SEISMIC PERFORMANCE EVALUATION OF REINFORCED CONCRETE MOMENT RESISTING FRAMES

by

Omar El Kafrawy

A Thesis Submitted to the

Department of Building, Civil & Environmental Engineering

Presented in Partial Fulfillment of the Requirements for the Degree of

Master of Applied Science (Civil Engineering)

at Concordia University

Montreal, Quebec, Canada

December 2006

Your file Votre référence
ISBN: 978-0-494-28897-9
Our file Notre référence
ISBN: 978-0-494-28897-9

ABSTRACT

Seismic Performance Evaluation of Reinforced Concrete Moment Resisting Frames

Omar El Kafrawy

During the last few decades, the concepts and procedures of seismic design of buildings have evolved significantly. Traditionally, seismic design was based on providing the structure with the strength it needs to resist equivalent lateral forces that represent the earthquake induced forces. This methodology has proven to be inefficient in providing the target level of seismic performance in all case scenarios. The new methodology in seismic design, performance-based design, aims at designing the structure to achieve the expected performance levels under different levels of seismic hazard. Although this goal is still not fully attained, several developments have taken place towards it. The latest edition of the National Building Code of Canada (NBCC 2005) requires the use of dynamic analysis for the seismic design, except in the case of short to medium height simple buildings with regular geometry where the use of the equivalent static load method is still allowed. In this study, three moment resisting frame buildings assumed to be located in Vancouver, British Columbia are designed based on the equivalent static load method of the NBCC 2005 and their performance is evaluated using nonlinear dynamic time-history analysis. Multiple ground motion records corresponding to different seismic hazard levels are used in the analysis. The effect of infill panels on the seismic response is studied as well. The results show that the design is satisfactory and, in some cases, conservative. The results also show that infill panels, in most cases, enhance the dynamic response of the buildings, while in some cases they may have a negative impact on it.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my thesis supervisor Dr. Ashutosh Bagchi, Assistant Professor, Department of Building, Civil and Environmental Engineering, Concordia University, for his guidance and encouragement throughout the course of this research. Without his motivation and support I wouldn't have been able to write this thesis.

I would like to thank all the faculty members of the Building, Civil and Environmental Engineering Department at Concordia for their help and guidance during the whole program.

The financial assistance provided for this research through the Faculty Research Development Program (FRDP) Grants Committee of Concordia University to Dr. Bagchi is also gratefully acknowledged.

I also acknowledge the financial assistance and the recognition given to me through the Avtar Pall Graduate Award in Earthquake Engineering during my graduate study.

Finally, I would like to thank my family, my friends, H.A.Y.T. and my colleagues for their moral support and encouragement to complete my research.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 GENERAL

In Canada, the west coast is the most vulnerable area to seismic excitations as it lies on two of earth's largest tectonic plates, the North American Plate and the Pacific Plate. According to seismologists a major seismic event could occur anytime in the near future along the Canadian west coast (Public Safety and Emergency Preparedness Canada, 2006). The magnitude of an earthquake is measured on the Richter scale. It is an indicator of the amount of energy the earthquake releases and is a function of the maximum displacement recorded on a seismograph located a hundred kilometers far from the epicenter. Earthquakes with magnitudes of five and above are strong enough to be recorded by any seismograph. During the last century, Canada has witnessed about twenty three earthquakes with such a magnitude, ten of which occurred only in the last thirty years (NRCAN, 2006). Major earthquakes are those with magnitudes between seven and eight, while great earthquakes are those with magnitudes between eight and nine. These earthquakes could cause massive damage over very large areas. A moderate earthquake of magnitude five can also cause damage in densely populated areas. The magnitude of an earthquake also depends on the distance from the epicenter as well as the soil properties of the site. Soil properties can significantly magnify the effect of an earthquake. Foo *et al.* (2001) reported that a 6.8 magnitude earthquake near Seattle on February the 28^{th}, 2001 jolted Vancouver and rattled the buildings and occupants, while in 1988 the Saguenay Region in Quebec was hit with a 6.0 magnitude earthquake, which is considered to be the strongest earthquake in eastern North America in the last fifty years.

The structural design of buildings to resist earthquakes has been a very important problem in civil engineering especially during the last few decades. It's a problem that involves a lot of uncertainties and complexities. The traditional seismic design methodology is to reduce the seismic design process in the case of simple and regular buildings from a dynamic problem to a static one using what is known as the equivalent static load method (ESLM). The ESLM as implied from its name assumes a static force applied laterally along the height of the building to represent the forces induced by the design level of earthquake. The design is then carried out using linear static analysis. However, the reduction of a dynamic problem to a static one is an approximation that doesn't necessarily ensure that the safety or economy of the structure is achieved. The latest edition of the National Building Code of Canada (NBCC 2005) recommends the use of dynamic analysis in the design, but still allows the use of the ESLM for seismic design in the case of simple and regular buildings that are below sixty meters of height. The need for performance evaluation and performance-based design becomes important as the buildings designed using the ESLM may not achieve the required level of performance or produce an over-performing and conservative design.

The methodology in seismic design nowadays is shifting from just specifying minimum forces and requirements for the design to a more performance-based approach that ensures as much as possible that the target seismic performance of the structures is attained. Although a coherent and integrated performance-based design methodology is still not fully developed, it's just a matter of time for this to happen. Meanwhile, and with the release of the NBCC 2005, research work should be conducted to examine the seismic provisions of this new code. The ESLM of the NBCC 2005 is based on a minimum static force that the structure should resist as well as a maximum amount of equivalent static inter-storey drift and the satisfaction of the "strong column – weak beam" requirement which will be later explained. The NBCC 2005 also specifies a limit for the maximum inter-storey drift to ensure the "Collapse Prevention" performance of a building under the design level of seismic hazard.

1.2 NBCC 2005

Although the NBCC 2005 continues to use the ESLM in the seismic design of regular buildings, there are some significant changes in the method compared to that of the previous code (NBCC 1995). The seismic provisions of the NBCC 2005 are discussed in detail in the articles published in the special issue of the Canadian Journal of Civil Engineering (CJCE, 2003). While the NBCC 1995 used a two parameter approach for seismic zoning and the design was based on a seismic event with 10% probability of being exceeded in 50 years or a recurrence interval of 475 years, the NBCC 2005 uses site-specific uniform hazard spectra (UHS) corresponding to 2% probability of being exceeded in 50 years or having a recurrence interval of 2500 years (Humar and Mahgoub, 2003). The design base shear is given out using Equation 1.1:

$$V = \frac{S(T_a)M_v I}{R_o R_d} W \geq \frac{S(2)M_v I}{R_o R_d} W$$

$$\leq \frac{2}{3} \times \frac{S(0.2)M_v I}{R_o R_d} W \qquad (1.1)$$

where V is the equivalent static force or base shear, $S(T_a)$ is the design spectral response acceleration expressed as a ratio to gravitational acceleration for the fundamental period of vibration T_a, M_v is a factor that accounts for the higher modes effect, I is the importance factor of the building, R_o is the overstrength related force modification factor that accounts for the dependable portion of reserve strength in a structure, R_d is the ductility related force modification factor that reflects the capability of a structure to dissipate energy through inelastic behaviour and W is the dead load of the structure plus 25% of the design snow load.

The code gives an empirical formula to estimate the value of the fundamental period of vibration of the structure T_a for reinforced concrete frame buildings:

$$T_a = 0.075(h_n)^{3/4} \tag{1.2}$$

where h_n is the height of the top storey above the base.

The design spectral response acceleration values $S(T)$ are calculated using Equation 1.3 while linearly interpolating for intermediate values of T.

$$\begin{aligned} S(T) &= F_a S_a(0.2) \text{ for } T \le 0.2\text{ s} \\ &= F_v S_a(0.5) \text{ or } F_a S_a(0.2) \text{ whichever is smaller for } T = 0.5 \\ &= F_v S_a(1.0) \text{ for } T = 1.0\text{ s} \\ &= F_v S_a(2.0) \text{ for } T = 2.0\text{ s} \\ &= F_v S_a(2.0)/2 \text{ for } T \ge 4.0\text{ s} \end{aligned} \tag{1.3}$$

where $S_a(T)$ is the 5% damped spectral response acceleration expressed as a ratio to gravitational acceleration for a period T, F_a is an acceleration based site coefficient and F_v is a velocity based site coefficient. F_a and F_v are site dependant factors that depend on the type of soil of the site. The design spectrum for a Vancouver site with reference soil (Type C) is shown in Figure 1.1 where F_a and F_v are both equal to 1. The reference soil (Type C) as defined in the NBCC 2005 is "very dense soil and soft rock". If the soil type in a given site is different, the modification factors (F_a and F_v) are used to magnify or reduce the design spectrum.

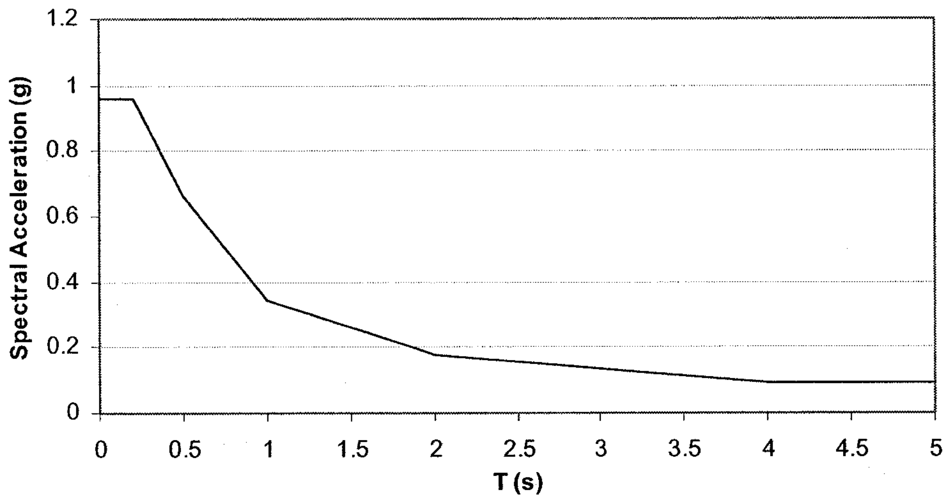


Figure 1.1: Design Response Spectrum for a Vancouver Site with Soil Type C

The lateral load distribution of the base shear is the same as in the previous code and is given out using Equation 1.4:

$$F_x = (V - F_t)\frac{W_x h_x}{\sum_{i=1}^{n} W_i h_i} \tag{1.4}$$

where F_x is the lateral force applied at level x, n is the total number of storeys, h_x and h_i are the heights above the base to level x and i respectively, W_x and W_i are the portion of W located at level x and i respectively. As implied by the previous equation not all of V is distributed on the storeys using the described formula. A portion of V is concentrated at the top storey of the building F_t and is determined from the Equation 1.5:

$$F_t = 0.07 T_a V \leq 0.25V \tag{1.5}$$

F_t may be taken as zero if T_a doesn't exceed 0.7 s.

1.3 PERFORMANCE EVALUATION METHODOLOGY

Several attempts have been made by researchers and practitioners in the last decade to transform the seismic design code provisions from prescription-based to performance-based format. The prescription-based code is one which specifies the design methodologies, materials and construction process in detail leaving very limited room for flexibility. The aim of performance-based design is to design the structure based on the desired performance level to be achieved under seismic excitations using any acceptable and feasible solution. The seismic performance of a structure is generally evaluated using inter-storey drifts, inelastic deformations, strains and also, although not as widely used, damage indices.

Several simplified methods are developed for performance-based design such as displacement-based design methods. The concept of displacement-based design is specifying the target displacement and using it to obtain the design force. Other methods, such as damage spectrum (Bozorgnia and Bertero, 2004) and yield point spectra (Aschheim, 2004) can also be used for performance-based design. Although several simplified methods are developed, these methods are still very approximate and differ significantly from each other (Yousuf, 2006). Since no reliable performance-based design methodology is developed yet, the NBCC 2005 still allows the use of a force-based approach utilizing the ESLM when dealing with regular and simple buildings. However, the NBCC 2005 requires the use of dynamic analysis when designing irregular and complex structures or with any structure exceeding sixty meters in height. While, dynamic analysis is the best way to deal with seismic design it's still not very practical for everyday office use as it requires a lot of time in solving for several representative ground motion records and in data pre and post-processing. Also there is a wide range of uncertainties related to the selection of appropriate ground motion records, material modelling and analysis algorithms. Until dynamic analysis becomes an easy task to perform, engineers will prefer to use the equivalent and approximate methods in the seismic design process. In this study, as mentioned before, the focus is on the ESLM and how the structures designed using this method perform under seismic ground excitations. Although the NBCC 2005 is not fully a performance-based design code, it is presented in an objective-based format where an acceptable solution can be produced based on alternate methods, not specifically mentioned by the code.

The NBCC 2005 doesn't specify different levels of seismic performance to be achieved by the structure under different levels of seismic hazard; instead it specifies 2.5% inter-storey drift as a maximum allowable value beyond which the structure is considered to have collapsed. However, the Vision 2000 (1995) committee of the Structural Engineers Association of California (SEAOC) have produced some guidelines for earthquake levels and the required performance under these types of earthquakes. Tables 1.1 and 1.2 show the classification of earthquakes as well as the performance levels defined by SEAOC Vision 2000 (1995) and their relationship with the maximum transient inter-storey drift. Figure 1.2 shows the performance objectives to be achieved by a basic facility (i.e. residential and office buildings) subjected to different types of earthquakes. It is noticed that this figure doesn't include the "Extremely Rare" earthquake type which is proposed in the seismic design provisions of the NBCC 2005 and the International Building Code (IBC, 2003).

Table 1.1: Earthquake Classification (SEAOC Vision 2000, 1995; Bagchi, 2001)

Earthquake Classification	Recurrence Interval	Probability of Occurrence
Frequent	43 years	50% in 30 years
Occasional	72 years	50% in 50 years
Rare	475 years	10% in 50 years
Very Rare	970 years	10% in 100 years
Extremely Rare	2500 years	2% in 50 years

Table 1.2: Performance Levels (SEAOC Vision 2000, 1995)

Performance Level	Damage Description	Maximum Allowable Inter-Storey Drift
Fully Functional	Negligible	0.2%
Operational	Minor local yielding at a few places. No observable fractures. Minor buckling or observable permanent distortion of members.	0.5%
Life Safe	Hinges form. Local buckling of some beam elements. Severe joint distortion. Isolated connection fractures. A few elements may experience fracture.	1.5%
Near Collapse	Extensive distortion of beams and column panels. Many fractures in connections.	2.5%

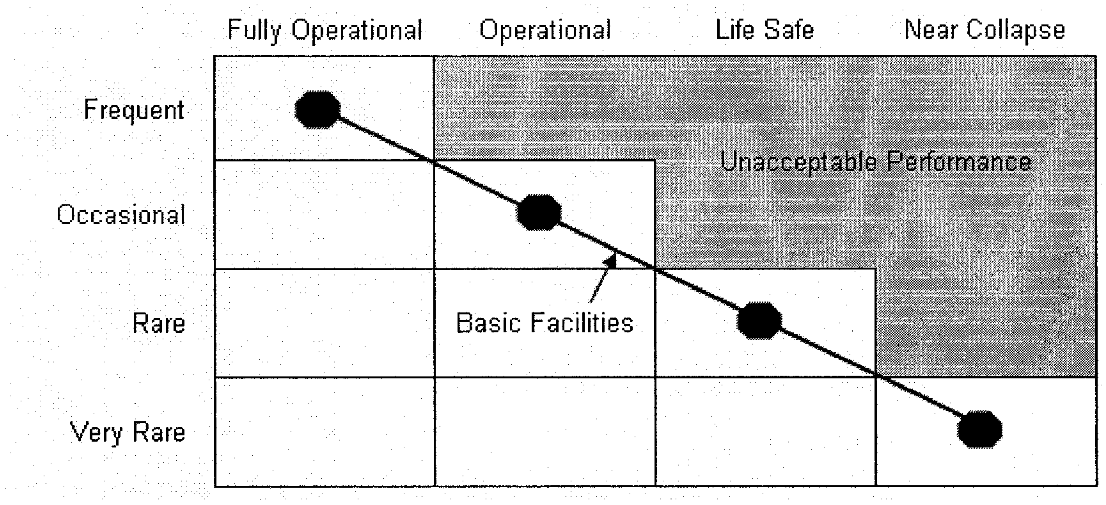


Figure 1.2: Performance Objectives for a Basic Building (SEAOC Vision 2000, 1995)

1.4 OBJECTIVES AND TASKS

The objectives of this research could be summarized in the following two points:

- The evaluation of the seismic performance achieved by reinforced concrete moment resisting frame buildings designed using the NBCC 2005 provisions, through nonlinear dynamic analysis for multiple levels of seismic hazard.
- The evaluation of the seismic performance of the structures when taking into account the effect of non-structural elements, in this case, infill panels.

In order to achieve these goals, the following tasks are identified:

- The implementation of the NBCC 2005 seismic design provisions in the design of a set of three concrete moment resisting frame structures located in Vancouver, British Columbia which represents one of the highest seismic zones in Canada.
- The identification of appropriate software tools for seismic performance evaluation of building frames and automation of the software tools used in the design and analysis of the buildings.

- The identification of an appropriate analysis model to represent the effect of infill panels on the seismic performance of the structures.
- The selection of appropriate ground motion records for the nonlinear dynamic analysis. For multiple levels of seismic hazard, the seismic events corresponding to the return periods of 475, 970 and 2500 years are considered. These hazard levels are denoted in this thesis by UHS-500, UHS-1000 and UHS-2500, respectively.
- The nonlinear dynamic time-history analysis of the structures and the evaluation of the performance achieved based on response and damage parameters.

1.5 ORGANIZATION OF THE THESIS

The thesis is divided into six chapters. The introduction and the objectives of the research are discussed in this first chapter. A summary of the literature survey is discussed in Chapter 2 while a description of the computer software used in this research and the development of the automation schemes for the design and analysis are presented in Chapter 3. The design of the buildings considered here is discussed in Chapter 4. The seismic performance evaluation of the buildings is presented in Chapter 5 and the summary and conclusions are presented in the final chapter, Chapter 6, which is followed by the list of References and Appendices.

CHAPTER 2

2. LITERATURE REVIEW

2.1 GENERAL

The first step in seismic design is providing the structure with lateral load resisting elements that are able to perform and resist ground motions occurring in any direction. The adequacy of a structure is decided based on the evaluation of a representative mathematical model for the effects of the design earthquake level (FEMA-368, 2001). Several types of lateral load resisting elements could be used. The most commonly used elements are shear walls, cores and moment resisting frames. The shear walls and cores are normally utilized in high rise buildings and provide a large amount of lateral stiffness to the structure. Moment resisting frames are designed to resist lateral loads by the framing action between the horizontal components and the vertical components through rigid joints. Additional elements could be used such as dampers which work to dissipate the earthquake energy as an addition to the damping of the structure. The choice of the type of lateral load resisting elements to be used in a structure depends on several factors such as the financial cost, the level of performance required, the importance of the building and the architectural design. Several types of lateral load resisting elements could be used in one building.

Researchers always look for new ways and methods to implement in earthquake engineering to provide the safety required for the occupants and for society in general. Other innovative resisting systems in earthquake engineering are being developed and updated such as isolation bearings which are mainly used in bridges. The concept behind isolation bearings is basically to allow the foundation to move laterally without affecting the vertical stability of the structure so as to isolate the building from the lateral ground motion as much as possible. This is done by using isolation devices in the foundation that have very high vertical stiffness while being able to elastically deform under the effect of lateral loads. The most widely used type of isolation devices is the lead-rubber bearing (LRB). However, the problem with the LRB is that due to large deformation, it may become unstable or some residual deformation may be unrecovered and may need to be replaced (Choi *et al.*, 2005). A new type of isolation device is developed and explained in Choi *et al.* (2005), the shape memory alloy (SMA) which is described as being super-elastic because it can recover its deformation after the release of the applied force.

The focus in this research is on buildings with reinforced concrete moment resisting frames. Concrete moment resisting frames are defined as lateral load resisting elements composed of beams and/or slabs representing the horizontal framing components, columns representing the vertical framing components and rigid joints connecting them (FEMA-356, 2000). To develop rigid connections in concrete frames, the beams and columns are cast monolithically.

2.2 TECHNICAL BACKGROUND OF THE NBCC 2005

The NBCC 2005 is not fully performance-based; however it incorporates the performance criteria by specifying the drift limit for "Collapse Prevention". As previously mentioned, the technical background of the seismic provisions of the NBCC 2005 has been published in a number of papers in the special issue of the Canadian

Journal of Civil Engineering (CJCE, 2003). The main change is the introduction of uniform hazard spectra (UHS) calculated at specific geographical locations (Adams and Atkinson, 2003). The return period of the design seismic hazard has changed from 475 years to approximately 2500 years. The UHS methodology is based on calculating response-spectral ordinates for a range of given periods (McGuire, 1977). Period-dependent site factors as well as ductility and overstrength reduction factors are also among the changes introduced in the NBCC 2005 (Heidebrecht, 2003). Dynamic analysis is described as the main method of analysis while the ESLM can still be used in some special cases.

2.3 PERFORMANCE-BASED DESIGN AND EVALUATION

As mentioned in the previous chapter, the earthquake engineering methodology is changing towards a performance-based approach. Although, the achievement of this goal might take a few more years, it seems almost inevitable. The old prescription-based approach has proven to be inadequate and can no longer be relied on to provide safety and economy to modern building structures. Performance-based design and evaluation depends on qualifying as well as quantifying the seismic performance of a structure based on response and damage parameters. As shown in Table 1.2, the Structural Engineers Association of California (SEAOC) has defined four structural performance levels and qualified the damage and quantified the inter-storey drift related to such performances (SEAOC Vision 2000, 1995). How much such definition and relation is acceptable remains a debatable topic that should be further explored and justified experimentally.

Response parameters used in seismic performance-based evaluation play a very important role. The appropriate selection of such parameters is a key factor in the performance evaluation process. Response or damage parameters such as inter-storey drifts and roof drifts which are displacement-based quantities are among the most widely used parameters (Bagchi, 2001). While roof drifts can be used in the damage assessment, they cannot really reflect the damage of each storey in the structure. For this reason, inter-storey drifts are more widely used and relied on in the evaluation process. However, an evaluation based solely upon the maximum inter-storey drifts that occur during an earthquake event might not be sufficient enough to capture the amount of damage induced by the cyclic loading nature of the earthquake. To take this factor into account, Park and Ang (1984) suggested a damage index that combines the effect of plastic deformation with the hysteretic energy dissipation (Bagchi, 2001). The hysteretic energy represents the inelastic energy dissipation during an earthquake ground motion (Bozorgnia and Bertero, 2001). The Park and Ang damage index for a structural element is expressed using Equation 2.1:

$$DI = \frac{\delta_m}{\delta_u} + \frac{\beta}{\delta_u P_y} \int dE_h \tag{2.1}$$

where δ_m is the maximum deformation experienced by the element during the loading history, δ_u is the ultimate deformation of the element, P_y is the yield strength of the element, $\int dE_h$ is the hysteretic energy absorbed by the element during the ground motion application and β is a model constant which depends on the strength deterioration characteristics of the element's material.

This damage index is included in IDARC2D (IDARC2D, 2006), a nonlinear dynamic analysis software which is discussed in the next chapter. IDARC2D calculates the damage indices of structural elements based on a modified version of the Park and Ang damage model (Kunnath *et al.*, 1992) which considers the end section in the damage calculation instead of the whole element. The modified damage index is then expressed by Equation 2.2:

$$DI = \frac{\theta_m - \theta_r}{\theta_u - \theta_r} + \frac{\beta}{\theta_u M_y} E_h \tag{2.2}$$

where θ_m is the maximum rotation experienced by the end section of an element during the loading history, θ_u is the ultimate rotation of the section, θ_r is the recoverable rotation after the unloading, M_y is the yield flexural strength of the section, E_h is the hysteretic energy absorbed by the section. The element damage is then taken as the maximum damage index of both end sections (Valles *et al.*, 1996).

The damage assessment based on structural elements is not sufficient to characterize the damage of the structure. Two additional damage parameters could be deduced from the element damage indices. By summing up the damage indices of all elements in a storey and multiplying each index by a weighting factor that is dependant on the amount of dissipated energy by the element with respect to the whole hysteretic energy of the storey, a storey damage index could be calculated. Similarly, by multiplying the damage indices of each storey by storey weighting factors that reflect the amount of dissipated energy by the storey relative to the total hysteretic energy absorbed by the structure, an overall structural damage index is obtained. Equations 2.3 through 2.4 show how the storey and damage indices are calculated:

$$DI_{storey} = \sum (\lambda_i)_{element} (DI_i)_{element} \; ; \; (\lambda_i)_{element} = \left(\frac{E_i}{\sum E_i} \right)_{element} \tag{2.3}$$

$$DI_{overall} = \sum (\lambda_i)_{storey} (DI_i)_{storey} \; ; \; (\lambda_i)_{storey} = \left(\frac{E_i}{\sum E_i} \right)_{storey} \tag{2.4}$$

where λ_i is the energy weighting factor of the element or storey i and E_i is the hysteretic energy absorbed by the element or storey.

According to Park *et al.* (1986), the damage observed in nine reinforced concrete buildings was used to relate the overall damage index to the degree of damage in a structure as well as the physical appearance of the damage. Table 2.1 shows the interpretation of the overall structural damage based on this study.

Table 2.1: Interpretation of the Overall Damage Index (Park *et al.,* 1986)

Damage Degree	Physical Apperance	Overall DI	State of Building
Collapse	Partial or total collapse of building	>1.0	Loss of building
Severe	Extensive crushing of concrete, buckled reinforcement	0.4-1.0	Beyond repair
Moderate	Large Cracks, spalling of concrete in weaker elements	<0.4	Repairable
Minor	Minor cracks, partial crushing of concrete		
Slight	Sporadic occurrence of cracking		

More research is needed in seismic engineering to take the performance-based design methodologies to a mature level to provide better safety for occupants and buildings in case of an earthquake. The Federal Emergency Management Agency in the USA (FEMA) which sponsors the development of building rehabilitation guidelines has published several standard reports for seismic rehabilitation guidelines starting with the FEMA-273 report (1997). These reports alongside with other publications (SEAOC Vision 2000, 1996) represent the first generation in performance-based seismic design (Hamburger *et al.*, 2004).

A new approach in performance-based design is emerging and is based on a better understanding of the probability or the likelihood of the building achieving a certain performance level under different kinds of scenarios. This is the next-generation performance-based earthquake engineering. The Pacific Earthquake Research Center (PEER) has adopted this goal and is trying to develop an integrated methodology for performance-based seismic engineering (Moehle and Deierlein, 2004). PEER has broken down the performance-based design into several components or steps. The first step is defining the "Intensity Measure" (IM) which involves categorizing ground motion records based on their intensity. The intensity could be measured using spectral acceleration or peak ground acceleration and is usually expressed as a mean probability of exceedance (p(IM)). The IM should depend on the location of the structure as well as the characteristics of the structure such as the fundamental period of vibration and the foundation and soil type. The next step is to define the "Engineering Demand Parameters" (EDP), which basically means expressing the seismic response of the structure in terms of response parameters such as inter-storey drifts, inelastic deformations and strains. The following step is to define "Damage Measures" (DM) using damage analysis and relate the DM to the EDP. The final step is calculating the "Decision Variables" (DV) which expresses in a probabilistic manner the likely losses that the decision makers would suffer under different scenarios. The losses could be direct financial losses, time losses or casualties. This approach would help the investors and decision makers to choose the desired level of performance with a better understanding of the risk and uncertainties accompanied with the design.

2.4 SUMMARY

It is noticed that the seismic design methodology of the past is changing and the performance-based design methodologies are evolving. Since the NBCC 2005 was recently published, not many studies on the performance characteristics of buildings designed according to its provisions have been conducted yet. The NBCC 2005 seismic provisions need to be tested and evaluated. To reach the goal of performance-based design, performance levels need to be well defined. This can start by performance evaluation. Performance evaluation is basically the backbone of performance-based design. The effect of non-structural elements needs also to be taken into account. In this study, the seismic performance of NBCC 2005 designed reinforced concrete moment resisting frame structures is evaluated for this purpose. The observed performance can offer some guidelines for the modification to the design methodologies such that the code-based design can be optimized for performance.

CHAPTER 3

3. COMPUTER APPLICATION

3.1 ANALYSIS SOFTWARE

Two dynamic analysis programs are used in the post-design analysis which is carried out to evaluate the seismic performance of the frames designed herein. The programs used here, IDARC2D and DRAIN-2DX, are non-commercial programs selected because they are specifically designed and used for nonlinear dynamic analysis of plane frame structures and are available for research purposes. The reason for using these two programs is to make a comparative study between two nonlinear dynamic analysis programs and to obtain a better assessment of the performance of the frames under ground acceleration and lateral loading in general. In other words, the comparison is intended to enhance the reliability of the results.

3.1.1 IDARC2D

In the beginning of this study IDARC2D was chosen because it's a nonlinear dynamic analysis program specifically designed for reinforced concrete frame structures. The program was developed at the State University of New York at Buffalo in 1987 for research purposes and the latest version was released in 2006 (IDARC2D, 2006). The user has the ability of defining the behaviour of the concrete material in a flexible manner. The user can define the hysteretic behaviour of concrete specifying different levels of stiffness degradation, strength decay and slip parameters as well as choosing a bilinear or a trilinear or a vertex oriented model. The program also has the ability to generate the moment-curvature relationships of the different members used as well as the P-M (axial force – moment) interaction curves. Four types of analyses can be performed using IDARC2D, inelastic incremental static analysis, monotonic push-over analysis, inelastic dynamic analysis and quasi-static cyclic analysis. The push-over analysis available in this software can be of the following two types, force control and displacement control. The force control analysis has several types of lateral load distribution that could be used, linear distribution or inverted triangular distribution, uniform distribution, modal adaptive distribution, distribution proportional to a power of the storey elevation, and finally, a user defined force distribution. The program does some post-processing and calculates the damage indices for the members as well as for the storeys, and finally, for the overall structure. The floors are assumed to be rigid so that each floor's displacement could be considered as a degree of freedom.

3.1.2 DRAIN-2DX

DRAIN-2DX was primarily used in this research in the gravity and lateral static analysis carried out during the design; however the program was used again in the post-design analysis to check the results of IDARC2D. It's a non-commercial general purpose program developed at the University of California, Berkeley, which is most suitable for steel structures but that can also be used for other material types (Parkash *et al.*, 1993). It has all the necessary analysis options for nonlinear gravity and lateral static analysis, push-over analysis and dynamic analysis. However, the hysteretic behaviour of the material is assumed to be without degradation and is based on a bilinear moment-curvature relationship. There is a special element for reinforced concrete modelling with strength degradation incorporated in it. However, it is complex and difficult to use.

3.2 TYPES OF ANALYSES USED IN THIS STUDY

The software tools (e.g. IDARC2D and DRAIN2DX) utilized in this study are used to perform different types of analyses. These types of analyses are described in this section and are all based on some form of the dynamic equilibrium equation, Equation 3.1 which is written in matrix form for multi-degree-of-freedom systems:

$$\underset{n\times n}{[M]}\underset{n\times 1}{\{\ddot{u}\}}+\underset{n\times n}{[C]}\underset{n\times 1}{\{\dot{u}\}}+\underset{n\times n}{[K]}\underset{n\times 1}{\{u\}}=\underset{n\times 1}{\{F\}} \tag{3.1}$$

where $[M]$ is the mass matrix of the structure, $[C]$ is the damping matrix of the structure, $[K]$ is the stiffness matrix of the structure which is derived from the cross sectional dimensions and the building configuration, $\{\ddot{u}\}$, $\{\dot{u}\}$, $\{u\}$ are respectively the nodal vectors of acceleration, velocity, displacement, $\{F\}$ is the nodal force vector which is derived from the applied forces and n is the number of degrees of freedom of the structure.

3.2.1 Linear Static Analysis

Linear (elastic) static analysis is used in the design process to produce the design shear forces, axial forces and bending moments due to the applied gravity loads and lateral equivalent static loads. For static analysis, the equilibrium Equation 3.1 is reduced to Equation 3.2 which is in matrix form:

$$\underset{n\times n}{[K]}\underset{n\times 1}{\{u\}}=\underset{n\times 1}{\{F\}} \tag{3.2}$$

Equation 3.2 is solved for the nodal displacements, and then the design forces and moments are obtained from these displacements. These design stress resultants are used to obtain the required area of reinforcement in all sections.

3.2.2 Modal Analysis

Modal analysis is a form of dynamic analysis used to simulate the free vibration characteristics of the structure. This analysis is required to obtain the natural frequencies (or periods) and mode shapes of vibration of the structure. The natural frequencies (or periods) are then used in updating the design base shear as they are also used in the record scaling process required for the dynamic analysis for actual ground motion records. The frequencies of vibration can also be used to reflect the stiffness of a structure with respect to its mass. The equilibrium equations used in this process are described in matrix form in Equation 3.3:

$$\underset{n\times n}{[M]}\underset{n\times 1}{\{\ddot{u}\}}+\underset{n\times n}{[K]}\underset{n\times 1}{\{u\}}=\underset{n\times 1}{\{0\}} \tag{3.3}$$

The free vibration response of a structure is a harmonic one, which transforms Equation 3.3 to Equation 3.4 which is solved through eigenvalue analysis:

$$\left[\underset{n\times n}{[K]}-\omega_i^2\underset{n\times n}{[M]}\right]\underset{n\times 1}{\{a_i\}}=\underset{n\times 1}{\{0\}} \tag{3.4}$$

where ω_i is the angular frequency of the i^{th} mode of vibration and $\{a_i\}$ is the corresponding modal vector which represents the amplitudes of vibration.

3.2.3 Nonlinear Static Push-Over Analysis

Nonlinear static push-over analysis is performed to simulate the inelastic response of the structure to lateral loading. Two types of push-over analyses can be conducted, force controlled and displacement controlled push-over analysis. In both types, the gravity loads are applied prior to the push-over analysis. In the force controlled push-over analysis, lateral loads corresponding to a specific distribution are monotonically increased from zero to the point of collapse, while in the displacement controlled analysis, a lateral roof displacement is incrementally applied to the structure. The push-over analysis in both cases produces the base shear versus the roof displacement curve which could be used to obtain:

- The roof displacement and base shear capacity corresponding to the point of failure of the structure.
- The equivalent yield point of the structure.
- The estimated ductility of the structure.
- The cracking and yielding sequence of the structural members.

The equations of equilibrium used in this type of analysis are shown in matrix form in Equation 3.5:

$$\underset{n\times n}{[K_t]}\underset{n\times 1}{\{\Delta u\}} = \underset{n\times 1}{\{\Delta F\}} \tag{3.5}$$

where $[K_t]$ is the tangent stiffness matrix, $\{\Delta u\}$ is the incremental displacement vector and $\{\Delta F\}$ is the incremental force vector.

3.2.4 Nonlinear Dynamic Analysis

Nonlinear dynamic analysis is conducted in this study to examine the response of the buildings to earthquakes. The earthquake energy is transmitted to the buildings by ground shaking. This ground motion is represented here as ground acceleration time-history. The equilibrium equations used for dynamic analysis are described in matrix form in Equation 3.6:

$$\underset{n\times n}{[M]}\underset{n\times 1}{\{\Delta\ddot{u}\}}+\underset{n\times n}{[C]}\underset{n\times 1}{\{\Delta\dot{u}\}}+\underset{n\times n}{[K]}\underset{n\times 1}{\{\Delta u\}}=\underset{n\times 1}{\{\Delta F_{eff}\}} \tag{3.6}$$

where $\{\Delta\ddot{u}\}$, $\{\Delta\dot{u}\}$ are respectively the incremental vectors of acceleration and velocity and $\{\Delta F_{eff}\}$ is the incremental effective dynamic forces caused by the ground acceleration.

Under the effect of dynamic vibrations, the structure dissipates a part of the earthquake energy through inelastic hysteretic behaviour and/or damping. The output parameters of this type of analysis, such as displacements and hysteretic energies could be used to evaluate the seismic performance of the structure.

3.3 SOFTWARE AUTOMATION

Because IDARC2D is a non-commercial program it lacks the friendly user interface for data input and output. Instead, the user has to input all data in a text file following a certain format in writing the input data. This method of input is not only cumbersome; it's time consuming especially when one is conducting a large set of analyses (El Kafrawy *et al.*, 2006). The output is also produced in text format and requires some post-processing to extract meaningful and important information from it.

For these reasons, a pre-processor and a post-processor have been developed here. Although DRAIN-2DX has the same problems in data input and output, the pre and post-processors are developed only for IDARC2D. Figure 3.1 shows the software automation scheme. First, input data is collected through the pre-processor which is a Microsoft

Excel file created with the use of Visual Basic (VB) scripts. Using the pre-processor, the input file is created in the format recognized by IDARC2D. Then, by running IDARC2D, output files are developed, the general output file, the deformation output file and the storey history output files. Using a MATLAB-based post-processor, relevant output data is extracted and plotted, such as mode shapes, push-over curves, displacements and inter-storey drifts time-history curves and maximum inter-storey drifts.

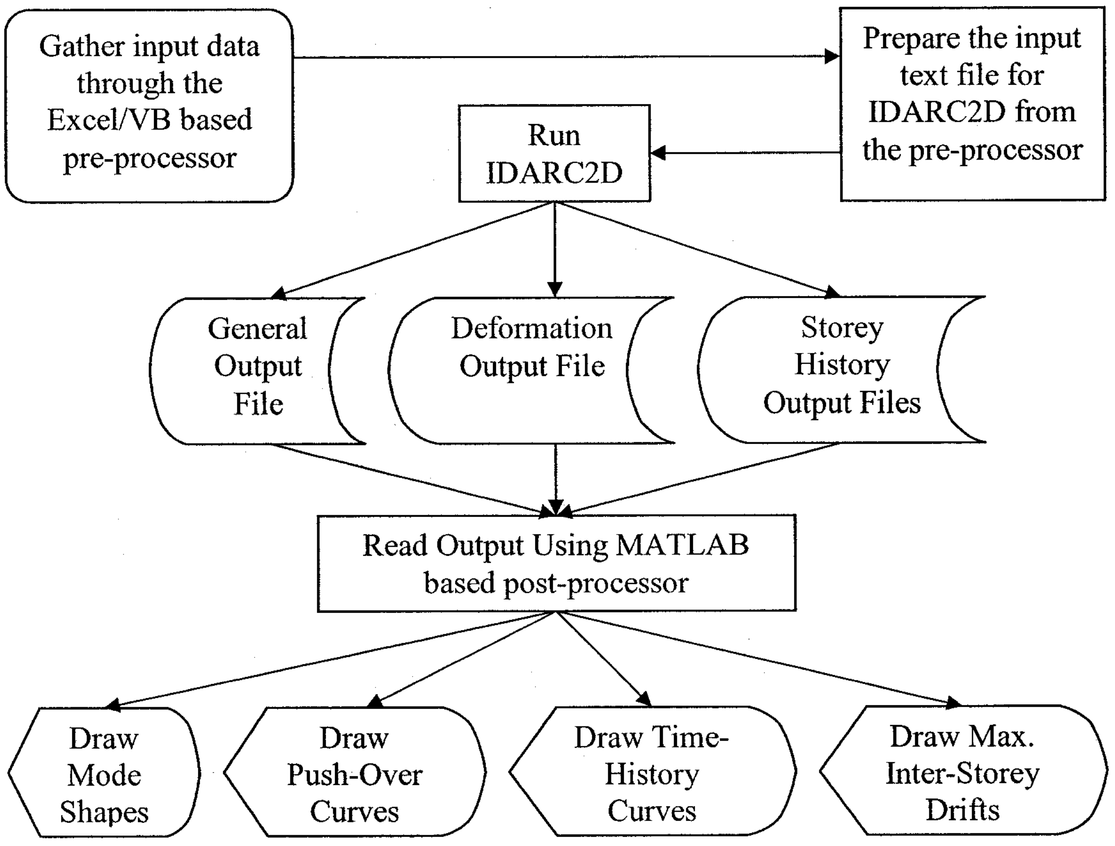


Figure 3.1: Schematic Architecture of the Automation System for IDARC2D

3.3.1 Pre-Processor for IDARC2D

Using Microsoft Excel and VB scripts a workbook is developed that contains seven worksheets where data is to be input. The worksheets are labelled: “General Info”, “Nodal Weights”, “Material Properties”, “Element Properties”, “Element Connectivities”, “Static Loads and Analysis”.

Since a large part of the input is in vector and matrix form, Excel seemed like a very good tool to use as a user interface for data input to IDARC. With the use of Visual Basic coding, the Excel workbook becomes dynamic and automatic in generating the required cells, vectors and matrices as well as retrieving this data and writing it in an input text file the name and location of which are specified by the user using a “Save As” window form. Furthermore, the worksheets are designed in such a manner as to give the user flexibility in inputting the data. Rows are added automatically when key cells are edited and cells are either locked or unlocked accordingly. Screenshots of four of the seven worksheets in the workbook of the pre-processor are shown in Figures 3.2 through 3.5.

Figure 3.2 shows the “General Info” worksheet where the general data is input. The worksheet is divided into six sections, “Building Data”, “Element types”, “Element Data”, “Material Properties Sets”, “Static Loads Data” and “Structural Behaviour”. The “Building Data” section is where the number of storeys of the structure is input as well as the floor heights, the number of non-identical frames, the number of duplicate frames and the number of column lines. The “Element Types” section is where the number of types of columns, beams, transverse beams and infill panels are entered. The next section, “Element Data”, is where the number of columns, beams, transverse beams and infill panels are input. The “Material Properties Sets” section is where the number of concrete material properties sets, reinforcement steel properties sets, masonry material properties sets and hysteretic modeling rules sets are entered. The number of uniform loads, concentrated vertical loads, lateral loads and concentrated moments applied on the structure are to be given in the "Static Loads Data" section. The final section in this worksheet is the "Structural Behaviour" section where the type of flexibility and plasticity is to be defined as well as the inclusion or the exclusion of the P-Delta effect. Two types of flexibility could be chosen, linearly distributed or uniformly distributed flexibility. Also two types of plasticity could be defined, spread plasticity and concentrated plasticity. Two unit systems could be chosen for the analysis. A dropdown list is located at the upper right corner of the worksheet and the user has the choice to use either millimeters and kilo Newtons (mm, kN) or inches and kilo pounds (inch, kips).

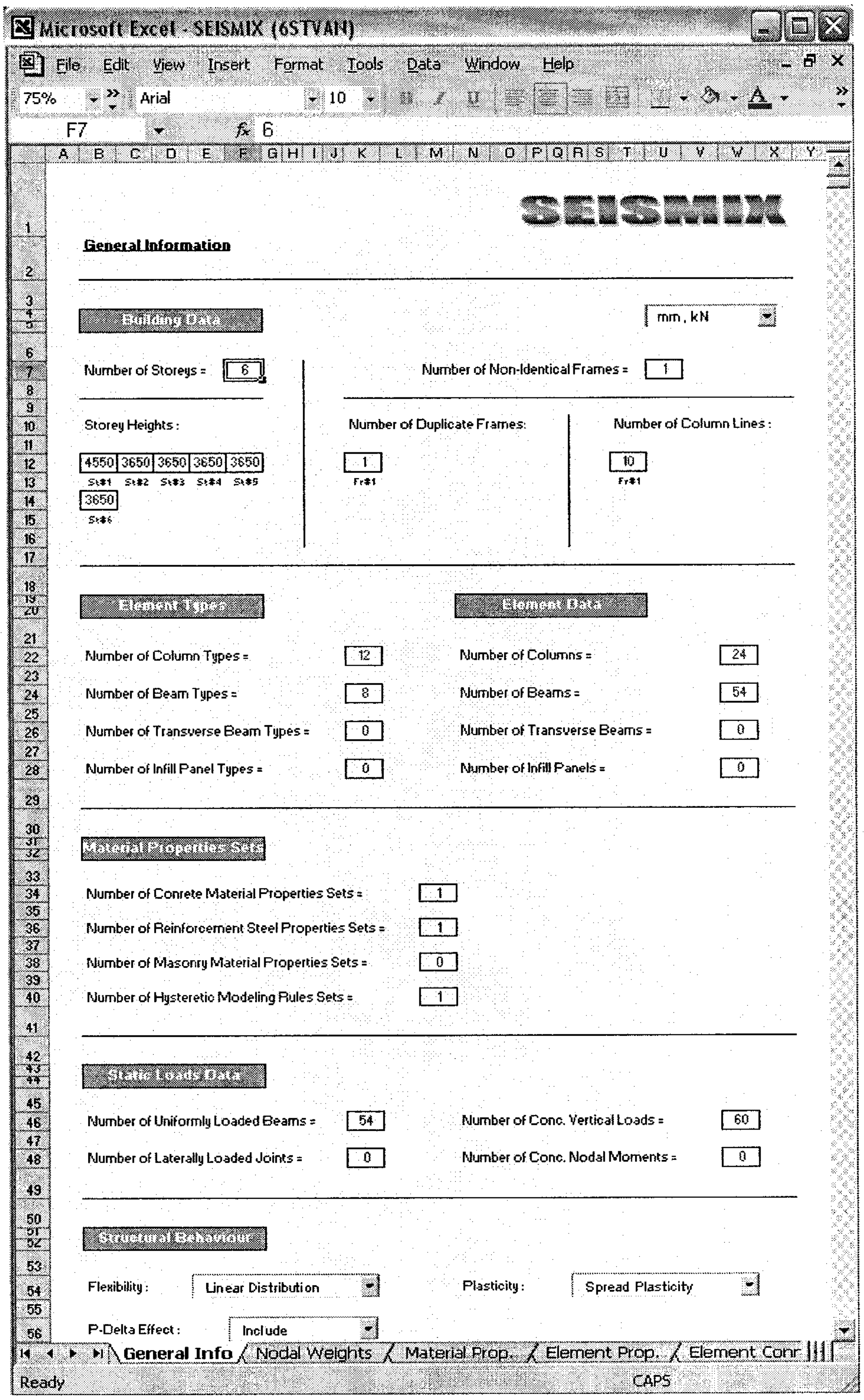
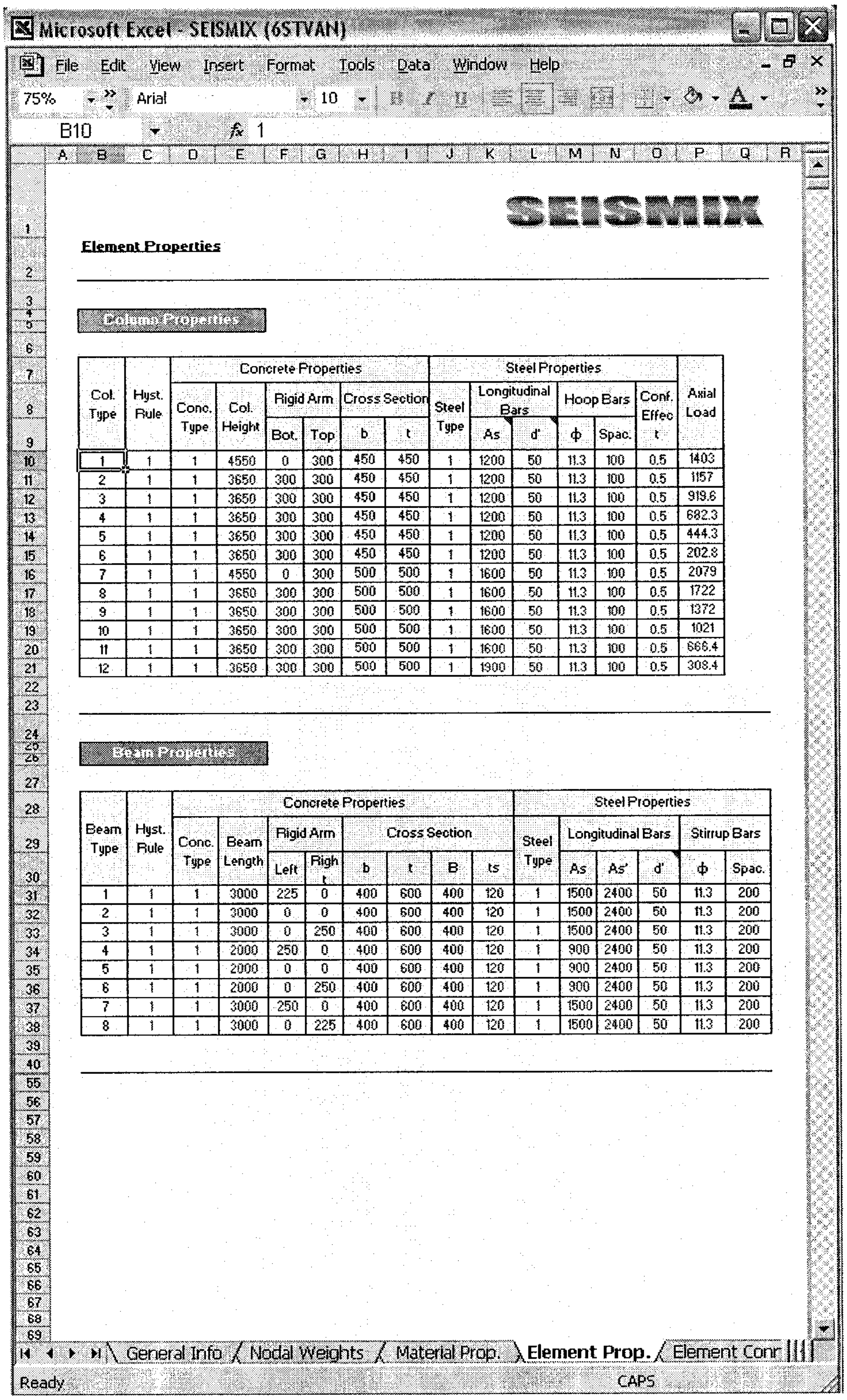


Figure 3.2: General Info Worksheet of the Pre-Processor

Figure 3.3 shows the "Element Properties" worksheet, which is divided into four sections, "Column Properties", "Beam Properties", "Transverse Beam Properties" and "Infill Panel Properties". Sections are automatically hidden if not needed. The first section, "Column Properties", is where the types of columns used are entered in a tabular form. The column type, hysteretic rule, concrete material type, column height, rigid arms, cross sectional dimensions, reinforcement steel type, longitudinal reinforcement and its position, transverse reinforcement (hoop bars) and its spacing, confinement effect and axial load are all parameters to be defined for each column type. The following Section, "Beam Properties", is where the types of beams are defined. Similarly, the beam type, hysteretic rule, concrete type, beam length, rigid arms, cross sectional dimensions, reinforcement steel types, longitudinal reinforcement and its spacing and transversal reinforcement (stirrup bars) and its spacing are all defined in this section. Transverse beams are defined in the following section, "Transverse Beam Properties", where the transverse beam type, the vertical stiffness, the torsional stiffness and the arm length are input for each transverse beam type. The final section in this worksheet is the "Infill Panels Properties" section. The infill panel type, masonry material property type, panel thickness, panel length, panel height, plastic moment of column, plastic moment of beam and plastic moment of joint are to be defined for each infill panel type.

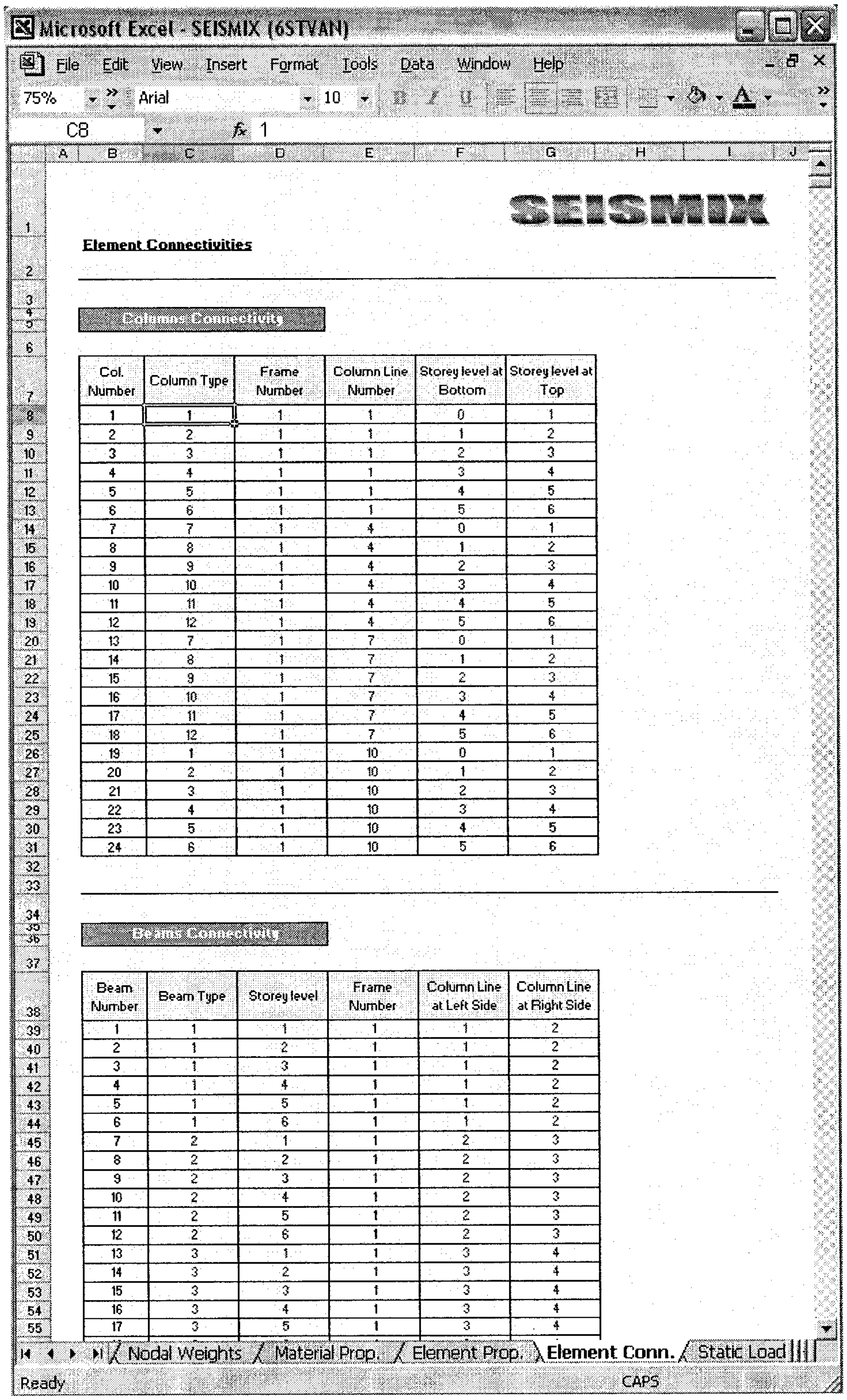


Col. Type	Hyst. Rule	Concrete Properties						Steel Properties						Axial Load
		Conc. Type	Col. Height	Rigid Arm		Cross Section		Steel Type	Longitudinal Bars		Hoop Bars		Conf. Effect	
				Bot.	Top	b	t		As	d'	ϕ	Spac.		
1	1	1	4550	0	300	450	450	1	1200	50	11.3	100	0.5	1403
2	1	1	3650	300	300	450	450	1	1200	50	11.3	100	0.5	1157
3	1	1	3650	300	300	450	450	1	1200	50	11.3	100	0.5	919.6
4	1	1	3650	300	300	450	450	1	1200	50	11.3	100	0.5	682.3
5	1	1	3650	300	300	450	450	1	1200	50	11.3	100	0.5	444.3
6	1	1	3650	300	300	450	450	1	1200	50	11.3	100	0.5	202.8
7	1	1	4550	0	300	500	500	1	1600	50	11.3	100	0.5	2079
8	1	1	3650	300	300	500	500	1	1600	50	11.3	100	0.5	1722
9	1	1	3650	300	300	500	500	1	1600	50	11.3	100	0.5	1372
10	1	1	3650	300	300	500	500	1	1600	50	11.3	100	0.5	1021
11	1	1	3650	300	300	500	500	1	1600	50	11.3	100	0.5	666.4
12	1	1	3650	300	300	500	500	1	1900	50	11.3	100	0.5	308.4

Beam Type	Hyst. Rule	Concrete Properties								Steel Properties					
		Conc. Type	Beam Length	Rigid Arm		Cross Section				Steel Type	Longitudinal Bars			Stirrup Bars	
				Left	Right	b	t	B	ts		As	As'	d'	ϕ	Spac.
1	1	1	3000	225	0	400	600	400	120	1	1500	2400	50	11.3	200
2	1	1	3000	0	0	400	600	400	120	1	1500	2400	50	11.3	200
3	1	1	3000	0	250	400	600	400	120	1	1500	2400	50	11.3	200
4	1	1	2000	250	0	400	600	400	120	1	900	2400	50	11.3	200
5	1	1	2000	0	0	400	600	400	120	1	900	2400	50	11.3	200
6	1	1	2000	0	250	400	600	400	120	1	900	2400	50	11.3	200
7	1	1	3000	250	0	400	600	400	120	1	1500	2400	50	11.3	200
8	1	1	3000	0	225	400	600	400	120	1	1500	2400	50	11.3	200

Figure 3.3: Element Properties Worksheet of the Pre-Processor

The "Element Connectivities" worksheet is shown in Figure 3.4. Similarly, this worksheet is divided into four sections, "Columns Connectivity", "Beams Connectivity", "Transverse Beams Connectivity" and "Infill Panels Connectivity". Only the relevant sections are available and the rest are automatically hidden. In this worksheet, the connectivity of all the elements used is defined. In the first section, "Columns Connectivity", the connectivity of the columns is defined. The column number, column type, frame number, column line, storey level at bottom and storey level at top are the parameters to be defined for all columns. Similarly, for all beams, the beam number, beam type, storey level, frame number and column line at left side and right side should be defined in the second section, "Beams Connectivity". The connectivity of the transverse beams is input in the following section. The transverse beam number, transverse beam type, storey level, frame number of the transverse beam, column line of the transverse beam, frame number of the connecting column and column line of the connecting column are the parameters to be input for all transverse beams. The connectivity of the infill panels is described in the last section. Infill panel number, frame number, infill panel type, storey level at top, storey level at bottom, column line at left side, column line at right side and beam type at top are the parameters needed to define the connectivity of each infill panel.



Microsoft Excel - SEISMIX (6STVAN)

SEISMIX

Element Connectivities

Columns Connectivity

Col. Number	Column Type	Frame Number	Column Line Number	Storey level at Bottom	Storey level at Top
1	1	1	1	0	1
2	2	1	1	1	2
3	3	1	1	2	3
4	4	1	1	3	4
5	5	1	1	4	5
6	6	1	1	5	6
7	7	1	4	0	1
8	8	1	4	1	2
9	9	1	4	2	3
10	10	1	4	3	4
11	11	1	4	4	5
12	12	1	4	5	6
13	7	1	7	0	1
14	8	1	7	1	2
15	9	1	7	2	3
16	10	1	7	3	4
17	11	1	7	4	5
18	12	1	7	5	6
19	1	1	10	0	1
20	2	1	10	1	2
21	3	1	10	2	3
22	4	1	10	3	4
23	5	1	10	4	5
24	6	1	10	5	6

Beams Connectivity

Beam Number	Beam Type	Storey level	Frame Number	Column Line at Left Side	Column Line at Right Side
1	1	1	1	1	2
2	1	2	1	1	2
3	1	3	1	1	2
4	1	4	1	1	2
5	1	5	1	1	2
6	1	6	1	1	2
7	2	1	1	2	3
8	2	2	1	2	3
9	2	3	1	2	3
10	2	4	1	2	3
11	2	5	1	2	3
12	2	6	1	2	3
13	3	1	1	3	4
14	3	2	1	3	4
15	3	3	1	3	4
16	3	4	1	3	4
17	3	5	1	3	4

Nodal Weights / Material Prop. / Element Prop. / Element Conn. / Static Load

Figure 3.4: Element Connectivities Worksheet of the Pre-Processor

Finally, Figure 3.5 shows the final worksheet labelled "Analysis". In this sheet, the type of analysis to be performed is chosen using a form which appears on the screen after pressing the "Analysis Settings" button. The user can then select between the options of inelastic incremental static analysis, monotonic push-over analysis, inelastic dynamic analysis and quasi-static cyclic analysis. Using this form, the user can also specify the number of incremental steps for the application of static loads and the steps between printing the output. Once the user chooses the type of analysis, the rows of the section corresponding to this analysis option appear on the screen and any other irrelevant rows belonging to another section are automatically hidden. This way, based on the analysis option chosen, the screen only contains the cells where data is to be entered for the selected analysis type. Figure 3.5 shows the "Dynamic Analysis" section, where the type of damping is to be specified using a dropdown list. Three types of damping are available, mass proportional, stiffness proportional and Rayleigh proportional. The damping coefficient, total duration of analysis, analysis time step, peak horizontal acceleration, peak vertical acceleration, time interval of the input wave, number of points in the wave file, wave title, filename of the horizontal component of the ground motion and filename of the vertical component are the parameters needed to define the inelastic dynamic analysis option. The user also has the choice whether to include the vertical component or not by checking or unchecking the "Include Vertical Component" checkbox. Finally, after all input data is entered, the user may create the input data file in the text format recognized by IDARC2D by pressing the "Write Input" button which allows a "Save As" window form to appear on the screen for the user to choose the directory where the file is to be saved.

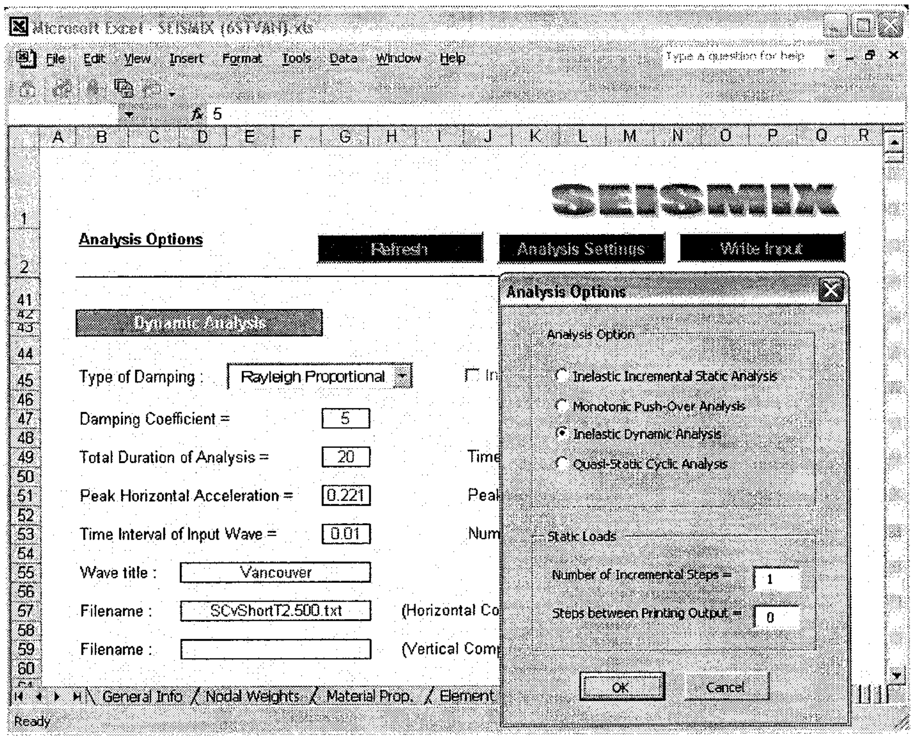


Figure 3.5: Analysis Worksheet of the Pre-Processor

3.3.2 Post-Processor for IDARC2D

Large output data is generated from the different types of analyses performed by IDARC2D. This data needs to be processed to extract the useful information and present the results in a compact form and plot relevant graphs. This job is done using a MATLAB-based Graphic User Interface (GUI) developed herein. The GUI is used to scan through the output files selected by the user and plot a number of response quantities like maximum inter-storey drifts, inter-storey drift time-history, displacement time-history, storey shear against storey drift, push-over curves and mode shapes. Figure 3.6 shows a screenshot of the MATLAB-based GUI post-processor. The post-processor is divided into four main sections corresponding to “Modal Analysis”, “Push-Over Analysis”, “Dynamic Analysis” and “Quasi-Static Analysis”. Figure 3.6 also shows two generated curves, the maximum inter-storey drift versus the storey level for a certain structure and the inter-storey drift versus time for a certain storey. The inter-storey drift in this figure is given in millimeters and represents the difference between the horizontal displacement of a certain storey and that of the storey below it.

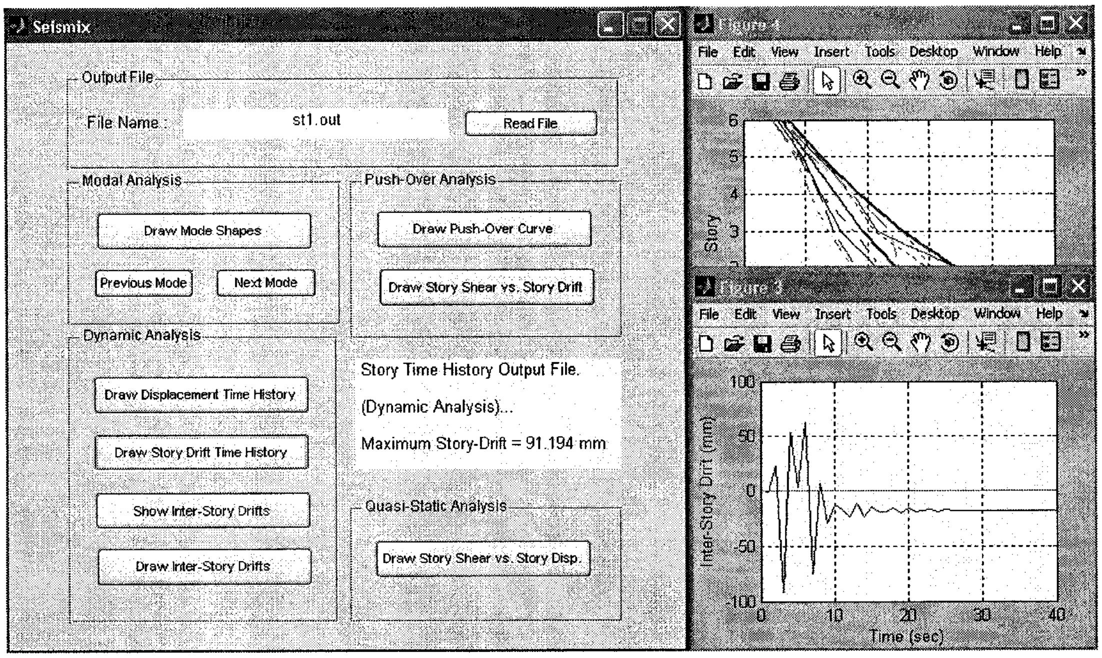


Figure 3.6: Screenshot of the MATLAB-based Post-Processor

3.4 OTHER COMPUTER APPLICATION

Computer use is essential in design of buildings nowadays, it's an important tool that facilitates the design process; not only in the analysis is computer software useful but also in the design of sections as well as in the calculation of loads and their combinations. Based on the NBCC 2005 provisions, several active Excel sheets are created to calculate the loads acting on the frames and combine them together using code specified combination factors. Static analysis output results are manipulated and used to design the cross sections of the frames also using Excel sheets. Two Excel files are created specifically for designing and calculating the capacity of rectangular reinforced concrete sections subjected to simple bending. Another Excel file is designed with the aid of VB scripts to generate the P-M interaction curves for rectangular reinforced concrete sections with uniform reinforcement. This data is required for designing the columns as well as satisfying the "strong column – weak beam" requirement and for data input in DRAIN-2DX.

Figure 3.7 shows a screenshot of the "Interaction Diagram" Excel worksheet. The advantage of this worksheet is that the user has the freedom of choosing the strength reduction factors for concrete and steel. The modulus of elasticity of concrete and steel are also specified by the user. The sectional dimensions and reinforcement are among the input parameters as well as position of the reinforcing steel. The stress-strain distribution of concrete is represented by an equivalent stress block and the user has the freedom of specifying the crushing strain of the concrete. After all data is entered, the P-M interaction curve is generated automatically by pressing the "Draw Diagram" button. First, the output data is written in another worksheet labelled "Output Data" which is linked to the "Interaction Diagram" chart.

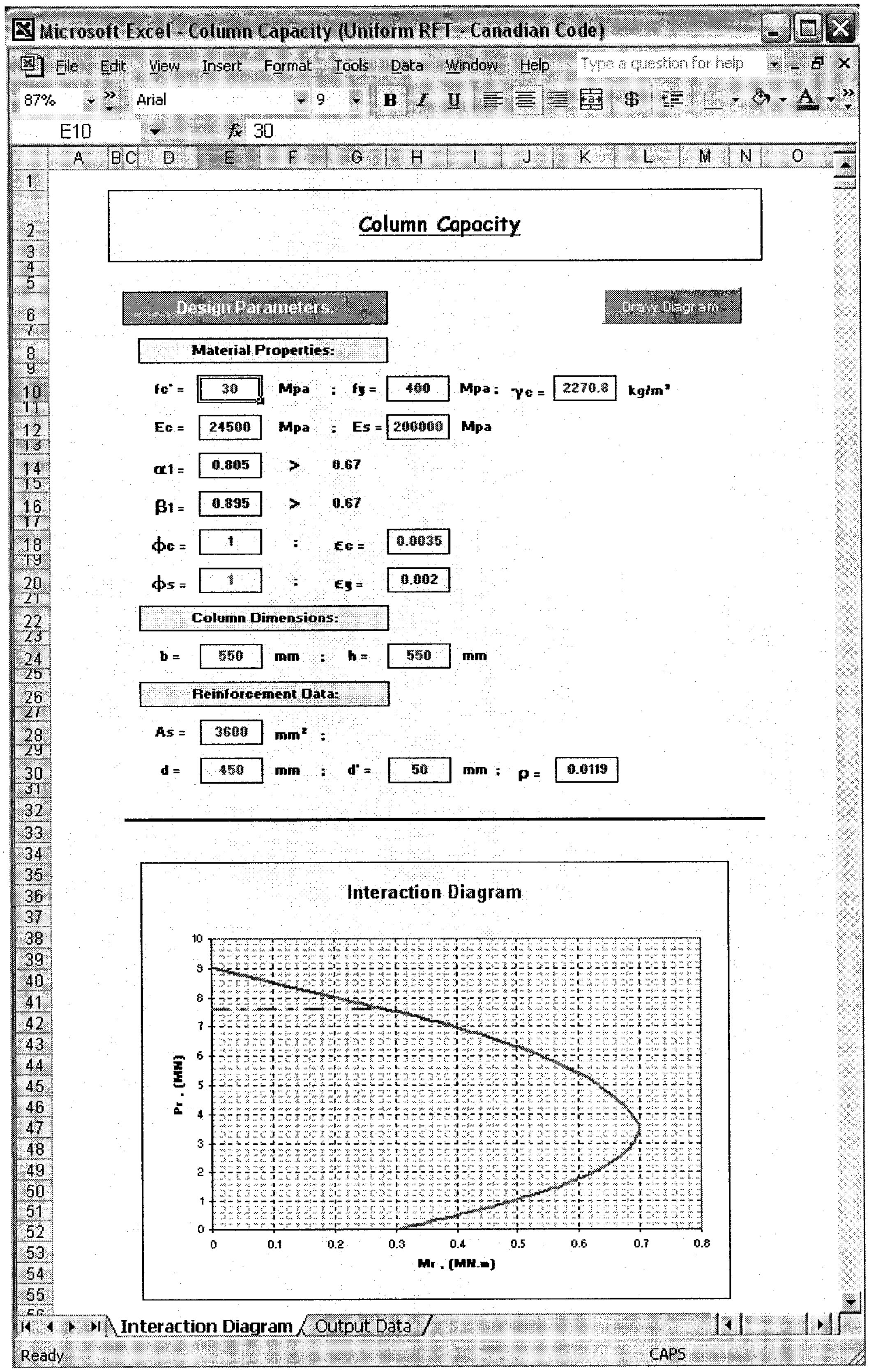


Figure 3.7: P-M Interaction Curve Generator Excel Worksheet

CHAPTER 4

4. DESIGN OF THE BUILDINGS

4.1 DESCRIPTION OF THE BUILDINGS

To evaluate the seismic performance of moment resisting frames, three office buildings are designed based on the NBCC 2005 provisions. The buildings are chosen to be located in Vancouver, British Columbia as it is a location of high seismicity in Canada. The layout plans of all three buildings are the same (Figure 4.1). The buildings are six, twelve and eighteen storeys high. The total height of the ground floor in each building is 4.85 m and the height of the rest of the floors is 3.65 m. Concrete moment resisting frames (CMRFs) are used in the short direction of the building to resist the lateral and gravity loads. The elevation of all three frames is shown in Figure 4.2. The layout of the buildings is symmetric and an interior frame close to the center of rigidity of the structure is considered such that the effect of accidental torsion can be ignored. In that case, the structure is reduced to a series of two-dimensional frames in two perpendicular directions. The lateral resistance of the building in the longer direction is not studied here. The focus of this research is mainly on the performance of the concrete moment resisting frames used in the weaker direction.

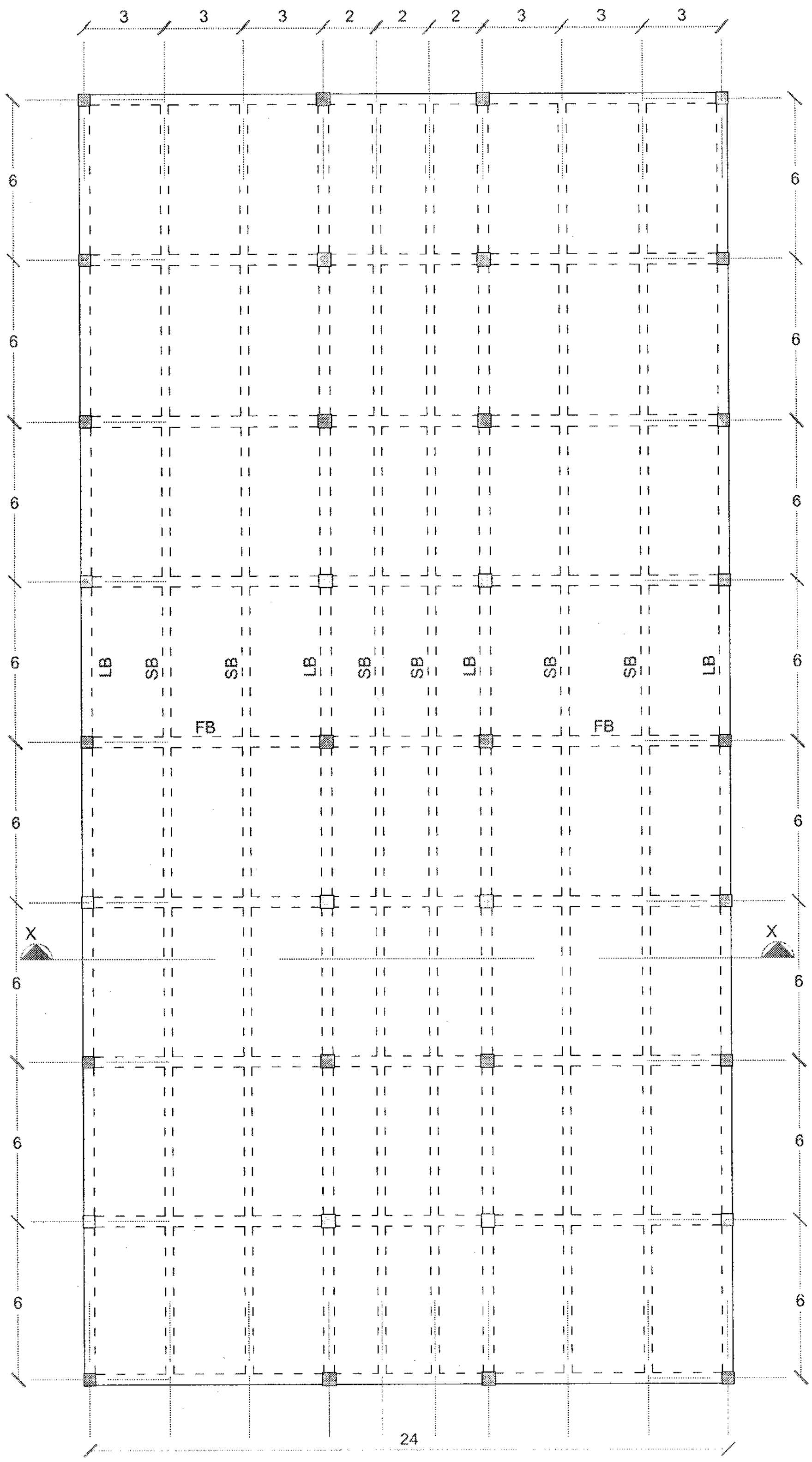


Figure 4.1: General Structural Plan (Dimensions are in Meters)

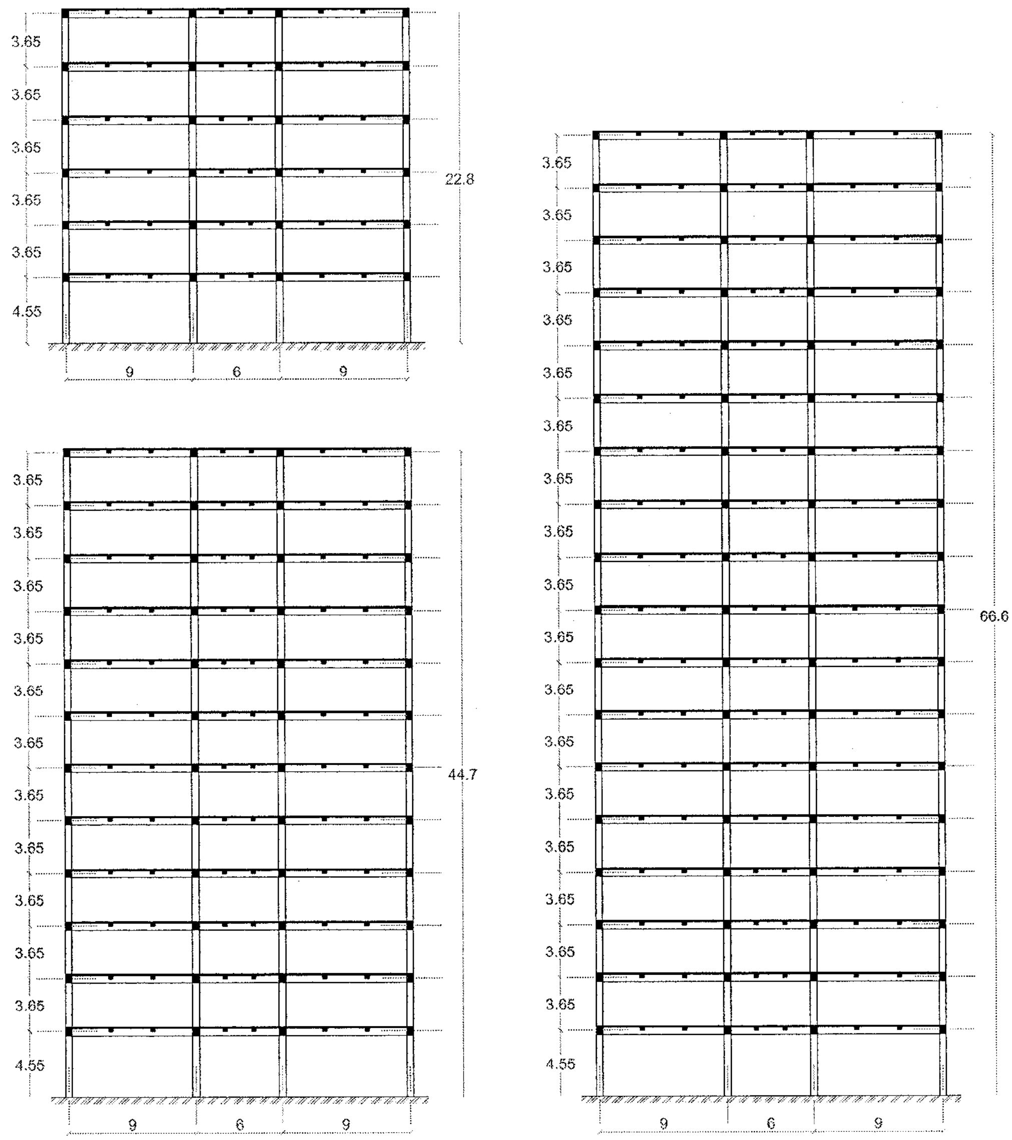


Figure 4.2: Elevation of the 6, 12 and 18 Storey CMRFs (Dimensions are in Meters)

4.2 DESIGN OF THE BUILDINGS

All frame beams are assumed to be (400x600 mm) for simplicity and the contribution of the slab in the resistance is neglected. All columns are assumed to be square columns for architectural purposes, but their dimensions and reinforcement may vary based on the design. To decrease the thickness of the reinforced concrete slab, six secondary beams are used as well as four longitudinal beams. The thickness of the slab is chosen to be 120 mm. All secondary beams are (300x350 mm) and all longitudinal beams are (400x600 mm). Those beams are not designed in this study, however their dimensions are assumed for gravity load estimation.

The unconfined compressive strength f_c' of the reinforced concrete is assumed to be 30 MPa. The initial modulus of elasticity of the concrete E_c is taken as 24500 MPa and the unit weight of reinforced concrete is assumed to be 24 kN/m^3. The reinforcing steel is assumed to have a yield strength f_y of 400 MPa. Also, masonry infill panels are assumed to be located in the interior bays of each storey of the three buildings and the unit weight of the infill material is taken as 24 kN/m^3 and the thickness of the panels is assumed to be 100 mm. However, the effect of the infill panels is not taken into account in the design. The infill panels are considered as non-structural elements. The design live loads used for all buildings are 2.4 kN/m^2 for all exterior bays and 4.8 kN/m^2 for all interior bays. The snow loads acting on the roof are assumed to be 2.2 kN/m^2. Three live load combinations were considered in the design as shown in Figure 4.3 to obtain the maximum possible straining actions and live load reduction factors were applied based on Equation 4.1:

$$LLRF = 0.3 + \sqrt{\frac{9.8}{A_t}} \tag{4.1}$$

where A_t is the tributary area in square meters.

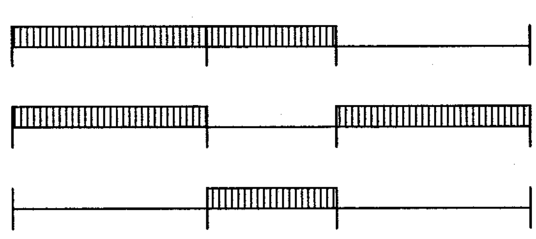


Figure 4.3: Live Load Combinations

Based on the NBCC 2005 seismic provisions, because the configuration of the six and twelve-storey buildings is simple and regular and the total height is below 60 m, the ESLM is used in the lateral load design. Since the eighteen-storey building is 66 m high, dynamic analysis should be used in the design. However, the ESLM is used to obtain preliminary design details. The base shear is calculated based on Equation 1.1. The frames are designed to be fully ductile therefore $R_d=4$ and $R_o=1.7$. The buildings sites are assumed to have firm soil conditions therefore $F_a=1$ and $F_v=1$. Since the buildings are office buildings the importance factor I is set equal to 1. The damping coefficient of the buildings is taken as 5% of the critical damping. Two loading combinations are used in the design, the first is the gravity load combination (1.25D+1.5L) and the second is the lateral load combination (D+0.5L+E), where D, L and E represent the dead, live and lateral earthquake induced loads, respectively.

The design fundamental period specified by the code is given by Equation 1.2. However, in most cases the fundamental period obtained from modal analysis is a lot larger than that obtained from the empirical equation suggested by the code. For this reason the NBCC 2005 allows the designer to increase the design fundamental period by 50% or up to the period obtained from modal analysis, whichever is less. This modification would increase the design period and thereby decrease the design base shear to allow for a more economic design.

4.2.1 Design Process

The initial design of the three buildings in this research is first performed based on the gravity load combination (1.25D+1.5L). Preliminary dimensions are assumed and the amount of reinforcement is obtained from the static analysis for gravity loads. This step is repeated until all cross sections are found to be adequate and able to resist this load combination with an acceptable amount of reinforcement. Following that, a modal analysis is performed to obtain the fundamental period of vibration of the structure. After that a comparison is made between the fundamental period of vibration obtained from the modal analysis and that calculated from the empirical code formula as given in Equation 1.2, and the design period is chosen based on the condition previously mentioned. The equivalent static lateral forces are then obtained using Equation 1.1, and the design is checked against the lateral load combination (D+0.5L+E). The estimated inter-storey drifts obtained from the static analysis and modified for the inelastic response by multiplying by R_d and R_o are checked to ensure that the drift at each storey level does not exceed 2.5%. Otherwise, the design should be updated and the abovementioned steps must be repeated. However, because the code (NBCC 2005) requires that the eighteen storey-building should be designed using dynamic analysis, this condition is not taken into account instead the inter-storey drifts of the building are later checked using dynamic analysis for ground acceleration time-history records. Finally, after the lateral load static design is performed, the rigid connections in the building are checked for the "strong column – weak beam" criterion, as required by the code.

To account for the cracking in concrete, the moments of inertia (I) are reduced for beams and columns. The reduction factor used in the case of beams is 0.4 while in the case of columns it is 0.7 for bottom columns and 0.6 for upper columns. This reduction is only done in the design process. However, a more accurate value obtained from IDARC2D is used in the performance evaluation.

4.2.2 Design Base Shear

The design base shear values of the 6, 12 and 18 storey buildings are found to be 294.33, 360.7 and 529.51 kN respectively and the design base shear coefficients *V/W* are 4.59%, 2.78% and 2.65% respectively. The design base shear coefficient is expressed as the ratio of the design base shear *V* to the dead load weight of the structure plus 25% of the snow load *W*. The lateral load distribution of the design base shear for the three buildings is shown in Figure 4.4.

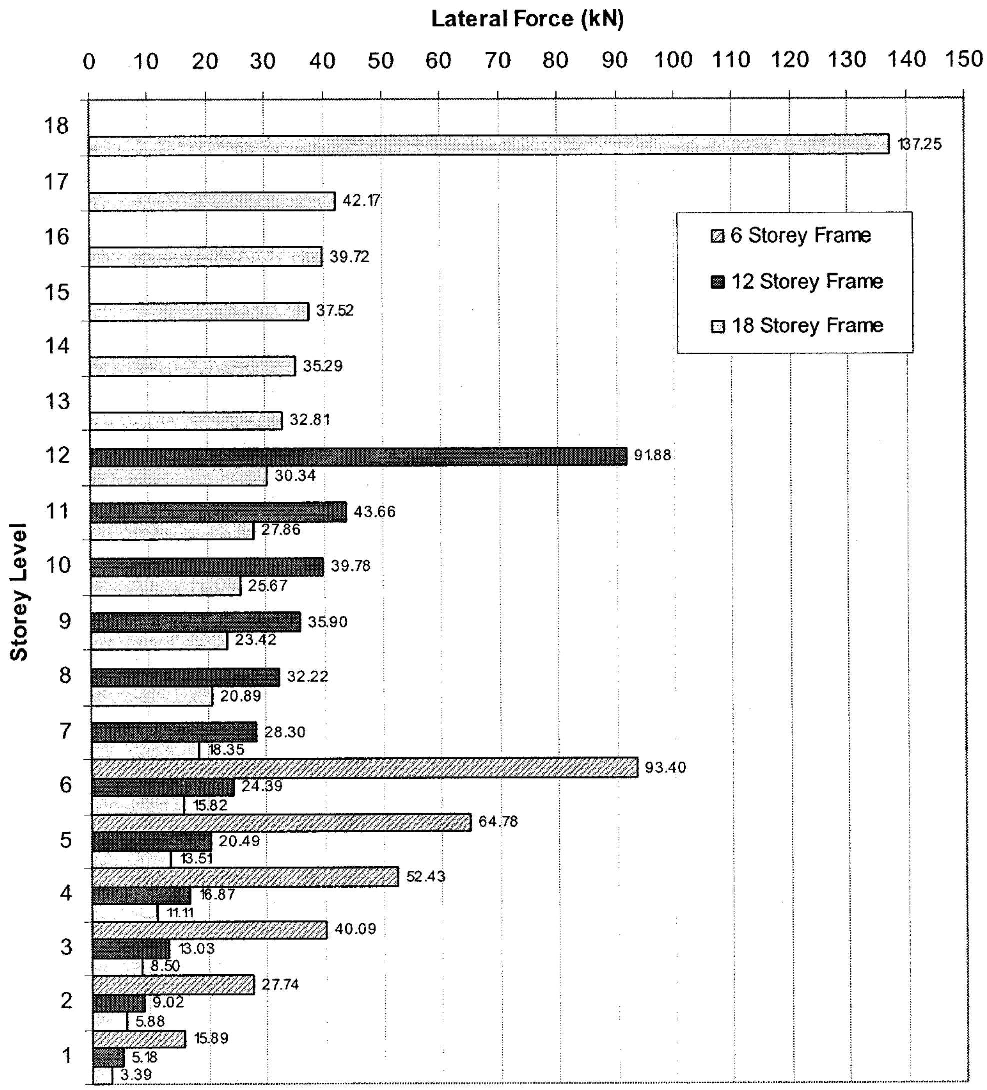


Figure 4.4: Design Base Shear Distribution on the 6, 12 and 18 Storey CMRFs

4.2.3 Inter-Storey Drifts

The estimated inter-storey drifts from the static analysis, as mentioned before, are checked to ensure that they do not exceed 2.5% of the storey height for the six and twelve-storey buildings. As for the eighteen-storey building, the inter-storey drifts are plotted just for comparison with the inter-storey drifts obtained from the dynamic analysis. Figure 4.5 shows the estimated inter-storey drifts for the three buildings under consideration. The maximum inter-storey drift ratio of the six-storey building occurs at the second storey level and is equal to 2.12% while the maximum inter-storey drift of the twelve-storey building is 2.31% and occurs at the fifth storey level. Both values are less than 2.5%, which means that the design is satisfactory from this point of view. On the other hand, however, the maximum inter-storey drift of the eighteen-storey building is 3.17% which occurs at the sixth storey level. Although this value exceeds 2.5%, the design of this building as specified by the NBCC 2005 needs to include dynamic analysis as the ESLM is technically not allowed for designing a structure with such height. Therefore, nonlinear dynamic analysis will be used to evaluate the validity of the design of the eighteen-storey building instead of the equivalent inter-storey static drifts.

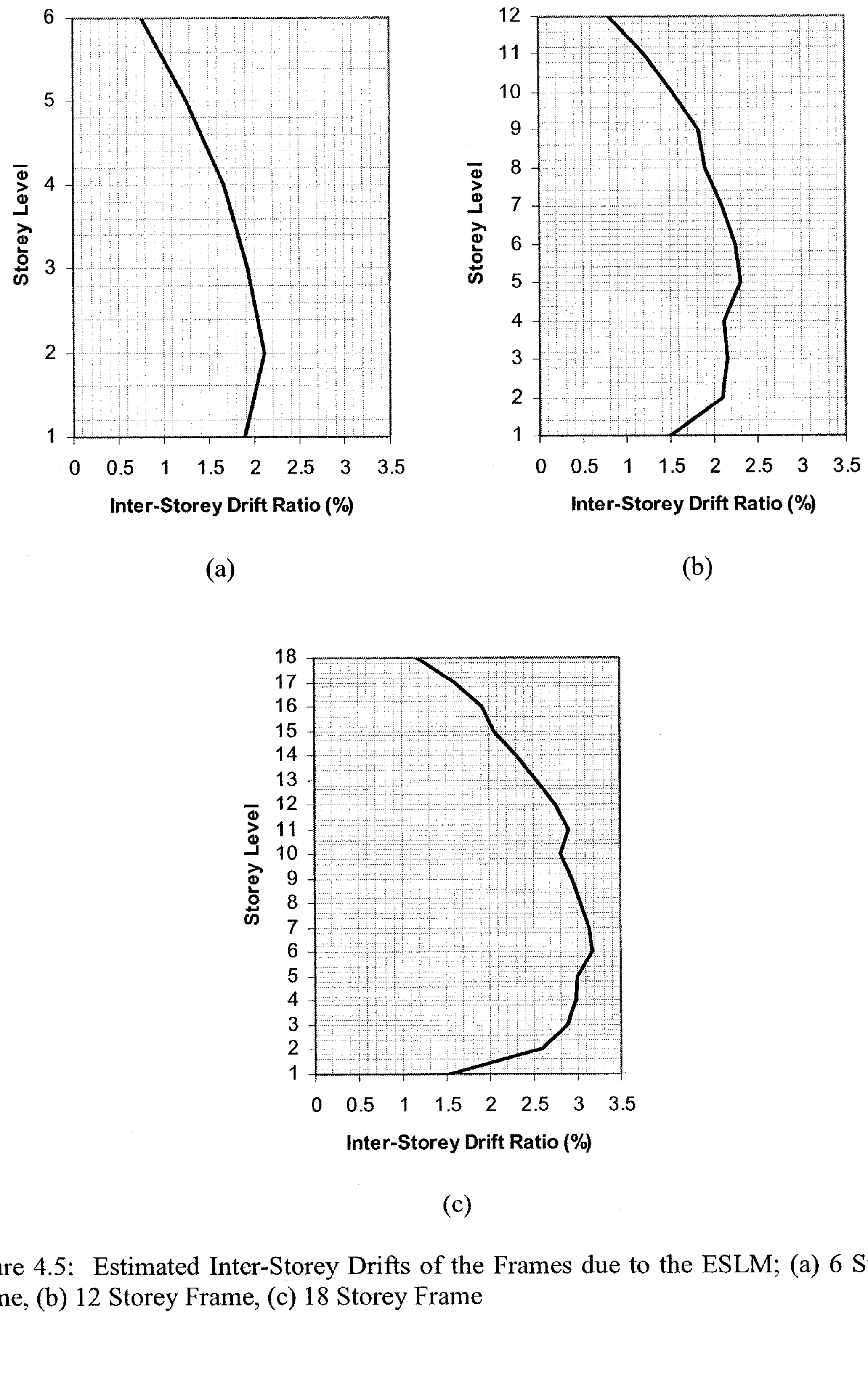


Figure 4.5: Estimated Inter-Storey Drifts of the Frames due to the ESLM; (a) 6 Storey Frame, (b) 12 Storey Frame, (c) 18 Storey Frame

4.2.4 "Strong Column – Weak Beam" Requirement

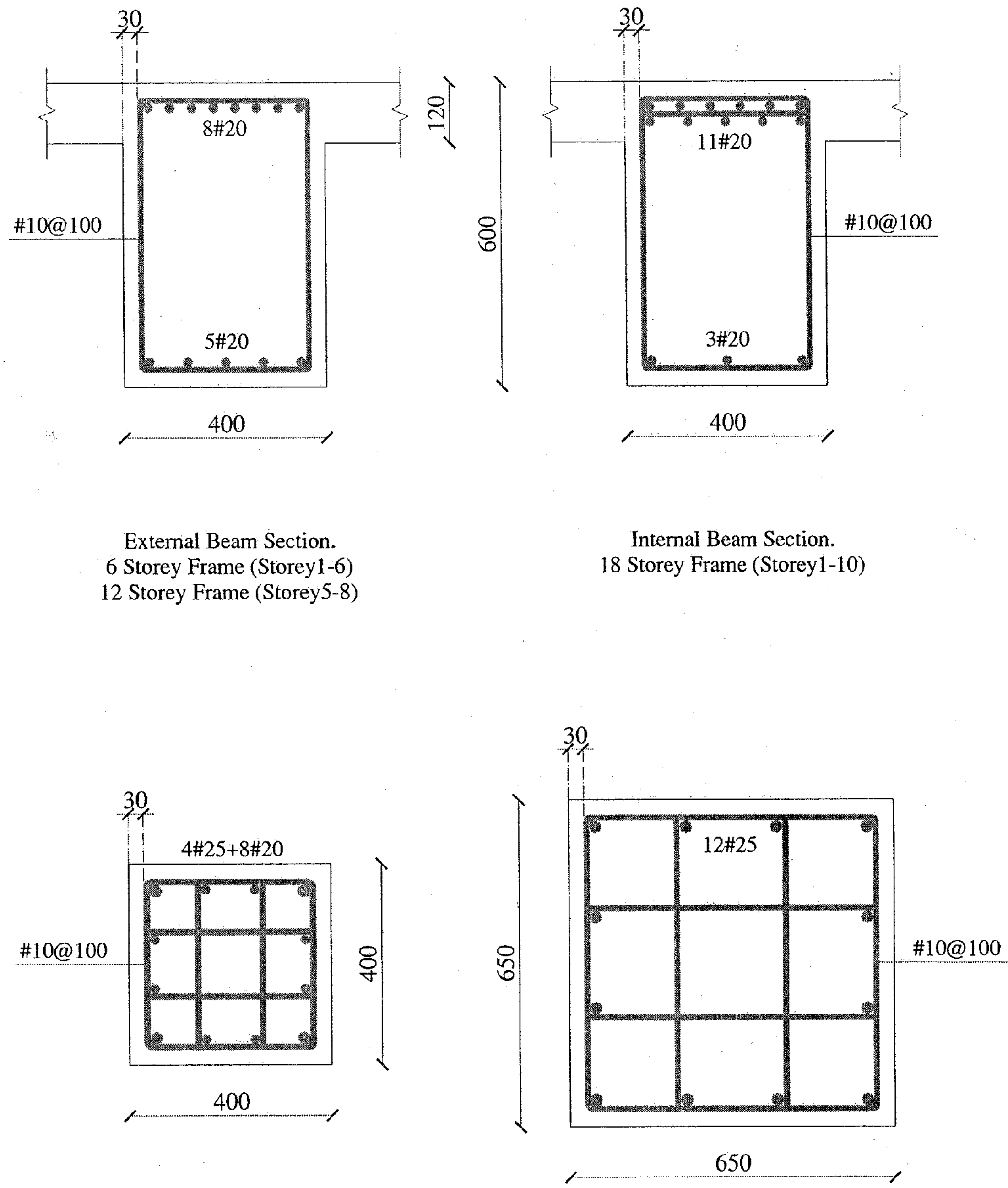
Based on the code provisions, plastic hinges should occur in horizontal members of the structure before occurring in vertical members to prevent the soft storey mechanism, which leads to a sudden failure, from occurring. This is why for a given joint the yielding capacity of the column(s) should be higher than that of the beam(s). To ensure that columns are stronger than beams, Equation 4.2 should be satisfied.

$$\sum M_{nc} \geq \sum M_{pb} \tag{4.2}$$

where $\sum M_{nc}$ is the sum of the nominal flexural resistances of the columns meeting at a specific joint while $\sum M_{pb}$ is the sum of the probable flexural resistances of the beams meeting at that same joint. This check is performed at all joints after the design based on gravity loads is checked against lateral loads. If any of the checks performed fails, the cross sections and/or reinforcement should be updated and the analysis and design should be repeated and rechecked in an iterative manner.

4.2.5 Final Design Details

After a couple of iterations, the final cross sections are obtained and their details are listed in Tables 4.1 and 4.2. Figure 4.6 also shows a sample of these cross sections.



External Beam Section.
6 Storey Frame (Storey1-6)
12 Storey Frame (Storey5-8)

Internal Beam Section.
18 Storey Frame (Storey1-10)

External Column Section.
12 Storey Frame (Storey12)

Internal Column Section.
18 Storey Frame (Storey6-10)

Figure 4.6: Sample Cross Sections (Dimensions are in Millimeters)

Table 4.1: Final Cross Sections of Beams

Storey #	External Beams			Internal Beams		
	6 Storey	12 Storey	18 Storey	6 Storey	12 Storey	18 Storey
1	8#20 Top	9#20 Top	10#20 Top	8#20 Top	8#20 Top	11#20 Top
	5#20 Bot.	5#20 Bot.	5#20 Bot.	3#20 Bot.	4#20 Bot.	3#20 Bot.
2	8#20 Top	9#20 Top	10#20 Top	8#20 Top	8#20 Top	11#20 Top
	5#20 Bot.	5#20 Bot.	5#20 Bot.	3#20 Bot.	4#20 Bot.	3#20 Bot.
3	8#20 Top	9#20 Top	10#20 Top	8#20 Top	8#20 Top	11#20 Top
	5#20 Bot.	5#20 Bot.	5#20 Bot.	3#20 Bot.	4#20 Bot.	3#20 Bot.
4	8#20 Top	9#20 Top	10#20 Top	8#20 Top	8#20 Top	11#20 Top
	5#20 Bot.	5#20 Bot.	5#20 Bot.	3#20 Bot.	4#20 Bot.	3#20 Bot.
5	8#20 Top	8#20 Top	10#20 Top	8#20 Top	8#20 Top	11#20 Top
	5#20 Bot.	5#20 Bot.	5#20 Bot.	3#20 Bot.	4#20 Bot.	3#20 Bot.
6	8#20 Top	8#20 Top	10#20 Top	8#20 Top	8#20 Top	11#20 Top
	5#20 Bot.	5#20 Bot.	5#20 Bot.	3#20 Bot.	4#20 Bot.	3#20 Bot.
7		8#20 Top	10#20 Top		8#20 Top	11#20 Top
		5#20 Bot.	5#20 Bot.		4#20 Bot.	3#20 Bot.
8		8#20 Top	10#20 Top		8#20 Top	11#20 Top
		5#20 Bot.	5#20 Bot.		4#20 Bot.	3#20 Bot.
9		7#20 Top	10#20 Top		6#20 Top	11#20 Top
		5#20 Bot.	5#20 Bot.		3#20 Bot.	3#20 Bot.
10		7#20 Top	10#20 Top		6#20 Top	11#20 Top
		5#20 Bot.	5#20 Bot.		3#20 Bot.	3#20 Bot.
11		7#20 Top	9#20 Top		6#20 Top	9#20 Top
		5#20 Bot.	5#20 Bot.		3#20 Bot.	3#20 Bot.
12		7#20 Top	9#20 Top		6#20 Top	9#20 Top
		5#20 Bot.	5#20 Bot.		3#20 Bot.	3#20 Bot.
13			9#20 Top			9#20 Top
			5#20 Bot.			3#20 Bot.
14			9#20 Top			9#20 Top
			5#20 Bot.			3#20 Bot.
15			9#20 Top			9#20 Top
			5#20 Bot.			3#20 Bot.
16			7#20 Top			6#20 Top
			5#20 Bot.			3#20 Bot.
17			7#20 Top			6#20 Top
			5#20 Bot.			3#20 Bot.
18			7#20 Top			6#20 Top
			5#20 Bot.			3#20 Bot.

* All cross sections are 400x600 mm

Table 4.2: Final Cross Sections of Columns

Storey #	External Columns			Internal Columns		
	6 Storey	12 Storey	18 Storey	6 Storey	12 Storey	18 Storey
1	450x450	550x550	650x650	500x500	600x600	750x750
	12#20	12#20	4#25+8#20	8#20+4#25	12#25+4#20	20#25
2	450x450	550x550	650x650	500x500	600x600	750x750
	12#20	12#20	4#25+8#20	8#20+4#25	8#25+4#20	16#25
3	450x450	550x550	650x650	500x500	600x600	750x750
	12#20	12#20	4#25+8#20	8#20+4#25	8#25	12#25
4	450x450	550x550	650x650	500x500	600x600	750x750
	12#20	12#20	4#25+8#20	8#20+4#25	8#25	12#25
5	450x450	450x450	650x650	500x500	550x550	750x750
	12#20	12#20	4#25+8#20	8#20+4#25	8#25	12#25
6	450x450	450x450	550x550	500x500	550x550	650x650
	12#20	12#20	4#25+4#20	12#20+4#25	8#25	12#25
7		450x450	550x550		550x550	650x650
		12#20	4#25+4#20		8#25	12#25
8		450x450	550x550		550x550	650x650
		12#20	4#25+4#20		8#25	12#25
9		400x400	550x550		500x500	650x650
		12#20	4#25+4#20		8#25	12#25
10		400x400	550x550		500x500	650x650
		12#20	4#25+4#20		8#25	12#25
11		400x400	500x500		500x500	550x550
		12#20	4#25+4#20		8#25	8#25
12		400x400	500x500		500x500	550x550
		4#25+8#20	4#25+4#20		8#25	8#25
13			500x500			550x550
			4#25+4#20			8#25
14			500x500			550x550
			4#25+4#20			8#25
15			500x500			550x550
			4#25+4#20			8#25
16			450x450			500x500
			4#25+4#20			8#25
17			450x450			500x500
			4#25+4#20			8#25
18			450x450			500x500
			4#25+8#20			8#25+4#20

CHAPTER 5

5. PERFORMANCE EVALUATION

5.1 INTRODUCTION

The seismic performance of the moment resisting frames is evaluated using nonlinear dynamic analysis as well as nonlinear static push-over analysis. Static push-over analysis is a static analysis method used to estimate the structure's strength and ductility capacities against lateral loading. This is done by applying static incremental lateral loads on the structure until a certain level of displacement is reached (displacement control) or until a certain level of applied force is reached (force control).

Nonlinear dynamic analysis is used to assess the structural seismic performance. This type of analysis is performed for UHS-2500 records as well as UHS-1000 and UHS-500 records. The analysis is done using eight, four and four synthesized records corresponding to a UHS-2500, UHS-1000 and UHS-500, respectively. Also sixteen real records scaled to represent a UHS-2500 are used in the analysis.

The dynamic performance evaluation consists of assessing the damage that the building has suffered during and after an earthquake or a ground motion. This is done by calculating damage indices and response parameters. The inter-storey drift is an important response parameter that reflects the amount of lateral deformation that occurred to each floor in the structure. It's an important indicator of the seismic performance of the structure. Based on the NBCC 2005 seismic provisions, buildings should be able to attain the "Collapse Prevention" performance level in the case of a UHS-2500 event. In that case, the inter-storey drift is limited to 2.5%.

5.2 METHODS AND MODELS USED FOR PERFORMANCE EVALUATION

While static push-over analysis is used here for the seismic performance evaluation, the main tool for performance evaluation is the nonlinear dynamic analysis of the structures for multiple ground acceleration records. The parameters utilized in quantifying and qualifying the performance and damage of the structures based on the dynamic analysis are mainly the inter-storey drifts and damage indices. The inter-storey drift which represents the relative displacement between a storey and the storey below it, expressed as a percentage of the storey height, is one of the most important parameters used in seismic performance evaluation. Damage indices, although not as widely used as inter-storey drifts, are also used in the damage assessment of the buildings. The damage indices used here are produced by IDARC2D and are based on the modified version of the Park and Ang damage model (Kunnath *et al.*, 1992) explained in Chapter 2. Equations 2.2 through 2.4 are used to produce the damage indices.

As stated earlier, the analysis models of the buildings considered in this study are two-dimensional concrete moment resisting frames. Two types of models are considered for each frame, bare frame and infilled frame models. The bare frame model consists of the main structural elements considered in the design process (i.e. beams and columns). However, to take into account the effect of the non-structural elements on the stiffness of the structure and the seismic behaviour and performance in general, masonry infill panels are considered in the analysis of the infilled frame model. The infill panels as mentioned in Chapter 4 are located in all the interior bays of the three buildings.

The analysis of the infilled frame is done with IDARC2D and DRAIN-2DX. The infill panels are modeled in DRAIN-2DX using diagonal compression struts, an approach widely used. Although IDARC2D has an infill panel element, it internally converts the infill panels to diagonal compression struts. However, the results of the infilled frame model analysis produced by IDARC2D seem to be inconsistent. For this reason, only the results of the DRAIN-2DX analysis are presented here and those of IDARC2D have been discarded. Figure 5.1 shows the generic elevation of the infilled frames and the equivalent diagonal compression struts model used in the DRAIN-2DX analysis. The parameter n shown in Figure 5.1 represents the number of storeys of the building.

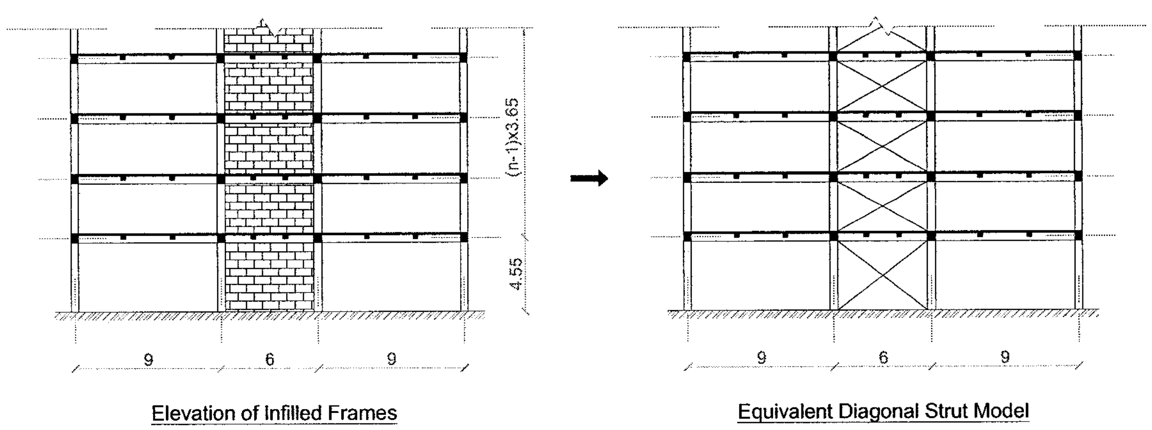


Figure 5.1: Generic Elevation of Infilled Frames

The struts are assumed to withstand only compressive stresses while being completely unable to resist any tensile stresses. The compressive strength of the masonry material is assumed to be 8.6 MPa. The modulus of elasticity of the masonry material E_m is calculated based on Equation 5.1:

$$E_m = kf_m \tag{5.1}$$

where k is a constant equal to 500 for masonry.

The provisions for the effective width calculation of a diagonal compression strut used to represent an infill panel in this analysis are taken from the FEMA-306 report (1998) and are based on the work of Mainstone and Weeks (1970) and Mainstone (1971). The infill panel in this case is assumed to be in full contact with the frame elements surrounding it. Equations 5.2 through 5.4 show how the effective width a is calculated:

$$a = 0.175(\lambda_1 h_{col})^{-0.4} r_i \tag{5.2}$$

where

$$\lambda_1 = \left[\frac{E_m t_i \sin 2\theta}{4 E_f I_{col} h_i} \right]^{1/4} \tag{5.3}$$

and h_{col} is the column height between the centerlines of the upper and lower beams, h_i is the height of the infill panel, r_i is the diagonal length of the infill panel, t_i is thickness of the equivalent strut which is the same as that of the infill panel, E_f is the modulus of elasticity of the frame material, I_{col} is the moment of inertia of the column and θ is the angle of inclination of the strut expressed as:

$$\theta = \tan^{-1}\left(\frac{h_i}{L_i} \right) \tag{5.4}$$

where L_i is the length of the infill panel.

While the factored resistance of the structural elements is considered in the design by applying strength reduction factors to the concrete and to the steel as specified by the code, the evaluation analyses are conducted using the nominal strength of the materials and no reduction is applied. This is done to obtain a better assessment of the actual behaviour of the materials and the structure as a whole.

5.3 ANALYSIS RESULTS

5.3.1 Modal Analysis

As mentioned before, the modal analysis is carried out during the design process to obtain the values of the fundamental periods of vibration of the structures and use it to calculate the equivalent lateral loads. Modal analysis is also used in scaling the ground acceleration records used in the nonlinear dynamic analysis as the first two periods are required in this process. A comparison of the dynamic characteristics of the frames with and without the inclusion of the infill panels could be made using modal analysis by comparing the periods of vibration which reflect the stiffness of the structures.

As clear from the fundamental period of vibration values shown in Table 5.1, the periods of the infilled frames are a lot closer to those specified by the empirical formula of the NBCC 2005 (Equation 1.2) than the periods of the bare frames. The first four mode shapes of the three bare frame models are shown in Figure 5.2 through 5.4.

Table 5.1: Fundamental Periods of Vibration

	Fundamental Period (s)		
	6 Storey	12 Storey	18 Storey
Bare Frame	1.3	2.31	3.32
Infilled Frame	0.86	1.69	2.55
NBCC 2005, T_a	0.78	1.3	1.75
$1.5T_a$	1.17	1.95	2.63

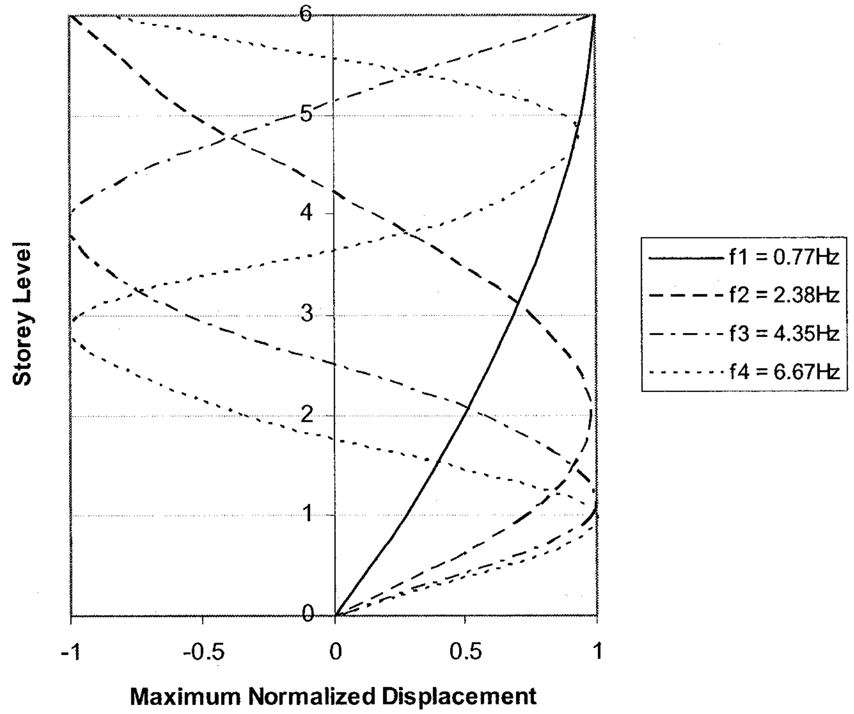


Figure 5.2: Mode Shapes of the 6 Storey Bare Frame (First 4 Modes)

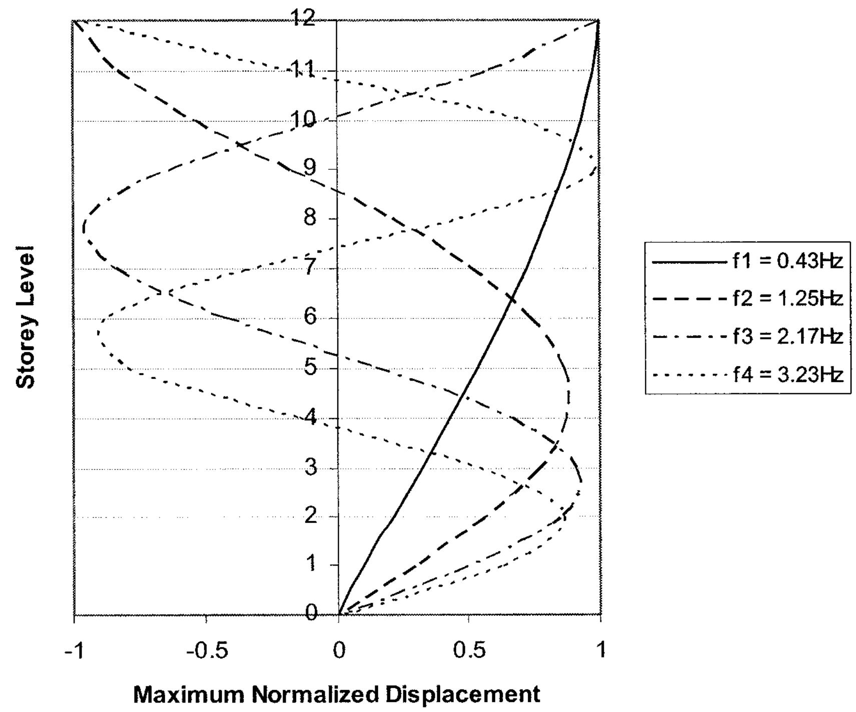


Figure 5.3: Mode Shapes of the 12 Storey Bare Frame (First 4 Modes)

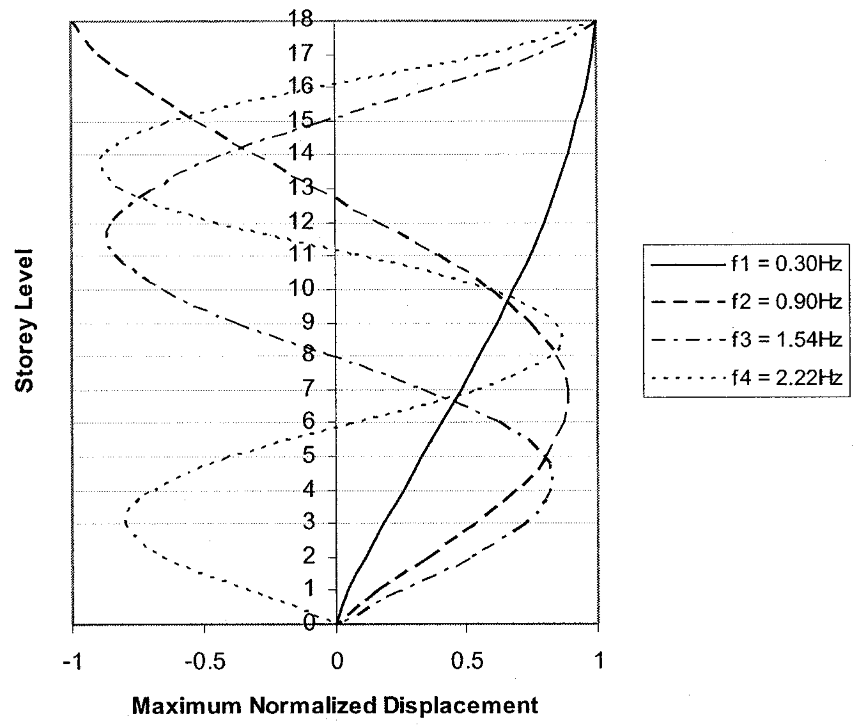


Figure 5.4: Mode Shapes of the 18 Storey Bare Frame (First 4 Modes)

5.3.2 Static Push-Over Analysis

The push-over analysis is performed using the two different computer programs previously mentioned: IDARC2D and DRAIN-2DX. The analysis is done for all three buildings, for the bare and infilled frames. A comparison is made between the responses of the structures with and without the inclusion of the P-Delta effect. The P-Delta effect is a second order (nonlinear) effect that occurs in axially loaded members, usually columns. This effect increases the stresses on the members and on the whole structure and is a result of axial forces (P) which initially are concentric, however under loading become eccentric. These eccentricities (Delta) are caused by lateral deformations as well as buckling. The P-Delta effect depends on the magnitude of the axial loads and on the slenderness of the elements and of the structure as a whole. One problem noticed is that the P-Delta effect option in the static push-over analysis of IDARC2D is not working properly. IDARC2D produces the same push-over results with and without the inclusion of the P-Delta effect, which implies that the P-Delta effect option is not properly taken into account.

The push-over analysis carried out in this study is done using the same distribution of the design base shear given out by the NBCC 2005. The results of these analyses are shown in the following figures. The failure point is shown on each curve and corresponds to the point of instability or to the point of 2.5% inter-storey drift, whichever occurs first. The point of instability is defined as the point on the push-over curve where the slope of the curve turns negative. In cases where the P-Delta effect is included, the failure point is taken as the instability point as no storey has reached the 2.5% inter-storey drift ratio defined by the code as the maximum allowable inter-storey drift. However, in all cases where the P-Delta effect is neglected, the failure point corresponds to the first point where any storey reaches 2.5% inter-storey drift. Figures 5.5 through 5.7 show all the push-over curves obtained from DRAIN-2DX and IDARC2D for all three structures.

It's clear from the results that the contribution of the infill panels to the overall stiffness of the structures which is expressed in the initial slope of the push-over curve is major. The overall stiffness increases 49%, 68% and 38% for the six, twelve and eighteen-storey building, respectively. The failure loads or the load capacities increase as well with the inclusion of infill panels in the analysis. For the six-storey frame the failure load increases 1.81 times with the P-Delta effect and 1.66 times without the P-Delta effect and for the twelve-storey frame 1.79 times with the P-Delta effect and 1.62 times without the P-Delta effect and for the eighteen-storey frame 1.78 times with the P-Delta effect and 1.52 times without the P-Delta effect. However, the (roof) displacement capacities of the buildings slightly decrease with the inclusion of the infill panels. Finally, by idealizing the push-over curves to bilinear curves and locating the equivalent yield point at the end of the linear (elastic) response, the ductility ratios of the buildings are approximately estimated by dividing the displacement capacity by the equivalent yield displacement. For the six-storey building, it is noticed that the estimated ductility capacity of the bare frame is 3.14 and 4.15 when including and excluding the P-Delta effect, respectively. While, for the infilled frames the ductility capacity is estimated to be 2.36 and 3.23 with the inclusion and exclusion of the P-Delta effect in the analysis, respectively. As for the twelve-storey building, the ductility capacity of the bare frame was found to be 2.43 and 4.36 while that of the infilled frame was found to be 1.53 and 3.13 in the case of the P-Delta effect included and excluded, respectively. Finally, for the eighteen-storey building, the estimated ductility capacity of the bare frame is 2.06 and 3.67 while that of the infilled frame is 1.7 and 3.23 when including and excluding the P-Delta effect, respectively. The ductility capacities of the buildings, as noticed, decrease when the infill panels are considered in the analysis. The P-Delta effect also, when considered, produces lower ductility ratios.

The yielding sequence of the frames is checked after the analysis to ensure that the "strong column – weak beam" requirement is fulfilled. The yielding of beams occurs before the yielding of columns in all bare frame cases, however, in the case of infilled frames this is not always the case. This is due to the fact that the infill panels were not taken into account in the "strong column – weak beam" design.

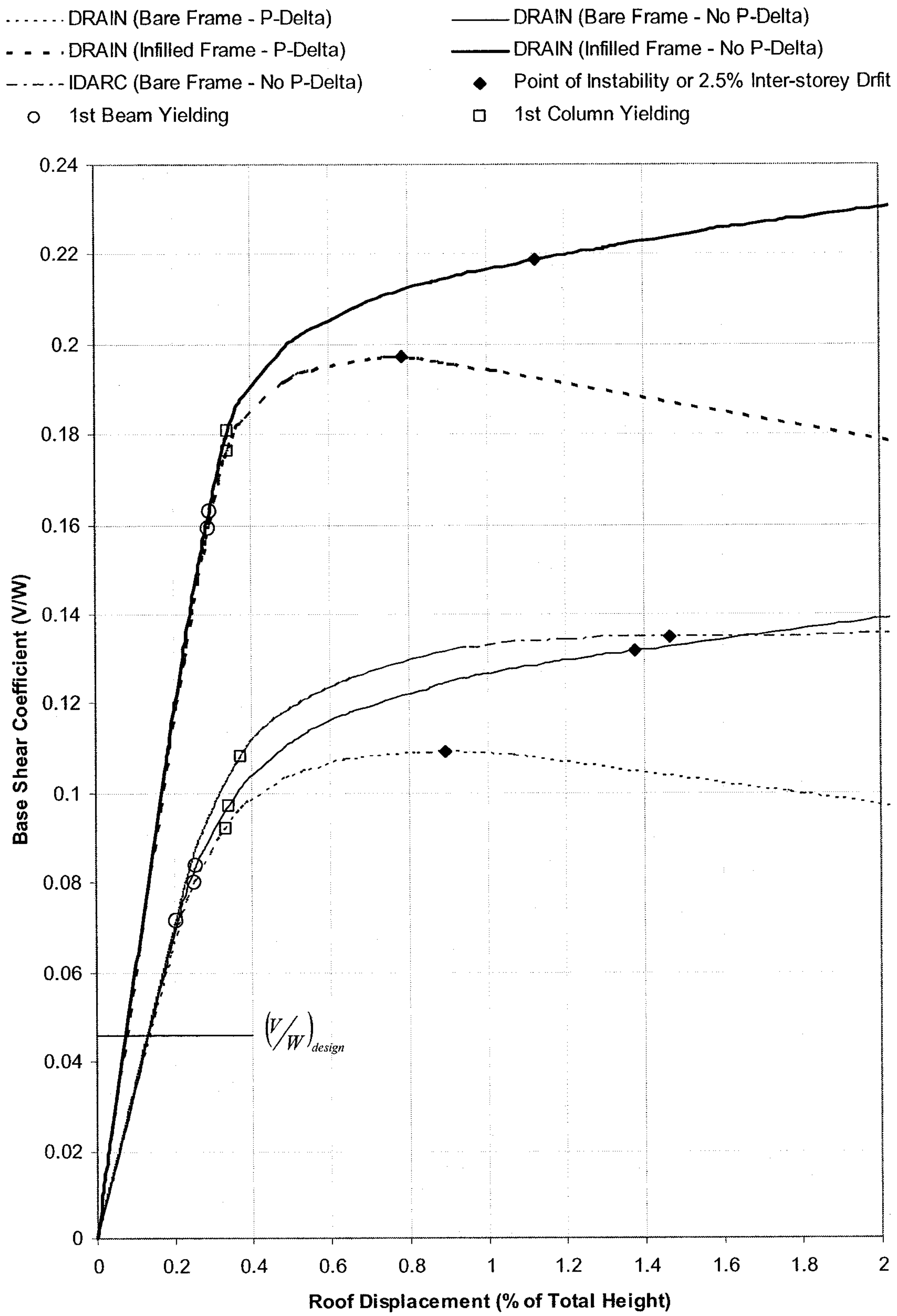


Figure 5.5: Push-Over Curves of the 6 Storey Frame

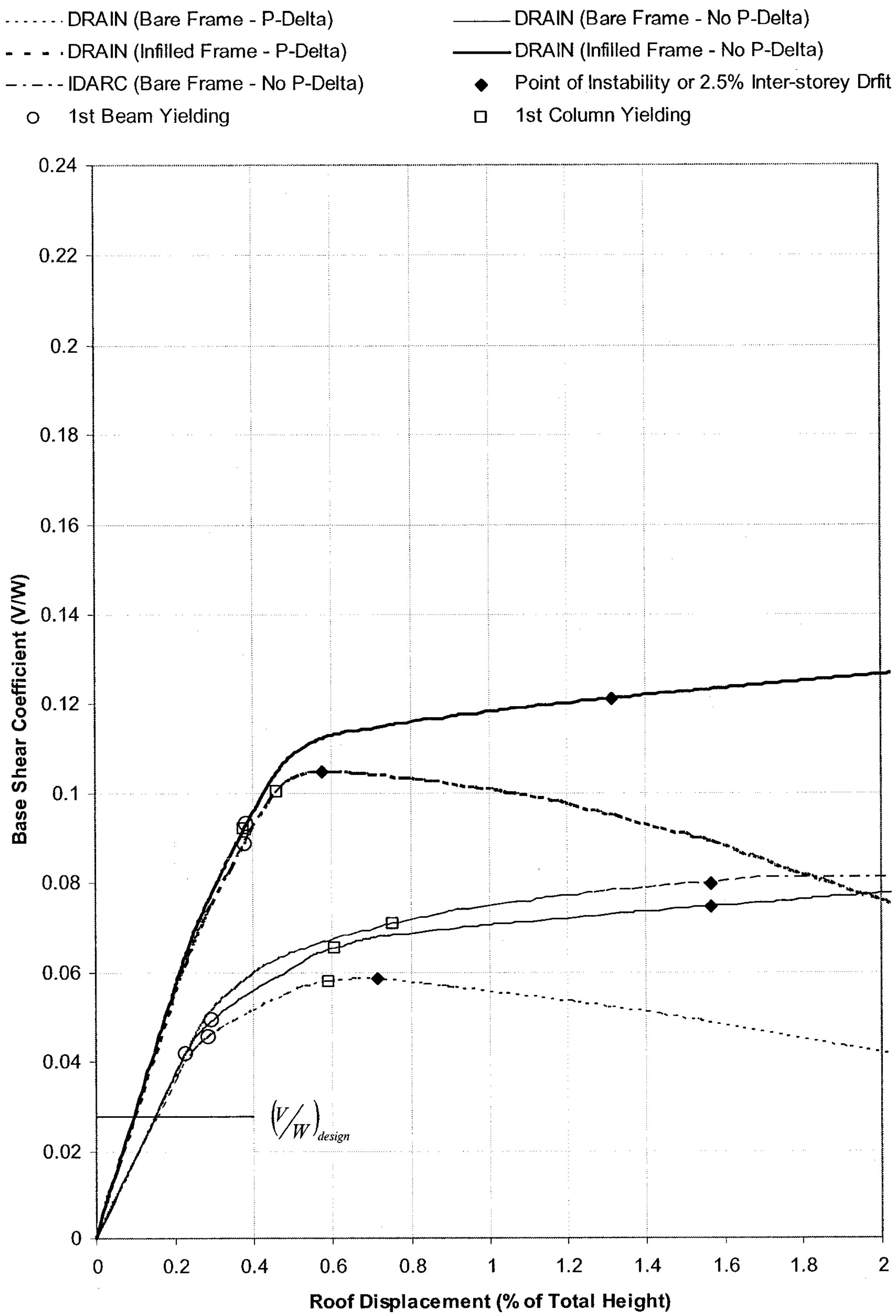


Figure 5.6: Push-Over Curves of the 12 Storey Frame

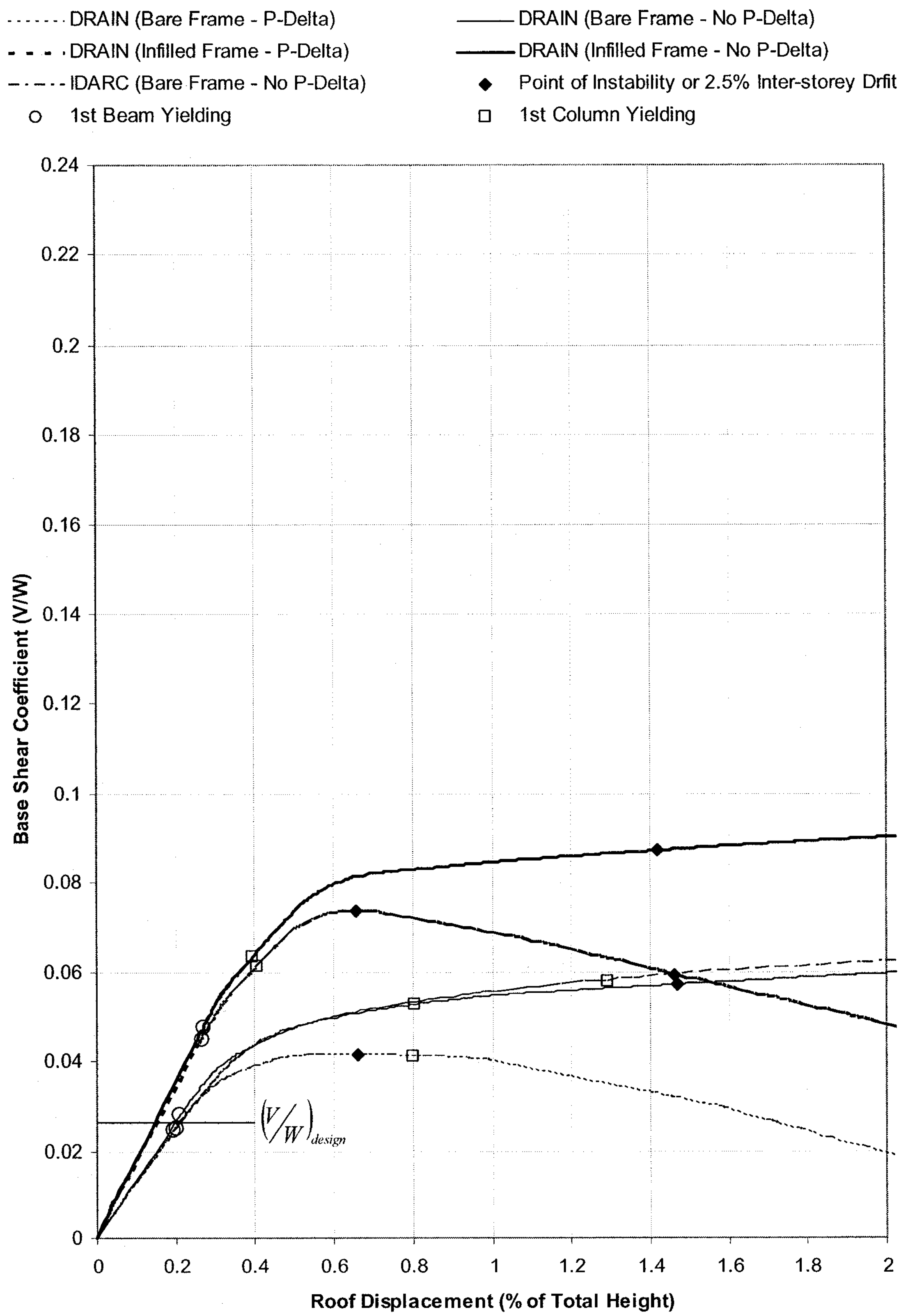


Figure 5.7: Push-Over Curves of the 18 Storey Frame

5.3.3 Nonlinear Dynamic Analysis

Nonlinear dynamic analysis is the best way to evaluate the seismic performance of buildings. However in order to obtain a good assessment, a large number of ground motion records (GMRs) should be used and rigorous analysis should be conducted. The more GMRs used in the dynamic analysis the more reliable the results become.

5.3.3.1 Synthesized GMRs

As previously mentioned, eight, four and four simulated stochastic ground acceleration records corresponding to a UHS-2500, UHS-1000 and UHS-500, respectively for Vancouver are used in the nonlinear dynamic analysis. Half of the simulated records used here are long duration records and the other half are short duration records. Long duration records have relatively moderate peak accelerations while short duration records have relatively high peak accelerations. The characteristics of these records are shown in the Tables 5.2, 5.3 and 5.4. The acceleration spectrum of the UHS-2500 compatible synthesized ground acceleration records used in the analysis is shown in Figure 5.8 and compared with the NBCC 2005 design spectrum. Figure 5.9 shows the time history of these records.

Table 5.2: Synthesized UHS-2500 Records

GMR	L1	L2	L3	L4	S1	S2	S3	S4
Total Duration (s)	18.18	18.18	18.18	18.18	8.53	8.53	8.53	8.53
Peak Acc. (cm/s^2)	244.2	221.1	248.6	242	523	416	567	339
Peak Acc. (g)	0.25	0.23	0.25	0.25	0.53	0.42	0.58	0.35

Table 5.3: Synthesized UHS-1000 Records

GMR	L1	L2	S1	S2
Total Duration (s)	17.98	17.98	6	6
Peak Acc. (cm/s^2)	163.24	146.7	230.4	275.8
Peak Acc. (g)	0.166	0.149	0.235	0.281

Table 5.4: Synthesized UHS-500 Records

GMR	L1	L2	S1	S2
Total Duration (s)	19.66	19.66	6	6
Peak Acc. (cm/s^2)	68.57	65.75	206.6	224
Peak Acc. (g)	0.07	0.067	0.211	0.228

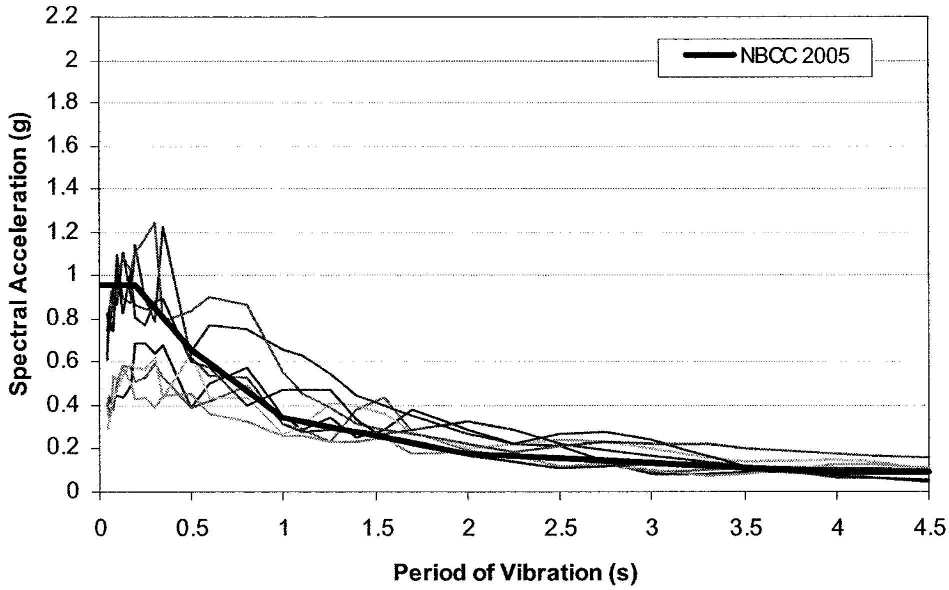


Figure 5.8: Response Spectrum of the Synthesized Records (UHS-2500)

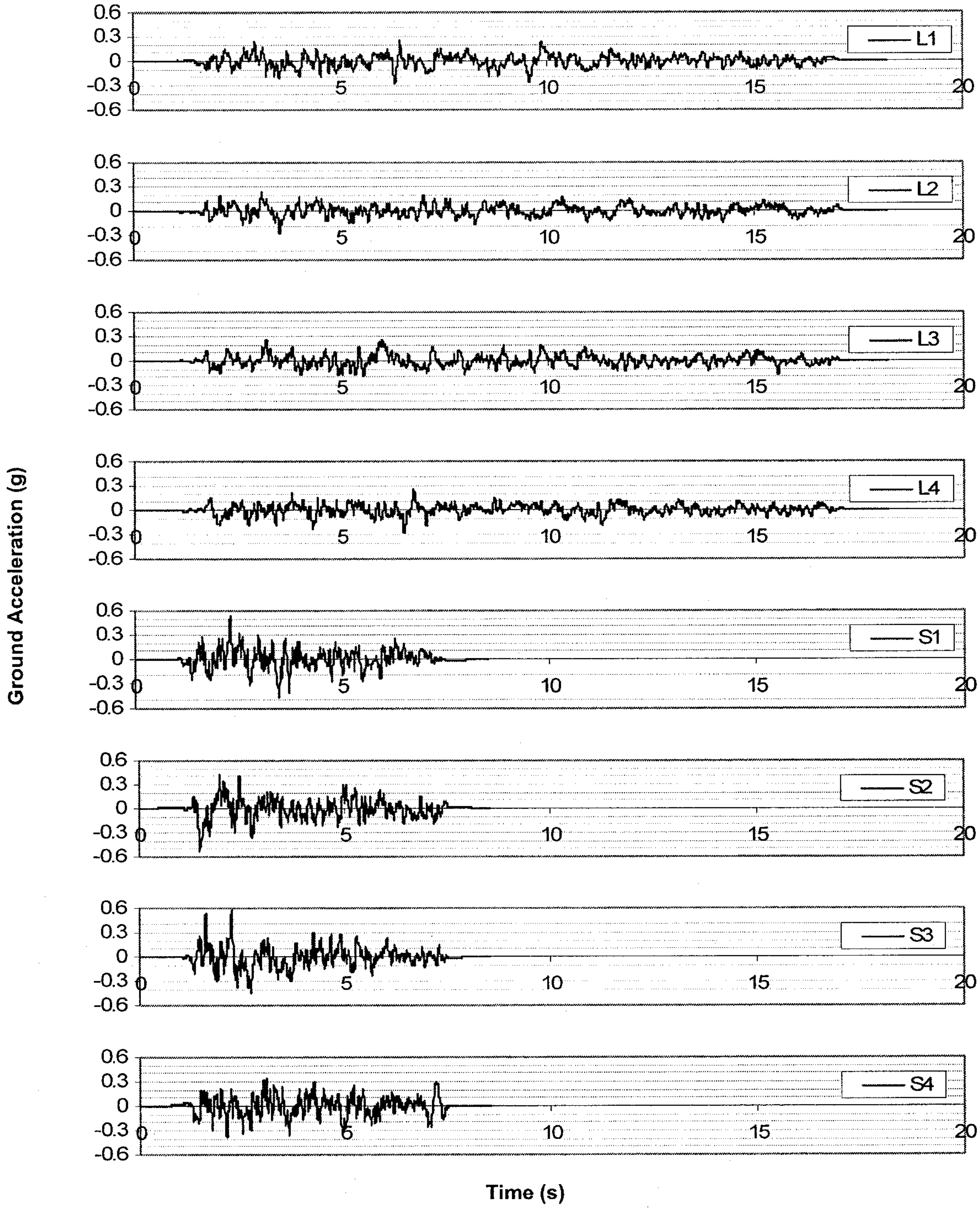


Figure 5.9: Time-History of the Synthesized Records (UHS-2500)

5.3.3.2 Scaled Actual GMRs

A collection of sixteen actual records are used in the nonlinear dynamic analysis. These records are scaled based on the design spectrum for Vancouver corresponding to a UHS-2500. Two methods of record scaling are used here, the ordinate method and the partial area method. First, spectral analysis is performed for each record. The response spectrum is then scaled to match the design spectrum of Vancouver. The characteristics of the actual records used in the analysis before scaling are shown in Table 5.5. The scaling methods used here are explained in the following two paragraphs based on Figure 5.10.

The ordinate method of record scaling is performed based on the fundamental period of vibration of the structure. The response spectral acceleration corresponding to the fundamental period is scaled up or down to the value of the design spectral acceleration corresponding to the same period. In other words, all record values are scaled based on the factor Sa_1 / Sa_2.

The partial area method of record scaling, on the other hand, is based on the first and second period of vibration of the structure. The area under the response spectral acceleration curve between 1.2 times the fundamental period value and the second period value is scaled to equal the area under the design spectral acceleration curve between the same period values. All the values of this record are scaled based on the factor A_1 / A_2.

Both scaling methods are compared by comparing the scaling factors as well as the response of the structures to the ground motions scaled using both methods. The scaling factors obtained from both methods are shown in table 5.6 and the response spectrum of the real GMRs scaled for the six-storey building is shown in Figure 5.11 and compared with the NBCC 2005 design spectrum.

Table 5.5: Actual GMRs

Record	Location / Record No.	Date	Amax (g)	Vmax (m/s)	A/V (s/m)	Duration (s)
1	Imperial Valley	18/05/1940	0.35	0.33	1.04	53.74
2	Kern County	21/07/1952	0.18	0.18	1.01	54.40
3	Kern County	21/07/1952	0.16	0.16	0.99	19.16
4	Borrego Mountain	08/04/1968	0.05	0.04	1.10	45.00
5	Friuli, Italy	15/09/1976	0.11	10.2	0.01	26.39
6	San Fernando	09/02/1971	0.15	0.15	1.01	65.18
7	San Fernando	09/02/1971	0.21	0.21	1.00	79.48
8	San Fernando	09/02/1971	0.17	0.17	0.99	62.58
9	San Fernando	09/02/1971	0.18	0.20	0.88	43.00
10	San Fernando	09/02/1971	0.20	0.17	1.19	47.08
11	Gazli, USSR	17/05/1976	0.608	65.4	0.01	16.27
12	Coalinga	22/07/1983	0.217	18.1	0.01	19.50
13	Monte Negro	15/04/1979	0.17	0.19	0.88	40.40
14	SUCH850919AL.T	19/09/1985	0.11	0.11	0.94	120.00
15	VILE850919AT.T	19/09/1985	0.09	0.11	0.83	128.00
16	Coyote Lake	06/08/1979	0.271	26.3	0.01	27.19

Table 5.6: Scaling Factors (Bare Frames)

Ground Motion Record		1	2	3	4	5	6	7	8
6 Storey	Ord. Meth.	1.24	2.43	2.04	6.63	3.44	1.89	1.01	1.72
	P.A. Meth.	0.84	1.87	1.81	7.49	2.72	2.60	1.44	2.28
12 Storey	Ord. Meth.	0.85	2.62	4.16	9.55	5.82	3.53	1.87	1.76
	P.A. Meth.	0.95	2.02	2.33	8.23	3.45	2.83	1.46	2.12
18 Storey	Ord. Meth.	1.34	3.07	4.79	9.20	9.64	3.52	1.12	2.30
	P.A. Meth.	1.14	2.30	2.90	9.26	4.67	3.10	1.41	2.16

Ground Motion Record		9	10	11	12	13	14	15	16
6 Storey	Ord. Meth.	3.10	2.13	0.79	2.54	2.01	0.82	1.52	1.48
	P.A. Meth.	1.81	2.14	0.62	2.31	1.63	1.49	1.57	1.28
12 Storey	Ord. Meth.	2.58	1.93	0.85	3.51	1.67	1.69	2.42	4.03
	P.A. Meth.	2.18	2.22	0.65	2.73	1.88	1.21	1.60	1.75
18 Storey	Ord. Meth.	1.81	1.26	0.75	4.86	2.14	1.64	2.14	4.71
	P.A. Meth.	2.45	1.90	0.68	3.57	1.85	1.28	1.80	2.55

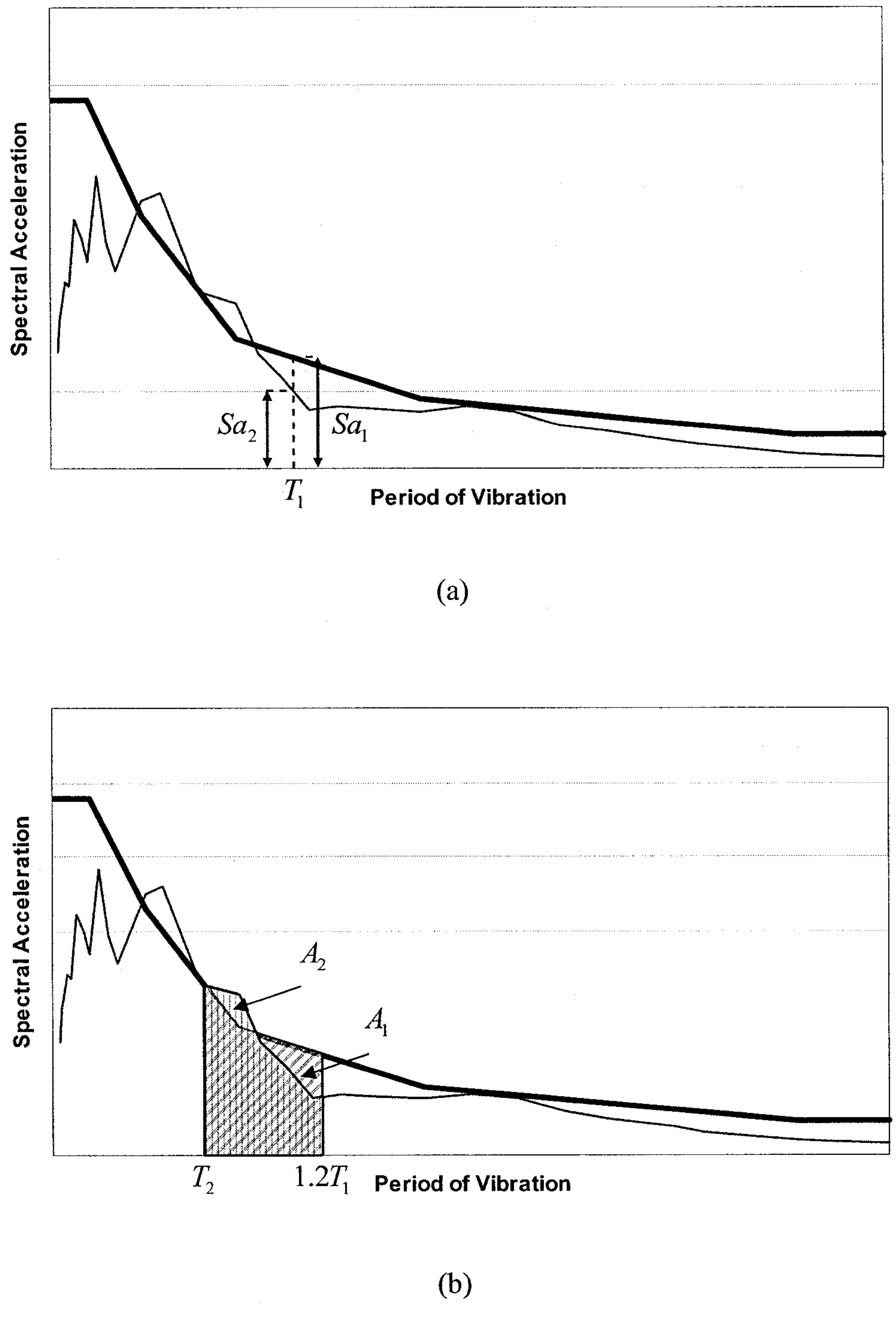
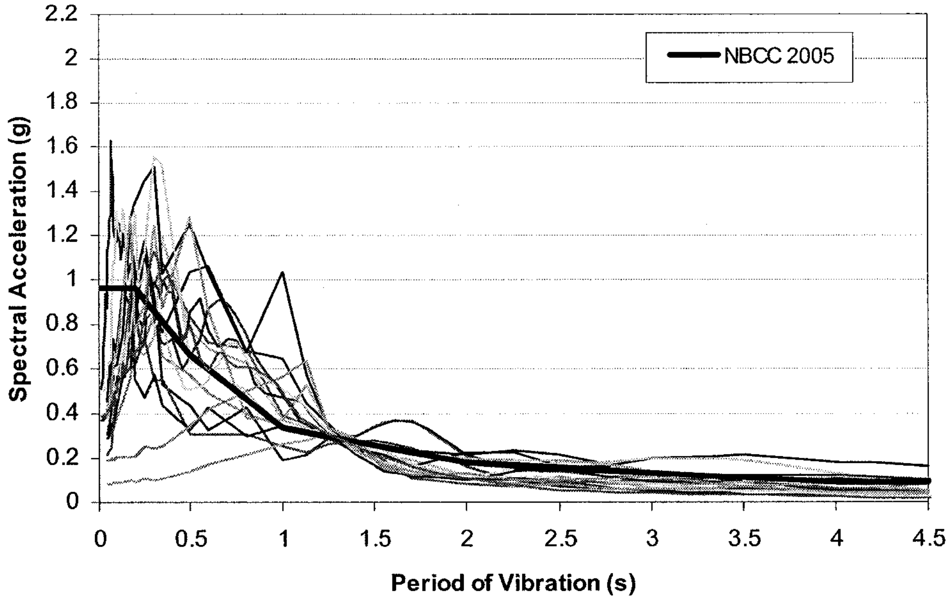
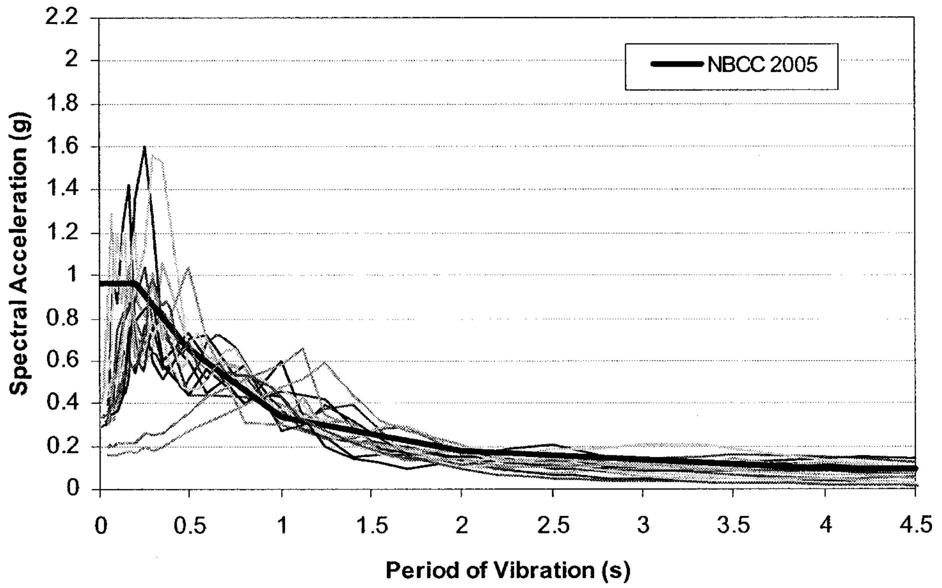


Figure 5.10: Record Scaling Methods; (a) Ordinate Method, (b) Partial Area Method



(a)



(b)

Figure 5.11: Response Spectrum of the Actual GMRs for the 6 Storey Frame; (a) Scaled Using the Ordinate Method, (b) Scaled Using the Partial Area Method

5.3.3.3 Inter-Storey Drifts of the Bare Frames due to the Synthesized GMRs

The maximum inter-storey drifts obtained from the analysis of the bare frame models for the synthesized records carried out by IDARC2D and DRAIN-2DX are shown in Figures 5.12 through 5.17. Since the number of synthesized records used in this analysis is small, the calculation of mean and standard deviation values is not so meaningful. For this reason, the maximum (envelope) values will be used in the performance evaluation instead.

Figure 5.12 shows the inter-storey drifts of the six-storey bare frame obtained from the IDARC2D analysis. The maximum inter-storey drift due to the UHS-500 long duration records is 0.33% of the storey height occurring at the first storey level. The maximum inter-storey drift due to the short duration records occurs at the second storey level and is equal to 0.54%. Due to the UHS-1000 records, the maximum inter-storey drifts are 0.86% and 0.71% caused by the long duration and short duration records, respectively, both values occurring at the first storey level. As for the UHS-2500, the maximum inter-storey drifts also occur at the first storey level and are equal to 2.07% and 1.46% due to the long duration and short duration records, respectively.

The maximum inter-storey drifts of the six-storey bare frame produced by DRAIN-2DX are shown in Figure 5.13. The maximum inter-storey drift due to the UHS-500 long duration records is found to be 0.35% occurring at the first storey level while that obtained from the short duration records is 0.55% occurring at the third storey level. The UHS-1000 records produce 0.76% and 1.07% maximum inter-storey drifts at the first storey level due to the long duration and short duration records, respectively. Finally, the UHS-2500 records cause 2% and 2.26% maximum inter-storey drifts due to the long duration and short duration records, respectively, both values occurring at the first storey level.

Figure 5.14 shows the maximum inter-storey drifts of the twelve-storey bare frame produced by IDARC2D. The maximum inter-storey drifts caused by the UHS-500 records are 0.47% at the second storey level due to the long duration records and 0.76% due to the short duration records also at the second storey level. As for the UHS-1000 records, the long duration records cause a 0.49% maximum inter-storey drift at the second storey level while the short duration records cause a maximum inter-storey drift of 0.54% at the fifth storey level. Finally, due to the UHS-2500 records the maximum inter-storey drift due to the long duration records is found to be 1.7% occurring at the fourth storey level and the maximum inter-storey drift due the short duration records is 2.25% occurring at the second storey level.

Figure 5.15 shows the results of the maximum inter-storey drifts of the twelve-storey bare frame obtained from the DRAIN-2DX analysis. The UHS-500 long duration and short duration records cause maximum inter-storey drifts of 0.44% and 0.6%, respectively, at the second storey level. The maximum inter-storey drift caused by the UHS-1000 long duration records is 0.5% occurring at the ninth storey level while the maximum inter-storey drift caused by the short records is 0.63% occurring at the fifth storey level. As for the UHS-2500, the maximum inter-storey drift due to the long duration records is 1.69% at the fourth storey level and that due to the short duration records is 1.76% at the third storey level.

The maximum inter-storey drifts of the eighteen-storey bare frame obtained from the IDARC2D analysis are shown in Figure 5.16. Due to the UHS-500 records, the maximum inter-storey drift caused by the long duration records is 0.39% occurring at the sixth and seventh storey levels and that caused by the short duration records is 0.46% also occurring at both the sixth and the seventh storey levels. The UHS-1000 long duration records produce a maximum inter-storey drift of 0.91% at the fifth storey level and the short duration records produce a maximum inter-storey drift of 0.48% at the eleventh storey level. The maximum inter-storey drifts caused by the UHS-2500 long duration and short duration records are 1.18% and 2.12%, respectively, both values occurring at the fourth storey level.

Figure 5.17 shows the maximum inter-storey drifts of the eighteen-storey bare frame produced by DRAIN-2DX. The maximum inter-storey drifts due to the UHS-500 records are found to be 0.36% at the seventh storey level and 0.44% at the fifth storey level due to the long duration and short duration records, respectively. Due to the UHS-1000 records, the maximum inter-storey drifts are 0.71% at the fifth storey level and 0.51% at the eleventh storey level due to the long duration and short duration records, respectively. Finally, the UHS-2500 long duration records cause a maximum inter-storey drift of 1.05% at the sixth storey level while the short duration records cause a maximum inter-storey drift of 1.62% at the fourth storey level.

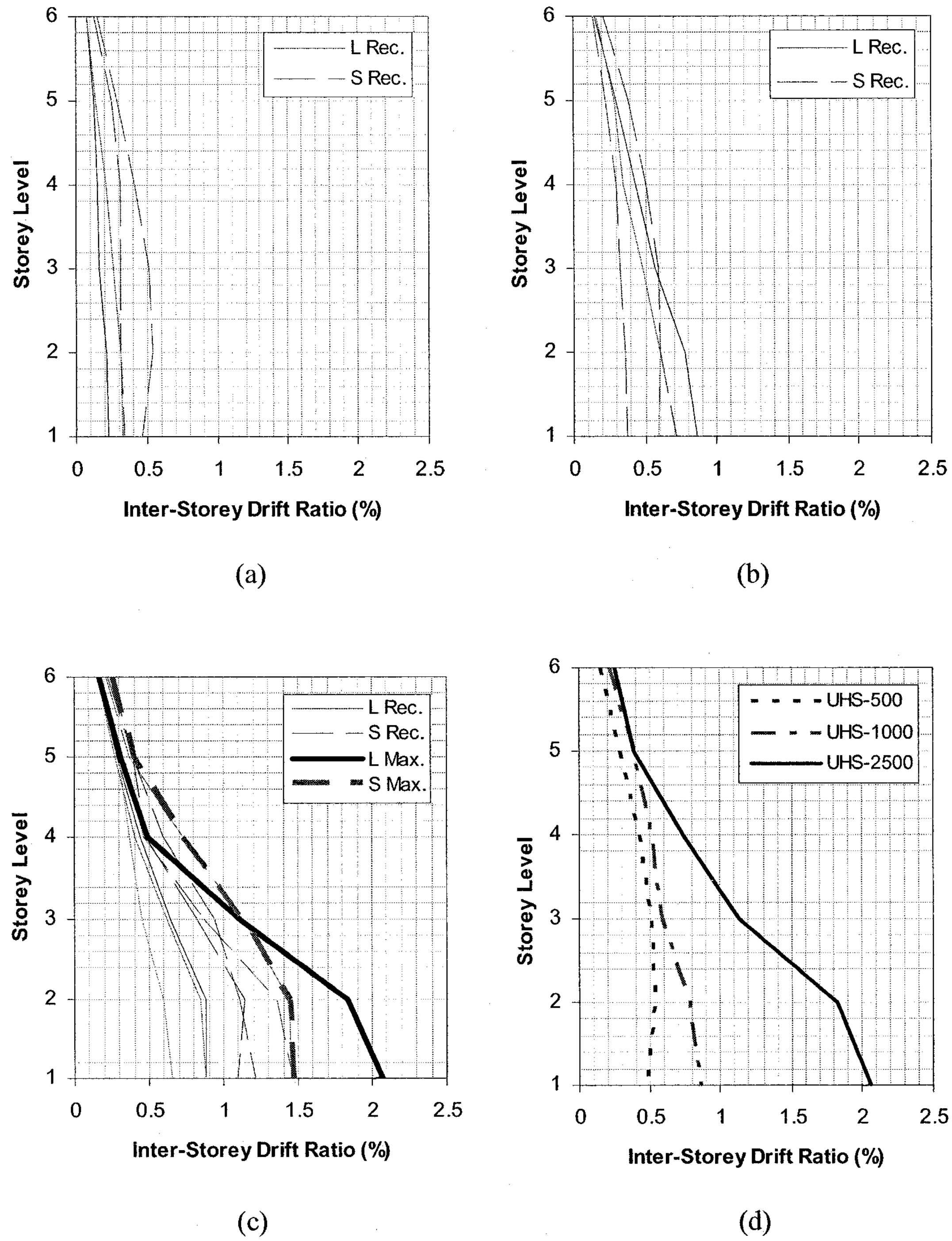


Figure 5.12: Inter-Storey Drifts of the 6 Storey Bare Frame due to the Synthesized GMRs (IDARC2D); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

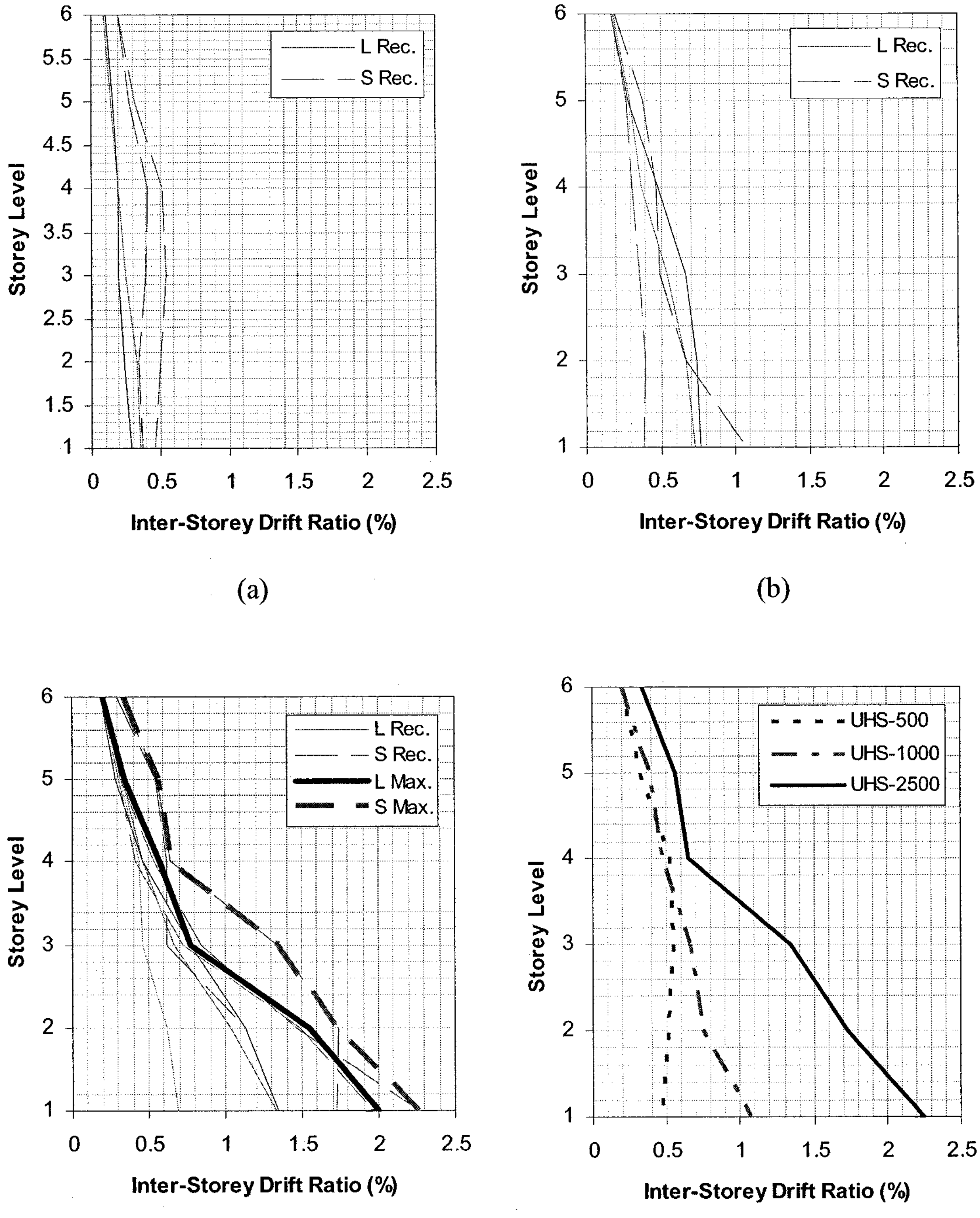


Figure 5.13: Inter-Storey Drifts of the 6 Storey Bare Frame due to the Synthesized GMRs (DRAIN-2DX); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

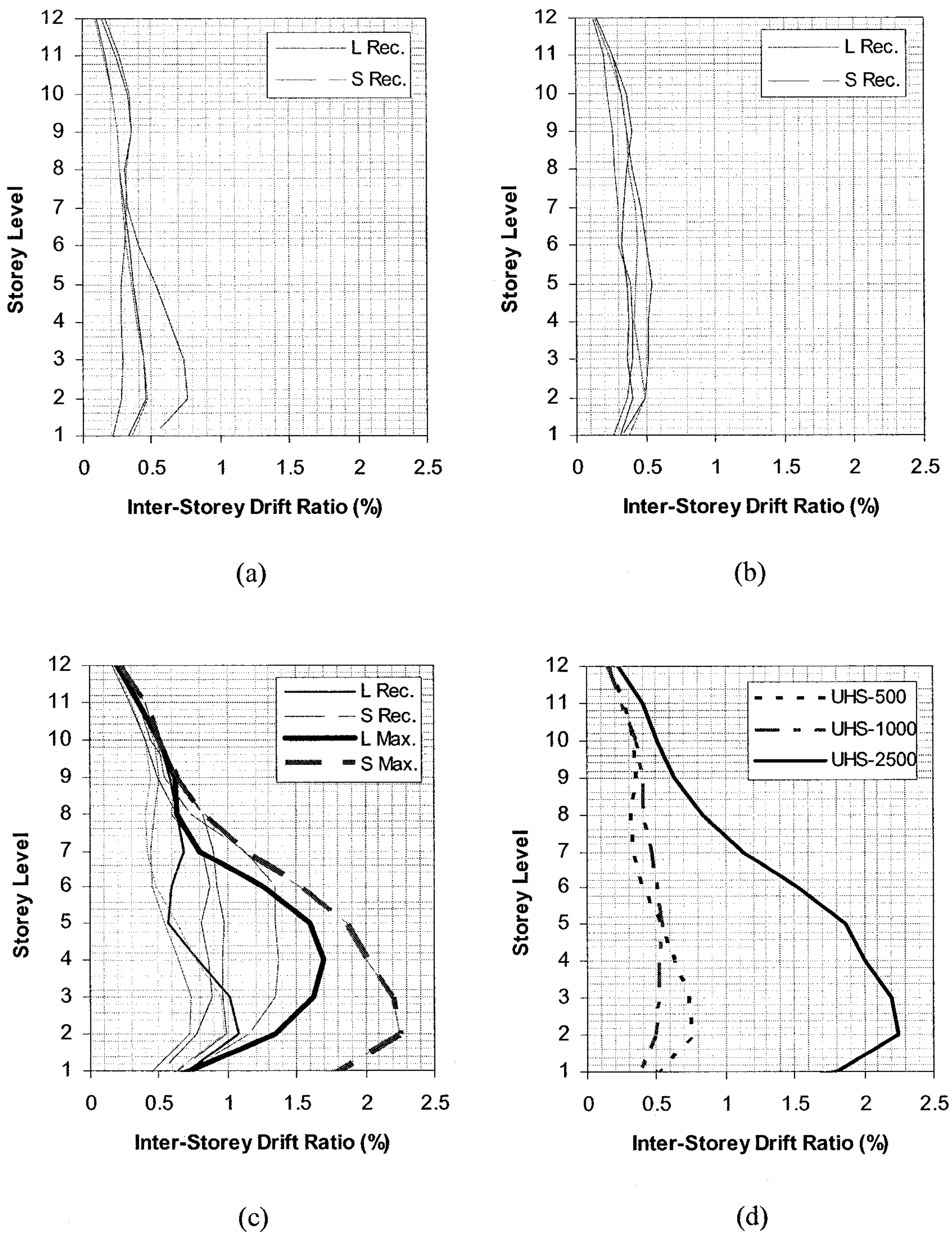


Figure 5.14: Inter-Storey Drifts of the 12 Storey Bare Frame due to the Synthesized GMRs (IDARC2D); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

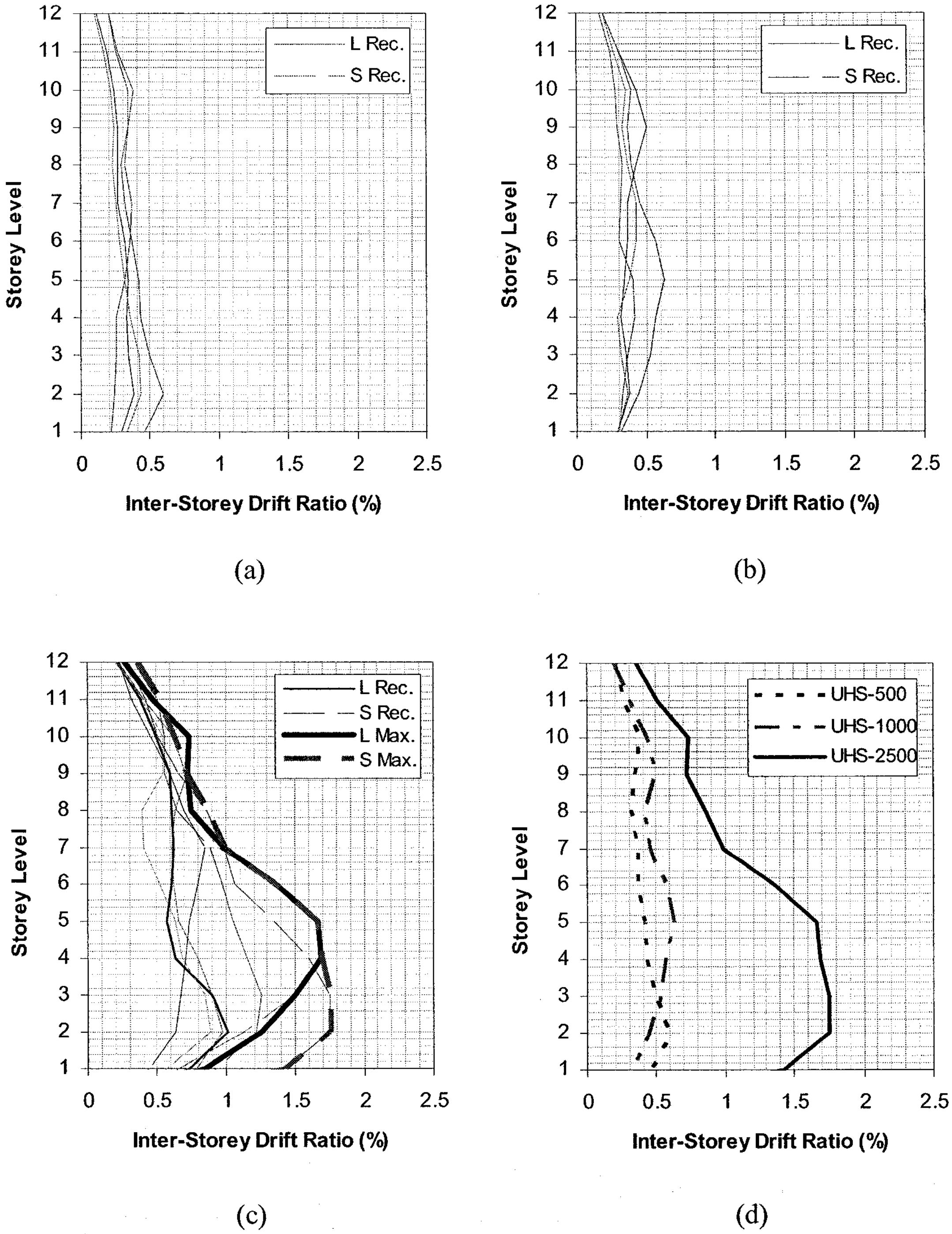


Figure 5.15: Inter-Storey Drifts of the 12 Storey Bare Frame due to the Synthesized GMRs (DRAIN-2DX); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

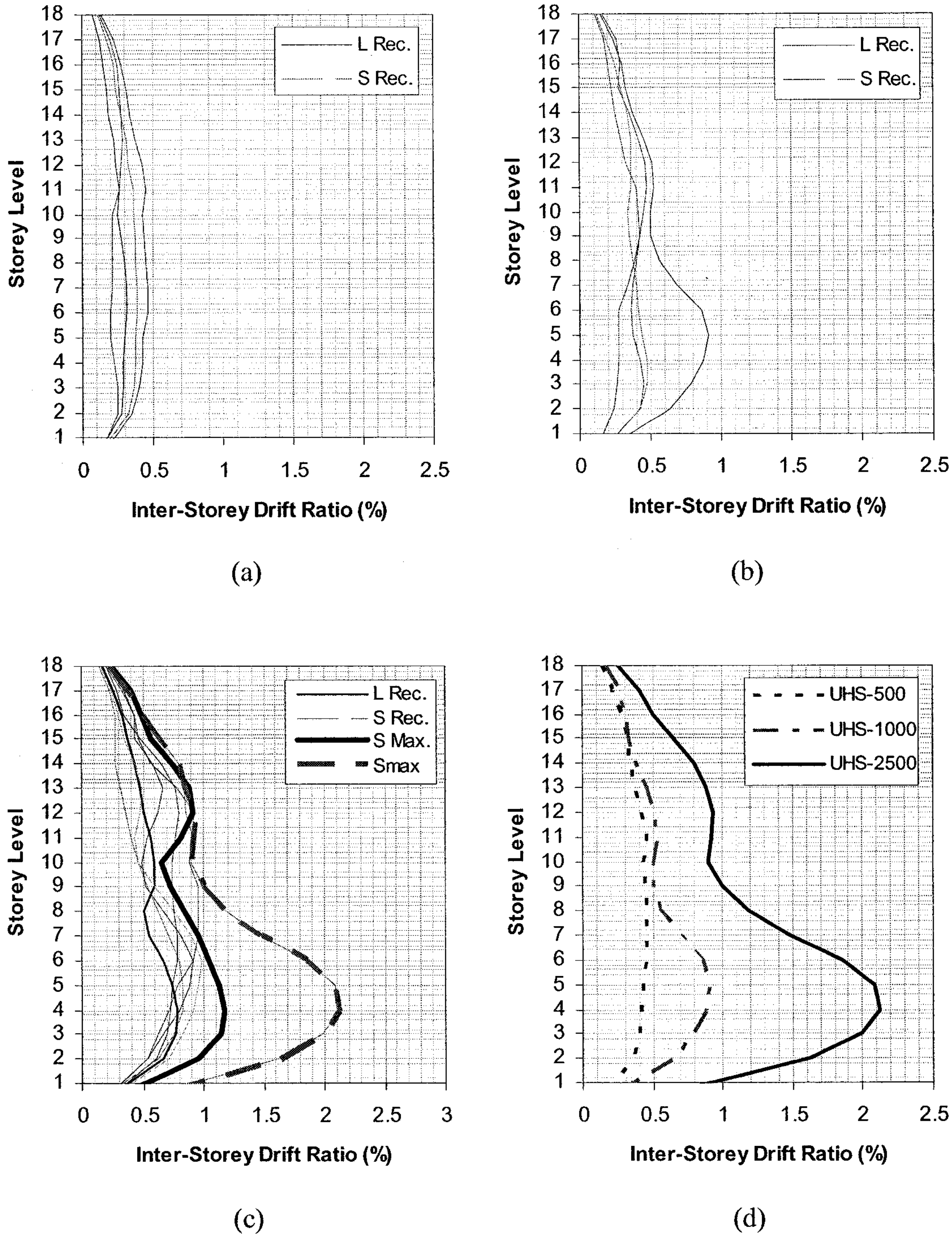


Figure 5.16: Inter-Storey Drifts of the 18 Storey Bare Frame due to the Synthesized GMRs (IDARC2D); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

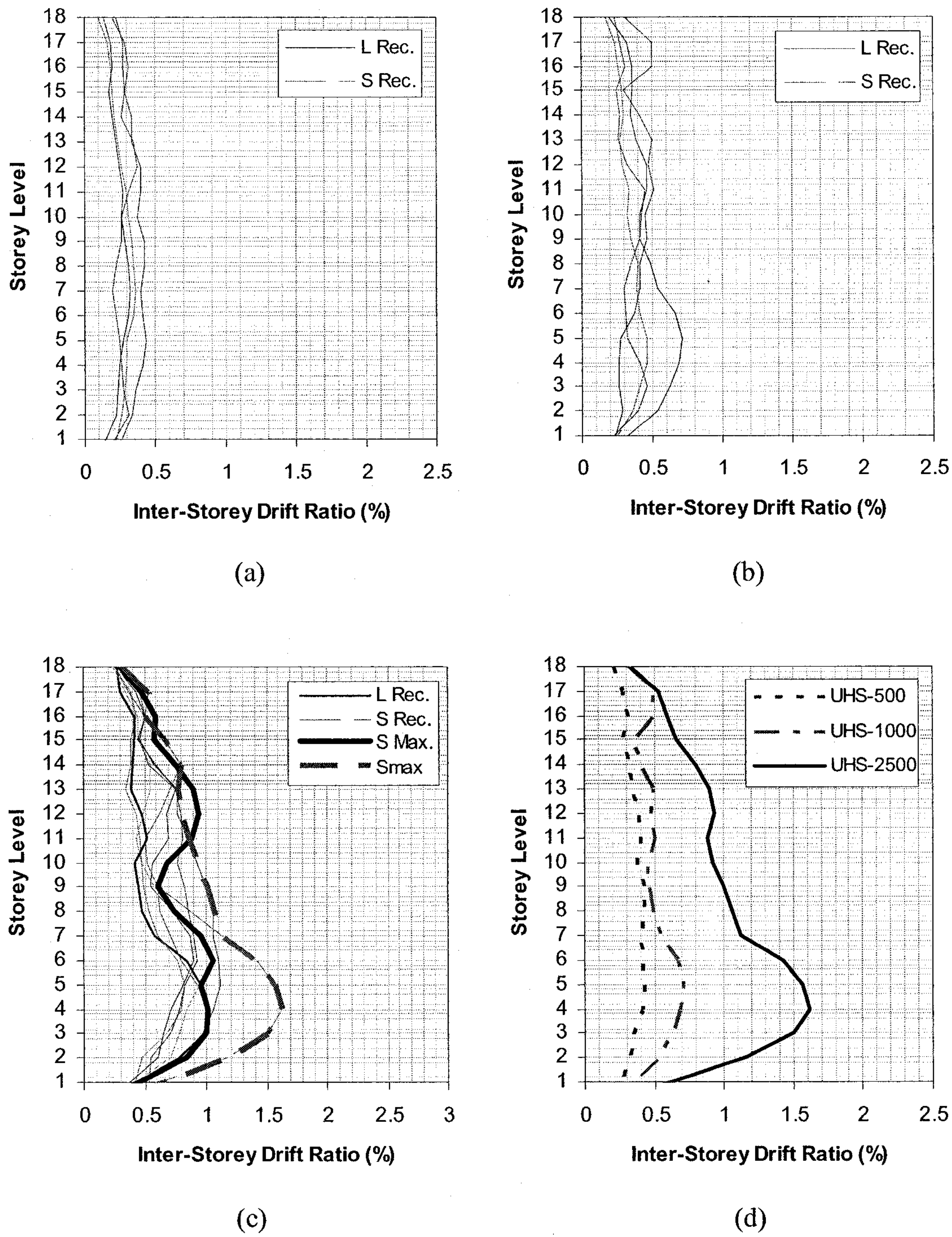


Figure 5.17: Inter-Storey Drifts of the 18 Storey Bare Frame due to the Synthesized GMRs (DRAIN-2DX); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

5.3.3.4 Inter-Storey Drifts of the Infilled Frames due to the Synthesized GMRs

The maximum inter-storey drifts obtained from the analysis of the infilled frame models for the synthesized records carried out by DRAIN-2DX are shown in Figures 5.18 through 5.20.

Figure 5.18 shows the inter-storey drifts of the six-storey infilled frame produced by DRAIN-2DX. Due to the UHS-500 records, the maximum inter-storey drift is 0.15% and 0.33% caused by the long duration and short duration records, respectively, both values occurring at the first storey level. The maximum inter-storey drifts due to the UHS-1000 long duration and short duration records are found to be 0.6% and 0.39%, respectively, both values occurring at the first storey level. The maximum inter-storey drifts caused by the UHS-2500 records also occur at the first storey level and are equal to 0.77% and 1.37% due to the long duration and short duration records, respectively.

Figure 5.19 shows the inter-storey drifts of the twelve-storey infilled frame produced by the DRAIN-2DX analysis. The maximum inter-storey drifts due to the UHS-500 records are found to be 0.25% at the fifth storey level and 0.33% at the fourth storey level due to the long duration and short duration records, respectively. As for the UHS-1000 records, the maximum inter-storey drifts are 0.42% at the seventh storey level and 0.52% at the third storey level due to the long duration and short duration records, respectively. The UHS-2500 long duration records cause a maximum inter-storey drift of 0.76% at the third storey level while the short duration records cause a maximum inter-storey drift of 0.98% at the fourth storey level.

The maximum inter-storey drifts of the eighteen-storey infilled frame obtained from the DRAIN-2DX analysis are shown in Figure 5.20. Due to the UHS-500, the maximum inter-storey drift due to the long duration records is found to be 0.28% occurring at the sixth storey level while that obtained from the short duration records is 0.42% occurring at the second storey level. The maximum inter-storey drifts produced by the UHS-1000 records are found to be 0.4% at the thirteenth storey level and 0.34% at the fourteenth storey level due to the long duration and short duration records, respectively. Finally, the UHS-2500 records cause 1.01% at the eighth storey level and 1.36% at the second storey level maximum inter-storey drifts due to the long duration and short duration records, respectively.

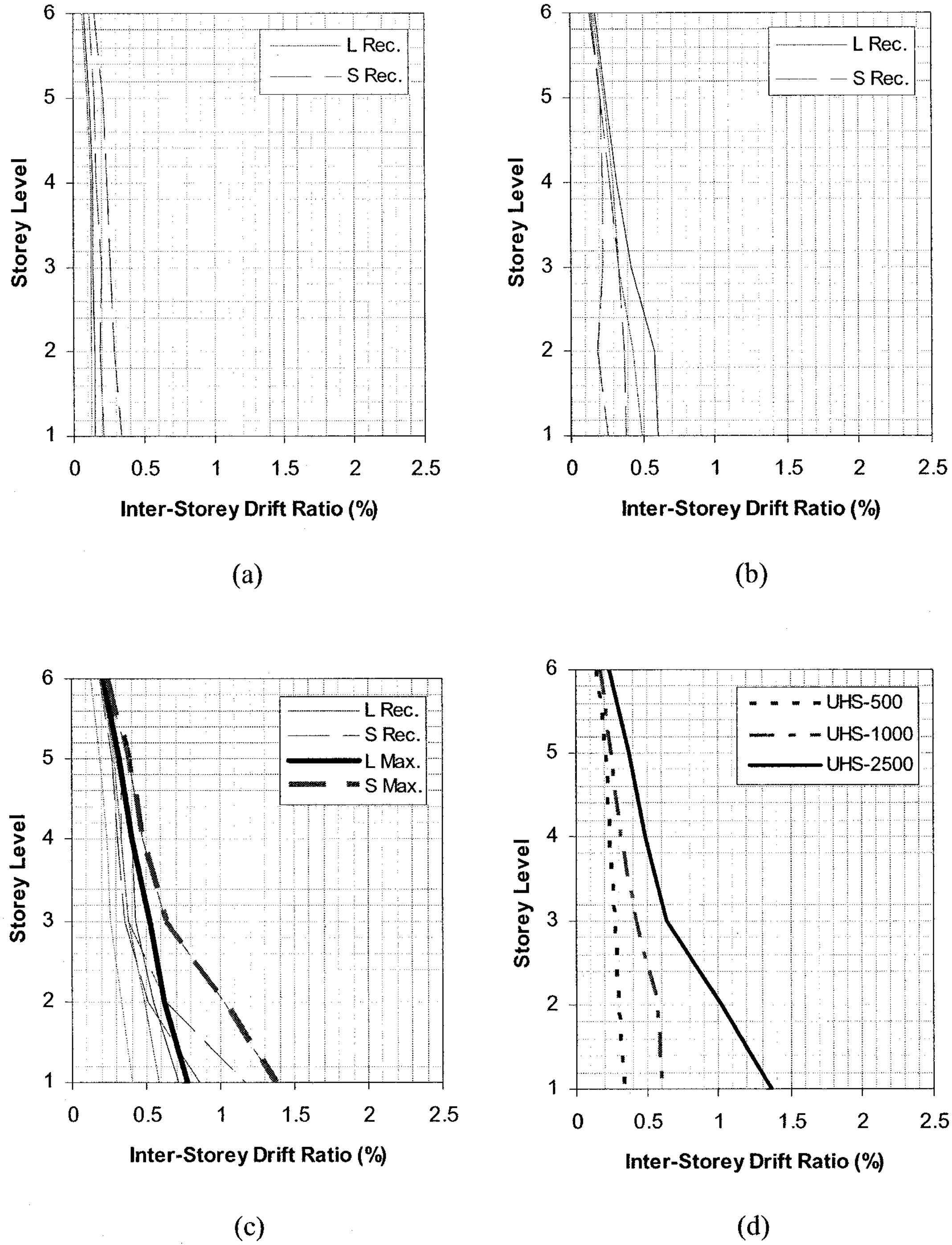


Figure 5.18: Inter-Storey Drifts of the 6 Storey Infilled Frame due to the Synthesized GMRs (DRAIN-2DX); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

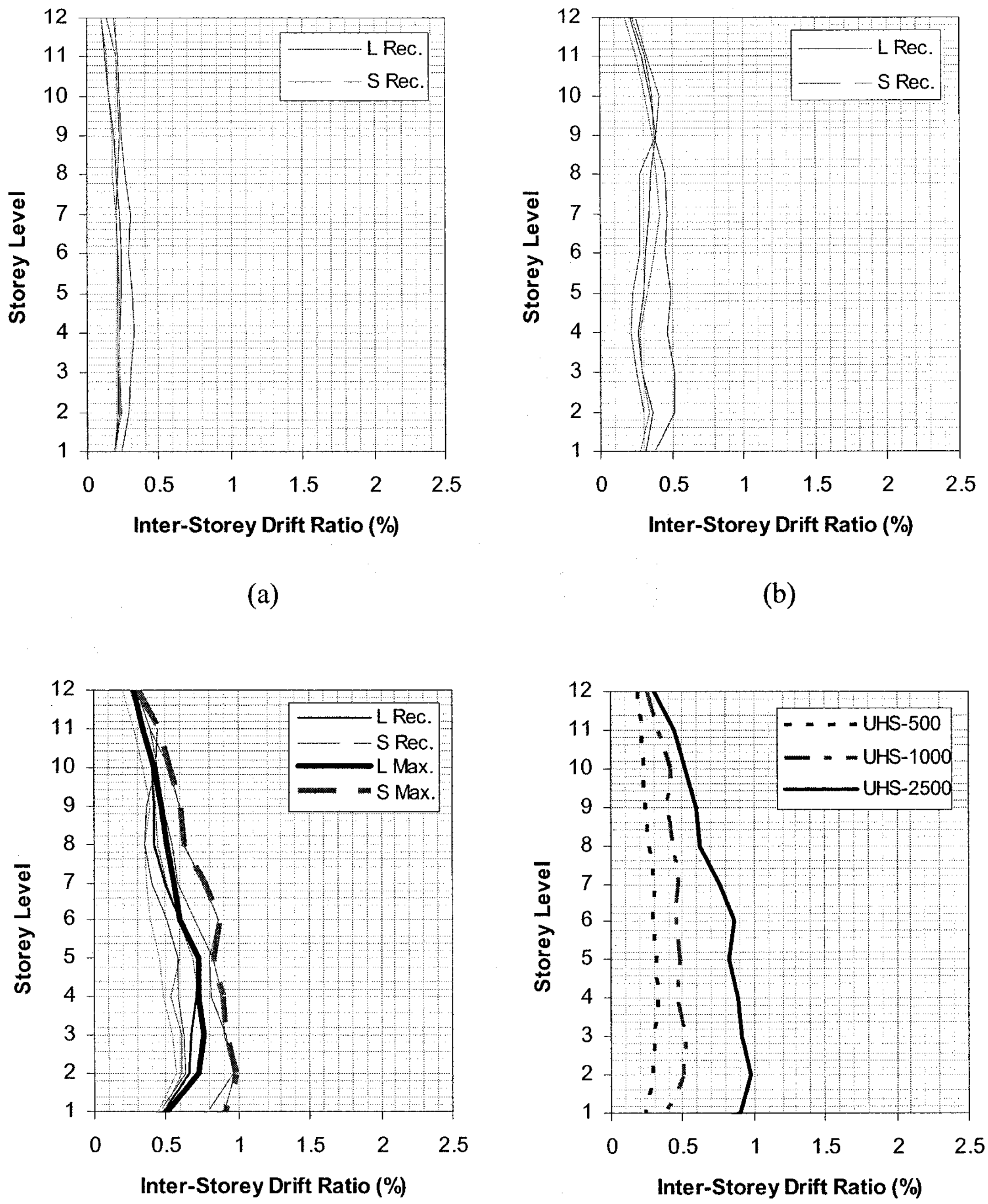
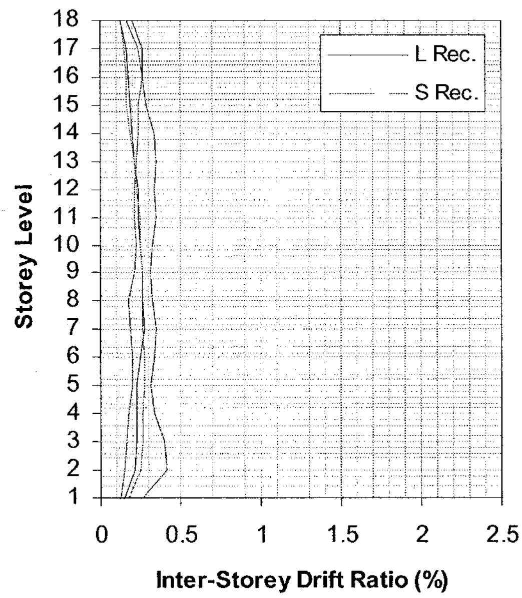
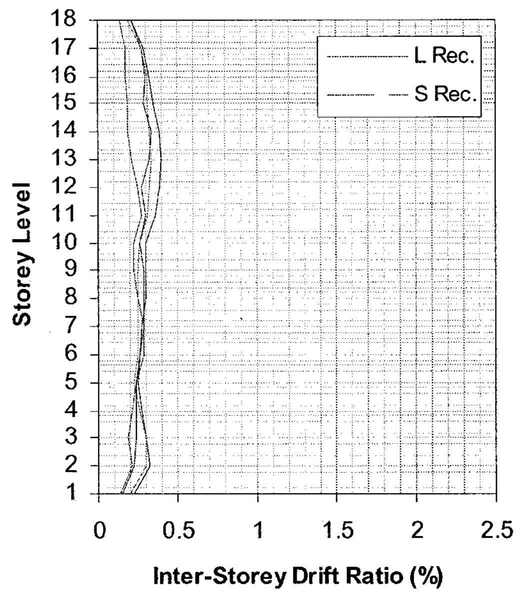


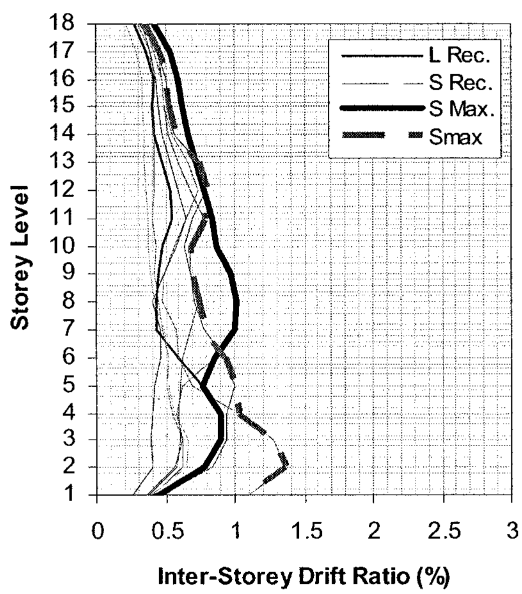
Figure 5.19: Inter-Storey Drifts of the 12 Storey Infilled Frame due to the Synthesized GMRs (DRAIN-2DX); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)



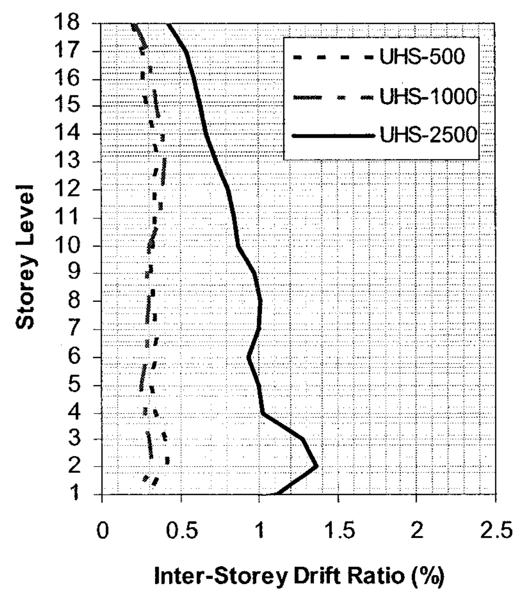
(a)



(b)



(c)



(d)

Figure 5.20: Inter-Storey Drifts of the 18 Storey Infilled Frame due to the Synthesized GMRs (DRAIN-2DX); (a) UHS-500, (b) UHS-1000, (c) UHS-2500, (d) Comparison of the Envelopes of Curves (a), (b) and (c)

5.3.3.5 Damage Indices of the Bare Frames due to the Synthesized GMRs

The storey damage indices obtained from the analysis of the bare frame models for the UHS-2500 compatible synthesized records carried out by IDARC2D are shown in Figure 5.21. The maximum storey damage indices due to the long duration records are 0.23 occurring at the first storey level, 0.34 occurring at the fourth storey level and 0.22 occurring at the third storey level for the six, twelve and eighteen-storey bare frames, respectively. The maximum storey damage indices in the case of short duration records are 0.21 occurring at the second storey level, 0.44 also occurring at the second storey level and 0.39 occurring at the fourth storey level for the six, twelve and eighteen-storey bare frames, respectively.

The overall structural damage indices of the three bare frame models are shown in Table 5.7.

Table 5.7: Overall Structural Damage Indices of the Bare Frames due to the Synthesized GMRs (IDARC2D)

GMR	L1	L2	L3	L4	S1	S2	S3	S4
6 Storey	0.06	0.05	0.06	0.05	0.05	0.11	0.12	0.09
12 Storey	0.16	0.13	0.27	0.14	0.24	0.17	0.33	0.16
18 Storey	0.13	0.12	0.18	0.14	0.17	0.13	0.30	0.13

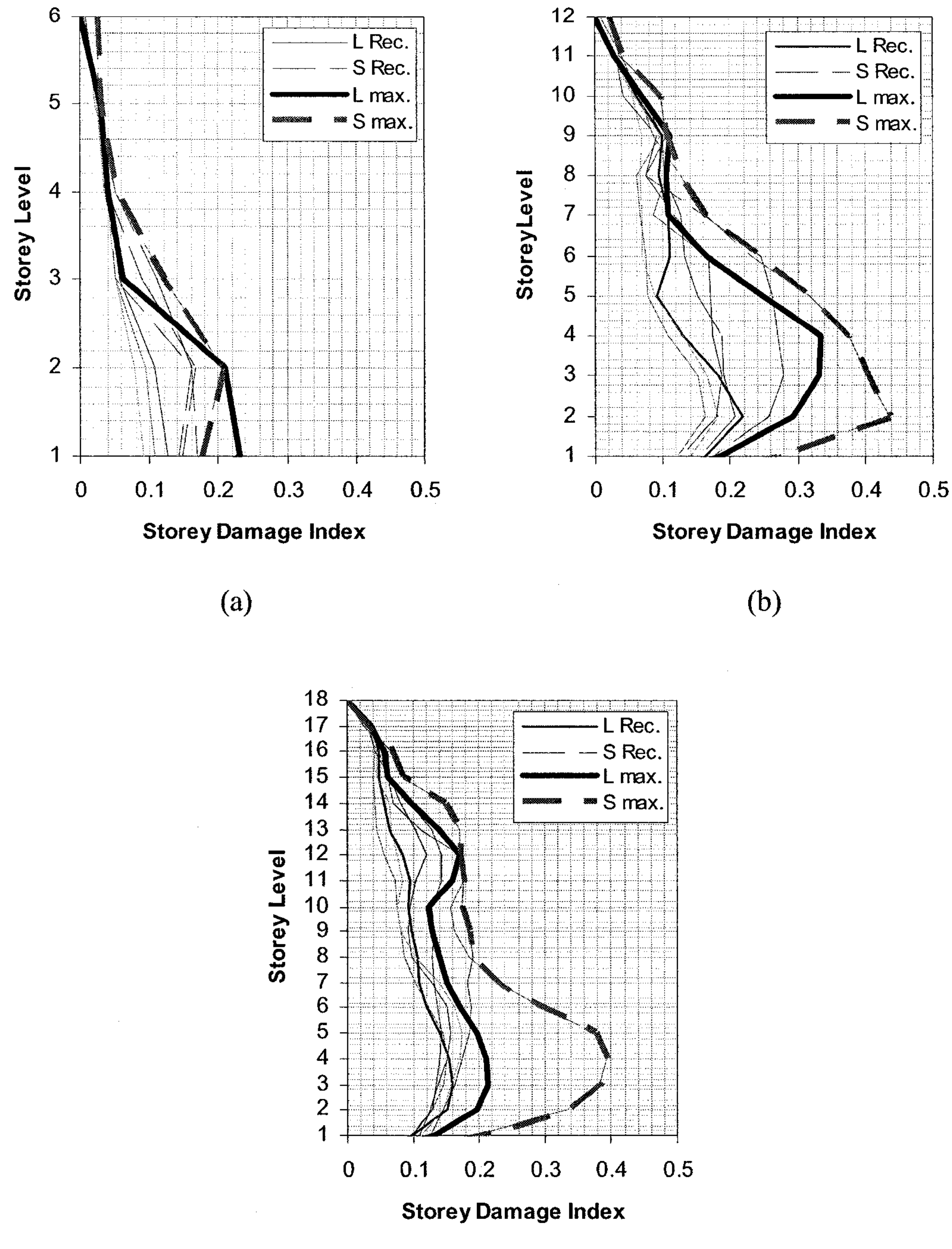


Figure 5.21: Storey Damage Indices of the Bare Frames due to the UHS-2500 Synthesized GMRs (IDARC2D); (a) 6 Storey Frame, (b) 12 Storey Frame, (c) 18 Storey Frame

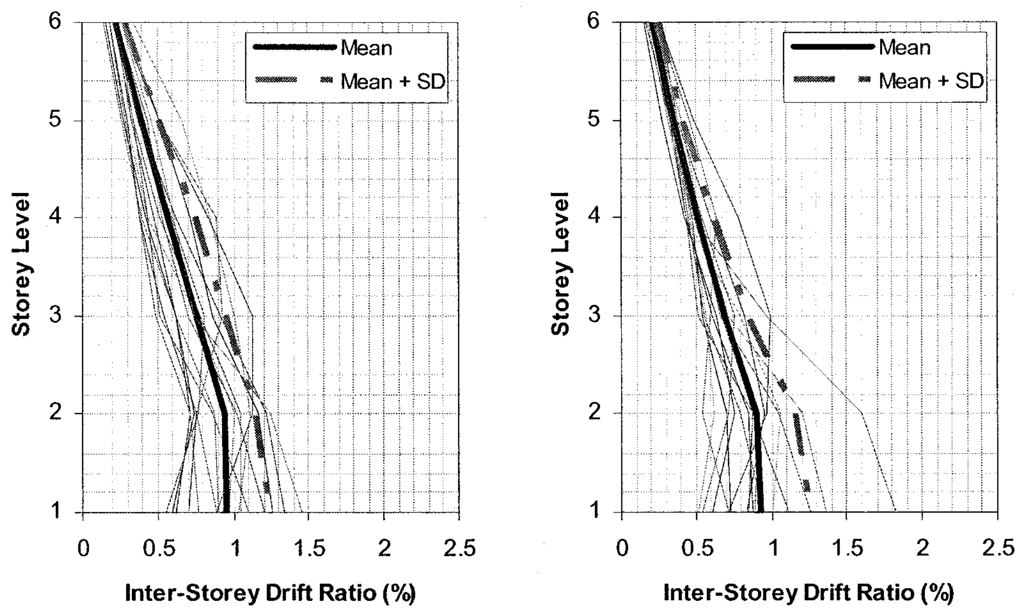
5.3.3.6 Inter-Storey Drifts of the Bare Frames due to the Scaled Actual GMRs

The maximum inter-storey drifts obtained from the analysis of the bare frame models for the UHS-2500 scaled records carried out by IDARC2D and DRAIN-2DX are shown in Figures 5.22 through 5.27. The DRAIN-2DX analysis is done for the ordinate method scaled records, while the IDARC2D analysis is done for both ordinate method and partial area method scaled records. Unlike the synthesized records set, the actual records set contains sixteen records, which is large enough for the mean and standard deviation values of the response parameters to be statistically meaningful. The mean values of the inter-storey drifts are shown in all figures (Figures 5.22 through 5.27) as well as the mean plus standard deviation values (mean+SD) which are later used in the evaluation of the performance instead of using the maximum values as in the case of the synthesized records. Assuming that the records used in the analysis follow a normal distribution, the confidence level of the evaluation based on the mean+SD values is approximately 84%. Which means that 84% of the maximum inter-storey drift values produced should be less than the mean+SD.

Figure 5.22 shows the maximum inter-storey drifts of the six-storey bare frame produced by IDARC2D. The maximum mean+SD inter-storey drift values are 1.23% and 1.26% for the ordinate method and partial area method scaled records, respectively, which occur at the first storey level. Figure 5.23 shows the maximum inter-storey drifts of the six-storey frame for the ordinate method scaled records obtained from DRAIN-2DX. The maximum mean+SD inter-storey drift is found to be 1.4% occurring at the first storey level.

Figure 5.24 shows the maximum inter-storey drifts of the twelve-storey bare frame obtained from the IDARC2D analysis. Due to the ordinate method and partial area method scaled records, the maximum mean+SD inter-storey drifts are 1.37% and 1.16%, respectively, both occurring at the third storey-level. The maximum inter-storey drifts due to the ordinate method scaled records produced by DRAIN-2DX are shown in Figure 5.25. The maximum mean+SD inter-storey drift occurs at the second storey level and is equal to 1.31%.

The maximum inter-storey drifts of the eighteen-storey bare frame obtained from IDARC2D are shown in Figure 5.26. The maximum mean+SD inter-storey drift due to the ordinate method scaled records is equal to 1.31% which occurs at the fifth storey level while the maximum mean+SD inter-storey drift due to the partial area method scaled records is equal to 1.19% occurring at the sixth storey level. Finally, the maximum inter-storey drifts of the eighteen-storey bare frame are shown in Figure 5.27 and the maximum mean+SD value due to the ordinate method scaled records is 1.56% which occurs at the fifth storey level.



(a) (b)

Figure 5.22: Inter-Storey Drifts of the 6 Storey Bare Frame due to the Scaled Actual GMRs (IDARC2D) Scaled Using; (a) the Ordinate Method, (b) the Partial Area Method

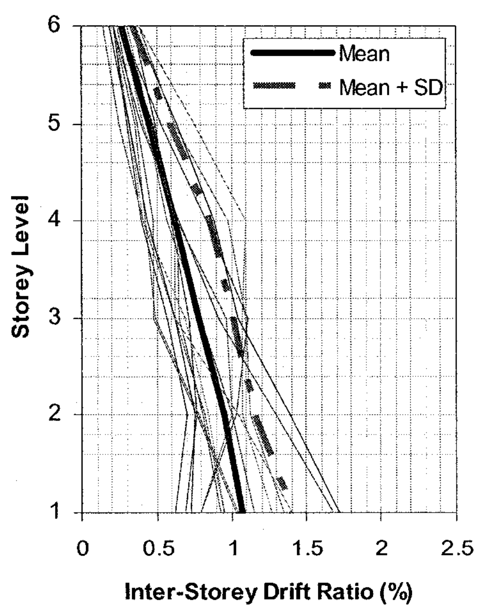


Figure 5.23: Inter-Storey Drifts of the 6 Storey Bare Frame due to the Scaled Actual GMRs (DRAIN-2DX) Scaled Using the Ordinate Method

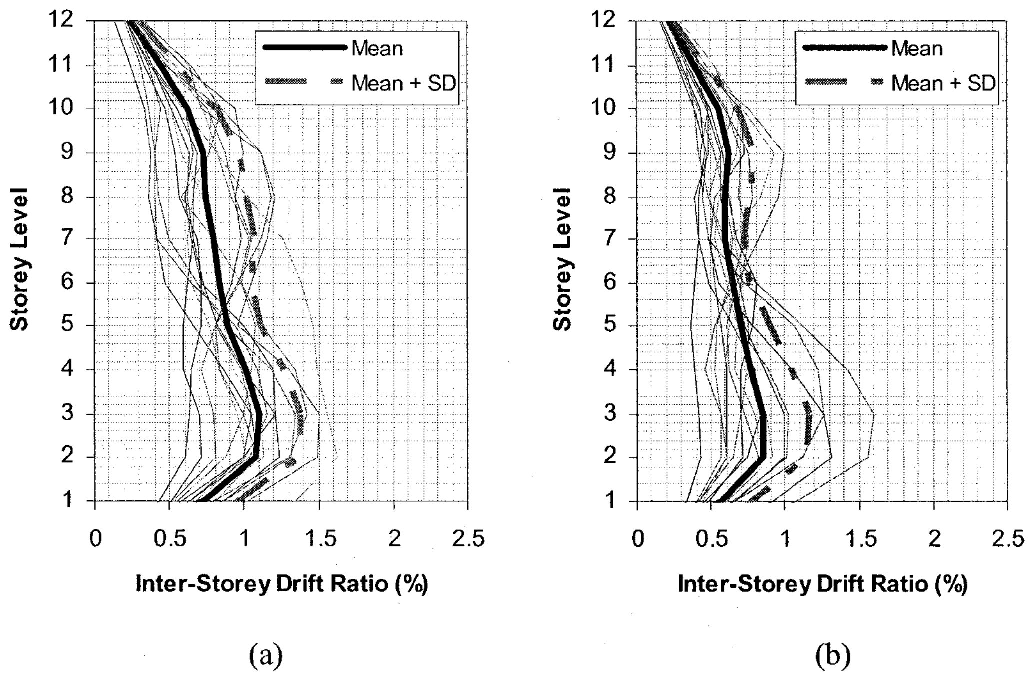


Figure 5.24: Inter-Storey Drifts of the 12 Storey Bare Frame due to the Scaled Actual GMRs (IDARC2D) Scaled Using; (a) the Ordinate Method, (b) the Partial Area Method

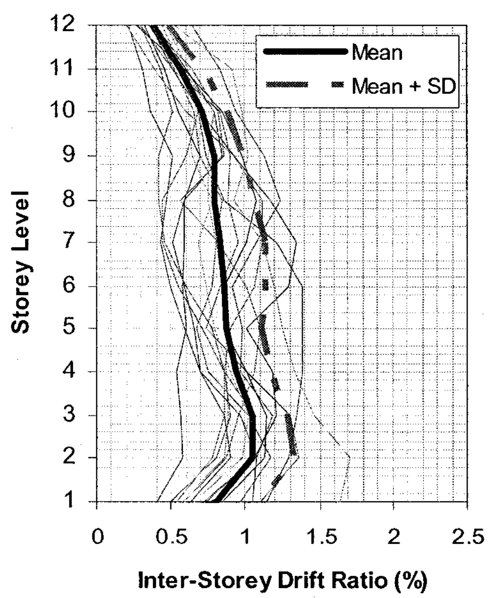


Figure 5.25: Inter-Storey Drifts of the 12 Storey Bare Frame due to the Scaled Actual GMRs (DRAIN-2DX) Scaled Using the Ordinate Method

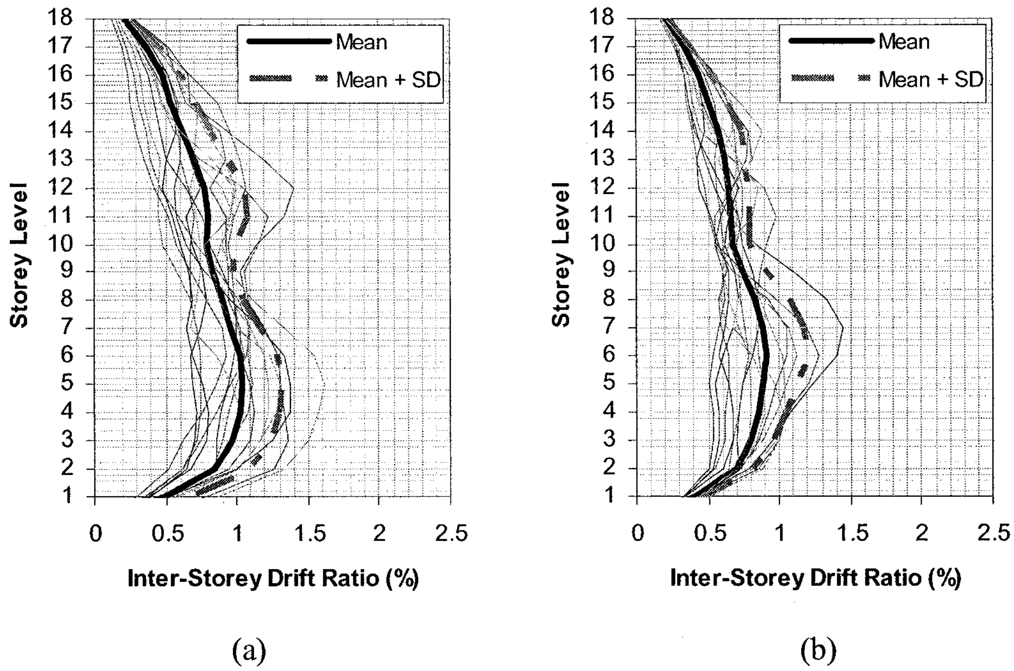


Figure 5.26: Inter-Storey Drifts of the 18 Storey Bare Frame due to the Scaled Actual GMRs (IDARC2D) Scaled Using; (a) the Ordinate Method, (b) the Partial Area Method

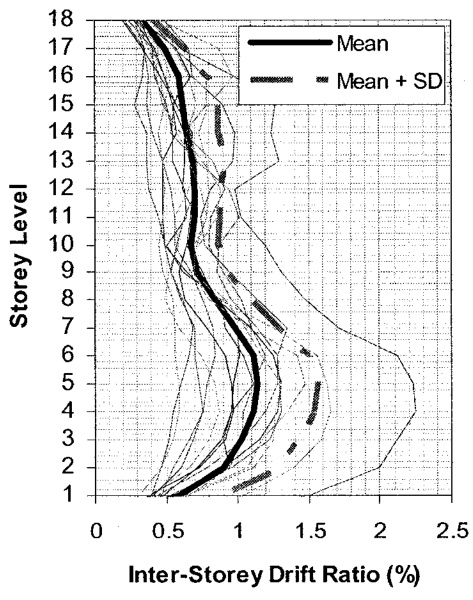


Figure 5.27: Inter-Storey Drifts of the 18 Storey Bare Frame due to the Scaled Actual GMRs (DRAIN-2DX) Scaled Using the Ordinate Method

5.3.3.7 Inter-Storey Drifts of the Infilled Frames due to the Scaled Actual GMRs

The maximum inter-storey drifts of the three infilled frames obtained from DRAIN-2DX are shown in Figure 5.28. These results are produced for the ordinate method scaled records. The periods of vibration of the infilled frames are used in the scaling of the actual records to obtain new scaling factors other than those used for the bare frames. The maximum mean+SD values are 1.02% occurring at the first storey level for the six-storey frame and 0.97% occurring at the second storey level for the eighteen-storey frame. The results of the twelve-storey building show that the structure fails due to six records and exceeds the 2.5% inter-storey drift limit due to one record. The infill panels have a negative effect on the twelve-storey building.

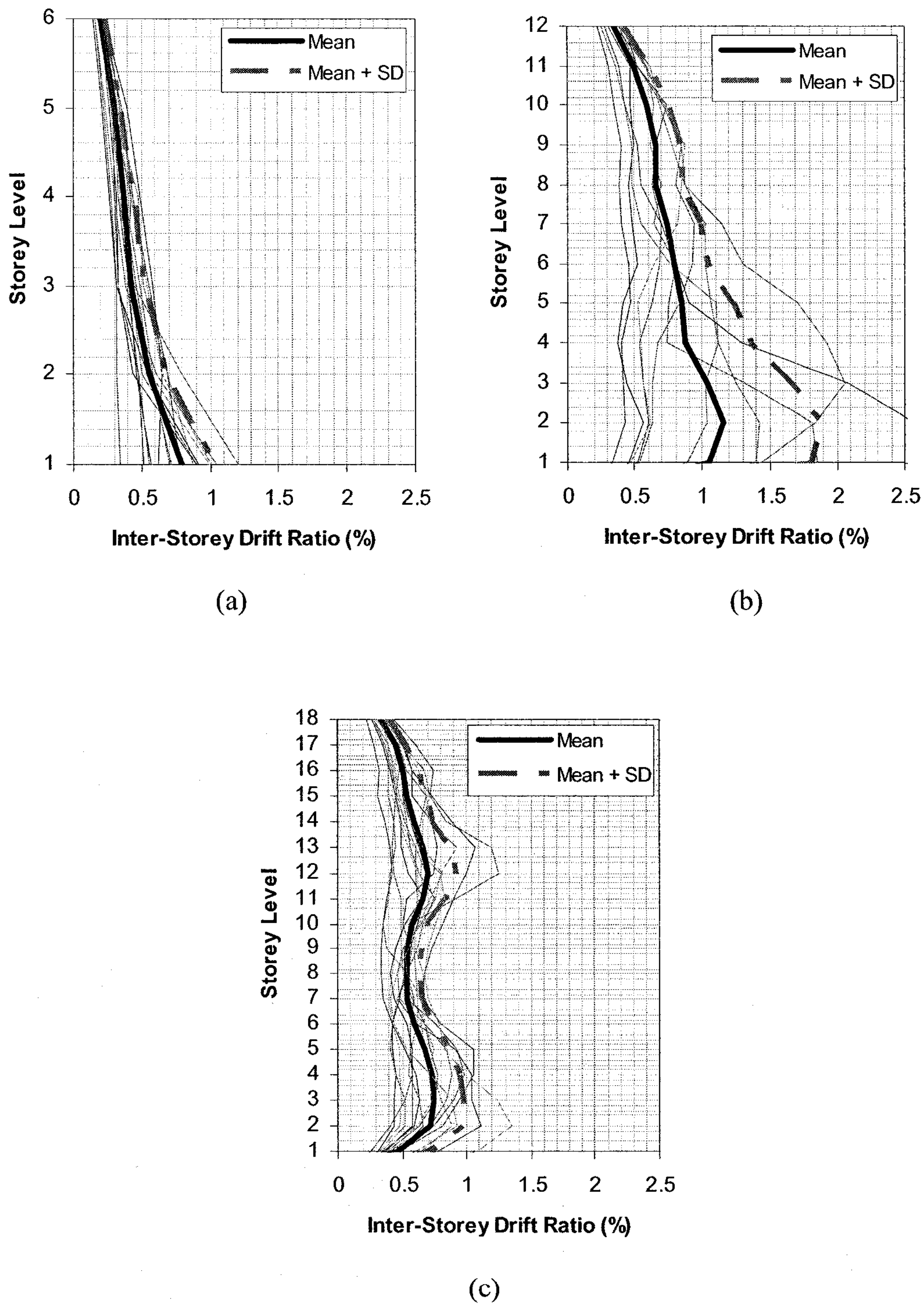


Figure 5.28: Inter-Storey Drifts of the Infilled Frames due to the Scaled Actual GMRs (DRAIN-2DX) Scaled Using the Ordinate Method; (a) 6 Storey Frame, (b) 12 Storey Frame, (c) 18 Storey Frame

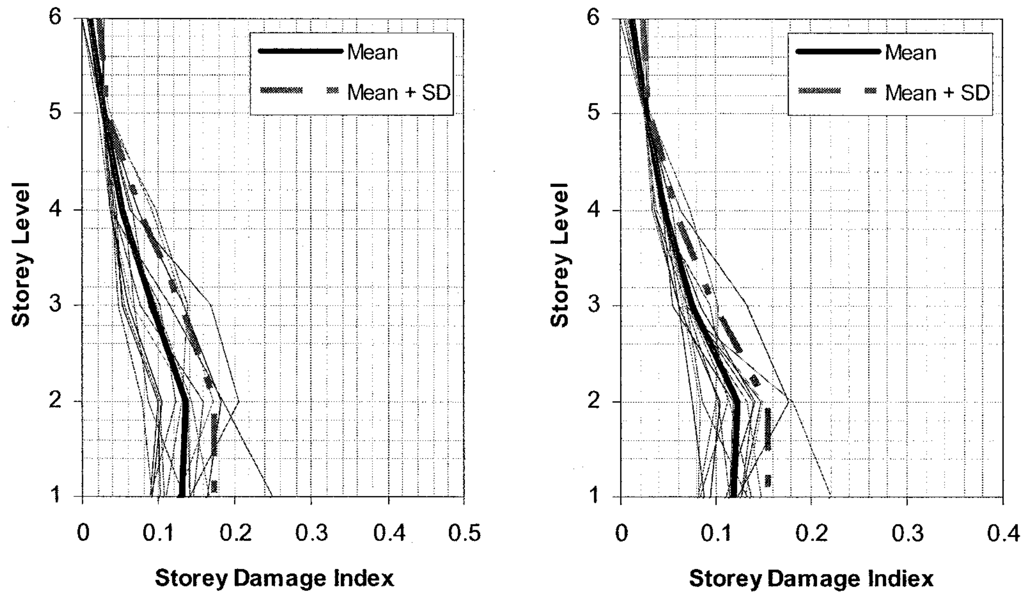
5.3.3.8 Damage Indices of the Bare Frames due to the Scaled Actual GMRs

Figures 5.29 through 5.31 show the storey damage indices of the bare frames calculated by IDARC2D due to the ordinate method and partial area method scaled records. Figure 5.29 shows that the maximum mean+SD storey damage index values of the six-storey bare frame are 0.17 and 0.15 for the ordinate method and partial area method scaled GMRs, respectively. Both values occur at the first and second storey level. The storey damage indices of the twelve-storey bare frame are shown in Figure 5.30. The maximum mean+SD values are found to be 0.26 and 0.23 for the ordinate method and partial area method scaled GMRs, respectively, occurring at the second-storey level in both cases. Finally, for the eighteen-storey bare frame, the storey damage indices are shown in Figure 5.31. The maximum mean+SD values occur at the fifth storey level and are found to be 0.24 and 0.2 for the ordinate method and partial area method scaled GMRs, respectively. The overall structural damage indices are shown in Table 5.8.

Table 5.8: Overall Structural Damage Indices of the Bare Frames due to the Scaled Actual GMRs (IDARC2D)

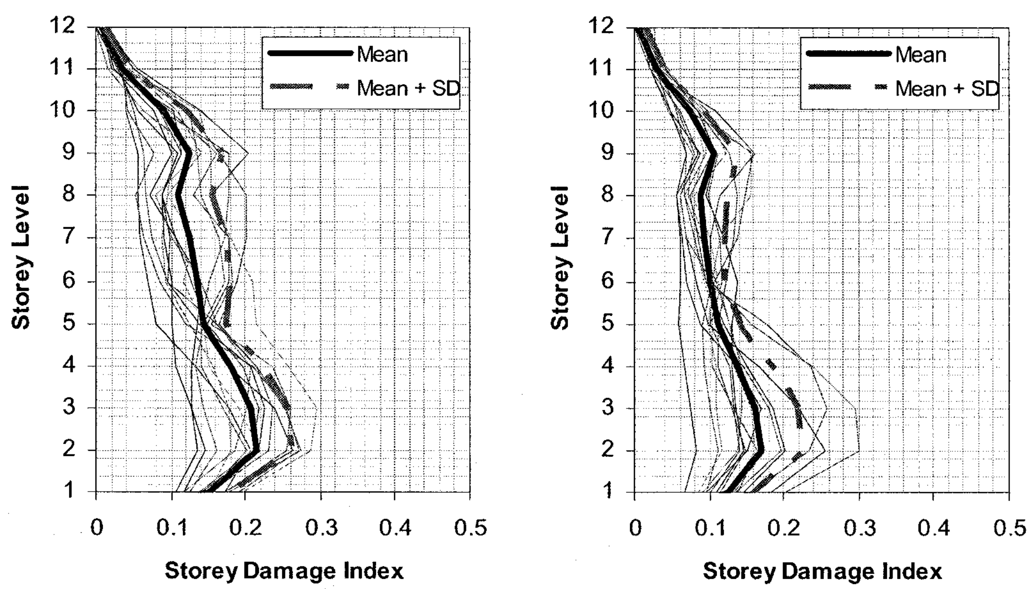
Ground Motion Record		1	2	3	4	5	6	7	8
6 Storey	Ord. Meth.	0.16	0.12	0.14	0.12	0.16	0.09	0.09	0.10
	P.A. Meth.	0.08	0.09	0.12	0.13	0.13	0.11	0.13	0.11
12 Storey	Ord. Meth.	0.12	0.16	0.20	0.16	0.18	0.18	0.20	0.17
	P.A. Meth.	0.12	0.11	0.12	0.14	0.12	0.15	0.15	0.20
18 Storey	Ord. Meth.	0.13	0.17	0.18	0.15	0.21	0.15	0.16	0.20
	P.A. Meth.	0.11	0.13	0.14	0.15	0.11	0.13	0.19	0.18

Ground Motion Record		9	10	11	12	13	14	15	16
6 Storey	Ord. Meth.	0.13	0.12	0.15	0.10	0.13	0.10	0.09	0.16
	P.A. Meth.	0.08	0.12	0.13	0.09	0.10	0.20	0.09	0.14
12 Storey	Ord. Meth.	0.14	0.21	0.17	0.16	0.19	0.14	0.12	0.22
	P.A. Meth.	0.13	0.23	0.12	0.14	0.20	0.10	0.07	0.12
18 Storey	Ord. Meth.	0.13	0.13	0.19	0.17	0.14	0.18	0.14	0.21
	P.A. Meth.	0.17	0.18	0.17	0.13	0.14	0.15	0.12	0.14



(a) (b)

Figure 5.29: Storey Damage Indices of the 6 Storey Bare Frame due to the Scaled Actual GMRs (IDARC2D) Scaled Using; (a) the Ordinate Method, (b) the Partial Area Method



(a) (b)

Figure 5.30: Storey Damage Indices of the 12 Storey Bare Frame due to the Scaled Actual GMRs (IDARC2D) Scaled Using; (a) the Ordinate Method, (b) the Partial Area Method

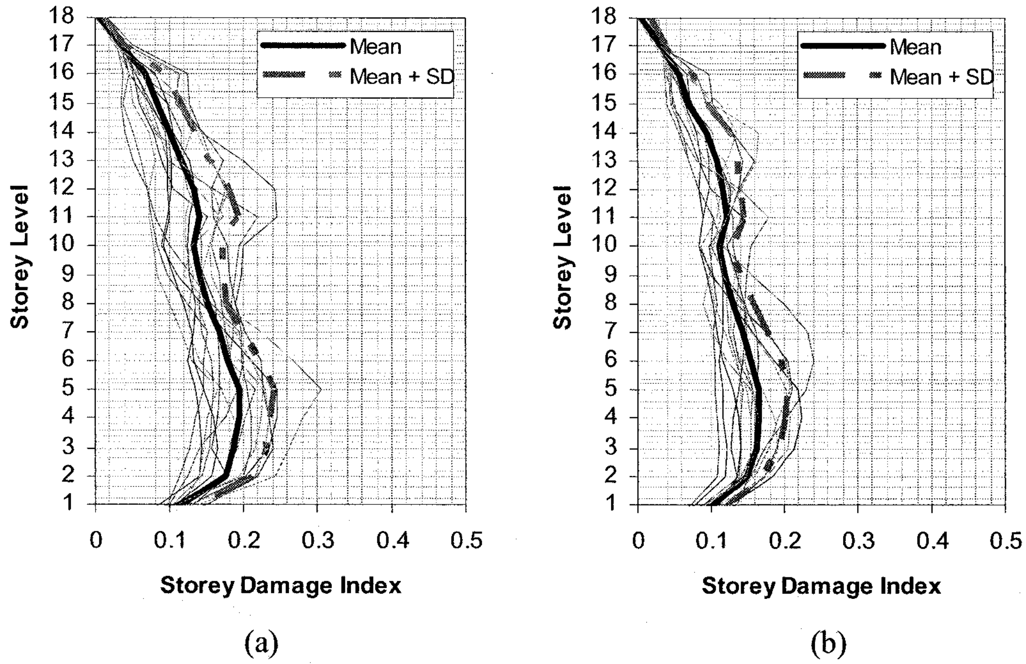


Figure 5.31: Storey Damage Indices of the 18 Storey Bare Frame due to the Scaled Actual GMRs (IDARC2D) Scaled Using; (a) the Ordinate Method, (b) the Partial Area Method

5.3.4 Performance Evaluation

The seismic performance of the structures is evaluated based on the response inter-storey drifts and damage indices obtained from the time-history dynamic analysis.

5.3.4.1 Evaluation Based on the Inter-Storey Drifts

Based on the NBCC 2005 provisions, all three structures have achieved the "Collapse Prevention" performance level required under UHS-2500 ground acceleration records since all inter-storey drift values did not exceed 2.5%. However, for a more detailed qualification of the achieved seismic performance, the SEAOC Vision 2000 (1995) provisions shown in Table 1.2 are used here to qualify the performance. Based on these provisions, the following tables are formulated to summarize the performance levels achieved. In the case of the synthesized records, the maximum values are used in the evaluation, while for the scaled records the mean+SD values are used instead. Tables 5.9 and 5.11 show the performance levels associated with the inter-storey drifts of the bare frame models due to all UHS-2500 compatible records, respectively. The results show that all records fulfill the "Collapse Prevention" or "Near Collapse" criterion specified by the NBCC 2005 provisions and even over-perform. Based on the synthesized records, the performance level achieved by all bare frames is "Near Collapse" while based on the scaled records, almost all the results point to a "Life Safe" performance. The reason that the inter-storey drifts due to the synthesized records are higher than those due to the scaled records could be explained by the fact that the maximum values are considered instead of the mean+SD values which is due to the small number of synthesized records used in the analysis.

Figure 5.32 shows the response time-history of the six-storey building due to the third actual record (*Kern County*) used in the analysis scaled using the ordinate method. The response of the structure is initially the same when using either program until several structural members yield, and then the post-yield response produced by each program is different even when the maximum inter-storey drift value is almost the same. However, because the analysis is conducted using multiple GMRs and the mean+SD values are considered as acceptable values for damage assessment, the overall performance level obtained from the analysis of both programs is very close in most cases. The variation in the mean+SD inter-storey drift values produced by IDARC2D and those produced by DRAIN-2DX ranges from 1% to 25%.

Going back to Table 5.6 where the scaling factors for the bare frames are displayed, it is clear that these factors may vary based on the scaling method applied. The results of the analysis also show the same point, although in some cases the differences in the seismic response parameters between records scaled using both scaling methods are minor, this is not always the case. The variation in the maximum mean+SD inter-storey drift values produced by the IDARC2D analysis for the actual records scaled using the ordinate method and the partial area method ranges from 2% to 16%.

Table 5.9: Performance Levels of the Bare Frames Based on the Inter-Storey Drifts due to the UHS-2500 GMRs (IDARC2D)

		Inter-Storey Drift (%)	Performance Level
6 Storey	Synthesized GMRs	2.07	Near Collapse
	Ord. Meth. Scaled GMRs	1.23	Life Safe
	P. Area Meth. Scaled GMRs	1.26	Life Safe
12 Storey	Synthesized GMRs	2.25	Near Collapse
	Ord. Meth. Scaled GMRs	1.37	Life Safe
	P. Area Meth. Scaled GMRs	1.16	Life Safe
18 Storey	Synthesized GMRs	2.12	Near Collapse
	Ord. Meth. Scaled GMRs	1.31	Life Safe
	P. Area Meth. Scaled GMRs	1.19	Life Safe

Table 5.10: Performance Levels of the Bare Frames Based on the Inter-Storey Drifts due to the UHS-2500 GMRs (DRAIN-2DX)

		Inter-Storey Drift (%)	Performance Level
6 Storey	Synthesized GMRs	2.26	Near Collapse
	Ord. Meth. Scaled GMRs	1.4	Life Safe
12 Storey	Synthesized GMRs	1.76	Near Collapse
	Ord. Meth. Scaled GMRs	1.31	Life Safe
18 Storey	Synthesized GMRs	1.62	Near Collapse
	Ord. Meth. Scaled GMRs	1.56	Near Collapse

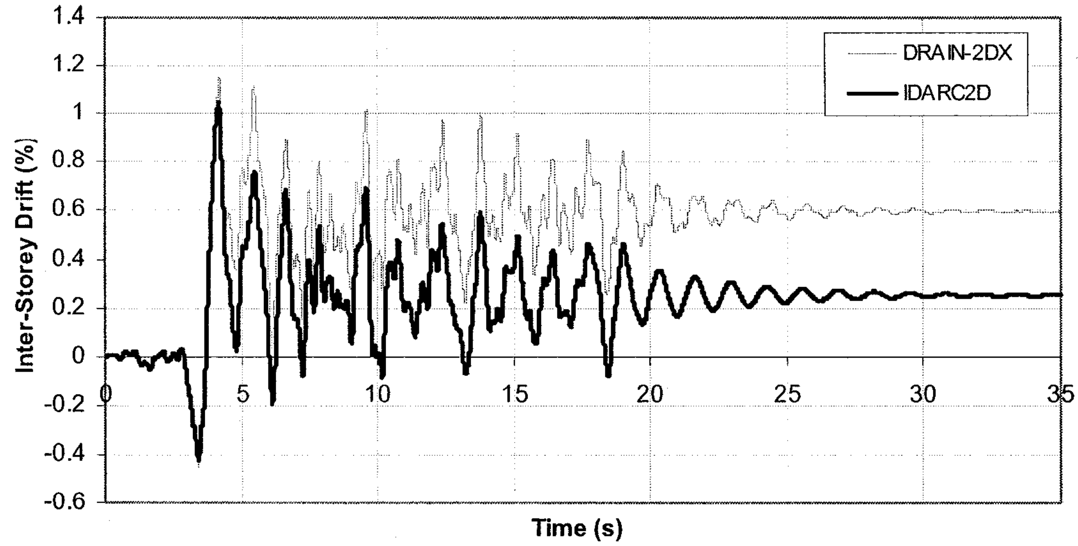


Figure 5.32: Response Time-History of the 6 Storey Bare Frame due to Scaled Actual Record No. 3 Scaled using the Ordinate Method

Table 5.11 shows the performance evaluation of the infilled frame models based on the DRAIN-2DX analysis results. The performance of the frames due to most records is found to be "Life Safe". The effect of infill panels is mostly positive and the inclusion of the panels decreases the inter-storey drifts in most cases, except in the case of the twelve-storey building analyzed for the scaled records.

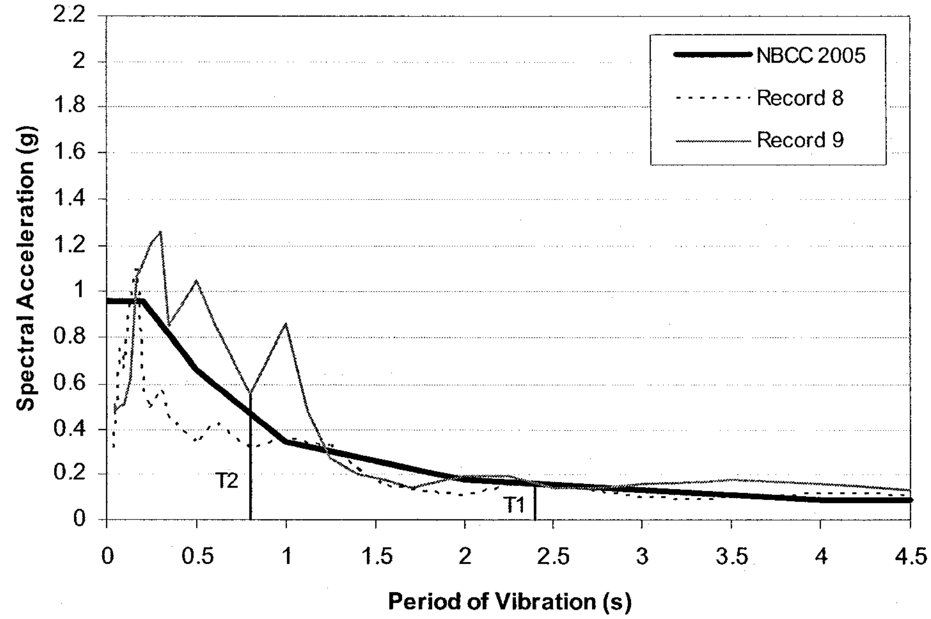
Figure 5.33 shows the response acceleration spectrum of Record 8 and Record 9 scaled using the ordinate method for the twelve-storey bare frame (Figure 5.33-(a)) and infilled frame (Figure 5.33-(b)) in comparison with the design response spectrum of the NBCC 2005. Record 8 is the record that caused 2.59% maximum inter-storey drift for the twelve-storey infilled frame and Record 9 caused failure to the same frame. The reasons for these unexpected results could be explained as follows:

- The infilled frame attracts more force from the earthquake because of its higher stiffness and the participation of the higher modes in the seismic response. The increase in strength of the structure due to the inclusion of the infill panels must at least match the increase in the attracted force or else the effect of the panels becomes negative. Infill panels improve the elastic response of the structure; however, they are not capable of enhancing the inelastic behaviour of the structure (Saatcioglu *et al.*, 2005).

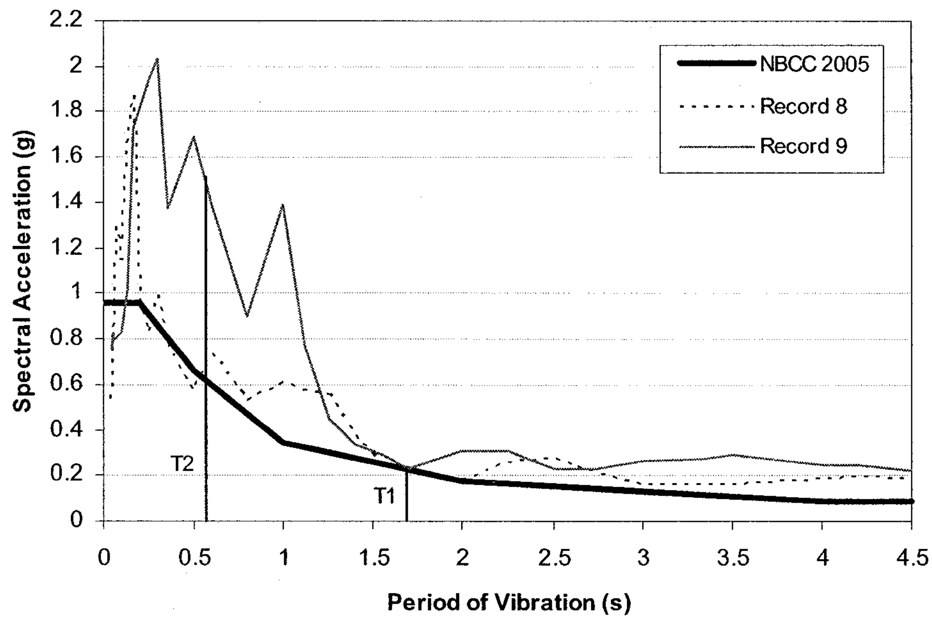
- Furthermore, the ordinate method of record scaling does not scale the records to fit the entire design spectrum prescribed by the NBCC 2005 and the spectral accelerations in the higher modes were found to be significantly higher. Because of the high fluctuations in the response spectrum in the higher modes zone, the ordinate method is not the best choice for scaling in this case. The scaling factor used for Record 9 in the case of the twelve-storey infilled frame is 4.17 using the ordinate method while it is found to be 2.09 when applying the partial area method. The partial area method seems more representative of the design spectrum in this case.

Table 5.11: Performance Levels of the Infilled Frames Based on the Inter-Storey Drifts due to the UHS-2500 GMRs (DRAIN-2DX)

		Inter-Storey Drift (%)	Performance Level
6 Storey	Synthesized GMRs	1.37	Life Safe
	Ord. Meth. Scaled GMRs	1.02	Life Safe
12 Storey	Synthesized GMRs	0.98	Life Safe
	Ord. Meth. Scaled GMRs	N/A	N/A
18 Storey	Synthesized GMRs	1.36	Life Safe
	Ord. Meth. Scaled GMRs	0.97	Life Safe



(a)



(b)

Figure 5.33: Response Spectrum of Actual Records 8 and 9; (a) 12 Storey Bare Frame, (b) 12 Storey Infilled Frame

Tables 5.12 and 5.13 show the evaluation of the inter-storey drifts of the bare frames resulting from the dynamic analysis carried out for the UHS-500 and UHS-1000 compatible synthesized records using both programs. The results show that the performance is "Life Safe" in both cases except for the eighteen-storey building which achieves an "Operational" level of performance.

Table 5.12: Performance Levels of the Bare Frames Based on the Inter-Storey Drifts due to the UHS-500 and UHS-1000 Synthesized GMRs (IDARC2D)

		Inter-Storey Drift (%)	Performance Level
6 Storey	UHS-500	0.54	Life Safe
	UHS-1000	0.86	Life Safe
12 Storey	UHS-500	0.76	Life Safe
	UHS-1000	0.54	Life Safe
18 Storey	UHS-500	0.46	Operational
	UHS-1000	0.91	Life Safe

Table 5.13: Performance Levels of the Bare Frames Based on the Inter-Storey Drifts due to the UHS-500 and UHS-1000 Synthesized GMRs (DRAIN-2DX)

		Inter-Storey Drift (%)	Performance Level
6 Storey	UHS-500	0.55	Life Safe
	UHS-1000	1.07	Life Safe
12 Storey	UHS-500	0.6	Life Safe
	UHS-1000	0.63	Life Safe
18 Storey	UHS-500	0.44	Operational
	UHS-1000	0.71	Life Safe

Table 5.14 shows the performance evaluation of the infilled frame models based on the DRAIN-2DX analysis carried out for the UHS-500 and UHS-1000 synthesized records. The performance of the infilled frames is found to be "Operational" in the case of the UHS-500 records. While, in the case of the UHS-1000 records, the performance of the

six and twelve-storey buildings is "Life Safe" and that of the eighteen-storey building is "Operational".

Table 5.14: Performance Levels of the Infilled Frames Based on the Inter-Storey Drifts due to the UHS-500 and UHS-1000 Synthesized GMRs (DRAIN-2DX)

		DRAIN-2DX	
		Inter-Storey Drift (%)	Performance Level
6 Storey	UHS-500	0.33	Operational
	UHS-1000	0.6	Life Safe
12 Storey	UHS-500	0.33	Operational
	UHS-1000	0.52	Life Safe
18 Storey	UHS-500	0.42	Operational
	UHS-1000	0.4	Operational

5.3.4.2 Evaluation Based on the Overall Damage Indices

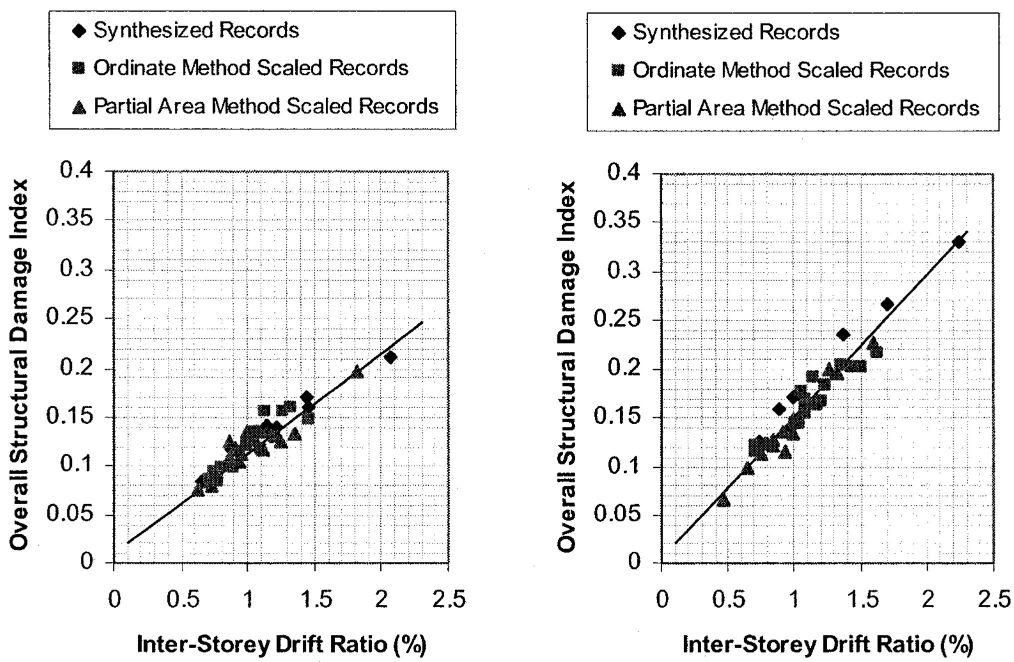
Based on the overall structural damage indices and using Table 2.1, the damage degree is assessed. As shown in Table 5.15, all structure suffered moderate damage and are repairable. The damage in the six-storey building, however, seems to be the least.

Table 5.15: Damage Degree of the Bare Frames Based on the Overall Structural Damage Indices due to the UHS-2500 GMRs (IDARC2D)

		Overall Damage Index	Damage Degree
6 Storey	Synthesized GMRs	0.12	Moderate
	Ord. Meth. Scaled GMRs	0.15	Moderate
	P. Area Meth. Scaled GMRs	0.14	Moderate
12 Storey	Synthesized GMRs	0.33	Moderate
	Ord. Meth. Scaled GMRs	0.20	Moderate
	P. Area Meth. Scaled GMRs	0.18	Moderate
18 Storey	Synthesized GMRs	0.30	Moderate
	Ord. Meth. Scaled GMRs	0.19	Moderate
	P. Area Meth. Scaled GMRs	0.17	Moderate

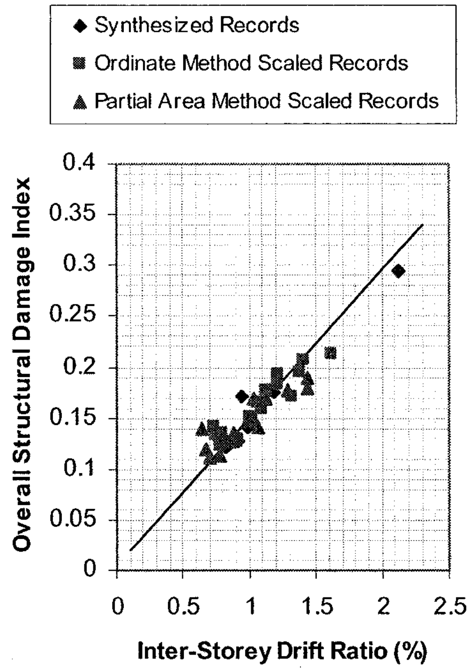
5.3.4.3 Relationship between the Overall Damage Indices and the Maximum Inter-Storey drifts

The overall damage indices are plotted in Figure 5.34 against the maximum inter-storey drifts for each UHS-2500 GMR. Both response parameters seem approximately linearly proportional. Using the least square methods, a straight line is fitted between the data points.



(a) (b)

Figure 5.34-(a) and (b): Overall Damage Indices versus Maximum Inter-Storey Drifts due to the UHS-2500 GMRs (IDARC2D); (a) 6 Storey Frame, (b) 12 Storey Frame



(c)

Figure 5.34-(c): Overall Damage Indices versus Maximum Inter-Storey Drifts due to the UHS-2500 GMRs (IDARC2D); (c) 18 Storey Frame

CHAPTER 6

6. SUMMARY AND CONCLUSIONS

6.1 SUMMARY

A set of three moment resisting frame buildings, six, twelve and eighteen storeys high are designed using the equivalent static load method (ESLM) of the NBCC 2005. Based on the NBCC 2005, the six and twelve-storey buildings are allowed to be designed using this method. However, the eighteen-storey building, because it exceeds sixty meters of height, should be designed using dynamic analysis. The ESLM in that case is used to obtain a preliminary design.

Two analysis models are used to represent the structures, a bare frame model consisting of only the reinforced concrete structural skeleton (i.e. beams and columns) and an infilled frame model that includes masonry infill panels. The infill panels are assumed to be located in the mid bays of all storeys of the three buildings, the inclusion of which is to give an assessment of the contribution of non-structural elements to the seismic performance of the structures.

Modal analysis is used in the ESLM to update the design fundamental period of the structures in order to reach a more economic design as well as to show the difference in the stiffness between the bare and infilled frames. The periods obtained from the modal analysis are also used in the scaling process of the real records used in the dynamic analysis. Non-linear static push-over analysis is conducted to examine the behaviour of the buildings when subjected to monotonic lateral loading as well as to reflect the behaviour of infill panels to the structural strength, stiffness, deformation capacity and ductility.

Using nonlinear dynamic time-history analysis, the structures are evaluated and their seismic performance and damage is assessed based on response parameters. Sixteen synthesized records are used in the dynamic analysis, eight of which correspond to a UHS-2500, four corresponding to a UHS-1000 and four corresponding to a UHS-500. Sixteen real records are scaled to match the design spectrum of Vancouver corresponding to a UHS-2500 and are used in the analysis as well. The response parameters used in the evaluation are the inter-storey drifts and the Park and Ang (1984) damage indices.

Based on the NBCC 2005 requirements, the performance levels achieved by all buildings are found acceptable. All structures have achieved the "Collapse Prevention" requirement based on the response inter-storey drifts due to the UHS-2500 compatible records. Based on the SEAOC Vision 2000 (1995) provisions, the bare frames have over-performed and reached a "Life Safe" performance level in the case of the scaled records where the mean+SD values are used in the evaluation, while they achieved a "Near Collapse" performance level in the case of the synthesized records where the maximum values are used in the evaluation because of the limited number of records used in the analysis. The results of the damage analysis show that the damage degree is moderate for all structures as they are deemed repairable based on the interpretation of the overall structural damage index shown in Table 2.1 (Park *et al.,* 1986).

The results of the analysis for the UHS-1000 records show that the bare frames have achieved a "Life Safe" performance level, while the results of the analysis for the UHS-500 records show that the six and twelve-storey bare frames have achieved a "Life Safe" performance level, while the eighteen-storey bare frame has achieved an "Operational" performance level.

The effect of non-structural elements has enhanced the performance of the structures in most cases. The performance is found to be "Life Safe" for all frames when subjected to the UHS-2500 records, except for the twelve storey frame under the scaled records. In this case, the structure fails due to six records and high inter-storey drifts are produced under several other records. The inter-storey drift exceeds the 2.5% "Collapse Prevention" limit in one case. These results are attributed to the fact that the increase in strength due to the inclusion of the infill panels does not match the increase in the attracted force from several ground motion records. Also, the scaling method applied in this case, the ordinate method, is based on the first period of vibration and because the infilled frame is stiffer than the bare frame, the higher modes have a greater effect on the response of the structure. The high fluctuations in the response spectrum near the periods of vibration of the infilled frame may suggest the use of the partial area method instead of the ordinate method.

The analysis of the infilled frames for the UHS-1000 records shows that the six and twelve-storey frames have achieved a "Life Safe" performance level while the eighteen-storey frame has achieved an "Operational" performance level. As for the UHS-500 records, all frames have reached an "Operational" performance level.

The results obtained from these analyses are proven to be reliable since two nonlinear dynamic analysis programs were used and the overall results from both programs are consistent.

6.2 CONCLUSIONS

Based on the study presented here and on the analysis results produced, the following conclusions are made. Further studies involving other building configurations should be conducted to generalize them.

- The ESLM of the NBCC 2005 produces a safe design that is in some cases a little conservative.
- Nonlinear dynamic analysis is only significant when using a large number of records where mean values and standard deviations could be used to give a reliable indication of the performance achieved.
- Although in most cases, infill panels enhance the seismic performance of a structure; this is not always the case. While infill panels improve the elastic behaviour of the structure, they do not enhance the inelastic deformation characteristics of the structure. As shown from the results of the push-over analysis, the stiffness and strength of the structure increase with the inclusion of infill panels, however, the deformation capacity and the ductility decrease. This increase in strength may not always match the increase in attracted force.
- The "strong column – weak beam" criterion is not always satisfied in the case of infilled frames since the infill panels are not considered in the design.
- The results of the analysis for real records depend a great deal on the scaling method utilized. The scaling method used in the analysis should take into account the shape of the response spectrum as well as the dynamic characteristics of the structure such as the effect of higher modes of vibration.

6.3 RECOMMENDATIONS AND SCOPE OF FUTURE WORK

The NBCC 2005 is a newly released building code. This study is one of the first studies to be conducted for the evaluation of the seismic performance of reinforced concrete moment resisting frames designed according to the new seismic provisions of the code. The recommendations for future research as well as the limitations of this research are summarized in the following paragraphs.

The NBCC 2005 specifies one objective performance (Collapse Prevention) under a specific level of seismic hazard (UHS-2500). The satisfaction of a certain target performance level under a certain level of seismic hazard does not necessarily mean that the objective performance of the structure will be satisfied under a different level of seismic hazard. To reach the goal of performance-based design and evaluation, performance objectives should be well defined and well related to different response parameters such as inter-storey drifts, roof drifts and damage indices. The stress level could also be used in the performance evaluation. To reach this goal, extensive experimental and analytical research should be conducted.

An attempt has been made in this research work to quantify and qualify the performance of symmetric and regular reinforced concrete moment resisting plane frames based on existing definitions of performance levels. This evaluation while useful doesn't take into account the three dimensional nature of real buildings. Out of plane and torsional effects were not included in this study. Further research work is needed to investigate the performance of three dimensional buildings with mass and shape irregularities which reflect many structures constructed nowadays. Furthermore, the effect of strength and stiffness degradation as well as pinching should was neglected in this study and should be considered in future research work.

The effect of non-structural elements such as infill panels is studied. However, based on the method of modelling the infill panels, the results could have a large variance. It is then also suggested to examine the effect of non-structural elements on the seismic behaviour of structures using different analysis models and also different infill panel arrangements. Infill panels should also be taken in to account in the design.

It is also recommended to conduct further studies to examine the performance of different types of lateral load resisting systems such as shear walls and frames with dampers.

REFERENCES

Aschheim M., 2004, *Yield Point Spectra: a Simple Alternative to the Capacity Spectrum Method*, ASCE online research library http://www.ascelibrary.org

Adams J. and Atkinson G., 2003, *Development of Seismic Hazard Maps for the Proposed 2005 Edition of the National Building Code of Canada*, Canadian Journal of Civil Engineering, Volume 30, pp. 255-271

Bagchi A., 2001, *Evaluation of the Seismic Performance of Reinforced Concrete Buildings*, PhD thesis, Department of Civil and Environmental Engineering, Carlton University, Ottawa, ON

Bozorgnia Y. and Bertero V., 2004, *Damage Spectrum and its Application to Performance-Based Earthquake Engineering*, 13th World Conference on Earthquake Engineering, Vancouver, BC, Paper No. 1497

Bozorgnia Y. and Bertero V., 2001, *Improved Shaking and Damage Parameters for Post-Earthquake Applications,* Proceedings of the SMIP01 Seminar on Utilization of Strong-Motion Data, Los Angeles, CA, pp. 1-22

Choi E., Nam T.H. and Cho B.S., 2005, *A New Concept of Isolation Bearings for Highway Steel Bridges Using Shape Memory Alloys,* Canadian Journal of Civil Engineering, Volume 32, pp. 957-967

CJCE, 2003, *Special Issue on NBCC 2005*, Canadian Journal of Civil Engineering, Volume30, No.4

Concrete Design Handbook, 2004, Cement Association of Canada, Ottawa, ON

REFERENCES

El Kafrawy O., Yousuf M. and Bagchi A., *Computer Aided Design and Analysis of RC Frame Buildings Subjected to Earthquakes*, Joint International Conference on Computing and Decision Making in Civil and Building Engineering (ICCCBE-XI), Montreal, QC, Paper No. IC-198

FEMA-306, 1998, *Evaluation of Earthquake Damaged Concrete and Masonry Wall Buildings – Basic Procedures Manual,* prepared by the Applied Technology Council (ATC) for the Partnership for Response and Recovery and funded by the Federal Emergency Management Agency, Washington, DC

FEMA-356, 2000, *Prestandard and Commentary for the Seismic Rehabilitation of Buildings,* prepared by the American Society of Civil Engineers (ASCE) for the Federal Emergency Management Agency, Washington, DC

FEMA-368, 2001, *NEHRP Recommended Provisions for Seismic Regulations for New Buildings and Other Structures,* prepared by the Building Seismic Safety Council (BSSC) for the Federal Emergency Management Agency, Washington, DC

Foo S., Naumoski N. and Saatcioglu M., 2001, *Seismic Hazard, Building Codes and Mitigation Options for Canadian Buildings,* Office of Critical Infrastructure Protection and Emergency Preparedness, 2nd Floor, Jackson Building, Ottawa, ON

Hamburger R., Rojahn C., Moehle J., Bachman R., Comartin C. and Whittaker A., 2004, *the ATC-58 Project: Development of Next-Generation Performance-Based Earthquake Engineering Design Criteria for Buildings,* 13th World Conference on Earthquake Engineering, Vancouver, BC, Paper No. 1819

REFERENCES

Humar J. and Mahgoub M.A., 2003, *Determination of Seismic Design Forces by Equivalent Static Load Method*, Canadian Journal of Civil Engineering, Volume 30, pp. 287-307

Heidebrecht A.C., 2003, *Overview of Seismic Provisions of the Proposed 2005 Edition of the National Building Code of Canada*, Canadian Journal of Civil Engineering, Volume 30, pp. 241-254

IBC 2003, 2003, *International Building Code,* International Code Council, Country Club Hills, IL

IDARC2D, 2006, A Computer Program for Seismic Inelastic Structural Analysis, Department of Civil, Structural and Environmental Engineering, University at Buffalo, Buffalo, NY, http://civil.eng.buffalo.edu/idarc2d50/

Kunnath S.K., Reinhorn A.M. and Lobo R.F., 1992, *IDARC Version 3.0: A Program for the Inelastic Damage Analysis of Reinforced Concrete Structures,* Report No. NCEER-92-0022, National Center for Earthquake Engineering Research, State University of New York at Buffalo, Buffalo, NY

Mainstone R.J. and Weeks G.A., 1970, *The Influence of Bounding Frame on the Racking Stiffness and Strength of Brick Walls,* 2nd International Brick and Masonry Conference, Building Research Establishment, Watford, England, pp. 165-171

Mainstone R.J., 1971, *On the Stiffness and Strength of Infilled Frames,* Proceedings of the Institution of Civil Engineers, London, England, Supplemental Volume IV, Paper 7360s, pp. 57-90

McGuire R., 1977, *Seismic Design Spectra and Mapping Procedures using Hazard Analysis Based Directly on Oscillator Response*, International Journal of Earthquake Engineering and Structural Dynamics, Volume 5, pp. 211-234

Moehle J. and Deierlein G.G., 2004, *A Framework Methodology for Performance-Based Earthquake Engineering,* 13th World Conference on Earthquake Engineering, Vancouver, BC, Paper No. 679

Naumoski N., Saatcioglu M. and Amiri-Hormozaki K., 2004, *Effects of Scaling of Earthquake Excitations on the Dynamic Response of Reinforced Concrete Frame Buildings*, 13th World Conference on Earthquake Engineering, Vancouver, BC, Paper No. 2917

NBCC 2005, 2005, *National Building Code of Canada,* Canadian Commission on Building and Fire Codes, National Research Council of Canada, Ottawa, ON

NRCAN, 2006, *Natural Resources Canada: the Atlas of Canada*, http://atlas.nrcan.gc.ca/site/english/maps/environment/naturalhazards/majorearthquakes

Park Y.J., Ang A.H.S. and Wen Y.K., 1984, *Seismic Damage Analysis and Damage-Limiting Design of R/C Buildings,* Civil Engineering Studies, Technical Report No. SRS 516, University of Illinois, Urbana, IL

Park Y.J., Ang A.H.S. and Wen Y.K., 1986, *Damage-Limiting Aseismic Design of Buildings,* Earthquake Spectra

Prakash V., Powell G.H., Graham H, Campbell S., Scott D., 1993, *Drain-2DX: Static and Dynamic Analysis of Inelastic Plane Structure*, Department of Civil Engineering, University of California, Berkeley, CA

PEER, 2006, *Pacific Earthquake Engineering Research Center: NGA Database*, http://peer.berkeley.edu/nga/

Public Safety and Emergency Preparedness Canada, 2006, *Earthquakes*, http://www.getprepared.ca/risks/earthquakes_e.asp#1

Saatcioglu M., Serrato F. and Foo S., 2005, *Seismic Performance of Masonry Infill Walls Retrofitted with CFRP Sheets*, 7th International Symposium on Fiber Reinforced Polymer (FRP) Reinforcement for Concrete Structures, Kansas City, MO

SEAOC Vision 2000 Committee, 1995, *Performance-Based Seismic Engineering*, Report prepared by the Structural Engineers Association of California, Sacramento, CA

SEAOC Vision 2000 Committee, 1996, *A Framework for Performance-Based Earthquake engineering*, Report prepared by the Structural Engineers Association of California, Sacramento, CA

Valles R.E., Reinhorn A.M., Kunnath S.K., Li C. and Madan A., 1996, *IDARC 2D Version 4.0: A Program for the Inelastic Damage Analysis of Buildings,* Report No. NCEER-96-0010, National Center for Earthquake Engineering Research, State University of New York at Buffalo, Buffalo, NY

Yousuf M., 2006, *Seismic Design and Performance of Steel Moment Resisting Frame Buildings Designed Using NBCC 2005,* M.A.Sc. Thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, QC

APPENDIX A. Visual Basic Code for the Pre-Processor

Workbook

```
Sub WorkBook_Open()

Start

End Sub
```

Module

```
Public NSO, NFR, NSONodalWeights, NFRNodalWeights As Integer
Public NCON, NSTL, NMSR As Integer
Public NCONProperties, NSTLProperties, NMSRProperties As Integer
Public MCOL, MBEM, MTRN, MIW As Integer
Public MCOLProperties, MBEMProperties, MTRNProperties, MIWProperties As Integer
Public NCOL, NBEM, NTRN, NIW As Integer
Public NCOLProperties, NBEMProperties, NTRNProperties, NIWProperties As Integer
Public NLU, NLJ, NLM, NLC As Integer
Public NLUProperties, NLJProperties, NLMProperties, NLCProperties As Integer
Public NLDED, NLDED2 As Integer
Public NLDEDAnalysis, NLDED2Analysis As Integer
Public NVLN(), NDUP(), NSTLD() As Integer
Public oo1, oo2, oo3 As Integer
Public OriginRow1, OriginCol1, OriginRow2, OriginCol2, OriginRow3, OriginCol3 As Integer
Public OriginRow4, OriginCol4, OriginRow5, OriginCol5, OriginRow6, OriginCol6 As Integer
Public OriginRow7, OriginCol7, OriginRow8, OriginCol8, OriginRow9, OriginCol9 As Integer
Public OriginRow10, OriginCol10, OriginRow11, OriginCol11, OriginRow12, OriginCol12 As Integer
Public OriginRow13, OriginCol13, OriginRow14, OriginCol14, OriginRow15, OriginCol15 As Integer
Public OriginRow16, OriginCol16, OriginRow17, OriginCol17, OriginRow18, OriginCol18 As Integer
Public OriginRow19, OriginCol19, OriginRow20, OriginCol20, OriginRow21, OriginCol21 As Integer
Public OriginRow22, OriginCol22, OriginRow23, OriginCol23, OriginRow24, OriginCol24 As Integer
Public i, j, k, X As Integer
Public FrameNumber As String
Public NumberOfTables, LastActiveRow As Integer
Public TenKey As Integer
Public xPushOver As Integer
Public NumberOfHistoryPoints, NumberOfHistoryPointsAnalysis As Integer
Public NPDEL, IFLEX, IFLEXDIST, IWV, NDATA, DTINP, ICNTRL, NPTS, NHYS, NHYSProperties,
JSTP As Integer
Public NAMEW, WHFILE, WVFILE As String
Public IU, KC, IMC, IMS, KB, KT, IPT, IMT, IL, IBN, LF, IBM, IFV, LV, JV, MSTEPS, NMOD, ITYP,
ITDMP, IR, IBILINEAR As Integer
Public FU, FL, FM1, FM2, FV, PMAX, DRFILM, POWER1, POWER2, PX(), GMAXH, GMAXV,
DTCAL, TDUR, DAMP, F(), HBD, HC, HBE, HS As Double
Public M, ITC, IC, JC, LBC, LTC, ITB, LB, IB, JLB, JRB, ITT, LT, IWT, JWT, IFT, JFT, IFnum, ITIW,
LTop, LBot, JL, JR, JBMT As Integer
```

APPENDIX A. VISUAL BASIC CODE FOR THE PRE-PROCESSOR

```
Public HIGT(), IM, FC, EC, EPS0, FT, EPSU, ZF, FS, FSU, ES, ESH, EPSH, FM, FMCR, EPSM, VM,
SIGMM, CFM As Double
Public AN, AMLC, RAMC1, RAMC2, KHYSC, D, B, DC, AT, HBS, CEF, DOuter, CVR, DST, NBAR,
BDIA, IOCRL As Double
Public AMLB, RAMB1, RAMB2, KHYSB, BSL, TSL, BC, AT1, AT2, AKV, ARV, ALV, TMP, VLMP,
VHMP, QMPC, QMPB, QMPJ As Double
Public refresh As Boolean
```

Public Sub Start()

```
'Specifying the origin point ("General Information" label cell)
OriginRow1 = 4
OriginCol1 = 2
OriginRow2 = 4
OriginCol2 = 2
OriginRow3 = 17
OriginCol3 = 2
OriginRow4 = 4
OriginCol4 = 2
OriginRow5 = 13
OriginCol5 = 2
OriginRow6 = 29
OriginCol6 = 2
OriginRow7 = 4
OriginCol7 = 2
OriginRow8 = 13
OriginCol8 = 2
OriginRow9 = 22
OriginCol9 = 2
OriginRow10 = 22
OriginCol10 = 2
OriginRow11 = 29
OriginCol11 = 2
OriginRow12 = 17

OriginCol12 = 14
OriginRow13 = 4
OriginCol13 = 2
OriginRow14 = 11
OriginCol14 = 2
OriginRow15 = 18
OriginCol15 = 2
OriginRow16 = 25
OriginCol16 = 2
OriginRow17 = 41
OriginCol17 = 2
OriginRow18 = 4
OriginCol18 = 2
OriginRow19 = 11
OriginCol19 = 2
OriginRow20 = 18
OriginCol20 = 2
OriginRow21 = 25
OriginCol21 = 2
OriginRow22 = 4
OriginCol22 = 2
```

```
OriginRow23 = 31
OriginCol23 = 2

NSO = Worksheets(1).Cells(3 + OriginRow1, 4 + OriginCol1)
NFR = Worksheets(1).Cells(3 + OriginRow1, 19 + OriginCol1)

Worksheets(1).Protect ("555")
Worksheets(2).Protect ("555")
Worksheets(3).Protect ("555")
Worksheets(4).Protect ("555")
Worksheets(5).Protect ("555")
Worksheets(6).Protect ("555")
Worksheets(7).Protect ("555")

Worksheets(1).EnableSelection = xlUnlockedCells
Worksheets(2).EnableSelection = xlUnlockedCells
Worksheets(3).EnableSelection = xlUnlockedCells
Worksheets(4).EnableSelection = xlUnlockedCells
Worksheets(5).EnableSelection = xlUnlockedCells
Worksheets(6).EnableSelection = xlUnlockedCells
Worksheets(7).EnableSelection = xlUnlockedCells

ReDim NVLN(NFR)
For i = 1 To NFR
  NVLN(i) = Worksheets(1).Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1)
  oo1 = (i - i Mod 5) / 5
Next

refresh = True
Worksheets(1).ComboBox1.Clear
Worksheets(1).ComboBox1.AddItem "Linear Distribution"
Worksheets(1).ComboBox1.AddItem "Uniform Distribution"
Worksheets(1).ComboBox2.Clear
Worksheets(1).ComboBox2.AddItem "Spread Plasticity"
Worksheets(1).ComboBox2.AddItem "Concentrated Plasticity"
Worksheets(1).ComboBox3.Clear
Worksheets(1).ComboBox3.AddItem "Ignore"
Worksheets(1).ComboBox3.AddItem "Include"
Worksheets(1).ComboBox4.Clear
Worksheets(1).ComboBox4.AddItem "inch, kips"
Worksheets(1).ComboBox4.AddItem "mm, kN"
Worksheets(7).ComboBox1.Clear
Worksheets(7).ComboBox1.AddItem "Force Control"
Worksheets(7).ComboBox1.AddItem "Displacement Control"
Worksheets(7).ComboBox2.Clear
Worksheets(7).ComboBox2.AddItem "Uniform"
Worksheets(7).ComboBox2.AddItem "Linear (Inverted Triangle)"
Worksheets(7).ComboBox2.AddItem "Exponential"
Worksheets(7).ComboBox2.AddItem "Modal Adaptive Push-Over"
Worksheets(7).ComboBox2.AddItem "User Defined"
Worksheets(7).ComboBox3.Clear
Worksheets(7).ComboBox3.AddItem "Mass Proportional"
Worksheets(7).ComboBox3.AddItem "Stiffness Proportional"
Worksheets(7).ComboBox3.AddItem "Rayleigh Proportional"
Worksheets(7).ComboBox4.Clear
Worksheets(7).ComboBox4.AddItem "Force Control"
```

```
Worksheets(7).ComboBox4.AddItem "Displacement Control"
Worksheets(1).ComboBox1.ListIndex = Worksheets(1).Cells(1, 41)
Worksheets(1).ComboBox2.ListIndex = Worksheets(1).Cells(1, 42)
Worksheets(1).ComboBox3.ListIndex = Worksheets(1).Cells(1, 43)
Worksheets(1).ComboBox4.ListIndex = Worksheets(1).Cells(4, 43) - 1
Worksheets(7).ComboBox1.ListIndex = Worksheets(7).Cells(1, 39)
Worksheets(7).ComboBox2.ListIndex = Worksheets(7).Cells(1, 30)
Worksheets(7).ComboBox3.ListIndex = Worksheets(7).Cells(2, 37)
Worksheets(7).ComboBox4.ListIndex = Worksheets(7).Cells(3, 37)
Worksheets(7).CheckBox1.Value = Worksheets(7).Cells(5, 37)
refresh = False

UserForm1.TextBox1.Value = Worksheets(7).Cells(5, 25)
UserForm1.TextBox2.Value = Worksheets(7).Cells(6, 26)

End Sub
```

Sheet1 (General Info)

```
Private Sub ComboBox1_Change()

If refresh = False Then
   Worksheets(1).Unprotect ("555")
   IFLEXDIST = ComboBox1.ListIndex
   Cells(1, 41) = IFLEXDIST
   Worksheets(1).Protect ("555")
End If

End Sub

Private Sub ComboBox2_Change()

If refresh = False Then
   Worksheets(1).Unprotect ("555")
   IFLEX = Worksheets(1).ComboBox2.ListIndex
   Cells(1, 42) = IFLEX
   Worksheets(1).Protect ("555")
End If

End Sub

Private Sub ComboBox3_Change()

If refresh = False Then
   Worksheets(1).Unprotect ("555")
   NPDEL = Worksheets(1).ComboBox3.ListIndex
   Cells(1, 43) = NPDEL
   Worksheets(1).Protect ("555")
End If

End Sub
```

APPENDIX A. VISUAL BASIC CODE FOR THE PRE-PROCESSOR

Private Sub ComboBox4_Change()

```
If refresh = False Then
   Worksheets(1).Unprotect ("555")
   IU = Worksheets(1).ComboBox4.ListIndex + 1
   Cells(4, 43) = IU
   Worksheets(1).Protect ("555")
End If
```

End Sub

Private Sub Worksheet_Change(ByVal Target As Excel.Range)

```
Dim StringNumber As String

'Specifying the origin point ("General Information" label cell)
OriginRow1 = 4
OriginCol1 = 2

'Editing the "Number of Stories" cell
If Not Application.Intersect(Target, Cells(3 + OriginRow1, 4 + OriginCol1)) Is Nothing Then
   Worksheets(1).Unprotect ("555")
''Minimum and Maximum limit for the "Number of Stories"
   If Cells(3 + OriginRow1, 4 + OriginCol1) <= 0 Then
      MsgBox ("Invalid Number of Stories")
      Cells(3 + OriginRow1, 4 + OriginCol1) = NSO
      Cells(3 + OriginRow1, 4 + OriginCol1).Select
      Exit Sub
   ElseIf Cells(3 + OriginRow1, 4 + OriginCol1) > 40 Then
      MsgBox ("Maximum Number of Stories is 40")
      Cells(3 + OriginRow1, 4 + OriginCol1) = NSO
      Cells(3 + OriginRow1, 4 + OriginCol1).Select
      Exit Sub
   End If
''Inserting or deleting Rows
   If NSO >= NFR Then
      If (Cells(3 + OriginRow1, 4 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 4 + OriginCol1) - 1) Mod 5) /
5 > (NSO - 1 - (NSO - 1) Mod 5) / 5 Then
         For i = 1 To 2 * (((Cells(3 + OriginRow1, 4 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 4 +
OriginCol1) - 1) Mod 5) / 5) - ((NSO - 1 - (NSO - 1) Mod 5) / 5))
            Rows(9 + OriginRow1 + i + 2 * (NSO - 1 - (NSO - 1) Mod 5) / 5).EntireRow.Insert
         Next
      ElseIf (Cells(3 + OriginRow1, 4 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 4 + OriginCol1) - 1) Mod
5) / 5 < (NSO - 1 - (NSO - 1) Mod 5) / 5 Then
         If NFR > Cells(3 + OriginRow1, 4 + OriginCol1) Then
            X = NFR
         Else
            X = Cells(3 + OriginRow1, 4 + OriginCol1)
         End If
         Range(Rows(10 + OriginRow1 + 2 * (X - 1 - (X - 1) Mod 5) / 5), Rows(9 + OriginRow1 + 2 * (X -
1 - (X - 1) Mod 5) / 5 + 2 * (((NSO - 1 - (NSO - 1) Mod 5) / 5) - ((X - 1 - (X - 1) Mod 5) / 5)))).Select
         Selection.Delete
      End If
   ElseIf NFR > NSO Then
      If (Cells(3 + OriginRow1, 4 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 4 + OriginCol1) - 1) Mod 5) /
5 > (NFR - 1 - (NFR - 1) Mod 5) / 5 Then
```

```
        For i = 1 To 2 * (((Cells(3 + OriginRow1, 4 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 4 +
OriginCol1) - 1) Mod 5) / 5) - ((NFR - 1 - (NFR - 1) Mod 5) / 5))
          Rows(9 + OriginRow1 + i + 2 * (NFR - 1 - (NFR - 1) Mod 5) / 5).EntireRow.Insert
        Next
      End If
    End If
''Expanding the "Storey Heights" vector
    If NSO < Cells(3 + OriginRow1, 4 + OriginCol1) Then
      oo1 = (NSO - NSO Mod 5) / 5
      For i = 1 To Cells(3 + OriginRow1, 4 + OriginCol1) - NSO
''''''''Expanding the "Storey Heights" vector
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + NSO + i - 5 * oo1).Locked = False
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + NSO + i - 5 * oo1).Borders.Color = RGB(0, 0,
0)
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + NSO + i - 5 * oo1).Font.Size = 10
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + NSO + i - 5 * oo1).Font.Color = vbRed
        StringNumber = NSO + i
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + NSO + i - 5 * oo1) = "St#" + StringNumber
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + NSO + i - 5 * oo1).Font.Size = 7
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + NSO + i - 5 * oo1).Font.Color = RGB(0, 128, 0)
        oo1 = (NSO + i - (NSO + i) Mod 5) / 5
      Next
''Shortening the "Storey Heights" vector
    ElseIf NSO > Cells(3 + OriginRow1, 4 + OriginCol1) Then
      oo1 = (Cells(3 + OriginRow1, 4 + OriginCol1) - Cells(3 + OriginRow1, 4 + OriginCol1) Mod 5) / 5
      If NFR < Cells(3 + OriginRow1, 4 + OriginCol1) Then
        X = (oo1 + 1) * 5 - Cells(3 + OriginRow1, 4 + OriginCol1)
      Else
        X = ((NFR - NFR Mod 5) / 5 + 1) * 5 - Cells(3 + OriginRow1, 4 + OriginCol1)
      End If
      For i = 1 To X
''''''''Shortening the "Storey Heights" vector
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + Cells(3 + OriginRow1, 4 + OriginCol1) + i - 5 *
oo1).Locked = True
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + Cells(3 + OriginRow1, 4 + OriginCol1) + i - 5 *
oo1).Borders.Color = RGB(255, 255, 255)
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + Cells(3 + OriginRow1, 4 + OriginCol1) + i - 5 *
oo1).Interior.Color = RGB(255, 255, 255)
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + Cells(3 + OriginRow1, 4 + OriginCol1) + i - 5 *
oo1) = Empty
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + Cells(3 + OriginRow1, 4 + OriginCol1) + i - 5 *
oo1) = Empty
        oo1 = (Cells(3 + OriginRow1, 4 + OriginCol1) + i - (Cells(3 + OriginRow1, 4 + OriginCol1) + i)
Mod 5) / 5
      Next
    End If
    NSO = Cells(3 + OriginRow1, 4 + OriginCol1)
    Worksheets(1).Protect ("555")
End If

'Editing the "Number of Frames" cell
If Not Application.Intersect(Target, Cells(3 + OriginRow1, 19 + OriginCol1)) Is Nothing Then
    Worksheets(1).Unprotect ("555")
''Minimum and Maximum limit for the "Number of Frames"
    If Cells(3 + OriginRow1, 19 + OriginCol1) <= 0 Then
      MsgBox ("Invalid Number of Frames")
```

```
        Cells(3 + OriginRow1, 19 + OriginCol1) = NFR
        Cells(3 + OriginRow1, 19 + OriginCol1).Select
        Exit Sub
    ElseIf Cells(3 + OriginRow1, 19 + OriginCol1) > 20 Then
        MsgBox ("Maximum Number of Frames is 20")
        Cells(3 + OriginRow1, 19 + OriginCol1) = NFR
        Cells(3 + OriginRow1, 19 + OriginCol1).Select
        Exit Sub
    End If
""Inserting or deleting Rows
    If NFR >= NSO Then
        If (Cells(3 + OriginRow1, 19 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 19 + OriginCol1) - 1) Mod
5) / 5 > (NFR - 1 - (NFR - 1) Mod 5) / 5 Then
            For i = 1 To 2 * (((Cells(3 + OriginRow1, 19 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 19 +
OriginCol1) - 1) Mod 5) / 5) - ((NFR - 1 - (NFR - 1) Mod 5) / 5))
                Rows(9 + OriginRow1 + i + 2 * (NFR - 1 - (NFR - 1) Mod 5) / 5).EntireRow.Insert
            Next
        ElseIf (Cells(3 + OriginRow1, 19 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 19 + OriginCol1) - 1)
Mod 5) / 5 < (NFR - 1 - (NFR - 1) Mod 5) / 5 Then
            If NSO > Cells(3 + OriginRow1, 19 + OriginCol1) Then
                X = NSO
            Else
                X = Cells(3 + OriginRow1, 19 + OriginCol1)
            End If
            Range(Rows(10 + OriginRow1 + 2 * (X - 1 - (X - 1) Mod 5) / 5), Rows(9 + OriginRow1 + 2 * (X -
1 - (X - 1) Mod 5) / 5 + 2 * (((NFR - 1 - (NFR - 1) Mod 5) / 5) - ((X - 1 - (X - 1) Mod 5) / 5)))).Select
            Selection.Delete
        End If
    ElseIf NSO > NFR Then
        If (Cells(3 + OriginRow1, 19 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 19 + OriginCol1) - 1) Mod
5) / 5 > (NSO - 1 - (NSO - 1) Mod 5) / 5 Then
            For i = 1 To 2 * (((Cells(3 + OriginRow1, 19 + OriginCol1) - 1 - (Cells(3 + OriginRow1, 19 +
OriginCol1) - 1) Mod 5) / 5) - ((NSO - 1 - (NSO - 1) Mod 5) / 5))
                Rows(9 + OriginRow1 + i + 2 * (NSO - 1 - (NSO - 1) Mod 5) / 5).EntireRow.Insert
            Next
        End If
    End If
""Expanding the "Number of Duplicate Frames" & "Number of Column Lines" vector
    If NFR < Cells(3 + OriginRow1, 19 + OriginCol1) Then
        oo1 = (NFR - NFR Mod 5) / 5
        For i = 1 To Cells(3 + OriginRow1, 19 + OriginCol1) - NFR
""""""Expanding the "Number of Duplicate Frames" & "Number of Column Lines" vector
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + NFR + i - 5 * oo1).Locked = False
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + NFR + i - 5 * oo1).Borders.Color = RGB(0, 0,
0)
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + NFR + i - 5 * oo1).Font.Size = 10
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + NFR + i - 5 * oo1).Font.Color = vbRed
            StringNumber = NFR + i
            Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + NFR + i - 5 * oo1) = "Fr#" + StringNumber
            Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + NFR + i - 5 * oo1).Font.Size = 7
            Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + NFR + i - 5 * oo1).Font.Color = RGB(0, 128, 0)
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1).Locked = False
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1).Borders.Color = RGB(0, 0,
0)
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1).Font.Size = 10
            Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1).Font.Color = vbRed
```

```
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1) = "Fr#" + StringNumber
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1).Font.Size = 7
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + NFR + i - 5 * oo1).Font.Color = RGB(0, 128,
0)
        oo1 = (NFR + i - (NFR + i) Mod 5) / 5
      Next
'''Shortening the "Number of Duplicate Frames" & "Number of Column Lines" vector
   ElseIf NFR > Cells(3 + OriginRow1, 19 + OriginCol1) Then
      oo1 = (Cells(3 + OriginRow1, 19 + OriginCol1) - Cells(3 + OriginRow1, 19 + OriginCol1) Mod 5) / 5
      If NSO < Cells(3 + OriginRow1, 19 + OriginCol1) Then
        X = (oo1 + 1) * 5 - Cells(3 + OriginRow1, 19 + OriginCol1)
      Else
        X = ((NSO - NSO Mod 5) / 5 + 1) * 5 - Cells(3 + OriginRow1, 19 + OriginCol1)
      End If
      For i = 1 To X
''''''''Shortening the "Number of Duplicate Frames" & "Number of Column Lines" vector
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + Cells(3 + OriginRow1, 19 + OriginCol1) + i - 5
* oo1).Locked = True
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + Cells(3 + OriginRow1, 19 + OriginCol1) + i - 5
* oo1).Borders.Color = RGB(255, 255, 255)
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + Cells(3 + OriginRow1, 19 + OriginCol1) + i - 5
* oo1).Interior.Color = RGB(255, 255, 255)
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + Cells(3 + OriginRow1, 19 + OriginCol1) + i - 5
* oo1) = Empty
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + Cells(3 + OriginRow1, 19 + OriginCol1) + i - 5
* oo1) = Empty
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + Cells(3 + OriginRow1, 19 + OriginCol1) + i -
5 * oo1).Locked = True
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + Cells(3 + OriginRow1, 19 + OriginCol1) + i -
5 * oo1).Borders.Color = RGB(255, 255, 255)
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + Cells(3 + OriginRow1, 19 + OriginCol1) + i -
5 * oo1).Interior.Color = RGB(255, 255, 255)
        Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + Cells(3 + OriginRow1, 19 + OriginCol1) + i -
5 * oo1) = Empty
        Cells(9 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + Cells(3 + OriginRow1, 19 + OriginCol1) + i -
5 * oo1) = Empty
        oo1 = (Cells(3 + OriginRow1, 19 + OriginCol1) + i - (Cells(3 + OriginRow1, 19 + OriginCol1) + i)
Mod 5) / 5
      Next
   End If
   NFR = Cells(3 + OriginRow1, 19 + OriginCol1)
   Worksheets(1).Protect ("555")
End If
```

End Sub

Sheet2 (Nodal Weights)

Private Sub Worksheet_Activate()

```
ReDim Preserve NVLN(NFR)
oo1 = 0
```

```
For i = 1 To NFR
   NVLN(i) = Worksheets(1).Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + i - 5 * oo1)
   oo1 = (i - i Mod 5) / 5
Next
OriginRow2 = 4
OriginCol2 = 2
oo1 = 0
NSONodalWeights = Cells(1, 26)
NFRNodalWeights = Cells(2, 26)
'Checking that the input hasn't changed
If NSONodalWeights = NSO Then
   If NFRNodalWeights = NFR Then
      For i = 1 To NFR
         If Cells(i, 27) <> NVLN(i) Then
            GoTo skip0:
         End If
      Next
      Exit Sub
   End If
End If
skip0:
'Checking that the "Number of Column Lines" is given for each non-identical frame
For i = 1 To NFR
   If Worksheets(1).Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + i - 5 * oo1) = Empty Then
      Worksheets(1).Select
      MsgBox ("Column Lines Input Incomplete")
      Exit Sub
   End If
   oo1 = (i - i Mod 5) / 5
Next
oo1 = 0
oo2 = 0
NumberOfTables = 0
For i = 1 To NFR
   X = ((NVLN(i) - 1) - (NVLN(i) - 1) Mod 10) / 10
   X = X + 1
   NumberOfTables = NumberOfTables + X
Next
Worksheets(2).Unprotect ("555")
LastActiveRow = Cells.Find(What:="*", After:=Cells(1, 1), searchorder:=xlByRows,
searchdirection:=xlPrevious).Row
Worksheets(2).Range(Cells(6, 1), Cells(LastActiveRow, 13)).Clear
Worksheets(2).Range(Cells(6, 1), Cells(LastActiveRow, 13)).Font.Color = RGB(255, 0, 0)
Worksheets(2).Range(Cells(6, 1), Cells(LastActiveRow, 13)).HorizontalAlignment = xlCenter
Worksheets(2).Range(Cells(6, 1), Cells(LastActiveRow, 13)).Interior.Color = RGB(255, 255, 255)
'Forming the Nodal Weights tables
For i = 1 To NFR
   FrameNumber = i
   Cells(OriginRow2 + 3 + oo2 * (4 + NSO), OriginCol2).Font.Color = RGB(0, 0, 128)
   Cells(OriginRow2 + 3 + oo2 * (4 + NSO), OriginCol2) = "Frame Number " + FrameNumber
   Cells(OriginRow2 + 3 + oo2 * (4 + NSO), OriginCol2).HorizontalAlignment = xlLeft
   Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2).Font.Color = RGB(0, 0, 128)
   Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2) = "Storey #"
   Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2).Interior.Color = RGB(255, 255, 153)
   Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2).Borders.Color = RGB(0, 0, 0)
   For j = 1 To Worksheets(1).Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + i - 5 * oo1)
```

```
    If j > 1 Then
      If (j - 1) Mod 10 = 0 Then
        oo2 = oo2 + 1
        oo3 = oo3 + 1
        TenKey = 10
      End If
    End If
    Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2 + j - TenKey * oo3).Font.Color = RGB(0, 0,
128)
    Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2 + j - TenKey * oo3).Interior.Color = RGB(255,
255, 153)
    Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2 + j - TenKey * oo3).Borders.Color = RGB(0, 0,
0)
    Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2 + j - TenKey * oo3) = j
    If TenKey = 10 Then
      Cells(OriginRow2 + 3 + oo2 * (4 + NSO), OriginCol2).Font.Color = RGB(0, 0, 128)
      Cells(OriginRow2 + 3 + oo2 * (4 + NSO), OriginCol2) = "Frame Number " + FrameNumber + "
(Cont.)"
      Cells(OriginRow2 + 3 + oo2 * (4 + NSO), OriginCol2).HorizontalAlignment = xlLeft
      Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2).Borders.Color = RGB(0, 0, 0)
      Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2).Font.Color = RGB(0, 0, 128)
      Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2) = "Storey #"
      Cells(OriginRow2 + 5 + oo2 * (4 + NSO), OriginCol2).Interior.Color = RGB(255, 255, 153)
    End If
    For k = 1 To NSO
      Cells(OriginRow2 + 5 + k + oo2 * (4 + NSO), OriginCol2 + j - TenKey * oo3).Locked = False
      Cells(OriginRow2 + 5 + k + oo2 * (4 + NSO), OriginCol2 + j - TenKey * oo3).Borders.Color =
RGB(0, 0, 0)
      Cells(OriginRow2 + 5 + k + oo2 * (4 + NSO), OriginCol2).Borders.Color = RGB(0, 0, 0)
      Cells(OriginRow2 + 5 + k + oo2 * (4 + NSO), OriginCol2).Font.Color = RGB(0, 0, 128)
      Cells(OriginRow2 + 5 + k + oo2 * (4 + NSO), OriginCol2) = k
    Next
  Next
  oo1 = (i - i Mod 5) / 5
  oo2 = oo2 + 1
  oo3 = 0
  TenKey = 0
Next
NSONodalWeights = NSO
Cells(1, 26) = NSONodalWeights
NFRNodalWeights = NFR
Cells(2, 26) = NFRNodalWeights
For i = 1 To NFR
  Cells(i, 27) = NVLN(i)
Next
Worksheets(2).Protect ("555")

End Sub
```

Sheet3 (Material Properties)

Private Sub Worksheet_Activate()

```
OriginRow6 = 29
OriginCol6 = 2
OriginRow7 = 4
OriginCol7 = 2
OriginRow8 = 13
OriginCol8 = 2
OriginRow9 = 22
OriginCol9 = 2
OriginRow23 = 31
OriginCol23 = 2

NCONProperties = Cells(1, 26)
NSTLProperties = Cells(2, 26)
NMSRProperties = Cells(3, 26)
NHYSProperties = Cells(4, 26)
Worksheets(3).Unprotect ("555")
If NSO > NFR Then
  X = ((NSO - 1 - (NSO - 1) Mod 5) / 5) * 2
Else
  X = ((NFR - 1 - (NFR - 1) Mod 5) / 5) * 2
End If
NCON = Worksheets(1).Cells(OriginRow6 + 3 + X, OriginCol6 + 11)
'Hiding or unhiding the 1st table
If NCON <> NCONProperties Then
  If NCON = 0 Then
    For i = 1 To 7
      Rows(OriginRow7 - 2 + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow7 + 6 + NCONProperties).EntireRow.Hidden = True
    Rows(OriginRow7 + 7 + NCONProperties).EntireRow.Hidden = True
  ElseIf NCONProperties = 0 Then
    For i = 1 To 7
      Rows(OriginRow7 - 2 + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow7 + 6 + NCONProperties).EntireRow.Hidden = False
    Rows(OriginRow7 + 7 + NCONProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 1st table
If NCON > NCONProperties Then
  For i = 1 To (NCON - NCONProperties)
    Rows(6 + OriginRow7 + NCONProperties + i).EntireRow.Insert
    For j = 1 To 7
      Cells(5 + OriginRow7 + NCONProperties + i, OriginCol7 - 1 + j).Borders.Color = RGB(0, 0, 0)
      Cells(5 + OriginRow7 + NCONProperties + i, OriginCol7 - 1 + j).Locked = False
    Next
    Cells(5 + OriginRow7 + NCONProperties + i, OriginCol7) = NCONProperties + i
    Cells(5 + OriginRow7 + NCONProperties + i, OriginCol7).HorizontalAlignment = xlCenter
    Cells(5 + OriginRow7 + NCONProperties + i, OriginCol7).Locked = True
  Next
ElseIf NCON < NCONProperties Then
  Range(Rows(6 + OriginRow7 + NCON), Rows(5 + OriginRow7 + NCONProperties)).Select
  Selection.Delete
End If
NCONProperties = NCON
```

```
Cells(1, 26) = NCONProperties
NSTL = Worksheets(1).Cells(OriginRow6 + 5 + X, OriginCol6 + 11)
'Hiding or unhiding the 2nd table
If NSTL <> NSTLProperties Then
  If NSTL = 0 Then
    For i = 1 To 7
      Rows(OriginRow8 - 2 + NCON + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow8 + 6 + NCON + NSTLProperties).EntireRow.Hidden = True
    Rows(OriginRow8 + 7 + NCON + NSTLProperties).EntireRow.Hidden = True
  ElseIf NSTLProperties = 0 Then
    For i = 1 To 7
      Rows(OriginRow8 - 2 + NCON + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow8 + 6 + NCON + NSTLProperties).EntireRow.Hidden = False
    Rows(OriginRow8 + 7 + NCON + NSTLProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 2nd table
If NSTL > NSTLProperties Then
  For i = 1 To (NSTL - NSTLProperties)
    Rows(6 + OriginRow8 + NCON + NSTLProperties + i).EntireRow.Insert
    For j = 1 To 6
      Cells(5 + OriginRow8 + NCON + NSTLProperties + i, OriginCol8 - 1 + j).Borders.Color = RGB(0,
0, 0)
      Cells(5 + OriginRow8 + NCON + NSTLProperties + i, OriginCol8 - 1 + j).Locked = False
    Next
    Cells(5 + OriginRow8 + NCON + NSTLProperties + i, OriginCol8) = NSTLProperties + i
    Cells(5 + OriginRow8 + NCON + NSTLProperties + i, OriginCol8).HorizontalAlignment = xlCenter
    Cells(5 + OriginRow8 + NCON + NSTLProperties + i, OriginCol8).Locked = True
  Next
ElseIf NSTL < NSTLProperties Then
  Range(Rows(6 + OriginRow8 + NCON + NSTL), Rows(5 + OriginRow8 + NCON +
NSTLProperties)).Select
  Selection.Delete
End If
NSTLProperties = NSTL
Cells(2, 26) = NSTLProperties
NMSR = Worksheets(1).Cells(OriginRow6 + 7 + X, OriginCol6 + 11)
'Hiding or unhiding the 3rd table
If NMSR <> NMSRProperties Then
  If NMSR = 0 Then
    For i = 1 To 7
      Rows(OriginRow9 - 2 + NCON + NSTL + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow9 + 6 + NCON + NSTL + NMSRProperties).EntireRow.Hidden = True
    Rows(OriginRow9 + 7 + NCON + NSTL + NMSRProperties).EntireRow.Hidden = True
  ElseIf NMSRProperties = 0 Then
    For i = 1 To 7
      Rows(OriginRow9 - 2 + NCON + NSTL + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow9 + 6 + NCON + NSTL + NMSRProperties).EntireRow.Hidden = False
    Rows(OriginRow9 + 7 + NCON + NSTL + NMSRProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 3rd table
```

```
If NMSR > NMSRProperties Then
  For i = 1 To (NMSR - NMSRProperties)
    Rows(6 + OriginRow9 + NCON + NSTL + NMSRProperties + i).EntireRow.Insert
    For j = 1 To 7
      Cells(5 + OriginRow9 + NCON + NSTL + NMSRProperties + i, OriginCol9 - 1 + j).Borders.Color
= RGB(0, 0, 0)
      Cells(5 + OriginRow9 + NCON + NSTL + NMSRProperties + i, OriginCol9 - 1 + j).Locked = False
    Next
    Cells(5 + OriginRow9 + NCON + NSTL + NMSRProperties + i, OriginCol9) = NMSRProperties + i
    Cells(5 + OriginRow9 + NCON + NSTL + NMSRProperties + i, OriginCol9).HorizontalAlignment =
xlCenter
    Cells(5 + OriginRow9 + NCON + NSTL + NMSRProperties + i, OriginCol9).Locked = True
  Next
ElseIf NMSR < NMSRProperties Then
  Range(Rows(6 + OriginRow9 + NCON + NSTL + NMSR), Rows(5 + OriginRow9 + NCON + NSTL +
NMSRProperties)).Select
  Selection.Delete
End If
NMSRProperties = NMSR
Cells(3, 26) = NMSRProperties
NHYS = Worksheets(1).Cells(OriginRow6 + 9 + X, OriginCol6 + 11)
'Hiding or unhiding the 4th table
If NHYS <> NHYSProperties Then
  If NHYS = 0 Then
    For i = 1 To 7
      Rows(OriginRow23 - 2 + NCON + NSTL + NMSR + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow23 + 6 + NCON + NSTL + NMSR + NHYSProperties).EntireRow.Hidden = True
    Rows(OriginRow23 + 7 + NCON + NSTL + NMSR + NHYSProperties).EntireRow.Hidden = True
  ElseIf NHYSProperties = 0 Then
    For i = 1 To 7
      Rows(OriginRow23 - 2 + NCON + NSTL + NMSR + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow23 + 6 + NCON + NSTL + NMSR + NHYSProperties).EntireRow.Hidden = False
    Rows(OriginRow23 + 7 + NCON + NSTL + NMSR + NHYSProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 4th table
If NHYS > NHYSProperties Then
  For i = 1 To (NHYS - NHYSProperties)
    Rows(6 + OriginRow23 + NCON + NSTL + NMSR + NHYSProperties + i).EntireRow.Insert
    For j = 1 To 6
      Cells(5 + OriginRow23 + NCON + NSTL + NMSR + NHYSProperties + i, OriginCol23 - 1 +
j).Borders.Color = RGB(0, 0, 0)
      Cells(5 + OriginRow23 + NCON + NSTL + NMSR + NHYSProperties + i, OriginCol23 - 1 +
j).Locked = False
    Next
    Cells(5 + OriginRow23 + NCON + NSTL + NMSR + NHYSProperties + i, OriginCol23) =
NHYSProperties + i
    Cells(5 + OriginRow23 + NCON + NSTL + NMSR + NHYSProperties + i,
OriginCol23).HorizontalAlignment = xlCenter
    Cells(5 + OriginRow23 + NCON + NSTL + NMSR + NHYSProperties + i, OriginCol23).Locked =
True
  Next
ElseIf NHYS < NHYSProperties Then
```

```
   Range(Rows(6 + OriginRow23 + NCON + NSTL + NMSR + NHYS), Rows(5 + OriginRow23 + NCON
+ NSTL + NMSR + NHYSProperties)).Select
   Selection.Delete
End If
NHYSProperties = NHYS
Cells(4, 26) = NHYSProperties
Worksheets(3).Protect ("555")
```

End Sub

Sheet4 (Element Properties)

Private Sub Worksheet_Activate()

```
OriginRow3 = 17
OriginCol3 = 2
OriginRow4 = 4
OriginCol4 = 2
OriginRow5 = 13
OriginCol5 = 2
OriginRow10 = 22
OriginCol10 = 2
OriginRow11 = 29
OriginCol11 = 2

MCOLProperties = Cells(1, 26)
MBEMProperties = Cells(2, 26)
MTRNProperties = Cells(3, 26)
MIWProperties = Cells(4, 26)
Worksheets(4).Unprotect ("555")
If NSO > NFR Then
   X = ((NSO - 1 - (NSO - 1) Mod 5) / 5) * 2
Else
   X = ((NFR - 1 - (NFR - 1) Mod 5) / 5) * 2
End If
MCOL = Worksheets(1).Cells(OriginRow3 + 3 + X, OriginCol3 + 9)
'Hiding or unhiding the 1st table
If MCOL <> MCOLProperties Then
   If MCOL = 0 Then
      For i = 1 To 7
         Rows(OriginRow4 - 2 + i).EntireRow.Hidden = True
      Next
      Rows(OriginRow4 + 6 + MCOLProperties).EntireRow.Hidden = True
      Rows(OriginRow4 + 7 + MCOLProperties).EntireRow.Hidden = True
   ElseIf MCOLProperties = 0 Then
      For i = 1 To 7
         Rows(OriginRow4 - 2 + i).EntireRow.Hidden = False
      Next
      Rows(OriginRow4 + 6 + MCOLProperties).EntireRow.Hidden = False
      Rows(OriginRow4 + 7 + MCOLProperties).EntireRow.Hidden = False
   End If
End If
```

```
'Expanding or shortening the 1st table
If MCOL > MCOLProperties Then
  For i = 1 To (MCOL - MCOLProperties)
    Rows(6 + OriginRow4 + MCOLProperties + i).EntireRow.Insert
    For j = 1 To 15
      Cells(5 + OriginRow4 + MCOLProperties + i, OriginCol4 - 1 + j).Borders.Color = RGB(0, 0, 0)
      Cells(5 + OriginRow4 + MCOLProperties + i, OriginCol4 - 1 + j).Locked = False
    Next
    Cells(5 + OriginRow4 + MCOLProperties + i, OriginCol4) = MCOLProperties + i
    Cells(5 + OriginRow4 + MCOLProperties + i, OriginCol4).HorizontalAlignment = xlCenter
    Cells(5 + OriginRow4 + MCOLProperties + i, OriginCol4).Locked = True
  Next
ElseIf MCOL < MCOLProperties Then
  Range(Rows(6 + OriginRow4 + MCOL), Rows(5 + OriginRow4 + MCOLProperties)).Select
  Selection.Delete
End If
MCOLProperties = MCOL
Cells(1, 26) = MCOLProperties
MBEM = Worksheets(1).Cells(OriginRow3 + 5 + X, OriginCol3 + 9)
'Hiding or unhiding the 2nd table
If MBEM <> MBEMProperties Then
  If MBEM = 0 Then
    For i = 1 To 7
      Rows(OriginRow5 - 2 + MCOL + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow5 + 6 + MCOL + MBEMProperties).EntireRow.Hidden = True
    Rows(OriginRow5 + 7 + MCOL + MBEMProperties).EntireRow.Hidden = True
  ElseIf MBEMProperties = 0 Then
    For i = 1 To 7
      Rows(OriginRow5 - 2 + MCOL + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow5 + 6 + MCOL + MBEMProperties).EntireRow.Hidden = False
    Rows(OriginRow5 + 7 + MCOL + MBEMProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 2nd table
If MBEM > MBEMProperties Then
  For i = 1 To (MBEM - MBEMProperties)
    Rows(6 + OriginRow5 + MCOL + MBEMProperties + i).EntireRow.Insert
    For j = 1 To 16
      Cells(5 + OriginRow5 + MCOL + MBEMProperties + i, OriginCol5 - 1 + j).Borders.Color =
RGB(0, 0, 0)
      Cells(5 + OriginRow5 + MCOL + MBEMProperties + i, OriginCol5 - 1 + j).Locked = False
    Next
    Cells(5 + OriginRow5 + MCOL + MBEMProperties + i, OriginCol5) = MBEMProperties + i
    Cells(5 + OriginRow5 + MCOL + MBEMProperties + i, OriginCol5).HorizontalAlignment = xlCenter
    Cells(5 + OriginRow5 + MCOL + MBEMProperties + i, OriginCol5).Locked = True
  Next
ElseIf MBEM < MBEMProperties Then
  Range(Rows(6 + OriginRow5 + MCOL + MBEM), Rows(5 + OriginRow5 + MCOL +
MBEMProperties)).Select
  Selection.Delete
End If
MBEMProperties = MBEM
Cells(2, 26) = MBEMProperties
MTRN = Worksheets(1).Cells(OriginRow3 + 7 + X, OriginCol3 + 9)
```

```
'Hiding or unhiding the 3rd table
If MTRN <> MTRNProperties Then
  If MTRN = 0 Then
    For i = 1 To 5
      Rows(OriginRow10 - 2 + MCOL + MBEM + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow10 + 4 + MCOL + MBEM + MTRNProperties).EntireRow.Hidden = True
    Rows(OriginRow10 + 5 + MCOL + MBEM + MTRNProperties).EntireRow.Hidden = True
  ElseIf MTRNProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow10 - 2 + MCOL + MBEM + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow10 + 4 + MCOL + MBEM + MTRNProperties).EntireRow.Hidden = False
    Rows(OriginRow10 + 5 + MCOL + MBEM + MTRNProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 3rd table
If MTRN > MTRNProperties Then
  For i = 1 To (MTRN - MTRNProperties)
    Rows(4 + OriginRow10 + MCOL + MBEM + MTRNProperties + i).EntireRow.Insert
    For j = 1 To 4
      Cells(3 + OriginRow10 + MCOL + MBEM + MTRNProperties + i, OriginCol10 - 1 +
j).Borders.Color = RGB(0, 0, 0)
      Cells(3 + OriginRow10 + MCOL + MBEM + MTRNProperties + i, OriginCol10 - 1 + j).Locked =
False
    Next
    Cells(3 + OriginRow10 + MCOL + MBEM + MTRNProperties + i, OriginCol10) = MTRNProperties
+ i
    Cells(3 + OriginRow10 + MCOL + MBEM + MTRNProperties + i,
OriginCol10).HorizontalAlignment = xlCenter
    Cells(3 + OriginRow10 + MCOL + MBEM + MTRNProperties + i, OriginCol10).Locked = True
  Next
ElseIf MTRN < MTRNProperties Then
  Range(Rows(4 + OriginRow10 + MCOL + MBEM + MTRN), Rows(3 + OriginRow10 + MCOL +
MBEM + MTRNProperties)).Select
  Selection.Delete
End If
MTRNProperties = MTRN
Cells(3, 26) = MTRNProperties
MIW = Worksheets(1).Cells(OriginRow3 + 9 + X, OriginCol3 + 9)
'Hiding or unhiding the 4th table
If MIW <> MIWProperties Then
  If MIW = 0 Then
    For i = 1 To 5
      Rows(OriginRow11 - 2 + MCOL + MBEM + MTRN + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow11 + 4 + MCOL + MBEM + MTRN + MIWProperties).EntireRow.Hidden = True
    Rows(OriginRow11 + 5 + MCOL + MBEM + MTRN + MIWProperties).EntireRow.Hidden = True
  ElseIf MIWProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow11 - 2 + MCOL + MBEM + MTRN + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow11 + 4 + MCOL + MBEM + MTRN + MIWProperties).EntireRow.Hidden = False
    Rows(OriginRow11 + 5 + MCOL + MBEM + MTRN + MIWProperties).EntireRow.Hidden = False
  End If
End If
```

```
'Expanding or shortening the 4th table
If MIW > MIWProperties Then
   For i = 1 To (MIW - MIWProperties)
      Rows(4 + OriginRow11 + MCOL + MBEM + MTRN + MIWProperties + i).EntireRow.Insert
      For j = 1 To 8
         Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + MIWProperties + i, OriginCol11 - 1 +
j).Borders.Color = RGB(0, 0, 0)
         Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + MIWProperties + i, OriginCol11 - 1 +
j).Locked = False
      Next
      Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + MIWProperties + i, OriginCol11) =
MIWProperties + i
      Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + MIWProperties + i,
OriginCol11).HorizontalAlignment = xlCenter
      Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + MIWProperties + i, OriginCol11).Locked =
True
   Next
ElseIf MIW < MIWProperties Then
   Range(Rows(4 + OriginRow11 + MCOL + MBEM + MTRN + MIW), Rows(3 + OriginRow11 +
MCOL + MBEM + MTRN + MIWProperties)).Select
   Selection.Delete
End If
MIWProperties = MIW
Cells(4, 26) = MIWProperties
Worksheets(4).Protect ("555")

End Sub
```

Sheet5 (Element Connectivities)

```
Private Sub Worksheet_Activate()

OriginRow12 = 17
OriginCol12 = 14
OriginRow13 = 4
OriginCol13 = 2
OriginRow14 = 11
OriginCol14 = 2
OriginRow15 = 18
OriginCol15 = 2
OriginRow16 = 25
OriginCol16 = 2

NCOLProperties = Cells(1, 26)
NBEMProperties = Cells(2, 26)
NTRNProperties = Cells(3, 26)
NIWProperties = Cells(4, 26)
Worksheets(5).Unprotect ("555")
If NSO > NFR Then
   X = ((NSO - 1 - (NSO - 1) Mod 5) / 5) * 2
Else
   X = ((NFR - 1 - (NFR - 1) Mod 5) / 5) * 2
```

```
End If
NCOL = Worksheets(1).Cells(OriginRow12 + 3 + X, OriginCol12 + 9)
'Hiding or unhiding the 1st table
If NCOL <> NCOLProperties Then
  If NCOL = 0 Then
    For i = 1 To 5
      Rows(OriginRow13 - 2 + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow13 + 4 + NCOLProperties).EntireRow.Hidden = True
    Rows(OriginRow13 + 5 + NCOLProperties).EntireRow.Hidden = True
  ElseIf NCOLProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow13 - 2 + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow13 + 4 + NCOLProperties).EntireRow.Hidden = False
    Rows(OriginRow13 + 5 + NCOLProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 1st table
If NCOL > NCOLProperties Then
  For i = 1 To (NCOL - NCOLProperties)
    Rows(4 + OriginRow13 + NCOLProperties + i).EntireRow.Insert
    For j = 1 To 6
      Cells(3 + OriginRow13 + NCOLProperties + i, OriginCol13 - 1 + j).Borders.Color = RGB(0, 0, 0)
      Cells(3 + OriginRow13 + NCOLProperties + i, OriginCol13 - 1 + j).Locked = False
    Next
    Cells(3 + OriginRow13 + NCOLProperties + i, OriginCol13) = NCOLProperties + i
    Cells(3 + OriginRow13 + NCOLProperties + i, OriginCol13).HorizontalAlignment = xlCenter
    Cells(3 + OriginRow13 + NCOLProperties + i, OriginCol13).Locked = True
  Next
ElseIf NCOL < NCOLProperties Then
  Range(Rows(4 + OriginRow13 + NCOL), Rows(3 + OriginRow13 + NCOLProperties)).Select
  Selection.Delete
End If
NCOLProperties = NCOL
Cells(1, 26) = NCOLProperties
NBEM = Worksheets(1).Cells(OriginRow12 + 5 + X, OriginCol12 + 9)
'Hiding or unhiding the 2nd table
If NBEM <> NBEMProperties Then
  If NBEM = 0 Then
    For i = 1 To 5
      Rows(OriginRow14 - 2 + NCOL + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow14 + 4 + NCOL + NBEMProperties).EntireRow.Hidden = True
    Rows(OriginRow14 + 5 + NCOL + NBEMProperties).EntireRow.Hidden = True
  ElseIf NBEMProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow14 - 2 + NCOL + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow14 + 4 + NCOL + NBEMProperties).EntireRow.Hidden = False
    Rows(OriginRow14 + 5 + NCOL + NBEMProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 2nd table
If NBEM > NBEMProperties Then
  For i = 1 To (NBEM - NBEMProperties)
```

```
      Rows(4 + OriginRow14 + NCOL + NBEMProperties + i).EntireRow.Insert
      For j = 1 To 6
        Cells(3 + OriginRow14 + NCOL + NBEMProperties + i, OriginCol14 - 1 + j).Borders.Color =
RGB(0, 0, 0)
        Cells(3 + OriginRow14 + NCOL + NBEMProperties + i, OriginCol14 - 1 + j).Locked = False
      Next
      Cells(3 + OriginRow14 + NCOL + NBEMProperties + i, OriginCol14) = NBEMProperties + i
      Cells(3 + OriginRow14 + NCOL + NBEMProperties + i, OriginCol14).HorizontalAlignment =
xlCenter
      Cells(3 + OriginRow14 + NCOL + NBEMProperties + i, OriginCol14).Locked = True
    Next
ElseIf NBEM < NBEMProperties Then
    Range(Rows(4 + OriginRow14 + NCOL + NBEM), Rows(3 + OriginRow14 + NCOL +
NBEMProperties)).Select
    Selection.Delete
End If
NBEMProperties = NBEM
Cells(2, 26) = NBEMProperties
NTRN = Worksheets(1).Cells(OriginRow12 + 7 + X, OriginCol12 + 9)
'Hiding or unhiding the 3rd table
If NTRN <> NTRNProperties Then
    If NTRN = 0 Then
      For i = 1 To 5
        Rows(OriginRow15 - 2 + NCOL + NBEM + i).EntireRow.Hidden = True
      Next
      Rows(OriginRow15 + 4 + NCOL + NBEM + NTRNProperties).EntireRow.Hidden = True
      Rows(OriginRow15 + 5 + NCOL + NBEM + NTRNProperties).EntireRow.Hidden = True
    ElseIf NTRNProperties = 0 Then
      For i = 1 To 5
        Rows(OriginRow15 - 2 + NCOL + NBEM + i).EntireRow.Hidden = False
      Next
      Rows(OriginRow15 + 4 + NCOL + NBEM + NTRNProperties).EntireRow.Hidden = False
      Rows(OriginRow15 + 5 + NCOL + NBEM + NTRNProperties).EntireRow.Hidden = False
    End If
End If
'Expanding or shortening the 3rd table
If NTRN > NTRNProperties Then
    For i = 1 To (NTRN - NTRNProperties)
      Rows(4 + OriginRow15 + NCOL + NBEM + NTRNProperties + i).EntireRow.Insert
      For j = 1 To 7
        Cells(3 + OriginRow15 + NCOL + NBEM + NTRNProperties + i, OriginCol15 - 1 +
j).Borders.Color = RGB(0, 0, 0)
        Cells(3 + OriginRow15 + NCOL + NBEM + NTRNProperties + i, OriginCol15 - 1 + j).Locked =
False
      Next
      Cells(3 + OriginRow15 + NCOL + NBEM + NTRNProperties + i, OriginCol15) = NTRNProperties +
i
      Cells(3 + OriginRow15 + NCOL + NBEM + NTRNProperties + i, OriginCol15).HorizontalAlignment
= xlCenter
      Cells(3 + OriginRow15 + NCOL + NBEM + NTRNProperties + i, OriginCol15).Locked = True
    Next
ElseIf NTRN < NTRNProperties Then
    Range(Rows(4 + OriginRow15 + NCOL + NBEM + NTRN), Rows(3 + OriginRow15 + NCOL +
NBEM + NTRNProperties)).Select
    Selection.Delete
End If
```

```
NTRNProperties = NTRN
Cells(3, 26) = NTRNProperties
NIW = Worksheets(1).Cells(OriginRow12 + 9 + X, OriginCol12 + 9)
'Hiding or unhiding the 4th table
If NIW <> NIWProperties Then
  If NIW = 0 Then
    For i = 1 To 5
      Rows(OriginRow16 - 2 + NCOL + NBEM + NTRN + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow16 + 4 + NCOL + NBEM + NTRN + NIWProperties).EntireRow.Hidden = True
    Rows(OriginRow16 + 5 + NCOL + NBEM + NTRN + NIWProperties).EntireRow.Hidden = True
  ElseIf NIWProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow16 - 2 + NCOL + NBEM + NTRN + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow16 + 4 + NCOL + NBEM + NTRN + NIWProperties).EntireRow.Hidden = False
    Rows(OriginRow16 + 5 + NCOL + NBEM + NTRN + NIWProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 4th table
If NIW > NIWProperties Then
  For i = 1 To (NIW - NIWProperties)
    Rows(4 + OriginRow16 + NCOL + NBEM + NTRN + NIWProperties + i).EntireRow.Insert
    For j = 1 To 8
      Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + NIWProperties + i, OriginCol16 - 1 +
j).Borders.Color = RGB(0, 0, 0)
      Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + NIWProperties + i, OriginCol16 - 1 +
j).Locked = False
    Next
    Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + NIWProperties + i, OriginCol16) =
NIWProperties + i
    Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + NIWProperties + i,
OriginCol16).HorizontalAlignment = xlCenter
    Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + NIWProperties + i, OriginCol16).Locked =
True
  Next
ElseIf NIW < NIWProperties Then
  Range(Rows(4 + OriginRow16 + NCOL + NBEM + NTRN + NIW), Rows(3 + OriginRow16 + NCOL
+ NBEM + NTRN + NIWProperties)).Select
  Selection.Delete
End If
NIWProperties = NIW
Cells(4, 26) = NIWProperties
Worksheets(5).Protect ("555")

End Sub
```

Sheet6 (Static Loads)

Private Sub Worksheet_Activate()

```
OriginRow17 = 41
```

```
OriginCol17 = 2
OriginRow18 = 4
OriginCol18 = 2
OriginRow19 = 11
OriginCol19 = 2
OriginRow20 = 18
OriginCol20 = 2
OriginRow21 = 25
OriginCol21 = 2

NLUProperties = Cells(1, 26)
NLJProperties = Cells(2, 26)
NLMProperties = Cells(3, 26)
NLCProperties = Cells(4, 26)
Worksheets(6).Unprotect ("555")
If NSO > NFR Then
  X = ((NSO - 1 - (NSO - 1) Mod 5) / 5) * 2
Else
  X = ((NFR - 1 - (NFR - 1) Mod 5) / 5) * 2
End If
NLU = Worksheets(1).Cells(OriginRow17 + 3 + X, OriginCol17 + 9)
'Hiding or unhiding the 1st table
If NLU <> NLUProperties Then
  If NLU = 0 Then
    For i = 1 To 5
      Rows(OriginRow18 - 2 + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow18 + 4 + NLUProperties).EntireRow.Hidden = True
    Rows(OriginRow18 + 5 + NLUProperties).EntireRow.Hidden = True
  ElseIf NLUProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow18 - 2 + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow18 + 4 + NLUProperties).EntireRow.Hidden = False
    Rows(OriginRow18 + 5 + NLUProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 1st table
If NLU > NLUProperties Then
  For i = 1 To (NLU - NLUProperties)
    Rows(4 + OriginRow18 + NLUProperties + i).EntireRow.Insert
    For j = 1 To 3
      Cells(3 + OriginRow18 + NLUProperties + i, OriginCol18 - 1 + j).Borders.Color = RGB(0, 0, 0)
      Cells(3 + OriginRow18 + NLUProperties + i, OriginCol18 - 1 + j).Locked = False
    Next
    Cells(3 + OriginRow18 + NLUProperties + i, OriginCol18) = NLUProperties + i
    Cells(3 + OriginRow18 + NLUProperties + i, OriginCol18).HorizontalAlignment = xlCenter
    Cells(3 + OriginRow18 + NLUProperties + i, OriginCol18).Locked = True
  Next
ElseIf NLU < NLUProperties Then
  Range(Rows(4 + OriginRow18 + NLU), Rows(3 + OriginRow18 + NLUProperties)).Select
  Selection.Delete
End If
NLUProperties = NLU
Cells(1, 26) = NLUProperties
NLJ = Worksheets(1).Cells(OriginRow17 + 5 + X, OriginCol17 + 9)
```

```
'Hiding or unhiding the 2nd table
If NLJ <> NLJProperties Then
  If NLJ = 0 Then
    For i = 1 To 5
      Rows(OriginRow19 - 2 + NLU + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow19 + 4 + NLU + NLJProperties).EntireRow.Hidden = True
    Rows(OriginRow19 + 5 + NLU + NLJProperties).EntireRow.Hidden = True
  ElseIf NLJProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow19 - 2 + NLU + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow19 + 4 + NLU + NLJProperties).EntireRow.Hidden = False
    Rows(OriginRow19 + 5 + NLU + NLJProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 2nd table
If NLJ > NLJProperties Then
  For i = 1 To (NLJ - NLJProperties)
    Rows(4 + OriginRow19 + NLU + NLJProperties + i).EntireRow.Insert
    For j = 1 To 4
      Cells(3 + OriginRow19 + NLU + NLJProperties + i, OriginCol19 - 1 + j).Borders.Color = RGB(0,
0, 0)
      Cells(3 + OriginRow19 + NLU + NLJProperties + i, OriginCol19 - 1 + j).Locked = False
    Next
    Cells(3 + OriginRow19 + NLU + NLJProperties + i, OriginCol19) = NLJProperties + i
    Cells(3 + OriginRow19 + NLU + NLJProperties + i, OriginCol19).HorizontalAlignment = xlCenter
    Cells(3 + OriginRow19 + NLU + NLJProperties + i, OriginCol19).Locked = True
  Next
ElseIf NLJ < NLJProperties Then
  Range(Rows(4 + OriginRow19 + NLU + NLJ), Rows(3 + OriginRow19 + NLU +
NLJProperties)).Select
  Selection.Delete
End If
NLJProperties = NLJ
Cells(2, 26) = NLJProperties
NLM = Worksheets(1).Cells(OriginRow17 + 5 + X, OriginCol17 + 21)
'Hiding or unhiding the 3rd table
If NLM <> NLMProperties Then
  If NLM = 0 Then
    For i = 1 To 5
      Rows(OriginRow20 - 2 + NLU + NLJ + i).EntireRow.Hidden = True
    Next
    Rows(OriginRow20 + 4 + NLU + NLJ + NLMProperties).EntireRow.Hidden = True
    Rows(OriginRow20 + 5 + NLU + NLJ + NLMProperties).EntireRow.Hidden = True
  ElseIf NLMProperties = 0 Then
    For i = 1 To 5
      Rows(OriginRow20 - 2 + NLU + NLJ + i).EntireRow.Hidden = False
    Next
    Rows(OriginRow20 + 4 + NLU + NLJ + NLMProperties).EntireRow.Hidden = False
    Rows(OriginRow20 + 5 + NLU + NLJ + NLMProperties).EntireRow.Hidden = False
  End If
End If
'Expanding or shortening the 3rd table
If NLM > NLMProperties Then
  For i = 1 To (NLM - NLMProperties)
```

```
        Rows(4 + OriginRow20 + NLU + NLJ + NLMProperties + i).EntireRow.Insert
        For j = 1 To 4
            Cells(3 + OriginRow20 + NLU + NLJ + NLMProperties + i, OriginCol20 - 1 + j).Borders.Color =
RGB(0, 0, 0)
            Cells(3 + OriginRow20 + NLU + NLJ + NLMProperties + i, OriginCol20 - 1 + j).Locked = False
        Next
        Cells(3 + OriginRow20 + NLU + NLJ + NLMProperties + i, OriginCol20) = NLMProperties + i
        Cells(3 + OriginRow20 + NLU + NLJ + NLMProperties + i, OriginCol20).HorizontalAlignment =
xlCenter
        Cells(3 + OriginRow20 + NLU + NLJ + NLMProperties + i, OriginCol20):Locked = True
    Next
ElseIf NLM < NLMProperties Then
    Range(Rows(4 + OriginRow20 + NLU + NLJ + NLM), Rows(3 + OriginRow20 + NLU + NLJ +
NLMProperties)).Select
    Selection.Delete
End If
NLMProperties = NLM
Cells(3, 26) = NLMProperties
NLC = Worksheets(1).Cells(OriginRow17 + 3 + X, OriginCol17 + 21)
'Hiding or unhiding the 4th table
If NLC <> NLCProperties Then
    If NLC = 0 Then
        For i = 1 To 5
            Rows(OriginRow21 - 2 + NLU + NLJ + NLM + i).EntireRow.Hidden = True
        Next
        Rows(OriginRow21 + 4 + NLU + NLJ + NLM + NLCProperties).EntireRow.Hidden = True
        Rows(OriginRow21 + 5 + NLU + NLJ + NLM + NLCProperties).EntireRow.Hidden = True
    ElseIf NLCProperties = 0 Then
        For i = 1 To 5
            Rows(OriginRow21 - 2 + NLU + NLJ + NLM + i).EntireRow.Hidden = False
        Next
        Rows(OriginRow21 + 4 + NLU + NLJ + NLM + NLCProperties).EntireRow.Hidden = False
        Rows(OriginRow21 + 5 + NLU + NLJ + NLM + NLCProperties).EntireRow.Hidden = False
    End If
End If
'Expanding or shortening the 4th table
If NLC > NLCProperties Then
    For i = 1 To (NLC - NLCProperties)
        Rows(4 + OriginRow21 + NLU + NLJ + NLM + NLCProperties + i).EntireRow.Insert
        For j = 1 To 5
            Cells(3 + OriginRow21 + NLU + NLJ + NLM + NLCProperties + i, OriginCol21 - 1 +
j).Borders.Color = RGB(0, 0, 0)
            Cells(3 + OriginRow21 + NLU + NLJ + NLM + NLCProperties + i, OriginCol21 - 1 + j).Locked =
False
        Next
        Cells(3 + OriginRow21 + NLU + NLJ + NLM + NLCProperties + i, OriginCol21) = NLCProperties +
i
        Cells(3 + OriginRow21 + NLU + NLJ + NLM + NLCProperties + i,
OriginCol21).HorizontalAlignment = xlCenter
        Cells(3 + OriginRow21 + NLU + NLJ + NLM + NLCProperties + i, OriginCol21).Locked = True
    Next
ElseIf NLC < NLCProperties Then
    Range(Rows(4 + OriginRow21 + NLU + NLJ + NLM + NLC), Rows(3 + OriginRow21 + NLU + NLJ +
NLM + NLCProperties)).Select
    Selection.Delete
End If
```

```
NLCProperties = NLC
Cells(4, 26) = NLCProperties
Worksheets(6).Protect ("555")

End Sub
```

Sheet7 (Analysis)

```
Private Sub CheckBox1_Click()

If refresh = False Then
   Unprotect ("555")
   Cells(5, 37) = CheckBox1.Value
   If CheckBox1.Value = False Then
      CheckBox1.ForeColor = RGB(128, 128, 128)
   Else
      CheckBox1.ForeColor = RGB(0, 0, 128)
   End If
   Protect ("555")
End If

End Sub

Private Sub ComboBox1_Change()

If refresh = False Then
Unprotect ("555")
If ComboBox1.Value = "Force Control" Then
   If Cells(1, 30) = 4 Then
      Rows(OriginRow22 + 5).Hidden = False
      Rows(OriginRow22 + 6).Hidden = False
      For i = 1 To 9 + 3 * Cells(1, 28)
         Rows(OriginRow22 + 26 + i).Hidden = False
      Next
   Else
      For i = 1 To 10
         Rows(4 + i + OriginRow22).Hidden = False
      Next
      For i = 1 To 9 + 3 * Cells(1, 28)
         Rows(OriginRow22 + 26 + i).Hidden = True
      Next
   End If
   If Cells(1, 30) = 3 Then
      For i = 1 To 8
         Rows(OriginRow22 + i + 14).Hidden = False
      Next
   ElseIf Cells(1, 30) = 2 Then
      For i = 1 To 4
         Rows(OriginRow22 + i + 22).Hidden = False
      Next
   End If
   ComboBox2.Top = Rows(5 + OriginRow22).Top
```

```
    ComboBox2.Left = Columns(6).Left
    ComboBox2.Width = 140
    ComboBox2.Height = 18
    Cells(1, 39) = ComboBox1.ListIndex
    For i = 1 To (1 + Cells(1, 28))
      Cells(31 + OriginRow22 + 3 * i, OriginCol22) = "Force"
    Next
  ElseIf ComboBox1.Value = "Displacement Control" Then
    If Cells(1, 30) = 4 Then
      Rows(OriginRow22 + 5).Hidden = True
      Rows(OriginRow22 + 6).Hidden = True
    Else
      For i = 1 To 10
        Rows(4 + i + OriginRow22).Hidden = True
      Next
    End If
    If Cells(1, 30) = 3 Then
      For i = 1 To 8
        Rows(OriginRow22 + i + 14).Hidden = True
      Next
    ElseIf Cells(1, 30) = 2 Then
      For i = 1 To 4
        Rows(OriginRow22 + i + 22).Hidden = True
      Next
    End If
    For i = 1 To 9 + 3 * Cells(1, 28)
      Rows(OriginRow22 + 26 + i).Hidden = False
    Next
    ComboBox2.Top = Rows(5 + OriginRow22).Top
    ComboBox2.Left = Columns(28).Left
    ComboBox2.Width = 140
    ComboBox2.Height = 18
    Cells(1, 39) = ComboBox1.ListIndex
    For i = 1 To (1 + Cells(1, 28))
      Cells(31 + OriginRow22 + 3 * i, OriginCol22) = "Displacement"
    Next
  End If
  Protect ("555")
  End If

End Sub

Private Sub ComboBox2_Change()

If refresh = False Then
If ComboBox2.ListIndex <> Cells(1, 30) Then
  Unprotect ("555")
  If Cells(1, 30) = 3 Then
    For i = 1 To 8
      Rows(OriginRow22 + i + 14).Hidden = True
    Next
  ElseIf Cells(1, 30) = 2 Then
    For i = 1 To 4
      Rows(OriginRow22 + i + 22).Hidden = True
    Next
  ElseIf Cells(1, 30) = 4 Then
```

```
    For i = 1 To 8
      Rows(6 + i + OriginRow22).Hidden = False
    Next
    For i = 1 To 9 + 3 * Cells(1, 28)
      Rows(OriginRow22 + 26 + i).Hidden = True
    Next
    ComboBox2.Top = Rows(5 + OriginRow22).Top
    ComboBox2.Left = Columns(6).Left
    ComboBox2.Width = 140
    ComboBox2.Height = 18
  End If
  If ComboBox2.ListIndex = 3 Then
    For i = 1 To 8
      Rows(OriginRow22 + i + 14).Hidden = False
    Next
  ElseIf ComboBox2.ListIndex = 2 Then
    For i = 1 To 4
      Rows(OriginRow22 + i + 22).Hidden = False
    Next
  ElseIf ComboBox2.ListIndex = 4 Then
    For i = 1 To 8
      Rows(6 + i + OriginRow22).Hidden = True
    Next
    For i = 1 To 9 + 3 * Cells(1, 28)
      Rows(OriginRow22 + 26 + i).Hidden = False
    Next
    For i = 1 To (1 + Cells(1, 28))
      Cells(31 + OriginRow22 + 3 * i, OriginCol22) = "Force"
    Next
    ComboBox2.Top = Rows(5 + OriginRow22).Top
    ComboBox2.Left = Columns(6).Left
    ComboBox2.Width = 140
    ComboBox2.Height = 18
  End If
  Cells(1, 30) = ComboBox2.ListIndex
  Protect ("555")
End If
End If

End Sub

Private Sub ComboBox3_Change()

If refresh = False Then
  Unprotect ("555")
  Cells(2, 37) = ComboBox3.ListIndex
  Protect ("555")
End If

End Sub

Private Sub ComboBox4_Change()

If refresh = False Then
  Dim CyclicAnalysisOption As String
  Dim StepNumber As String
```

```
  Unprotect ("555")
  Cells(3, 37) = ComboBox4.ListIndex
  NumberOfHistoryPoints = Cells(5, 33)
  oo1 = Cells(4, 28)
  If ComboBox4.Value = "Force Control" Then
    CyclicAnalysisOption = "Force at step "
  Else
    CyclicAnalysisOption = "Displacement at step "
  End If
  For i = 1 To (1 + oo1)
    For j = 1 To NumberOfHistoryPoints
      StepNumber = NumberOfHistoryPoints + 1 - j
      Cells(69 + OriginRow22 + 3 * Cells(1, 28) + (2 + NumberOfHistoryPoints) * (oo1 + 2 - i) - j,
OriginCol22) = CyclicAnalysisOption + StepNumber
    Next
  Next
  Protect ("555")
End If
```

End Sub

Private Sub CommandButton1_Click()

```
If Cells(3, 33) = 1 Then
  UserForm1.OptionButton1.Value = True
ElseIf Cells(3, 33) = 2 Then
  UserForm1.OptionButton2.Value = True
ElseIf Cells(3, 33) = 3 Then
  UserForm1.OptionButton3.Value = True
Else
  UserForm1.OptionButton4.Value = True
End If
If NSO > NFR Then
  X = ((NSO - 1 - (NSO - 1) Mod 5) / 5) * 2
Else
  X = ((NFR - 1 - (NFR - 1) Mod 5) / 5) * 2
End If
NLU = Worksheets(1).Cells(OriginRow17 + 3 + X, OriginCol17 + 9)
NLJ = Worksheets(1).Cells(OriginRow17 + 5 + X, OriginCol17 + 9)
NLM = Worksheets(1).Cells(OriginRow17 + 5 + X, OriginCol17 + 21)
NLC = Worksheets(1).Cells(OriginRow17 + 3 + X, OriginCol17 + 21)
If NLC = 0 And NLM = 0 And NLJ = 0 And NLU = 0 Then
  UserForm1.TextBox1.Enabled = False
  UserForm1.TextBox2.Enabled = False
Else
  UserForm1.TextBox1.Enabled = True
  UserForm1.TextBox2.Enabled = True
End If
UserForm1.TextBox1.Value = Cells(5, 25)
UserForm1.TextBox2.Value = Cells(6, 26)
UserForm1.Show
```

End Sub

Private Sub CommandButton3_Click()

```
Start

End Sub

Private Sub Worksheet_Change(ByVal Target As Excel.Range)

If Not Application.Intersect(Target, Cells(27 + OriginRow22, 5 + OriginCol22)) Is Nothing Then
   Unprotect ("555")
   NLDED = Cells(27 + OriginRow22, 5 + OriginCol22)
   NLDEDAnalysis = Cells(1, 26)
   oo1 = (NLDEDAnalysis - 1 - (NLDEDAnalysis - 1) Mod 10) / 10
""Inserting or deleting rows
   If (NLDED - 1 - (NLDED - 1) Mod 10) / 10 > oo1 Then
      For i = 1 To (NLDED - 1 - (NLDED - 1) Mod 10) / 10 - oo1
         For j = 1 To 3
            Rows(36 + OriginRow22 + 3 * oo1).EntireRow.Insert
         Next
      Next
      For i = 1 To (NLDED - 1 - (NLDED - 1) Mod 10) / 10 - oo1
         Range(Cells(33 + OriginRow22, OriginCol22), Cells(34 + OriginRow22, OriginCol22 + 2)).Select
         Selection.Copy
         Cells(33 + OriginRow22 + 3 * (i + oo1), OriginCol22).Select
         Selection.PasteSpecial
      Next
   ElseIf (NLDED - 1 - (NLDED - 1) Mod 10) / 10 < oo1 Then
      Range(Rows(36 + OriginRow22 + 3 * (NLDED - 1 - (NLDED - 1) Mod 10) / 10), Rows(35 +
OriginRow22 + 3 * (NLDED - 1 - (NLDED - 1) Mod 10) / 10 + 3 * (oo1 - (NLDED - 1 - (NLDED - 1)
Mod 10) / 10))).Select
      Selection.Delete
   End If
""Expanding push-over table
   If NLDED > NLDEDAnalysis Then
      oo1 = (NLDEDAnalysis - NLDEDAnalysis Mod 10) / 10
      For i = 1 To (NLDED - NLDEDAnalysis)
         Cells(33 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDEDAnalysis + i - 10 * oo1).Locked =
False
         Cells(34 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDEDAnalysis + i - 10 * oo1).Locked =
False
         Cells(33 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDEDAnalysis + i - 10 *
oo1).Borders.Color = RGB(0, 0, 0)
         Cells(34 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDEDAnalysis + i - 10 *
oo1).Borders.Color = RGB(0, 0, 0)
         oo1 = (NLDEDAnalysis + i - (NLDEDAnalysis + i) Mod 10) / 10
      Next
""Shortening push-over table
   ElseIf NLDED < NLDEDAnalysis Then
      oo1 = (NLDED - 1 - (NLDED - 1) Mod 10) / 10
      For i = 1 To (oo1 + 1) * 10 - NLDED
         Cells(33 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDED + i - 10 * oo1).Borders.Color =
RGB(255, 255, 255)
         Cells(34 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDED + i - 10 * oo1).Borders.Color =
RGB(255, 255, 255)
         Cells(33 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDED + i - 10 * oo1) = Empty
         Cells(34 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDED + i - 10 * oo1) = Empty
         Cells(33 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDED + i - 10 * oo1).Locked = True
         Cells(34 + OriginRow22 + oo1 * 3, OriginCol22 + 2 + NLDED + i - 10 * oo1).Locked = True
```

```
      Next
    End If
    NLDEDAnalysis = NLDED
    Cells(1, 26) = NLDEDAnalysis
    Cells(1, 28) = (NLDED - 1 - (NLDED - 1) Mod 10) / 10
    Protect ("555")
End If

''''Quasi Static table

If Not Application.Intersect(Target, Cells(64 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)) Is
Nothing Then
    Unprotect ("555")
    NLDED2 = Cells(64 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    NLDED2Analysis = Cells(2, 30)
    oo1 = (NLDED2Analysis - 1 - (NLDED2Analysis - 1) Mod 10) / 10
''''Inserting or deleting rows
    If (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10 > oo1 Then
      For i = 1 To (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10 - oo1
        For j = 1 To (2 + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22))
          Rows(70 + OriginRow22 + 3 * Cells(1, 28) + (2 + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5
+ OriginCol22)) * (oo1 + 1)).EntireRow.Insert
        Next
      Next
      For i = 1 To (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10 - oo1
          Range(Cells(70 + OriginRow22 + 3 * Cells(1, 28), OriginCol22), Cells(70 + OriginRow22 + 3 *
Cells(1, 28) + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22), OriginCol22 + 3)).Select
        Selection.Copy
        Cells(70 + OriginRow22 + 3 * Cells(1, 28) + (2 + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 +
OriginCol22)) * (i + oo1), OriginCol22).Select
        Selection.PasteSpecial
      Next
    ElseIf (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10 < oo1 Then
      Range(Rows(70 + OriginRow22 + 3 * Cells(1, 28) + (2 + Cells(66 + OriginRow22 + 3 * Cells(1, 28),
5 + OriginCol22)) * ((NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10 + 1)), Rows(69 + OriginRow22 + 3 *
Cells(1, 28) + (2 + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)) * ((NLDED2 - 1 -
(NLDED2 - 1) Mod 10) / 10 + 1) + (2 + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)) *
(oo1 - (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10))).Select
      Selection.Delete
    End If
''''Expanding push-over table
    If NLDED2 > NLDED2Analysis Then
      oo1 = (NLDED2Analysis - NLDED2Analysis Mod 10) / 10
      For i = 1 To (NLDED2 - NLDED2Analysis)
        For j = 1 To (1 + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22))
          Cells(69 + j + OriginRow22 + 3 * Cells(1, 28) + oo1 * (2 + Cells(66 + OriginRow22 + 3 *
Cells(1, 28), 5 + OriginCol22)), OriginCol22 + 3 + NLDED2Analysis + i - 10 * oo1).Locked = False
          Cells(69 + j + OriginRow22 + 3 * Cells(1, 28) + oo1 * (2 + Cells(66 + OriginRow22 + 3 *
Cells(1, 28), 5 + OriginCol22)), OriginCol22 + 3 + NLDED2Analysis + i - 10 * oo1).Borders.Color =
RGB(0, 0, 0)
        Next
        oo1 = (NLDED2Analysis + i - (NLDED2Analysis + i) Mod 10) / 10
      Next
''''Shortening push-over table
    ElseIf NLDED2 < NLDED2Analysis Then
      oo1 = (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10
```

```
      For i = 1 To (oo1 + 1) * 10 - NLDED2
        For j = 1 To (1 + Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22))
          Cells(69 + j + OriginRow22 + 3 * Cells(1, 28) + oo1 * (2 + Cells(66 + OriginRow22 + 3 *
Cells(1, 28), 5 + OriginCol22)), OriginCol22 + 3 + NLDED2 + i - 10 * oo1).Borders.Color = RGB(255,
255, 255)
          Cells(69 + j + OriginRow22 + 3 * Cells(1, 28) + oo1 * (2 + Cells(66 + OriginRow22 + 3 *
Cells(1, 28), 5 + OriginCol22)), OriginCol22 + 3 + NLDED2 + i - 10 * oo1) = Empty
          Cells(69 + j + OriginRow22 + 3 * Cells(1, 28) + oo1 * (2 + Cells(66 + OriginRow22 + 3 *
Cells(1, 28), 5 + OriginCol22)), OriginCol22 + 3 + NLDED2 + i - 10 * oo1).Locked = True
        Next
      Next
    End If
    NLDED2Analysis = NLDED2
    Cells(2, 30) = NLDED2Analysis
    Cells(4, 28) = (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10
    Protect ("555")
End If

If Not Application.Intersect(Target, Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)) Is
Nothing Then
    Unprotect ("555")
    Dim CyclicAnalysisOption As String
    Dim StepNumber As String
    If ComboBox4.Value = "Force Control" Then
      CyclicAnalysisOption = "Force at step "
    Else
      CyclicAnalysisOption = "Displacement at step "
    End If
    NLDED2 = Cells(64 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    NumberOfHistoryPoints = Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    NumberOfHistoryPointsAnalysis = Cells(5, 33)
    oo1 = (NLDED2 - 1 - (NLDED2 - 1) Mod 10) / 10
""Inserting or deleting rows
    If NumberOfHistoryPoints > NumberOfHistoryPointsAnalysis Then
      For i = 1 To (1 + oo1)
        For j = 1 To (NumberOfHistoryPoints - NumberOfHistoryPointsAnalysis)
          Rows(69 + OriginRow22 + 3 * Cells(1, 28) + (2 + NumberOfHistoryPointsAnalysis) * (oo1 + 2 -
i)).EntireRow.Insert
          Range(Cells(70 + OriginRow22 + 3 * Cells(1, 28) + (2 + NumberOfHistoryPointsAnalysis) *
(oo1 + 1 - i), OriginCol22), Cells(70 + OriginRow22 + 3 * Cells(1, 28) + (2 +
NumberOfHistoryPointsAnalysis) * (oo1 + 1 - i), OriginCol22 + 13)).Select

          Selection.Copy
          Cells(69 + OriginRow22 + 3 * Cells(1, 28) + (2 + NumberOfHistoryPointsAnalysis) * (oo1 + 2 -
i), OriginCol22).Select
          Selection.PasteSpecial
          StepNumber = NumberOfHistoryPoints + 1 - j
          Cells(69 + OriginRow22 + 3 * Cells(1, 28) + (2 + NumberOfHistoryPointsAnalysis) * (oo1 + 2 -
i), OriginCol22) = CyclicAnalysisOption + StepNumber
        Next
      Next
    ElseIf NumberOfHistoryPoints < NumberOfHistoryPointsAnalysis Then
      For i = 1 To (1 + oo1)
        Range(Rows(69 + OriginRow22 + 3 * Cells(1, 28) + (2 + NumberOfHistoryPointsAnalysis) * (oo1
+ 2 - i) - (NumberOfHistoryPointsAnalysis - NumberOfHistoryPoints)), Rows(68 + OriginRow22 + 3 *
```

```
Cells(1, 28) + (2 + NumberOfHistoryPointsAnalysis) * (oo1 + 2 - i) - (NumberOfHistoryPointsAnalysis -
NumberOfHistoryPoints) + (NumberOfHistoryPointsAnalysis - NumberOfHistoryPoints))).Select
      Selection.Delete
    Next
  End If
  NumberOfHistoryPointsAnalysis = NumberOfHistoryPoints
  Cells(5, 33) = NumberOfHistoryPointsAnalysis
  Protect ("555")
End If
```

End Sub

Private Sub CommandButton2_Click()

```
If NSO > NFR Then
  X = ((NSO - 1 - (NSO - 1) Mod 5) / 5) * 2
Else
  X = ((NFR - 1 - (NFR - 1) Mod 5) / 5) * 2
End If
NCON = Worksheets(1).Cells(OriginRow6 + 3 + X, OriginCol6 + 11)
NSTL = Worksheets(1).Cells(OriginRow6 + 5 + X, OriginCol6 + 11)
NMSR = Worksheets(1).Cells(OriginRow6 + 7 + X, OriginCol6 + 11)
NHYS = Worksheets(1).Cells(OriginRow6 + 9 + X, OriginCol6 + 11)
NPDEL = Worksheets(1).Cells(1, 43)
IFLEX = Worksheets(1).Cells(1, 42)
IFLEXDIST = Worksheets(1).Cells(1, 41)
MCOL = Worksheets(1).Cells(OriginRow3 + 3 + X, OriginCol3 + 9)
MBEM = Worksheets(1).Cells(OriginRow3 + 5 + X, OriginCol3 + 9)
MTRN = Worksheets(1).Cells(OriginRow3 + 7 + X, OriginCol3 + 9)
MIW = Worksheets(1).Cells(OriginRow3 + 9 + X, OriginCol3 + 9)
NCOL = Worksheets(1).Cells(OriginRow12 + 3 + X, OriginCol12 + 9)
NBEM = Worksheets(1).Cells(OriginRow12 + 5 + X, OriginCol12 + 9)
NTRN = Worksheets(1).Cells(OriginRow12 + 7 + X, OriginCol12 + 9)
NIW = Worksheets(1).Cells(OriginRow12 + 9 + X, OriginCol12 + 9)
NLU = Worksheets(1).Cells(OriginRow17 + 3 + X, OriginCol17 + 9)
NLJ = Worksheets(1).Cells(OriginRow17 + 5 + X, OriginCol17 + 9)
NLM = Worksheets(1).Cells(OriginRow17 + 5 + X, OriginCol17 + 21)
NLC = Worksheets(1).Cells(OriginRow17 + 3 + X, OriginCol17 + 21)
IU = Worksheets(1).Cells(4, 43)
ReDim HIGT(NSO)
oo1 = 0
For i = 1 To NSO
  HIGT(i) = Worksheets(1).Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 - 1 + i - 5 * oo1)
  oo1 = (i - i Mod 5) / 5
Next
ReDim NDUP(NFR)
oo1 = 0
For i = 1 To NFR
  NDUP(i) = Worksheets(1).Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 8 + i - 5 * oo1)
  oo1 = (i - i Mod 5) / 5
Next
ReDim NVLN(NFR)
oo1 = 0
For i = 1 To NFR
  NVLN(i) = Worksheets(1).Cells(8 + OriginRow1 + 2 * oo1, OriginCol1 + 17 + i - 5 * oo1)
  oo1 = (i - i Mod 5) / 5
```

```
Next
Dim FullFileName As String
Dim FileName As String
Dim SFileName As String
Dim PathName As String
Dim posn As Integer
FullFileName = Application.GetSaveAsFilename("", "IDARC Input File (*.dat),*.dat")
If FullFileName = "False" Then
   Exit Sub
End If
posn = 0
'find the position of the last "\" character in filename
For i = 1 To Len(FullFileName)
   If (Mid(FullFileName, i, 1) = "\") Then posn = i
Next
'get filename without path
FileName = Right(FullFileName, Len(FullFileName) - posn)
'get path name
PathName = Left(FullFileName, posn)
'get filename without extension
posn = InStr(FileName, ".")
If posn <> 0 Then
   SFileName = Left(FileName, posn - 1)
End If
Open PathName + "idarc.dat" For Output As #1
Print #1, FileName
Print #1, SFileName + ".out"
Close #1
Open FullFileName For Output As #1
Print #1, "kaf"
Print #1, "CONTROL DATA"
Print #1, NSO; NFR; NCON; NSTL; NMSR; NPDEL; IFLEX; IFLEXDIST; 1
Print #1, "ELEMENT TYPES"
Print #1, MCOL; MBEM; 0; 0; MTRN; 0; 0; 0; 0; MIW
Print #1, "ELEMENT DATA"
Print #1, NCOL; NBEM; 0; 0; NTRN; 0; 0; 0; NIW
Print #1, "UNITS SYSTEM"
Print #1, IU
Print #1, "FLOOR ELEVATIONS"
Dim elevation As Double
elevation = 0
For i = 1 To NSO
   elevation = elevation + HIGT(i)
   Print #1, elevation;
Next
Print #1,
Print #1, "IDENTICAL FRAMES"
For i = 1 To NFR
   Print #1, NDUP(i);
Next
Print #1,
Print #1, "PLAN CONFIGURATION"
For i = 1 To NFR
   Print #1, NVLN(i);
Next
Print #1,
```

```
Print #1, "NODAL WEIGHTS"
oo2 = 0
For i = 1 To NSO
   Print #1, i;
   For j = 1 To NFR
      If j > 1 Then
         Print #1, "   ";
      End If
      Print #1, j;
      If j > 1 Then
         If (j - 1) Mod 10 = 0 Then
            oo2 = oo2 + 1
            oo3 = oo3 + 1
         End If
      End If
      For k = 1 To NVLN(j)
         Print #1, Worksheets(2).Cells(OriginRow2 + 5 + i + (4 + NSO) * oo2, OriginCol2 + k - 10 * oo3);
      Next
      Print #1,
      oo2 = oo2 + 1
      oo3 = 0
   Next
   oo2 = 0
Next
Print #1, "CODE FOR SPECIFICATION OF USER PROPERTIES"
Print #1, 0
If NCON > 0 Then
   Print #1, "CONCRETE PROPERTIES"
   For i = 1 To NCON
      IM = Worksheets(3).Cells(5 + OriginRow7 + i, OriginCol7)
      FC = Worksheets(3).Cells(5 + OriginRow7 + i, OriginCol7 + 1)
      EC = Worksheets(3).Cells(5 + OriginRow7 + i, OriginCol7 + 2)
      EPS0 = Worksheets(3).Cells(5 + OriginRow7 + i, OriginCol7 + 3)
      FT = Worksheets(3).Cells(5 + OriginRow7 + i, OriginCol7 + 4)
      EPSU = Worksheets(3).Cells(5 + OriginRow7 + i, OriginCol7 + 5)
      ZF = Worksheets(3).Cells(5 + OriginRow7 + i, OriginCol7 + 6)
      Print #1, IM; FC; EC; EPS0; FT; EPSU; ZF
   Next
End If
If NSTL > 0 Then
   Print #1, "REINFORCEMENT PROPERTIES"
   For i = 1 To NSTL
      IM = Worksheets(3).Cells(5 + OriginRow8 + NCON + i, OriginCol8)
      FS = Worksheets(3).Cells(5 + OriginRow8 + NCON + i, OriginCol8 + 1)
      FSU = Worksheets(3).Cells(5 + OriginRow8 + NCON + i, OriginCol8 + 2)
      ES = Worksheets(3).Cells(5 + OriginRow8 + NCON + i, OriginCol8 + 3)
      ESH = Worksheets(3).Cells(5 + OriginRow8 + NCON + i, OriginCol8 + 4)
      EPSH = Worksheets(3).Cells(5 + OriginRow8 + NCON + i, OriginCol8 + 5)
      Print #1, IM; FS; FSU; ES; ESH; EPSH
   Next
End If
If NMSR > 0 Then
   Print #1, "MASONRY INFILL PROPERTIES"
   For i = 1 To NMSR
      IM = Worksheets(3).Cells(5 + OriginRow9 + NCON + NSTL + i, OriginCol9)
      FM = Worksheets(3).Cells(5 + OriginRow9 + NCON + NSTL + i, OriginCol9 + 1)
```

```
        FMCR = Worksheets(3).Cells(5 + OriginRow9 + NCON + NSTL + i, OriginCol9 + 2)
        EPSM = Worksheets(3).Cells(5 + OriginRow9 + NCON + NSTL + i, OriginCol9 + 3)
        VM = Worksheets(3).Cells(5 + OriginRow9 + NCON + NSTL + i, OriginCol9 + 4)
        SIGMM = Worksheets(3).Cells(5 + OriginRow9 + NCON + NSTL + i, OriginCol9 + 5)
        CFM = Worksheets(3).Cells(5 + OriginRow9 + NCON + NSTL + i, OriginCol9 + 6)
        Print #1, IM; FM; FMCR; EPSM; VM; SIGMM; CFM
    Next
End If
Print #1, "HYSTERETIC MODELING RULES"
Print #1, NHYS
For i = 1 To NHYS
    IR = Worksheets(3).Cells(5 + OriginRow23 + NCON + NSTL + NMSR + i, OriginCol23)
    HC = Worksheets(3).Cells(5 + OriginRow23 + NCON + NSTL + NMSR + i, OriginCol23 + 1)
    HBD = Worksheets(3).Cells(5 + OriginRow23 + NCON + NSTL + NMSR + i, OriginCol23 + 2)
    HBE = Worksheets(3).Cells(5 + OriginRow23 + NCON + NSTL + NMSR + i, OriginCol23 + 3)
    HS = Worksheets(3).Cells(5 + OriginRow23 + NCON + NSTL + NMSR + i, OriginCol23 + 4)
    IBILINEAR = Worksheets(3).Cells(5 + OriginRow23 + NCON + NSTL + NMSR + i, OriginCol23 + 5)
    Print #1, IR; 1; HC; HBD; HBE; HS; IBILINEAR
Next
If NCOL <> 0 And MCOL <> 0 Then
    Print #1, "COLUMN PROPERTIES"
    Print #1, 0
    Print #1, "COLUMN DATA"
    For i = 1 To MCOL
        KC = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4)
        KHYSC = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 1)
        IMC = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 2)
        AMLC = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 3)
        RAMC1 = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 4)
        RAMC2 = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 5)
        B = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 6)
        D = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 7)
        IMS = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 8)
        AT = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 9)
        DC = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 10)
        HBD = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 11)
        HBS = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 12)
        CEF = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 13)
        AN = Worksheets(4).Cells(5 + OriginRow4 + i, OriginCol4 + 14)
        Print #1, 1
        Print #1, KC; IMC; IMS; AN; AMLC; RAMC1; RAMC2
        Print #1, -KHYSC; D; B; DC; AT; HBD; HBS; CEF
    Next
End If
If NBEM <> 0 And MBEM <> 0 Then
    Print #1, "BEAM PROPERTIES"
    Print #1, 0
    Print #1, "BEAM DATA"
    For i = 1 To MBEM
        KB = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5)
        KHYSB = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 1)
        IMC = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 2)
        AMLB = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 3)
        RAMB1 = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 4)
        RAMB2 = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 5)
        B = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 6)
```

```
        D = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 7)
        BSL = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 8)
        TSL = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 9)
        IMS = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 10)
        AT1 = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 11)
        AT2 = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 12)
        BC = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 13)
        HBD = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 14)
        HBS = Worksheets(4).Cells(5 + OriginRow5 + MCOL + i, OriginCol5 + 15)
        Print #1, 1
        Print #1, KB; IMC; IMS; AMLB; RAMB1; RAMB2
        Print #1, -KHYSB; D; B; BSL; TSL; BC; AT1; AT2; HBD; HBS
    Next
End If
If NTRN <> 0 And MTRN <> 0 Then
    Print #1, "TRANSVERSE BEAM PROPERTIES"
    Print #1, 0
    Print #1, "TRANSVERSE BEAM DATA"
    For i = 1 To MTRN
        KT = Worksheets(4).Cells(3 + OriginRow10 + MCOL + MBEM + i, OriginCol10)
        AKV = Worksheets(4).Cells(3 + OriginRow10 + MCOL + MBEM + i, OriginCol10 + 1)
        ARV = Worksheets(4).Cells(3 + OriginRow10 + MCOL + MBEM + i, OriginCol10 + 2)
        ALV = Worksheets(4).Cells(3 + OriginRow10 + MCOL + MBEM + i, OriginCol10 + 3)
        Print #1, KT; AKV; ARV; ALV
    Next
End If
If NIW <> 0 And MIW <> 0 Then
    Print #1, "INFILL PANEL PROPERTIES"
    Print #1, 0
    Print #1, "INFILL PANEL DATA"
    For i = 1 To MIW
        IPT = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11)
        IMT = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11 + 1)
        TMP = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11 + 2)
        VLMP = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11 + 3)
        VHMP = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11 + 4)
        QMPC = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11 + 5)
        QMPB = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11 + 6)
        QMPJ = Worksheets(4).Cells(3 + OriginRow11 + MCOL + MBEM + MTRN + i, OriginCol11 + 7)
        Print #1, IPT; 0
        Print #1, IMT; TMP; VLMP; VHMP
        Print #1, QMPC; QMPB; QMPJ; QMPC
        Print #1, 1; 0.1; 0.9; 2; 0.01
        Print #1, 1; 0.3; 0.1; 0
        Print #1, 0.1; 0.8; 1; 5
    Next
End If
If NCOL <> 0 And MCOL <> 0 Then
    Print #1, "COLUMN CONNECTIVITIES"
    For i = 1 To NCOL
        M = Worksheets(5).Cells(3 + OriginRow13 + i, OriginCol13)
        ITC = Worksheets(5).Cells(3 + OriginRow13 + i, OriginCol13 + 1)
        IC = Worksheets(5).Cells(3 + OriginRow13 + i, OriginCol13 + 2)
        JC = Worksheets(5).Cells(3 + OriginRow13 + i, OriginCol13 + 3)
        LBC = Worksheets(5).Cells(3 + OriginRow13 + i, OriginCol13 + 4)
        LTC = Worksheets(5).Cells(3 + OriginRow13 + i, OriginCol13 + 5)
```

```
    Print #1, M; ITC; IC; JC; LBC; LTC
  Next
End If
If NBEM <> 0 And MBEM <> 0 Then
  Print #1, "BEAM CONNECTIVITIES"
  For i = 1 To NBEM
    M = Worksheets(5).Cells(3 + OriginRow14 + NCOL + i, OriginCol14)
    ITB = Worksheets(5).Cells(3 + OriginRow14 + NCOL + i, OriginCol14 + 1)
    LB = Worksheets(5).Cells(3 + OriginRow14 + NCOL + i, OriginCol14 + 2)
    IB = Worksheets(5).Cells(3 + OriginRow14 + NCOL + i, OriginCol14 + 3)
    JLB = Worksheets(5).Cells(3 + OriginRow14 + NCOL + i, OriginCol14 + 4)
    JRB = Worksheets(5).Cells(3 + OriginRow14 + NCOL + i, OriginCol14 + 5)
    Print #1, M; ITB; LB; IB; JLB; JRB
  Next
End If
If NTRN <> 0 And MTRN <> 0 Then
  Print #1, "TRANSVERSE BEAM CONNECTIVITIES"
  For i = 1 To NTRN
    M = Worksheets(5).Cells(3 + OriginRow15 + NCOL + NBEM + i, OriginCol15)
    ITT = Worksheets(5).Cells(3 + OriginRow15 + NCOL + NBEM + i, OriginCol15 + 1)
    LT = Worksheets(5).Cells(3 + OriginRow15 + NCOL + NBEM + i, OriginCol15 + 2)
    IWT = Worksheets(5).Cells(3 + OriginRow15 + NCOL + NBEM + i, OriginCol15 + 3)
    JWT = Worksheets(5).Cells(3 + OriginRow15 + NCOL + NBEM + i, OriginCol15 + 4)
    IFT = Worksheets(5).Cells(3 + OriginRow15 + NCOL + NBEM + i, OriginCol15 + 5)
    JFT = Worksheets(5).Cells(3 + OriginRow15 + NCOL + NBEM + i, OriginCol15 + 6)
    Print #1, M; ITT; LT; IWT; JWT; IFT; JFT
  Next
End If
If NIW <> 0 And MIW <> 0 Then
  Print #1, "INFILL PANEL CONNECTIVITIES"
  For i = 1 To NIW
    M = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16)
    IFnum = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16 + 1)
    ITIW = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16 + 2)
    LTop = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16 + 3)
    LBot = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16 + 4)
    JL = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16 + 5)
    JR = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16 + 6)
    JBMT = Worksheets(5).Cells(3 + OriginRow16 + NCOL + NBEM + NTRN + i, OriginCol16 + 7)
    Print #1, M; IFnum; ITIW; LTop; LBot; JL; JR; JBMT
  Next
End If
Print #1, "ANALYSIS OPTION"
Print #1, Cells(3, 33)
Print #1, "STATIC LOADS"
Print #1, NLU; NLJ; NLM; NLC
If NLU <> 0 Or NLJ <> 0 Or NLM <> 0 Or NLC <> 0 Then
  JSTP = Cells(5, 25)
  IOCRL = Cells(6, 26)
  Print #1, JSTP; IOCRL
End If
If NLU <> 0 Then
  Print #1, "UNIFORMLY LOADED BEAMS DATA"
  For i = 1 To NLU
    IL = Worksheets(6).Cells(3 + OriginRow18 + i, OriginCol18)
    IBN = Worksheets(6).Cells(3 + OriginRow18 + i, OriginCol18 + 1)
```

```
      FU = Worksheets(6).Cells(3 + OriginRow18 + i, OriginCol18 + 2)
      Print #1, IL; IBN; FU
    Next
End If
If NLJ <> 0 Then
    Print #1, "LATERALLY LOADED JOINTS DATA"
    For i = 1 To NLJ
      IL = Worksheets(6).Cells(3 + OriginRow19 + NLU + i, OriginCol19)
      LF = Worksheets(6).Cells(3 + OriginRow19 + NLU + i, OriginCol19 + 1)
      IFnum = Worksheets(6).Cells(3 + OriginRow19 + NLU + i, OriginCol19 + 2)
      FL = Worksheets(6).Cells(3 + OriginRow19 + NLU + i, OriginCol19 + 3)
      Print #1, IL; LF; IFnum; FL
    Next
End If
If NLM <> 0 Then
    Print #1, "NODAL MOMENT DATA"
    For i = 1 To NLM
      IL = Worksheets(6).Cells(3 + OriginRow20 + NLU + NLJ + i, OriginCol20)
      IBM = Worksheets(6).Cells(3 + OriginRow20 + NLU + NLJ + i, OriginCol20 + 1)
      FM1 = Worksheets(6).Cells(3 + OriginRow20 + NLU + NLJ + i, OriginCol20 + 2)
      FM2 = Worksheets(6).Cells(3 + OriginRow20 + NLU + NLJ + i, OriginCol20 + 3)
      Print #1, IL; IBM; FM1; FM2
    Next
End If
If NLC <> 0 Then
    Print #1, "CONCENTRATED VERTICAL LOADS DATA"
    For i = 1 To NLC
      IL = Worksheets(6).Cells(3 + OriginRow21 + NLU + NLJ + NLM + i, OriginCol21)
      IFV = Worksheets(6).Cells(3 + OriginRow21 + NLU + NLJ + NLM + i, OriginCol21 + 2)
      LV = Worksheets(6).Cells(3 + OriginRow21 + NLU + NLJ + NLM + i, OriginCol21 + 1)
      JV = Worksheets(6).Cells(3 + OriginRow21 + NLU + NLJ + NLM + i, OriginCol21 + 3)
      FV = Worksheets(6).Cells(3 + OriginRow21 + NLU + NLJ + NLM + i, OriginCol21 + 4)
      Print #1, IL; IFV; LV; JV; FV
    Next
End If
If Cells(3, 33) = 2 Then
    Print #1, "MONOTONIC PUSH-OVER ANALYSIS"
    Print #1, Cells(1, 39) + 1
    If Cells(1, 39) + 1 = 1 Then
      Print #1, "FORCE CONTROLLED INPUT"
      If Cells(1, 30) = 0 Then
        ITYP = 2
      ElseIf Cells(1, 30) = 1 Then
        ITYP = 1
      ElseIf Cells(1, 30) = 2 Then
        ITYP = 5
      ElseIf Cells(1, 30) = 3 Then
        ITYP = 3
      Else
        ITYP = 4
      End If
      Print #1, ITYP
      If ITYP <> 4 Then
        PMAX = Cells(9 + OriginRow22, 9 + OriginCol22)
        MSTEPS = Cells(11 + OriginRow22, 9 + OriginCol22)
        DRFILM = Cells(13 + OriginRow22, 9 + OriginCol22)
```

```
            Print #1, PMAX; MSTEPS; DRFILM
        End If
        If ITYP = 3 Then
            NMOD = Cells(17 + OriginRow22, 9 + OriginCol22)
            POWER1 = Cells(19 + OriginRow22, 5 + OriginCol22)
            POWER2 = Cells(21 + OriginRow22, 5 + OriginCol22)
            Print #1, NMOD; POWER1; POWER2
        End If
        If ITYP = 5 Then
            Print #1, Cells(25 + OriginRow22, 5 + OriginCol22)
        End If
        If ITYP = 4 Then
            Print #1, "USER DEFINED"
        End If
    End If
    If Cells(1, 39) + 1 = 2 Then
        Print #1, "DISPLACEMENT CONTROL INPUT"
    End If
    If Cells(1, 39) + 1 = 2 Or ITYP = 4 Then
        NLDED = Cells(27 + OriginRow22, 5 + OriginCol22)
        MSTEPS = Cells(29 + OriginRow22, 5 + OriginCol22)
        DRFILM = Cells(31 + OriginRow22, 9 + OriginCol22)
        Print #1, NLDED
        ReDim NSTLD(NLDED)
        ReDim PX(NLDED)
        oo1 = 0
        For i = 1 To NLDED
            NSTLD(i) = Cells(33 + OriginRow22 + 3 * oo1, 2 + OriginCol22 + i - 10 * oo1)
            PX(i) = Cells(34 + OriginRow22 + 3 * oo1, 2 + OriginCol22 + i - 10 * oo1)
            oo1 = (i - i Mod 10) / 10
            Print #1, NSTLD(i);
        Next
        Print #1,
        For i = 1 To NLDED
            Print #1, PX(i);
        Next
        Print #1,
        Print #1, MSTEPS; DRFILM
    End If
ElseIf Cells(3, 33) = 3 Then
    Print #1, "DYNAMIC ANALYSIS"
    GMAXH = Cells(47 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    GMAXV = Cells(47 + OriginRow22 + 3 * Cells(1, 28), 13 + OriginCol22)
    DTCAL = Cells(45 + OriginRow22 + 3 * Cells(1, 28), 13 + OriginCol22)
    TDUR = Cells(45 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    DAMP = Cells(43 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    If Cells(2, 37) = 0 Then
        ITDMP = 1
    ElseIf Cells(2, 37) = 1 Then
        ITDMP = 2
    ElseIf Cells(2, 37) = 2 Then
        ITDMP = 3
    End If
    If CheckBox1.Value = True Then
        IWV = 1
    Else
```

```
      IWV = 0
    End If
    NDATA = Cells(49 + OriginRow22 + 3 * Cells(1, 28), 13 + OriginCol22)
    DTINP = Cells(49 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    NAMEW = Cells(51 + OriginRow22 + 3 * Cells(1, 28), 2 + OriginCol22)
    WHFILE = Cells(53 + OriginRow22 + 3 * Cells(1, 28), 2 + OriginCol22)
    WVFILE = Cells(55 + OriginRow22 + 3 * Cells(1, 28), 2 + OriginCol22)
    Print #1, GMAXH; GMAXV; DTCAL; TDUR; DAMP; ITDMP
    Print #1, "INPUT WAVE"
    Print #1, IWV; NDATA; DTINP
    Print #1, NAMEW
    Print #1, WHFILE
    If IWV = 1 Then
      Print #1, WVFILE
    End If
  ElseIf Cells(3, 33) = 4 Then
    Print #1, "QUASI-STATIC CYCLIC ANALYSIS"
    NLDED = Cells(64 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    NPTS = Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22)
    DTCAL = Cells(68 + OriginRow22 + 3 * Cells(1, 28), 7 + OriginCol22)
    ReDim NSTLD(NLDED)
    ReDim F(NPTS, NLDED)
    For i = 1 To NLDED
      NSTLD(i) = Cells(70 + OriginRow22 + 3 * Cells(1, 28) + (2 + NPTS) * oo1, 3 + OriginCol22 + i - 10
  * oo1)
      For j = 1 To NPTS
        F(j, i) = Cells(70 + OriginRow22 + j + 3 * Cells(1, 28) + (2 + NPTS) * oo1, 3 + OriginCol22 + i -
  10 * oo1)
      Next
      oo1 = (i - i Mod 10) / 10
    Next
    Print #1, Cells(3, 37)
    Print #1, NLDED
    For i = 1 To NLDED
      Print #1, NSTLD(i);
    Next
    Print #1,
    Print #1, NPTS
    For i = 1 To NLDED
      For j = 1 To NPTS
        Print #1, F(j, i);
      Next
      Print #1,
    Next
    Print #1, DTCAL
  End If
  If Cells(3, 33) = 2 Then
    Print #1, "PUSH-OVER ANALYSIS SNAPSHOTS"
    Print #1, 0
  ElseIf Cells(3, 33) = 3 Then
    Print #1, "DYNAMIC ANALYSIS SNAPSHOTS"
    Print #1, 0
  ElseIf Cells(3, 33) = 4 Then
    Print #1, "QUASI-STATIC ANALYSIS SNAPSHOTS"
    Print #1, 0
  End If
```

```
Print #1, 0; 0; 0; 0; 0
Print #1, "STORY OUTPUT CONTROL"
Print #1, 0; 0.01 'NSO has been changed to 0 for this particular case and a semicolon has been deleted from
the end
'For i = 1 To NSO
'   Print #1, (NSO + 1 - i);
'Next
'Print #1,
'Dim number As String
'For i = 1 To NSO
'   number = (NSO + 1 - i)
'   Print #1, "st" + number + ".out"
'Next
Print #1, "ELEMENT HYSTERESIS OUTPUT"
Print #1, 0; 0; 0; 0; 0; 0
Close #1
```

End Sub

User Form (Analysis Options)

Private Sub CommandButton1_Click()

```
Worksheets(7).Unprotect ("555")
If OptionButton1.Value = True Then
  If Worksheets(7).Cells(3, 33) = 2 Then
    Worksheets(7).ComboBox1.Top = Worksheets(7).Rows(3 + OriginRow22).Top
    Worksheets(7).ComboBox1.Left = Worksheets(7).Columns(28).Left
    Worksheets(7).ComboBox1.Width = 140
    Worksheets(7).ComboBox1.Height = 18
    Worksheets(7).ComboBox2.Top = Worksheets(7).Rows(5 + OriginRow22).Top
    Worksheets(7).ComboBox2.Left = Worksheets(7).Columns(28).Left
    Worksheets(7).ComboBox2.Width = 140
    Worksheets(7).ComboBox2.Height = 18
    For i = 1 To (38 + 3 * Worksheets(7).Cells(1, 28))
      Worksheets(7).Rows(OriginRow22 + i - 2).Hidden = True
    Next
  ElseIf Worksheets(7).Cells(3, 33) = 3 Then
    Worksheets(7).ComboBox3.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
    Worksheets(7).ComboBox3.Left = Worksheets(7).Columns(28).Left
    Worksheets(7).ComboBox3.Width = 120
    Worksheets(7).ComboBox3.Height = 18
    Worksheets(7).CheckBox1.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
    Worksheets(7).CheckBox1.Left = Worksheets(7).Columns(28).Left
    Worksheets(7).CheckBox1.Width = 140
    Worksheets(7).CheckBox1.Height = 18
    For i = 1 To 21
      Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 36).Hidden = True
    Next
  ElseIf Worksheets(7).Cells(3, 33) = 4 Then
```

```
        Worksheets(7).ComboBox4.Top = Worksheets(7).Rows(62 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).ComboBox4.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox4.Width = 140
        Worksheets(7).ComboBox4.Height = 18
        For i = 1 To (13 + (2 + Worksheets(7).Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22))
* (Worksheets(7).Cells(4, 28) + 1))
            Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 57).Hidden = True
        Next
    End If
    Worksheets(7).Cells(3, 33) = 1
ElseIf OptionButton2.Value = True Then
    If Worksheets(7).Cells(3, 33) = 3 Then
        Worksheets(7).ComboBox3.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).ComboBox3.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox3.Width = 120
        Worksheets(7).ComboBox3.Height = 18
        Worksheets(7).CheckBox1.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).CheckBox1.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).CheckBox1.Width = 140
        Worksheets(7).CheckBox1.Height = 18
        For i = 1 To 21
            Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 36).Hidden = True
        Next
    ElseIf Worksheets(7).Cells(3, 33) = 4 Then
        Worksheets(7).ComboBox4.Top = Worksheets(7).Rows(62 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).ComboBox4.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox4.Width = 140
        Worksheets(7).ComboBox4.Height = 18
        For i = 1 To (13 + (2 + Worksheets(7).Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22))
* (Worksheets(7).Cells(4, 28) + 1))
            Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 57).Hidden = True
        Next
    End If
    If Worksheets(7).Cells(3, 33) <> 2 Then
        For i = 1 To 6
            Worksheets(7).Rows(OriginRow22 + i - 2).Hidden = False
        Next
        Worksheets(7).ComboBox1.Top = Worksheets(7).Rows(3 + OriginRow22).Top
        Worksheets(7).ComboBox1.Left = Worksheets(7).Columns(6).Left
        Worksheets(7).ComboBox1.Width = 140
        Worksheets(7).ComboBox1.Height = 18
        If Worksheets(7).ComboBox1.Value = "Force Control" Then
            If Worksheets(7).Cells(1, 30) = 4 Then
                Worksheets(7).Rows(OriginRow22 + 5).Hidden = False
                Worksheets(7).Rows(OriginRow22 + 6).Hidden = False
                For i = 1 To 9 + 3 * Worksheets(7).Cells(1, 28)
                    Worksheets(7).Rows(OriginRow22 + 26 + i).Hidden = False
                Next
            Else
                For i = 1 To 10
                    Worksheets(7).Rows(4 + i + OriginRow22).Hidden = False
                Next
```

```
            End If
            If Worksheets(7).Cells(1, 30) = 3 Then
                For i = 1 To 8
                    Worksheets(7).Rows(OriginRow22 + i + 14).Hidden = False
                Next
            ElseIf Worksheets(7).Cells(1, 30) = 2 Then
                For i = 1 To 4
                    Worksheets(7).Rows(OriginRow22 + i + 22).Hidden = False
                Next
            End If
            Worksheets(7).ComboBox2.Top = Worksheets(7).Rows(5 + OriginRow22).Top
            Worksheets(7).ComboBox2.Left = Worksheets(7).Columns(6).Left
            Worksheets(7).ComboBox2.Width = 140
            Worksheets(7).ComboBox2.Height = 18
        ElseIf Worksheets(7).ComboBox1.Value = "Displacement Control" Then
            For i = 1 To 9 + 3 * Worksheets(7).Cells(1, 28)
                Worksheets(7).Rows(OriginRow22 + 26 + i).Hidden = False
            Next
        End If
        Worksheets(7).Rows(OriginRow22 + 36 + 3 * Worksheets(7).Cells(1, 28)).Hidden = False
    End If
    Worksheets(7).Cells(3, 33) = 2
ElseIf OptionButton3.Value = True Then
    If Worksheets(7).Cells(3, 33) = 4 Then
        Worksheets(7).ComboBox4.Top = Worksheets(7).Rows(62 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).ComboBox4.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox4.Width = 140
        Worksheets(7).ComboBox4.Height = 18
        For i = 1 To (13 + (2 + Worksheets(7).Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22))
* (Worksheets(7).Cells(4, 28) + 1))
            Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 57).Hidden = True
        Next
    ElseIf Worksheets(7).Cells(3, 33) = 2 Then
        Worksheets(7).ComboBox1.Top = Worksheets(7).Rows(3 + OriginRow22).Top
        Worksheets(7).ComboBox1.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox1.Width = 140
        Worksheets(7).ComboBox1.Height = 18
        Worksheets(7).ComboBox2.Top = Worksheets(7).Rows(5 + OriginRow22).Top
        Worksheets(7).ComboBox2.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox2.Width = 140
        Worksheets(7).ComboBox2.Height = 18
        For i = 1 To (38 + 3 * Worksheets(7).Cells(1, 28))
            Worksheets(7).Rows(OriginRow22 + i - 2).Hidden = True
        Next
    End If
    If Worksheets(7).Cells(3, 33) <> 3 Then
        For i = 1 To 21
            Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 36).Hidden = False
        Next
        Worksheets(7).ComboBox3.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).ComboBox3.Left = Worksheets(7).Columns(5).Left
        Worksheets(7).ComboBox3.Width = 120
        Worksheets(7).ComboBox3.Height = 18
```

```
        Worksheets(7).CheckBox1.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).CheckBox1.Left = Worksheets(7).Columns(10).Left
        Worksheets(7).CheckBox1.Width = 140
        Worksheets(7).CheckBox1.Height = 18
    End If
    Worksheets(7).Cells(3, 33) = 3
ElseIf OptionButton4.Value = True Then
    If Worksheets(7).Cells(3, 33) = 2 Then
        Worksheets(7).ComboBox1.Top = Worksheets(7).Rows(3 + OriginRow22).Top
        Worksheets(7).ComboBox1.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox1.Width = 140
        Worksheets(7).ComboBox1.Height = 18
        Worksheets(7).ComboBox2.Top = Worksheets(7).Rows(5 + OriginRow22).Top
        Worksheets(7).ComboBox2.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox2.Width = 140
        Worksheets(7).ComboBox2.Height = 18
        For i = 1 To (38 + 3 * Worksheets(7).Cells(1, 28))
            Worksheets(7).Rows(OriginRow22 + i - 2).Hidden = True
        Next
    ElseIf Worksheets(7).Cells(3, 33) = 3 Then
        Worksheets(7).ComboBox3.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).ComboBox3.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).ComboBox3.Width = 120
        Worksheets(7).ComboBox3.Height = 18
        Worksheets(7).CheckBox1.Top = Worksheets(7).Rows(41 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).CheckBox1.Left = Worksheets(7).Columns(28).Left
        Worksheets(7).CheckBox1.Width = 140
        Worksheets(7).CheckBox1.Height = 18
        For i = 1 To 21
            Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 36).Hidden = True
        Next
    End If
    If Worksheets(7).Cells(3, 33) <> 4 Then
        For i = 1 To (13 + (2 + Worksheets(7).Cells(66 + OriginRow22 + 3 * Cells(1, 28), 5 + OriginCol22))
* (Worksheets(7).Cells(4, 28) + 1))
            Worksheets(7).Rows(OriginRow22 + 3 * Worksheets(7).Cells(1, 28) + i + 57).Hidden = False
        Next
        Worksheets(7).ComboBox4.Top = Worksheets(7).Rows(62 + OriginRow22 + 3 *
Worksheets(7).Cells(1, 28)).Top
        Worksheets(7).ComboBox4.Left = Worksheets(7).Columns(7).Left
        Worksheets(7).ComboBox4.Width = 140
        Worksheets(7).ComboBox4.Height = 18
    End If
    Worksheets(7).Cells(3, 33) = 4
End If
Worksheets(7).Cells(5, 25) = UserForm1.TextBox1.Value
Worksheets(7).Cells(6, 26) = UserForm1.TextBox2.Value
Worksheets(7).Protect ("555")
UserForm1.Hide

End Sub

Private Sub CommandButton2_Click()
```

```
UserForm1.Hide

End Sub
```

APPENDIX B. MATLAB Code for the Post-Processor

```
function varargout = Seismix(varargin)
% SEISMIX M-file for Seismix.fig
%      SEISMIX, by itself, creates a new SEISMIX or raises the existing
%      singleton*.
%
%      H = SEISMIX returns the handle to a new SEISMIX or the handle to
%      the existing singleton*.
%
%      SEISMIX('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in SEISMIX.M with the given input arguments.
%
%      SEISMIX('Property','Value',...) creates a new SEISMIX or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Seismix_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Seismix_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Seismix

% Last Modified by GUIDE v2.5 03-Mar-2006 23:32:19

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Seismix_OpeningFcn, ...
                   'gui_OutputFcn',  @Seismix_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Seismix is made visible.
function Seismix_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Seismix (see VARARGIN)

clear all;
global FileName;

% UIWAIT makes Seismix wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Seismix_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global FileName;
global NSO
global NFR
global NModes;
global ModeShapes;
global ModeShapes2;
global Mode;
global Item;
global Story;
global Period;
if NSO > 8
   NModes = 8;
else
   NModes = NSO;
end
Index = find(strcmp(Item,'NORMALIZED)'));
Index1 = Index(2);
ModeShapes = zeros(NSO+1,NModes+1);
ModeShapes2 = zeros(NSO+1,NModes+1);
for i = 1:NSO
   for j = 1:(NModes+1)
      ModeShapes(i,j) = str2double(cell2mat(Item(Index1+2+NModes+(i-1)*(NModes+1)+j)));
   end
end
Index = find(strcmp(Item,'PARTICIPATION'));
Index1 = Index(1);
Period = zeros(NModes,1);
for i = 1:NModes
   Period(i,1) = str2double(cell2mat(Item(Index1+12+(i-1)*6)));
end
set(handles.text8, 'String', [' Fundamental Period = ' mat2str(Period(1,1)) ' s']);
Mode = 0;
ModeShapes2 = ModeShapes;
Story = zeros(NSO+1,1);
```

```
ModeShapes2(:,1) = [];
figure(1);
set(1,'Position',[660 260 560 420]);
plot(ModeShapes2,ModeShapes(:,1));
xlabel('Mass Normalized Displacement');
ylabel('Story');
grid on;


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global NSO;
global NModes;
global ModeShapes;
global Mode;
global Period;
if Mode < NModes
   Mode = Mode + 1;
else
   Mode = 1;
end
Z = zeros(NSO+1,2);
for i = 1:NSO
   Z(i,1) = ModeShapes(i,Mode+1);
end
if Mode == 1
   set(handles.text8, 'String', [' Fundamental Period = ' mat2str(Period(1,1)) ' s']);
else
   set(handles.text8, 'String', [' Period # ' mat2str(Mode) ' = ' mat2str(Period(Mode,1)) ' s']);
end
figure(1);
set(1,'Position',[660 260 560 420]);
plot(Z,ModeShapes(:,1));
xlabel('Mass Normalized Displacement');
ylabel('Story');
grid on;


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global NSO;
global NModes;
global ModeShapes;
global Mode;
global Period;
if Mode > 1
   Mode = Mode - 1;
else
   Mode = NModes;
```

```
end
Z = zeros(NSO+1,2);
for i = 1:NSO
   Z(i,1) = ModeShapes(i,Mode+1);
end
if Mode == 1
   set(handles.text8, 'String', [' Fundamental Period = ' mat2str(Period(1,1)) ' s']);
else
   set(handles.text8, 'String', [' Period # ' mat2str(Mode) ' = ' mat2str(Period(Mode,1)) ' s']);
end
figure(1);
set(1,'Position',[660 260 560 420]);
plot(Z,ModeShapes(:,1));
xlabel('Mass Normalized Displacement');
ylabel('Story');
grid on;


% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global FileName;
global Item;
Index = find(strcmp(Item,'VS.'));
Index1 = Index(1);
Index = find(strcmp(Item,'PLOT'));
Index2 = Index(1);
x = 0;
x = (Index2 - Index1 - 16) / 3;
BaseShear = zeros(x+1,1);
OverallDef = zeros(x+1,1);
for i = 1:x
   BaseShear(i+1,1) = str2double(cell2mat(Item(Index1+14+2+3*(i-1))));
   OverallDef(i+1,1) = str2double(cell2mat(Item(Index1+14+3+3*(i-1))));
end
figure(2);
set(2,'Position',[660 260 560 420]);
plot(OverallDef,BaseShear);
xlabel('Overall Deformation (% of Building Height)');
ylabel('Base Shear Coefficient (V/W)');
grid on;


% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global FileName;
global Item;
global Max;
Index1 = find(strcmp(Item,'COL-WALL'));
```

```
x = size(Item);
Index2 = x(1);
y = (Index2-Index1)/7;
Time = zeros(y+1,1);
Displacement = zeros(y+1,2);
Max = 0;
for i = 1:y
   Time(i+1,1) = str2double(cell2mat(Item(Index1+1+(i-1)*7)));
   Displacement(i+1,1) = str2double(cell2mat(Item(Index1+2+(i-1)*7)));
   if abs(Displacement(i+1,1)) > Max
      Max = abs(Displacement(i+1,1));
   end
end
set(handles.text8, 'String', [' Maximum Disp. = ' mat2str(Max) ' mm']);
figure(3);
set(3,'Position',[660 260 560 420]);
plot(Time,Displacement);
xlabel('Time (sec)');
ylabel('Displacement (mm)');
grid on;


% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global FileName;
global Item;
Index = find(strcmp(Item,'SHEAR'));
Index1 = Index(2);
x = size(Item);
Index2 = x(1);
y = (Index2-Index1-6)/5;
Drift = zeros(y+1,1);
StoryShear = zeros(y+1,2);
for i = 1:y
   Drift(i+1,1) = str2double(cell2mat(Item(Index1+8+(i-1)*5)));
   StoryShear(i+1,1) = str2double(cell2mat(Item(Index1+11+(i-1)*5)));
end
figure(4);
set(4,'Position',[660 260 560 420]);
plot(Drift,StoryShear);
xlabel('Inter-Story Drift (mm)');
ylabel('Story Shear (kN)');
grid on;


% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global FileName;
```

```
global Item;
Index1 = find(strcmp(Item,'COL-WALL'));
x = size(Item);
Index2 = x(1);
y = (Index2-Index1-3)/5;
Displacement = zeros(y+1,2);
StoryShear = zeros(y+1,1);
for i = 1:y
   Displacement(i+1,1) = str2double(cell2mat(Item(Index1+4+(i-1)*5)));
   StoryShear(i+1,1) = str2double(cell2mat(Item(Index1+6+(i-1)*5)));
end
figure(5);
set(5,'Position',[660 260 560 420]);
plot(Displacement,StoryShear);
xlabel('Story Displacement (mm)');
ylabel('Story Shear (kN)');
grid on;


% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global FileName;
global Item;
global Max;
Index1 = find(strcmp(Item,'COL-WALL'));
x = size(Item);
Index2 = x(1);
y = (Index2-Index1)/7;
Time = zeros(y+1,1);
Drift = zeros(y+1,2);
Max = 0;
for i = 1:y
   Time(i+1,1) = str2double(cell2mat(Item(Index1+1+(i-1)*7)));
   Drift(i+1,1) = str2double(cell2mat(Item(Index1+3+(i-1)*7)));
   if abs(Drift(i+1,1)) > Max
      Max = abs(Drift(i+1,1));
   end
end
set(handles.text8, 'String', [' Maximum Story-Drift = ' mat2str(Max) ' mm']);
figure(3);
set(3,'Position',[660 260 560 420]);
plot(Time,Drift);
xlabel('Time (sec)');
ylabel('Inter-Story Drift (mm)');
grid on;


% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global FileName
global Item
global NSO;
global NFR;
[FileName,PathName] = uigetfile('*.out','Read File');
set(handles.text6, 'String', FileName);
FileName = [PathName FileName];
Item = textread(FileName,'%s');
if strcmp(Item(1),'TIME') == 0
   Index = find(strcmp(Item,'STORIES'));
   Index1 = Index(1);
   NSO = str2double(cell2mat(Item(Index1+2)));
   NFR = str2double(cell2mat(Item(Index1+7)));
   set(handles.text5, 'String', ' General Output File.');
   set(handles.text7, 'String', [' Number of Stories = ' mat2str(NSO)]);
   set(handles.text8, 'String', '');
elseif strcmp(Item(17),'COL-WALL') == 1
   set(handles.text5, 'String', ' Story Time History Output File.');
   set(handles.text7, 'String', ' (Dynamic Analysis)...');
   set(handles.text8, 'String', '');
elseif strcmp(Item(18),'SHEAR') == 1
   set(handles.text5, 'String', ' Story History Output File.');
   set(handles.text7, 'String', ' (Push-Over Analysis)...');
   set(handles.text8, 'String', '');
elseif strcmp(Item(15),'COL-WALL') == 1
   set(handles.text5, 'String', ' Story History Output File.');
   set(handles.text7, 'String', ' (Quasi-Static Analysis)...');
   set(handles.text8, 'String', '');
end


% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global FileName;
global Item;
global NSO;
DriftRatio = zeros(NSO,1);
IDriftRatio = zeros(NSO,1);
Index1 = find(strcmp(Item,'RATIO(%)'));
for i = 1:NSO
   DriftRatio(i,1) = str2double(cell2mat(Item(Index1+13+(i-1)*8)));
end
assignin('base','IDriftRatio',DriftRatio);
open IDriftRatio;


% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global FileName;
global Item;
global NSO;
DriftRatio = zeros(NSO+1,1);
Story = zeros(NSO+1,1);
Index1 = find(strcmp(Item,'RATIO(%)'));
for i = 1:NSO
   DriftRatio(NSO-i+2,1) = str2double(cell2mat(Item(Index1+13+(i-1)*8)));
   Story(i+1,1) = i;
end
figure(6);
set(6,'Position',[660 260 560 420]);
plot(DriftRatio,Story);
xlabel('Inter-Story Drift Ratio (%)');
ylabel('Story');
grid on;


% % --- Executes on button press in pushbutton13.
% function pushbutton13_Callback(hObject, eventdata, handles)
% % hObject    handle to pushbutton13 (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% % handles    structure with handles and user data (see GUIDATA)
%
% global FileName;
% global Item;
% global FlexCol;
% Index = find(strcmp(Item,'COLUMNS'));
% Index1 = Index(3);
% NCOL = str2double(cell2mat(Item(Index1+2)));
% Index = find(strcmp(Item,'EI3N'));
% Index1 = Index(3);
% FlexCol = zeros(NCOL,6);
% for i = 1:NCOL
%    for j = 1:6
%       FlexCol(i,j) = str2double(cell2mat(Item(Index1+2+j+25*(i-1))));
%    end
% end
% Col = 1;
% PDef = zeros(4,2);
% PDef(2,1) = FlexCol(Col,2)/FlexCol(Col,1);
% PDef(3,1) = FlexCol(Col,4);
% PDef(4,1) = FlexCol(Col,5);
% PDef(2,2) = FlexCol(Col,2);
% PDef(3,2) = FlexCol(Col,3);
% PDef(4,2) = FlexCol(Col,3)+FlexCol(Col,6)*(FlexCol(Col,5)-FlexCol(Col,4));
% display(PDef);
% display(PDef(2,1));
% display(PDef(3,1));
% display(PDef(4,1));
% figure(7);
% set(7,'Position',[660 260 560 420]);
% plot(PDef(:,1),PDef(:,2));
% xlabel('Deformation');
% ylabel('Force');
```

```
% grid on;
```

APPENDIX C. Visual Basic Code for the Interaction Diagram

```
Private Sub Interaction()

Dim PM() As Double

For i = 1 To 200
   For j = 1 To 10
      Worksheets("Output Data").Cells(i + 2, j) = Empty
   Next
Next

n = 180
fc = [e10]
fy = [h10]
Ec = [e12]
Es = [h12]
alpha1 = [u8]
beta1 = [u10]
phic = [e18]
phiS = [e20]
epsC = [h18]
epsY = [h20]
b = [e24]
h = [h24]
Ast = [e28]
d = [e30]
dt = [h30]

ReDim PM(n, 2)

For i = 0 To (n - 1)
   c = h * (n - i) / n
   epsB = (d - c) / c * 0.0035
   epsT = (dt - c) / c * 0.0035
   Slope = c / 0.0035
   If epsB > phiS * fy / Es Then
      fs = phiS * fy
      B1 = phiS * fy / Es * Slope
      B2 = d - c - B1
   ElseIf epsB > 0 Then
      fs = epsB * Es
      B1 = d - c
      B2 = 0
   ElseIf epsB < -phiS * fy / Es Then
      fs = -phiS * fy
      B1 = phiS * fy / Es * Slope
      B2 = d - c - B1
   ElseIf epsB < 0 Then
      fs = epsB * Es
      B1 = d - c
      B2 = 0
   End If
```

```
    If epsT < -phiS * fy / Es Then
        fst = -phiS * fy
        T1 = phiS * fy / Es * Slope
        T2 = c - dt - T1
    ElseIf epsT < 0 Then
        fst = epsT * Es
        T1 = c - dt
        T2 = 0
    ElseIf epsT > phiS * fy / Es Then
        fst = phiS * fy
        T1 = phiS * fy / Es * Slope
        T2 = c - dt - T1
    ElseIf epsT > 0 Then
        fst = epsB * Es
        T1 = c - dt
        T2 = 0
    End If
    Pc = -phic * alpha1 * fc * beta1 * c * b
    Ps = Ast / 4 * (fst + fs + 2 / (d - dt) * (fs * (B1 / 2 + B2) + fst * (T1 / 2 + T2)))
    PM(i, 1) = Pc + Ps
    PM(i, 2) = -phic * alpha1 * fc * beta1 * c * b * (h / 2 - beta1 * c / 2) + Ast / 4 * (fst * (h / 2 - dt) - fs * (d
- h / 2) + 2 / (d - dt) * (-fs * (B1 * (Abs(B1)) / 3 + B2 * (Abs(B1) + Abs(B2) / 2)) + fst * (T1 * (Abs(T1)) /
3 + T2 * (Abs(T1) + Abs(T2) / 2)))) + Ast / 2 / (d - dt) * (fs * (B1 / 2 + B2) + fst * (T1 / 2 + T2)) * (h / 2 -
c)
    If PM(i, 2) > 0 Then
        GoTo Finito
    End If
    If PM(i, 1) > 0 Then
        GoTo Finito
    End If
    Worksheets("Output Data").Cells(i + 3, 1) = -PM(i, 1) / b / h
    Worksheets("Output Data").Cells(i + 3, 2) = -PM(i, 2) / b / h ^ 2
    Worksheets("Output Data").Cells(i + 3, 3) = -PM(i, 1) / 1000000
    Worksheets("Output Data").Cells(i + 3, 4) = -PM(i, 2) / 1000000000
    Worksheets("Output Data").Cells(i + 3, 5) = -phic * alpha1 * fc * beta1 * c * b
    Worksheets("Output Data").Cells(i + 3, 6) = c
    Worksheets("Output Data").Cells(i + 3, 7) = epsT
    Worksheets("Output Data").Cells(i + 3, 8) = epsB
Next

Finito:

Worksheets("Output Data").Cells(3, 9) = Worksheets("Output Data").Cells(3, 1)
Worksheets("Output Data").Cells(3, 10) = Worksheets("Output Data").Cells(3, 3)
Worksheets("Output Data").Cells(2, 1) = Worksheets("Output Data").Cells(2, 3) * 1000000 / b / h

End Sub

Private Sub CommandButton1_Click()

Interaction

End Sub
```