A NUMERICAL PROCEDURE FOR THE NONLINEAR ANALYSIS OF REINFORCED CONCRETE FRAMES WITH INFILL WALLS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT EFE GÜNEY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

JULY 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Erdal Çokça
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. M. Uğur Polat
Supervisor

Examining Committee Members

Prof. Dr. S. Tanvir Wasti (METU, CE)

Prof. Dr. Mehmet Utku (METU, CE)

Prof. Dr. Süha Oral (METU, ME)

Prof. Dr. Turgut Tokdemir (METU, ES)

Assoc. Prof. Dr. M Uğur Polat (METU, CE)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Murat Efe Güney

Signature :

ABSTRACT

A NUMERICAL PROCEDURE FOR THE NONLINEAR ANALYSIS OF REINFORCED CONCRETE FRAMES WITH INFILL WALLS

Güney, Murat Efe

M. Sc., Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. M. Uğur Polat

July 2005, 125 pages

Materially non-linear analysis of reinforced concrete frame structures with infill walls requires appropriate mathematical models to be adopted for the beams and the columns as well as the infill walls. This study presents a mathematical model for frame elements based on a 3D Hermitian beam/column finite element and an equivalent strut model for the infill walls. The spread-of-plasticity approach is employed to model the material nonlinearity of the frame elements. The cross-section of the frame element is divided into triangular sub regions to evaluate the stiffness properties and the response of the element cross-section. By the help of the triangles spread over the actual area of the section, the bi-axial bending and the axial deformations are coupled in the inelastic range. A frame super-element is also formed by combining a number of frame finite elements.

Two identical compression-only diagonal struts are used for modeling the infill. The equivalent geometric and material properties of the struts are determined from the geometry of the infill and the strength of the masonry units

A computer code is developed using the object-oriented design paradigm and the models are implemented into this code. Efficiency and the effectiveness of the models are investigated for various cases by comparing the numerical response predictions produced by the program with those obtained from experimental studies.

Keywords: non-linear finite element method; reinforced concrete frames; 3D frame element; spread-of-plasticity; infill wall model; object-oriented design.

ÖZ

DOLGU DUVARLI BETONARME ÇERÇEVELERİN DOĞRUSAL OLMAYAN ANALİZİ İÇİN SAYISAL BİR YÖNTEM

Güney, Murat Efe

Yüksek Lisans, İnşaat Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. M. Uğur Polat

Temmuz 2005, 125 sayfa

Dolgu duvarlı betonarme çerçeve yapıların malzeme açısından doğrusal olmayan 3 boyutlu analizinde gerek kiriş ve kolonlar için, gerekse de dolgu duvarlar için uygun matematiksel modelleriin seçilmesi gerekir. Bu çalışmada kolon ve kirişler için Hermit tipi kiriş elemanına dayanan bir çerçeve sonlu elemanı modeli ile dolgu duvarlar için eşdeğer çapraz çubuk modeli verilmektedir. Betonarme çerçeve elemanlarının malzeme açısından doğrusal olmayan analizi için plastisitenin dağılımı yaklaşımı kullanılmıştır. Çerçeve elemanının rijitlik özellikleri ile deformasyona olan tepkisini hesaplamak üzere elemanın gerçek kesiti üçgenlere bölünmüştür. Tüm kesit alanı üzerine yayılmış bu üçgenler vasıtsıyla çift eksenli eğilme deformasyonları ile eksenel deformasyonlar plastikleşme sonrası birbirleriyle ilişkilendirilmiştir. Ayrıca birden çok çerçeve sonlu elemanı kullanılarak bir çerçeve super elemanı oluşturulmuştur.

Duvar modeli olarak iki adet eşdeğer ve yanlızca basınç altında çalışan çapraz çubuklar kullanılmıştır. Çubukların alanı ve uzunlukları duvarın geometrik özellikleri ve dolgu bloklarının malzeme özellikleri kullanılarak belirlenmiştir.

Yukarıda belirtilen modellerin uygulamasını içeren bir bilgisayar programı nesne yönelimli tasarım anlayışıyla geliştirilmiştir. Çerçeve elemanının etkinliği test edilmiş, ayrıca programın sonuçları deney verileriyle karşılaştırılmıştır.

Anahtar Sözcükler: Doğrusal olmayan sonlu elemanlar yöntemi; betonarme çerçeveler; 3 boyutlu çerçeve elemanı; plastisitenin dağılımı; dolgu duvar modeli; nesne yönelimli tasarım.

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Uğur Polat for his help and guidance. It was my good fortune to study with such an exceptional academician and person. I admire his knowledge and inspiration.

I also want to thank my family for their never ending support. I am grateful for their altruism and patience.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

1.1 Statement of the Problem

Most of the structures, today, are designed based on the results of a linear elastic analysis and the principle of superposition. However, a realistic assessment of an existing structure requires a nonlinear analysis. This is especially important in the process of retrofitting and upgrading such structures for various reasons.

Since the nonlinear analysis of real life structures is computationally very expensive, most of the analysis tools simplify the problem by, for instance, dividing the structure into 2D frames, neglecting the bi-axial bending, assuming the plasticity will take place at certain preset locations along the length of the members and neglecting the gradual yielding of the member cross-section. However, as a result of the rapid improvements in the computer technology in recent years, some procedures, which are computationally more expensive and more precise, are being proposed by researchers.

Frame members and the infill walls are the main components of a typical frame structure. Although the infill walls are mostly considered as non-structural elements, it is known that they have important impact on the behavior of the frames. Several researches have been performed and some procedures have been proposed in the literature to model the nonlinearity of these components. A brief summary of such procedures are given below together with the key features of the object-oriented design in finite elements programming for their implementation.

1.2 Literature Survey

Nonlinear analysis of the frame structures has been studied by researchers for a very long time. Research results reported in the literature show that the presence of infill walls changes the behavior of the structure drastically. Therefore, their influence on the frame structures should not be neglected.

1.2.1 Infill Models

The behavior of infilled frames has been investigated by many researchers. Studies show that the interaction of the infill and the bounding frame determines the behavior of the structure. However, a detailed inclusion of the infills in the finite element model of the frame structures renders the model bulky and sluggish beyond practical limits. Therefore, there has always been an effort to represent their influence on the rest of the structure by a simplified model. Stafford Smith [1] proposed two pin- connected equivalent diagonal struts for modeling the infill walls. The model is based on the elastic theory. He has related the effective width of the diagonal strut with a dimensionless relative stiffness parameter as given below.

$$\lambda = \sqrt[4]{\frac{E_i t \cdot \sin 2\theta}{4 E_c I_c h}} \qquad (1.1)$$

Here,

E_i : Elastic modulus of infill

t : Thickness of infill

θ : Angle between diagonal and horizontal of infill

h : Height of infill

$E_c I_c$: Bending stiffness of columns

Having seen the large gap between the theoretical and the experimental results, Stafford Smith [2] and Mainstone & Weeks [3] have separately proposed an empirical relationship between the effective width of an equivalent strut and the stiffness parameter of Smith [2] for masonry infills.

There are also methods proposed by different researchers, which are based on the theory of plasticity. Woods [4] performed limit analysis of plasticity to determine the collapse load of infilled frame systems with the assumption of perfect plasticity. He has introduced a frame/infill strength parameter, and an empirical penalty factor to reduce the infill strength. Liauw & Kwan [5] have simplified this technique. However, they have neglected the shear forces at the frame/infill interface. They have proposed four types of failure modes by considering the plastic hinges on frames and infill crushing at corners. Both methods adjusted the infill strength in order to narrow the gap between the theoretical and the experimentally observed results. Saneinejad & Hobbs [6] have stated that the limit analysis based on the assumption of perfect plasticity may not be realistic since no plastic collapse mechanism develops at the peak load. Instead, a lower-bound solution was proposed by considering the infill crushing at the corners.

Force equilibrium of a frame with an infill wall under lateral load is shown below in Figure 1.1.

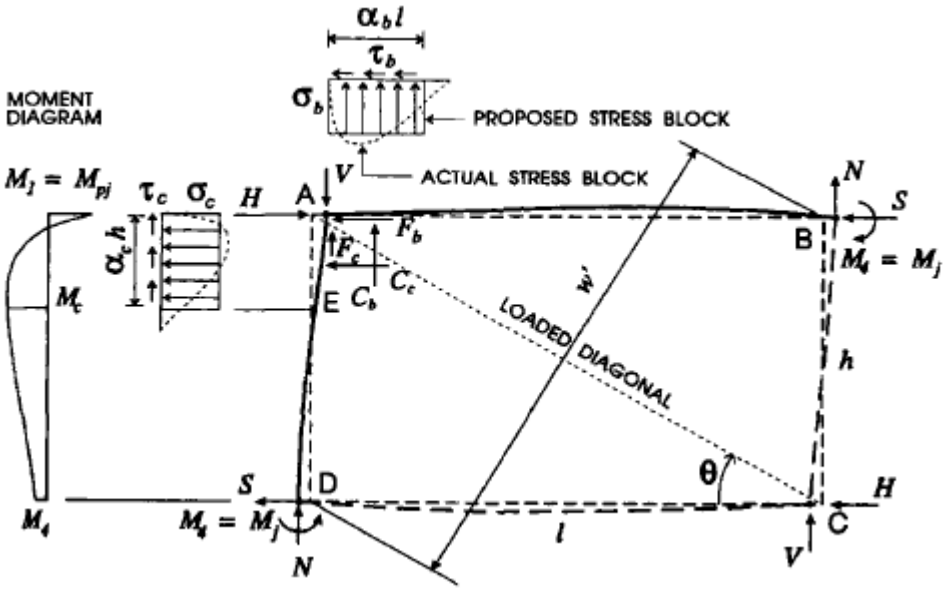

Figure 1.1: Frame Force Equilibrium, Saneinejad & Hobbs [6]

Saneinejad & Hobbs [6] have calculated the collapse load as:

$$H = \sigma_c t(1-\alpha_c)\alpha_c h + \tau_b t \alpha_b l + 2\frac{(M_{pj} + M_j)}{h} \tag{1.2}$$

Here, σ_c and τ_b are the uniform frame infill contact normal and shear stresses, respectively; α_c, α_b are the normalized contact lengths of the column and the beam, respectively; h, l and t are the height, length and thickness of the infill, respectively; M_{pj} is the joint resisting plastic moment on loaded corner and M_j is the moment at the unloaded corner at collapse load level.

For multistory structures Saneinejad & Hobbs [6] have proposed a pin connected equivalent diagonal strut with the cross-sectional area given as:

$$A_d = \frac{(1-\alpha_c)\alpha_c th\frac{\sigma_c}{f_c} + \alpha_b tl\frac{\tau_b}{f_c}}{\cos\theta} \tag{1.3}$$

In Equation (1.3) f_c represents uniform compression strength. This equation was further modified for stability of infill and shear sliding of masonry infilled frames

according to ACI 312.1-89 [7] and ACI 530-88 [8] respectively. Later Madan et al. [9] extended this model with hysteric rule that accounts for strength and stiffness degradation as well as pinching resulting from opening and closing of the gaps between the infill and the bounding frame.

Based on the model developed by Saneinejad & Hobbs[6], a three-strut model was proposed by El-Dakhakhni W. W., Elgaaly M. & Hamid A.A. [10]. In this model the shear stress developing between the infill and the frame is neglected. In other words, the friction coefficient between the infill and the frame is taken as zero, based on the suggestions of Flanagan et al. [11]. El-Dakhakhni W. W. et al. [10] proposed three equivalent struts which have a total area given as:

$$A = \frac{(1-\alpha_c)\alpha_c ht}{\cos\theta} \tag{1.4}$$

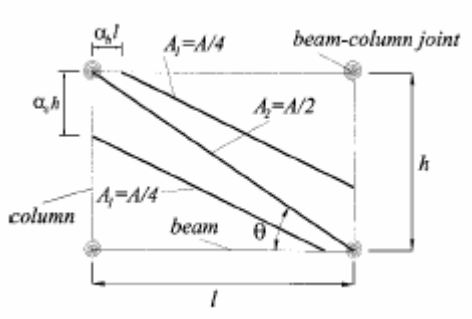

Figure 1.2: Concrete masonry-infilled steel frame model [10]

Distribution of the total area among the diagonal struts is shown in Figure 1.2. It was stated that three-strut model had better simulated the moment distribution along beams and columns. It should be noted, however, that this method only considers the infill corner crushing which is the most common type of failure mode in infilled steel frames.

Shing P.B. and Mehrabi A.B. [12] stated that no single analytical model can account for all possible load resistance mechanisms. It was also noted that the limit analysis methods that account for a variety of possible failure modes are the most promising approaches. They have summarized the failure mechanisms as given in Figure 1.3. It was also stated that the mechanism that results in the lowest lateral

resistance is the dominant failure mechanism and the corresponding load determines the maximum lateral load. Mainly, five failure mechanisms and the corresponding frame and infill load resistances proposed by Shing and Mehrabi [12] are as follows.

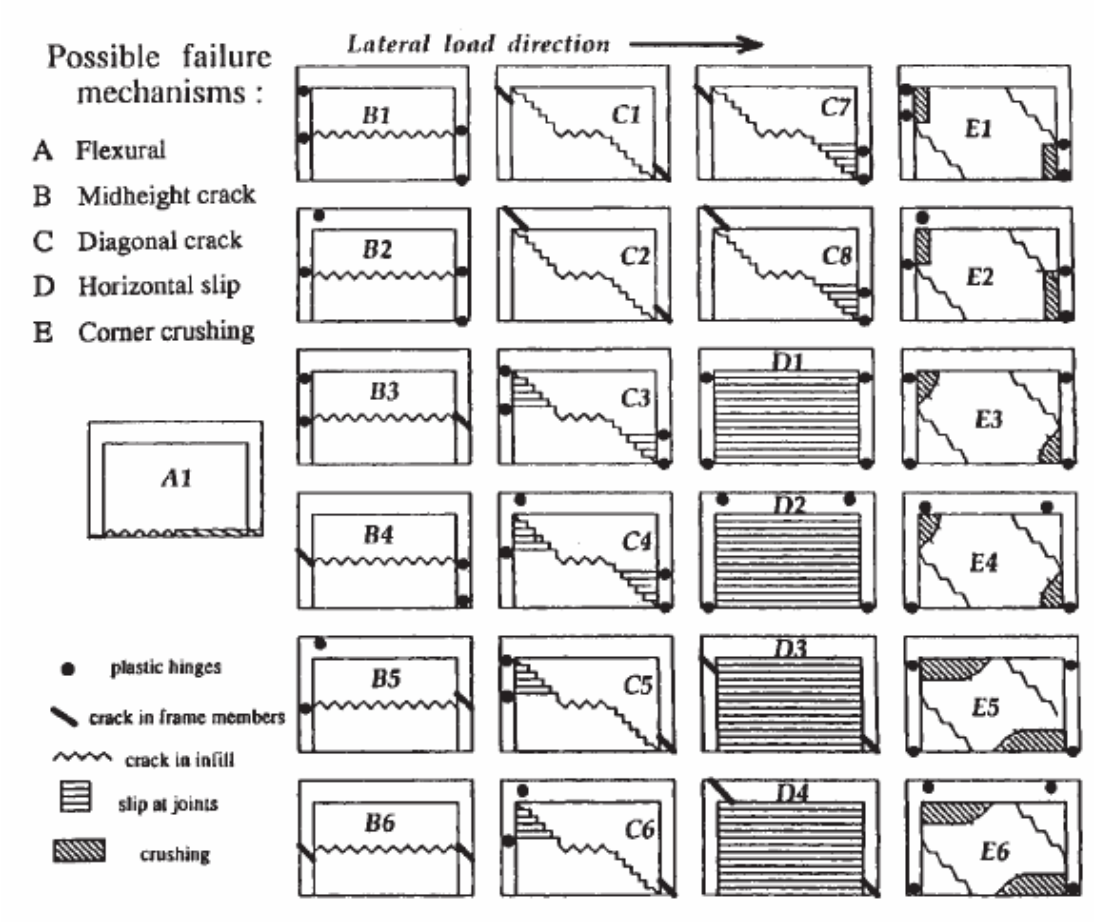

Figure 1.3: Failure Mechanisms of infilled frames [12]

FAILURE MECHANISM 1

This mechanism corresponds to horizontal sliding failure of the infill at mid-point. The lateral resistance in this case is the sum of the shear forces in the columns and the residual shear resistance of the wall. The resistance of the frame is governed by the hinges formed at one end and the mid-height of each column.

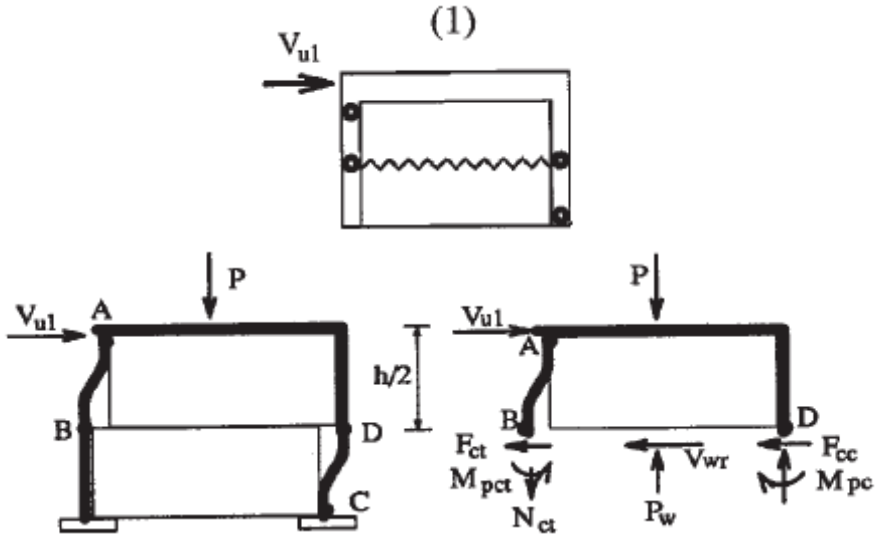

Figure 1.4: Failure Mechanism 1 [12]

FAILURE MECHANISM 2

Here, the shear failure develops at one or more locations in the columns. This is the main distinction form the mechanism 1. Lateral resistance is the sum of the ultimate shear resistance of the windward column, the shear force in the leeward column and the residual shear resistance along the horizontal crack of infill.

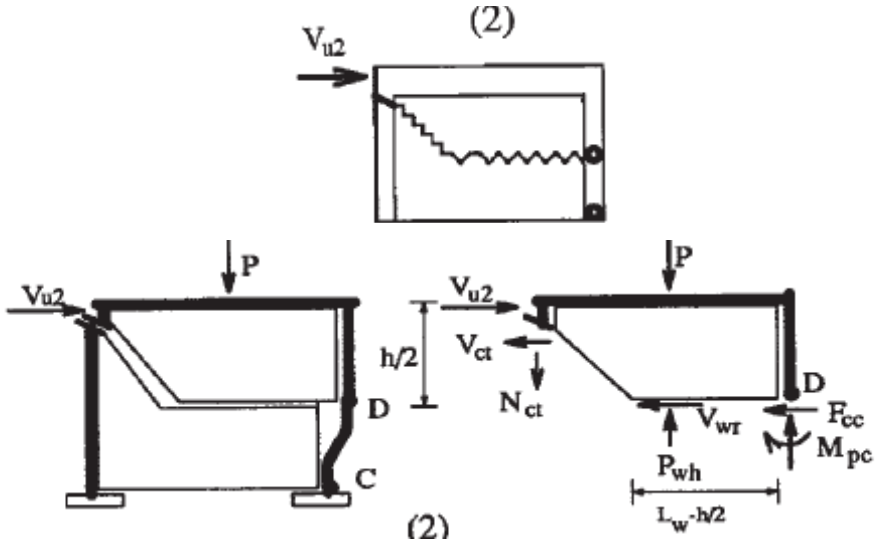

Figure 1.5: Failure Mechanism 2 [12]

FAILURE MECHANISM 3

In this mechanism masonry reaches the crushing strength along the wall to frame interface. Also, plastic hinges develop near the beam-to-column joints and at points B as shown in Figure 1.6.

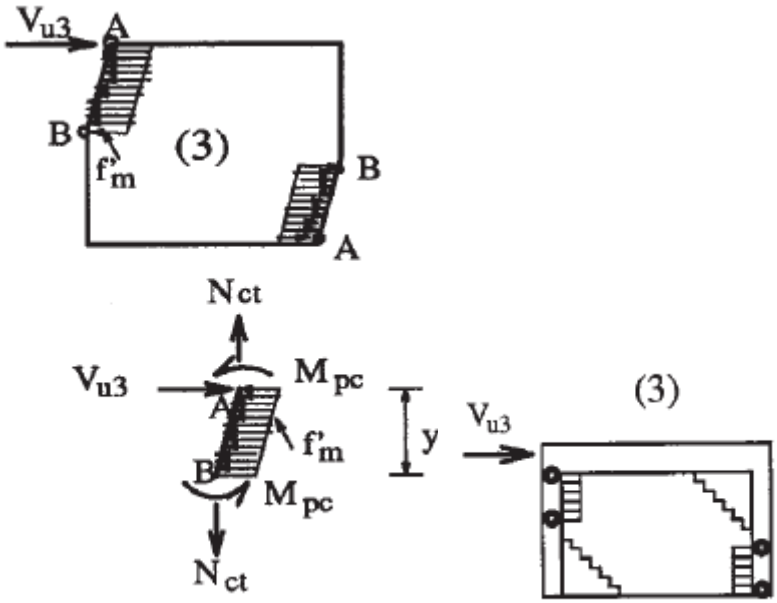

Figure 1.6: Failure Mechanism 3 [12]

FAILURE MECHANISM 4

Infill reaches its compressive strength at corners and plastic hinges are formed at both ends of the column. The wall-to-column interface has a parabolic distribution along the contact length.

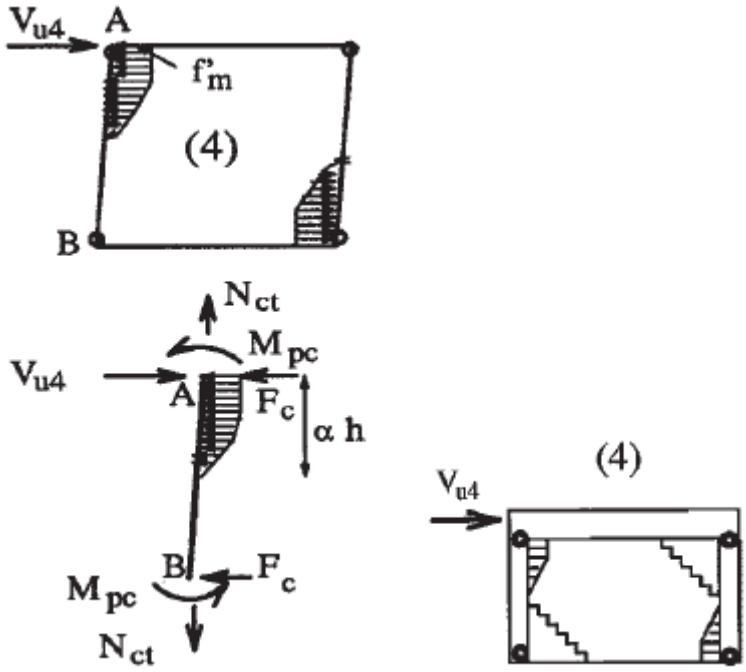

Figure 1.7: Failure Mechanism 4 [12]

FAILURE MECHANISM 5

In this mechanism the infill and the frame are considered as two parallel systems with compatible displacements at the compression corners. The lateral resistance is the sum of the flexural resistance of the frame and the residual shear resistance of the wall.

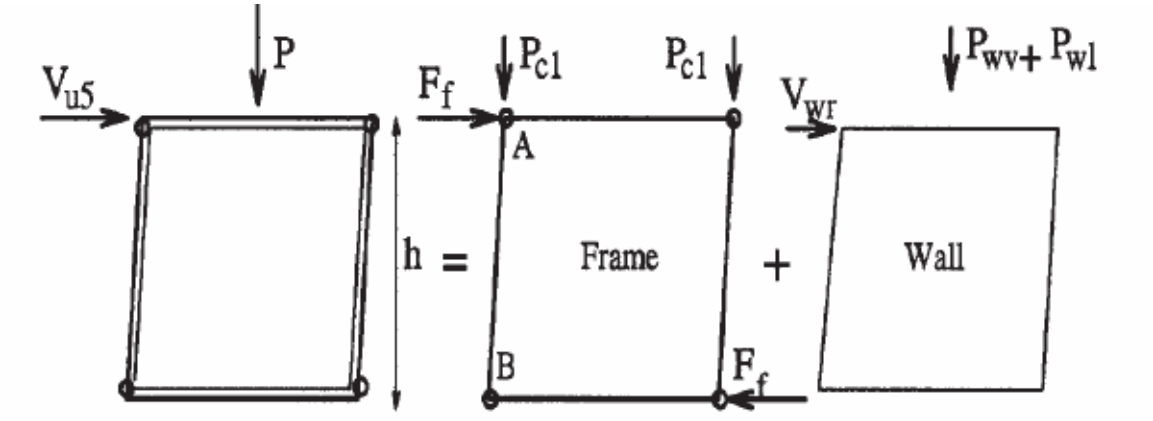

Figure 1.8: Failure Mechanism 5 [12]

Shing and Mehrabi [12] have also remarked that the Mechanism 5 is dominant for the specimens with weak infills. On the other hand, Mechanism 2 dominates for the specimens with strong infills.

Various material models were proposed for the equivalent struts. According to the available literature, the stressed part of the infill panel is a diagonal region connecting the loaded corners of the frame. Similarly, the material properties of the infill are governed by the properties in diagonal direction.

Saneinejad & Hobbs [6] have proposed secant modulus of elasticity of their diagonal strut as:

$$E_d = \frac{f_c}{\varepsilon_c} = \frac{d \cdot f_c}{\Delta_d} \tag{1.5}$$

where Δ_d is the diagonal deflection of the infill at the peak load and d is the diagonal length of the infill and f_c is effective compressive strength of the infill. Δ_d may be related to lateral deflection Δ_h by $\Delta_d = \Delta_h \cos\theta$. Saneinejad & Hobbs [6] have given a lateral deflection value at peak loads based on the nonlinear finite element analyses as follows.

$$\Delta_h = 5.8\varepsilon_c h \cos\theta (\alpha_c{}^2 + \alpha_b{}^2)^{0.333} \tag{1.6}$$

where ε_c is the infill strain at peak uniaxial (unconfined) compression, other terms are as defined previously. Saneinejad & Hobbs [6] states that the initial stiffness of the strut can be taken as twice its secant stiffness.

El-Dakhakhni W. W. et al. [10] have given the elastic modulus of the panel in a diagonal direction as:

$$E_\theta = \frac{1}{\frac{1}{E_0}\cos^4\theta + (-\frac{2v_{0-90}}{E_0} + \frac{1}{G})\cos^2\theta + \frac{1}{E_{90}}\sin^4\theta} \tag{1.7}$$

The ultimate compressive strength $f'_{m-\theta}$ was related to the E_θ by the same factor relating f'_{m-90} .to E_{90}. Knowing $f'_{m-\theta}$ and E_θ the strain corresponding to the peak load ε_p was calculated. Additionally, a tri-linear relation was proposed for the material model of the struts as given in Figure 1.9.

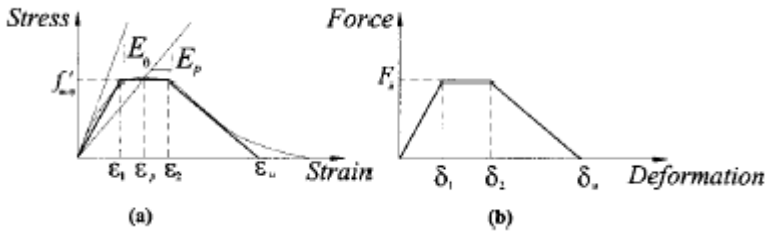

Figure 1.9: Simplified tri-linear relations: a) stress–strain relation of concrete masonry; b) typical force–deformation relation for struts [10]

1.2.2 Nonlinear Frame Element Models

Nonlinearity is divided into two parts. First one is the material nonlinearity and the second is the geometric nonlinearity. In this study only the material nonlinearity is considered. Therefore, a review of the literature only about this source of nonlinearity is given here.

There are mainly two approaches in handling the material nonlinearity. In the first approach the nonlinearity is assumed to happen at discrete locations of the beam. Plastic hinges with zero lengths are assembled at certain points of the frame (generally at the member ends) to take he material nonlinearity into account. Some early studies about plastic hinges were performed by Heyman, J [18], Porter FL and Powell GH [19] and Nigam NC [20]. In these studies a single yield surface was proposed which behaves perfectly plastic after a certain strain level. Later, plastic hinges composed of several yield surfaces were proposed to take gradual yielding of the section into account by El-Tawil and Deierlein [21] and Liew and Tang [22].

Second approach for the materially nonlinear analysis of the frames is the spread of plasticity procedure. Here, a number of control points are selected along the frame

element and the element cross-section is discretized into fibers (or monitoring grids). Meek and Lin [24] investigated nonlinear analysis of closed thin-walled beams and columns with the spread of plasticity approach. Bild et al. [25] has proposed a formulation to determine the strength of the steel frames under bi-axial bending and torsion. Teh and Clarke [26] claimed that interaction between the axial force and the bi-axial moment could be modeled more accurately by using plastic zone approach. They have proposed a co-rotational formulation for materially and geometrically nonlinear analysis of steel frames composed of the compact tubular sections and open sections. They derived governing equations using virtual work principle, and used von Mises yield criterion with an associated flow rule. Effect of torsional warping was neglected in their study.

Jiang et al. [27] stated that most of the studies about the spread of plasticity had dealt with planar frames or behavior of the members and little work has performed for large scale 3D analyses. They employed an updated Lagrangian formulation for the formulations of the small displacement element stiffness matrix and geometric stiffness matrix. They discretized element cross section into a number of monitoring fibers. Yielding of the fibers was determined according to the von Mises yield criterion and associated flow rule. Discretized cross section of Jiang et al. [27] is given in Figure 1.10.

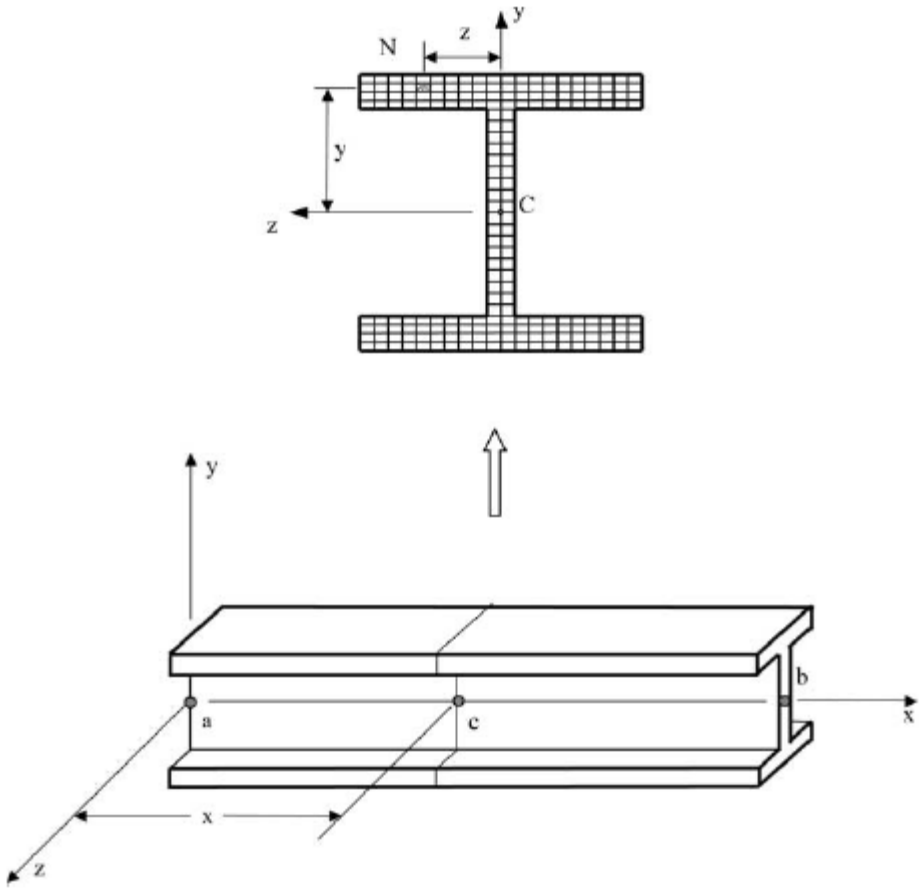

Figure 1.10: Discretization of the frame's cross section in the spread-of-plasticity analysis [27]

Additionally Jiang et al. [27] used mixed element technique for the analysis of a large scale 3D steel structure. In mixed element technique they have used beam-column elements and spread-of-plasticity elements concurrently.

Researchers mostly used displacement based frame elements in the studies summarized up to now. Neuenhofer A, Filippou FC [28] has evaluated the accuracy and effectiveness of the displacement based and force based frame elements. They stated that displacement based elements gave approximate results for nonlinear problems and to obtain more accurate results one must refine the mesh or use higher order polynomials to approximate the actual displacement fields along the element. They also stated that in flexibility approach, force fields of the element could easily be reproduced by using force interpolation functions in the absence of geometric

nonlinearity. This was because the force equilibrium conditions are rather simple to satisfy for a frame element. Constant axial force and linearly varying moment give the exact force field if there is no distributed load on the element. Neuenhofer A, Filippou FC [28] also noted that it was hard to integrate flexibility based state determination of an element into a displacement based nonlinear program. However, they claim that recent proposals overcome this shortcoming by using an iterative state determination approach to calculate the stiffness matrix and the response of the element. However, this is a computationally intensive approach. Finally, they have proposed a non-iterative state determination approach for flexibility based frame elements.

Recently, Taylor et al. [29] proposed a frame finite element, which was formulated with three-field variational form, which was based on Hu-Washizu variational principle. Axial force, moment and discontinuous strain fields were approximated at each element. They have considered the effect of axial, bending and shear for material nonlinearity. It is stated that the formulations created a shear deformable element, which is free of locking. The formulations are for 2D case and valid for the case of small displacements.

1.2.3 Object-Oriented Finite Element Programming

Many researchers have studied the potential of object-oriented finite element programming for structural problems. It is stated in the literature that object-oriented programming is very suitable for developing robust and efficient algorithms to scientific problems. Commend and Zimmerman [31] summarizes the key features of object-oriented design as follows:

- **Robustness and modularity: encapsulation of the data**

  Each object stores its data and procedures under a single class. Additionally members of the class may be hidden from the other components of the program.

- **Inheritance and polymorphism: the hierarchy of the classes**

  Every object is an instance of a class, which is an abstract data type. Classes can be built in a hierarchy. A derived class inherits the members and procedures of its parent class. Additionally, procedures of the classes may act differently according to the type of the object, which is called polymorphism.

- **Non-anticipation and state encapsulation**

  It is stated that if the content of a method does not depend on the state of the variables, then robustness of the code will increase.

Object-oriented designs up to now mostly came up with the following finite element objects: node, domain, element, load, material model etc. Node object stores its degrees of freedom and displacements. Domain is composed of the elements, nodes, loads and other components of the finite element method. Components of the domain can be queried or new components can be added to the domain. Element class represents the base class for finite elements. Loading of the system is defined by load class. Finally, element's material properties are stored in material model class.

McKenna [16] stated that existing object-oriented finite element designs considered the analysis class as a black box, since its operations were opaque. He has proposed an analysis class, which has an interface that allows the use of different analysis algorithms. Supplying adequate class instances for the constructor of the analysis class, the class defines its components for the solution algorithm. Later, analysis is performed using the components of the analysis class.

Pelerin & Pegon [34] separated the domain from the problem type to increase the modularity of the program. Additionally they have proposed a new intermediate class for handling the data like stiffness matrix, mass matrix etc. in an efficient way.

1.3 Aim of the Study

Objective of this study is to develop a tool for the nonlinear analysis of 3D reinforced concrete frame structures with or without infill walls. For this purpose a 3D Hermitian beam finite element is used which is based on the usual Euler-Bernoulli assumptions. Spread of plasticity approach is used to take the material nonlinearity into account. Here, the cross-section of the frame element is discretized into triangular elements to monitor gradual yielding of the cross section. Since the frame finite element gives approximate displacement field for the nonlinear case, a frame member is further discretized into several sub elements along its length. Additionally, an infill model, which is mainly based on, the model proposed by Saneinejad and Hobbs [6] is used to take the infill walls into account.

Later, a program is developed for the nonlinear analysis of the R/C frame structures which includes implementation of the items stated above. Program is developed using the object-oriented design to allow the changes and addition of new finite elements or some other components to the code in the future.

Next, the parameters affecting the accuracy of the solution are investigated. Some sample problems are solved to determine the optimum values of these variables. Later, the response predicted by the proposed procedure is compared with the experimental data available.

Finally, the limitations of the program and a guideline for the future studies are given.

CHAPTER 2

FORMULATIONS

2.1 Infill Wall Modeling

2.1.1 Equivalent Strut Model

The most popular approach for infill modeling is taken in this study and the masonry infill is modeled by two diagonal struts connecting the opposite corners of the frame. For the equivalent area of the struts, basically, the model proposed by Saneinejad and Hobbs [6] is used. The equivalent strut area is given in Equation (1.3). The ultimate load capacity of the infill is also taken as proposed by Saneinejad and Hobbs [6]

$$H = \sigma_c t(1-\alpha_c)\alpha_c h + \tau_b t \alpha_b l \tag{2.1}$$

where α_c and α_b are the normalized lengths of contact between the infill and the beam and the column, respectively. Their upper-bound values of these contact lengths are defined as.

$$\alpha_c h = \sqrt{\frac{2(M_{pj} + 0.2M_{pc})}{\sigma_{c0} t}} \leq 0.4h \tag{2.2}$$

$$\alpha_b l = \sqrt{\frac{2(M_{pj} + 0.2M_{pb})}{\sigma_{b0} t}} \leq 0.4l \tag{2.3}$$

where M_{pc} and M_{pb} are the effective plastic moment capacities of the column and the beam, respectively and M_{pj} is the effective joint plastic moment which is the smaller of the beam or column plastic moment capacity. In Equation (2.2) and Equation (2.3) σ_{c0} and σ_{b0} are the nominal contact stresses defined as.

$$\sigma_{c0} = \frac{f_c}{\sqrt{1+3\mu^2 r^4}} \tag{2.4}$$

$$\sigma_{b0} = \frac{f_c}{\sqrt{1+3\mu^2}} \tag{2.5}$$

in which f_c is the factored (effective) compressive strength of the infill taken as $0.39 f_m^{'}$ where $f_m^{'}$ is the masonry strength. μ is the frictional coefficient along the frame-infill interface. El-Dakhakhni et al. [10] proposes that $f_{m-0}^{'}$ (masonry strength parallel to bed joints) and $f_{m-90}^{'}$ (masonry strength normal to bed joints) can be used for the masonry strengths in Equation (2.4) and Equation (2.5), respectively.

A study reported by Flanagan et al. [11] shows that the model proposed by Saneinejad and Hobbs [6] overestimates the corner-crushing load by as much as twice the actual capacity based on the experimentally observed results. They performed a parametric study for the coefficient of friction μ in the formulation of Saneinejad and Hobbs [6]. It is reported that taking $\mu = 0$ gives the best results with a mean ratio of 74%.

In light of the above discussion Equations (2.2) and (2.3) can be rewritten in the following forms as given by El-Dakhakhni et al. [10].

$$\alpha_c h = \sqrt{\frac{2(M_{pj} + 0.2M_{pc})}{f'_{m-0} t}} \leq 0.4h \tag{2.6}$$

$$\alpha_b l = \sqrt{\frac{2(M_{pj} + 0.2M_{pb})}{f'_{m-90} t}} \leq 0.4l \tag{2.7}$$

Also, the Equations (2.4) and (2.5) can be expressed as:

$$\sigma_{c0} = f'_{m-0} \tag{2.8}$$

$$\sigma_{b0} = f'_{m-90} \tag{2.9}$$

Finally, the ultimate horizontal load for corner crushing reduces to the following form.

$$H = f'_{m-0} t(1-\alpha_c)\alpha_c h \tag{2.10}$$

Hence, the ultimate load for the diagonal strut can be written as:

$$R = \frac{f'_{m-0} t(1-\alpha_c)\alpha_c h}{\cos\theta} \tag{2.11}$$

2.1.2 Material Model for the Equivalent Struts

Saneinejad and Hobbs [6] suggest equivalent uniform stress block for the stress distribution at the middle of the infill wall. Uniform stress block is used to check the stability of the middle of the infill. In this study width of the stress block is used to determine the initial stiffness of the infill wall together with the modulus of elasticity of the infill in diagonal direction. This approach is acceptable since stress distribution along the diagonal is known to be the similar at the initial stages of loading. Width of the uniform stress block (effective width of the infill) is given as:

$$w = 0.5h / \cos\theta \tag{2.12}$$

Here h is the height of the infill, and θ is the angle between the diagonal of the infill and the horizontal line. Using the effective width of the infill, initial stiffness of the infill can be found as:

$$K = \frac{E_\theta wt}{l} \tag{2.13}$$

Where w is effective width of the infill, t is the thickness of the infill and l is the length of the diagonal of the infill and E_θ is the modulus of elasticity of the infill in diagonal direction. Knowing E_0 and E_{90} (Young's moduli in the directions parallel and normal to the bed joints respectively) El-Dakhakhni et al. [10] state that modulus of elasticity in θ direction can be found by Equation (1.7). It is also stated that using Equation (1.7) results with E_θ values about %80 of E_{90}. This is consistent with the test results reported by Bennet et al. [13]. Bennet et al. [13] reports modulus of elasticity of the masonry prisms loaded in different orientations. According to the results ratios of $E_{22.5}, E_{45}$ and $E_{67.5}$ to E_{90} has a mean of %78. Therefore E_θ may be related to E_{90} as:

$$E_\theta = 0.8E_{90} \tag{2.14}$$

E_{90} can be found from the Equation (2.15) given by Bennet et al. [13] which relates prism strength of the masonry and modulus of elasticity at the half of the prism strength.

$$E_{90} = 0.68 f_{m-90}{}' + 1.5 \tag{2.15}$$

Here E_{90} is modulus of elasticity in vertical direction (GPa) and $f_{m-90}{}'$ is masonry prism strength in vertical direction (MPa).

Additionally prism strength of the masonry is related to unit strength of masonry in vertical direction by Bennet et al. [13] as:

$$f'_{m-90} = 0.3 f_{u-90} + 0.9 \tag{2.16}$$

Therefore knowing the unit masonry strength f_{u-90}, f_{m-90}' can be determined using Equation (2.16), then E_{90} and E_θ are found using Equations (2.15) and (2.14) respectively.

To determine R from Equation (2.11), one needs to know f_{m-0}' (prism strength in horizontal direction). Based on the results of the past experiments presented by Bennet et al. [13], linear regression analysis is performed to determine relationship between unit masonry strength and prism strength in horizontal direction. Equation found is given as:

$$f_{m-0}' = 0.11 f_{u-0} + 1.18 \tag{2.17}$$

The displacement at the ultimate load and corresponding strain value can be expressed as:

$$\Delta = \frac{R}{K} \tag{2.18}$$

$$\varepsilon_u = \frac{\Delta}{l} \tag{2.19}$$

where R is given in Equation (2.11) and l is the lengths of the diagonal struts. For the material model of the struts El-Dakhakhni et al. [10] proposed a yield plateau with a length of 0.002. Additionally, in the material model the stress of the strut goes down to zero at a strain value of 0.01. Using these values a material model similar to the one proposed by El-Dakhakhni et al. [10] is constituted. Figure 2.1 shows the material model of the diagonal struts in terms of the variables given in this chapter.

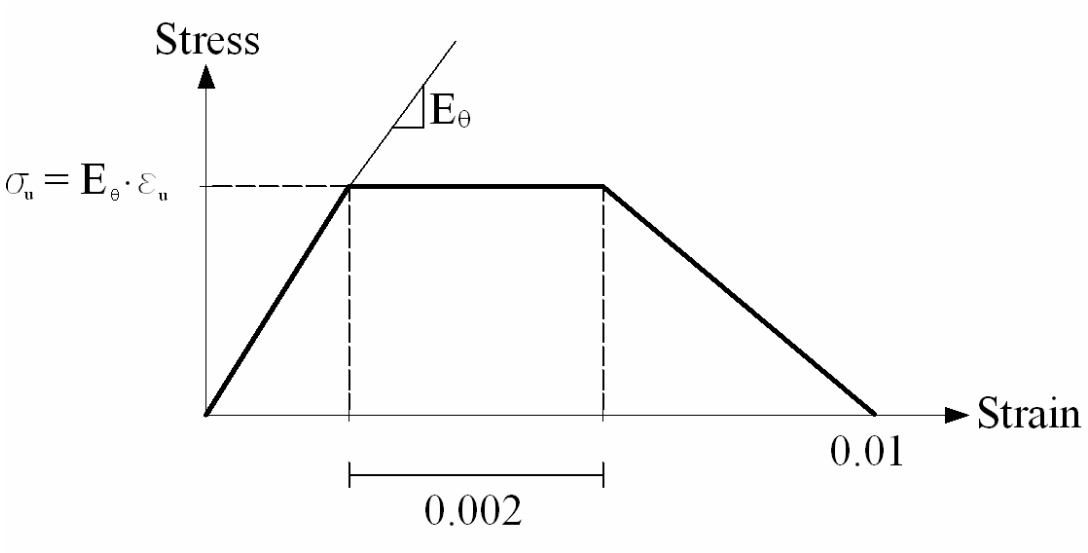

Figure 2.1: Stress strain relationship of the equivalent diagonal struts

2.2 Nonlinear Frame Element

2.2.1 Displacement Based Frame Element

Displacement based Hermitian beam finite element with the usual Euler-Bernoulli assumptions is used in this study for modeling the frame elements. The nodal displacements of a 2D Hermitian beam element are shown in Figure 2.2.

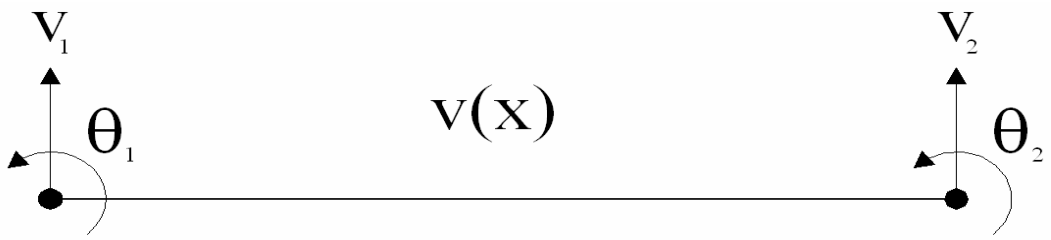

Figure 2.2: Degree of freedoms of 2D Euler-Bernoulli beam

The shape functions used for approximating the transverse displacement field along this element are shown in Figure 2.3.

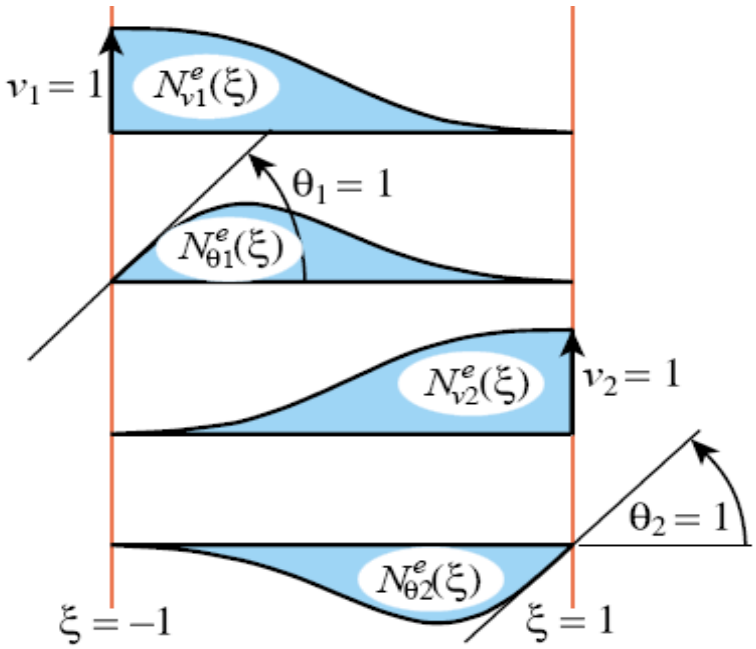

Figure 2.3: Hermitian cubic shape functions for the beam element in local coordinates [41]

Expressions of the shape functions in terms of the local coordinate ξ are.

$$N^e{}_{v1} = \frac{1}{4} \cdot (1-\xi)^2 \cdot (2+\xi)$$

$$N^e{}_{\theta1} = \frac{1}{8} \cdot l \cdot (1-\xi)^2 \cdot (1+\xi)$$

$$N^e{}_{v2} = \frac{1}{4} \cdot (1+\xi)^2 \cdot (2-\xi)$$

$$N^e{}_{\theta2} = \frac{1}{8} \cdot l \cdot (1+\xi)^2 \cdot (1-\xi) \qquad (2.20)$$

The transverse displacement field along the beam can be approximated in terms of the element shape functions and the nodal displacements as.

$$v^e = \begin{bmatrix} N^e{}_{v1} & N^e{}_{\theta 1} & N^e{}_{v2} & N^e{}_{\theta 2} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{bmatrix} \tag{2.21}$$

$$v^e = N \cdot u^e \tag{2.22}$$

The curvature of the beam can be expressed in terms of the nodal displacements by the help of the **B** matrix. **B** matrix relates the nodal displacements to the curvature strain of the element.

$$B = \frac{1}{l}\left[6\frac{\xi}{l} \quad 3\xi - 1 \quad -\frac{6\xi}{l} \quad 3\xi + 1 \right] \tag{2.23}$$

$$\kappa = B \cdot u^e \tag{2.24}$$

Tangential stiffness matrix and the response of the element can be expressed as in Equation (2.25) and (2.26), respectively.

$$K^e = \int_{-1}^{1} EI \cdot B^T \cdot B \cdot \frac{1}{2 \cdot l} \cdot d\xi \tag{2.25}$$

$$R^e = \int_{-1}^{1} B^T \cdot M \cdot \frac{1}{2 \cdot l} \cdot d\xi \tag{2.26}$$

In Equation (2.26), M is the distributed moment along the element.

The element formulation given for a planar beam can generalized for a 3D beam element. The nodal displacements to be considered are shown in Figure 2.4. Transverse displacements in the local xy plane can be approximated by the nodal displacements v_{y1}, v_{y2}, θ_{z1} and θ_{z2}. Similarly, the transverse displacements in the local xz plane can be approximated by the nodal displacements v_{z1}, v_{z2}, θ_{y1} and θ_{y2}. Therefore, the 2D formulation can easily be generalized to 3D. However, it

should be noted that some adjustment in the signs of shape functions are necessary for bending in xz plane.

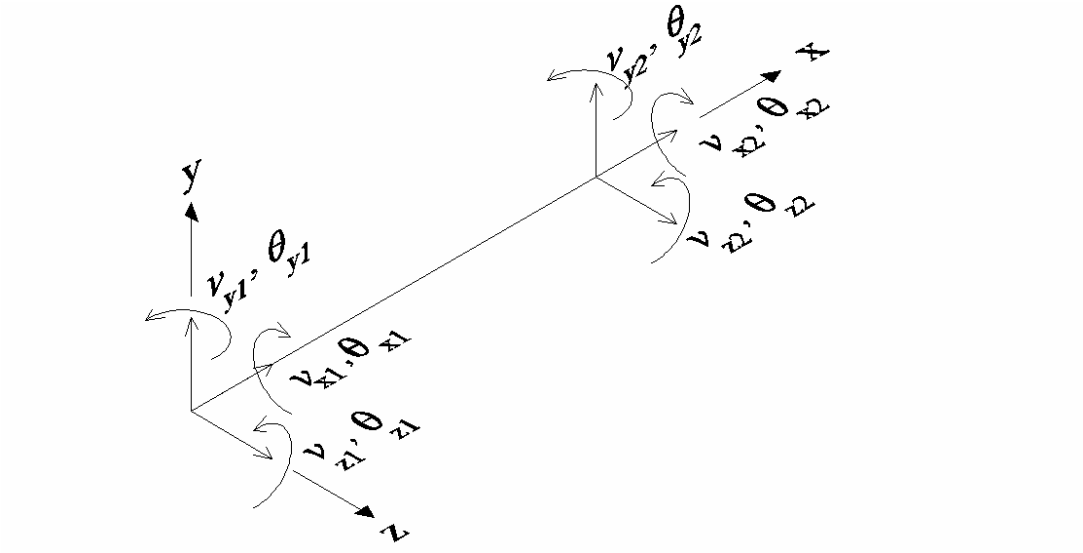

Figure 2.4: Degrees of freedoms of 3D beam element

The shear deformations are neglected in this formulation. Hence, the element formulation is valid for shallow beams, which is usually the case in practice.

Besides the transverse displacements an uncoupled and linearly varying axial displacement field is assumed along the element. The axial strain-displacement matrix of the element is of the form

$$B = \frac{1}{l} \cdot \begin{bmatrix} -1 & 1 \end{bmatrix} \tag{2.27}$$

Using the **B** matrix the stiffness matrix and the axial response of the bar element can be determined as

$$K^e = \int_{-1}^{1} B^T \cdot B \cdot EA \cdot \frac{1}{2 \cdot l} \cdot d\xi \tag{2.28}$$

$$R^e = \int_{-1}^{1} B^T \cdot F \cdot \frac{1}{2 \cdot l} \cdot d\xi \tag{2.29}$$

Note that F in Equation (2.29) is the axial force distributed along the length of the frame element.

2.2.2 Material Model for the Frame Element

Material nonlinearity of the beam is modeled by the spread-of-plasticity approach. In steel frames plastic hinges may be used for the nonlinear analysis since the plastification occurs at certain locations such as member ends. However, in reinforced concrete frames, the plastic region is usually not localized but generally spread along the member length and over the cross-section. In order to monitor the spread of plastification, the element cross-section is divided into a number of triangular subregions at some pre-selected critical points along the member length. In this study, these point are either equally spaced along the member or they are selected as the Gaussian integration points. First step in the calculation of the element response and the element tangential stiffness matrix is the calculation and then integration over the cross-section of the element stresses (Figure 2.5). Integrations in Equations (2.25), (2.26), (2.28) and (2.29) along the element are performed afterward.

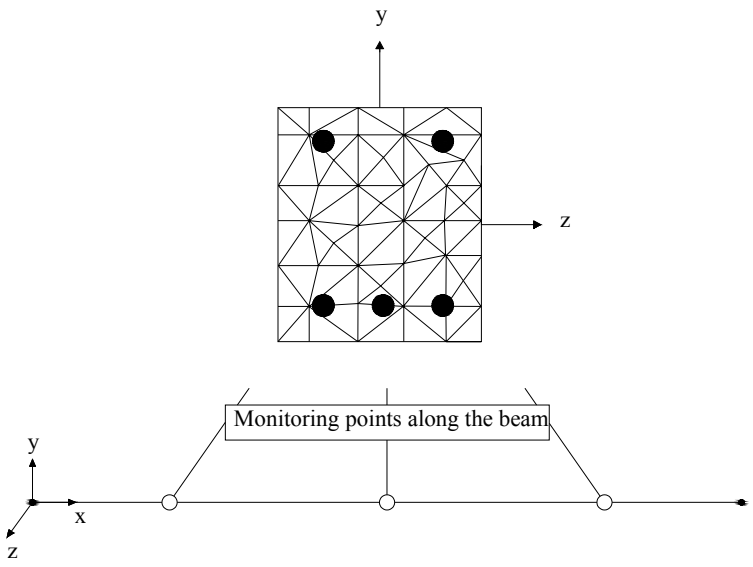

Figure 2.5: Discretization of the element cross-section at monitoring locations along the reinforced concrete frame element.

When calculating the element response the curvature in both directions and the axial strain due to axial deformations are coupled. The normal strain created by the curvature is calculated first at the centroid of each triangular sub-region. Later, the axial strain due to axial deformations (which is the same for all triangular sub-regions) is added to the strain due to curvature. Then, the axial force response is calculated for each triangle and the moment response is obtained about a reference point on the section by taking the moment about this point of the triangular element forces. The reference point is chosen to be the attachment point of the element for compatibility.

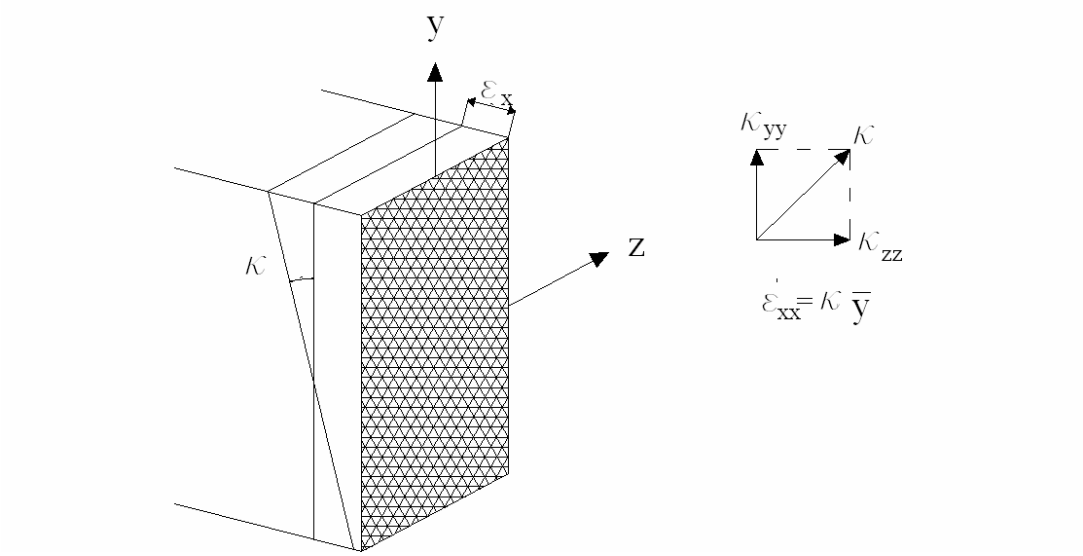

Figure 2.6: Strain distribution over the element cross-section for a given axial strain and curvature.

Hognestad parabola shown in Figure 2.7 is used for the stress-strain relationship of concrete. The effect of confinement is neglected for simplicity. Additionally, it is assumed that the concrete takes no tension. An idealized bilinear elasto-plastic material model with strain hardening is used for calculating the steel response.

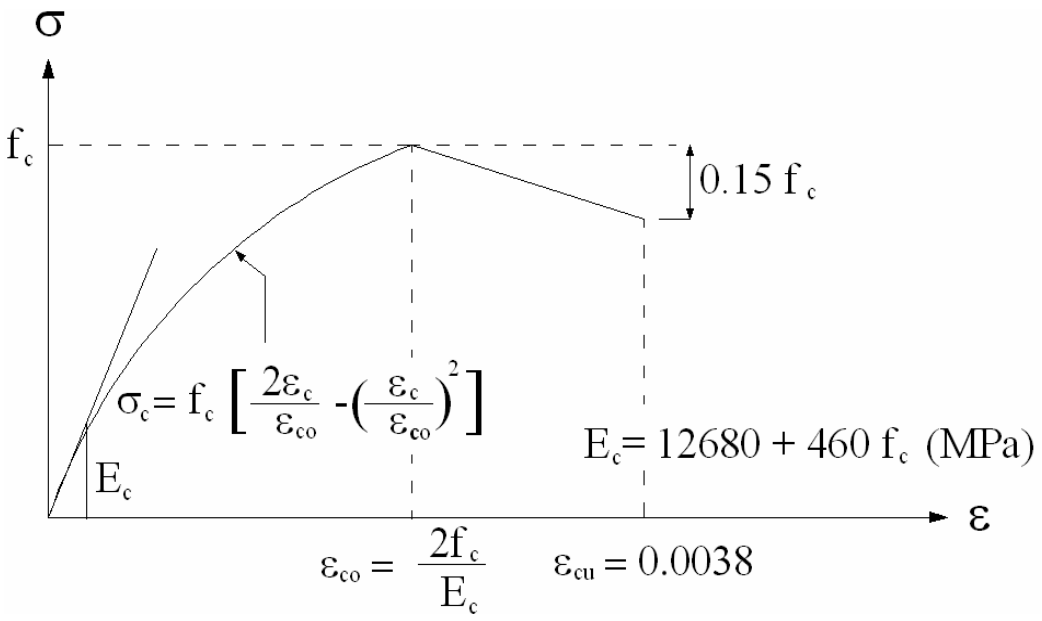

Figure 2.7: Hognestad Parabola [30]

2.2.3 Tangential Stiffness Matrix for Coupled Material Model

The axial and the flexural response of the linear Hermitian frame element are uncoupled. However, as stated earlier, the axial and the flexural response become coupled in the nonlinear plastic range. In other words, a displacement or a rotation corresponding to one of the degrees of freedom will generally produce forces and moments at each of the other degrees of freedom. If the stiffness matrix of a regular frame element is used for the response calculations in the inelastic phase by ignoring this coupling effect, each load increment requires a large number of iterations to converge. Therefore, the stiffness matrix should be reconstructed by considering this coupling effect.

A numerical procedure is adopted for computing the tangential element stiffness matrix in the inelastic range. First, a tangential sectional rigidity matrix is formed at each integration point by using a numerical procedure as follows.

The current strain state at the section is incremented by a small amount $\Delta\varepsilon_{xx}$ in the axial x-direction and the corresponding increase in the sectional response is calculated. As shown in Figure 2.8, in general, an axial force increase of ΔN_{xx} accompanied by moment response increments of ΔM_{yx} and ΔM_{zx} about the y and z axes is calculated.

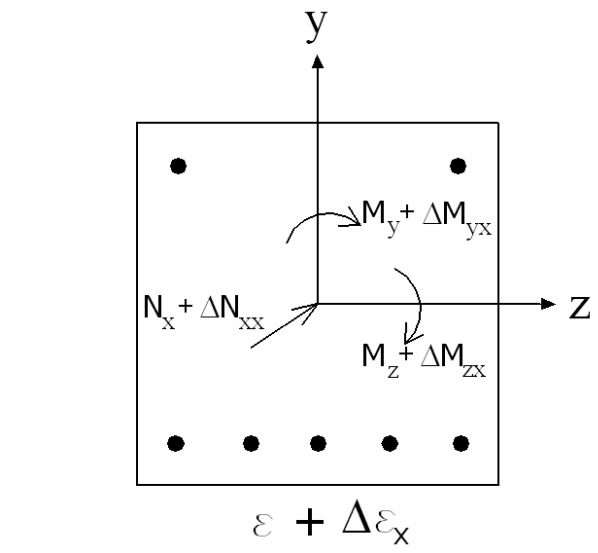

Figure 2.8: Response of the section for small axial strain increment

These response increments are then divided by the given axial strain increment $\Delta\varepsilon_{xx}$ to obtain the tangential section rigidity parameters for a unit axial strain.

The procedure is repeated by applying small curvature increments $\Delta\kappa_y$ and $\Delta\kappa_z$ about y- and z-axes, respectively, and calculating the corresponding increments in the axial force and the moment response of the section, as shown in Figure 2.9.

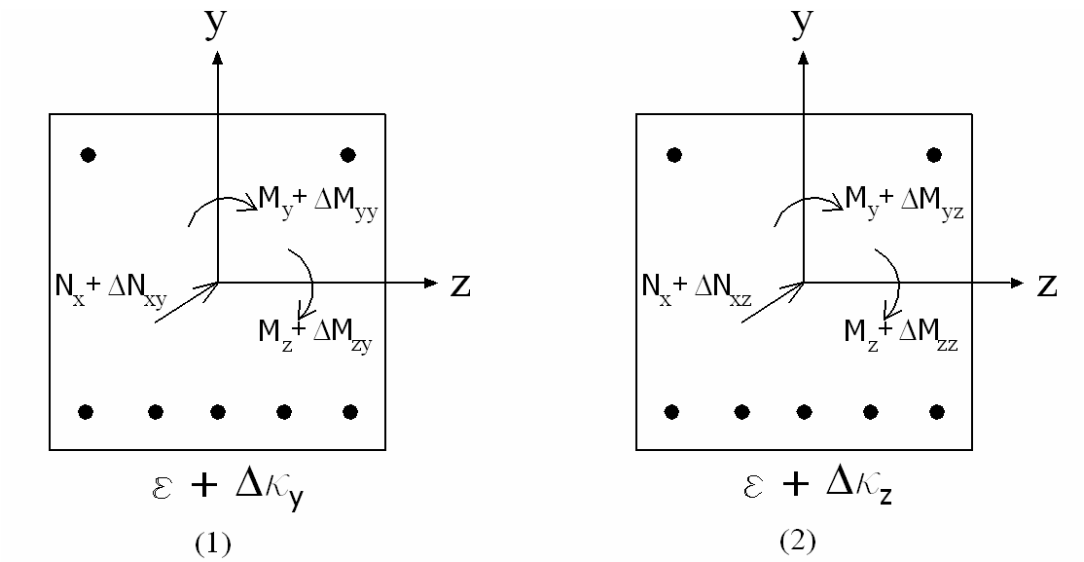

Figure 2.9: Response of the section for small curvature strain increments

Finally, the sectional rigidity matrix can be expressed as

$$\begin{bmatrix} N_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} \frac{\Delta N_{xx}}{\Delta \varepsilon_x} & \frac{\Delta N_{xy}}{\Delta \kappa_y} & \frac{\Delta N_{xz}}{\Delta \kappa_z} \\ \frac{\Delta M_{yx}}{\Delta \varepsilon_x} & \frac{\Delta M_{yy}}{\Delta \kappa_y} & \frac{\Delta M_{yz}}{\Delta \kappa_z} \\ \frac{\Delta M_{zx}}{\Delta \varepsilon_x} & \frac{\Delta M_{zy}}{\Delta \kappa_y} & \frac{\Delta M_{zz}}{\Delta \kappa_z} \end{bmatrix} \cdot \begin{bmatrix} \Delta \varepsilon_x \\ \Delta \kappa_y \\ \Delta \kappa_z \end{bmatrix} \tag{2.30}$$

Having obtained the sectional rigidity matrices at each integration point along the frame element, it is possible to obtain the coupled tangential element stiffness matrix by the basic definition of nodal stiffness coefficients. A unit value is prescribed for each of the element degrees of freedom in turn while keeping the rest of them at zero. The axial strain or curvature increment corresponding to this pivotal degree of freedom is calculated at each integration point and this strain increment is converted to axial force and moment increments at the section using the sectional rigidity matrix at the integration point. Finally, the element response vector is obtained by integrating these incremental sectional responses via Equation (2.26) and (2.29). This way, a column of the element tangential stiffness matrix corresponding to the pivotal degree of freedom is obtained.

First, unit values corresponding to axial degrees of freedom v_{x1} and v_{x2} are imposed separately. Later, the strain values corresponding to these unit axial displacements are calculated at the integration points by using the strain-displacement matrix of the bar element. As shown in Figure 2.10 the strain distribution corresponding to axial displacements at element nodes is constant along the element length. Therefore, the sectional response for the imposed axial displacements at integration points can be obtained using the sectional rigidity matrix as follows.

$$N_x^1 = -\frac{1}{L}\frac{\Delta N_{xx}}{\Delta \varepsilon_{xx}} \tag{2.31}$$

$$M_y^1 = -\frac{1}{L}\frac{\Delta M_{yx}}{\Delta \varepsilon_{xx}} \tag{2.32}$$

$$M_z^1 = -\frac{1}{L}\frac{\Delta M_{zx}}{\Delta \varepsilon_{xx}} \tag{2.33}$$

$$N_x^2 = \frac{1}{L}\frac{\Delta N_{xx}}{\Delta \varepsilon_{xx}} \tag{2.34}$$

$$M_y^2 = \frac{1}{L}\frac{\Delta M_{yx}}{\Delta \varepsilon x_x} \tag{2.35}$$

$$M_z^2 = \frac{1}{L}\frac{\Delta M_{zx}}{\Delta \varepsilon_{xx}} \tag{2.36}$$

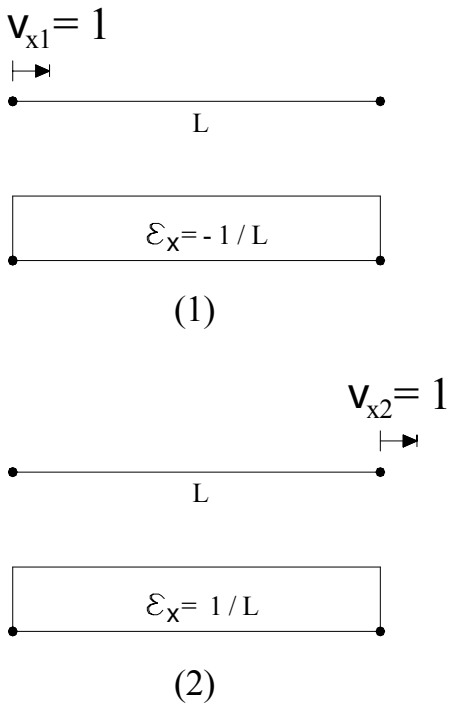

Figure 2.10: Strain distribution along frame element caused by unit axial displacement at element ends.

Equations (2.31) to (2.33) give the axial force and moment responses at the integration points for a given unit axial displacement at the start node of the element. Similarly, Equations (2.34) to (2.36) give the axial force and moment response at the integration points for a given unit axial displacement at the end node of the element.

This procedure is repeated by imposing a unit value for each of the remaining degrees of freedom v_{y1}, v_{y2}, θ_{z1}, θ_{z2}, v_{z1}, v_{z2}, θ_{y1} and θ_{y2}. Relevant equations are used for computing the member end forces, which are nothing but the nodal stiffness coefficients. Finally, these nodal stiffness coefficients are assembled into proper locations in the element tangential stiffness matrix.

2.3 Condensation and Constraint Equations

2.3.1 Condensation of the Frame Elements

The displacement field assumed over an element is approximate in the inelastic range. In order to improve the accuracy of the predicted displacement response in the inelastic range each frame element is further divided into a number of sub elements. This, in turn, increases the total number of unknowns and the number of equations to be solved. Consequently, the solution time increases drastically. Static condensation is used to eliminate the internal degrees of freedom and keep each frame element as a single macro element. This way, the total number of equations to be handled is kept at a reasonable level for the whole system.

In this study each frame element is subdivided into several finite elements. Later the internal degrees of freedoms are condensed out as shown in Figure 2.11. Here, the displacement components with the subscript *c* are those to be condensed out.

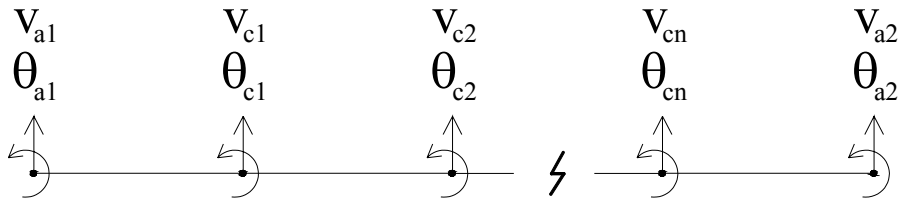

Figure 2.11: Condensation of the frame element into the end degree of freedoms

Condensed stiffness matrix and the member end forces are given below. In the following formulation the terms with the subscript *a* are those to be kept and the terms with the subscript *c* are those which are to be condensed.

$$\begin{bmatrix} K_{aa} & K_{ac} \\ K_{ca} & K_{cc} \end{bmatrix} \cdot \begin{bmatrix} U_a \\ U_c \end{bmatrix} = \begin{bmatrix} R_a \\ R_c \end{bmatrix} \tag{2.37}$$

$$K_{condensed} \cdot U_a = F_{condensed} \tag{2.38}$$

Where;

$$K_{condensed} = (K_{aa} - K_{ac} \cdot K_{cc}^{-1} \cdot K_{cc}) \tag{2.39}$$

$$F_{condensed} = R_a - K_{ac} \cdot K_{cc}^{-1} \cdot R_c \tag{2.40}$$

Response and current load state of a super element should also be condensed out since the nonlinear computations are performed with the structural degree of freedom. However, it is the residual forces that will be used for further calculations. Therefore, the residual forces are calculated at super element level and later they are condensed out.

It is more efficient to apply the coordinate transformation to frame super element after the condensation of the internal degrees of freedom. Less effective approach is transforming each beam finite element to global coordinates and later condensing the internal degrees of freedom. This will certainly take considerably more time than the former approach especially if the number of subdivisions in the super element is high. Moreover, in order to take the rigid zones into account one needs property matrices of the super element in local coordinates not in global ones.

2.3.2 Constraint Equations and Rigid Zones

The element degrees of freedom may be different from global structural degrees of freedom for several reasons. Two of them are considered in this study.

- Rigid End Zones: Rigid zones exist at member ends where beams and column are connected.

- Incompatibility between the centroid of the section and the material points on the structure where the global degrees of freedom are defined. Spread-of-plasticity formulation is performed by assuming the centroid of the section as the reference point. However, this may not, and usually is not, the material point on the structure where the global degrees of freedom are defined.

The element degrees of freedom at element end points 1 and 2 and the structural degrees of freedom defined at nodal points *m1* and *m2* on the structure are shown in Figure 2.12 for a typical element. If it is assumed that the centroid of the element's end section and the material point where the structural degrees of freedom are defined are both in the same rigid zone (this is usually the case) than, for a typical element, the constraint equations can be written as follows (Wilson [35]).

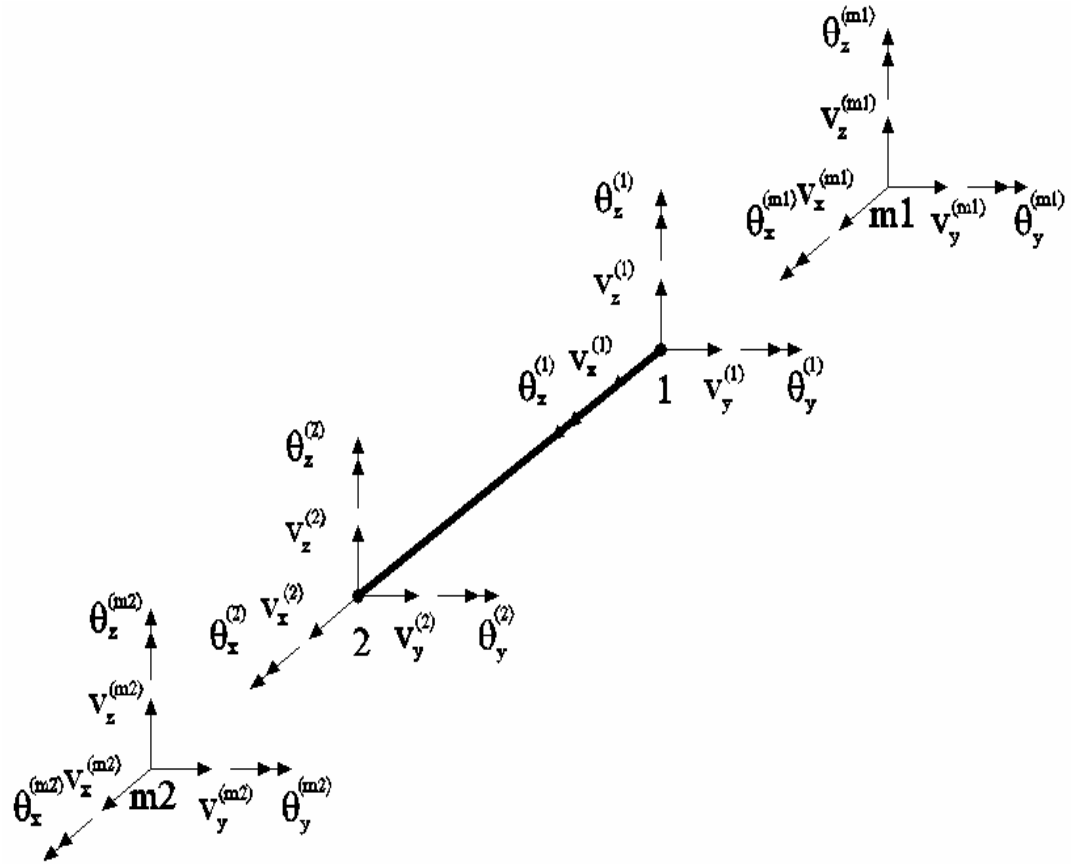

Figure 2.12 : Frame element's and global degree of freedoms

$$v_x^1 = v_x^{m1} + (z^1 - z^{m1})\theta_y^{m1} - (y^1 - y^{m1})\theta_z^{m1}$$

$$v_y^1 = v_y^{m1} - (z^1 - z^{m1})\theta_x^{m1} + (x^1 - x^{m1})\theta_z^{m1}$$

$$v_z^1 = v_z^{m1} + (y^1 - y^{m1})\theta_x^{m1} - (x^1 - x^{m1})\theta_y^{m1} \qquad (2.41)$$

$$\theta_x^1 = \theta_x^{m1}$$

$$\theta_y^1 = \theta_y^{m1}$$

$$\theta_z^1 = \theta_z^{m1}$$

$$v_x^2 = v_x^{m2} + (z^2 - z^{m2})\theta_y^{m2} - (y^2 - y^{m2})\theta_z^{m2}$$

$$v_y^2 = v_y^{m2} - (z^2 - z^{m2})\theta_x^{m2} + (x^2 - x^{m2})\theta_z^{m2}$$

$$v_z^2 = v_z^{m2} + (y^2 - y^{m2})\theta_x^{m2} - (x^2 - x^{m2})\theta_y^{m2} \tag{2.42}$$

$$\theta_x^2 = \theta_x^{m2}$$

$$\theta_y^2 = \theta_y^{m2}$$

$$\theta_z^2 = \theta_z^{m2}$$

Alternatively, these equations can be written in a matrix form as follows

$$u^s = T \cdot u^m \tag{2.43}$$

$$\begin{bmatrix} v_x^1 \\ v_y^1 \\ v_z^1 \\ \theta_x^1 \\ \theta_y^1 \\ \theta_z^1 \\ v_x^2 \\ v_y^2 \\ v_z^2 \\ \theta_x^2 \\ \theta_y^2 \\ \theta_z^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & z^1 - z^{m1} & y^{m1} - y^1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & z^{m1} - z^1 & 0 & x^1 - x^{m1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & y^1 - y^{m1} & x^{m1} - x^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & z^2 - z^{m2} & y^{m2} - y^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & z^{m2} - z^2 & 0 & x^1 - x^{m1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & y^2 - y^{m2} & x^{m2} - x^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x^{m1} \\ v_y^{m1} \\ v_z^{m1} \\ \theta_x^{m1} \\ \theta_y^{m1} \\ \theta_z^{m1} \\ v_x^{m2} \\ v_y^{m2} \\ v_z^{m2} \\ \theta_x^{m2} \\ \theta_y^{m2} \\ \theta_z^{m2} \end{bmatrix} \tag{2.44}$$

The element stiffness matrix and the load vector transformed to master nodes can then be expressed as.

$$K^m = T^T \cdot K \cdot T \tag{2.45}$$

$$R^m = T^T \cdot R \tag{2.46}$$

Constraint modifications are performed before the element matrices are transformed to global coordinate system. For the frame super-elements, the condensation of the

internal degrees of freedom is carried out first before the constraint equations are imposed on the remaining degrees of freedom.

After completing the analysis of the system the structural displacements need to be transformed back to the element local coordinate system. This is accomplished by using the Equation (2.43).

The procedure stated above assumes a rigid zone between the element ends and the points on the structure where the structural degrees of freedom are defined. However, the portion of the frame element that remains in this region is neither axially nor torsionally rigid. It has an axial and torsional rigidity similar to the rest of the element. To take this fact into account the tangential stiffness matrix constructed before the equation solution phase and the displacements that are back-substituted after analysis are further modified as follows.

Forming the element stiffness matrix using L shown in Figure 2.13 as the flexible length renders it stiffer then it actually is for axial deformations. To compensate for this extra stiffness the relevant terms of the stiffness matrix are multiplied by the modification factor given below.

$$c = 1 - \frac{dx1 + dx2}{L} \tag{2.47}$$

Figure 2.13: Member end offsets of the frame in axial direction

The terms of the element stiffness matrix that are modified with the coefficient c are shown below:

$$K^* = \begin{bmatrix} K_{1,1}\cdot c & K_{1,2}\cdot c & K_{1,3}\cdot c & K_{1,4}\cdot c & K_{1,5}\cdot c & K_{1,6}\cdot c & K_{1,7}\cdot c & K_{1,8}\cdot c & K_{1,9}\cdot c & K_{1,10}\cdot c & K_{1,11}\cdot c & K_{1,12}\cdot c \\ K_{2,1}\cdot c & & & & & & K_{2,7}\cdot c & & & & & \\ K_{3,1}\cdot c & & & & & & K_{3,7}\cdot c & & & & & \\ K_{4,1}\cdot c & & & & & & K_{4,7}\cdot c & & & & & \\ K_{5,1}\cdot c & & & & & & K_{5,7}\cdot c & & & & & \\ K_{6,1}\cdot c & & & & & & K_{6,7}\cdot c & & & & & \\ K_{7,1}\cdot c & K_{7,2}\cdot c & K_{7,3}\cdot c & K_{7,4}\cdot c & K_{7,5}\cdot c & K_{7,6}\cdot c & K_{7,7}\cdot c & K_{7,8}\cdot c & K_{7,9}\cdot c & K_{7,10}\cdot c & K_{7,11}\cdot c & K_{7,12}\cdot c \\ K_{8,1}\cdot c & & & & & & K_{8,7}\cdot c & & & & & \\ K_{9,1}\cdot c & & & & & & K_{9,7}\cdot c & & & & & \\ K_{10,1}\cdot c & & & & & & K_{10,7}\cdot c & & & & & \\ K_{11,1}\cdot c & & & & & & K_{11,7}\cdot c & & & & & \\ K_{12,1}\cdot c & & & & & & K_{12,7}\cdot c & & & & & \end{bmatrix} \tag{2.48}$$

Similarly, the axial strains along the element are overestimated since the flexible length of the element is taken smaller than it actually is. Therefore, the axial displacements at the member ends are modified as follows:

$$u^* = \begin{bmatrix} v_x^{m1}\cdot c \\ v_y^{m1} \\ v_z^{m1} \\ \theta_x^{m1} \\ \theta_y^{m1} \\ \theta_z^{m1} \\ v_x^{m2}\cdot c \\ v_y^{m2} \\ v_z^{m2} \\ \theta_x^{m2} \\ \theta_y^{m2} \\ \theta_z^{m2} \end{bmatrix} \tag{2.49}$$

The axial forces will remain the same, so there is no need to modify them.

A similar adjustment is also due for the torsional stiffness coefficients, displacements and loads. Since the material is assumed to be linear in torsion, the adjustment is accomplished through modifying the terms $T_{6,6}$ and $T_{12,12}$ in the transformation matrix of Equation (2.44) by multiplying them by a correction factor:

$$c^T = \sqrt{1 - \frac{dx1 + dx2}{L}} \tag{2.50}$$

2.4 Solution Algorithm

The nonlinear solution procedure is composed of many linear trial solutions of the system. First, the load increments are determined. Then, each load increment is applied on the system and a linear solution for the current loading level is performed by using the tangential stiffness matrix of the structure. The iterations are continued until the response of the structure becomes equal to the total loading currently on the structure.

In the elasto-plastic analysis of the structures the loading sequence changes the final response. The self-weight and other permanent loads must be applied on the structure first for a realistic solution. Therefore, the solution strategy should include a sequential loading scheme of the structure. It is also important to have an option to adjust the number of increments for each loading stage separately. Since it is usually unlikely to have serious plastification self-weight and other permanent loads, the analyst may want to step these sequences with less number of increments. The same goes for the earlier stages of lateral loads.

For the solution of the nonlinear structural equations the modified Newton Raphson algorithm (see Section 2.4.1) is preferred rather than the full Newton Raphson iterations However, since the material nonlinearity is modeled with the spread-of-plasticity approach which divides the cross section into many triangular monitoring elements, stiffness of the beam elements may vary sharply especially in the first iteration and when the yield state of the section changes. This leads to oscillations in the displacement increments and may lead the system to an incorrect solution. In some occasions load increments may fail to converge. To eliminate this short come, abnormal changes in the element's state are monitored and the solution algorithm is changed to full Newton Raphson when a numerical instability is detected. If the

same situation remains after changing the solution algorithm, than the solution for that load increment is accepted as correct.

2.4.1 Modified Newton Raphson Algorithm

In the modified Newton-Raphson algorithm (Figure 2.14) the stiffness of the structure is not updated for each linear solution. Instead, the same tangential stiffness is used for a certain number of iterations. After some number of fruitless iteration the stiffness matrix is updated to accelerate the convergence and reduce the solution time.

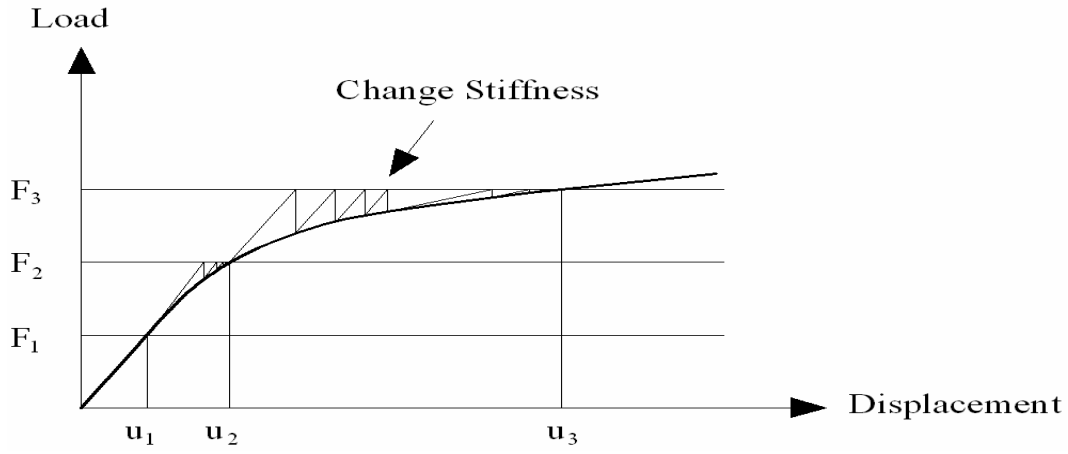

Figure 2.14: Modified Newton Raphson Algorithm

2.4.2 Solution Algorithm for Prescribed Displacements

In the case of displacement loading on the structure such as support settlements, a special solution algorithm is needed to convert these displacements to a set of nodal forces. When performing a linear solution, the nodal displacements can be converted to nodal forces as follows.

$$\begin{bmatrix} F_a \\ F_c \end{bmatrix} = \begin{bmatrix} K_{aa} & K_{ac} \\ K_{ca} & K_{cc} \end{bmatrix} \cdot \begin{bmatrix} U_a \\ U_c \end{bmatrix} \tag{2.51}$$

$$F_a = K_{aa} \cdot U_a + K_{ac} \cdot U_c \tag{2.52}$$

$$(F_a - K_{ac} \cdot U_c) = K_{aa} \cdot U_a \tag{2.53}$$

where U_c are the prescribed nodal displacements. However, for non-linear analysis the support displacements are assembled to the corresponding element's degrees of freedom incrementally and the system of equations is solved iteratively until no residual force remains on the degrees of freedom. To increase the numerical stability and decrease the number of iterations needed, the displacements U_a obtained from Equation (2.53) can be back-substituted into the corresponding element's nodes for each displacement increments U_c. While doing this it is best to use the tangential stiffness matrix at the current time and to take the load vector F_a as zero.

2.4.3 Mixed Element Technique

In some structures, e.g., steel structures, the element plastification occurs at certain locations of the structure. Moreover, it does not even occur up to a certain level of loading. It is obvious that with the use of Hermitian beam elements mesh refinement does not change the solution in the linear phase. Consequently, starting the computations with a fully discretized system leads to unnecessary intensity. Using the mixed element technique can alleviate this shortcoming. In the mixed element technique each structural frame member is modeled with a single frame finite element initially. After plastification occurs for an element, it is replaced with a super-element which consists of a certain number of frame finite elements. While replacing the element the displacement boundary conditions and the equilibrium should be satisfied for the substitute element. Unfortunately, this method is not suitable for reinforced concrete members since the plastification occurs at very low load levels (tensional cracking of concrete).

CHAPTER 3

DESIGN OF THE ANALYSIS PROGRAM

3.1 Design Environment

For the development of the analysis program C++ programming language is chosen. C++ is an object-oriented programming language. Object-oriented design has many advantages like sustainable development, easy design and reuse of the code. Additionally using the features of the object-oriented programming interface of the program can easily be defined. This provides a good framework for the libraries since the unnecessary details are hidden from the user.

Main features of the object-oriented design are: data abstraction, encapsulation, polymorphism and inheritance.

Real world problems are too complicated to be managed as whole. So the problem is divided into several objects and each object is considered separately. Objects are different from each other, however some have similarities. Abstraction is emphasis of these similarities by creating a class which has common attributes and procedures for a set of objects. Encapsulation means storing the data and the procedures under a single class. Additionally some procedures and data of the class can be protected from the outside use.

For the objects that have additional features that are not abstracted in the base class, a new class can be derived from the base class. In that situation derived class contains the procedures and members of the base class. This is called inheritance. By the help of inheritance only the additional features are implemented for an object since others are inherited from the base class. Moreover certain procedures of

the object may be different from the base class. Using polymorphism, procedures of the derived class can act differently by overriding existing procedures of the base class.

Components of nonlinear structural analysis can easily be represented by the objects. At the first glance elements, loads, structure and material models can be thought as main objects of the problem. Details of the design are given in the next chapter.

For the development tool Microsoft Visual C++ 6.0 is used. User interface of the analysis module is developed with OpenGL (Open Graphics Library) and MFC (Microsoft Foundation Classes).

OpenGL is a library for performing 3D graphical operations. It is a procedural library which can be used building graphical models using geometric primitives such as points, lines and polygons. For the details oen can check out the related reference.

MFC is a c++ library for windows programming. It includes classes for creating dialogs, views, frames etc. For the details about MFC you can check out the related reference.

3.2 Architecture

Program is developed in object-oriented manner; similar objects are grouped and derived from the same abstract class. Mainly a domain object is used to store elements, substructures, loads and boundary conditions of the main structure. This class is named as CStrSystem. Additionally functions those are necessary for nonlinear solutions such as forming tangential stiffness matrix, response calculations, back-substitution of the global displacements to the nodes of the elements are implemented. For the initialization of the class CStrSystem, an intermediate class called CSystemIntegrator is implemented. CSystemIntegrator gets definition of the structure either from an input file or some other appropriate

source. It creates instances of necessary classes such as elements, material properties of elements and assigns them to related container objects according to the definition of the structure. Additionally DOF numbering is performed by the class.

Each structural element is derived from an abstract base class named CSElements. An element class which is derived from CSElements must override member functions that computes tangential stiffness matrix of the element, converts distributed load to member end forces, calculates response of the element and assigns material model of the element.

Material properties of the elements are represented by the class CMaterialElements. Following classes are derived from this class:

1. CFrame3dLinear: Linear material model of 3D frame element.
2. CFrame3dNonLinear: Linear material model of 3D frame element. Two classes are derived from this class.
    a. CFrame3dNonLinearUncoupled: Moment response in local y-y direction, in local z-z direction and axial response are uncoupled.
    b. CFrame3dNonLinearCoupled: Moment response in local y-y direction, in local z-z direction and axial response are coupled.
3. CBar3dLinear: Linear material model of 3D bar element
4. CBar3dNonLinear: Non-linear Material model of 3D bar element.

Substructures or super elements are considered as special objects which have similarities both with CStrSystem and CSElements. They are similar to CStrSystem object since some number of elements forms a substructure. Nevertheless they are assembled to CStrSystem object to form the whole structure. Therefore they may also be treated as an element of the structure. Considering the situation, CSubStrSystem class is derived from both CSElements and CStrSystem. Assembly

of the substructure is handled by the functions of CSElements and the domain object CStrSystem stores the elements of the substructure.

Both CSElements and CStrSystem class definitions include a member class for handling the structural equations. This class is named CKFMath which contains the stiffness matrix, load vector and displacement vector of the structure/element. Additionally vectors that hold total load and response are stored by the class to be used for nonlinear analysis.

Constraint equations, static condensation and transformation of the element matrices from local coordinates to global coordinates are all categorized as modifications to the structural equations. Accordingly the classes that are related to these operations are derived from the same abstract base class which is called CModifyEquations. Classes that are derived from CModifyEquations must override functions which perform modifications of stiffness matrix and load vector. Additionally they must override the function that back-substitutes modified displacements to original displacements. After creating an instance of class which is derived form CModifyEquations, it must be assigned to a corresponding CKFMath class which stores the structural/element stiffness matrix and load vector. More than one CModifyEquations object can be stacked on each other. In this case the modifications will be performed starting with the first CModifyEquations object that is assigned to the CKFMath class. Three classes that are derived from CModifyEquations are listed above.

1. CTransformation: Modifies the structural equations for transformation of degree of freedoms from one coordinate system to another.

2. CCondensation: Modifies the structural equations for static condensation of certain degree of freedoms.

3. CConstraint: Modifies the structural equations for degree of freedoms constrained with each other.

Loading of the system is stored in a class which is called CLoadSystem. Load cases are contained by this base class. Each load case is represented by CLoadCases class. A load case consists of two types of loads. First one is nodal forces and the other one is element loads. There are also two types of element loading: distributed load and point load. As stated before it is responsibility of the element to convert loads on it to fixed end forces.

For sequential loading of the structure CSequentialLoading class is derived from CLoadSystem. This class stores the load cases at each sequence and their weights. It is also responsible for carrying the system to next load increment. It is performed by stepping the nodal loads and element forces to next load increment.

In nonlinear solution, history of the system should be stored for future use. CSystemStorage class is implemented to store the system instances at the end of each load increment. It stores the CSystemInstances classes. Each CSystemInstances class instance is a copy of the CStrSystem class at a certain stage of solution. At the end of the nonlinear solution the amount of CSystemInstances classes that are stored in CSystemStorage class will be equal to total number of load increments.

For geometric operations 3D point and vector classes are implemented. Two classes works interactively by operators working with each other.

Matrix operations are performed with an available matrix library. The library is named as TBCI NumLib. Mainly it contains vector and matrix classes and related operations. It has also special types of matrices like band matrix. An LU solver is implemented in the library that works with the band matrices. Details about the library can be reached from the related reference. Some extensions were added to the library such as sub-matrix operations. Duration of the solution is calculated with an open source duration class CDuration.

For the graphical representation of the system following classes are used. CDrawMain class initializes the OpenGL view for a given window handle. It also draws the drawing objects it includes for each refreshments of the view. Drawing

objects are represented by the abstract base class DOMain. Each drawing object that is derived from DOMain should implement its own drawing on the screen.

3.2.1 Class Hierarchy & Relationship between Classes.

Classes in the analysis program and their relationships are shown in Figure 3.1. Rumbaugh notation (Rimbaugh et al. [17]) is used for the relationships of the classes. Classes are represented inside rectangles in Rumbaugh notation. Mckenna FT [16] summarizes the relationships of the classes in object oriented design and their representations in Rumbaugh notation as follows:

- **knows-a**: Object of the one class knows about the object of the another class. It is shown by a line connecting rectangles.

- **is-a:** This term is used for a class that is derived from a parent class. It is represented by a line with a triangle between the classes.

- **has-a:** has-a relation is for a class which consists of a number of other classes. It is represented by a diamond in the aggregate class and line from the diamond to the component classes.

Using above notations a class diagram similar to the one drawn by Mckenna [16] is formed as given in Figure 3.1. Similarly relationship between the classes CSElements, CStrSystem CKFMath and CModifyEquations are drawn in Figure 3.2.

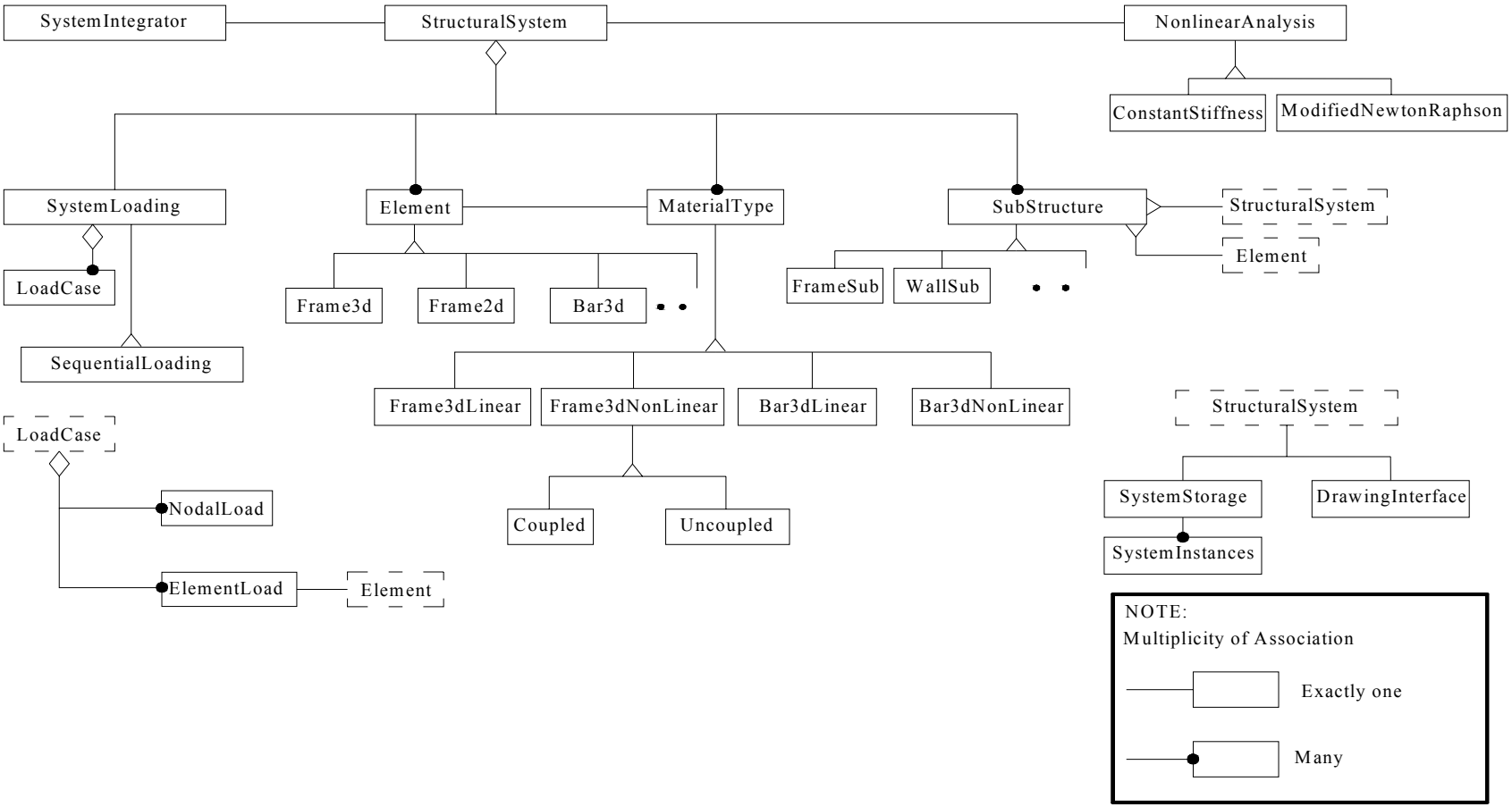

Figure 3.1 : Class Diagram of the Nonlinear Analysis Program

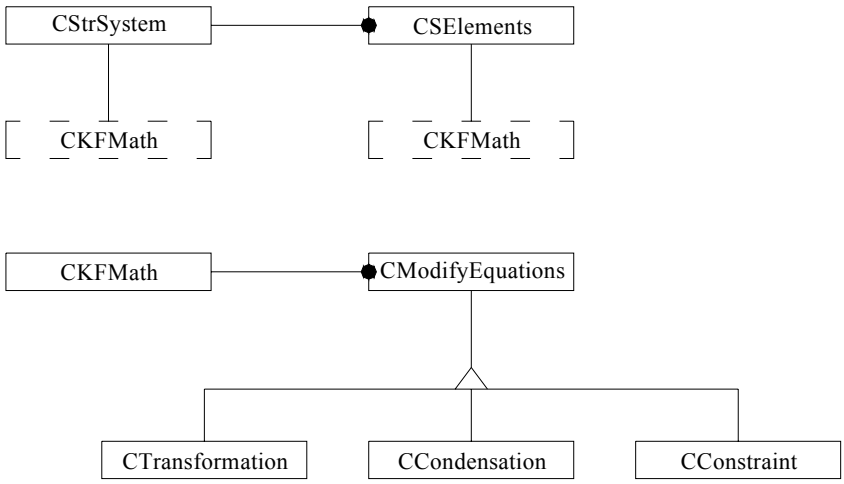

Figure 3.2: Relationship between classes CStrSystem, CSElements, CKFMath and CModifyEquations

3.2.2 Class Members and Procedures

Classes implemented in the program are listed here with their procedures and members. Basic definition is also given for each class. In members and procedures, items apart from those defined in the inherited class, are listed. Procedures that are listed under the name of pure virtual functions are needed to be defined for each derived class. If not the class cannot be instantiated.

3.2.2.1 CStrSystem

Definition:

It is the domain object that stores everything about the structure. Only a single instance of this class is created during execution of the program.

Inherits From:

None

Members:

- Array storing CSElements class instances
- Array storing CSubStrSystem class instances
- Array storing CMaterialElements class instances
- CLoadSystem class instance
- CSystemStorage class instance
- CSystemIntegrator class instance
- CKFMath class instance for structural equations.

Main Procedures:

- Adds element/substructure to domain.
- Queries an element/substructure of the domain.
- Defines a new material property for the domain
- Sets/queries the loading of the system
- Back-substitutes the displacement to the elements
- Calculates the response of the system for the current displacement increment
- Steps the loading of the system to next load increment
- Forms initial and tangential stiffness matrix of the whole structure
- Solves the system for the current loading state
- Writes structural, load, displacement matrices to a specified file

3.2.2.2 CSystemIntegrator

Definition:

This class forms the system according to a valid system definition. Definition may be done by file input or setting the variables of this class directly. The job of this class ends after initialization of the system which is creation of the elements and their material models and performing other necessary initializations before analysis. Although CStrSystem includes a pointer to this class instance, it is independent of this class.

Inherits From:

None

Members:

- Array storing the coordinates of the nodes.
- Array storing the elements' connectivity.
- Array storing local y-axes of the 3D frames.
- Array storing the element types used in the structure.
- Array storing the material model for each element type used in the structure.
- Array storing the nodal loads of the structure.
- Array storing the element loads of the structure.
- Array storing the support conditions of the structure.
- Array storing the degree of freedoms.
- CStrSystem class instance.

Main Procedures:

- Reads input file and fills the member variables of the class.
- Forms degree of freedoms of the class according to the active degree of freedoms.
- Creates CStrSystem class instance, creates the elements which are used in structure and assigns them to the structure.
- Creates the load cases and assigns them to the structure.

3.2.2.3 CSElements

Definition:

This is abstract base class for structural elements. Since it is abstract, class cannot be instantiated. Structural elements should be derived from this class.

Inherits From:

None

Members:

- CKFMath class instance for structural equations
- Coordinates of the element
- Assembly information of the element
- Connectivity of the element
- CMaterialElements class instance which stores the material properties of the element
- Element specific information such as degree of freedoms, node number, degree of freedom per node etc...

- Latest displacement of the element

Main Procedures:

Pure Virtual Functions

Pure virtual functions are one of the key concept of abstraction. Followings are the list of pure virtual procedures that must to be implemented in each derived class. The operation that needs to be performed in these functions is summarized below.

- Computation of the response of the element for a given displacement field
- Computation of the tangential stiffness matrix of the element
- Computation of the fixed end forces for a given distributed load

Other Procedures:

- Sets/queries the global coordinates of the element.
- Sets/queries the connectivity of the element.
- Assembles the stiffness matrix or load vector of the element to global stiffness matrix and global load vector according to the assembly info of the element.
- Assembles the global displacement vector back to element.
- Calculates the residual forces on the element and sets them to the next load increment.
- Computes response by calling related pure virtual function and performing common necessary operations for element (such as modification in load vector due to transformation, condensation etc…).
- Computes tangent stiffness matrix by calling related pure virtual function of the element and performing common necessary operations for element (such

as modification in stiffness matrix due to transformation, condensation etc…).

3.2.2.4 CBar3d

Definition:

This class represents 3D bar element.

Inherits From:

CSElements

Members:

- Vector storing the local x axis of the bar element.
- CTransformation class instance for coordinate transformation of the bar element from local coordinates to global ones.

Main Procedures:

- Computes the tangential stiffness matrix of the bar element.
- Computes the response of the bar element.
- Calculates the local axis of the bar element according to its coordinates

3.2.2.5 CFrame3d

Definition:

This class represents 3D frame element.

Inherits from:

CSElements

Members:

- Vectors storing local x, y and z axis.
- CTransformation class instance for coordinate transformation of the beam element from local coordinates to global ones.

Main Procedures:

- Sets the direction of the local y axis of the beam element.
- Computes the tangential stiffness matrix of the beam element.
- Computes the response of the beam element.
- Calculates the local axis of the beam element according to its coordinates.
- Calculates the shape function of the beam element for a given local coordinate.
- Calculates the strain-displacement (B) matrix for a given local coordinate.
- Calculates the curvatures of the beam element for a given displacement field.

3.2.2.6 CSubStrSystem

Definition:

This is abstract base class for substructures. It has two parents which are CSElements and CStrSystem. This approach is not very common and it is arguable. Later this class may be modified like deriving it from one of them or neither of them.

Inherits From:

CSElements and CStrSystem

Main Procedures:

Pure Virtual Functions

The operation that needs to be performed in these functions is summarized below.

- Generation of internal elements of the substructure. Creating instances of the internal elements, performing the necessary initializations, and also assigning static condensation if there exist.

Other Procedures:

- Back-substitutes the displacements to the elements in substructure.
- Calculates the residual force and sets it as the next load increment.
- Assembles the stiffness matrix and load vector of the substructure to global stiffness matrix according to the assembly info of the substructure.

3.2.2.7 CFrameSubStr

Definition:

It is 3D frame super element. It may also be called as super element. This class is initialized similar to a single frame element in system integrator. Various frame super elements can be instantiated using this class. The class can easily be extended to have a custom frame super element.

Inherits From:

CSubStrSystem

Members:

- Vector storing local y axis of the substructure.
- Integer storing the type of the substructure.

Main Procedures:

- Generates the internal elements of the substructure.
- Sets the orientation of the substructure by defining the local y axis.

3.2.2.8 CWallSubStr

Definition:

This class models the infill walls of the structure. It is composed of two diagonal bars which only work in compression.

Inherits From:

CSubStrSystem

Members:

- Array storing the bounding frame elements.
- Material properties of the wall such as modulus of elasticity in horizontal and vertical directions and prism strength of the wall.
- Thickness of the wall.

Main Procedures:

- Generates the internal elements of the substructure.

3.2.2.9 CLoadSystem

Definition:

Loading information of the system is stored in this class. This class basically stores the load cases of the system. Special loading types (i.e. sequential loading) can be derived from this class.

Inherits From:

None

Members:

- Array storing load case instances

Procedures:

- Adds load case to the class.
- Queries load cases.

3.2.2.10 CSequentialLoading

Definition:

This class is used for sequential loading of the structure. It is responsible for assigning load vector of the structural system according to the loading sequence.

Inherits From:

CLoadSystem

Members:

- Array storing load vector for each sequence
- Array storing the number of increments for each sequence.
- Array storing load cases and their weights for each sequence.

Procedures:

- Adds a sequence of loading for given load cases and their weights.
- Forms the load vector of the structure for the next increment of loading.

- Steps back the structure to the previous load increment.

3.2.2.11 CLoadCases

Definition:

This class stores a load case for the structure.

Members:

- Nodal load vector.
- Nodal displacement vector.
- Distributed and point loads of the elements.

3.2.2.12 CKFMath

Definition:

Structural stiffness matrix, load vector and displacements are stored here. Additionally total load and response records are kept here for nonlinear solution. There is an instance of this class in each CSElements and CStrSystem classes. Modification in structural equations such as transformation of coordinates or condensation can be assigned to the class.

Inherits From:

None

Members:

- Stiffness matrix.
- Modified stiffness matrix.
- Load vector.

- Modified load vector.
- Displacements.
- Modified displacements.
- Response vector.
- Total load vector.
- Array storing CModifyEquations class instances.

Procedures:

- Sets number of degree of freedoms. Allocates memory for structural matrices stored in the class.
- Queries number of degree of freedoms.
- Solves the current system.
- Assigns next loading increment to total load minus response vector.
- Modifies the stiffness matrix according to the CModifyEquations class instances.
- Modifies the load vector according to the CModifyEquations class instances.
- Back-substitutes the displacement vector according to the CModifyEquations class instances.

3.2.2.13 CModifyEquations

Definition:

This is abstract base class for the modifications in structural equations. Derived classes should implement procedures of the class which modifies the stiffness matrix, load vector and displacements.

Inherits From:

None

Members:

- Pointer to the unmodified stiffness matrix.
- Pointer to the modified stiffness matrix
- Pointer to the load vector.
- Pointer to the modified load vector.
- Pointer to the displacement vector.
- Pointer to the modified displacement vector.
- Original stiffness matrix stored after modification is applied.
- Original load vector stored after modification is applied.

Main Procedures:

Pure Virtual Functions

The operation that needs to be performed in these functions is summarized below.

- Modifying the stiffness matrix.
- Modifying the load vector.
- Back-substituting the displacement vector.

Other Procedures:

- Assigns pointer to the unmodified and modified stiffness matrix to related class member.

- Assigns pointer to the unmodified and modified load vector to related class member.

- Assigns pointer to the unmodified and modified load vector to related class member.

3.2.2.14 CTransformation

Definition:

This class handles transformation of degree of freedoms from one coordinate axis to another (generally from local axis to global).

Inherits From:

CModifyEquations

Main Procedures:

- Modifies the stiffness matrix.

- Modifies the load vector.

- Back-substitutes the load vector.

- Sets transformation properties from three base vectors. This procedure is used for 3D frame elements.

- Sets the transformation properties from one vector. This procedure is used for bar elements.

3.2.2.15 CCondensation

Definition:

This class handles condensation of certain degree of freedoms. It is used in substructures and super elements.

Inherits From:

CModifyEquations

Members:

- Number of degree of freedoms to be retained.
- Inverse of the stiffness matrix.
- Sub-matrices to be used for condensation operation.

Main Procedures:

- Modifies the stiffness matrix.
- Modifies the load vector.
- Back-substitutes the load vector.
- Sets number of degree of freedoms to be retained after condensation.

3.2.2.16 CConstraint

Definition:

This class is used for rigid zone constraints of the frame elements**.**

Inherits From:

CModifyEquations

Main Procedures:

- Modifies the stiffness matrix.
- Modifies the load vector.
- Back-substitutes the load vector.

- Forms the constraint equations by taking four points' coordinates as input.

3.2.2.17 CMaterialElements

Definition:

This is base class for the material properties of the elements.

Inherits From:

None

Members:

- Array of pointer to the elements whose material properties are defined by this class.

Procedures:

- Stores the pointer of the CSElements class instance.

3.2.2.18 CFrame3dLinear

Definition:

Class defines linear material properties of a 3D frame element.

Inherits From:

CMaterialElements

Members:

- Modulus of elasticity of the frame.
- Moment of inertia about local y-y axis.
- Moment of inertia about local z-z axis.

- Area of the cross section
- Shear modulus of cross section.
- Torsional constant for cross section.

3.2.2.19 CFrame3dNonlinear

Definition:

This is abstract base class that defines nonlinear material properties of a 3D frame element.

Inherits From:

CMaterialElements

Members:

- Initial modulus of elasticity of the frame.
- Initial moment of inertia about local y-y axis.
- Initial moment of inertia about local z-z axis.
- Area of the cross section
- Shear modulus of cross section.
- Torsional constant for cross section.

Main Procedures:

Pure Virtual Functions

The operation that needs to be performed in these functions is summarized below.

- Calculating response according to a given strain filed.

- Calculating tangential stiffness matrix according to a given strain filed.

3.2.2.20 CFrame3dNonLinearUncoupled

Definition:

Response and the tangential stiffness matrix are calculated by assuming moment about y-y axis, moment about z-z and axial force are uncoupled. Therefore relation between curvature and moment about y-y axis and z-z axis should be defined separately. Additionally a linear relationship between axial strains and stresses is assumed. This model can predict the response of the beam elements accurately. However in columns coupled material model should be used since there is significant amount of axial force in columns. In addition to that, under certain load conditions, bi-axial bending occurs.

Inherits From:

CFrame3dNonlinear

Members:

- Array storing the negative moment values in y-y direction for specified strain increments.
- Array storing the positive moment values in y-y direction for specified strain increments.
- Array storing the negative moment values in z-z direction for specified strain increments.
- Array storing the positive moment values in z-z direction for specified strain increments.

Procedures:

- Calculates response according to a given strain filed.

- Calculates tangential stiffness matrix according to a given strain filed given.
- Reads nonlinear material properties from an input file.
- Writes nonlinear material properties to an output file.

3.2.2.21 CFrame3dNonLinearCoupled

Definition:

Response and the tangential stiffness matrix are calculated by coupling moment about y-y axis, moment about z-z and axial force. This material model gives more realistic results. However it is computationally more intensive. Additionally whole system needs more iteration to converge for a load increment when structure is composed of elements with this material type. First cross section is meshed using a FORTRAN dynamic link library. Meshed section is stored in the class to prevent re-meshing. Response and stiffness properties of the section are computed using the meshed section stored in the class again by using FORTRAN dynamic link library.

Inherits From:

CFrame3dNonlinear

Members:

- Various variables storing the cross section properties and discretized cross section.

Procedures:

- Calculates response according to a given strain filed.
- Calculates tangential stiffness matrix according to a given strain filed given.
- Reads section geometric and material properties from an input file.
- Writes section geometric and material properties to an output file.

3.2.2.22 CBar3dLinear

Definition:

This class stores the linear properties of the 3D bar properties.

Inherits From:

CMaterialElements

Members:

- Modulus of elasticity of the bar.
- Cross sectional area of the bar.

3.2.2.23 CBar3dNonLinear

Definition:

This class is for nonlinear 3D bar element with multi-linear material model.

Inherits From:

CMaterialElements

Members:

- Array storing modulus of elasticity for certain strain levels.
- Array storing strain levels at which hardening begins.
- Array storing stress values correspond to the strain levels stated above.

Main Procedures:

- Defines material model by giving stress value and modulus of elasticity.
- Calculates the stress for a given strain.

- Calculates the tangent modulus of elasticity for a given strain state.

3.2.2.24 CNonLinSol

Definition:

This class is the abstract base class for the nonlinear solution algorithms. It has a pure virtual function ‘solve’ that needs to be overridden in each derived class. In addition to that class includes some common procedures for each nonlinear solution algorithm.

Inherits From:

None

Members:

- CStrSystem class instance.
- Error tolerance of the solution in percentage.
- CDuration class instance for determining the solution time.

Main Procedures:

Pure Virtual Functions

The operation that needs to be performed in these functions is summarized below.

- Solving the system that is stored as member variable of the class

Other Procedures:

- Check whether the system is converged according to the given load.
- Check whether the system is converged according to the given displacements.

- Starts the timer of CDuration.
- Stops the timer of CDuration
- Reports the iterations.
- Reports the displacements after each convergence.

3.2.2.25 CConstantStiffness

Definition:

This class solves the system with Modified Newton Rapshon algorithm. It uses the initial stiffness matrix throughout the solution.

Inherits From:

CNonLinSol

Procedures:

- Solves the system.

3.2.2.26 CModifiedNewtonRapshon

Definition:

It starts solution with initial stiffness matrix. After a number of iterations that are not converged the stiffness matrix is updated. Later the same stiffness is used until iterative solution does not give satisfactory displacement increments.

Inherits From:

CNonLinSol

Procedures:

- Solves the system.

3.2.2.27 CModifiedNewtonRapsonWC_1

Definition:

Implementation of the class is the same with the CModifiedNewtonRapshon except this one changes the algorithm to Full-Newton Raphson when solution converges to an unrealistic result.

Inherits From:

CNonLinSol

Procedures:

- Solves the system.

3.2.2.28 CModifiedNewtonRapsonDisplacement

Definition:

This class is capable of solving a system which is under the effect of pre-described displacement.

Inherits From:

CNonLinSol

Procedures:

- Solves the system.

3.2.2.29 CSystemStorage

Definition:

This class keeps the record of the structural system at each increment of the load. It mainly includes an array of CSystemInstances objects.

Inherits From:

None

Members:

- Array of pointer to the CSystemInstances class instances.
- Pointer to the CStrSystem class instance.

Procedures:

- Saves the current state of the system.
- Restores the system at the given index.

3.2.2.30 CSystemInstances

Definition:

This class stores the system, particularly its elements. State of the elements are stored by copying the elements.

Inherits From:

None

Members:

- Array of pointer to the CSElements class instances.

CHAPTER 4

CASE STUDIES

4.1 Numerical Test Problems

The validity and the effectiveness of the proposed algorithm are tested by a number of numerical test problems. The performance of the numerical scheme is influenced by a number of parameters that can affect the predicted response directly. Mesh density of the cross section, the number of integration points taken along the beam, discretization of the frame element along its length are some of them. First, the program results are verified for linear and nonlinear cases. Later, the parameters of the solution algorithm are investigated to find their optimum values. Especially, a frame super-element that expected to represent the behavior of the structure within an acceptable level of accuracy is investigated. Finally, the structural response predicted by the program is compared with the experimental results available.

4.1.1 Linear Case

To verify the results produced by the program for linear structures, a one-bay single-story space frame structure shown in Figure 4.1 is used. All beams and the columns in the example structure have identical geometric and material properties. The structural analysis program SAP2000 v8.2.3 is used for comparison.

Material and geometric properties of the frame elements in some consistent units are as follows. $E = 2.0x10^{7}$, $I_{yy} = 2.0x10^{-4}$, $I_{zz} = 2.0x10^{-3}$, $J = 1.0x10^{-4}$, $A = 2.0x10^{-2}$ and $v = 0.2$. The load is applied at *node 2* in global x-direction is F= 5000.

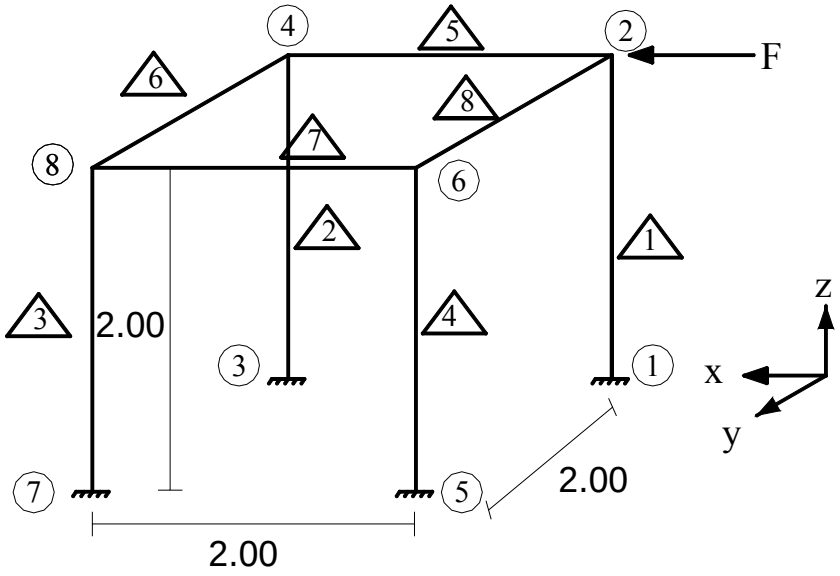

Figure 4.1: Geometry of the sample frame (in some suitable units)

The load is applied in a number of steps. Since the material models of the frame elements are linear, the solution is expected to converge in a single iteration for each load increment. This is observed to be the case. The final displacement response of the structure is given in Table 4.1 together with the response predicted by SAP2000.

Table 4.1 : Displacements given by the program and SAP2000

Node Number	DOF	Displacements (Code Developed)	Displacements SAP2000
	Ux	0,070322	0,070322
	Uy	-0,013354	-0,013354
	Uz	0,008911	0,008911
node:2	θx	-0,003083	-0,003083
	θy	0,027544	0,027544
	θz	0,023681	0,023681
	ux	0,058741	0,058741
	uy	0,013354	0,013354
	uz	-0,008911	-0,008911
node:4	θx	0,003083	0,003083
	θy	0,021773	0,021773
	θz	0,019465	0,019465
	ux	0,003474	0,003474
	uy	-0,013354	-0,013354
node:6	uz	0,000957	0,000957
	θx	-0,003083	-0,003083
	θy	0,001679	0,001679
	θz	0,023681	0,023681
	ux	0,003428	0,003428
	uy	0,013354	0,013354
node:8	uz	-0,000957	-0,000957
	θx	0,003083	0,003083
	θy	0,001636	0,001636
	θz	0,019465	0,019465

Table 4.1 shows that, as expected, the program gives the same results as SAP2000 for linear structures.

Later, another analysis is performed by considering the rigid zones of length 0.1 at the member ends of the beams and top of the columns. The results are listed in Table 4.2.

Table 4.2: Displacements given by the program and SAP200 when rigid zones are taken into account

Node Number	DOF	Displacements (Code Developed)	Displacements SAP2000
	ux	0,062692	0,062692
	uy	-0,012518	-0,012518
	uz	0,009509	0,009509
node:2	θx	-0,003408	-0,003408
	θy	0,024799	0,024799
	θz	0,021574	0,021573
	ux	0,051174	0,051174
	uy	0,012518	0,012518
	uz	-0,009509	-0,009509
node:4	θx	0,003408	0,003408
	θy	0,018884	0,018884
	θz	0,017148	0,017148
	ux	0,003572	0,003572
	uy	-0,012518	-0,012518
node:6	uz	0,001135	0,001135
	θx	-0,003408	-0,003408
	θy	0,001756	0,001756
	θz	0,021574	0,021573
	ux	0,003519	0,003519
	uy	0,012518	0,012518
node:8	uz	-0,001135	-0,001135
	θx	0,003408	0,003408
	θy	0,001712	0,001712
	θz	0,017148	0,017148

From Table 4.2 it is clear that the results produced by the program developed are in perfect agreement with those given by SAP2000 when the rigid zones at element ends are considered.

4.1.2 Nonlinear Case

4.1.2.1 Coupled Material Model for Frame Elements

Coupled material model of the frame element is verified using a simply supported beam structure. The geometry and the cross-sectional details of the structure are given in Figure 4.2. The beam is modeled by a single frame element. The concentrated end moments are applied incrementally at both ends to create a single curvature along the element. The curvature values at each load increment are recorded and they are compared with the moment-curvature diagram of the element cross-section. The moment-curvature diagram of the reinforced concrete section is obtained by the code RCMOD given by Polat [36]. A modified version of the same code is used for the material model of the frame element in this study. Therefore, the results obtained from RCMOD and the program developed in this study is expected to match. The analyses are later repeated in the presence of an axial load. The results of the analyses are presented in Figure 4.3, Figure 4.4, and Figure 4.5.

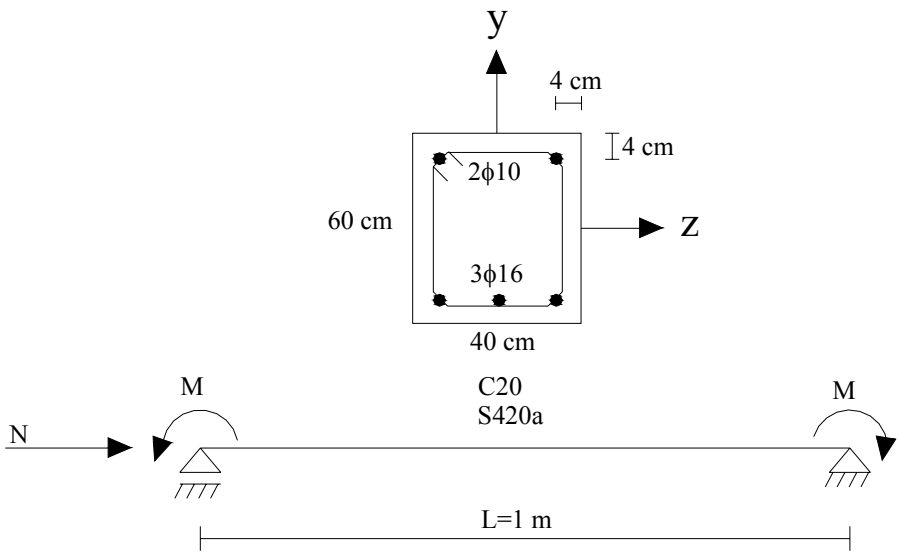

Figure 4.2: Test of the material model of the beam element under constant moment and axial load

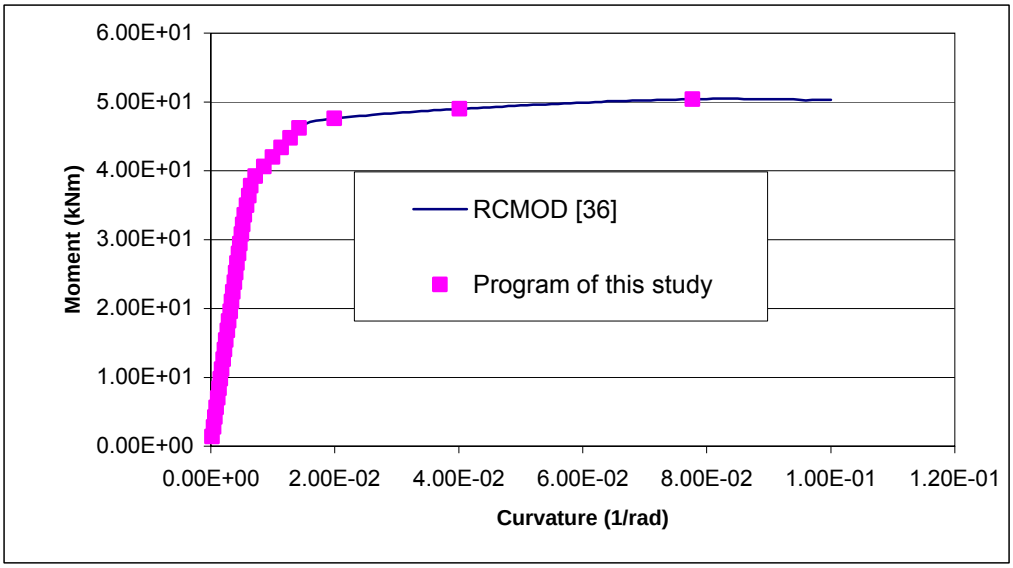

Figure 4.3: Moment-curvature relationship about *y axis* produced by the program of this study and RCMOD [36]

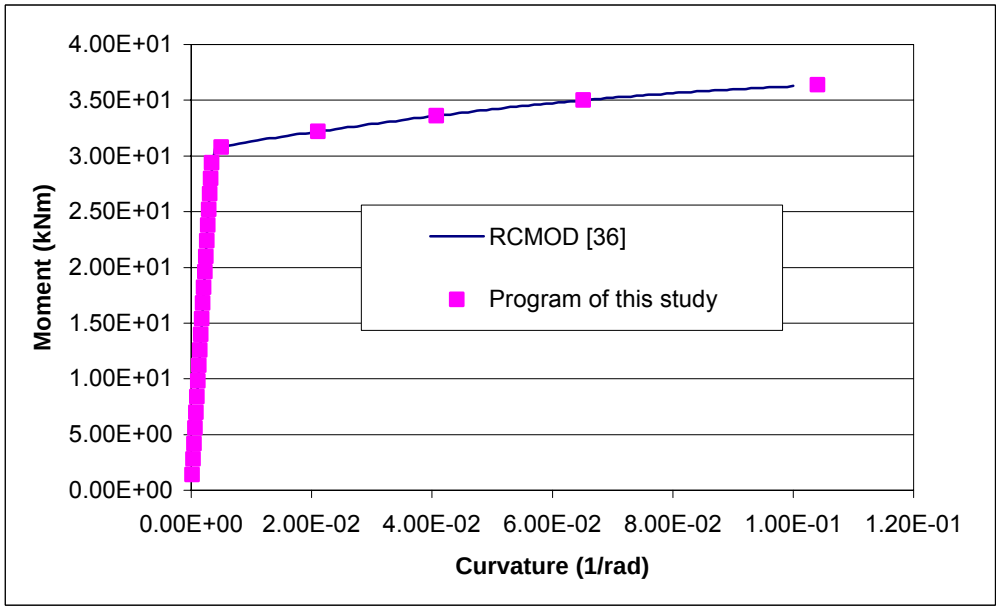

Figure 4.4: Moment-curvature relationship about *positive z* axis produced by the program of this study and RCMOD [36]

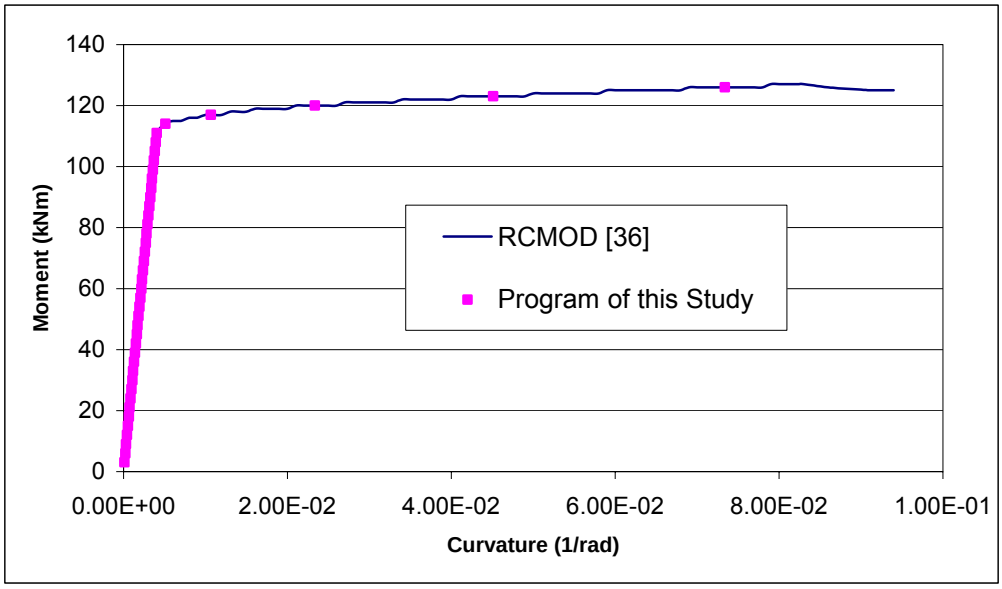

Figure 4.5: Moment curvature relationship about *negative* z axis produced by the program of this study and RCMOD [36]

The same analysis is repeated in the presence of an axial load of *1600 kN* to test the material model under a more practical situation. The axial load is applied to the beam as the first sequence of loading and then the end moments are applied. The results are presented below.

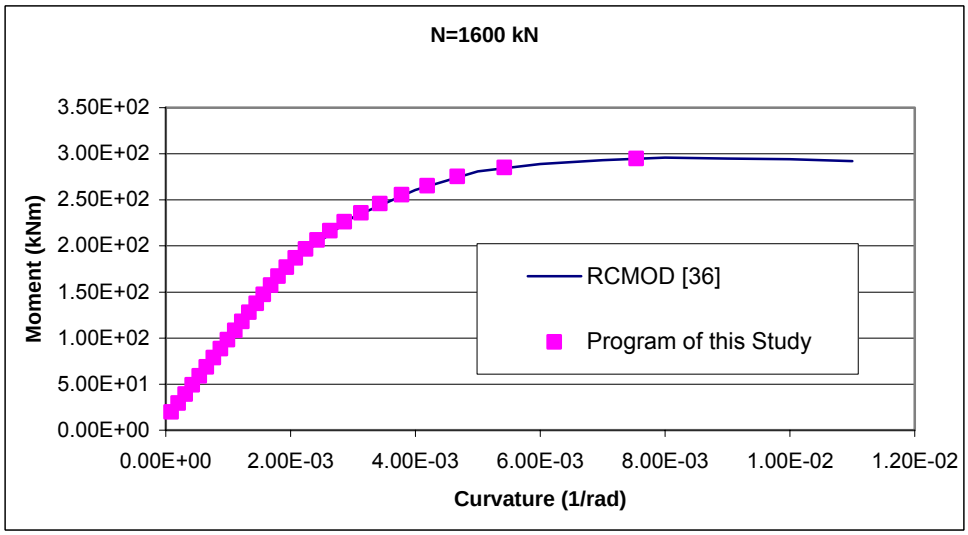

Figure 4.6: Moment-curvature relationship about *negative z* axis under axial load produced by the program of this study and RCMOD [36]

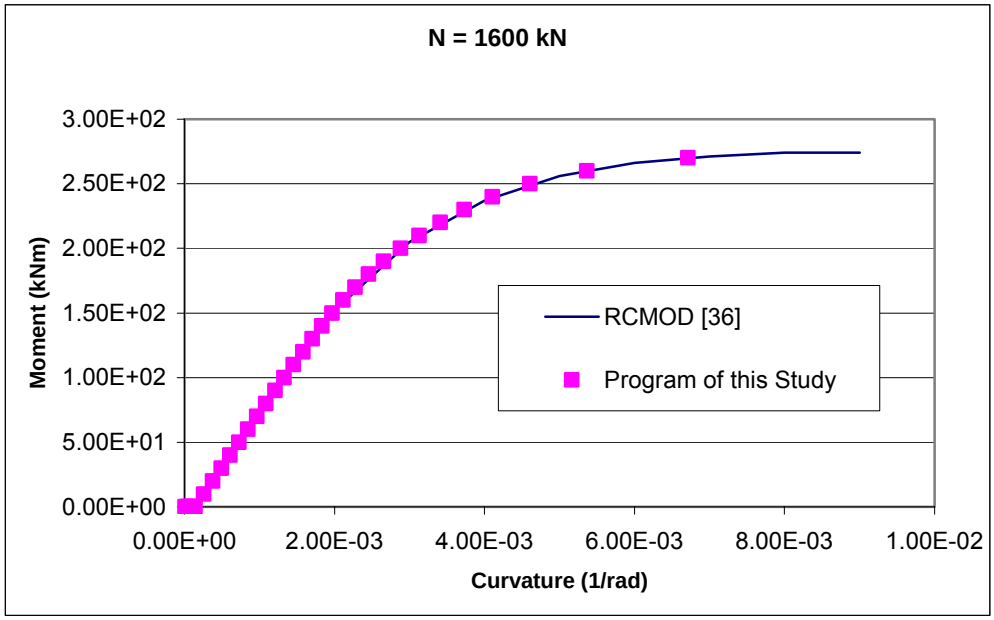

Figure 4.7: Moment-curvature relationship about *negative z* axis under axial load produced by the program of this study and RCMOD [36]

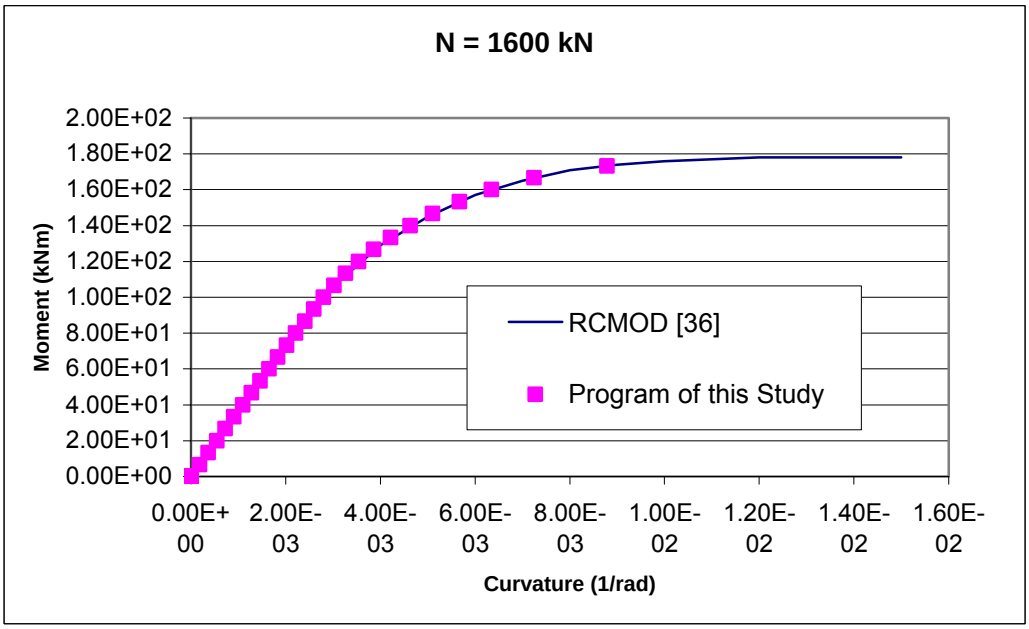

Figure 4.8: Moment-curvature relationship about *y* axis under axial load produced by the program of this study and RCMOD [36]

4.1.3 Efficiency and Effectiveness of the Frame Element

4.1.3.1 Influence of the Number of Integration Points

Gauss-Legendre integration is used for numerical integration of the frame elements. With *n* Gauss integration points a polynomial function of order *2n-1* can be integrated exactly. In the linear case using 2 Gauss integration points is sufficient to obtain the exact results since the *B* matrix and the curvature along the beam varies linearly. Consequently, the expressions such as $B^T \cdot EI \cdot B$ *and* $B^T \cdot M$ are second degree polynomials. However, when the material model of the beam element is nonlinear using a larger number of integration points is normally needed since the order of the expressions increases. However, when the spread-of-plasticity approach is adopted as the material model, it is not possible to specify the exact order of integration since the exact order of the integrant expressions is not clear. Therefore, the effect of the number of integration points to be used for the numerical integration is investigated next.

A cantilever column shown in Figure 4.9 is used as a test structure to explore the effect of the number of integration points (integration order) on the predicted response. The cross-sectional details of the structure are as given in Figure 4.2. The lateral tip displacements calculated for a lateral load of F=*12.74 kN,* which is very close to its ultimate value, are reported in Table 4.3. From Table 4.3 it is clear that increasing the number of Gauss integration points used do not increase the accuracy of the solution significantly. Nevertheless, using fewer numbers of Gauss integration points decreases the solution time drastically.

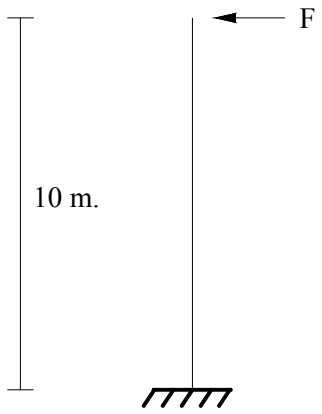

Figure 4.9: Cantilever column used to test the effect of the number of Gauss points used on the solution accuracy.

Table 4.3: Lateral tip deflection for different number of Gauss integration points used for computations

Number of Beam Elements	Number of Gaussian Points			
	2	3	4	5
1	0.13801	0.139351	0.139186	0.139117
3	0.152701	0.15274	0.152736	0.152752
5	0.186641	0.186599	0.186686	0.186644
20	0.448504	0.448502	0.448505	0.448502

The effect of the frame super-element mesh density on its predictive capacity is investigated with the test structure shown in Figure 4.10. The cross-sectional details of the structure are again as given in Figure 4.2. The structure is analyzed using different meshing schemes for the frame super-elements representing the beam and the columns. The load is applied until total collapse of the structure. The results are given in Figure 4.11, Figure 4.12 and Figure 4.13.

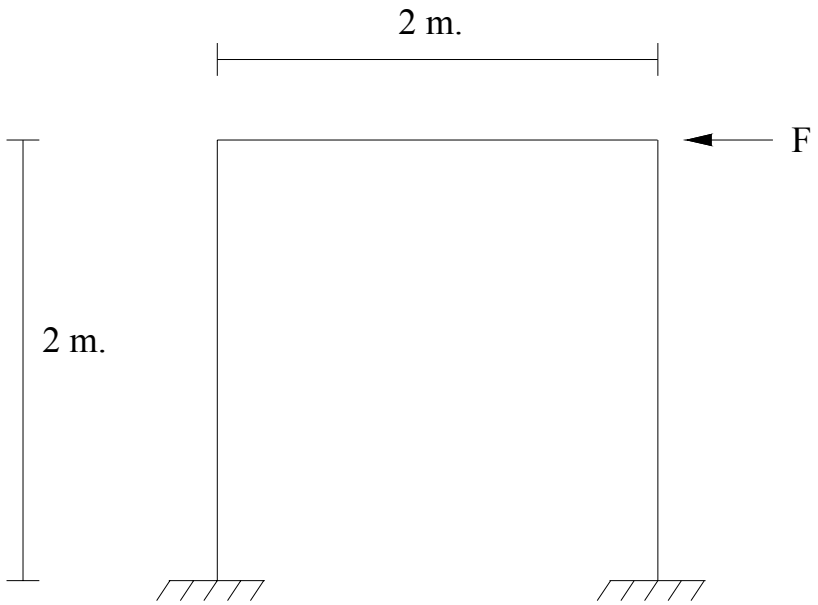

Figure 4.10: Portal frame for testing the influence of mesh density on the predictive capacity of the frame super-element.

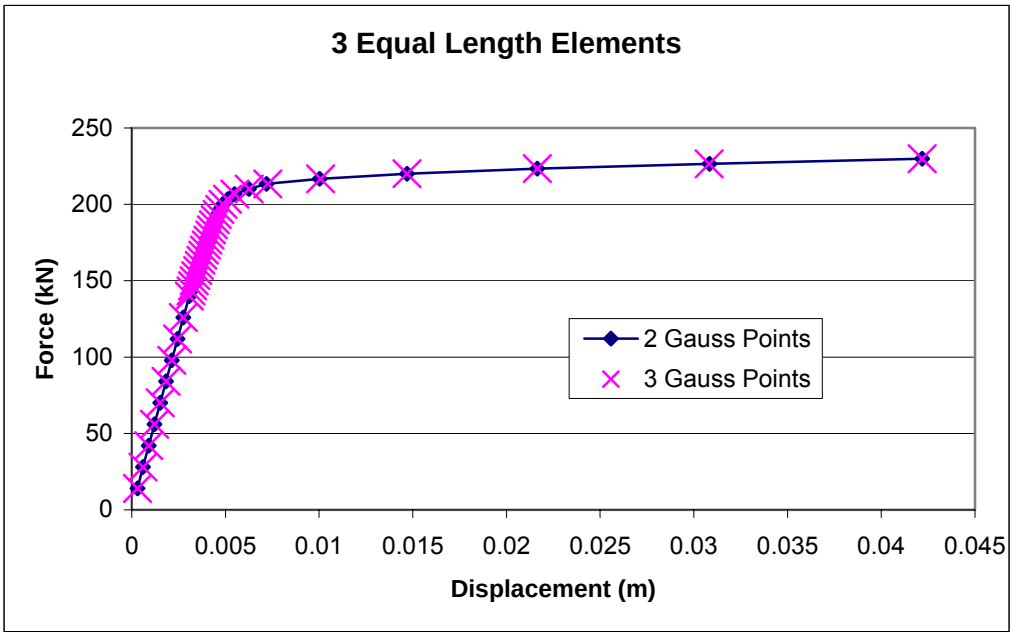

Figure 4.11: Force-displacement relationship at the loaded node of the portal frame for 3 equal-length divisions of frame super-elements.

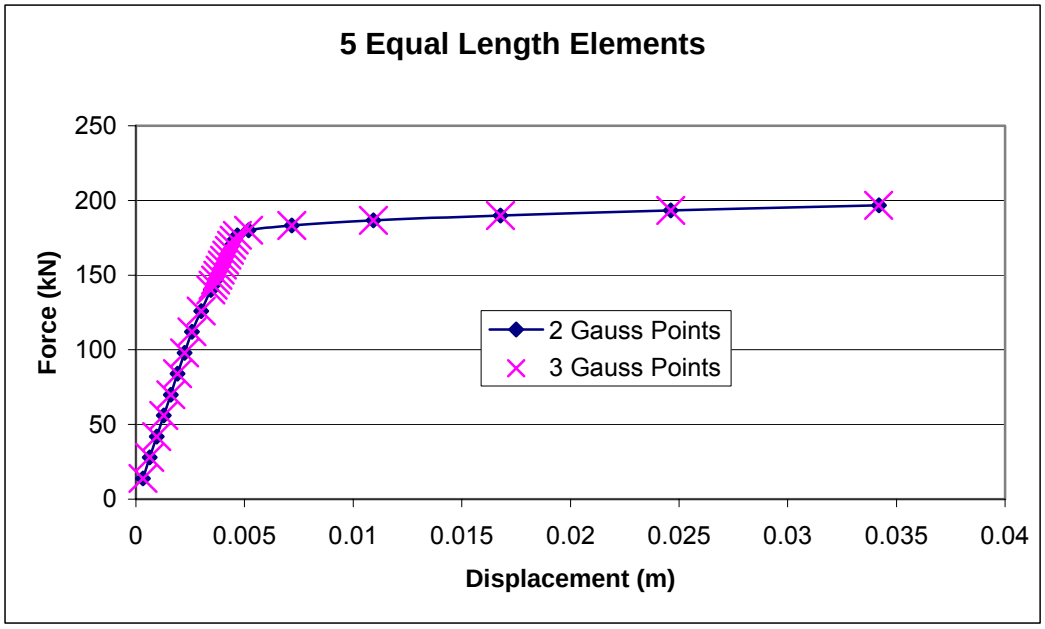

Figure 4.12: Force-displacement relationship at the loaded node of the portal frame for 5 equal-length divisions of frame super-elements

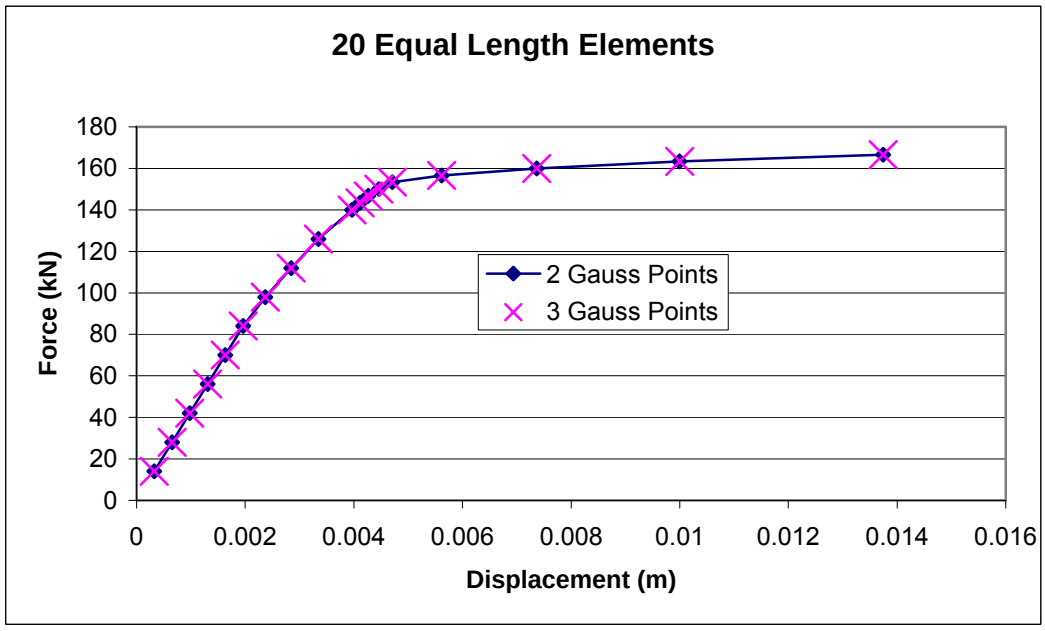

Figure 4.13: Force-displacement relationship at the loaded node of the portal frame for 20 equal-length divisions of frame super-elements

Examination of the results of these test problems reveals that using more than 2 Gauss integration points does not increase the accuracy of the solution.

4.1.3.2 Mesh Density of the Cross Section

Discretization of the cross-section of the frame elements is investigated. For this purpose a cantilever column of Figure 4.14 is used. The cross-sectional details of the column are as shown in Figure 4.2.

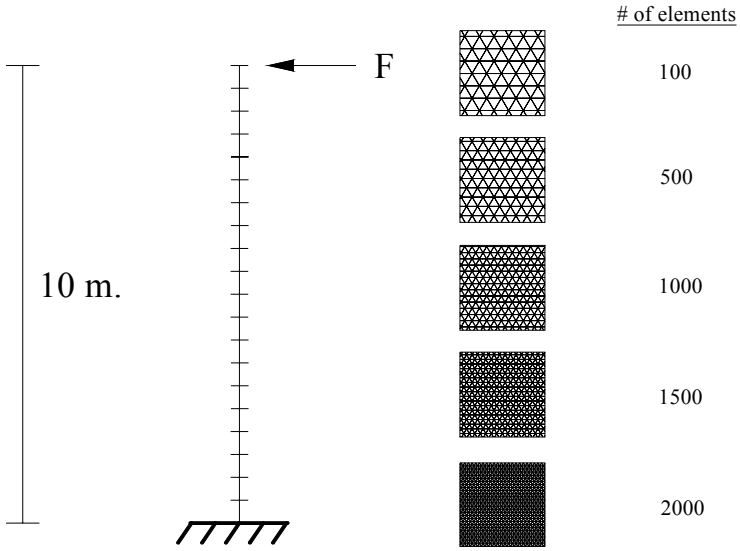

Figure 4.14: Analysis of a cantilever column with varying cross-sectional mesh densities.

The frame super-element is modeled with 20 equal length finite beam elements. Analyses are performed for cross section that is divided into 100, 500, 1000, 1500 and 2000 triangular elements. Frame is subjected to a lateral load of *12.74 kN* that is near its ultimate load capacity. Force-displacement relationships are shown in Figure 4.15.

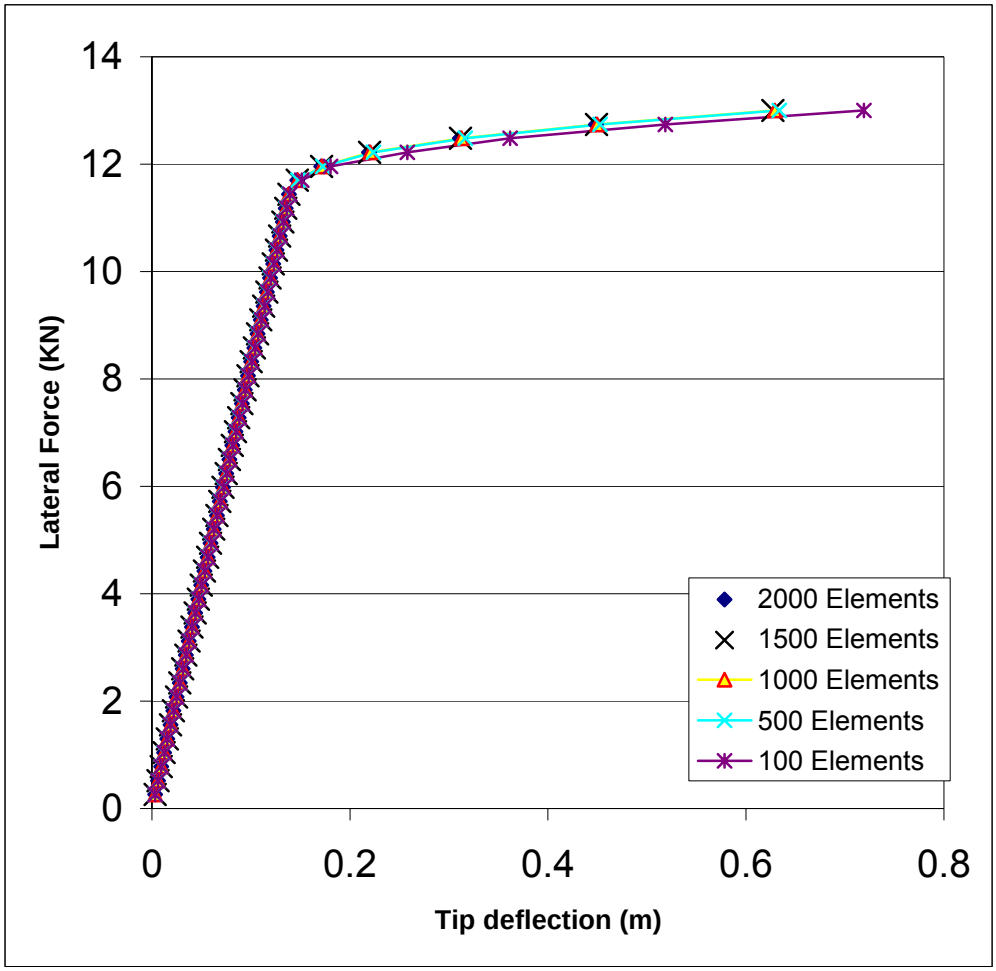

Figure 4.15: Force-displacement relations of a cantilever column with different cross-sectional mesh densities

Inspection of Figure 4.15 shows that it is possible to predict the response even with a rather coarse mesh up to a load level very close to its ultimate capacity. Taking the response for the case with a cross-sectional mesh density of 2000 elements as the true response the relative error for coarser cross-sectional mesh densities are shown Table 4.4.

Table 4.4: Relative error in the predicted tip deflection for varying cross-sectional mesh densities.

100 Elements	500 Elements	1000 Elements	1500 Elements
%16.354	%1.764	%0.508	%0.200

From Table 4.4 it is clear that increasing the mesh density of the cross-section beyond 1000 elements does not increase the accuracy significantly. Therefore, a mesh density of 1000 triangular elements is used in the rest of the study. It should be noted that using even 500 elements is sufficient to meet the precision demand in most practical situations.

4.1.3.3 Mesh Schemes along the Frame Super-Element

In real-life frame structures the degree of plastification varies along the length of the frame element. For instance, the level of stressing is quite high at the member ends when the structure is subject to lateral loads during an earthquake while their middle portions are relatively less stressed. Therefore, the use of a graded meshing of the elements as opposed to uniform meshing is expected to be more effective. In an attempt to explore the effect of mesh density and the relative effectiveness of various graded meshes the test structure of Figure 4.14 is modeled by using a uniform mesh of 5, 10, 50 and 100 elements and graded mesh of 3, 7, and 9 elements shown in Figure 4.16. The relative size of each sub-element in the graded mesh is also indicated in Figure 4.16.

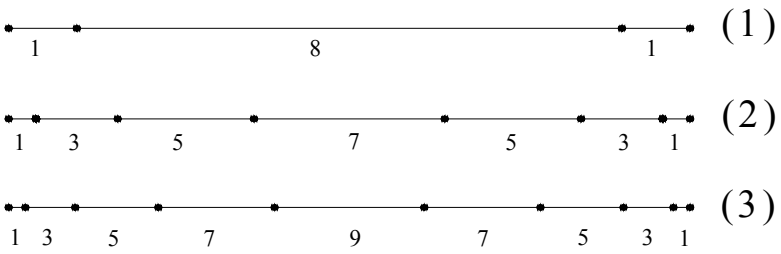

Figure 4.16: Special graded meshing of frame super-elements.

The cantilever column structure of Figure 4.14 is analyzed for the two mesh types with various mesh densities. The force-deflection relationship at the tip is obtained as shown in Figure 4.17.

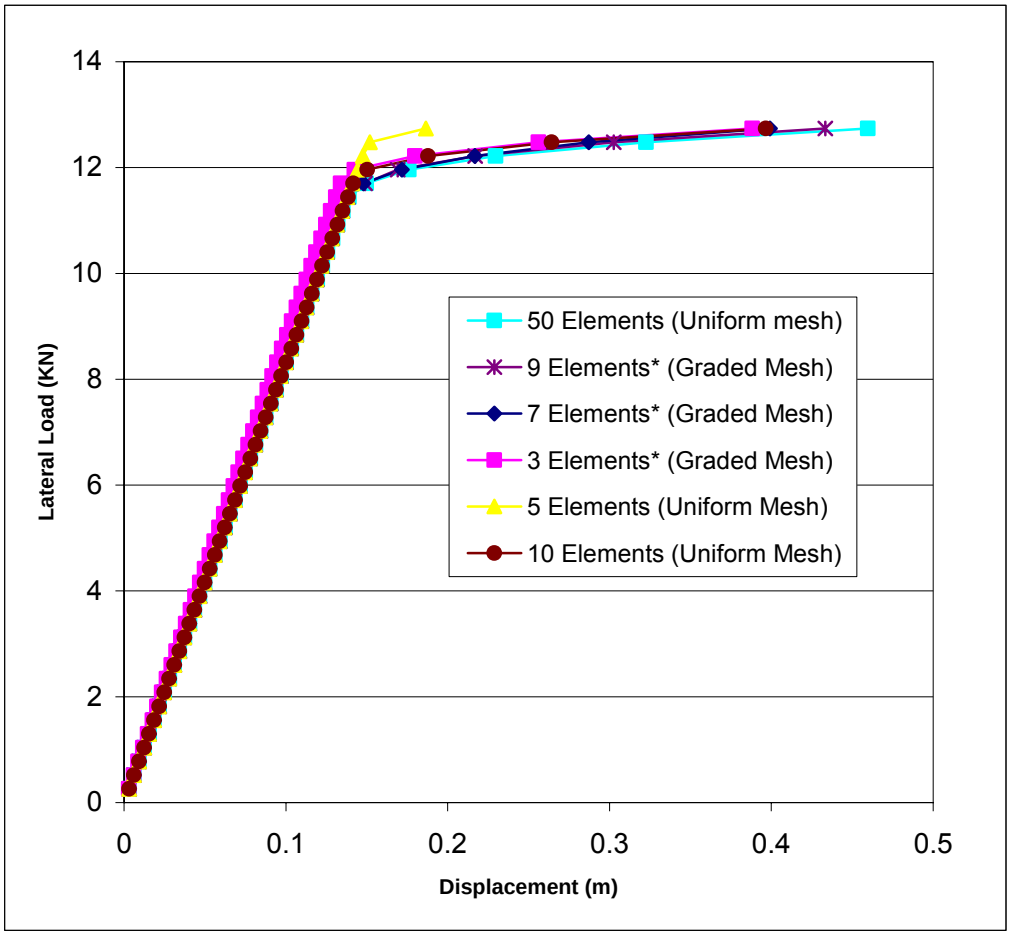

Figure 4.17: Force-displacement relationships of the frame super-element for various densities of uniform and graded mesh.

The maximum deflection of the frame for the case of a uniform mesh with 50 elements is taken as the true response and the relative departure from the true response of the other meshing schemes and densities are given in Table 4.5.

Table 4.5: Relative departure from the true response of various meshing schemes and densities for the frame super-element.

5 Elements	10 Elements	3 Elements*	7 Elements*	9 Elements*
%-59.399	%-13.701	%-15.510	%-13.170	%-4.477

It is obvious from Figure 4.17 and Table 4.5 that, for the finite element modeling of the frame super-element, the graded meshing schemes of Figure 4.16 are clearly

more effective than the uniform meshing with comparable densities. Moreover, 9-elements graded mesh seems to be the most effective one for modeling the frame super-element in the plastic range.

In order to further investigate the super-element meshing schemes and densities, the portal frame structure of Figure 4.10 is analyzed again. The loading is applied incrementally until collapse and the elements are modeled with uniform and graded meshes of varying densities. The response predictions for uniform meshing schemes are in Figure 4.18 and those for the graded mesh schemes are shown in Figure 4.19.

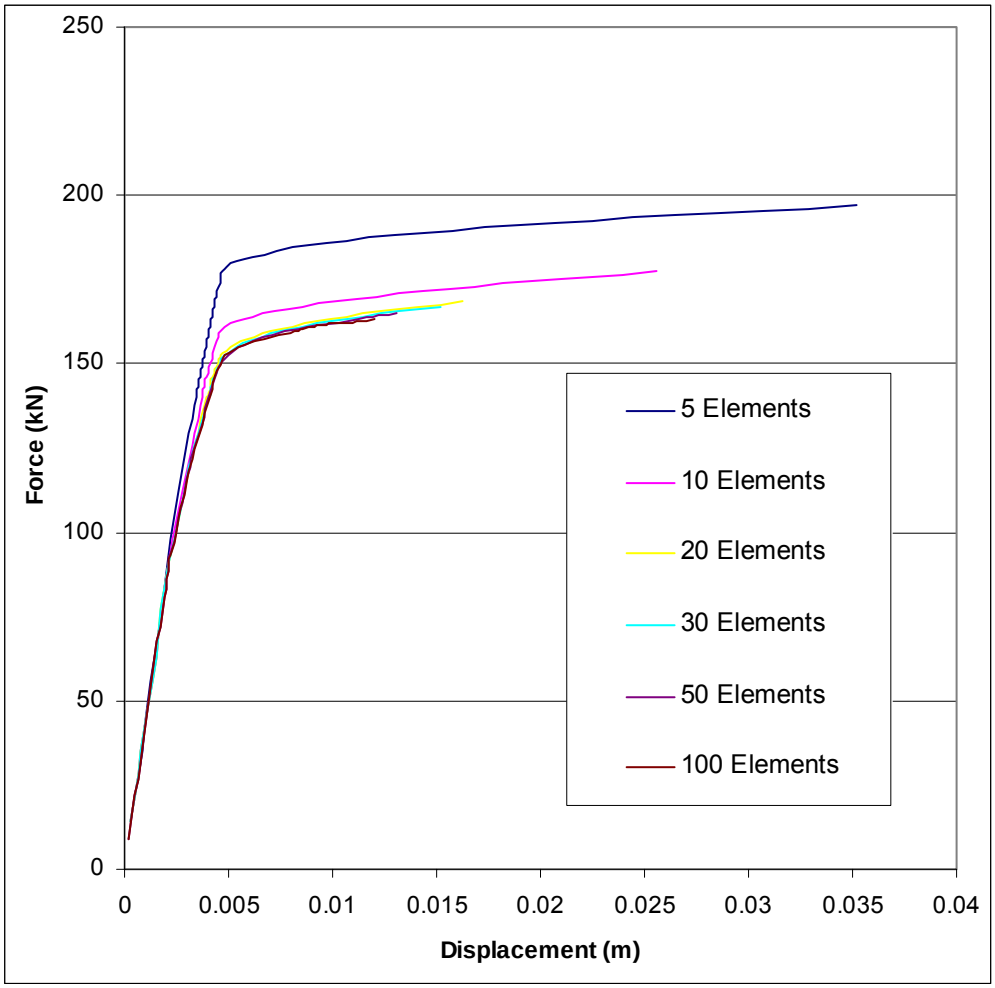

Figure 4.18: Load-deflection relationships for various uniform mesh densities of the frame super-element.

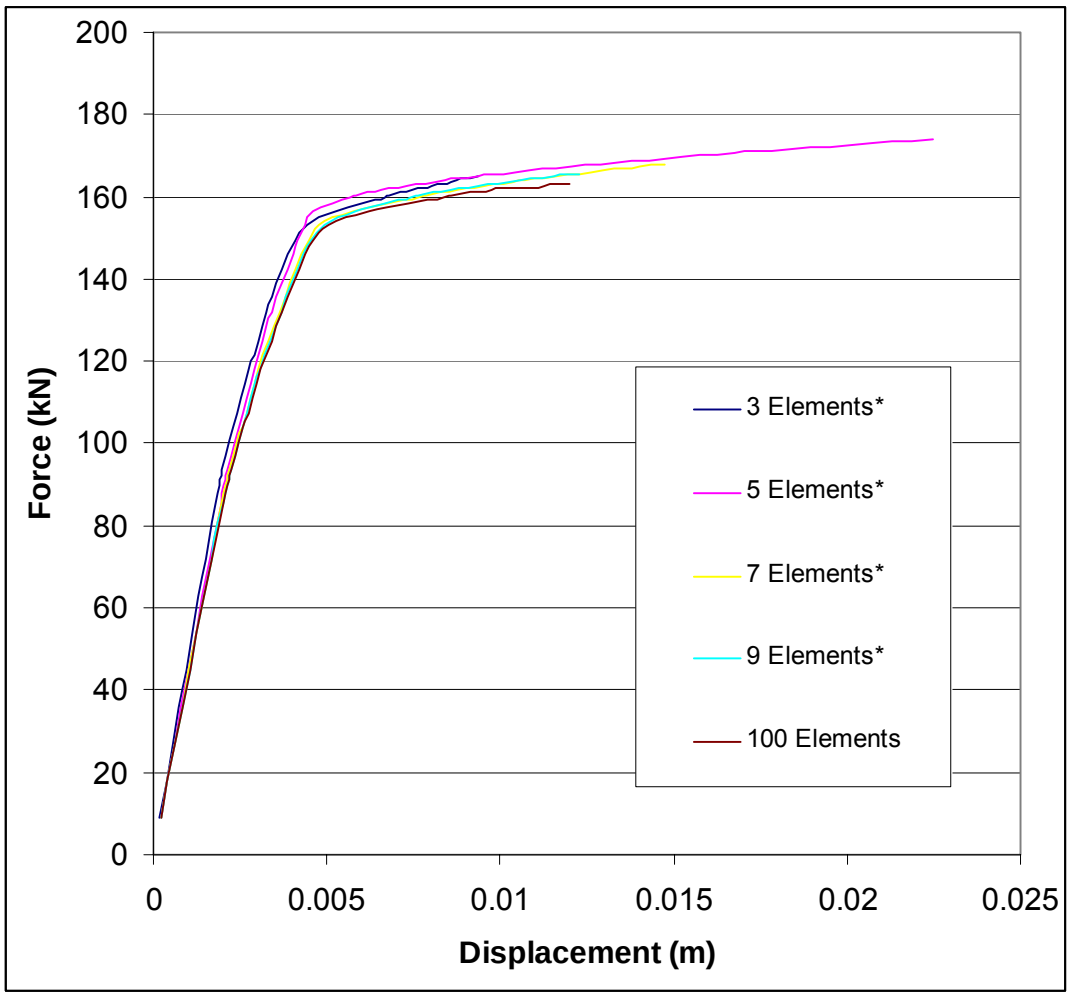

Figure 4.19: Force-displacement relationships for various graded mesh densities of the frame super-element

It is again seen that the use of a graded meshing scheme for the frame super-elements makes them more effective.

4.2 Comparison of the Program Results with Experimental Measurements

First response predictions of the program are compared with the experimental results of a beam without shear reinforcement under four-point loading.

As part of another study on the strengthening of reinforced concrete frames with masonry walls, some experiments have been performed in the Materials of

Construction Laboratory. The data obtained from these experiments are available and used for the verification of the proposed numerical procedure in Section 4.2.2 and Section 4.2.3.

4.2.1 Beam under Four-Point Loading

A lightweight reinforced concrete beam is tested under four-point loading by Walraven, JC [23]. There is no web reinforcement in the beam. The geometry and the cross-sectional details of the beam are given in Figure 4.20. The concrete cubic compressive strength and bar yield strength is given as 34.2 MPa and 440 MPa, respectively. Experimental data obtained are given in Figure 4.21 by Pamin J & Bors R [39].

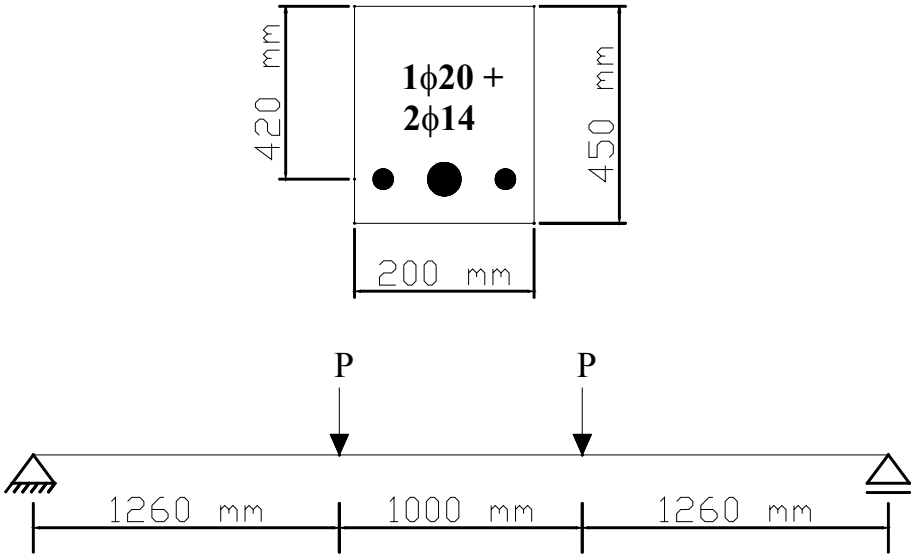

Figure 4.20 : Beam under four-point loading [39]

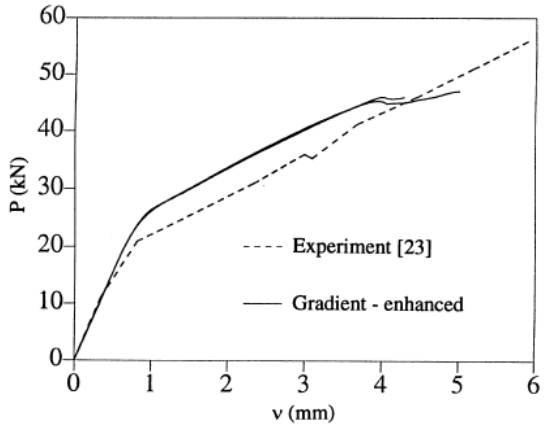

Figure 4.21: Load-deflection diagram for the beam under four-point loading [39]

The test is simulated by the program for two cases: First, it is assumed that the concrete takes tension according to the stress-strain diagram given in Figure 4.22. Later, the analysis is repeated by assuming that the concrete will not take any tension. Material models of concrete and the steel are taken as C35 and S420a respectively. The beam is divided into three frame super elements at the location of the point loads. Each frame super element is composed of equally spaced 50 frame finite elements. The load deflection diagrams that are predicted by the program developed in this study are given in Figure 4.23.

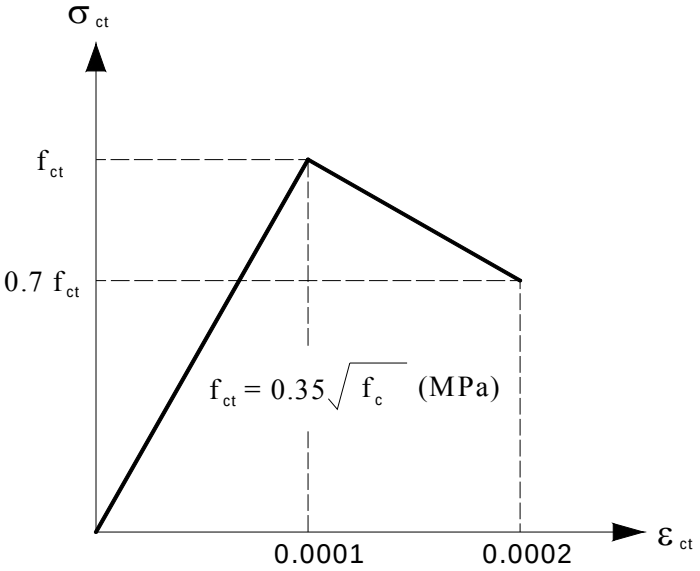

Figure 4.22: Stress-strain relationship for concrete in tension [42]

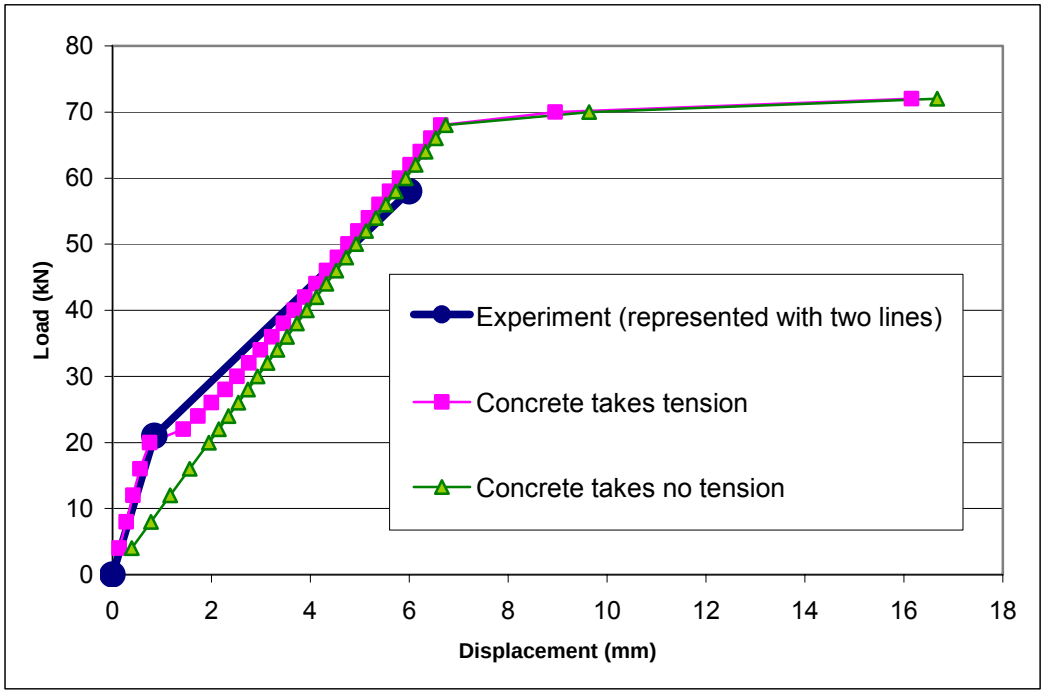

Figure 4.23: Load deflection diagrams predicted by the program developed in this study together with the representation of the experimental data with two straight lines.

Figure 4.23 shows that the program developed gives satisfactory results compared with the experimental data. Overestimation of the stiffness of the beam at the final stages of loading can be explained with the perfect bond assumption of the program.

Finally, it should be noted that considering concrete will take no tension becomes a valid assumption after the load level where tension cracks occur in concrete.

4.2.2 R/C Frame Only

The sketch of the bare frame specimen used in the experiments is shown in Figure 4.24 and the cross-sectional details of the frame members are given in Figure 4.25.

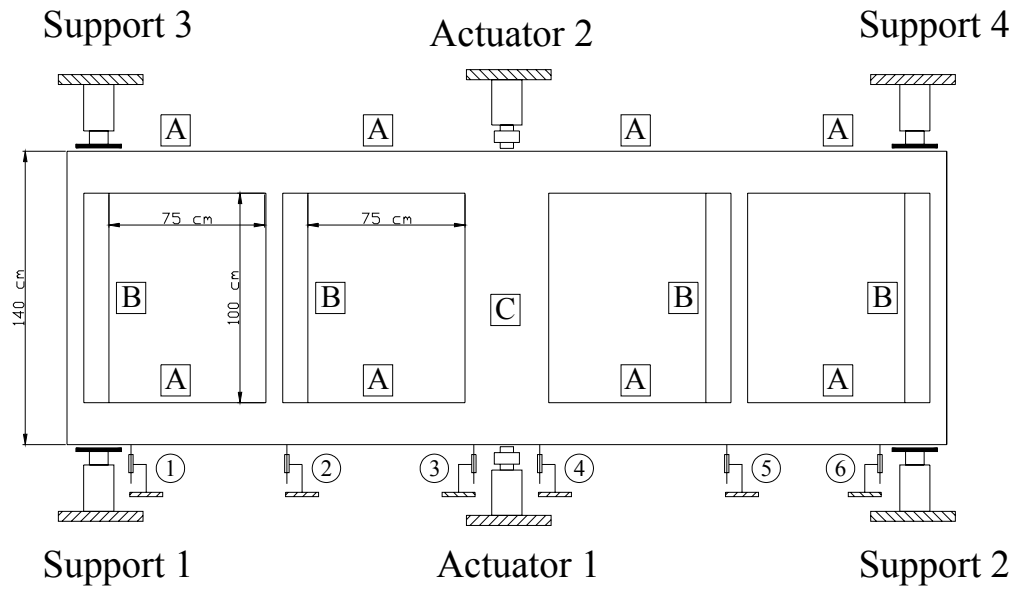

Figure 4.24: Experimental setup and the bare fame test specimen.

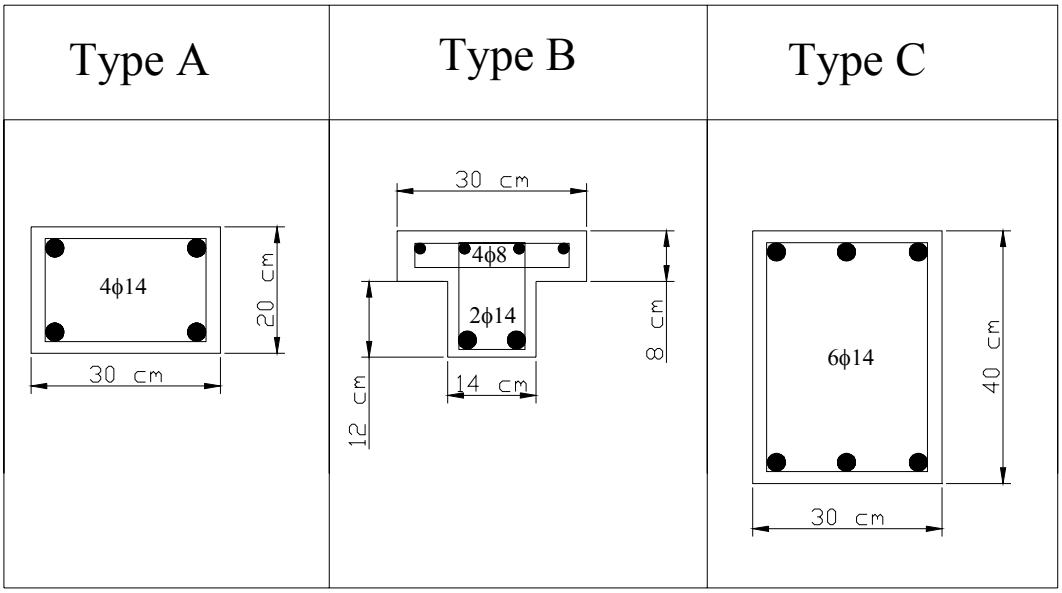

Figure 4.25: Cross-sectional details of the frame members used in the experiments.

The concrete class of the test frame specimen is C20 and the steel class of the reinforcement is S420a. There are load cells attached to each actuator hydraulic cylinder. A cyclic load is applied by *actuators 1* and *2*. Displacements are measured by LVDTs, which are located as shown in Figure 4.24. After processing the raw

data recorded during experiments, the load-displacement curve for the load cell 1 and LVDT 3 is obtained as given in Figure 4.26.

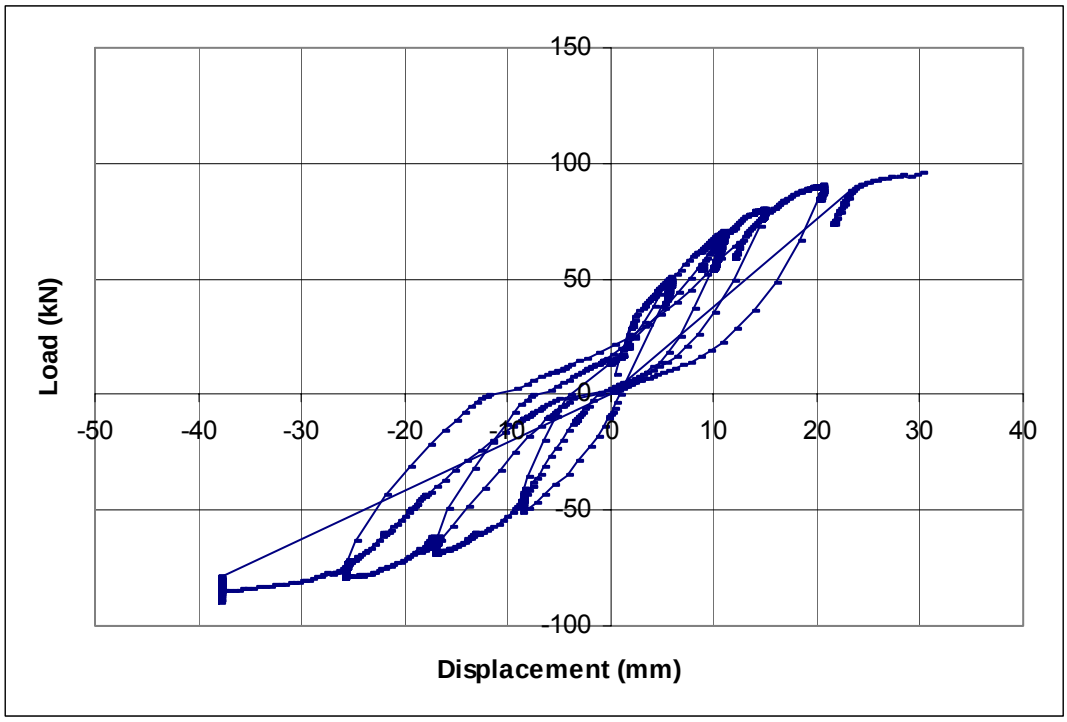

Figure 4.26: Experimentally observed load-displacement relation of test specimen for load cell 1 & LVDT 3

The experimental specimen is modeled numerically with the program as given in Figure 4.27. Here, rigid zones are only considered at connections 5 and 6. The load is applied at *Node 5*. The characteristic strengths of steel and concrete are used in the calculations. The analysis is performed by neglecting the self-weight of the frame. Initially, fixed support conditions are assumed at nodal points 2 and 10. The calculated displacement corresponding to each load increment at *node 5* is plotted in Figure 4.28.

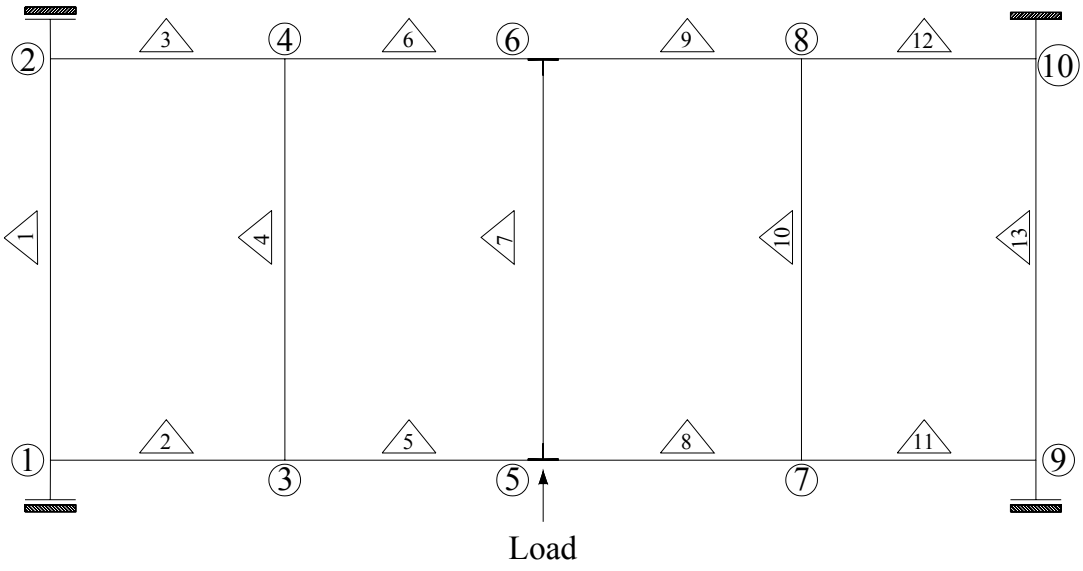

Figure 4.27: Finite element model of the test specimen used for analysis.

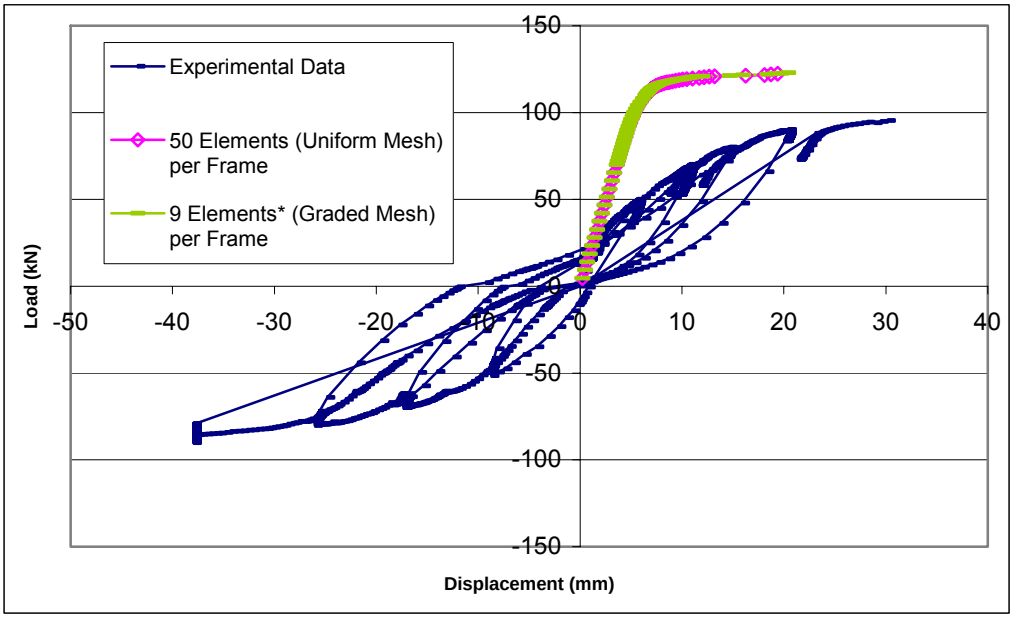

Figure 4.28: Experimentally measured and numerically predicted behavior of the R/C frame (fixed ends)

The actual support conditions of the test frame are complicated. However, a close examination of the support devices shows that it is closer to a pin rather than a clamp. As a result, the modeling the frame as given in Figure 4.27 creates a stiffer system. Therefore, the same analysis is performed again using pin support

conditions at *Node 10* and a roller support at *Node 2*. The load-displacement relationship of *Node 5* is presented in Figure 4.29. It is expected that the actual behavior of the frame will be between these two limit states.

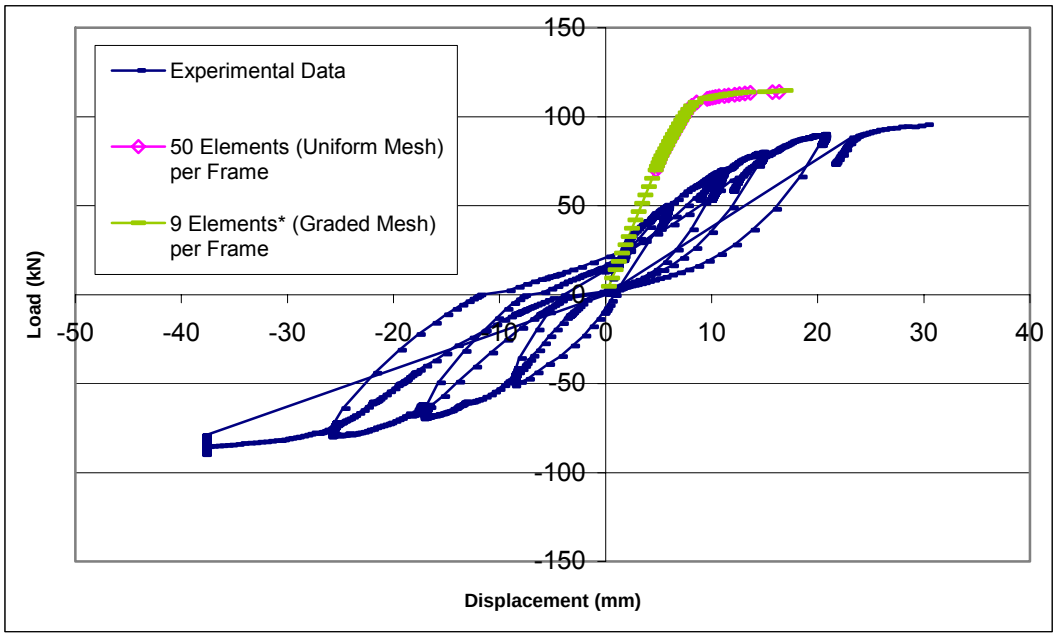

Figure 4.29: Experimentally measured and numerically predicted behavior of the R/C frame (pin ends)

Based on the response predicted by the program it is observed that the numerical model is stiffer than the actual structure. Beside the obvious fact that the displacement based finite element models are always stiffer than the actual structures, part of the reason for this discrepancy can be attributed to the following details not included in the mathematical model.

- The loading is cyclic in the experiments and it is possible that some strength reduction is experienced by the frame members as a result of damage caused by load reversals.

- The loading was not continued up to failure since the same damaged frames were planned to be used again after strengthening.

- It is possible that some local crushing of concrete occurred during experiments at support locations and at the point where the load is applied as a result of high local stresses. This is not accounted for in the numerical simulation.

- Strength reduction in the frame members of the test specimen is possible due some slippage of the rebars whereas a perfect bond is assumed in the numerical model.

4.2.3 R/C Frame with Masonry Infill

In another experiment, an identical R/C frame with masonry infill was tested. The same experimental setup was used and each bay was filled with a 9cm thick masonry wall as shown in Figure 4.30.

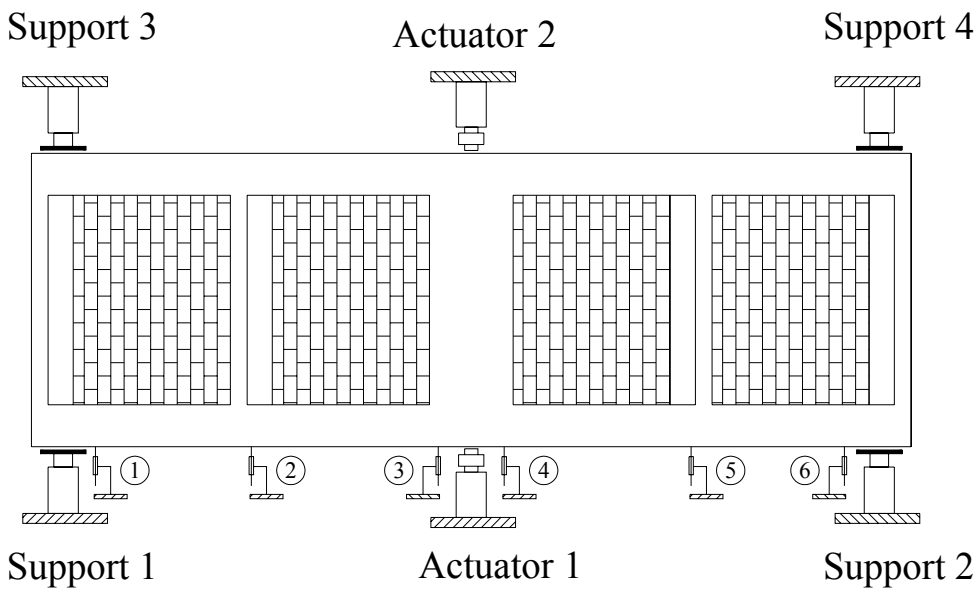

Figure 4.30: R/C frame with masonry infill

The test was conducted by applying the load monotonically in one direction only. The load-displacement variation recorded by the load cell attached to Actuator1 and LVDT3 is shown in Figure 4.31.

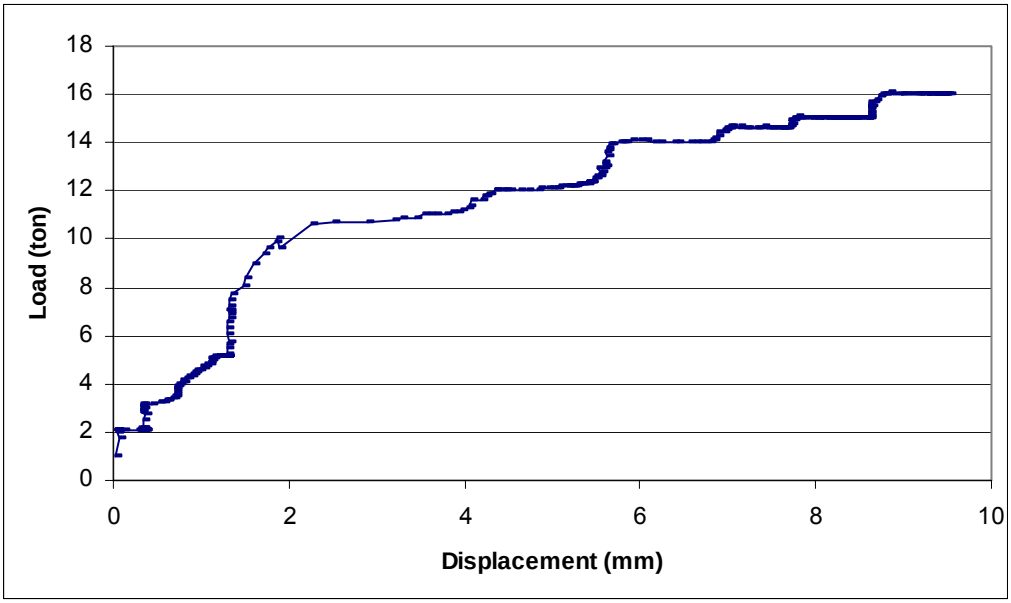

Figure 4.31: The load-displacement diagram of the frame with masonry infill (Actuator1 – LVDT3)

The R/C frame is modeled by using 9-element graded mesh scheme for the frame super-elements. The infill walls are modeled by two diagonal compression-only struts described previously in Section 2.1.1. To determine the material properties of the masonry infill relevant to the strut model used, the masonry units are tested in the vertical and horizontal directions. The unit strength of the masonry block is determined as 8.5 (MPa) in the vertical direction, and 13.5 (MPa) in the horizontal direction. The equivalent strut model parameters are then calculated by the equations given in Section 2.1.2.

The response calculations of the numerical model are done by incrementally increasing the load applied at *Node 5* until total collapse of the structure. The calculations are repeated once for the case of clamped end conditions and once for the case pin end conditions at support nodes. The results are presented in Figure 4.32 and Figure 4.33, respectively. The numerically predicted load level and the location of the first infill damage, the damage pattern and the progression of the damage state follows very closely those which are experimentally observed. The final collapse of the structure predicted by the numerical model comes after the

capacity of the outer bay infill walls is exhausted. This is simultaneously followed by collapse of the R/C frame under excessive load released by the failing infills. This ultimate capacity was not reached in the experimental program in order not to damage the specimen excessively since it was planned to be used again after repair and strengthening of the damaged infills by the application of FRP on their surfaces.

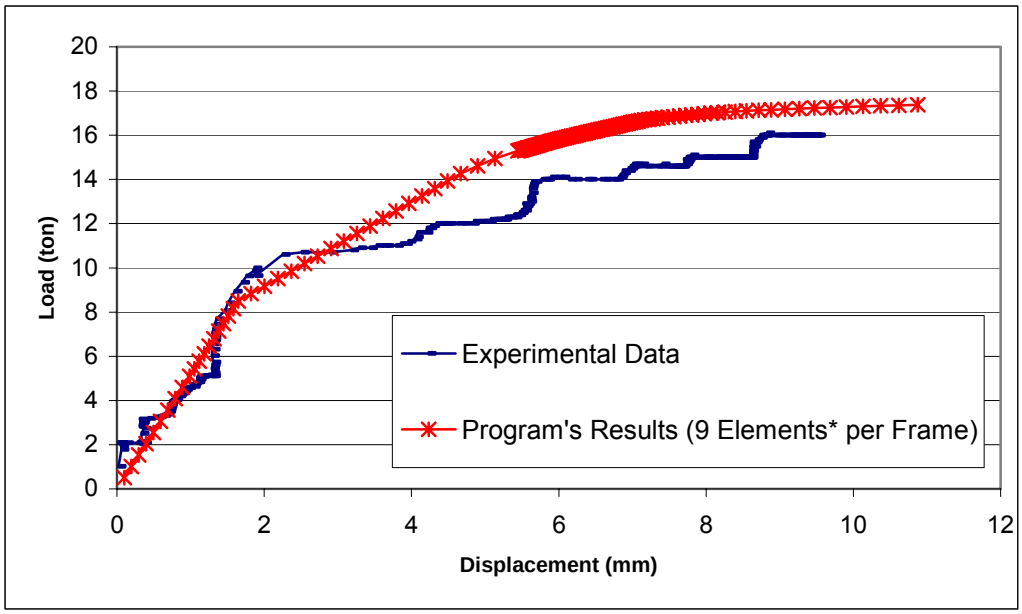

Figure 4.32: The numerically predicted and the experimentally measured response of the R/C frame with masonry infill (fixed ends)

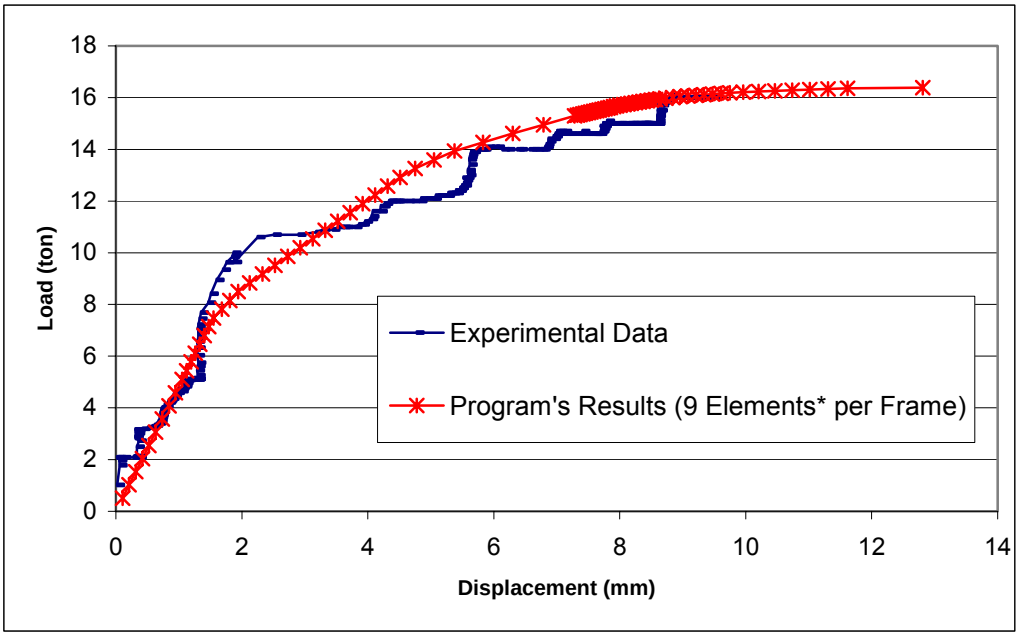

Figure 4.33: The numerically predicted and the experimentally measured response of the R/C frame with masonry infill (pin ends)

4.3 Non-linear Analysis of a Sample 3D Structure

A sample 3D structure is analyzed to test the program. Geometry of the structure is given in Figure 4.34. Section properties are given in Figure 4.25. In this example for columns section *type A* and for beams *type B* is used. Concrete and steel classes of the frame members are C20 and S420a respectively. All frame members are modeled with specially discretized 9 frame finite elements (see Figure 4.16). Cross section of the frame elements are subdivided into 2000 triangular elements.

Two sequence of loading is applied to the structure. First same support displacement u is applied at the nodes shown in Figure 4.34. Value of the support displacement is 7 mm. Then system is loaded laterally. Point loads on structure are shown in Figure 4.34. The loads F_{1a} and F_{2a} are 1.5 times of the loads F_{1b} and F_{2b} respectively. Additionally loads on the second storey are two times of the corresponding loads in the first stories. Load displacement diagrams of loads F_{2a} and F_{2b} corresponding displacements are drawn in Figure 4.35.

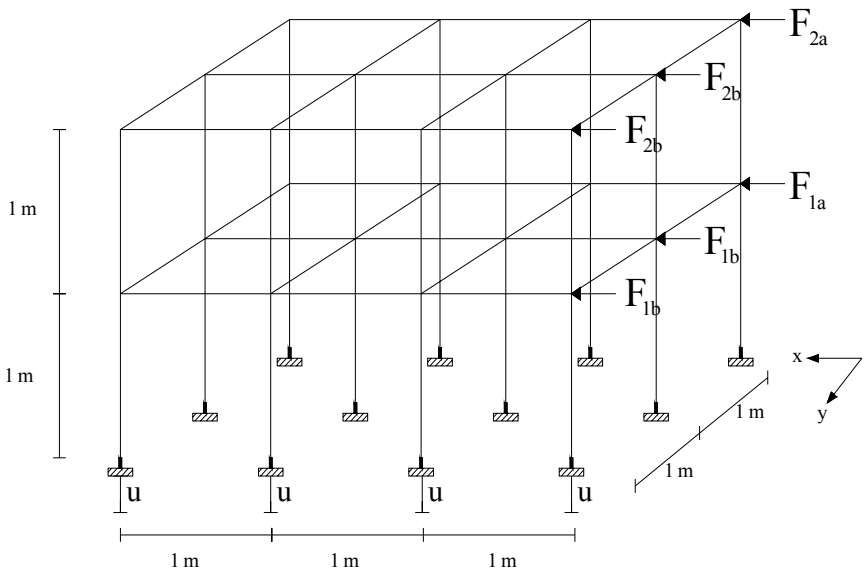

Figure 4.34: Sample 3D structure

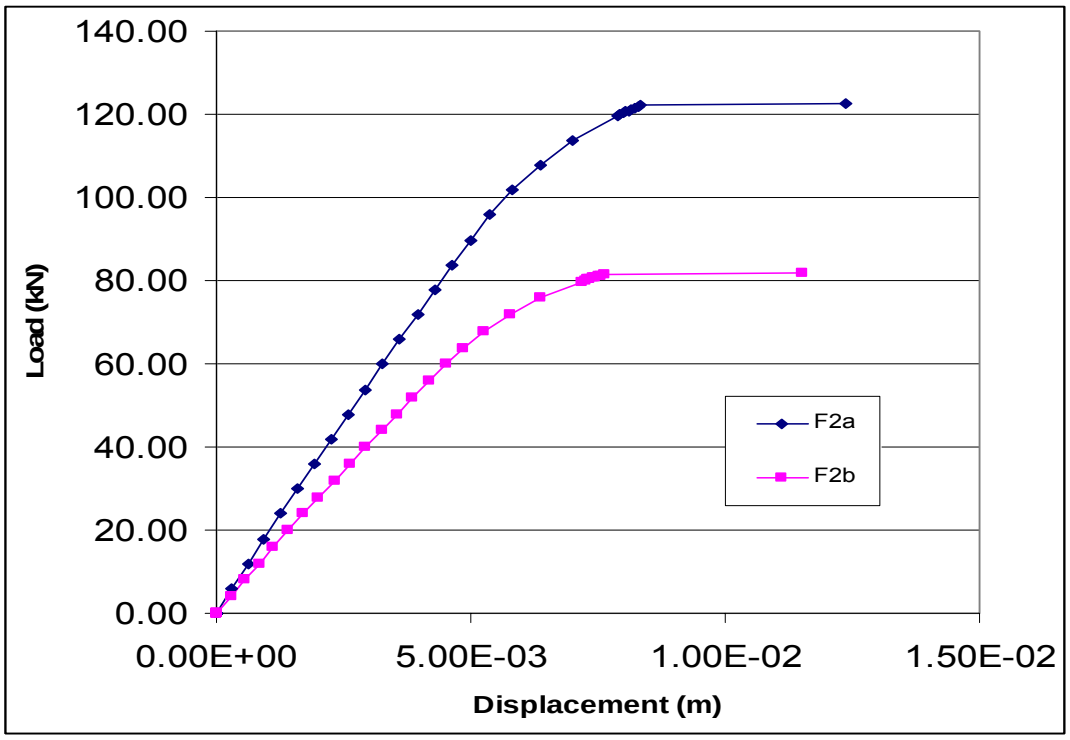

Figure 4.35: Load displacement diagram for the load F2a and F2b and the corresponding displacements

Deflected shape of the system after imposing the prescribed displacements is given in Figure 4.36 with magnifying the displacements by a factor of 50. In addition to that deflected shape of the system at the final state is given in Figure 4.37 and Figure 4.38.

It should be noted that the solution time of the problem is about 200 seconds for a computer with AMD 1700+ CPU.

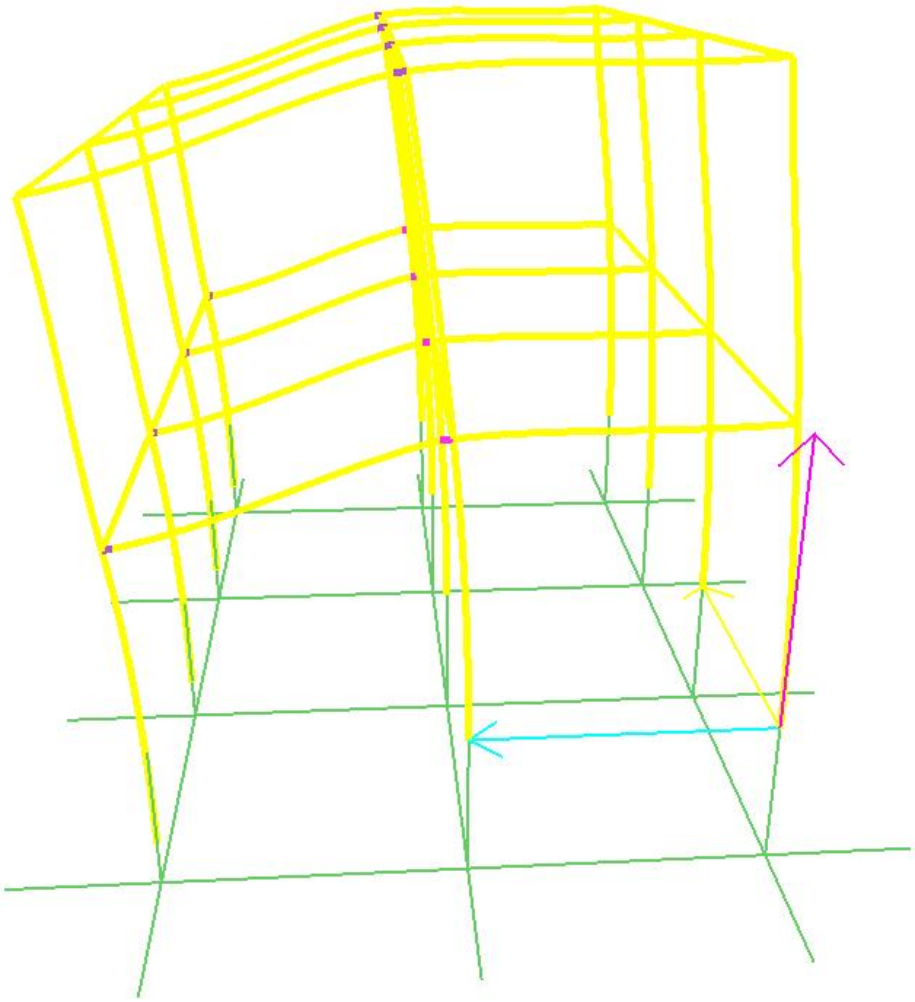

Figure 4.36: Deflected shape of the sample 3D structure after imposing prescribed support displacements (looking through positive x direction)

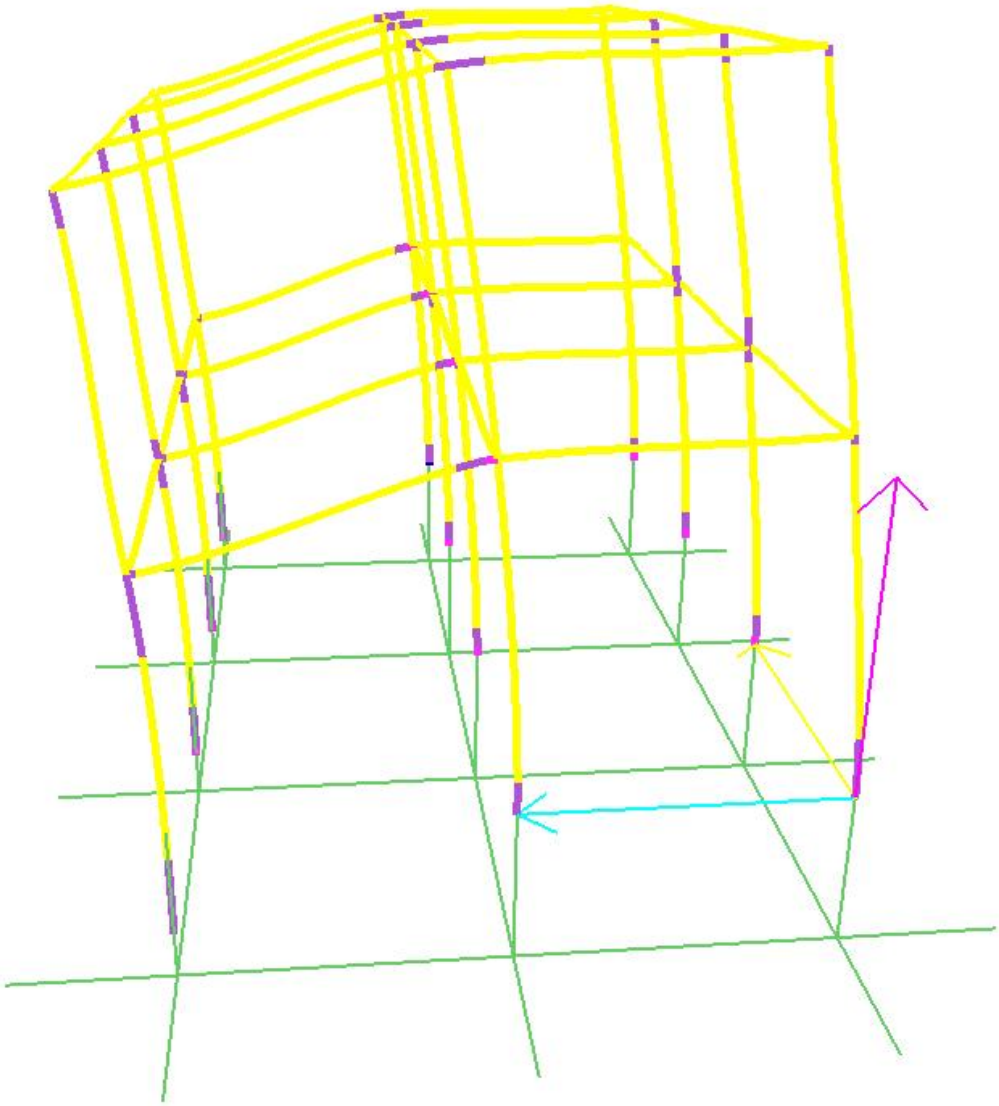

Figure 4.37: Deflected shape of the sample 3D structure at the final stage of loading (looking through positive x direction)

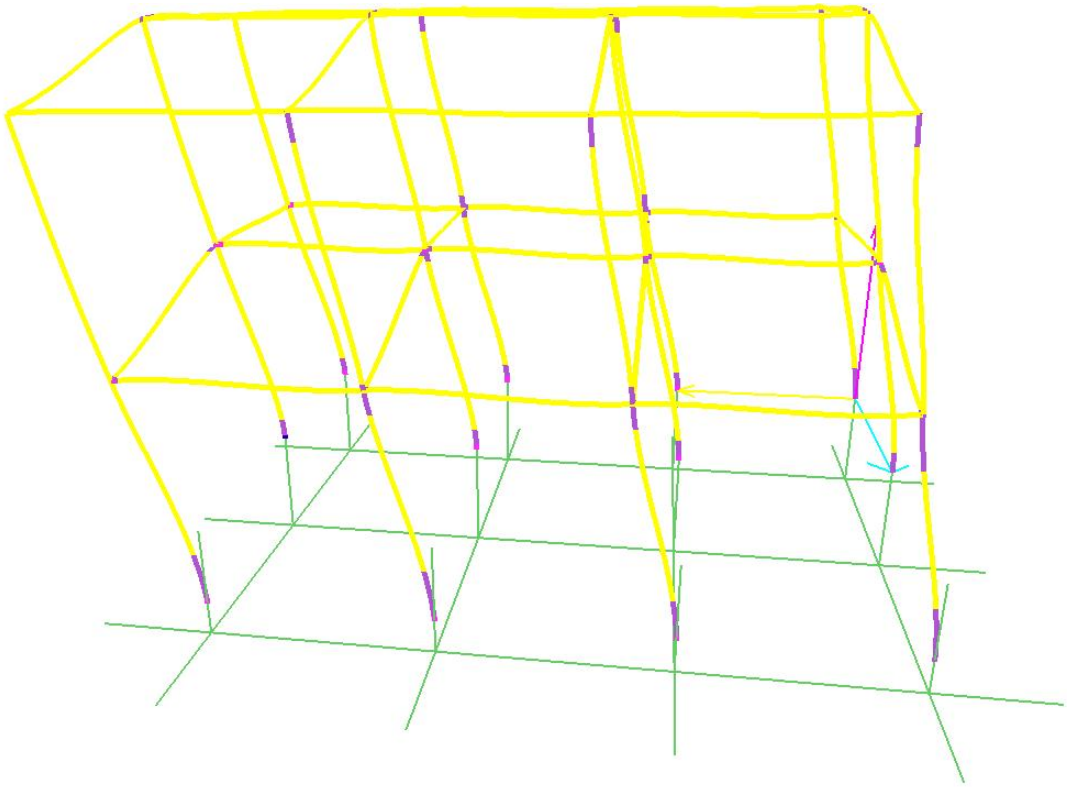

Figure 4.38: Deflected shape of the sample 3D structure at the final stage of loading (looking through the same direction as in Figure 4.34)

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 Summary

In this study modeling nonlinearity of masonry infilled reinforced concrete frames are investigated. Reinforced concrete frames are modeled with 3D displacement based Hermitian beam finite elements. Material nonlinearity of the frame elements is modeled by the spread-of-plasticity approach. Cross section of the frame is divided into monitoring triangles at the integration points. This allows monitoring the gradual yielding of the cross-section. Bending and the axial displacements are coupled in the inelastic range. Therefore, a special tangential stiffness matrix is derived that couples the displacements. Newly derived tangential stiffness matrix is not symmetric. Therefore the full matrix is stored in a band form. Constraint equations are implemented for the frame element by assuming that the member end nodes are in the rigid zone. Constraint equations are further modified since these zones are assumed as axially and torsionally flexible.

A super element is formed by using a number of sub elements since the predicted displacement response of the elements is only approximate in the inelastic range. Internal degrees of freedoms of the super element are condensed out.

For modeling the infill wall, a non-linear infill wall model which only considers the corner crushing failure mode of the infill is presented and used in this study.

For the implementation of the procedure, a program is developed with object-oriented approach. The components of the procedure stated above are each represented with a separate object.

Finally, some case studies are performed to test the program. Different discretization schemes of the frame members are tested. Moreover, some other parameters controlling the solution such as the number of integration points used and the discretization of the element cross-section are tested. The response predictions of the program are compared with the available experimental data.

5.2 Limitations

Limitations and the assumptions of the nonlinear analysis program are listed below.

- Shear deformations are neglected for the frame element. Formulations include flexural and axial deformations only.
- Warping effect is neglected.
- Torsional behavior of the frame element is assumed to be linear.
- Geometric nonlinearity is neglected.
- Hardening of the steel is neglected. Additionally, no hysteretic damage model is adopted in the program.
- Web reinforcement and its contribution to concrete strength (due to confinement) are neglected.
- In the infill wall formulation, only the corner crushing of the infill wall is taken as the limit state. Other limit states like the shear failure and diagonal crushing are not taken into account.
- While binding the element degrees of freedom to the structural degrees of freedom the space between the two is assumed to be rigid.

5.3 Conclusions

A numerical procedure is suggested for the nonlinear analysis of reinforced concrete frames with infill walls. The use of the spread-of-plasticity approach for the frame elements in the inelastic range makes it possible to observe the gradual yielding of the cross-section. The axial and the bending deformations of a reinforced concrete frame element in the inelastic range are, in general, coupled and this coupling effect can easily be handled in this approach since the element cross-section is taken into consideration by its actual geometry. However, this procedure is computationally more intensive than other approaches such as using isolated concentrated plastic hinges at various locations along the member. It is seen that ignoring this coupling effect in the formulation results in an excessive number of equilibrium iterations for convergence to equilibrium state in each load increment. Therefore, a more populated stiffness matrix is formulated that couples axial and the bending displacements of the section. However, the resulting stiffness matrix is not symmetric. Therefore, symmetric storage schemes and solvers cannot be used anymore.

It is observed that the elasto-plastic super elements, which are composed of several nonlinear 3D frame finite elements, overestimate the ultimate load capacity of the actual frames when the predicted response is compared with a limited number of experimental results. This may be due to a lack of hysteretic damage model for the frame elements of this study since the actual experiment are performed under cyclic loading. Additionally, the capacity of the reinforced concrete sections may be less than the one expected due to uncertainties such as bond damage during cyclic load applications.

The response of frames with infill walls predicted by this procedure is also satisfactory. A better correlation with the experimental results is observed in a case study involving infill walls. However, this may be partly due to the fact that in the experimental study the loading was monotonic in one direction only.

The program developed in this study for the nonlinear analysis of reinforced concrete frames with infill walls is a good framework for the future studies. It is developed considering the possible extensions in the future. Additional components can easily be implemented by deriving from existing classes or creating new classes that is consistent with the current design.

5.4 Recommendations for Future Studies

Following are suggested for future studies.

- The response predictions of the procedure are needed to be verified with more experimental data.
- The program can easily be adopted for steel frames by introducing steel material properties into the DLL, which calculates the sectional response.
- Infill model can be extended to take other failure modes into account.
- Other structural members can be implemented in the program.
- Shear deformations can be included in the beam finite element to model deep beams and columns.
- Force based and mixed formulated elements may be implemented and tested to see whether they give more realistic results with less number of elements.
- Accuracy of the frame elements can further be tested. Alternative discretization schemes can be explored.
- Performance of the program can be improved by various optimizations.
- In addition to the band solver, a profile solver can be implemented.
- Since the spread-of-plasticity approach is used to model the material nonlinearity, iterations may converge to some un-realistic states, or even not

converge in some occasions. This is due to the rise and falls in the tangential stiffness of the structure. A better solution strategy can be implemented to avoid such numerical instabilities.

- In the case studies, it is observed that up to a certain load level, the frame element behaves basically as a linear element. Additionally, it is seen that the initial stiffness of the frame members can be very well predicted with a fewer number of elements. Consequently, the system can be modeled with frame members that are discretized into smaller number of elements initially. Later, as plastification starts, members can be substituted with the super elements with finer meshes.

5.4.1 Extending the Code

Since the program is developed in an object-oriented manner any extensions should comply with the principles of design for the sustainable development of the code. For instance, to add a new finite or structural element to the code a new class shall be derived from the base class CSElements. For connecting the newly derived element with the rest of the code necessary functions should be overridden. This requirement is also dictated by the compiler since the common functions that integrate the element with the domain object and the solution algorithm are declared as pure virtual functions. This means that a newly derived element could not be instantiated unless these pure virtual functions are not implemented. The same applies to the other components of the program.

REFERENCES

1. **Stafford Smith B** "Lateral stiffness of infilled frames." Journal of Structural Division (ASCE) (1962): 88(6): 183-199.

2. **Stafford Smith B** "Methods for predicting the lateral stiffness and strength of multi-storey infilled frames." Building Science (1967): 2: 247–257.

3. **Mainstone RJ & Weeks GA** "The influence of bounding frame on the racking stiffness and strength of brick walls." Proceedings of the 2nd International Conference on Brick Masonry, Stoke-on-Trent, UK, (1970): 165–171.

4. **Wood RH** "Plasticity, composite action and collapse design of unreinforced shear wall panels in frames." Proceedings of the Institute of Civil Engineers (1978): 65(2): 381–441.

5. **Liauw TC & Kwan KH** "Unified plastic analysis for infilled frames." Journal of Structural Engineering (ASCE) (1985): 111(7): 1427–1448.

6. **Saneinejad A & Hobbs B** "Inelastic design of infilled frames." Journal of Structural Engineering (ASCE) (1995): 121(4): 634–650.

7. **"**Building cod requirements for structural plain concrete and commentary**"** (1989) **ACI 312.1-89** Am. Concrete Inst.(ACI), Detroit, Mich

8. **"**Building code requirements for masonry structures **"** (1988) **ACI 530-88/ASCE 5-88**, Am. Concrete Institute (ACI), Detroit, Mich

9. **Madan A, Reinhorn AM, Mander JB & Valles RE** "Modeling of masonry infill panels for structural analysis." Journal of Structural Engineering (ASCE) (1997) 123(10): 1295–1302.

10. **El-Dakhakhni WW, Elgaaly M & Hamid AA "**Three-Strut Model for Concrete Masonry-Infilled Steel Frames." Journal of Structural Engineering (2003) 129(2): 177-185.

11. **Flanagan, RD, Bennett RM, & Barclay GA** ''In-plane behavior of structural clay tile infilled frames.'' Journal of Structural Engineering (1999) 125(6), 590–599.

12. **Shing PB and Mehrabi AB "**Behavior and analysis of masonry-infilled frames." Prog. Struct. Engng Mater. (2002) 4:320–331.

13. **Bennet RM, Kurt AB, Flanagan RD "**Compressive properties of structural clay tile prisms.**"** Journal of Structural Engineering 1997: 123(7): 920-926

14. **Silicon Graphics Inc. "**OpenGL Programming Guide", Addison-Wesley Publishing Company, 1997.

15. **MSDN (Microsoft Developers Network)** "Microsoft Foundation Class Library", (2005) http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vcmfc98/html/ mfchm.asp

16. **Mckenna FT & Fenves GL "**Object-oriented finite element programming: frameworks for analysis, algorithms and parallel computing", PHD Dissertation, University of California, Berkeley, 1997.

17. **Rumbaugh, J, Blaha M, Premerhani W, Eddy F & Lorensen W** "Object-Oriented Modeling and Design" Prentice-Hall (1991), Englewood Cliffs, New Jersey 07632.

18. **Heyman J** "Plastic design of portal frames." Cambridge University Press (1957), Cambridge, England.

19. **Porter FL & Powell GH** ''Static and dynamic analysis of inelastic framed structures.'' Rep. No. EERC 71-3, Earthquake Engineering Research Center, (1971) University of California at Berkeley, California.

20. **Nigam, NC** ‘‘Yielding in framed structures under dynamic loads.’’ J. Eng. Mech. Div., (1970) 96(5), 687–709.

21. **El-Tawil S & Deierlein GG** ‘‘Stress-resultant plasticity for frame structures.’’ J. Eng. Mech. (1998), 124(12), 1360–1370.

22. **Liew JYR & Tang LK** ‘‘Advanced plastic-hinge analysis for the design of tubular space frames.’’ Eng. Struct. (2000), 22(7), 769–783.

23. **Walraven, JC** “The influence of depth on the shear strength of lightweight concrete beams without shear reinforcement.” (1978) TU-Delft Report no. 5-78-4, Stevin Laboratory, Delft University of Technology

24. **Meek JL, Lin LW** “Geometric and material nonlinear analysis of thin-walled beam–columns.” ASCE Journal of Structural Engineering (1990), 116(6), 1473–90.

25. **Bild S, Chen G, Trahair NS** “Out-of-plane strength of steel beams.” ASCE Journal of Structural Engineering (1992), 118(8), 1987–2003.

26. **Teh LH, Clarke MJ** “Plastic-zone analysis of 3D steel frames using beam elements.” ASCE Journal of Structural Engineering (1999), 125(11), 1328–37.

27. **Jiang XM, Chen H, Liew JYR “**Spread-of-plasticity analysis of three dimensional steel frames.**”** Journal of Constructional Steel Research 58 (2002), 193-212

28. **Neuenhofer A & Filippou FC** “Evaluation of nonlinear frame finite element models.” J. Struct. Eng. (1997) ASCE 123, 958–966.

29. **Taylor RL, Filippou FC, Saritas A, Auricchio F “**A mixed finite element method for beam and frame problems” Computational Mechanics 31 (2003), 192–203.

30. **Hognestad E**, "A study of Combined Bending and Axial Load in R.C. Members" University of Illinois Enginering Exp. Sta. Bull. No.399, (1951)

31. **Commend S, Zimmermann T "**Object-Oriented Nonlinear Finite Element Programming: a Primer**"** Advances in Engineering Software 32 (2001), 8, 611-628.

32. **Zimmermann T, Dubois-Pélerin Y, Bomme P** "Object-oriented finite element programming: I Governing principles" (1992) Computer Methods in Applied Mechanics and Engineering (98)

33. **Dubois-Pélerin Y, Zimmermann T, Bomme P** "Object-oriented finite element programming: II A prototype program in Smalltalk" (1992) Computer Methods in Applied Mechanics and Engineering (98).

34. **Lerin YD & Pegon P** "Improving modularity in object-oriented finite element programming" Commun. numer. methods engin., vol. 13, (1997) 193-198 .

35. **Wilson EL** "Three dimensional static and dynamic analysis of structures." Computers and structures Inc., (2000) Berkeley, California.

36. **Numerical applications with object oriented design**, "TCBI NumLib", (2005) http://plasimo.phys.tue.nl/TBCI/

37. **Polat, MU** "RCMOD – A Code for the Response Calculation of Reinforced Concrete Sections", Department of Civil Engineering, METU, 06531, Ankara, Turkey (2005)

38. **Bathe, KJ "**Finite Element Procedures**",** Prentice-Hall,(1996) New Jersey

39. **Pamin J, Bors R** "Simulation of crack spacing using a reinforced concrete model with an internal length parameter" (1998) Archive of Applied Mechanics (68) 613-625

40. **Computers & Structures Inc. "**SAP2000 – Structural Analysis Program" (2003), California Berkeley.

41. **Carlos AF** "Introduction to Finite Element Methods" Department of Aerospace Engineering Sciences and Center for Aerospace Structures University of Colorado (2004).

42. **Ersoy U "**Reinforced Concrete**"** Department of Civil Engineering, METU, 06531, Ankara, Turkey (2000).

APPENDIX A

USING THE PROGRAM

After locating necessary DLL files on the search path of the executable file, program can be run by double-clicking the file named Tez_Prog.Exe. Later user should click “Read from Input File” button from the toolbar of the program. Program will prompt for an input file in a browser window. If a valid input file is entered, geometry of the structure should appear on the screen. After seeing the structure on the screen, analysis can be performed by clicking “Analyze” button on the toolbar. Program will solve the system updating the view of the system after each loading step. After analysis is completed, user can walk through the load steps by clicking rewind and forward buttons on the toolbar. Strain of the frame and bar members can be investigated by right clicking on the members after the analysis is completed.

View of the system can be adjusted anytime with the following keys.

- Ctrl + Left Mouse Botton + Mouse Move : Rotates the model.
- Ctrl + Right Mouse Buttton + Mouse Move : Zooms the model.
- Shift + Left Mouse Botton + Mouse Move : Pans the model.

APPENDIX B

INPUT FILE FORMAT

Input file consists of keywords defining the types of the data immediately followed by the actual data in the next line(s). For example, to input number of supports of the structure as 2, one needs to write:

```
NumSupport
2
```

To state there are fixed supports on nodes 1 and 4, followings need to be written.

```
Supports
1 1 1 1 1 1 1
4 1 1 1 1 1 1
```

All keywords are listed with the definitions below. Under the definition of each keyword, type of data is given with the following convention. For integer numbers symbol *'#'* and real numbers symbol *'#.#'* is used. In addition to that Symbol *'b'* is used for the Boolean terms. Alternatively term *'string'* states that given input is a text. Each of these items starts with a number to refer in the definition. The block of symbols that remains between the brackets '<' and '>' indicates that same format is used for more than one times to define a multiple data. Using this convention example given previously for the supports can be represented as follows.

```
NumSupport
1#

Supports
<1# 2b 3b 4b 5b 6b 7b> x NumSupport
```

Definitions of the keywords are given in the order of appearance in the input file. This order may not be followed; however, related items should obey an order logically. For example number of supports should be given before the definition of the supports. It should be noted that units of the program is kN and m. Keywords that are used in the input file and their definitions are:

- **NSType:** Nonlinear solution algorithm used for the solution of the problem.

NSType
1#

1#

1 :Constant Stiffness

2 :Modified Newton Raphson

3 :Modified Newton Raphson that changes to full Newton Raphson when the equilibrium cannot be satisfied with the iterations.

4 :Modified Newton Raphson that can handle prescribed support displacements.

- **NumElemTypes:** Number of element types that is defined in the structure.

NumElemTypes
1#

- **ElemTypes:** Element type definitions that is used in the structure.

ElemTypes
<1string
2# (optional)
3string
4string or 4#.# 5#.# …**> x NumElemTypes**

1string

This is mathematical model of the element to be used; alternatives are:

FR3D :3D frame finite element

BAR3D :3D bar finite element

FRAMESUBELEMNO :Frame super element which is formed with equally spaced frame finite elements

FRAMESUB :Frame super element which is formed with specially discretized frame finite elements

WALL :Infill wall super element which is formed with two compressive struts.

2#

This is an optional field and should not be used with the element types other than FRAMESUBELEMNO and FRAMESUB. This variable is number of frame finite elements that will be used in the frame super element.

3string

Material model used for the element. This field is not used for WALL super element.

LINEAR :Linear material model for BAR3D, FR3D, FRAMESUBELEMNO and FRAMESUB.

NONLINEAR :Nonlinear material model of FR3D, BAR3D, FRAMESUBELEMNO and FRAMESUB

NONLINEARCP :Nonlinear material model of FR3D, FRAMESUBELEMNO and FRAMESUB.

4string

This is name of the input file to be read to form material model of the element type. This field is used for material model NONLINEARCP and NONLINEAR for frame finite elements and frame super elements.

4#.# 5#.# ...

These are real numbers which are used for forming the material model of linear and nonlinear 3D bar finite element, linear 3D frame finite element and infill wall super element.

For linear 3D bar finite element **4#.#** represents the modulus of elasticity of the bar and **5#.#** represents the area of the bar.

For nonlinear 3D bar finite element a series of stress and modulus of elasticity values are given to form multi-linear stress strain relationship of the bar.

For liner 3D frame finite element **4#.#, 5#.#, 6#.#, 7#.#** and **8#.#** are modulus of elasticity, moment of inertia about local y-y axis, moment of inertia about local z-z axis, torsional constant of the section and area of the section respectively.

For infill wall super element, **4#.#, 5#.#, 6#.#, 7#.#** and **8#.#** are modulus of elasticity of the unit masonry in vertical direction, strength of the unit masonry in vertical direction, modulus of elasticity of the unit masonry in horizontal direction, strength of the unit masonry in horizontal direction and thickness of the infill wall respectively.

- **NumNode:** Number of nodes of the structure

NumNode
1#

- **Coord:** Coordinates of the nodes

Coord
<1#.# 2#.# 3#.#> x NumNode

1#.#, 2#.# and 3#.# are x, y and z coordinates of the node respectively.

- **NumElem:** Number of elements that is used in the structure**.**

NumElem
1#

- **Connect:** Connectivity of the elements together with the element types

Connect
<1#
2# 3#**> x NumElem**

1# is the index of the element type used for the element. **2**# and **3**# are start and end nodes of the frame and bar elements.

An extra term **4**# is given next to **3**# for infill walls. The terms **2#, 3**# **and 4**# represents the indices of the frame elements bounding the wall. The bounding frames can either be entered clockwise or counter clockwise.

- **OrientFrame:** Vector defining the direction of the local y axis of the frame elements.

OrientFrame
<1#
2#.# 3#.# 4#.#**> x** (Number of frame finite elements plus frame super elements)

1# is index of the frame**, 2#.#, 3#.#** and **4#.#** defines the vector that is in local y direction of the frame.

- **NumSupport:** Number of supports exist on the structure.

NumSupport
1#

- **Support:** Location and types of the supports.

Support
<1# 2b 3b 4b 5b 6b 7b **> x NumSupport**

1# is the node index which the support exists. **2b**, **3b**, **4b**, **5b**, **6b** and **7b** are the restraints at dofs ux, uy, uz, θx θy and θz of the node respectively. If the value is 1 then the displacement or rotation of the degree of freedom is prevented.

- **NumLoad:** Number of point loads on the structure. Each **NumLoad** entry defines the number of point loads for different load cases. First **NumLoad** defines the number of point loads for load case 1, second defines for load case 2 etc...

NumLoad
1#

- **Load:** Definition of the point loads are given under the item. Each **Load** entry defines the point loads for different load cases. If there exists a loading on a degree of freedom which is restrained, than the loading is automatically recognized as prescribed displacement.

Load
<1# 2#.# 3#.# 4#.# 5#.# 6#.# 7#.#> x NumLoad

1# is the node number on which the load is applied**. 2#.#, 3#.#, 4#.#, 5#.#, 6#.#** and **7#.#** are the loading values on the dofs ux, uy, uz, θx θy and θz of the node respectively.

- **NumEdgeLoad:** Number of edge loads on elements of the structure. Each **NumEdgeLoad** entry defines the number of edge loads for different load cases. First **NumEdgeLoad** defines the number of edge loads for load case 1, second defines for load case 2 etc...

NumEdgeLoad
1#

- **EdgeLoad:** Definition of the edge loads are given under the item. Each **EdgeLoad** entry defines the edge loads for different load cases.

EdgeLoad
<1# 2#.# 3#.# 4#.#>

1# is the element index on which the load is applied**. 2#.#, 3#.#** and **4#.#** are the distributed load values in x, y and z direction.

- **NumNodeDisps:** Number of nodes whose displacements will be reported after each load step.

```
NumNodeDisps
1#
```

- **MonitDisps:** Displacements of these degrees of freedoms are reported in a file after each loading step.

```
MonitDisps
<1# 2b 3b 4b 5b 6b 7b> x NumNodeDisps
```

1# is the node number whose displacements will be monitored. **2b, 3b, 4b, 5b, 6b** and **7b** determine the monitored degree of freedoms of the node.

- **SeqLoad:** The items under this keyword define the sequence of loading of the structure.

```
SeqLoad
1#
<2#
3# 4#.# …> x 1#
```

1# is the number of sequence of loading. 2# is the number of iterations for the loading sequence. **3#** and **4#.#** are the index of the load case and weight of the load case that forms the sequence. Many load cases and weight of the load cases can be given to form a sequence. Combination of the given load cases by considering the load weights will be taken to find the total load of the sequence.

- **DimProblem:** This key defines the degree of freedoms that are active in global sense.

```
DimProblem
1b 2b 3b 4b 5b 6b
```

1b, 2b, 3b, 4b, 5b and 6b are translation degree of freedoms in x, y z direction and rotational degree of freedoms in x, y, z direction respectively. For instance a 2D problem can be solved in xz plane by defining **DimProblem** as:

```
DimProblem
1 0 1 0 1 0
```