest threes' are spring connection.

3.5.4. Materials

Once the nodes have been defined, the next step towards defining elements is the material definition. Concrete02 and steel02 will be used for the structure under consideration of this work.The cover and core concrete will be modeled as different materials, using the same material type, but different stress and strain characteristics and different material tags.

3.5.5. Element Cross Section

Some element types require that the element cross section be defined a-priori, this is done using the section command. The section is used to represent force-deformation (or resultant stress-strain) relationships at beam-column and plate sample points. While there are many types of sections available; the fiber section will be used to define the cross section of the column in the structure under consideration.A fiber section has a general geometric configuration formed by sub regions of simpler, regular shapes (e.g. quadrilateral, circular, and triangular regions) called patches. In addition, individual or layers of reinforcement bars can be specified. Both beams and columns cross section's core concrete cover and reinforcements are defined using fiber section command.In this work the fiber section was formed by quadrilateral region for both beam and column section and reinforcement bars were arranged in layer.

3.5.6. Elements and Elements Connectivity

Once the element cross section has been defined, additional mechanical properties must be associated (aggregated) to it.Elastic torsion needs to be added to the column under consideration, using an elastic uniaxial material. The geometric transformation is used to relate the local element, and section, coordinates to the global system coordinates.In this work therefore, elastic torsion is used in the modeling of frame structure.

3.5.7. Nonlinear Beam Column Element

This is used to construct a nonlinearBeam Column element object, which is based on the non-iterative (or iterative) force formulation, and considers the spread of plasticity along the element. There are basically two types of Nonlinear Beam-Column Elements

Force based elements

- Distributed plasticity (nonlinearBeamColumn)
- Concentrated plasticity with elastic interior (beamWithHinges)

Displacement based element

- Distributed plasticity with linear curvature distribution (dispBeamColumn)

In this work Displacement based element is used.

3.5.8. Loads and Analysis in Opensees

3.5.8.1. Gravity and other Constant Loads

Gravity loads are independent of the type of lateral loading and are considered part of the structural model.

3.5.8.2. Gravity loads (live and dead loads)

Using EBCS provision for two ways slab, the gravity load is transferred distributed loads from slabs to Beams. And these loads are then converted into nodal loads (gravity load and moment) for opensees modeling and are applied to respective nodes based on the following calculation and sign is changed for moment.

Fixed end moment is given by: FEM=$WL^2/_{12}$ (KNm) and a nodal gravity load that is applied to each respective node is calculated by applying (WL/2) loads to each Node of the beam end (axial load).

Gravity loads applied in each nodes of the frame for opensees modeling are given in the table below. These loads are applied in each nodes of the frame except the footings' nodes.These loads are applied for opensees modeling with axial load always downward and the two momements with their analysis direction.i.e clockwise negative and counterwise positive as can be seen in the table.

Table 6.Gravity loads for opensees modeling

Node	nodal loads		
	P	Mx	My
9 ,17 ,25, 33, 41, 49, 57, 65, 73	-8.75	-20	20
10 ,18, 26, 34, 42, 50, 58, 66, 74	-39.87	-59.7	-61.7
11, 19, 27, 35, 43, 51, 59, 67, 75	-49.150	-74.6	-39.6
12, 20, 28 ,36, 44 ,52, 60, 68, 76	-22.670	-30.19	30.23
13 ,21 ,29, 37, 45, 53, 61, 69, 77	-22.670	-30.19	30.23
14, 22, 30 ,38, 46, 54, 62, 70 ,78	-49.150	74.57	-39.6
15, 23, 31, 39, 47, 55, 63, 71 ,79	-39.780	59.7	-59.3
16 ,24, 32, 40, 48, 56 ,64, 72, 80	-8.750	20.31	-20.31
81	-8.750	-11.7	-11.7
82	-20.140	-33.7	-27
83	-20.104	-33	-27.1
84	-8.750	-11.7	11.7
85	-8.750	11.7	11.7
86	-20.140	33	-27.1
87	-20.140	33	27.1
88	-8.750	11.7	-11.7

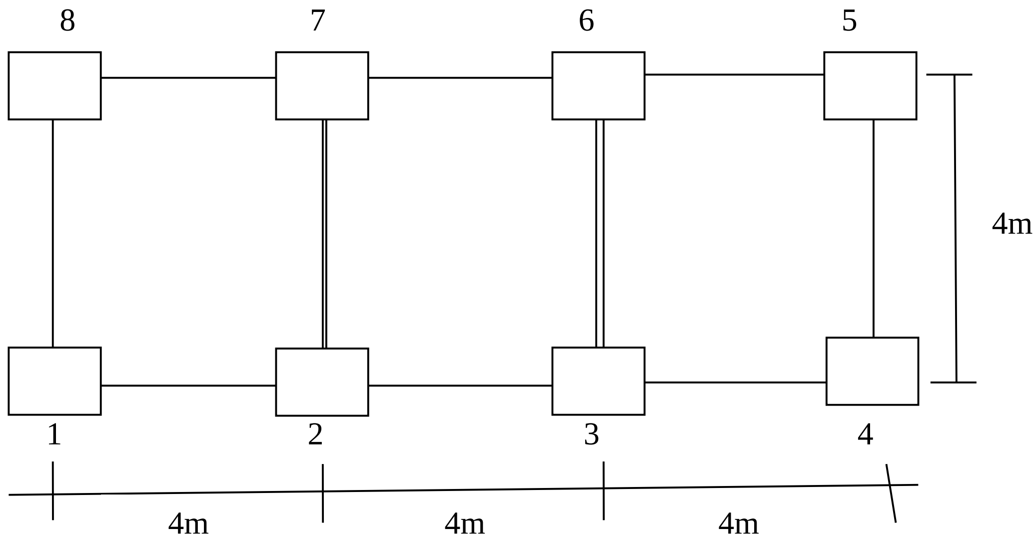


Figure. 13.Foundation layout of the building

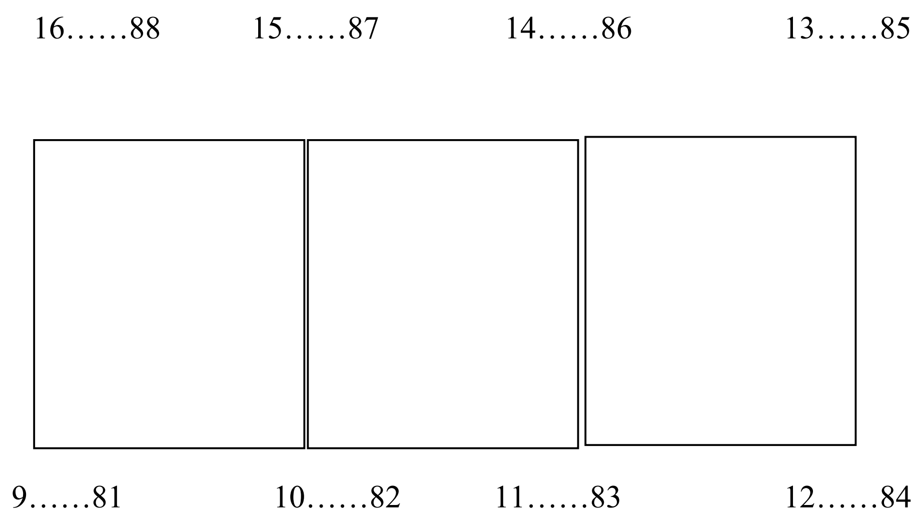


Figure. 14.nodding in opensees modeling (plan of the building frame from ground slab to roof slab)

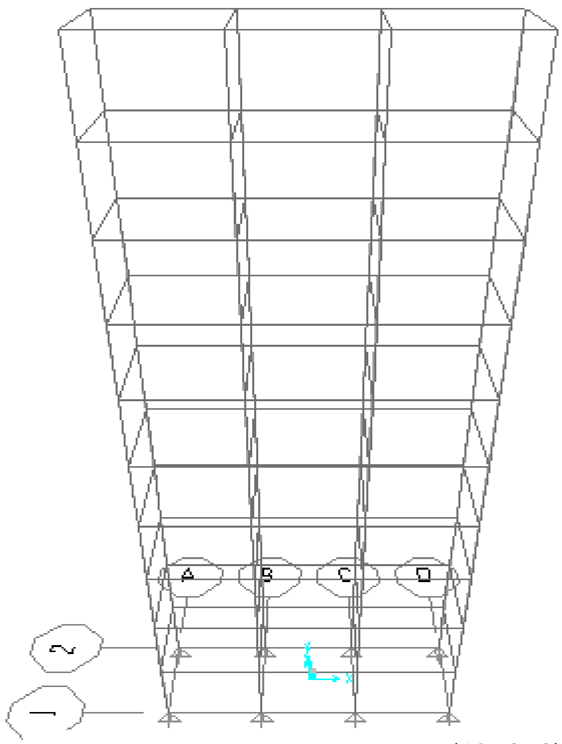


Figure. 15.3D Frame for opensees modeling

3.5.8.3. Lateral Loads

Lateral load that is applied to the framed structural nodes in opensees modeling was calculated using EBCS 8 provision. These loads are applied to their respective nodes in the horizontal direction.Node number starts from (0 0 0) coordinate in the counter clockwise direction. Total numbers of nodes are 88 and total numbers of elements are 172 .Nodal loads are applied at each respective node. Element number starts from column and then beams.

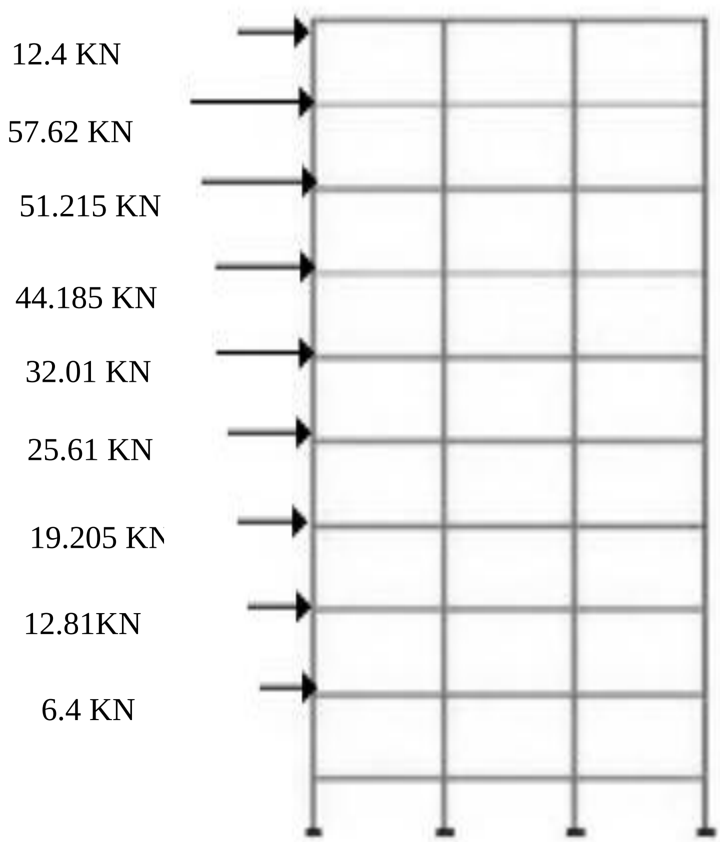


Figure. 16.Lateral Loads on the Frame

The frame was modeled in the above nodding system for ten story reinforced concrete building as can be seen in the plan of building in the Fig.11 .The nodes 1,2,3,4,5,6,7 and 8 are the footings. The rest are beam column nodes at each storey level and the numbering of the node are in the counter clockwise direction.For 10 storey 88 nodes are available as can be seen in the fig 16 and 17.

In Opensees loads is applied in a three-step process:

- Loads must be defined in a load pattern.
- The analysis must be then defined and its features.
- The loads are then applied when the analysis is executed.

3.6. Recorder Object

- Monitors user-defined parameters in the model during the analysis.
- Monitors the state of a domain component (node, element, etc.) during an analysis.
- Writes this state to a file or to a database at selected intervals during the analysis.
- Recorders can also be placed anywhere on a fiber section to measure fiber stresses and strains. When more than one material may occupy the location specified (such as a steel bar at the edge of the confined-concrete core), a preferred material can be specified. The location of the recorder is specified using the local coordinate system. If no fiber is located at that coordinate, a blank file will be output (very common error).In this work Recorder Object is placed before pushover analysis but after the defining lateral loads.

3.7. Pushover Analysis

Pushover analysis is a static, nonlinear procedure using simplified nonlinear technique to estimate seismic structural deformations. It is an incremental static analysis used to determine the force-displacement relationship, or the capacity curve, for a structure or structural element. The analysis involves applying horizontal loads, in a prescribed pattern, to the structure incrementally, i.e. pushing the structure and plotting the total applied shear force and associated lateral displacement at each increment, until the structure collapse. In this technique a computer model of the building is subjected to a lateral load of a certain shape (i.e., inverted triangular or uniform). In this work gravity loads (factored Live and Dead Loads transferred from two way slab to beams are converted to nodal loads) were applied to each nodes of the frame. Next, lateral loads are applied to each node with appropriate direction but, prior to it is an opensees script called "loadConst -time 0.0" which means the opensees program understand that gravity load is constant load applied at each node of the element, then additional loads of lateral load is applied. Finally pushover analysis in a loop is made to run both constant gravity load and lateral load with increasing condition and after the prescribed iteration, the pushover analysis will be successful or fail.

3.8. Procedure of the Program for Frame Structure (Without Soil Effect) modeling

The Flowchart for the analysis of three dimensional frames for static loads is presented in this section. Only flowchart to show the steps used to develop the program is presented here.

- Enter dimension of the frame structure/3D
- Enter geometry of the frame structure/LxWxD
- Enter node of the frame structure/coordinates of each node.
- Enter boundary condition of the frame structure/fixity
- Enter material properties of the frame structure
- Enter nominal concrete compressive strength
- Enter concrete elastic modulus
- Enter concrete shear modulus
- Enter compressive strength of confined concrete [1.3x nominal concerete compressive strength]
- Enter strain at maximum stress for confined concrete
- Enter strin at ultimate stress for confined concrete
- Enter unconfined concrete strength of unconfined concrete
- Enter strain at maximum strength of unconfined concrete

↓

- Enter ultimate stress for unconfined concrete
- Enter strain at ultimate stress for unconfined concrete
- Enter lambda (ratio between unloading slope and initial slope) for unconfined concrete
- Enter steel yield stress
- Enter tensile strength +tensions
- Enter tension softening stiffness
- Build core concrete (confined)
- Build cover concrete (unconfined)
- Build reinforcement material
- Enter section geometry (width X depth)
- Enter number of longitudinal reinforcement
- Enter cover for reinforcement bar

↓

- ➢ Enter fiber section properties
- Enter cover and core for both axes
- Enter the core patch, four cover patches, and reinforcement layer respectively
- Enter torsional stiffness and combine section properties
- Enter geometric transformation, integration points
- ➢ Create non linear column/beam element (element dispbeamColumn)

↓

- ➢ Analysis object
- Enter constraints transformation, numberer plain, system BandGeneral, test NormDispIncr, algorithm Newoton,IntegratorDisplacementControl, analyze static, analyze

↓

- ➢ Recorder Object
- Recorder node (displacement reaction forces at each node)
- Recorder element (local, global, section forces, deformation, stress strain, stiffness)
- Recorders drift (storey drift)

↓

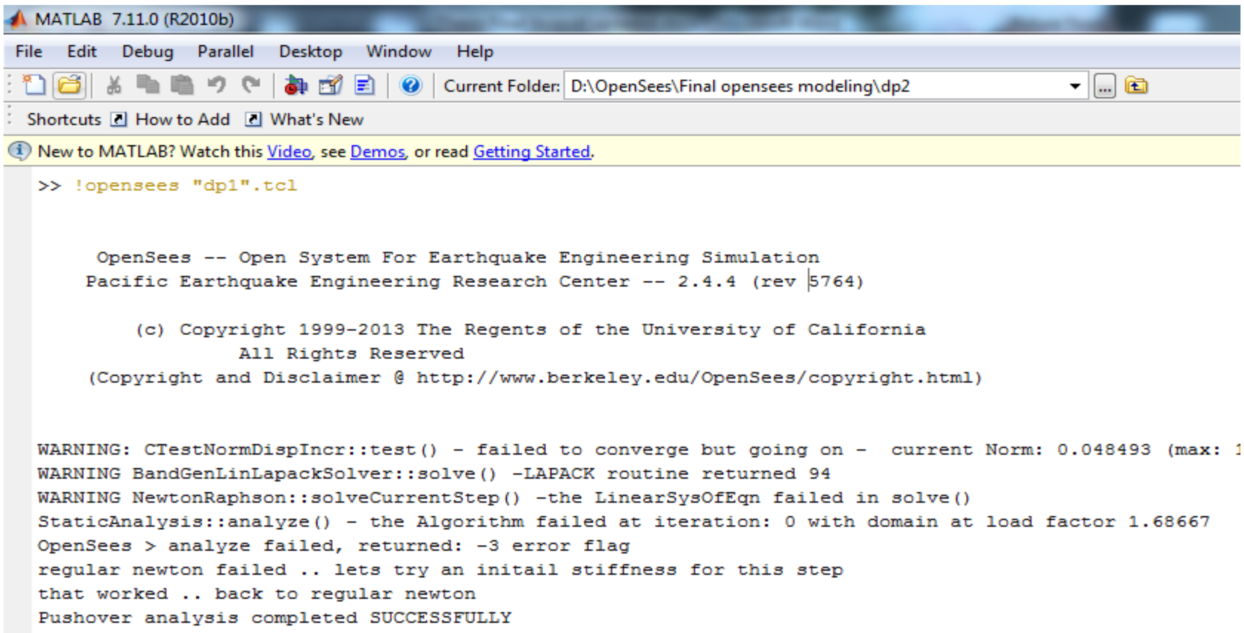
- ➢ Enter pushover analysis in loops

↓

- ➢ Print node
- ➢ Print ele
- ➢ Print sec

3.9 Running the Program

After the program is written in the flow chart sequence, then it is possible to run it in matlab.For this work, two models are taken-with (soil.tcl) and without soil (frame) effect. For example, dp1 (Frame structure without soil) was run in mat lab and showed the following message.



3.10. Foundation Model in opensees

3.10.1. Selection of structures and soil properties

A 10-storey, 3x1-bay with 4m each reinforced concrete moment resisting frame building that satisfy architectural requirement. The building was designed based on EBCS provision, with a floor area of 4x12 m^2. The section properties and geometric details of the foundation and columns are: Foundation depth=3 m, Foundation size 3x3x0.6 m, Column section: 0.6x0.6 m. The columns of the building are assumed to be supported on an isolated foundation resting on clay following soil properties: cohesion: 100 kPa, unit weight: 16kN/m3, shear modulus: 20MPa and Poisson’s ratio: 0.4, angle of friction: $Ø=0^0$. The effective shear modulus is obtained by reducing the maximum shear modulus corresponding to small strain values by 50% to represent the high strain modulus during significant earthquake loadings.

3.10.2. Beam-on-Nonlinear Winkler Foundation (BNWF) Models

A spring responds only to loads acting parallel to its axis, so loads acting in a perpendicular direction have no effect on the response of the spring. Nonlinear springs for shallow foundations have been used in conjunction with gapping and damper elements.BNWF approach assumes that soil-foundation interface is closely-spaced, independent, inelastic spring elements. The BNWF model implemented into OpenSees consists of elastic beam-column elements that capture the structural footing behavior with independent zero-length soil elements that model the soil-footing behavior. Currently it is developed for two-dimensional analysis only for opensees modeling. Therefore, the one-dimensional elastic beam-column elements used for the footing have three degrees-of-freedom per node (i.e., horizontal, vertical, and rotation). One-dimensional uniaxial springs are used to simulate the vertical load displacement behavior (q-z), horizontal passive load-displacement behavior against the side of a footing (p-y), and horizontal shear-sliding behavior at the base of a footing (t-y). Moment-rotation behavior is captured by distributing vertical springs along the base of the footing.

The footing capacity is derived using the general bearing capacity equation from Terzaghi (1943) with shape, depth and inclination factors after Meyerhof (1963) as shown in the equations below.

$$Q_{Ult} = C\mathrm{N}_c F_{CS} F_{cd} F_{ci} + \Upsilon D_f N_q F_{qs} F_{qd} F_{qi} + 0.5\gamma B N_\gamma F\gamma_y F_{\gamma s} F_{\gamma d} F_{\gamma i} \dots\dots\dots\dots\dots\dots..3.10.2.1$$

Nc =5.7, Nq=1, Nγ=0, Fcs=1.3, Fqs=1.2, Fγs=0.8, Fcd=1.09, Fqd=1.09, Fγd=1, Fci=1, Fqi=1, Fγd=1

Qult=100*5.7*1.3*1.09*1+16*3*1*1.2*1+0=865.29 KN/m

For the p-x material, the ultimate lateral load capacity is determined as the total passive resisting force acting on the front side of the embedded footing. For homogeneous backfill against the footing, the passive resisting force can be calculated using a linearly varying pressure distribution resulting in the following expression:

$$\mathrm{Pult} = 0.5\Upsilon \mathrm{Df}^2\,\mathrm{Kp} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots 3.10.2.2$$

Pult=0.5*1*3^2*16=72KN/m^2

Where,p_{ult} = passive earth pressure per unit length of footing, and Kp= passive earth pressure coefficient. For the t-x material, the lateral load capacity is the total sliding (frictional) resistance, which can be defined as the shear strength between the soil and the footing as:

$$\mathrm{t_{ult}} = \mathrm{W_g}\tan\delta + \mathrm{A_b c} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots..3.10.2.3$$

Where,t_{ult} = frictional resistance per unit area of foundation, Wg = vertical force acting at the base of the foundation, δ = angle of friction between the foundation and soil (typically varying from 1/3φ to 2/3 φ) and Ab =the area of the base of footing in contact with the soil (=L × B).C=cohesion of soil=100KN/$m^{2,}$ φ=0

t_{ult}=0+3*3*100 KN=900KN/m

The initial elastic stiffness (vertical and lateral) of the footing is derived from Gazetas (1991) as follows:

Table 7.Vertical springs

Material	Capacity
101	$q_1=(Q/L)*L_1$
102	$q_2=(Q/L)*L_2$
103	$q_3=(Q/L)*L_3$
Q=Total capacity L=total length	

Extreme end spacing=0.05m

Mid region spacing=0.25m

Extreme end (q_1) = (865.29/3)*(0.05)/2 =7.21 KN/m^2

end region (q_2)=(865.29/3)*(0.05) =14.42 KN/m^2

mid region(q_3) =(865.29/3)*(0.25) =72.11 KN/m^2

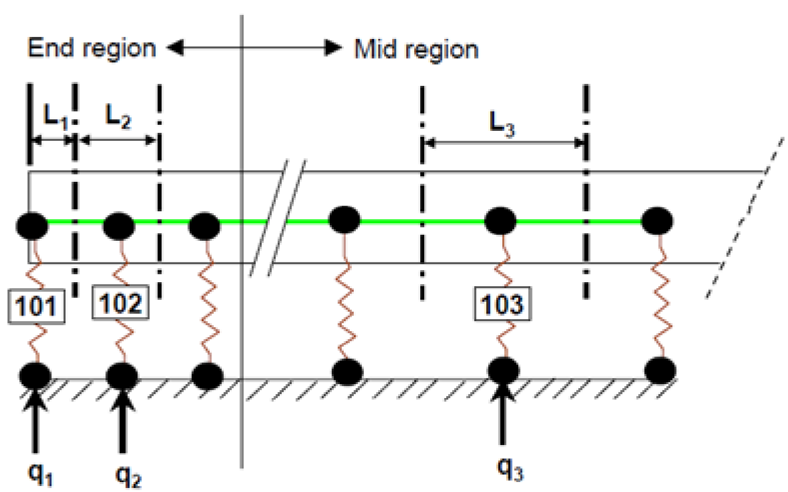


Figure. 17.Tributary capacity calculation

rotational damping (c_z) =0.05

tension capacity=0.1

Intial stiffness (vertical and horizontal)

$$K_V = \frac{GL}{1-V}[0.73 + 1.5\left(\frac{B}{L}\right)^{0.75} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.10.2.4$$

K_V =(20,000*3)/(1-0.4)*(0.73+1.5$^{0.75}$) =20,700KN/m

$$K_h = \frac{GL}{2-V}[2 + 2.5\left(\frac{B}{L}\right)^{0.85} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.10.2.5$$

K_h=20,000*3/(2-0.4)*(2+2.5$^{0.85}$) =156,712.5 KN/m

where k_v and k_h are the vertical and lateral initial elastic stiffness of the footing, respectively; G is the shearmodulus of soil; ν is the Poisson's ratio of soil; and Band L are the footing width and length, respectively.

In the modeling of soil-structure, the above ones are the inputs in soil modeling in opensees software.

3.11. Procedure of Shallow Foundation Modeling in Opensees

Define dimension of the soil modeling, nodding in CCW, Boundary condition (spring fixity), identify foundation base condition, define foundation material (Qzsimple2 (vertical spring), Pysimple2 (lateral spring), Tzsimple2 (sliding resistance)), Create element (zero-length element for vertical spring and horizontal, and elastic element for foundation element), finally, soil modeling with 2-Dimension and super structure with 3-Dimension are connected with multi-support constraints called " equalDOF".

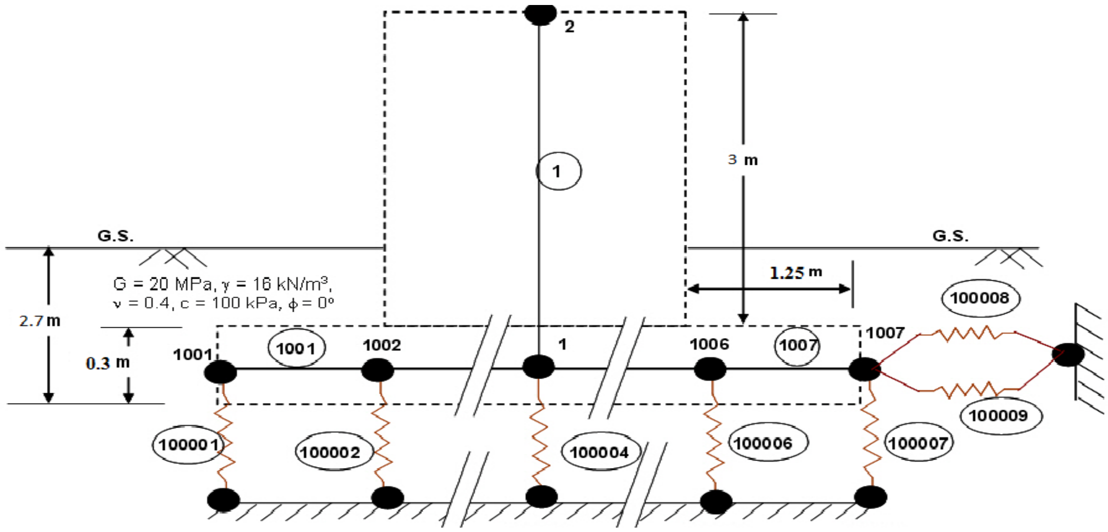


Figure. 18.Soil data used in the modeling of soil modeling (for only Type I soil) and shallow Foundation modeling for opensees

3.12. Procedure of the Program for Frame Structure (With Soil Effect) modeling

The Flow chart for the analysis of three dimensional frames for static loads with soil effect is presented in this section. Only flow chart to show the steps used to develop the program is presented here.

- Enter dimension of the soil structure/2D
- Enter foundation base condition for each foundation
- Enter multi-point constraints (equaldof command at column-foundation connection)
- Enter materials for shallow foundation
- Enter vertical spring element connectivity
- Enter horizontal spring element connectivity (rocking and sliding)
- Enter foundation element connectivity
- Define 3D Frame Structure Modeling
- Enter dimension of the frame structure/3D
- Enter geometry of the frame structure/LxWxD
- Enter node of the frame structure/coordinates of ech node

↓

- Connect soil modeling and structure with multi-point constraints called equalDof command
- Enter boundary condition of the frame structure/fixity
- Puts “soil modeling finished”
- Define 3D Frame Structure Modeling
- Enter dimension of the frame structure/3D
- Enter geometry of the frame structure/LxWxD
- Enter node of the frame structure/coordinates of ech node
- Connect soil modeling and structure with multi-point constraints called equalDof command
- Enter boundary condition of the frame structure/fixity
- Enter material properties of the frame structure
- Enter nominal concrete compressive strength of the frame structure
- Enter concrete elastic modulus of the frame structure
- Enter concrete shear modulus of the frame structure
- Enter compressive strength of confined concrete [1.3x nominal concrete compressive strength] for the frame structure
- Enter strain at maximum stress for confined concrete for the frame structure
- Enter strain at ultimate stress for confined concrete for the frame structure
- Enter unconfined concrete strength of unconfined concrete for the frame structure
- Enter strain at maximum strength of unconfined concrete for the frame structure
- Enter ultimate stress for unconfined concrete for the frame structure
- Enter strain at ultimate stress for unconfined concrete for the frame structure

- Enter section geometry (width depth)
  - Enter number of longitude reinforcement
  - Enter area of longitude reinforcement
  - Enter cover for reinforcement bar
  - Enter fiber section properties
  - Enter cover and core for both axes
  - Enter the core patch, four cover patches, and reinforcement layer respectively
  - Enter torsional stiffness and elastic torsional stiffness
  - Enter combine section properties
- Enter geometric transformation, integration points
- Create nonlinear column/beam element (element dispbeamColumn
  - Enter gravity loads at each load (factored dead and live load)
  - Enter loadConst-time 0.0
  - Enter lateral loads
- Analysis Object
  - Enter constraints transformation, numberer plain, system BandGeneral, and test NormDispIncr, algorithm Newton, Integrator Displacementcontrol, analysis static, analyze
- Define recorder
  - Recorder node (displacement reaction forces at each node)
  - Recorder element (local, global, section forces, deformation, stress strain, stiffness)
  - Recorder drifts (story drifts)
- Enter pushover analysis in loops
- Print node
- Print ele
- Print sec

4. Result and Discussions

This thesis work focuses on the geometric nonlinear modeling of reinforced concrete structure under gravity and seismic load (equivalent static method) to investigate the ten story three dimensional reinforced concrete structure behavior using Opensees software.

The output of the developed object oriented program is produced in the external file.

The outputs consist of nodal displacement, story drift, reaction forces, local and global forces, section forces, and section deformation.

Opensees program has its own command called “put” command that assure whether the modeling is correct or wrong ,so “put” command was used and push over analysis was run successfully .The accuracy of the solution may depend on the number of nodes ,number of iteration ,number of integration points ,number of fiber sections,etc.But as we increase the mentioned condition ,super computer may be needed to solve the problem and it may be time consuming.

In this work 2-bay in x direction 1-bay in y-direction with ten story reinforced concrete structure was modeled for two boundary conditions (with and without soil) using Opensees and outputs were obtained and compared.

Opensees is still under development and no clear programming tcl command is prepared for modeling of three dimensional soil-structure interactions for shallow foundation. So, in this work two dimensional soil modeling and three dimensional reinforced concrete frame structure modeling were made and the shallow foundations were connected to soil spring with TCL command ‘equalDOF’ [17].This command connect two nodes of two different modeling with different dimension and these nodes act the same-and is placed after the structural node is defined (modeling attached at appendeces).This modeling was compared with sap. The outputs of opensees for force demand and displacements are larger than sap outputs. This is good for structural strength. In the modeling of frame in sap, incorporation of SSI is a little bit smaller than the force demand without considering soil effect. Except my for fixed support modeling is smaller than with soil modeling as can be seen in the table below. This is because the earth quake load was changed into static equivalent and applied at each node in the horizontal direction in the modeling. So, foundation does not move compare to dynamic loading. That is why noticeable difference was not observed in sap modeling for both cases. But, opensees shows good result for static seismic loading as can be seen in table 8.When nodal displacements are compared in sap for fixed and spring modeling, incorporation of SSI make the frame to have more displacement. However, when we compare the outputs for both fixed and spring modeling in opensees, the force demand for SSI are smaller than fixed and larger displacements for spring modeling case for static pushover analysis. Therefore, opensees is best finite element software in modeling of frame structures in seismic zones for static loading.

The force based element can suffer from convergence issues as it needs to iterate to achieve convergence so does if a displacement beam element is subdivided, similar convergence issues. In this work displacement based element was used.In opensees the common problem that is thought to be occurred is tension softening issues-which occur at critical sections (critical sections are any points of beam or column).So, tension softening stiffness, were modeled in the material modeling for this problem.

The elements forces in the table 8 (base shear and base moments) are at story 1.For this elements 8 columns were used. The element 1 connects node 1 to node 9, element 2 connects node 2 to node 10, etc.

In this paper the effects of nonlinear SSI on the structural response is studied in terms of base moment, base shear, story displacement, and inter-story drift.

Sample examples:

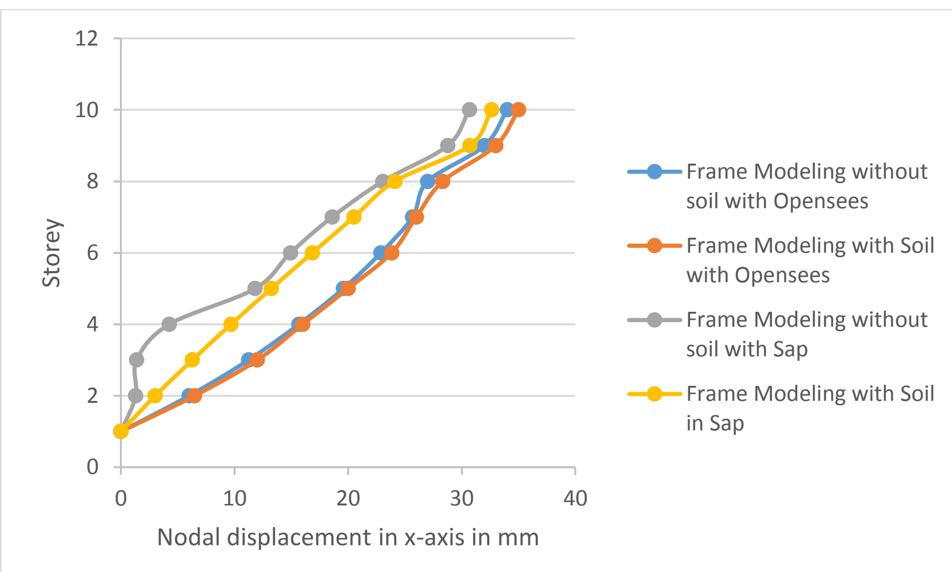


Fig. 19.story vs. Nodal Displacement in x-direction

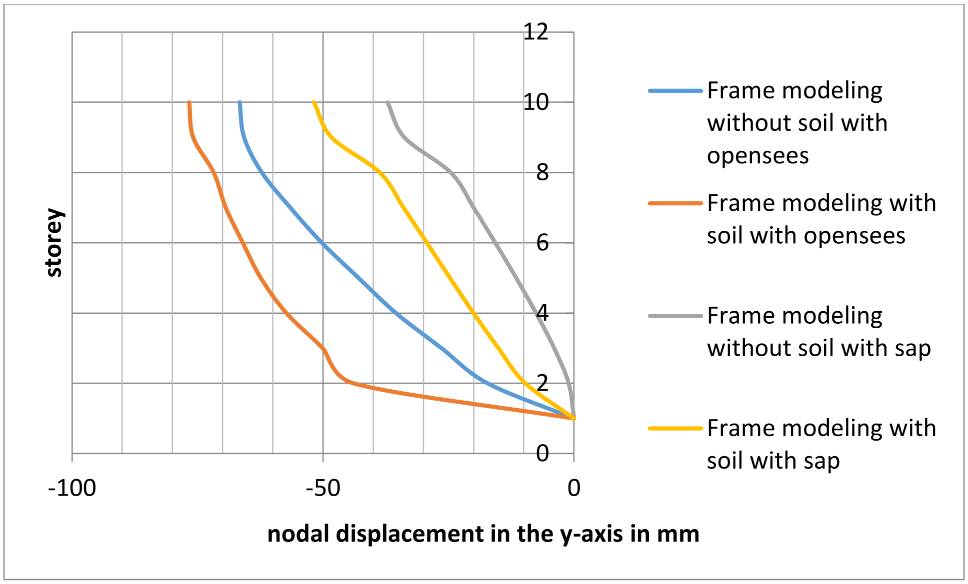


Figure. 20.story vs. Nodal Displacement in y-direction

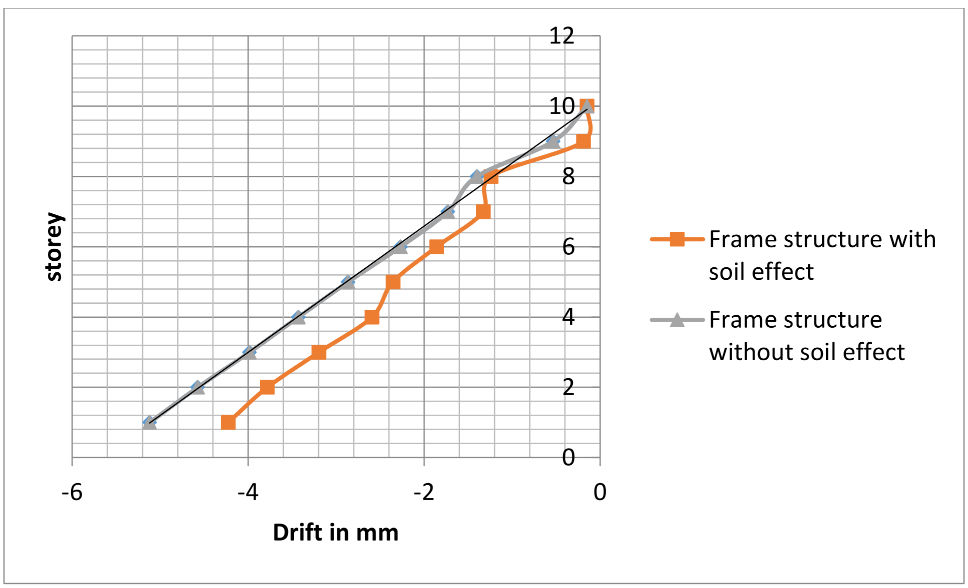


Figure. 21.Story vs. story drift

Table 8.Nodal forces for column elements at story-2 with opensees

Frame without soil (x-z plane) at y=0 with sap					Frame with soil (x-z plane) at y=0 with sap				
Element	Mz (Kn m)	Vy(Kn)	my(Knm)	Vz(Kn)	element	Mz(Knm)	Vy(Kn)	My(Knm)	vz(Kn)
1	-0.772	-7.974	-7.667	11.707	1	2.67E-14	-3.295	0	-5.82E-11
2	-15.436	-11.383	-30.761	20.346	2	-16.3017	-9.847	-56.364	30.737
3	-14.248	-10	-27.492	18.847	3	-13.949	-8.98	-24.162	17.954
4	-14.15	-9.556	-28.272	19	4	-13.8175	-8.948	-28.89	19.559
Total	**-44.606**	**-38.913**	**-94.192**	**69.9**	**Total**	**-44.0682**	**-31.07**	**-109.416**	**68.25**
Frame without soil (x-z plane) at y=4 with sap					Frame with soil (x-z plane) at y=4 with sap				
Element	mz	Vy	my	Vz	element	Mz	Vy	My	Vz
5	-13.713	-9.177	-28.45	19	5	13.414	9	-28.355	19.19
6	-13.23	-8.854	-28.734	19.075	6	-12.938	-8.57	-28.64	19.164
7	-12.749	-8.561	-29.116	19.19	7	-12.466	-8.333	-29	19.219
8	-12.3	-8.186	-29.6	19.377	8	-12	-7.982	-29.5185	19.374
Total	**-51.992**	**-34.778**	**-115.9**	**76.642**	**Total**	**-23.99**	**-15.885**	**-115.514**	**76.947**
Frame without soil (x-z plane) at y=0 with opensees					Frame with soil (x-z plane) at y=0 with opensees				
Element	mz	Vy	my	Vz	element	Mz	Vy	My	Vz
1	-57.63	9.84	-14.77	-86.44	1	-9.794	-11.705	9.243	-11.106
2	-9.1	-45.84	68.76	-13.66	2	37.402	3.33E+01	-3.79E+01	34.276
3	-28.82	-38.1	57.1	-43.23	3	3.515	4.421	-19.901	26.545
4	-60	-0.87	1.3	-90	4	4.164	2.518	98.959	-93.606
Total	**-155.55**	**-74.97**	**112.39**	**-233.33**	Total	**35.287**	**28.495**	**50.371**	**-43.891**

Frame without soil (x-z plane) at y=4 with opensees					Frame with soil (x-z plane) at y=4 with opensees				
element	Mz	Vy	My	Vz	element	Mz	Vy	My	Vz
5	60	-3.70E-15	3.20E-15	90.05	5	-8.844	-4.272	87.756	-86.825
6	53.67	-16.86	25.3	80	6	-0.00388	-0.0048	-66.862	62.899
7	24.18	-41	61.33	36.27	7	-0.0365	-0.0398	-40.32	37.715
8	50.1	21.86	-32.78	75.14	8	-12.659	-16.458	15.711	-17.087
Total	**187.95**	**-36**	**5.39E+01**	**281.46**	**Total**	**-21.5434**	**-20.7746**	**-3.715**	**-3.298**

Note: the unit of the moment and shear force is Knm and Kn respectively.

5. Conclusion and Recommendation

5.1. Conclusion

Based on the analysis result on ten-story reinforced concrete building frame using opensees finite element modeling, the following conclusions were arrived:

(1) Pushover analysis results indicate that with incorporation of SSI, the global force demand of a structure reduces while the nodal displacement demand increases. However, this alteration is more significant (reduction in force demand and increase in displacement demand) when the inelastic behavior of the soil-foundation interface is taken into account as can be seen in the Fig.19 and 20 and Table 8.

(2) Pushover analysis results indicate that with incorporation of nonlinear SSI, the drift demand of a structure is reduced, indicating that ignoring nonlinear SSI may lead to an over-conservative estimation of the structural drift as can be seen in Fig.21. However, the drift for both cases approach similar as the effects of support condition decrease as we move away from support.

(3) Finally, it may be concluded from this study that nonlinear behavior of the soil-foundation interface may play a crucial role in altering the seismic demands of a structure, indicating the necessity for incorporation of inelastic foundation behavior in modern design codes to accomplish more economic, yet safe structural design in a performance-based design framework.

(4) In general, the geometric non-linear static analysis was conducted using the Opensees software. The outputs indicate that the SSI can considerably affect the seismic response of building founded on clay soil conditions.Soil-structure interaction greatly depends on the ground motion characteristics, number of stories and horizontal capacity of earthquake resistance of building. If the lateral loading were dynamic, big output difference between frame structure with and without soil was expected as ground motion affects the response of the frame structure.

5.2. Recommendation

Recommendations drawn from the results are:

1. It is good to use dynamic loading in nonlinear modeling of framed structure to get a better result because dynamic loads moves the bed rock so it intensify the dynamic effects of the site and changes the structural response and give a clear clues how structural modeling with soil modeling and without affect structural response.

2. Even though using more nodes and more fiber are costly, better results are obtained. The input data should be arranged according to the format set in the modeling (Appendix 1, 2).

3. The developed modeling is advisable to be used for framed nonlinear modeling of reinforced concrete framed structure with and without underlying soil only, but, the modeling can be modified and used for any similar structure.

4. Since the modeling is geometric non linearity for three dimensional framed reinforced concrete structures, it is possible to find a separate stress-strain relation of rebar, confined concrete, or unconfined concrete in a single modeling. This work was done for very low seismic loads with earthquake zone II (with ground acceleration 0.05g), with soil data of shear modulus of 20Mpa, poison's ratio of 0.4, C=100Kpa, and $\text{Ø}=0^0$ and without considering soil effect and under shallow foundation. And, the results showed that soil effect should be considered in the modeling of structures in seismic prone areas. One can also see how a structure (reinforced or steel composite, etc.) responds in different soil type and foundation types.

Bibliography

[1].C. Zeris, A. K. (2012). Seismic Performance of a Typical Existing Core Frame RC Building With Soil Struct. Interaction. 15 WCEE , 1-10.

[2]Hamid Reza Tabatabaiefar, B. S. (2010). Effects of Dynamic Soil-Structure Interaction on Inelastic Behaviour of Mid-Rise Moment Resisting Buildings on Soft Soils. Centre for Built Infrastructure Research, University of Technology Sydney (UTS), , 1-11.

[3]Hokmabadi A.S., F. B. (2007). Seismic Response of Superstructure on Soft Soil Considering Soil-Pile-Structure Interaction. Broadway NSW , 557-550.

[4]Hora, V. G. (2012). A review on interaction behaviour of structure-foundation-soil system. International Journal of Engineering Research and Applications , 639-644.

[5]Jinu Mary Mathew1*, C. A. (2014). Seismic Response Of Rc Building By Considering Soil Structure Interaction. International Journal of Structural and Civil engineering Research , 159-172.

[6]LEWIS EDGERS, M. S. (2005). Modeling the Effects of Soil-Structure Interaction on a Tall Building Bearing on a Mat Foundation. Georgia: Civil Engineering Practice.

[7]Pallavi Ravishankar, D. D. (2013). Finite Element Modelling To Study Soil Structure Interaction Of Asymmetrical Tall Building. Paris: Centre for Earthquake Engineering International Institute of Information Technology Hyderabad - 500 032, INDIA .

[8]Rao*, R. C. (2012). Study of soil interaction in a model building frame with plinth beam supported by pile group. International Journal Advanced Structural Engineering , 1-15.

[9]Sara Casciati *, R. I. (2004). Dynamic FE analysis of South Memnon Colossus including 3D soil–foundation–structure interaction. computers and structures , 1720-1736.

[10]Shehata E Abdel Raheem1, 2. a. (2013). Soil-structure interaction modeling effects on seismic response of cable-stayed bridge tower. International Journal of Advanced Structural Engineering , 1-17.

[11]Dr.K.R.ARORA. (2004). Siol Mechanics And Foundation Engineering. Delhi: A.K.Jain.

[12]Hutchinson2,P.R.(2008).ShallowFoundationGenOpenSees-Documentation. ShallowFoundationGen OpenSees Documentation , 1-26.

[13]Kumari, B. (2010). Finite Element Modelling Of Reinforced Concrete FramE.

[14]Mo, T. T. (2010). Unified Theory Of Concrete Structures. University of Houston, USA: A John Wiley and Sons, Ltd., Publication.

[15]Silvia Mazzoni, F. M. (2007). OpenSees Command Language Manual. San Diego: UC.

[16] EBCS 1,2,7 and 8 1995

[17]http://opensees.berkely.edu/community/index.php

[18]http://opensees.berkeley.edu/wiki/index.php/Getting_Started

[19]http://opensees.berkeley.edu/wiki/index.php/Command_Manual

[20]http://opensees.berkeley.edu/wiki/index.php/OpenSees_User

[21]http://opensees.berkeley.edu/wiki/index.php/Examples_Manual

[22]http://opensees.berkeley.edu/wiki/index.php/OpenSees_Developer

[23]http://opensees.berkeley.edu/cgi-bin/cvsweb2.cgi/OpenSees/SRC/

Appendix I-Modeling of Frame structure with opensees without soil effect

units# units KN,m, sec

start of model generation

create modelBuilder with 3 dimension and 6 DOF/node

model BasicBuilder -ndm 3 -ndf 6

Define Geometry

set parameters for model geometry set h 3;# storey height set bx 4;# bay width in x- direction set by 4;# bay width in y- direction

Create nodes

tag x y z

node 1 0 0 0;node 2 [expr $bx] 0 0;node 3 [expr $bx*2] 0 0;node 4 [expr $bx*3] 0 0;node 5 [expr $bx*3] [expr $by] 0;node 6 [expr $bx*2] [expr $by] 0

node 7 [expr $bx*1] [expr $by] 0;node 8 0 [expr $by] 0;node 9 0 0 [expr $h];node 10 [expr $bx] 0 [expr $h];node 11 [expr $bx*2] 0 [expr $h] node 12 [expr $bx*3] 0 [expr $h];node 13 [expr $bx*3] [expr $by] [expr $h];node 14 [expr $bx*2] [expr $by] [expr $h];node 15 [expr $bx*1] [expr $by] [expr $h] node 16 0 [expr $by] [expr $h];node 17 0 0 [expr $h*2];node 18 [expr $bx] 0 [expr $h*2];node 19 [expr $bx*2] 0 [expr $h*2];node 20 [expr $bx*3] 0 [expr $h*2]

node 21 [expr $bx*3] [expr $by] [expr $h*2];node 22 [expr $bx*2] [expr $by] [expr $h*2];node 23 [expr $bx*1] [expr $by] [expr $h*2];node 24 0 [expr $by] [expr $h*2] node 25 0 0 [expr $h*3];node 26 [expr $bx] 0 [expr $h*3];node 27 [expr $bx*2] 0 [expr $h*3];node

28 [expr $bx*3] 0 [expr $h*3] node 29 [expr $bx*3] [expr $by] [expr $h*3];node 30 [expr $bx*2] [expr $by] [expr $h*3];node 31 [expr $bx*1] [expr $by] [expr $h*3]

node 32 0 [expr $by] [expr $h*3];node 33 0 0 [expr $h*4];node 34 [expr $bx] 0 [expr $h*4];node 35 [expr $bx*2] 0 [expr $h*4]

node 36 [expr $bx*3] 0 [expr $h*4];node 37 [expr $bx*3] [expr $by] [expr $h*4];node 38 [expr $bx *2] [expr $by] [expr $h*4] node 39 [expr $bx*1] [expr $by] [expr $h*4];node 40 0 [expr $by] [expr $h*4];node 41 0 0 [expr $h*5];node 42 [expr $bx] 0 [expr $h*5] node 43 [expr $bx*2] 0 [expr $h*5];node 44 [expr $bx*3] 0 [expr $h*5];node 45 [expr $bx*3] [expr

$by] [expr $h*5] node 46 [expr $bx*2] [expr $by] [expr $h*5];node 47 [expr $bx*1] [expr $by] [expr $h*5];node 48 0 [expr $by] [expr $h*5] node 49 0 0 [expr $h*6];node 50 [expr $bx] 0 [expr $h*6];node 51 [expr $bx*2] 0 [expr $h*6];node 52 [expr $bx*3] 0 [expr $h*6] node 53 [expr

$bx*3] [expr $by] [expr $h*6];node 54 [expr $bx*2] [expr $by] [expr $h*6];node 55 [expr $bx*1] [expr $by] [expr $h*6] node 56 0 [expr $by] [expr $h*6];node 57 0 0 [expr $h*7] ;node 58 [expr $bx] 0 [expr $h*7];node 59 [expr $bx*2] 0 [expr $h*7]

node 60 [expr $bx*3] 0 [expr $h*7];node 61 [expr $bx*3] [expr $by] [expr $h*7];node 62 [expr $bx *2] [expr $by] [expr $h*7] node 63 [expr $bx*1] [expr $by] [expr $h*7];node 64 0 [expr $by] [expr $h*7];node 65 0 0 [expr $h*8];node 66 [expr $bx] 0 [expr $h*8]

node 67 [expr $bx*2] 0 [expr $h*8];node 68 [expr $bx*3] 0 [expr $h*8];node 69 [expr $bx*3] [expr $by] [expr $h*8] node 70 [expr $bx*2] [expr $by] [expr $h*8];node 71 [expr $bx*1] [expr $by] [expr $h*8];node 72 0 [expr $by] [expr $h*8] node 73 0 0 [expr $h*9];node 74 [expr $bx] 0 [expr $h*9];node 75 [expr $bx*2] 0 [expr $h*9]; node 76 [expr $bx*3] 0 [expr $h*9]

node 77 [expr $bx*3] [expr $by] [expr $h*9];node 78 [expr $bx*2] [expr $by] [expr $h*9];node 79 [expr $bx*1] [expr $by] [expr $h*9] node 80 0 [expr $by] [expr $h*9];node 81 0 0 [expr $h*10];node 82 [expr $bx] 0 [expr $h*10]; node 83 [expr $bx*2] 0 [expr $h*10]

node 84 [expr $bx*3] 0 [expr $h*10];node 85 [expr $bx*3] [expr $by] [expr $h*10];node 86 [expr $bx*2] [expr $by] [expr $h*10] node 87 [expr $bx*1] [expr $by] [expr $h*10];node 88 0 [expr $by] [expr $h*10]

set base constraints(boundery conditions)

tag Dx Dy Dz Rx Ry Rz

fix $nodeTag (ndf $ConstrValues)

foreach node {1 2 3 4 5 6 7 8} {fix $node 1 1 1 1 1 1}

foreach node {9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 } {fix $node 0 0 1 0 0 1}

MATERIAL parameters ---set IDconcCore 1;# material ID tag -- confined core concrete set IDconcCover 2;# material ID tag -- unconfined cover concrete set IDreinf 3;# material ID tag -- reinforcement

nominal concrete compressive strength

set fc 25000;# CONCRETE Compressive Strength,(+Tension, -Compression) set Ec 29000000;# Concrete Elastic Modulus

set GC [expr $Ec/2.4] # confined concrete

set Kfc 1.3;# ratio of confined to unconfined concrete strength set fc1C [expr $Kfc*$fc];# CONFINED concrete (mander model), maximum stress set eps1C [expr 2.*$fc1C/$Ec];# strain at maximum stress set fc2C [expr 0.2*$fc1C];# ultimate stress

set eps2C [expr 5*$eps1C];# strain at ultimate stress

unconfined concrete

set fc1U $fc;# UNCONFINED concrete (todeschini parabolic model), maximum stress set eps1U -0.003;# strain at maximum strength of unconfined concrete

set fc2U [expr 0.2*$fc1U]; # ultimate stress set eps2U -0.01;# strain at ultimate stress

set lambda 0.1;# ratio between unloading slope at $eps2 and initial slope $ Ec

tensile-strength properties set ftC [expr -0.14*$fc1C];# tensile strength +tension set ftU [expr -0.14*$fc1U];# tensile strength +tension set Ets [expr $ftU/0.002];# tension softening stiffness

----------set Fy 260870;# STEEL yield stress set Es 200000000;# modulus of steel set Bs 0.01;# strain-hardening ratio set R0 18;# control the transition from elastic to plastic branches set cR1 0.925;# control the transition from elastic to plastic branches set cR2 0.15;# control the transition from elastic to plastic branches

uniaxialMaterial Concrete02 $IDconcCore $fc1C $eps1C $fc2C $eps2C $lambda $ftC $Ets;# build core concrete (confined)

uniaxialMaterial Concrete02 $IDconcCover $fc1U $eps1U $fc2U $eps2U $lambda $ftU $Ets;# build cover concrete (unconfined)

uniaxialMaterial Steel02 $IDreinf $Fy $Es $Bs $R0 $cR1 $cR2;# build reinforcement material

section GEOMETRY --set HSec 0.5;# Column Depth set BSec 0.4;# Column Width set coverSec 0.025; # Column cover to reinforcing steel NA. set numBarsSec 6;# number of longitudinal-reinforcement bars in steel layer. (symmetric top & bot)

set barAreaSec [expr 0.25*3.14*0.4];# area of longitudinal-reinforcement bars(diam.=20mm)

set SecTag 1;# set tag for symmetric section

FIBER SECTION properties ---# column section: set coverx [expr $HSec/2.0]; # The distance from the section y-axis to the edge of the cover concrete -- outer edge of cover concrete

set covery [expr $BSec/2.0]; # The distance from the section x-axis to the edge of the cover concrete -- outer edge of cover concrete set corex [expr $coverx-$coverSec]; # The distance from the section y-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete

set corey [expr $covery-$coverSec]; # The distance from the section x-axis to the edge of

the core concrete -- edge of the core concrete/inner edge of cover concrete set nfCorex 30;# number of fibers for concrete in x-direction -- core concrete set nfCorey 24;# number of fibers for concrete in y-direction

set nfCoverx 30;# number of fibers for concrete in x-direction -- cover concrete set nfCovery 24;# number of fibers for concrete in y-direction

Define the fiber section (define the core patch,four cover patchs and Reinforcement layer respectively)

section fiberSec $SecTag {patch quadr $IDconcCore $nfCorey $nfCorex -$corex $corey -$corex $corey $corex -$corey $corex $corey patch quadr $IDconcCover 1 $nfCoverx -$coverx $covery -$corex $corey $corex $corey $coverx $covery patch quadr $IDconcCover 1 $nfCoverx -$corex $corey -$coverx -$covery $coverx -$covery $corex -$corey patch quadr $IDconcCover $nfCovery 1 -$coverx $covery -$coverx -$covery -$corex -$corey -$corex $corey patch quadr $IDconcCover $nfCovery 1 $corex $corey $corex -$corey $coverx -$covery $coverx $covery layer straight $IDreinf $numBarsSec $barAreaSec $corex $corey $corex -$corey layer straight $IDreinf $numBarsSec $barAreaSec -$corex $corey -$corex -$corey }

assign torsional Stiffness for 3D Model set SecTagTorsion 98;# ID tag for torsional section behavior set SecTag3D 3; # ID tag for combined behavior for 3D model set GJ [expr $GC *0.1685*.50* 1/3*pow(0.4,3)];# torsional stiffness

uniaxialMaterial Elastic $SecTagTorsion $GJ;# define elastic torsional stiffness

section Aggregator $SecTag3D $SecTagTorsion T -section $SecTag; # combine section properties

number of column integration points (sections)

ID tag for column transformation, defining element normal set IDcolTransf 1

#geomTransf PDelta $transfTag $vecxzX $vecxzY $vecxzZ <-jntOffset $dXi $dYi $dZi $dXj $dYj $dZj> geomTransf PDelta $IDcolTransf 1 0 0 set colSec 3

set np 4

Create the non-linear column elements' connectivity

element dispBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens> element dispBeamColumn 1 1 9 $np $colSec 1;element dispBeamColumn 2 2 10 $np $colSec 1;element dispBeamColumn 3 3 11 $np $colSec 1

element dispBeamColumn 4 4 12 $np $colSec 1;element dispBeamColumn 5 5 13 $np $colSec 1;element dispBeamColumn 6 6 14 $np $colSec 1

element dispBeamColumn 7 7 15 $np $colSec 1;element dispBeamColumn 8 8 16 $np $colSec 1;element dispBeamColumn 9 9 17 $np $colSec 1

element dispBeamColumn 10 10 18 $np $colSec 1;element dispBeamColumn 11 11 19 $np $colSec 1; element dispBeamColumn 12 12 20 $np $colSec 1

element dispBeamColumn 13 13 21 $np $colSec 1;element dispBeamColumn 14 14 22 $np $colSec 1; element dispBeamColumn 15 15 23 $np $colSec 1

element dispBeamColumn 16 16 24 $np $colSec 1;element dispBeamColumn 17 17 25 $np $colSec 1; element dispBeamColumn 18 18 26 $np $colSec 1

element dispBeamColumn 19 19 27 $np $colSec 1;element dispBeamColumn 20 20 28 $np $colSec 1; element dispBeamColumn 21 21 29 $np $colSec 1

element dispBeamColumn 22 22 30 $np $colSec 1;element dispBeamColumn 23 23 31 $np $colSec 1; element dispBeamColumn 24 24 32 $np $colSec 1

element dispBeamColumn 25 25 33 $np $colSec 1;element dispBeamColumn 26 26 34 $np $colSec 1; element dispBeamColumn 27 27 35 $np $colSec 1

element dispBeamColumn 28 28 36 $np $colSec 1;element dispBeamColumn 29 29 37 $np $colSec 1; element dispBeamColumn 30 30 38 $np $colSec 1

element dispBeamColumn 31 31 39 $np $colSec 1;element dispBeamColumn 32 32 40 $np $colSec 1; element dispBeamColumn 33 33 41 $np $colSec 1

element dispBeamColumn 34 34 42 $np $colSec 1;element dispBeamColumn 35 35 43 $np $colSec 1; element dispBeamColumn 36 36 44 $np $colSec 1

element dispBeamColumn 37 37 45 $np $colSec 1;element dispBeamColumn 38 38 46 $np $colSec 1; element dispBeamColumn 39 39 47 $np $colSec 1

element dispBeamColumn 40 40 48 $np $colSec 1;element dispBeamColumn 41 41 49 $np $colSec 1; element dispBeamColumn 42 42 50 $np $colSec 1

element dispBeamColumn 43 43 51 $np $colSec 1;element dispBeamColumn 44 44 52 $np $colSec 1; element dispBeamColumn 45 45 53 $np $colSec 1

element dispBeamColumn 46 46 54 $np $colSec 1;element dispBeamColumn 47 47 55 $np $colSec 1; element dispBeamColumn 48 48 56 $np $colSec 1

element dispBeamColumn 49 49 57 $np $colSec 1;element dispBeamColumn 50 50 58 $np $colSec 1; element dispBeamColumn 51 51 59 $np $colSec 1

element dispBeamColumn 52 52 60 $np $colSec 1;element dispBeamColumn 53 53 61 $np $colSec 1; element dispBeamColumn 54 54 62 $np $colSec 1

element dispBeamColumn 55 55 63 $np $colSec 1;element dispBeamColumn 56 56 64 $np $colSec 1; element dispBeamColumn 57 57 65 $np $colSec 1

element dispBeamColumn 58 58 66 $np $colSec 1;element dispBeamColumn 59 59 67 $np $colSec 1; element dispBeamColumn 60 60 68 $np $colSec 1

element dispBeamColumn 61 61 69 $np $colSec 1;element dispBeamColumn 62 62 70 $np $colSec 1; element dispBeamColumn 63 63 71 $np $colSec 1

element dispBeamColumn 64 64 72 $np $colSec 1;element dispBeamColumn 65 65 73 $np $colSec 1; element dispBeamColumn 66 66 74 $np $colSec 1

element dispBeamColumn 67 67 75 $np $colSec 1;element dispBeamColumn 68 68 76 $np $colSec 1; element dispBeamColumn 69 69 77 $np $colSec 1

element dispBeamColumn 70 70 78 $np $colSec 1;element dispBeamColumn 71 71 79 $np $colSec 1; element dispBeamColumn 72 72 80 $np $colSec 1

element dispBeamColumn 73 73 81 $np $colSec 1;element dispBeamColumn 74 74 82 $np $colSec 1; element dispBeamColumn 75 75 83 $np $colSec 1

element dispBeamColumn 76 76 84 $np $colSec 1;element dispBeamColumn 77 77 85 $np $colSec 1; element dispBeamColumn 78 78 86 $np $colSec 1

element dispBeamColumn 79 79 87 $np $colSec 1;element dispBeamColumn 80 80 88 $np $colSec 1

define beam element # section GEOMETRY of beam set d 0.5 ;# bema Depth set b 0.25; # beam Width set coverSec1 0.025;# beam cover to reinforcing steel NA.

set numBarsSec 4;# number of longitudinal-reinforcement bars in steel layer. (symmetric top & bot)

set barAreaSec [expr 3.14*0.25*0.196]; # area of longitudinal-reinforcement bars(diam=14mm)
set SecTag 4;# set tag for symmetric section #Beam section:

set coverx1 [expr $d/2.0]; # The distance from the section z-axis to the edge of the cover concrete -- outer edge of cover concrete

set covery1 [expr $b/2.0]; # The distance from the section y-axis to the edge of the cover concrete -- outer edge of cover concrete

set corex1 [expr $coverx1-$coverSec1]; # The distance from the section z-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete

set corey1 [expr $covery1-$coverSec1]; # The distance from the section y-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete set nfCorex1 30;# number of fibers for concrete in y-direction -- core concrete set nfCorey1 25;# number of fibers for concrete in z-direction

set nfCoverx1 30;# number of fibers for concrete in y-direction -- cover concrete set nfCovery1 25;# number of fibers for concrete in z-direction

Define the fiber section (define the core patch,four cover patchs and Reinforcement layer respectively) set IDconcCover1 1 set IDconcCore1 2

section fiberSec $SecTag {patch quadr $IDconcCore $nfCorey1 $nfCorex1 -$corex1 $corey1 -$corex1 -$corey1 $corex1 -$corey1 $corex1 $corey1 patch quadr $IDconcCover1 $nfCoverx1 -$coverx1 $covery1 -$corex1 $corey1 $corex1 $corey1 $coverx1 $covery1 patch quadr $IDconcCover1 $nfCoverx1 -$corex1 -$corey1 -$coverx1 -$covery1 $coverx1 -$covery1 $corex1 -$corey1 patch quadr $IDconcCover1 $nfCovery1 -$coverx1 $covery1 -$coverx1 -$covery1 -$corex1 -$corey1 $corex1 $corey1 patch quadr $IDconcCover $nfCovery1 $corex1 $corey1 $corex1 -$corey1 $coverx1 -$covery1 $coverx1 $covery1 layer straight $IDreinf $numBarsSec $barAreaSec $corex1 $corey1

$corex1 -$corey1 layer straight $IDreinf $numBarsSec $barAreaSec -$corex1 $corey1 -$corex1 $corey1 }

assign torsional Stiffness for 3D Model set SecTagTorsion 99;# ID tag for torsional section behavior set SecTag3D 5; # ID tag for combined behavior for 3D model set GJbeam [expr $GC *0.163*0.30* 1/3*pow(0.250,3)]

uniaxialMaterial Elastic $SecTagTorsion $GJ; # define elastic torsional stiffness

section Aggregator $SecTag3D $SecTagTorsion T -section $SecTag; # combine section properties; set IDbeamTransf 2

Geometric transformation for beams set IDbeamTransf 2

geomTransf Linear $transfTag $vecxzX $vecxzY $vecxzZ <-jntOffset $dXi $dYi $dZi $dXj $dYj $dZj> geomTransf Linear $IDbeamTransf 1 1 0 set beamSec 5

number of beam integration points (sections) set np 4

Create beam element

element nonlinearBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens> <-iter $maxIters $tol>

Create the non-linearbeam elements' connectivity

element dispBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens element dispBeamColumn 81 9 10 $np $beamSec 2;element dispBeamColumn 82 10 11 $np $beamSec 2; element dispBeamColumn 83 11 12 $np $beamSec 2

element dispBeamColumn 84 12 13 $np $beamSec 2;element dispBeamColumn 85 13 14 $np $beamSec 2; element dispBeamColumn 86 14 15 $np $beamSec 2

element dispBeamColumn 87 15 16 $np $beamSec 2;element dispBeamColumn 88 10 15 $np $beamSec 2; element dispBeamColumn 89 11 14 $np $beamSec 2

element dispBeamColumn 90 17 18 $np $beamSec 2;element dispBeamColumn 91 18 19 $np $beamSec 2; element dispBeamColumn 92 19 20 $np $beamSec 2

element dispBeamColumn 93 20 21 $np $beamSec 2;element dispBeamColumn 94 21 22 $np $beamSec 2; element dispBeamColumn 95 22 23 $np $beamSec 2

element dispBeamColumn 96 23 24 $np $beamSec 2;element dispBeamColumn 97 18 23 $np $beamSec 2; element dispBeamColumn 98 19 22 $np $beamSec 2

element dispBeamColumn 99 25 26 $np $beamSec 2;element dispBeamColumn 100 26 27 $np $beamSec 2; element dispBeamColumn 101 27 28 $np $beamSec 2

element dispBeamColumn 102 28 29 $np $beamSec 2;element dispBeamColumn 103 29 30 $np $beamSec 2; element dispBeamColumn 104 30 31 $np $beamSec 2

element dispBeamColumn 105 31 32 $np $beamSec 2;element dispBeamColumn 106 26 31 $np element dispBeamColumn 107 27 30 $np $beamSec 2

element dispBeamColumn 108 33 34 $np $beamSec 2;element dispBeamColumn 109 34 35 $np $beamSec 2; element dispBeamColumn 110 35 36 $np $beamSec 2

element dispBeamColumn 111 36 37 $np $beamSec 2;element dispBeamColumn 112 37 38 $np $beamSec 2; element dispBeamColumn 113 38 39 $np $beamSec 2

element dispBeamColumn 114 39 40 $np $beamSec 2;element dispBeamColumn 115 34 39 $np $beamSec 2; element dispBeamColumn 116 35 38 $np $beamSec 2

element dispBeamColumn 117 40 41 $np $beamSec 2;element dispBeamColumn 118 41 42 $np $beamSec 2; element dispBeamColumn 119 42 43 $np $beamSec 2

element dispBeamColumn 120 43 44 $np $beamSec 2;element dispBeamColumn 121 44 45 $np $beamSec 2; element dispBeamColumn 122 45 46 $np $beamSec 2

element dispBeamColumn 123 46 47 $np $beamSec 2;element dispBeamColumn 124 47 48 $np $beamSec 2; element dispBeamColumn 125 42 47 $np $beamSec 2

element dispBeamColumn 126 43 46 $np $beamSec 2;element dispBeamColumn 127 49 50 $np $beamSec 2; element dispBeamColumn 128 50 51 $np $beamSec 2

element dispBeamColumn 129 51 52 $np $beamSec 2;element dispBeamColumn 130 52 53 $np $beamSec 2; element dispBeamColumn 131 53 54 $np $beamSec 2

element dispBeamColumn 132 54 55 $np $beamSec 2;element dispBeamColumn 133 55 56 $np $beamSec 2; element dispBeamColumn 134 50 55 $np $beamSec 2

element dispBeamColumn 135 51 54 $np $beamSec 2;element dispBeamColumn 136 57 58 $np $beamSec 2; element dispBeamColumn 137 58 59 $np $beamSec 2

element dispBeamColumn 138 59 60 $np $beamSec 2;element dispBeamColumn 139 60 61 $np $beamSec 2; element dispBeamColumn 140 61 62 $np $beamSec 2

element dispBeamColumn 141 62 63 $np $beamSec 2;element dispBeamColumn 142 63 64 $np $beamSec 2; element dispBeamColumn 143 58 63 $np $beamSec 2

element dispBeamColumn 144 59 62 $np $beamSec 2;element dispBeamColumn 145 65 66 $np $beamSec 2; element dispBeamColumn 146 66 67 $np $beamSec 2

element dispBeamColumn 147 67 68 $np $beamSec 2;element dispBeamColumn 148 68 69 $np $beamSec 2; element dispBeamColumn 149 69 70 $np $beamSec 2

element dispBeamColumn 150 70 71 $np $beamSec 2;element dispBeamColumn 151 71 72 $np $beamSec 2; element dispBeamColumn 152 66 71 $np $beamSec 2

element dispBeamColumn 153 67 60 $np $beamSec 2;element dispBeamColumn 154 73 74 $np $beamSec 2; element dispBeamColumn 155 74 75 $np $beamSec 2

element dispBeamColumn 156 75 76 $np $beamSec 2;element dispBeamColumn 157 76 77 $np $beamSec 2; element dispBeamColumn 158 77 78 $np $beamSec 2

element dispBeamColumn 159 78 79 $np $beamSec 2;element dispBeamColumn 160 79 80 $np $beamSec 2; element dispBeamColumn 161 74 79 $np $beamSec 2

element dispBeamColumn 162 75 78 $np $beamSec 2;element dispBeamColumn 163 81 82 $np $beamSec 2; element dispBeamColumn 164 82 83 $np $beamSec 2

element dispBeamColumn 165 83 84 $np $beamSec 2;element dispBeamColumn 166 84 85 $np $beamSec 2; element dispBeamColumn 167 86 87 $np $beamSec 2

element dispBeamColumn 168 87 88 $np $beamSec 2;element dispBeamColumn 169 82 87 $np $beamSec 2; element dispBeamColumn 170 83 86 $np $beamSec 2

Gravity loads applied at each corner node

Define gravity loads

pattern Plain 1 Linear {foreach node {9 17 25 33 41 49 57 65 73 } {load $node 0.0 0.0 -8.750 -20 0.0 }} pattern Plain 2 Linear {foreach node {10 18 26 34 42 50 58 66 74 } {load $node 0.0 0.0 -39.870 -59.7 -61.7 0.0}}

pattern Plain 3 Linear {foreach node {11 19 27 35 43 51 59 67 75 } {load $node 0.0 0.0 -49.150 -74.6 -39.6 0.0}}

pattern Plain 4 Linear {foreach node {12 20 28 36 44 52 60 68 76 } {load $node 0.0 0.0 -22.670

-30.19 30.23 0.0}}

pattern Plain 5 Linear {foreach node {13 21 29 37 45 53 61 69 77 } {load $node 0.0 0.0 30.19 30.23 0.0}} pattern Plain 6 Linear {foreach node {14 22 30 38 46 54 62 70 78 } {load $node 0.0 0.0 -49.150

74.57 -39.6 0.0}} pattern Plain 7 Linear {foreach node {15 23 31 39 47 55 63 71 79 } {load $node 0.0 0.0 -39.780

59.7 -59.3 0.0}} pattern Plain 9 Linear {foreach node {16 24 32 40 48 56 64 72 80 } {load $node 0.0 0.0 -8.750

20.31 -20.31 0.0}} pattern Plain 10 Linear {load 81 0.0 0.0 -8.750 -11.7 -11.7 0.0};pattern Plain 11 Linear {load

82 0.0 0.0 -20.140 -33.7 -27 0.0}

pattern Plain 12 Linear { load 83 0.0 0.0 -20.104 -33 -27.1 0.0};pattern Plain 13 Linear { load 84 0.0 0.0 -8.750 -11.7 11.7 0.0}

pattern Plain 14 Linear { load 85 0.0 0.0 -8.750 11.7 11.7 0.0};pattern Plain 15 Linear { load

86 0.0 0.0 -20.140 33 -27.1 0.0}

pattern Plain 16 Linear { load 87 0.0 0.0 -20.140 33 27.1 0.0};pattern Plain 17 Linear { load 88 0.0 0.0 -8.750 11.7 -11.7 0.0} # Start of analysis generation # Create the constraint handler constraints Transformation

Create the RCM Numberer numberer Plain

Create the system of equation storage and solver system BandGeneral

Create the convergent test

test NormDispIncr $tol $maxNumIter <$printFlag

test NormDispIncr 1.0e-5 1000 5 # create the solution algorithm

algorithm Newton

Create the time integration scheme

integrator LoadControl $dLambda1 <$Jd $minLambda $maxLambda> integrator LoadControl 0.1 # Create the static analysis

analysis Static initialize

analyze $numIncr <$dt> <$dtMin $dtMax $Jd>

analyze 1000 0.01

start of lateral load analysis

loadConst -time 0.0 # Define lateral loads

pattern Plain 18 Linear {foreach node {9 10 11 12} {load $node 0.565 0 0.0 0.0 0.0 0.0}} pattern Plain 19 Linear {foreach node{17 18 19 20} {load $node 1.1275 0 0.0 0.0 0.0 0.0}} pattern Plain 20 Linear {foreach node{25 26 27 28} {load $node 1.6925 0 0.0 0.0 0.0 0.0}} pattern Plain 21 Linear {foreach node{33 34 35 36} {load $node 2.2575 0 0.0 0.0 0.0 0.0}} pattern Plain 22 Linear {foreach node{41 42 43 44} {load $node 2.82 0 0.0 0.0 0.0 0.0}} pattern Plain 23 Linear {foreach node{49 50 51 52} {load $node 3.385 0 0.0 0.0 0.0 0.0}} pattern Plain 24 Linear {foreach node{57 58 59 60} {load $node 3.950 0 0.0 0.0 0.0 0.0}} pattern Plain 25 Linear {foreach node{65 66 67 68} {load $node 4.5125 0 0.0 0.0 0.0 0.0}} pattern Plain 26 Linear {foreach node{73 74 75 76} {load $node 5.0775 0 0.0 0.0 0.0 0.0}} pattern Plain 27 Linear {foreach node{81 82 83 84} {load $node 8.86 0 0.0 0.0 0.0 0.0}} # Start of recorder generation

recorder Node <-file $fileName> <-xml $fileName> <-time> <-node ($node1 $node2 ...)> <-nodeRange $startNode $endNode> <-region $RegionTag> <-node all> -dof ($dof1 $dof2 ...) $respType

```
recorder Node -file nodeDisp.out -time -node 9 10 11 12 17 18 19 20 25 26 27 28 33 34 35 36 41
42 43 44 49 50 51 52 57 58 59 60 65 66 67 68 73 74 75 76 81 82 83 84 -dof 1 2 6 disp
recorder Node -file Rnode.out -time -node 1 2 3 4 5 6 7 8 -dof 1 2 6 reaction
#recorder Drift -file $fileName <-time> -iNode ($inode1 $inode2 ...) -jNode($jnode1 $jnode2 ...)
- dof ($dof1 $dof2 ...) -perpDirn ($perpDirn1 $perpDirn 2 ...)
recorder Drift -file drift1.out -time -iNode 5 6 7 8 -jNode 13 14 15 16 -dof 1 2 -perpDirn 3
recorder Drift -file drift2.out -time -iNode 13 14 15 16 -jNode -perpDirn 3         21 22 23 24 -dof 1 2
recorder Drift -file drift3.out -time -iNode 21 22 23 24 –jNode 29 30 31 32 -dof 112 3
2 -perpDirn 3
recorder Drift -file drift4.out -time -iNode 29 30 31 32 -jNode 37 38 39 40 -dof 1
2-perpDirn 3
recorder Drift -file drift5.out -time -iNode 37 38 39 40 –jNode 45 46 47 48 -dof 1
2 -perpDirn 3
recorder Drift -file drift6.out -time -iNode 45 46 47 48 -jNode 53 54 55 56 –dof
1 2 – perpDirn
recorder Drift -file drift7.out -time -iNode 53 54 55 56 –jNode 61 62 63 64 -dof 1
2 –perpDirn
recorder Drift -file drift8.out -time -iNode 61 62 63 64 -jNode -perpDirn 3
recorder Drift -file drift9.out -time -iNode 69 70 71 72 -jNode 77 78 79 80 -dof 1
2 – perpDirn
recorder Drift -file drift10.out -time -iNode 77 78 79 80 –jNode 85 86 87 88 -dof
1 2 -perpDirn 3
# recorder Element <-file $fileName> <-time> <-ele ($ele1 $ele2 ...)> <-eleRange $startEle
$endEle> <-region $regTag> <-ele all> ($arg1 $arg2 ...)
recorder Element -file ele1global.out -time -ele 1 2 81 82 globalForce recorder Element -file
el1local.out -time -ele 1 2 81 82 localForce recorder Element -file ele1sec3Force.out -time -ele 1
2 81 82 section 4 force recorder Element -file ele1sec3Defo.out -time -ele 1 2 81 82 section 4
deformation recorder Element -file ele1sec2Stiff.out -time -ele 1 2 81 82 section 4 stiffness
recorder Element -file ele1sec2StressStrain.out -time -ele 1 2 81 82 section 4 fiber 0.25 0.125
1 stressStrain
integrator LoadControl 1.0 4 0.02 2.0 # Finally perform the analysis
```

```
# set some parameters
set maxU 0.3810; # Max displacement
set ok 0
set currentDisp 0.0 ;# perform the analysis;
while {$ok == 0 && $currentDisp < $maxU} { set ok [analyze 1]}; # if the analysis fails try initial tangent iteration;
if {$ok != 0} { puts "regular newton failed .. lets try an initial stiffness for this step"; test NormDispIncr 1.0e-5 1000; algorithm Newton -initial; set ok [analyze 1];if {$ok == 0} { puts "that worked .. back to regular newton"} ;test NormDispIncr 1.0e-12 1000; algorithm Newton
} if {$ok == 0} { puts "Pushover analysis completed SUCCESSFULLY"; } else { puts "Pushover analysis FAILED"; }
# Print out
print node
print ele
print sec
```

Appendix II-Soil structure modeling in opensees

units KN,m, sec

start of soil model generation

create modelBuilder with 2 dimension and 3 DOF/node

model BasicBuilder -ndm 2 -ndf 3; # Define the model builder, ndm=#dimension, ndf=#dofs # Foundation tag=1

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord $zcoord

node 1001 0 -1.5 0;node 100001 0 -1.5 0 ;node 1002 0 -1 0 ;node 100002 0 -1 0 ;node 1003 0 -0.5 0 ;node 100003 0 -0.5 0 ;node 1004 0 0 0 ;node 100004 0 0 0

node 1005 0 0.5 0 ;node 100005 0 0.5 0 0 ;node 1006 0 1 0 0 ;node 100006 0 1 0;node 1007 0 1.5 0 ;node 100007 0 1.5 0 0 ;node 100008 0 1.5 0;node 100009 0 1.5 0 0 equalDOF 1004 100004 1 2 3

Foundation tag=2

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord $zcoord

node 2001 0 -1.5 0;node 200001 0 -1.5 0;node 2002 0 -1 0;node 200002 0 -1 0;node 2003 0 -0.50;node 200003 0 -0.5 0;node 2004 0 0 0;node 200004 0 0 0;node 2005 0 0.5 0 ;node 200005 0 0.5 0 node 2006 0 1 0 ;node 200006 0 1 0 ;node 2007 0 1.5 0;node 200007 0 1.5 0;node 200008 0 1.5 0 node 200009 0 1.5 0 equalDOF 2004 200004 1 2 3

Foundation tag=3

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord $zcoord

node 3001 0 -1.5 0 ;node 300001 0 -1.5 0;node 3002 0 -1 0;node 300002 0 -1 0 ;node 3003 0
-0.5 0 node 300003 0 -0.5 0 0;node 3004 0 0 0;node 300004 0 0 0;node 3005 0 0.5 0 ;node 300005 0 0.5 0 node 3006 0 1 0;node 300006 0 1 0;node 3007 0 1.5 0 ;node 300007 0 1.5 0 ;node 300008 0 1.5 0 node 300009 0 1.5 0 equalDOF 3004 300004 1 2 3

Foundation tag=4

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord $zcoord

node 4001 0 -1.5 0;node 400001 0 -1.5 0 ;node 4002 0 -1 0;node 400002 0 -1 0 ;node 4003 0-0.5 0;node 400003 0 -0.5 0;node 4004 0 0 0;node 400004 0 0 0;node 4005 0 0.5 0;node 400005 0 0.5 0; node 4006 0 1 0;node 400006 0 1 0;node 4007 0 1.5 0;node 400007 0 1.5 0;node 400008 0 1.5 0;node 400009 01.5 0;equalDOF 4004 400004 1 2 3;# Foundation tag=5

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord

node 5001 0 -1.5 0;node 500001 0 -1.5 0 ;node 5002 0 -1 0 0;node 500002 0 -1 0 ;node 5003 0-0.5 0;node 500003 0 -0.5 0;node 5004 0 0 0;node 500004 0 0 0;node 5005 0 0.5 0 ;node 500005 0 0.5 0 ; node 5006 0 1 0;node 500006 0 1 0 ;node 5007 0 1.5 0;node 500007 0 1.5 0;node 500008 0 1.5 0;node 500009 0;1.5 0 equalDOF 5004 500004 1 2 3

Foundation tag=6

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord $zcoord

node6001 0 -1.5 0;node 600001 0 -1.5 0;node 6002 0 -1 0;node 600002 0 -1 0;node 6003 0 -0.5 0;node600003 0 -0.5 0;node 6004 0 0 0;node600004 0 0 0;node 6005 0 0.5 0;node 600005 0 0.5 0;node 6006 0 1 0 ;node 600006 0 10 ; node 6007 0 1.5 0 ;node 600007 0 1.5 0 node 600008 0 1.5 0;node 600009 0 1.5 0;equalDOF 6004 600004 1 2 3

Foundation tag=7

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord $zcoord

node 7001 0 -1.5 0;node 700001 0 -1.5 0;node 7002 0 -1 0;node 700002 0 -1 0;node 7003 0 -0.5 0;node 700003 0 -0.5 0;node 7004 0 0 0;node 700004 0 0 0;node 7005 0 0.5 0 ;node 700005 0 0.5 0;node 7006 0 1 0 ;node 700006 0 1 0; ;node 7007 0 1.5 0 ;node 700007 0 1.5 0;node 700008 0 1.5 0 node 700009 0 1.5 0 equalDOF 7004 700004 1 2 3

Foundation tag=8

Foundation Base Condation Tag=5

node $nodeTag $xcoord $ycoord $zcoord

node 8001 0 -1.5 0 ;node 800001 0 -1.5 0 ;node 8002 0 -1 0 ;node 800002 0 -1 0 ;node 8003 0 -0.5 0 node 800003 0 -0.5 0;node 8004 0 0 0;node 800004 0 0 0;node 8005 0 0.5 0 0;node 800005 0 0.5 0 node 8006 0 1 0;node 800006 0 1 0 ;node 8007 0 1.5 0 ;node 800007 0 1.5 0 ;node 800008 0 1.5 0;;node 800009 0 1.5 0;equalDOF 8004 800004 1 2 3

fix $nodeTag (ndf $ConstrValues)

#foreach varName1 list1 <varName2 list2 …> {body}

fix $nodeTag (ndf $ConstrValues)

fixity of foundation -1

foreach node { 100001 100002 100003 100004 100005 100006 100007 100008 100009} {fix $node 1 1 1 } set endFootNodeL_1 1001 ;set endSprNodeR_1 1007 ;set endSprEleL_1 100001 set endSprEleR_1 100007;set midSprEle_1 100004

fixity of Foundation-2

foreach node {200001 200002 200003 200004 200005 200006 200007 200008 200009} {fix $node 1 1 1 } set endFootNodeL_2 2001;set endSprNodeR_2 2007 ;set endSprEleL_2 200001;set endSprEleR_2 200007 ;set midSprEle_2 200004

equalDof $rNodeTag $cNodeTag $dof1 $dof2 $dof3

foreach node {300001 300002 300003 300004 300005 300006 300007 300008 300009} {fix $node 1 1 1 } set endFootNodeL_3 3001;set endSprNodeR_3 3007 ;set endSprEleL_3 300001 ;set endSprEleR_3 300007 ;set midSprEle_3 300004

fixity of Foundation-4

foreach node {400001 400002 400003 400004 400005 400006 400007 400008 400009} { fix $node 1 1 1 } set endFootNodeL_4 4001 ;set endSprNodeR_4 4007;set endSprEleL_4 400001 ;set endSprEleR_4 400007 ;set midSprEle_4 400004

fixity-5

foreach node {500001 500002 500003 500004 500005 500006 500007 500008 500009} { fix $node 1 1 1 } set endFootNodeL_5 5001 ;set endSprNodeR_5 5007;set endSprEleL_5 500001;set endSprEleR_5 500007 ;set midSprEle_5 500004

fixity-6

foreach node {600001 600002 600003 600004 600005 600006 600007 600008 600009 } {fix $node 1 1 1 } set endFootNodeL_6 6001 ;set endSprNodeR_6 6007;set endSprEleL_6 600001;set endSprEleR_6 600007 ;set midSprEle_6 600004

fixity-7

foreach node {700001 700002 700003 700004 700005 700006 700007 700008 700009} { fix $node 1 1 1} setendFootNodeL_7 7001;set endSprNodeR_7 7007;set endSprEleL_7 700001 ;set endSprEleR_7 700007; setmidSprEle_7 700004

fixity-8

foreachnode {800001 800002 800003 800004 800005 800006 800007 800008 800009} { fix $node 1 1 11}

set endFootNodeL_8 8001 ;set endSprNodeR_8 8007;set endSprEleL_8 800001 ;set endSprEleR_8 800007 ;set midSprEle_8 800004

Material for shallow Foundation-1

#uniaxialMaterial PySimple1 $matTag $soilType $ pult $Y50 $ Cd <$c>.

uniaxialMaterial QzSimple2 101 1 41840 0.002770946 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 102 1 83680 0.002770946 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 103 1 251040 0.00541893 0.1 0.05

#uniaxialMaterial PzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 105 1 208000 0.0197215 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 106 1 100000 0.00948148 0.1 0.05

Material for shallow Foundation-2

#uniaxialMaterial QzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 201 1 127872 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 202 1 155744 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 203 1 167232 0.004512320 0.1 0.05

#uniaxialMaterial PySimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 205 1 208000 0.0197215 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $soilType $ Qult -end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 206 1 100000 0.00948148 0.1 0.05

Material for shallow Foundation-3

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 301 1 127872 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 302 1 155744 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 303 1 167232 0.004512320 0.1 0.05

#uniaxialMaterial PySimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 305 1 208000 0.0197215 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 306 1 100000 0.00948148 0.1 0.05

Material for shallow Foundation-4

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 401 1 127872 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 402 1 155744 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 403 1 167232 0.004512320 0.1 0.05

#uniaxialMaterial PySimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 405 1 208000 0.0197215 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 406 1 100000 0.00948148 0.1 0.05

Material for shallow Foundation

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 501 1 127872 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil>uniaxialMaterial QzSimple2 502 1 155744 0.002256160 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil>uniaxialMaterial QzSimple2 503 1 167232 0.004512320 0.1 0.05

#uniaxialMaterial PySimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 505 1 208000 0.0197215 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 506 1 100000 0.00948148 0.1 0.05

Material for shallow Foundation
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 601 1 127872 0.002256160 0.1 0.05
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 602 1 155744 0.002256160 0.1 0.05
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 603 1 167232 0.004512320 0.1 0.05
#uniaxialMaterial PySimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 605 1 208000 0.0197215 0.1 0.05
#uniaxialMaterial TzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 606 1 100000 0.00948148 0.1 0.05
Material for shallow Foundation
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 701 1 127872 0.002256160 0.1 0.05
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 702 1 155744 0.002256160 0.1 0.05
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 703 1 167232 0.004512320 0.1 0.05
#uniaxialMaterial PySimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 705 1 208000 0.0197215 0.1 0.05
#uniaxialMaterial TzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 706 1 100000 0.00948148 0.1 0.05
Material for shallow Foundation-8
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 801 1 127872 0.002256160 0.1 0.05
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 802 1 155744 0.002256160 0.1 0.05
#uniaxialMaterial QzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial QzSimple2 803 1 167232 0.004512320 0.1 0.05
#uniaxialMaterial PySimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial PySimple2 805 1 208000 0.0197215 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $soilType $Qult-end-extreme $z-50-end <TpSoil><CradSoil> uniaxialMaterial TzSimple2 806 1 100000 0.00948148 0.1 0.05

Foundation Element connectivity

Define elastic section for Footing

set Af [expr 3*3];set Jf [expr 0.141*3*pow(3,3)];set Iy [expr 3*pow(3,3)/12] set Iz [expr 3*pow(3,3)/12];set Ec 29000;set Gc [expr $Ec/2.4]

Vertical spring element connectivity

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 100001 100001 1001 -mat 101 -dir 2;element zeroLength 100002 100002 1002 – mat-dir 2 element zeroLength 100003 100003 1003 -mat 103 -dir 2;element zeroLength 100004 100004 1004 – mat-dir 2 element zeroLength 100005 100005 1005 -mat 103 -dir 2;element zeroLength 100006 100006 1006 - mat

102 -dir 2 element zeroLength 100007 100007 1007 -mat 101 -dir 2 # Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir

element zeroLength 100008 1007 100008 -mat 105 -dir 1;element zeroLength 100009 1007 100009 -mat106-dir 1

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj

section Elastic $secTag $E $A $Iz <$Iy $G $J>set secTag 6

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

geomTransf Linear 10

Foundation element connectivity

element elasticBeamColumn $eleTag $iNode $jNode $A $E $G $J $Iy $Iz $transfTag

element elasticBeamColumn 1001 1001 1002 $Af EcIz10; element elasticBeamColumn 1002 1002

1003 $Af $Ec $Iz 10

element elasticBeamColumn 1003 1003 1004 $Af EcIz10;element elasticBeamColumn 1004 1004

1005 $Af $Ec $Iz 10

element elasticBeamColumn 1005 1005 1006 $Af $Ec $Iz10;element elasticBeamColumn 1006 1006

1007 $Af $Ec $Iz 10

Vertical spring elemenet connectivity

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 200001 200001 2001 -mat 201 -dir 2;element zeroLength 200002 200002 2002 – mat-dir 2 element zeroLength 200003 200003 2003 -mat 203 -dir 2;element zeroLength 200004 200004 2004 – mat-dir 2 element zeroLength 200005 200005 2005 -mat 203 -dir 2;element zeroLength 200006 200006 2006 – mat202 -dir 2 element zeroLength 200007 200007 2007 -mat 201 -dir 2

Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir

element zeroLength 200008 2007 200008 -mat 205 -dir 1;element zeroLength 200009 2007 200009 -mat 206 -dir 1

section Elastic $secTag $E $A $Iz <$Iy $G $J>set secTag 7

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj geomTransf Linear 20

Foundation connectivity

element elasticBeamColumn $eleTag $iNode $jNode $A $E $G $J $Iy $Iz $transfTag

element elasticBeamColumn 2001 2001 2002 $Af $Ec 2003$Iz 20;

$Af $Ec $Iz 20 element elasticBeamColumn 2002 2002

element elasticBeamColumn 2003 2003 2004 $Af $Ec 2005$Iz 20 ;

$Af $Ec $Iz 20 element elasticBeamColumn 2004 2004

element elasticBeamColumn 2005 2005 2006 $Af $Ec $Iz 20 ;

element elasticBeamColumn 2006 1006

2007 $Af $Ec $Iz 20

Vertical spring element connectivity

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 300001 300001 3001 -mat 301 -dir 2;element zeroLength 300002 300002 3002 – mat-dir 2 element zeroLength 300003 300003 3003 -mat 303 -dir 2;element zeroLength 300004 300004 3004 - mat-dir 2 element zeroLength 300005 300005 3005 -mat 303 -dir 2;element zeroLength 300006 300006 3006 - mat

302 -dir 2 element zeroLength 300007 300007 3007 -mat 301 -dir2;

Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir

element zeroLength 300008 3007 300008 -mat 305 -dir 1;element zeroLength 300009 3007 300009 -mat306-dir 1

section Elastic $secTag $E $A $Iz <$Iy $G $J>setsecTag 8

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj geomTransf Linear 30

Foundation connectivity

element elasticBeamcolumn $eleTag $iNode $jNode $A $E Iz $trnsfTag

element elasticBeamColumn 3001 3001 3002 3003$Af $Ec $Iz 30 ;element elasticBeamColumn 3002 3002 $Af $Ec $Iz 30

element elasticBeamColumn 3003 3003 3004 3005$Af $Ec $Iz 30;element elasticBeamColumn 3004 3004 $Af $Ec $Iz 30

element elasticBeamColumn 3005 3005 3006 $Af $Ec $Iz 30;element elasticBeamColumn 3006 3006 3007 $Af $Ec $Iz 30

Vertical spring elemenet connectivity of Foundation-4

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 400001 400001 4001 -mat 401 -dir 2;element zeroLength 400002 400002 4002 – mat-dir 2 element zeroLength 400003 400003 4003 -mat 403 -dir 2;element zeroLength 400004 400004 4004 – mat-dir 2 element zeroLength 400005 400005 4005 -mat 403 -dir 2;element zeroLength 400006 400006 4006 - mat

402 -dir 2 element zeroLength 400007 400007 4007 -mat 401 dir2

Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir

element zeroLength 400008 4007 400008 -mat 405 -dir 1;element zeroLength 400009 4007 400009 -mat 406 -dir 1

section Elastic $secTag $E $A $Iz <$Iy $G $J>set secTag 9

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj geomTransf Linear 40

Foundation connectivity

element elasticBeamcolumn $eleTag $iNode $jNode $A $E Iz $trnsfTag

element elasticBeamColumn 4001 4001 4002 $Af $Ec 4003$Iz 40;element elasticBeamColumn 4002 4002 $Af $Ec $Iz 40 element elasticBeamColumn 4003 4003 4004 $Af $Ec 4005$Iz 40 ;element elasticBeamColumn 4004 4004 $Af $Ec $Iz 40

element elasticBeamColumn 4005 4005 4006 $Af $Ec $Iz 40;element elasticBeamColumn 4006 4006 4007 $Af $Ec $Iz 40

Vertical spring elemenet connectivity -5

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 500001 500001 5001 -mat 501 -dir 2;element zeroLength 500002 500002 5002 – mat-dir 2 element zeroLength 500003 500003 5003 -mat 503 -dir 2;element zeroLength 500004 500004 5004 – mat-dir 2 element zeroLength 500005 500005 5005 -mat 503 -dir 2;element zeroLength 500006 500006 5006 – mat502 -dir 2 element zeroLength 500007 500007 5007 -mat 501 -dir 2;

Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir

element zeroLength 500008 5007 500008 -mat 505 -dir 1;element zeroLength 500009 5007 500009

-mat 506 -dir 1

section Elastic $secTag $E $A $Iz <$Iy $G $J>setsecTag 10

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj geomTransf Linear 50

Foundation connectivity

element elasticBeamcolumn $eleTag $iNode $jNode $A $E Iz $trnsfTag

element elasticBeamColumn 5001 5001 5002 $Af $Ec 5003$Iz 50;element elasticBeamColumn 5002 5002 $Af $Ec $Iz 50

element elasticBeamColumn 5003 5003 5004 $Af $Ec 5005$Iz 50 ;element elasticBeamColumn 5004 5004 $Af $Ec $Iz 50

element elasticBeamColumn 5005 5005 5006 $Af $Ec $Iz 50 ;element elasticBeamColumn 5006 5006 5007 $Af $Ec $Iz 50

Vertical spring elemenet connectivity

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 600001 600001 6001 -mat 601 -dir 2;element zeroLength 600002 600002 6002 – mat-dir 2 element zeroLength 600003 600003 6003 -mat 603 -dir 2;element zeroLength 600004 600004 6004 – mat-

dir 2 element zeroLength 600005 600005 6005 -mat 603 -dir 2;element zeroLength 600006 600006 6006 - mat

602 -dir 2 element zeroLength 600007 600007 6007 -mat 601 -dir2

Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir

element zeroLength 600008 6007 600008 -mat 605 -dir 1;element zeroLength 600009 6007 600009 -mat 606 -dir 1

section Elastic $secTag $E $A $Iz <$Iy $G $J>set secTag 11

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj geomTransf Linear 60

Foundation connectivity

element elasticBeamcolumn $eleTag $iNode $jNode $A $E Iz $trnsfTag

element elasticBeamColumn 6001 6001 6002 $Af $Ec $Iz 60;element elasticBeamColumn 6002 6002

6003 $Af $Ec $Iz 60

element elasticBeamColumn 6003 6003 6004 $Af $Ec $Iz 60;element elasticBeamColumn 6004 6004

6005 $Af $Ec $Iz 60

element elasticBeamColumn 6005 6005 6006 $Af $Ec $Iz 60;element elasticBeamColumn 6006 6006 6007 $Af $Ec $Iz 60

Vertical spring elemenet connectivity

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 700001 700001 7001 -mat 701 -dir 2;element zeroLength 700002 700002 7002 – mat-dir 2 element zeroLength 700003 700003 7003 -mat 703 -dir 2;element zeroLength 700004 700004 7004 – mat-dir 2 element zeroLength 700005 700005 7005 -mat 703 -dir 2;element zeroLength 700006 700006 7006 – mat702 -dir 2 element zeroLength 700007 700007 7007 -mat 701 -dir2

Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir

element zeroLength 700008 7007 700008 -mat 705 -dir 1;element zeroLength 700009 7007 700009 -mat 706 -dir 1

section Elastic $secTag $E $A $Iz <$Iy $G $J>set secTag 12

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj geomTransf Linear 70

element elasticBeamcolumn $eleTag $iNode $jNode $A $E Iz $trnsfTag

element elasticBeamColumn 7001 7001 7002 $Af $Ec 7003$Iz 70;element elasticBeamColumn 7002 7002 $Af $Ec $Iz 70

element elasticBeamColumn 7003 7003 7004 $Af $Ec 7005$Iz 70 ;element elasticBeamColumn 7004 7004 $Af $Ec $Iz 70

element elasticBeamColumn 7005 7005 7006 $Af $Ec $Iz 70;element elasticBeamColumn 7006 7006 7007 $Af $Ec $Iz 70

Vertical spring elemenet connectivity

element zeroLength $eleTag iNode $jNode -mat$matTag -dir $dir element zeroLength 800001 800001 8001 -mat 801 -dir 2;element zeroLength 800002 800002 8002 – mat-dir 2 element zeroLength 800003 800003 8003 -mat 803 -dir 2;element zeroLength 800004 800004 8004 – mat-dir 2 element zeroLength 800005 800005 8005 -mat 803 -dir 2;element zeroLength 800006 800006 8006 - mat

802 -dir 2 element zeroLength 800007 800007 8007 -mat 801 -dir2

Horizontal spring element connectivity

element zeroLength $eleTag $Node$jNode -mat$matTag -dir $dir element zeroLength 800008 8007 800008 -mat 805 -dir 1 element zeroLength 800009 8007 800009 -mat 806 -dir 1 # section Elastic $secTag $E $A $Iz <$Iy $G $J>set secTag 13

section Elastic $secTag $Ec $Af $Iz $Iy $Gc $Jf

GoemetricTransf Linear $trasTag <-jntOffset $dXi $dYi $dXj $dYj geomTransf Linear 80

Foundation connectivity

element elasticBeamcolumn $eleTag $iNode $jNode $A $E Iz $trnsfTag

element elasticBeamColumn 8001 8001 8002 $Af $Ec 8003$Iz 80 ;element elasticBeamColumn 8002 8002 $Af $Ec $Iz 80

element elasticBeamColumn 8003 8003 8004 $Af $Ec 8005$Iz 80;element elasticBeamColumn 8004 8004 $Af $Ec $Iz 80

element elasticBeamColumn 8005 8005 8006 $Af $Ec $Iz 80;element elasticBeamColumn 8006 8006 8007 $Af $Ec $Iz 80

gravity load (weight of soil) puts "soil Modeling is Finished..."

units# units KN,m, sec

start of model generation

create modelBuilder with 3 dimension and 6 DOF/node

model BasicBuilder -ndm 3 -ndf 6

Define Geometry

set parameters for model geometry set h 3;# storey height set bx 4;# bay width in x- direction set by 4;# bay width in y- direction

Create nodes

tag x y z

node 1 0 0 0;node 2 [expr $bx] 0 0;node 3 [expr $bx*2] 0 0;node 4 [expr $bx*3] 0 0;node 5 [expr $bx*3] [expr $by] 0;node 6 [expr $bx*2] [expr $by] 0

node 7 [expr $bx*1] [expr $by] 0;node 8 0 [expr $by] 0;node 9 0 0 [expr $h];node 10 [expr $bx] 0 [expr $h];node 11 [expr $bx*2] 0 [expr $h]

node 12 [expr $bx*3] 0 [expr $h];node 13 [expr $bx*3] [expr $by] [expr $h];node 14 [expr $bx*2] [expr$by] [expr $h];node 15 [expr $bx*1] [expr $by] [expr $h]

node160[expr $by] [expr $h];node 17 0 0 [expr $h*2];node 18 [expr $bx] 0 [expr $h*2];node19 [expr$bx*2] 0 [expr $h*2];node 20 [expr $bx*3] 0 [expr $h*2]

node21[expr $bx*3] [expr $by] [expr $h*2];node 22 [expr $bx*2] [expr $by] [expr $h*2];node23 [expr $bx*1] [expr $by] [expr $h*2];node 24 0 [expr $by] [expr $h*2] node 25 0 0 [expr $h*3];node 26 [expr $bx] 0 [expr $h*3];node 27 [expr $bx*2] 0 [expr $h*3];node

28 [expr $bx*3] 0 [expr $h*3] node 29 [expr $bx*3] [expr $by] [expr $h*3];node 30 [expr $bx*2] [expr $by] [expr $h*3];node 31[expr $bx*1] [expr $by] [expr $h*3]node 32 0 [expr $by] [expr $h*3];node 33 0 0 [expr $h*4];node 34 [expr $bx] 0 [expr $h*4];node35 [expr $bx*2] 0 [expr $h*4]node 36 [expr $bx*3] 0 [expr $h*4];node 37 [expr $bx*3] [expr $by] [expr $h*4];node 38 [expr $bx*2] [expr $by] [expr $h*4]

node 39 [expr $bx*1] [expr $by] [expr $h*4];node 40 0 [expr $by] [expr $h*4];node 41 0 0 [expr $h*5];node 42 [expr $bx] 0 [expr $h*5] node 43 [expr $bx*2] 0 [expr $h*5];node 44 [expr $bx*3] 0 [expr $h*5];node 45 [expr $bx*3] [expr$by] [expr $h*5] node 46 [expr $bx*2] [expr $by] [expr $h*5];node 47 [expr $bx*1] [expr $by] [expr $h*5];node 480 [expr $by] [expr $h*5] node 49 0 0 [expr $h*6];node 50 [expr $bx] 0 [expr $h*6];node 51 [expr $bx*2] 0 [expr $h*6];node52 [expr

$bx*3] 0 [expr $h*6] node 53 [expr $bx*3] [expr $by] [expr $h*6];node 54 [expr $bx*2] [expr $by] [expr $h*6];node 55[expr $bx*1] [expr $by] [expr $h*6]

node 56 0 [expr $by] [expr $h*6];node 57 0 0 [expr $h*7] ;node 58 [expr $bx] 0 [expr $h*7];node 59 [expr $bx*2] 0 [expr $h*7]node 60 [expr $bx*3] 0 [expr $h*7];node 61 [expr $bx*3] [expr $by] [expr $h*7];node 62 [expr $bx*2] [expr $by] [expr $h*7]

node 63 [expr $bx*1] [expr $by] [expr $h*7];node 64 0 [expr $by] [expr $h*7];node 65 0 0 [expr $h*8];node 66 [expr $bx] 0 [expr $h*8]

node 67 [expr $bx*2] 0 [expr $h*8];node 68 [expr $bx*3] 0 [expr $h*8];node 69 [expr $bx*3] [expr $by] [expr $h*8]

node 70 [expr $bx*2] [expr $by] [expr $h*8];node 71 [expr $bx*1] [expr $by] [expr $h*8];node 72 0 [expr $by] [expr $h*8]

node 73 0 0 [expr $h*9];node 74 [expr $bx] 0 [expr $h*9];node 75 [expr $bx*2] 0 [expr $h*9]; node 76 [expr $bx*3] 0 [expr $h*9]

node 77 [expr $bx*3] [expr $by] [expr $h*9];node 78 [expr $bx*2] [expr $by] [expr $h*9];node 79 [expr $bx*1] [expr $by] [expr $h*9]

node 80 0 [expr $by] [expr $h*9];node 81 0 0 [expr $h*10];node 82 [expr $bx] 0 [expr $h*10]; node 83 [expr $bx*2] 0 [expr $h*10]

node 84 [expr $bx*3] 0 [expr $h*10];node 85 [expr $bx*3] [expr $by] [expr $h*10];node 86 [expr $bx*2] [expr $by] [expr $h*10]

node 87 [expr $bx*1] [expr $by] [expr $h*10];node 88 0 [expr $by] [expr $h*10]

set base constraints(boundery conditions)

tag Dx Dy Dz Rx Ry Rz

fix $nodeTag (ndf $ConstrValues)

foreach node {1 2 3 4 5 6 7 8} {fix $node 1 1 1 1 1 1 }

foreach node {9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68

69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 } {fix $node 0 0 0 0 0 0}

set frame constraints

tag Dx Dy Dz Rx Ry Rz

fix $nodeTag (ndf $ConstrValues)

Fixity b/n soil modeling and structure

equalDof $rNodeTag $cNodeTag $dof1 $dof2 $dof3

equalDOF1 1004 1 2 3 equalDOF2 2004 1 2 3 equalDOF3 3004 1 2 3 equalDOF4 4004 1 2 3 equalDOF 5 5004 1 2 3 equalDOF 6 6004 1 2 3 equalDOF 7 7004 1 2 3 equalDOF 8 8004 1 2 3

MATERIAL parameters ---

set IDconcCore 1;# material ID tag -- confined core concrete set IDconcCover 2;# material ID tag -- unconfined cover concrete set IDreinf 3;# material ID tag -- reinforcement

nominal concrete compressive strength

set fc 25000;# CONCRETE Compressive Strength,(+Tension, -Compression) set Ec 29000000;# Concrete Elastic Modulus

set GC [expr $Ec/2.4] # confined concrete

set Kfc 1.3;# ratio of confined to unconfined concrete strength set fc1C [expr $Kfc*$fc];# CONFINED concrete (mander model), maximum stress set eps1C [expr 2.*$fc1C/$Ec];# strain at maximum stress set fc2C [expr 0.2*$fc1C];# ultimate stress

set eps2C [expr 5*$eps1C];# strain at ultimate stress

unconfined concrete

set fc1U $fc;# UNCONFINED concrete (todeschini parabolic model), maximum stress set eps1U -0.003;# strain at maximum strength of unconfined concrete

set fc2U [expr 0.2*$fc1U]; # ultimate stress set eps2U -0.01;# strain at ultimate stress

set lambda 0.1;# ratio between unloading slope at $eps2 and initial slope $Ec

tensile-strength properties set ftC [expr -0.14*$fc1C];# tensile strength +tension set ftU [expr -0.14*$fc1U];# tensile strength +tension set Ets [expr $ftU/0.002];# tension softening stiffness# ---set Fy 260870;# STEEL yield stress set Es 200000000;# modulus of steel set Bs 0.01;# strain-hardening ratio set R0 18;# control the transition from elastic to plastic branches set cR1 0.925;# control the transition from elastic to plastic branches set cR2 0.15;# control the transition from elastic to plastic branches

uniaxialMaterial Concrete02 $IDconcCore $fc1C $eps1C $fc2C $eps2C $lambda $ftC $Ets;# build core concrete (confined)

uniaxialMaterial Concrete02 $IDconcCover $fc1U $eps1U $fc2U $eps2U $lambda $ftU $Ets;# build cover concrete (unconfined)

uniaxialMaterial Steel02 $IDreinf $Fy $Es $Bs $R0 $cR1 $cR2;# build reinforcement material

section GEOMETRY --set HSec 0.5;# Column Depth set BSec 0.4;# Column Width set coverSec 0.025; # Column cover to reinforcing steel NA. set numBarsSec 6;# number of longitudinal-reinforcement bars in steel layer. (symmetric top & bot)

set barAreaSec [expr 0.25*3.14*0.4];# area of longitudinal-reinforcement bars(diam.=20mm) set SecTag 1;# set tag for symmetric section

FIBER SECTION properties ---# column section:

setcoverx [expr $HSec/2.0]; # The distance from the section y-axis to the edge of the cover concrete -- outer edge of cover concrete

setcovery [expr $BSec/2.0]; # The distance from the section x-axis to the edge of the cover concrete -- outer edge of cover concrete

set corex [expr $coverx-$coverSec]; # The distance from the section y-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete

set corey [expr $covery-$coverSec]; # The distance from the section x-axis to the edge of

the core concrete -- edge of the core concrete/inner edge of cover concrete set nfCorex 30;# number of fibers for concrete in x-direction -- core concrete set nfCorey 24;# number of fibers for concrete in y-direction

set nfCoverx 30;# number of fibers for concrete in x-direction -- cover concrete set nfCovery 24;# number of fibers for concrete in y-direction

Define the fiber section (define the core patch,four cover patchs and Reinforcement layer respectively)

section fiberSec $SecTag {patch quadr $IDconcCore $nfCorey $nfCorex -$corex $corey -$corex $corey $corex -$corey $corex $corey patch quadr $IDconcCover 1 $nfCoverx -$coverx $covery -$corex $corey $corex $corey $coverx $covery patch quadr $IDconcCover 1 $nfCoverx -$corex $corey -$coverx -$covery $coverx -$covery $corex -$corey patch quadr $IDconcCover $nfCovery 1 -$coverx $covery -$coverx -$covery -$corex -$corey -$corex $corey patch quadr $IDconcCover $nfCovery 1 $corex $corey $corex -$corey $coverx -$covery $coverx $covery layer straight $IDreinf $numBarsSec $barAreaSec $corex $corey $corex -$corey layer straight $IDreinf $numBarsSec $barAreaSec -$corex $corey -$corex -$corey }

assign torsional Stiffness for 3D Model set SecTagTorsion 98;# ID tag for torsional section behavior set SecTag3D 3; # ID tag for combined behavior for 3D model set GJ [expr $GC *0.1685*.50* 1/3*pow(0.4,3)];# torsional stiffness

uniaxialMaterial Elastic $SecTagTorsion $GJ;# define elastic torsional stiffness

section Aggregator $SecTag3D $SecTagTorsion T -section $SecTag; # combine section properties

number of column integration points (sections)

ID tag for column transformation, defining element normal set IDcolTransf 1

#geomTransf PDelta $transfTag $vecxzX $vecxzY $vecxzZ <-jntOffset $dXi $dYi $dZi $dXj $dYj $dZj> geomTransf PDelta $IDcolTransf 1 0 0 set colSec 3

set np 4

Create the non-linear column elements' connectivity

element dispBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens> element dispBeamColumn 1 1 9 $np $colSec 1;element dispBeamColumn 2 2 10 $np $colSec 1;element dispBeamColumn 3 3 11 $np $colSec 1

element dispBeamColumn 4 4 12 $np $colSec 1;element dispBeamColumn 5 5 13 $np $colSec 1;element dispBeamColumn 6 6 14 $np $colSec 1

element dispBeamColumn 7 7 15 $np $colSec 1;element dispBeamColumn 8 8 16 $np $colSec 1;element dispBeamColumn 9 9 17 $np $colSec 1

element dispBeamColumn 10 10 18 $np $colSec 1;element dispBeamColumn 11 11 19 $np $colSec 1; element dispBeamColumn 12 12 20 $np $colSec 1

element dispBeamColumn 13 13 21 $np $colSec 1;element dispBeamColumn 14 14 22 $np $colSec 1; element dispBeamColumn 15 15 23 $np $colSec 1

element dispBeamColumn 16 16 24 $np $colSec 1;element dispBeamColumn 17 17 25 $np $colSec 1; element dispBeamColumn 18 18 26 $np $colSec 1

element dispBeamColumn 19 19 27 $np $colSec 1;element dispBeamColumn 20 20 28 $np $colSec 1; element dispBeamColumn 21 21 29 $np $colSec 1

element dispBeamColumn 22 22 30 $np $colSec 1;element dispBeamColumn 23 23 31 $np $colSec 1; element dispBeamColumn 24 24 32 $np $colSec 1

element dispBeamColumn 25 25 33 $np $colSec 1;element dispBeamColumn 26 26 34 $np $colSec 1; element dispBeamColumn 27 27 35 $np $colSec 1

element dispBeamColumn 28 28 36 $np $colSec 1;element dispBeamColumn 29 29 37 $np $colSec1; element dispBeamColumn 30 30 38 $np $colSec 1

element dispBeamColumn 31 31 39 $np $colSec 1;element dispBeamColumn 32 32 40 $np $colSec1; element dispBeamColumn 33 33 41 $np $colSec 1

element dispBeamColumn 34 34 42 $np $colSec 1;element dispBeamColumn 35 35 43 $np $colSec 1; element dispBeamColumn 36 36 44 $np $colSec 1

element dispBeamColumn 37 37 45 $np $colSec 1;element dispBeamColumn 38 38 46 $np $colSec 1; element dispBeamColumn 39 39 47 $np $colSec 1

element dispBeamColumn 40 40 48 $np $colSec 1;element dispBeamColumn 41 41 49 $np $colSec 1; element dispBeamColumn 42 42 50 $np $colSec 1

element dispBeamColumn 43 43 51 $np $colSec 1;element dispBeamColumn 44 44 52 $np $colSec 1; element dispBeamColumn 45 45 53 $np $colSec 1

element dispBeamColumn 46 46 54 $np $colSec 1;element dispBeamColumn 47 47 55 $np $colSec 1; element dispBeamColumn 48 48 56 $np $colSec 1

element dispBeamColumn 49 49 57 $np $colSec 1;element dispBeamColumn 50 50 58 $np $colSec 1; element dispBeamColumn 51 51 59 $np $colSec 1

element dispBeamColumn 52 52 60 $np $colSec 1;element dispBeamColumn 53 53 61 $np $colSec 1; element dispBeamColumn 54 54 62 $np $colSec 1

element dispBeamColumn 55 55 63 $np $colSec 1;element dispBeamColumn 56 56 64 $np $colSec 1; element dispBeamColumn 57 57 65 $np $colSec 1

element dispBeamColumn 58 58 66 $np $colSec 1;element dispBeamColumn 59 59 67 $np $colSec 1; element dispBeamColumn 60 60 68 $np $colSec 1

element dispBeamColumn 61 61 69 $np $colSec 1;element dispBeamColumn 62 62 70 $np $colSec 1; element dispBeamColumn 63 63 71 $np $colSec 1

element dispBeamColumn 64 64 72 $np $colSec 1;element dispBeamColumn 65 65 73 $np $colSec 1; element dispBeamColumn 66 66 74 $np $colSec 1

element dispBeamColumn 67 67 75 $np $colSec 1;element dispBeamColumn 68 68 76 $np $colSec 1; element dispBeamColumn 69 69 77 $np $colSec 1

element dispBeamColumn 70 70 78 $np $colSec 1;element dispBeamColumn 71 71 79 $np $colSec 1; element dispBeamColumn 72 72 80 $np $colSec 1

element dispBeamColumn 73 73 81 $np $colSec 1;element dispBeamColumn 74 74 82 $np $colSec 1; element dispBeamColumn 75 75 83 $np $colSec 1

element dispBeamColumn 76 76 84 $np $colSec 1;element dispBeamColumn 77 77 85 $np $colSec 1; element dispBeamColumn 78 78 86 $np $colSec 1

element dispBeamColumn 79 79 87 $np $colSec 1;element dispBeamColumn 80 80 88 $np $colSec 1

define beam element # section GEOMETRY of beam set d 0.5 ;# bema Depth set b 0.25; # beam Width set coverSec1 0.025;# beam cover to reinforcing steel NA.

set numBarsSec 4;# number of longitudinal-reinforcement bars in steel layer. (symmetric top & bot)

set barAreaSec [expr 3.14*0.25*0.196]; # area of longitudinal-reinforcement bars(diam=14mm)

set SecTag 4;# set tag for symmetric section #Beam section: set coverx1 [expr $d/2.0]; # The distance from the section z-axis to the edge of the cover concrete -- outer edge of cover concrete

set covery1 [expr $b/2.0]; # The distance from the section y-axis to the edge of the cover concrete -- outer edge of cover concrete

set corex1 [expr $coverx1-$coverSec1];   # The distance from the section z-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete

set corey1 [expr $covery1-$coverSec1];   # The distance from the section y-axis to the edge of the core concrete -- edge of the core concrete/inner edge of cover concrete

setnfCorex1 30;# number of fibers for concrete in y-direction -- core concrete setnfCorey1 25;# number of fibers for concrete in z-direction

setnfCoverx1 30;# number of fibers for concrete in y-direction -- cover concrete setnfCovery1 25;# number of fibers for concrete in z-direction

Define the fiber section (define the core patch,four cover patchs and Reinforcement layer respectively) set IDconcCover1 1 set IDconcCore1 2

section fiberSec $SecTag {patch quadr $IDconcCore $nfCorey1 $nfCorex1 -$corex1 $corey1 -$corex1 -$corey1 $corex1 -$corey1 $corex1 $corey1 patch quadr $IDconcCover1 $nfCoverx1 -$coverx1

$covery1 -$corex1 $corey1 $corex1 $corey1 $coverx1 $covery1 patch quadr $IDconcCover1 $nfCoverx1 -$corex1 -$corey1 -$coverx1 -$covery1 $coverx1 -$covery1 $corex1 -$corey1 patch quadr $IDconcCover1 $nfCovery1 -$coverx1 $covery1 -$coverx1 -$covery1 -$corex1 -$corey1

$corex1 $corey1 patch quadr $IDconcCover $nfCovery1 $corex1 $corey1 $corex1 -$corey1 $coverx1

-$covery1 $coverx1 $covery1 layer straight $IDreinf $numBarsSec $barAreaSec $corex1 $corey1 $corex1 -$corey1 layer straight $IDreinf $numBarsSec $barAreaSec -$corex1 $corey1 -$corex1 $corey1 }

assign torsional Stiffness for 3D Model set SecTagTorsion 99;# ID tag for torsional section behavior set SecTag3D 5; # ID tag for combined behavior for 3D model set GJbeam [expr $GC *0.163*0.30* 1/3*pow(0.250,3)]

uniaxialMaterial Elastic $SecTagTorsion $GJ;   # define elastic torsional stiffness

section Aggregator $SecTag3D $SecTagTorsion T -section $SecTag; # combine section propertiesset

IDbeamTransf 2

Geometric transformation for beams set IDbeamTransf 2

geomTransf Linear $transfTag $vecxzX $vecxzY $vecxzZ <-jntOffset $dXi $dYi $dZi $dXj $dYj $dZj> geomTransf Linear $IDbeamTransf 1 1 0 set beamSec 5

number of beam integration points (sections) set np 4

Create beam element

element nonlinearBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens><-iter $maxIters $tol>

Create the non-linearbeam elements' connectivity

element dispBeamColumn $eleTag $iNode $jNode $numIntgrPts $secTag $transfTag <-mass $massDens element dispBeamColumn 81 9 10 $np $beamSec 2;element dispBeamColumn 82 10 11 $np $beamSec 2; element dispBeamColumn 83 11 12 $np $beamSec 2

element dispBeamColumn 84 12 13 $np $beamSec 2;element dispBeamColumn 85 13 14 $np $beamSec 2; element dispBeamColumn 86 14 15 $np $beamSec 2

element dispBeamColumn 87 15 16 $np $beamSec 2;element dispBeamColumn 88 10 15 $np $beamSec 2; element dispBeamColumn 89 11 14 $np $beamSec 2

element dispBeamColumn 90 17 18 $np $beamSec 2;element dispBeamColumn 91 18 19 $np $beamSec 2; element dispBeamColumn 92 19 20 $np $beamSec 2

element dispBeamColumn 93 20 21 $np $beamSec 2;element dispBeamColumn 94 21 22 $np $beamSec 2; element dispBeamColumn 95 22 23 $np $beamSec 2

element dispBeamColumn 96 23 24 $np $beamSec 2;element dispBeamColumn 97 18 23 $np $beamSec 2; element dispBeamColumn 98 19 22 $np $beamSec 2

element dispBeamColumn 99 25 26 $np $beamSec 2;element dispBeamColumn 100 26 27 $np $beamSec 2; element dispBeamColumn 101 27 28 $np $beamSec 2

element dispBeamColumn 102 28 29 $np $beamSec 2;element dispBeamColumn 103 29 30 $np $beamSec 2; element dispBeamColumn 104 30 31 $np $beamSec 2

element dispBeamColumn 105 31 32 $np $beamSec 2;element dispBeamColumn 106 26 31 $np $beamSec 2; element dispBeamColumn 107 27 30 $np $beamSec 2

element dispBeamColumn 108 33 34 $np $beamSec 2;element dispBeamColumn 109 34 35 $np $beamSec2; element dispBeamColumn 110 35 36 $np $beamSec 2

element dispBeamColumn 111 36 37 $np $beamSec 2;element dispBeamColumn 112 37 38 $np $beamSec2; element dispBeamColumn 113 38 39 $np $beamSec 2

element dispBeamColumn 114 39 40 $np $beamSec 2;element dispBeamColumn 115 34 39 $np $beamSec 2; element dispBeamColumn 116 35 38 $np $beamSec 2

element dispBeamColumn 117 40 41 $np $beamSec 2;element dispBeamColumn 118 41 42 $np $beamSec 2; element dispBeamColumn 119 42 43 $np $beamSec 2

element dispBeamColumn 120 43 44 $np $beamSec 2;element dispBeamColumn 121 44 45 $np $beamSec 2; element dispBeamColumn 122 45 46 $np $beamSec 2

element dispBeamColumn 123 46 47 $np $beamSec 2;element dispBeamColumn 124 47 48 $np $beamSec 2; element dispBeamColumn 125 42 47 $np $beamSec 2

element dispBeamColumn 126 43 46 $np $beamSec 2;element dispBeamColumn 127 49 50 $np $beamSec 2; element dispBeamColumn 128 50 51 $np $beamSec 2

element dispBeamColumn 129 51 52 $np $beamSec 2;element dispBeamColumn 130 52 53 $np $beamSec 2; element dispBeamColumn 131 53 54 $np $beamSec 2

element dispBeamColumn 132 54 55 $np $beamSec 2;element dispBeamColumn 133 55 56 $np $beamSec 2; element dispBeamColumn 134 50 55 $np $beamSec 2

element dispBeamColumn 135 51 54 $np $beamSec 2;element dispBeamColumn 136 57 58 $np $beamSec 2; element dispBeamColumn 137 58 59 $np $beamSec 2

element dispBeamColumn 138 59 60 $np $beamSec 2;element dispBeamColumn 139 60 61 $np $beamSec 2; element dispBeamColumn 140 61 62 $np $beamSec 2

element dispBeamColumn 141 62 63 $np $beamSec 2;element dispBeamColumn 142 63 64 $np $beamSec 2; element dispBeamColumn 143 58 63 $np $beamSec 2

element dispBeamColumn 144 59 62 $np $beamSec 2;element dispBeamColumn 145 65 66 $np $beamSec 2; element dispBeamColumn 146 66 67 $np $beamSec 2

element dispBeamColumn 147 67 68 $np $beamSec 2;element dispBeamColumn 148 68 69 $np $beamSec 2; element dispBeamColumn 149 69 70 $np $beamSec 2

element dispBeamColumn 150 70 71 $np $beamSec 2;element dispBeamColumn 151 71 72 $np $beamSec 2; element dispBeamColumn 152 66 71 $np $beamSec 2

element dispBeamColumn 153 67 60 $np $beamSec 2;element dispBeamColumn 154 73 74 $np $beamSec 2; element dispBeamColumn 155 74 75 $np $beamSec 2

element dispBeamColumn 156 75 76 $np $beamSec 2;element dispBeamColumn 157 76 77 $np $beamSec 2; element dispBeamColumn 158 77 78 $np $beamSec 2

element dispBeamColumn 159 78 79 $np $beamSec 2;element dispBeamColumn 160 79 80 $np $beamSec 2; element dispBeamColumn 161 74 79 $np $beamSec 2

element dispBeamColumn 162 75 78 $np $beamSec 2;element dispBeamColumn 163 81 82 $np $beamSec 2; element dispBeamColumn 164 82 83 $np $beamSec 2

element dispBeamColumn 165 83 84 $np $beamSec 2;element dispBeamColumn 166 84 85 $np $beamSec 2; element dispBeamColumn 167 86 87 $np $beamSec 2

element dispBeamColumn 168 87 88 $np $beamSec 2;element dispBeamColumn 169 82 87 $np $beamSec 2; element dispBeamColumn 170 83 86 $np $beamSec 2

Gravity loads applied at each corner node

Define gravity loads

pattern Plain 1 Linear {foreach node {9 17 25 33 41 49 57 65 73 } {load $node 0.0 0.0 -8.750 -20 0.0 }} pattern Plain 2 Linear {foreach node {10 18 26 34 42 50 58 66 74 } {load $node 0.0 0.0 -39.870

-59.7 -61.7 0.0}}

pattern Plain 3 Linear {foreach node {11 19 27 35 43 51 59 67 75 } {load $node 0.0 0.0 -49.150 -74.6 -39.6 0.0}}

pattern Plain 4 Linear {foreach node {12 20 28 36 44 52 60 68 76 } {load $node 0.0 0.0 -22.670 -30.19 30.23 0.0}}

pattern Plain 5 Linear {foreach node {13 21 29 37 45 53 61 69 77 } {load $node 0.0 0.0 -22.670

```
30.19 30.23 0.0}}
pattern Plain 6 Linear {foreach node {14 22 30 38 46 54 62 70 78 } {load $node 0.0 0.0
74.57 -39.6 0.0}} pattern Plain 7 Linear {foreach node {15 23 31 39 47 55 63 71 79 } {load $node
0.0 0.0
59.7 -59.3 0.0}} pattern Plain 9 Linear {foreach node {16 24 32 40 48 56 64 72 80 } {load $node
0.0 0.0 -8.750
20.31 -20.31 0.0}} pattern Plain 10 Linear {load 81 0.0 0.0 -8.750 -11.7 -11.7 0.0};pattern Plain
11 Linear {load
82 0.0 0.0 -20.140 -33.7 -27 0.0}
pattern Plain 12 Linear { load 83 0.0 0.0 -20.104 -33 -27.1 0.0};pattern Plain 13 Linear { load
84 0.0 0.0 -8.750 -11.7 11.7 0.0}
pattern Plain 14 Linear { load 85 0.0 0.0 -8.750 11.7 11.7 0.0};pattern Plain 15 Linear { load
86 0.0 0.0 -20.140 33 -27.1 0.0}
pattern Plain 16 Linear { load 87 0.0 0.0 -20.140 33 27.1 0.0};pattern Plain 17 Linear { load 88
0.0 0.0 -8.750 11.7 -11.7 0.0} # Start of analysis generation # Create the constraint handler
constraints Transformation
# Create the RCM Numberer numberer Plain
# Create the system of equation storage and solver system BandGeneral
# Create the convergent test
# test NormDispIncr $tol $maxNumIter <$printFlag
test NormDispIncr 1.0e-5 1000 5 # create the solution algorithm
algorithm Newton
# Create the time integration scheme
# integrator LoadControl $dLambda1 <$Jd $minLambda $maxLambda> integrator LoadControl
0.1 # Create the static analysis
analysis Static initialize
# analyze $numIncr <$dt><$dtMin $dtMax $Jd>
analyze 1000 0.01
# start of lateral load analysis
loadConst -time 0.0 # Define lateral loads
```

pattern Plain 18 Linear {foreach node {9 10 11 12} {load $node 0.565 0 0.0 0.0 0.0 0.0}} pattern Plain 19 Linear {foreach node{17 18 19 20} {load $node 1.1275 0 0.0 0.0 0.0 0.0}} pattern Plain 20 Linear {foreach node{25 26 27 28} {load $node 1.6925 0 0.0 0.0 0.0 0.0}} pattern Plain 21 Linear {foreach node{33 34 35 36} {load $node 2.2575 0 0.0 0.0 0.0 0.0}} pattern Plain 22 Linear {foreach node{41 42 43 44} {load $node 2.82 0 0.0 0.0 0.0 0.0}} pattern Plain 23 Linear {foreach node{49 50 51 52} {load $node 3.385 0 0.0 0.0 0.0 0.0}} pattern Plain 24 Linear {foreach node{57 58 59 60} {load $node 3.950 0 0.0 0.0 0.0 0.0}} pattern Plain 25 Linear {foreach node{65 66 67 68} {load $node 4.5125 0 0.0 0.0 0.0 0.0}} pattern Plain 26 Linear {foreach node{73 74 75 76} {load $node 5.0775 0 0.0 0.0 0.0 0.0}} pattern Plain 27 Linear {foreach node{81 82 83 84} {load $node 8.86 0 0.0 0.0 0.0 0.0}} # Start of recorder generation

recorder Node <-file $fileName><-xml $fileName><-time><-node ($node1 $node2 ...)> <-nodeRange $startNode $endNode><-region $RegionTag><-node all> -dof ($dof1 $dof2 ...) $respType

recorder Node -file nodeDisp.out -time -node 9 10 11 12 17 18 19 20 25 26 27 28 33 34 35 36 41 42 43 44 49 50 51 52 57 58 59 60 65 66 67 68 73 74 75 76 81 82 83 84 -dof 1 2 6 disp recorder Node -file Rnode.out -time -node 1 2 3 4 5 6 7 8 -dof 1 2 3 4 5 6 reaction

#recorder Drift -file $fileName <-time> -iNode ($inode1 $inode2 ...) -jNode($jnode1 $jnode2 ...) -dof ($dof1 $dof2 ...) -perpDirn ($perpDirn1 $perpDirn2 ...)

recorder Drift -file drift1.out -time -iNode 5 6 7 8 -jNode 13 14 15 16 -dof 1 2 -perpDirn3

recorder Drift -file drift2.out -time -iNode 13 14 15 16 -jNode -perpDirn 3 21 22 23 24 -dof 1 2

recorder Drift -file drift3.out -time -iNode 21 22 23 24 -jNode -perpDirn 3 29 30 31 32 -dof 1 2

recorder Drift -file drift4.out -time -iNode 29 30 31 32 -jNode -perpDirn 3 37 38 39 40 -dof 1 2

recorder Drift -file drift5.out -time -iNode 37 38 39 40 -jNode -perpDirn 3 45 46 47 48 -dof 1 2

recorder Drift -file drift6.out -time -iNode 45 46 47 48 -jNode53 54 55 56 dof
1 2 - perpDirn 3

recorder Drift -file drift7.out -time -iNode 53 54 55 56 -jNode -perpDirn 61 62 63 64 -dof 1 2

recorder Drift -file drift8.out -time -iNode 61 62 63 64 -jNode -perpDirn 69 70 71 72 -dof 1 2

recorder Drift -file drift9.out -time -iNode 69 70 71 72 –jNode 77 78 79 80 --perpDirn 3
dof 1 2

recorder Drift -file drift10.out -time -iNode 77 78 79 80 -jNode -perpDirn 3 85 86 87 88 -dof 1 2

recorder Element <-file $fileName><-time><-ele ($ele1 $ele2 ...)><-eleRange $startEle $endEle><-region $regTag><-ele all> ($arg1 $arg2 ...)

recorder Element -file ele1global.out -time -ele 1 2 81 82 globalForce recorder Element -file el1local.out -time -ele 1 2 81 82 localForce recorder Element -file ele1sec3Force.out -time -ele 1 2 81 82 section 4 force recorder Element -file ele1sec3Defo.out -time -ele 1 2 81 82 section 4 deformation recorder Element -file ele1sec2Stiff.out -time -ele 1 2 81 82 section 4 stiffness

recorder Element -file ele1sec2StressStrain.out -time -ele 1 2 81 82 section 4 fiber 0.25 0.125 1 stressStrain

Change the integrator to take a min and max load increment

integrator LoadControl 1.0 4 0.02 2.0 # Finally perform the analysis

set some parameters

set maxU 0.3810; # Max displacement

set ok 0

set currentDisp 0.0 ;# perform the analysis;

while {$ok == 0 &&$currentDisp <$maxU} { set ok [analyze 1]}; # if the analysis fails try initial tangent iteration; if {$ok != 0} { puts "regular newton failed .. lets try an initial stiffness for this step"; test NormDispIncr 1.0e-5 1000; algorithm Newton -initial; set ok [analyze 1];if {$ok == 0} { puts "that worked .. back to regular newton"} ;test NormDispIncr 1.0e-12 1000; algorithm Newton

} if {$ok == 0} { puts "Pushover analysis completed SUCCESSFULLY"; } else { puts "Pushover analysis FAILED"; } # Print out print node print ele print sec