

ANALYSIS OF ASSUMPTIONS MADE IN DESIGN OF REINFORCEMENT IN SLENDER
REINFORCED CONCRETE (TILT-UP) PANELS WITH OPENINGS

by

BRANDON SCHWABAUER

B.S., Kansas State University, 2010

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Architectural Engineering and Construction Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2010

Approved by:

Major Professor
Kimberly Waggle Kramer, P.E., S.E.

Abstract

This report uses and references (Analysis of Vertical Reinforcement in Slender Reinforced Concrete (Tilt-up) Panels with Openings & Subject to Varying Wind Pressures) (Bartels, 2010) to investigate the design philosophy and assumptions used in Section 14.8 of the ACI 318-08 (ACI Committee 318, 2008). The design philosophy and assumptions are analyzed to determine the applicability and accuracy of Section 14.8 of the ACI 318-08 (ACI Committee 318, 2008) to the design and analysis of slender concrete panels with openings. Special emphasis is placed on identifying and quantifying the degree of effect that each assumption has on the final design of the panel. These topics include stress distribution around openings, the effect of varying stiffness of the member on the $P-\delta$ effect, stiffness variations due to workmanship and tolerances, and the effect of axial load on the stiffness of the member. This is accomplished through the use of specially designed computer analyses that isolate an assumption or effect to determine its impact on the final design.

This study shows that two-way effects are almost non-existent, the portion of the panel above the opening has very little effect on the $P-\delta$ effects, the code specified reduction in bending stiffness due to workmanship and tolerances appear to be appropriate, and the effective area of reinforcement overestimates the stiffness of the panel.

Table of Contents

List of Figures	v
List of Tables	vi
List of Terms	vii
Acknowledgements	viii
§1.0. Introduction	1
§2.0. Finite Element Modeling (FEM)	4
§2.1. Finite Element Process	4
§2.1.1. Discretization	5
§2.1.2. Idealization – Lump stiffness into the nodes	6
§2.1.3. Element Stiffness Matrix	7
§2.1.4. Assembly	9
§2.1.5. Skyline Storage	11
§2.1.6. Boundary Conditions	13
§2.1.7. Node Reordering	15
§2.1.8. Solver	15
§2.1.8.1. Gaussian Elimination	16
§2.1.8.2. Skyline Solver	17
§2.1.8.3. Sparse Solvers	18
§2.1.9. Element Forces	19
§2.2. Types of Analysis	19
§2.2.1. Static	19
§2.2.2. Dynamic	20
§2.2.3. Linear Elastic	20
§2.2.4. Geometrically Nonlinear	21
§2.2.4.1. P- Δ Effect	21
§2.2.4.2. P- δ Effect	21
§2.2.5. Materially Nonlinear	22
§2.2.5.1. Steel	23
§2.2.5.2. Concrete	23

§2.3. Elements Used and their Properties	24
§2.3.1. Bar Element	24
§2.3.2. Bernoulli-Euler Plane Beam	25
§2.3.3. Two & Three Dimensional Elements	28
§2.3.3.1. Displacement Based Elements	28
§2.3.3.2. Isoparametric Elements	30
§2.3.4. Plane Stress Quad	33
§2.3.5. Plate	33
§2.3.6. Shell Element	34
§2.3.7. Hexahedron (Brick) Element	34
§3.0. Results	36
§3.1. One-Way Bending Assumption Test	36
§3.2. Constant Bending Stiffness Along Member Assumption Test	39
§3.3. Bending Stiffness Reduction Factor Assumption Test	40
§3.4. Effect of Axial Force on Bending Stiffness Test	42
§4.0. Conclusion	44
Bibliography	45
Appendix A - Panel Loading (Bartels, 2010)	46
Appendix B - Stiffness Matrix Assembly MS Excel VBA Code	51
Appendix C - Stiffness Matrix Formulation MS Excel VBA Code	57
Appendix D - Skymatrix Solver MS Excel VBA Code	61
Appendix E - Gaussian Elimination Solver with Pivoting MS Excel VBA Code	63

List of Figures

Figure 1-1 Panel with Opening Design Strip Diagram.....	2
Figure 2-1 Discretization of a Frame	5
Figure 2-2 Discretization of a Tilt-up Panel with an Opening.....	6
Figure 2-3 Idealization of a Beam-Column	6
Figure 2-4 Truss or Bar Element	7
Figure 2-5 Assembly Process.....	9
Figure 2-6 Skyline Matrix Representation.....	12
Figure 2-7 Skyline Matrix Representation Numerical Example.....	12
Figure 2-8 Boundary Conditions Example	13
Figure 2-9 P- Δ Effect.....	21
Figure 2-10 P- δ Effect.....	22
Figure 2-11 Concrete Stress Strain Plot.....	23
Figure 2-12 Euler Bernoulli Plane Beam Element	25
Figure 2-13 Natural Coordinates	31
Figure 2-14 Plane Stress Quadrilateral	33
Figure 2-15 Quadrilateral Plate.....	34
Figure 2-16 Shell Element	34
Figure 2-17 Hexahedron (Brick) Element	35
Figure 3-1 Panel Geometry	36
Figure 3-2 Stress Distribution in a 40' x 24' Panel with a 12' x 12' Opening	38
Figure 3-3 Constant Bending Stiffness Along Member Test Results.....	40
Figure 3-4 Bending Stiffness Workmanship Reduction Results	41
Figure 3-5 Effect of Axial Force on Bending Stiffness	43

List of Tables

Table 3-1 Constant Bending Stiffness Along Member Test Results	39
--	----

List of Terms

ACI - American Concrete Institute

ASCE - American Society of Civil Engineers

A_{se} - Effective Area of Reinforcement

DOF - Degrees of Freedom

ELFP - Equivalent Lateral Force Procedure

FEM - Finite Element Method

I_{cr} - Cracked Moment of Inertia

I_{eff} - Effective Moment of Inertia

MS - Microsoft

VBA - Visual Basic for Applications

Acknowledgements

I would like to thank the members of my committee, Professor Kimberly Kramer, P.E., S.E., Professor Sutton Stephens, Ph.D., P.E., S.E., and Professor David Fritchen for their guidance and support during the writing of this paper.

§1.0. Introduction

This report uses and references (Analysis of Vertical Reinforcement in Slender Reinforced Concrete (Tilt-up) Panels with Openings & Subject to Varying Wind Pressures) (Bartels, 2010) to investigate the design philosophy and assumptions used in Section 14.8 of the ACI 318-08 (ACI Committee 318, 2008). The design philosophy and assumptions will be analyzed to determine the applicability and accuracy of Section 14.8 of the ACI 318-08 (ACI Committee 318, 2008) to the design and analysis of slender concrete panels with openings. Special emphasis is placed on identifying and quantifying the degree of effect that each design philosophy or assumption has on the final design of the panel. These topics include stress distribution around openings, the effect of varying stiffness of the member on the $P-\delta$ effect, stiffness variations due to workmanship and tolerances, and the effect of axial load on the stiffness of the member. This will be accomplished through the use of specially designed computer analyses that isolate an assumption, effect, or design philosophy to determine its impact on the final design. Although Section 14.8 of the ACI 318-08 makes many assumptions, only the assumptions that apply directly to the design of tilt-up panels with openings are investigated.

The first main assumption is how stresses distribute around openings. The ACI 318-08 does not address how to design tilt-up panels with openings. However, in Section 14.8.2.2 of the ACI 318-08 states “The cross section shall be constant over the height of the panel”. Clearly, if the panel has openings, then it does not fit this requirement and the panel is not eligible to be designed using Section 14.8. However, the ACI318-08 commentary states “Panels that have windows or other large openings are not considered to have constant cross section over the height of the panel. Such walls are to be designed taking into account the effects of openings. (ACI Committee 318, 2008)” How are these effects taken into account? The Design Guide for Tilt-Up Concrete Panels (ACI Committee 551, 2010) suggests that these effects can be “approximated by a simple one-dimensional strip analysis that provides accuracy and economy for most designs.” This is illustrated in Figure 1-1 and would imply that only the shaded regions provide bending moment resistance. An analysis of this assumption is provided in §3.0 Results.

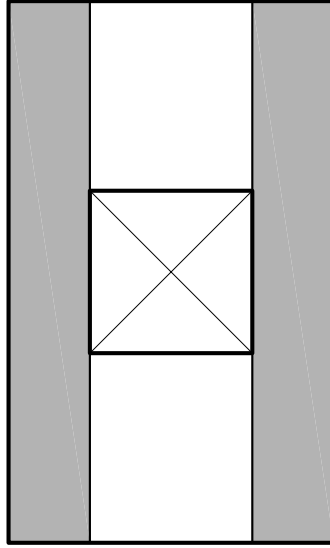


Figure 1-1 Panel with Opening Design Strip Diagram

The second major assumption is related to the first. Following the assumption that only the full height strip (shaded region in Figure 1-1) provides bending moment resistance, the assumption is made that the deflection calculation should only include the stiffness of the panel leg. The effectiveness of this assumption will be determined in §3.2 Constant Bending Stiffness Along Member Assumption Test. The deflection calculation is important for these panels because it has a direct impact on the P- δ effect discussed in §2.2.4.2. The P- δ effect increases the design moments by a significant amount due to the slenderness of the panels.

The third assumption is related to the code requiring a 0.75 factor on the cracked moment of inertia “to account for variations in material properties and workmanship. (ACI Committee 551, 2010)” In addition, “small variations in the position of reinforcement in the concrete section will have a significant effect on the strength of the wall panel and bending stiffness properties.” (ACI Committee 551, 2010) This assumption will be tested by varying the placement of reinforcement up to the code allowed tolerances of the section.

The fourth major design philosophy or assumption made in Section 14.8 of the ACI 318-08 is the effective area of reinforcement, A_{se} . A_{se} is an adjustment to the area of tension reinforcement, A_s , to account for the effect of axial load on the section. A_{se} is defined as:

$A_{se} = A_s + \frac{P_u}{f_y} \left(\frac{h}{2d} \right)$. The results of this equation will be compared with more exact methods to determine the equation’s validity throughout a range of axial forces.

Several methods can be used to analyze these assumptions. These include a moment curvature analysis, the alternative design provisions for slender walls in Section 14.8 of the ACI 318-08, and the finite element method.

The rigorous moment curvature analysis can be used to analyze two of the four design philosophies or assumptions stated previously; the exception is for $P-\delta$ effects and 2-way effects. Moment curvature analysis cannot account for $P-\delta$ effects and 2-way effects because it analyzes the structure at only one part of the member. $P-\delta$ effects and 2-way effects must take into account the whole length/area of the member. Therefore, moment curvature analysis will only be used in calculating the effect of tolerances on the bending stiffness and the effects of axial force on the bending stiffness.

The alternative design provisions for slender walls in Section 14.8 of the ACI 318 is used as a baseline or a control analysis method because this is the method employed by Bartels (Bartels, 2010). This method is important as a baseline analysis because the assumptions being tested come from this method. These calculations will not be performed for this report; instead Bartels (Bartels, 2010) will be referenced as a comparison to this report's results.

The finite element method is capable of accounting for all four of the assumptions stated above and is especially useful in analyzing two-way bending effects and varying stiffness along the length of the member. Along with the advantage of being able to analyze the four design philosophies or assumptions, the finite element method contains a few pitfalls that must be avoided.

The first caution about using a commercial finite element software is the solution calculations cannot be traced easily. This is similar to other computer based engineering solutions. Inputs can be verified but what the program performs with those inputs comes from the programmer. Without coding the program yourself, the results must be taken in faith that the programmer made no mistakes in the program. Therefore, it is important to confirm the results using a different method or approximation to make sure the results are accurate.

The second caution with the finite element method is that it is an upperbound method - the solution should converge to the “exact” solution, but it approaches this level from the unconservative side. This is especially true for “coarse” meshes. Therefore, special attention must be placed on a proper “meshing” to provide accurate results.

§2.0. Finite Element Modeling (FEM)

Finite element analysis in its broadest sense is a method of analysis that can be applied to solving differential equations, heat transfer problems, and structural analysis problems. However, in this report, only structural applications are discussed. Finite element analysis is a very useful analysis technique especially for problems having no conventional solution. For instance, FEM originates from aircraft analysis because it would be extremely difficult if not impossible to design modern aircraft by conventional methods with the precision and efficiency provided by FEM. However, the designer must be careful to model the problem correctly to obtain suitable results. Blind acceptance of what the computer outputs is not advised. It must be checked against engineering judgment and knowledge. Some of the topics and ideas involved in finite element analysis are discussed, so it can be properly applied to real world structural problems. This will by no means be an exhaustive explanation of the subject, merely an overview for practicing engineers using FEM software.

This chapter explores finite element analysis in several different areas. First, the basics of the finite element analysis is presented. While no prior FEM knowledge is required for this chapter, some working knowledge of matrices and advanced mathematics topics is assumed; for more detailed explanations of matrix operations and advanced mathematics, reference (Bathe, 2006) and (Felippa, 2004). Next, different types of analysis are covered; These include static, dynamic, linear elastic, materially nonlinear, and geometrically nonlinear. All of these types of analysis have their particular place in the designer's repertoire. Some problems might require a geometrically nonlinear dynamic analysis whereas for others, a static linear elastic analysis might suffice. Third, several different types of elements will be presented. These are by no means the only elements that can be derived; they are only a tasting of some of the most commonly used elements.

§2.1. Finite Element Process

The finite element method is a process that has many steps and details. In addition, steps can often be rearranged, added, removed, or altered to suit the needs of the analysis. Therefore, some of the more basic and pertinent steps needed for almost every analysis are presented.

These steps include discretization, idealization, assembly, skyline storage, boundary conditions, node reordering, solving, and element forces.

§2.1.1. Discretization

The first step in the finite element process is discretization. Discretization is the process of dividing a continuous problem into distinct parts. For example, consider the frame shown in Figure 2-1(a). Although this is a relatively simply problem, it would be difficult and time consuming to derive a closed form solution to this structural analysis problem. The solution is made much easier and simpler by separating the frame into individual members as shown in Figure 2-1(b), solving for each individual member, assembling them all together, and then solving for the whole frame. This difference in difficulty of solution is much more apparent in larger systems, where a classical solution is more complex to derive.

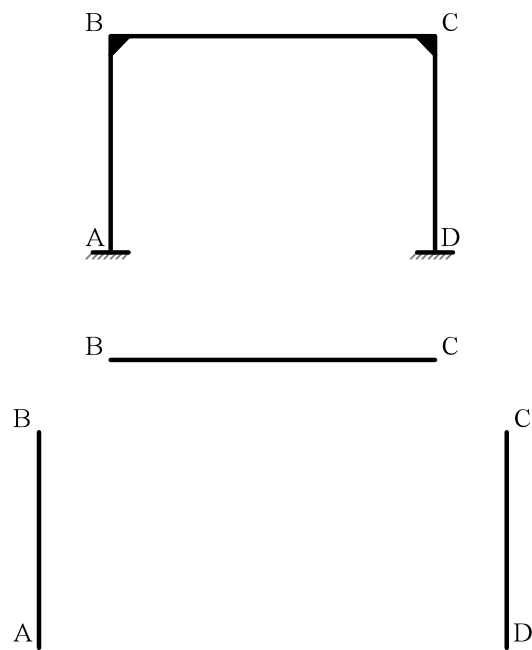


Figure 2-1 Discretization of a Frame

One of these types of problems is when considering the analysis of a tilt-up wall panel with an opening as shown in Figure 2-2(a). It would be complex to solve this problem without discretizing it similar to as shown in Figure 2-2(b). In addition, this discretization makes stress calculations and distributions around the opening more accurate.

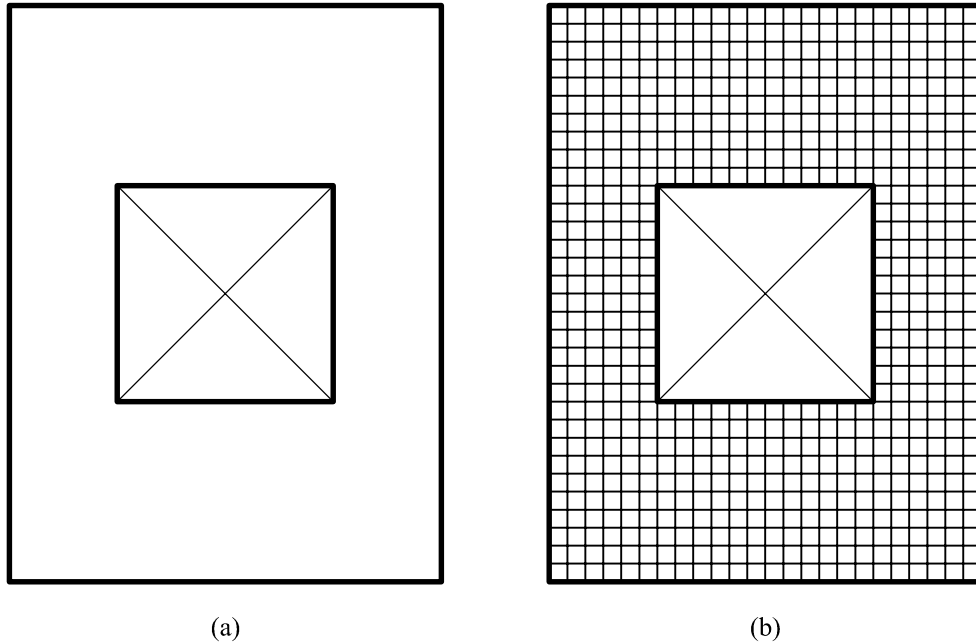


Figure 2-2 Discretization of a Tilt-up Panel with an Opening

§2.1.2. Idealization – Lump stiffness into the nodes

Another step in finite element analysis is idealization. Idealization is the process of converting a realistic structure into something more basic. For example, in Figure 2-3(b) a “realistic” (it has thickness) beam-column in contact with the fixed ground, is idealized as a line with perfectly pinned end conditions as shown in Figure 2-3(a). The intent of idealization is to provide an accurate representation of the realistic conditions of the problem. While this idealization holds several assumptions about the realistic conditions, accurate results can be obtained by applying some engineering judgment and knowledge to the problem.

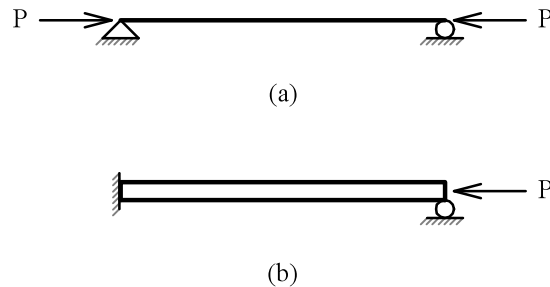


Figure 2-3 Idealization of a Beam-Column

§2.1.3. Element Stiffness Matrix

After the structure is both properly discretized and idealized, the next step is to compute the stiffness matrix for all elements in the analysis. The stiffness matrix, \mathbf{K} , is a matrix relating “the element end forces to corresponding unit element end displacements” (Bathe, 2006). For one dimensional bars and beams, this relationship can be derived exactly according to the beam bending theory of mechanics of materials (the so called direct stiffness method). However, for two dimensional or three dimensional elements, this relationship cannot be derived exactly and must be approximated. This relationship is shown in general below:

$$\mathbf{K}\mathbf{U} = \mathbf{P}$$

Where \mathbf{K} is the stiffness matrix, \mathbf{U} is the displacement vector, and \mathbf{P} is the force vector. As a more practical example consider a truss or bar element as shown in Figure 2-4.

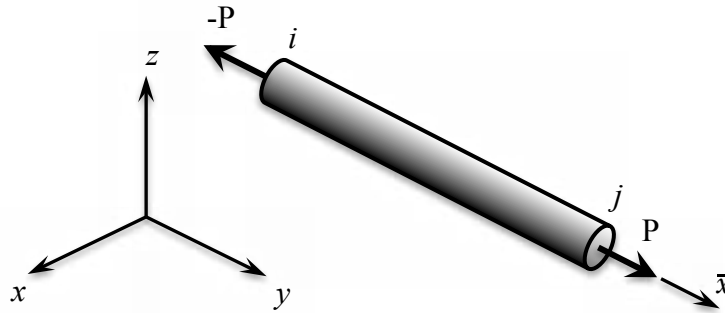


Figure 2-4 Truss or Bar Element

Each node can move either in the positive or negative direction along its own coordinate system. Therefore, this system has two degrees of freedom (DOF) and can be written as:

$$\bar{p}_{\bar{x}i} = \frac{AE}{L} \bar{u}_i - \frac{AE}{L} \bar{u}_j$$

$$\bar{p}_{\bar{x}j} = -\frac{AE}{L} \bar{u}_i + \frac{AE}{L} \bar{u}_j$$

Or, in matrix form as:

$$\begin{Bmatrix} \bar{p}_{\bar{x}i} \\ \bar{p}_{\bar{x}j} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \bar{u}_i \\ \bar{u}_j \end{Bmatrix}$$

Applying this one dimensional element to a two dimensional coordinate system yields the following:

$$\begin{Bmatrix} \bar{p}_{\bar{x}i} \\ \bar{p}_{\bar{y}i} \\ \bar{p}_{\bar{x}j} \\ \bar{p}_{\bar{y}j} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \bar{u}_i \\ \bar{v}_i \\ \bar{u}_j \\ \bar{v}_j \end{Bmatrix}$$

This element stiffness matrix is in its own local coordinates. To solve for the structure as a whole, the elements need to be in the same coordinate system. This can be accomplished through the use of a transformation matrix as shown:

$$\begin{Bmatrix} \bar{u}_i \\ \bar{v}_i \\ \bar{v}_j \\ \bar{v}_j \end{Bmatrix} = \begin{bmatrix} \cos \Phi & \sin \Phi & 0 & 0 \\ -\sin \Phi & \cos \Phi & 0 & 0 \\ 0 & 0 & \cos \Phi & \sin \Phi \\ 0 & 0 & -\sin \Phi & \cos \Phi \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

where Φ is the angle between the member and the global coordinate system. To get the stiffness matrix in global coordinates, the following must be calculated:

$$\mathbf{k} = \mathbf{T}^T \bar{\mathbf{k}} \mathbf{T}$$

For truss or bar elements, this operation can be performed symbolically to arrive at the element stiffness matrix in global coordinates:

$$[k] = \frac{AE}{L} \begin{bmatrix} \cos^2 \Phi & \sin \Phi \cos \Phi & -\cos^2 \Phi & \sin \Phi \cos \Phi \\ \sin \Phi \cos \Phi & \sin^2 \Phi & \sin \Phi \cos \Phi & -\sin^2 \Phi \\ -\cos^2 \Phi & \sin \Phi \cos \Phi & \cos^2 \Phi & \sin \Phi \cos \Phi \\ \sin \Phi \cos \Phi & -\sin^2 \Phi & \sin \Phi \cos \Phi & \sin^2 \Phi \end{bmatrix}$$

The derivation of this truss or bar element is critical to the development of other more complex elements. The beam and frame elements can relatively easily be extended from this truss or bar element. In addition, the concept of the stiffness matrix and “lumping” the stiffness of an element at its nodes leads to the derivation of the finite element method.

§2.1.4. Assembly

The assembly process takes the individual element stiffness matrices and assembles or compiles them into the master global stiffness matrix in their proper place. For example, where two frame elements are connected at a joint, their stiffness terms are added together. In this way, the effects of both members are taken into account.

As an example, consider the structure in Figure 2-5. The assembled master global stiffness matrix for the whole structure is calculated by assembling the individual elements together. This process is described in detail below:

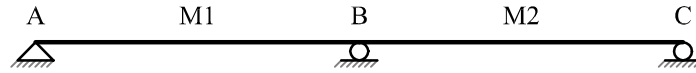


Figure 2-5 Assembly Process

The stiffness matrix for a plane beam element is:

$$[k] = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

From observation, there are 6 DOF's present in this example problem. Therefore,

$$\begin{Bmatrix} p_{v1} \\ p_{\theta1} \\ p_{v2} \\ p_{\theta2} \\ p_{v3} \\ p_{\theta3} \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \begin{Bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \\ v_3 \\ \theta_3 \end{Bmatrix}$$

Evaluating the first element with the stiffness element terms in their global locations yields:

$$[k]_1 = \begin{matrix} & \begin{matrix} v_1 & \theta_1 & v_2 & \theta_2 & v_3 & \theta_3 \end{matrix} \\ \begin{matrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & 0 \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & 0 \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & 0 \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{matrix}$$

Evaluating the second element with the stiffness element terms in their global locations yields:

$$[k]_2 = \begin{matrix} & \begin{matrix} v_1 & \theta_1 & v_2 & \theta_2 & v_3 & \theta_3 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ 0 & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{matrix} \end{matrix}$$

Adding these two matrices yields the global stiffness matrix for the whole structure:

$$[k] = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & 0 \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & 0 \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} + \frac{12EI}{L^3} & \frac{6EI}{L^2} - \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & \frac{6EI}{L^2} - \frac{6EI}{L^2} & \frac{4EI}{L} + \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ 0 & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

Reducing yields:

$$[k] = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & 0 \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & 0 \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{24EI}{L^3} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{8EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ 0 & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

In practice, it is advantageous to perform the assembly at the same time as the element stiffness matrix formulations because the element stiffness matrices do not all need to be stored in computer memory prior to assembly. This greatly reduces the memory requirements of the finite element analysis. For a VBA code example of this, see Appendix B - Stiffness Matrix Assembly MS Excel VBA Code.

§2.1.5. Skyline Storage

The skyline storage scheme is a very important development in finite element analysis. It is suitable for small to large size problems (however, other methods such as sparse methods are more appropriate for extremely large problems). Skyline storage allows for a great reduction in memory requirements because only the nonzero terms need to be stored in computer memory. In

addition, because the stiffness matrix is always positive definite symmetric, only half of the skyline matrix must be stored. These concepts are shown in Figure 2-6.

$$[k] = \begin{bmatrix} X & X & 0 & X & 0 \\ & X & 0 & 0 & 0 \\ & & X & X & X \\ & sym & & X & X \\ & & & & X \end{bmatrix}$$

$$[k] = [X \ X \ X \ X \ X \ X \ \dots \ X]$$

Figure 2-6 Skyline Matrix Representation

As can be seen in Figure 2-6, the elements for each column from the diagonal terms until the last nonzero term are the only terms stored. The skyline is illustrated by the line running above the last nonzero terms in the column. These columns are then compiled serially into a one-dimensional array as shown in Figure 2-6. In addition, to be able to access the entries of the skyline matrix, the locations of the diagonal elements of the matrix must be stored in the Global Skyline Diagonal Location Table (GSDLT). A numerical example is worked out in Figure 2-7:

$$[k] = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ & 7 & 8 & 0 & 2 \\ & & 4 & 5 & 0 \\ & sym & & 1 & 6 \\ & & & & 2 \end{bmatrix}$$

$$[k] = [3 \ 7 \ 1 \ 4 \ 8 \ 1 \ 5 \ 2 \ 6 \ 0 \ 2]$$

$$GSDLT = [1 \ 2 \ 4 \ 6 \ 8]$$

Figure 2-7 Skyline Matrix Representation Numerical Example

These savings are minimal for this example. However, consider an analysis with 1000's of DOF's. This can make a HUGE difference in storage requirements for the matrix. In addition, this type of storage lends itself to speedy solutions because operations do not have to be performed on entries outside of the skyline. For a VBA code based example of this type of

storage, see Appendix B - Stiffness Matrix Assembly MS Excel VBA Code and Appendix D - Skymatrix Solver MS Excel VBA Code.

§2.1.6. Boundary Conditions

The boundary conditions of a problem describe the end fixity conditions for the problem. For example, consider the beam in Figure 2-8. The right end of the beam consists of a roller that resists any and all vertical deflection and does not provide any moment resistance. On the left side there is a fixed roller that resists moment and does not resist vertical deflection from the boundary conditions.



Figure 2-8 Boundary Conditions Example

To solve this problem, we must first assemble the global stiffness matrix as shown below:

$$\begin{Bmatrix} p_{v1} \\ p_{\theta1} \\ p_{v2} \\ p_{\theta2} \end{Bmatrix} = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ \frac{12EI}{L^3} & \frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ -\frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{Bmatrix}$$

Once the stiffness matrix is assembled, we must apply the loads and boundary conditions.

This includes setting

$$\begin{Bmatrix} p_{v1} \\ p_{\theta1} \\ p_{v2} \\ p_{\theta2} \end{Bmatrix} = \begin{Bmatrix} -1 \\ p_{\theta1} \\ p_{v2} \\ 0 \end{Bmatrix}$$

which means that a unit load is applied in the downward direction at the left node and there is no load applied at the other DOF's. The variables $p_{\theta1}$ and p_{v2} are left because they represent the reactions and will be solved in the following steps. The known displacements must all be applied at this point resulting in:

$$\begin{Bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{Bmatrix} = \begin{Bmatrix} v_1 \\ 0 \\ 0 \\ \theta_2 \end{Bmatrix}$$

This shows that no rotation is allowed at the first node and no vertical displacement is allowed at the second node. Leaving the variables v_1 and θ_2 in the equation means they will get solved for in the coming steps. So applying these to the complete global stiffness matrix yields:

$$\begin{Bmatrix} -1 \\ p_{\theta_1} \\ p_{v_2} \\ 0 \end{Bmatrix} = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ \frac{12EI}{L^3} & \frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ -\frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} v_1 \\ 0 \\ 0 \\ \theta_2 \end{Bmatrix}$$

To solve for the displacements, we need to pull out the equations containing a variable in the displacement matrix. This yields the following:

$$\begin{Bmatrix} -1 \\ 0 \end{Bmatrix} = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} v_1 \\ \theta_2 \end{Bmatrix}$$

This can be solved for the displacements using any method for a system of equations. Once v_1 and θ_2 are known, the other two variables, p_{θ_1} and p_{v_2} can be solved using the following:

$$\begin{Bmatrix} p_{\theta_1} \\ p_{v_2} \end{Bmatrix} = \begin{bmatrix} \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ \frac{12EI}{L^3} & \frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \end{bmatrix} \begin{Bmatrix} v_1 \\ 0 \\ 0 \\ \theta_2 \end{Bmatrix}$$

This method works well for a hand solution of a relatively small solution. It involves rearranging equations and extracting and rebuilding the needed equations and matrices. On a computer, this extraction and rebuilding are very expensive in terms of both processing time and memory requirements. Therefore, a more advantageous solution has been developed. This consists of simply zeroing out the columns relating to the boundary conditions, and then adding a 1 to the diagonal to avoid a singular solution. For the previous example, this would yield:

$$\begin{Bmatrix} -1 \\ p_{\theta_1} \\ p_{v_2} \\ 0 \end{Bmatrix} = \begin{bmatrix} \frac{12EI}{L^3} & 0 & 0 & \frac{6EI}{L^2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{6EI}{L^2} & 0 & 0 & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} v_1 \\ 0 \\ 0 \\ \theta_2 \end{Bmatrix}$$

The solution of the displacements will be exactly the same as calculated previously.

§2.1.7. Node Reordering

Node reordering is an important step in finite element analysis, especially for large problems. Node reordering is the process of renumbering (or reordering) the node (and DOF's) to provide a smaller bandwidth for the global stiffness matrix. This is important for several reasons. First, it reduces the memory requirements of the global stiffness matrix. This is due to more of the terms in the global stiffness matrix being moved closer to the diagonal of the matrix. This means that fewer zeros must be stored inside of the global skyline stiffness matrix. Fewer terms in the global skyline stiffness matrix leads to the second benefit of node reordering: solution speed. As the bandwidth of the skyline matrix is reduced, its solution speed increases. This speed increase occurs because fewer terms need to be calculated inside of the skyline matrix. Therefore, this process can lead to much faster solutions; especially nonlinear solutions since the node reordering only needs to be done once for a problem whereas the solution needs to be repeated several times until convergence is met.

Although several bandwidth reduction algorithms exist, the details of these methods are not critical to a basic understanding of finite element analysis. The important thing to remember is node reordering or renumbering can lead to a more efficient solution.

§2.1.8. Solver

The solver is one of the most important parts of the finite element process because of the need for fast and accurate solutions. The solver has the task of solving for the displacements of a set of simultaneous equations. In a commercial finite element software package, 1000s upon 1000s of equations (or DOFs) must be solved simultaneously. This can be a very time consuming task and often is one of the longest parts of the solution time. This can be especially

true for nonlinear solutions or problems with multiple load cases. Additionally, the solution must be accurate. If the solution is not accurate, but is fast, what good is it to the designer?

Several methods exist that attempt to account for the issues stated above. Each of these methods is suited to a different type or size of problems. The methods most commonly used include: Gaussian elimination, skyline solvers, sparse solvers, and iterative solvers. The former three methods are all related and have evolved from the same methodology. The latter method, iterative solvers, uses a different methodology in its solution.

§2.1.8.1. *Gaussian Elimination*

Gaussian elimination is the basis for most of the methods of solving simultaneous equations. It is the process of adding and subtracting equations together to eliminate unknowns from the equations. For example, consider the system below:

$$x + 2y = 5$$

$$-x + 3y = 7$$

Adding these together yields:

$$5y = 12$$

And therefore, $y = 12/5$.

In this way, the solution to almost any size problem can be calculated (although other methods become more efficient as the size of the matrix becomes large). This solution can be divided into two main steps: forward reduction and back substitution.

The first step in the process is forward reduction. The purpose of forward reduction is to reduce the stiffness matrix to an upper triangular matrix (including the diagonal). This step is important because it results in the solution of the last variable in our system. It is also a convenient form for solving the other variables. Consider the general form of a system of equations given below:

$$\begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix}$$

This would reduce to:

$$\begin{bmatrix} 1 & X' & X' \\ 0 & 1 & X' \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} a' \\ b' \\ c' \end{Bmatrix}$$

In this example, $u_3 = c'$. The prime indication in the above indicates that the term has been reduced during the process of forward reduction.

In addition to adding or subtracting equations together, equations can also swap positions in the system. This is often referred to as pivoting. Pivoting is not always required for numerical accuracy; for example symmetrical positive definitive matrices (stiffness matrices are positive definite in linear elastic analysis) do not require pivoting. However, in other cases, pivoting might be required to provide numerical stability and accuracy.

The next step after this is back substitution. This is the process of plugging the variables we know the solution to into the previous equations to solve for one more variable. For example, the back substitution of the second equation in the above would look like:

$$0 + u_2 + X' * c' = b'$$

and

$$u_2 = b' - X' * c'$$

This can be repeated for the remaining variables to obtain the complete solution to the problem. A computer algorithm demonstrating this is written in Visual Basic for Applications (VBA) for Microsoft Excel is included in Appendix E - Gaussian Elimination Solver with Pivoting MS Excel VBA Code.

§2.1.8.2. Skyline Solver

A skyline solver is a solver using skyline storage in carrying out its solution. Several different variations of a skyline solver can be put to use most of which are variations of Gaussian elimination. One such variation is Crout elimination. In Crout elimination, “one column at a time is completely factorized, beginning with the first column and not involving subsequent columns. (Hughes, 2000)” Additionally, Crout elimination can be performed completely within the skyline matrix thereby providing a convenient solution for a skyline solver algorithm. Because there is no need for storage outside of the skyline matrix, it is also convenient, beneficial, and possible to overwrite the stiffness matrix with the factorization of itself. This greatly reduces the storage requirements of the solution because only one copy of the stiffness

matrix must be stored. The storage requirements are also reduced further because for linear elastic analysis, the stiffness matrix is symmetrical resulting in only half of the skyline matrix needing to be stored. The steps in Crout elimination include the following (for a stiffness matrix A_{ij} and load vector b_j) (Hughes, 2000):

1. **Factorization** - The algorithm for factorizing A_{ij} is:

For $j = 1$ to n ,

$$L_{ji} = A_{ij} - \sum_{k=1}^{i-1} U_{ki} L_{jk}, \quad 1 \leq i \leq j-1$$

$$U_{ij} = \frac{L_{ji}}{D_{ii}}$$

$$D_{jj} = A_{jj} - \sum_{i=1}^{j-1} L_{ji} U_{ij}$$

where \mathbf{U} is the upper triangular matrix and \mathbf{D} is the diagonal matrix from \mathbf{A} .

These matrices, \mathbf{U} and \mathbf{D} overwrite \mathbf{A} .

2. **Forward Reduction** - The algorithm for forward reduction is:

$$b_j \leftarrow b_j - \sum_{i=1}^{j-1} A_{ij} b_i, \quad 2 \leq j \leq n$$

3. **Diagonal Scaling** - The algorithm for diagonal scaling is:

$$b_j \leftarrow \frac{b_j}{A_{jj}}, \quad 1 \leq j \leq n$$

4. **Back Substitution** - The algorithm for diagonal scaling is:

For $j = n, n-1, \dots, 2$

For $i = 1, 2, \dots, j-1$

$$b_i \leftarrow b_i - A_{ij} b_j$$

An implementation of this algorithm written in Microsoft Excel VBA is included in Appendix D - Skymatrix Solver MS Excel VBA Code.

§2.1.8.3. Sparse Solvers

Sparse solvers work similarly to skyline solvers by only storing non-zero terms. For example, a skyline matrix might contain zeros within its data structure; a sparse solver would not

store these values because they are not needed and just take up extra space. These types of solvers can be very beneficial for massive problems in which a large number of simultaneous equations need to be solved. But, the matrix needs to be sparse; meaning that it mostly consists of zeros with nonzero values distributed fairly evenly throughout. The logic in these types of solvers becomes much more complicated. Therefore, they will not be presented here.

§2.1.9. *Element Forces*

The next step in the finite element process is determining the element forces and stresses. This is a very important step and is the ultimate goal of the analysis. With these forces and stresses, the designer can determine the best member for the problem. The element forces and stresses are calculated differently for each element, and will therefore not be covered explicitly in this section. However, it is important to note the process in which this happens. After the global displacements have been calculated at each node, the finite element program must loop through all of the elements and plug in these displacements in order to determine the forces and stresses. This is important because if we are only interested in a select portion of a rather large analysis, only the part we are interested in must be calculated. This can save precious time in which the designer would otherwise be waiting for results.

§2.2. Types of Analysis

In any structural analysis method, several different types of analysis exist. These different types of analysis all have their own strengths and weaknesses which makes them best suited for certain types of problems. In the following sections, several different types of analysis will be discussed including their strengths, their weaknesses, and the type of problem best suits them.

§2.2.1. *Static*

A static analysis is an analysis using only loads that do not vary with time. This type of analysis is widely used for most structural engineering problems. Its relative simplicity, accurate results, and code acceptance make it a valuable tool in the engineers repertoire. For example, this method is used in conjunction with the Equivalent Lateral Force Procedure (ELFP) of the (ASCE 7-05 Minimum Design Loads for Buildings and Other Structures, 2006). This provides acceptable results even though as the name implies, the equivalent lateral force procedure

“converts” dynamic loads (such as wind and seismic loads) to equivalent static loads. However, this static analysis method has several limitations based on the geometry, location, and loads applied to the building as outlined in the (ASCE 7-05 Minimum Design Loads for Buildings and Other Structures, 2006). When these limitations are encountered (things such as soft stories, horizontal irregularities, or vertical irregularities) a more complex analysis must be used - dynamic analysis.

§2.2.2. *Dynamic*

A dynamic analysis includes the effects of loads varying over time. It is a much more complex analysis used for problems where a static analysis is not appropriate. In addition, a dynamic analysis increases solution time because the problem must be solved along the timeline specified. In addition to this extra time, a dynamic analysis also includes two extra matrices needed to solve the differential equation governing the analysis. These are the mass matrix, **M**, and the damping matrix, **C**. The mass matrix is similar to the stiffness matrix in that it lumps the mass of the elements at their nodes. Similarly, the damping matrix lumps damping to the nodes of the model. The equation governing the analysis is shown below:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R}$$

As shown above, dynamic analysis adds greatly to the complexity of the problem. In addition to the several added matrices, the problem must be solved at different time increments to analyze for the worst case forces and stresses. Therefore, this problem is by definition nonlinear.

§2.2.3. *Linear Elastic*

A linear elastic analysis makes two basic assumptions: the structure responds completely elastically and the structure responds in a linear way. When the structure is assumed to respond linear elastically, members are “allowed” to take an infinite amount of forces and displacements; no provision is made for members failing due to buckling or yielding at a level lower than the analysis provides. However, this potential problem is not typically an issue for a properly designed building because the members are designed for the proper forces, deflections are limited, and the members are kept within their elastic limit. However, several cases of nonlinearity are encountered commonly in structures and must be properly accounted for.

§2.2.4. Geometrically Nonlinear

Geometric nonlinearities occur in all structures; however, the effect of geometric nonlinearities in some problems may be minimal. They are so called secondary effects caused by a force acting through a deformed member or members. This type of analysis is mutually exclusive with a linear elastic analysis. However, it can be applied with either a static or dynamic analysis. The two main types of geometric nonlinearities for beam-column elements are outline in the following sections.

§2.2.4.1. $P-\Delta$ Effect

The $P-\Delta$ effect occurs when the end of a member deflects laterally and the axial load on the member moves with it producing an eccentricity, Δ . This adds a moment, $P-\Delta$, to the end of the member that has deflected laterally. This $P-\Delta$ moment tapers off along the length of this member. This is illustrated in Figure 2-9.

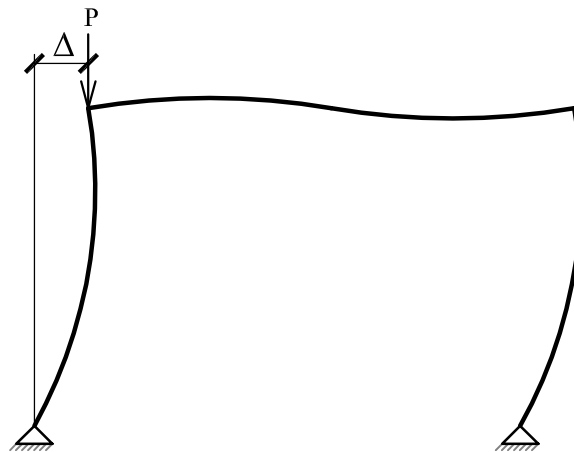


Figure 2-9 $P-\Delta$ Effect

This effect is greatest when large lateral deflections are allowed. This is one of the reasons why the code limits the lateral drift allowed for buildings. Moment frames suffer from this effect more than braced frames because moment frames are much more flexible in the lateral direction.

§2.2.4.2. $P-\delta$ Effect

The $P-\delta$ effect is similar to the $P-\Delta$ effect but the secondary moment distribution is different. With the $P-\Delta$ effect, the largest moment is applied at joints whereas with the $P-\delta$

effect, the secondary moment is largest around midspan and approaches zero near the ends. This is illustrated in Figure 2-10. The $P-\delta$ effect has the largest effect when a slender member is subjected to both bending and compression. These reasons are why it is so important to consider $P-\delta$ effects for tilt-up wall design.

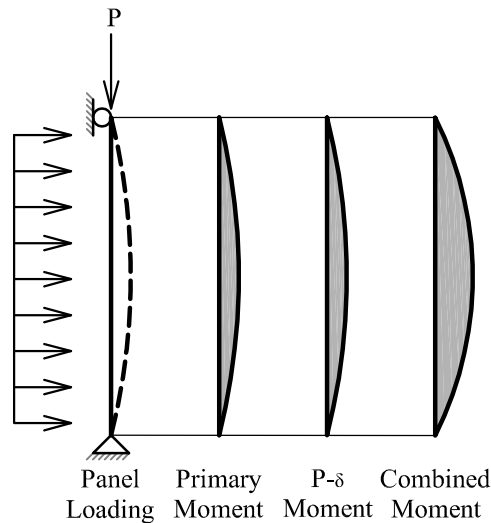


Figure 2-10 $P-\delta$ Effect

In structural analysis programs, the $P-\delta$ effect is often not taken into account. However, usually the $P-\Delta$ effect is taken into account. This is most likely because of the relative simplicity of calculating the $P-\Delta$ effect over the $P-\delta$ effect. For a full derivation of the equations needed to perform either type of analysis, see (Chen & Lui, 1987).

§2.2.5. Materially Nonlinear

Another type of nonlinearity often encountered in real world problems is material nonlinearities. These occur when the stress strain curve of the material is not linear; which is true for almost all materials to a certain extent. However, certain materials, such as most metals, have a nearly linear range - especially for service level loads. However, when predicting the ultimate capacity of members, nonlinear material properties can be especially important. Therefore, material nonlinear properties of both steel and concrete are described.

§2.2.5.1. Steel

Steel exhibits a very linear stress strain curve until it “yields”. After yielding, the stress strain curve can still be approximated linearly, but it has a much lower slope. For the same amount of stress after yielding, the steel will deform much more than before yielding.

Usually, this material nonlinearity does not present much of an issue for steel because we do not want it to yield as this would cause excessive deformation. However, when predicting the theoretical ultimate capacity of a member, reduced stiffness needs to be addressed. This is easily done in a computer analysis by checking to see if the stress is greater than the yield stress and if it is, then it uses a reduced modulus of elasticity rather than the normal modulus of elasticity.

§2.2.5.2. Concrete

Concrete exhibits many nonlinear behaviors. It cracks, has a nonlinear stress strain curve, and it is subject to increased deflections when load duration is increased. These nonlinearities are more important than those of steel. The stress strain curve of concrete is shown in Figure 2-11. It is relatively linear for the first part of the curve, but then becomes highly nonlinear as the strain is increased. This is important because unlike in steel, concrete cannot be approximated linearly for the load that it will see in a typical building structure. Because of this, nonlinear material models such as the Mander model must be used.

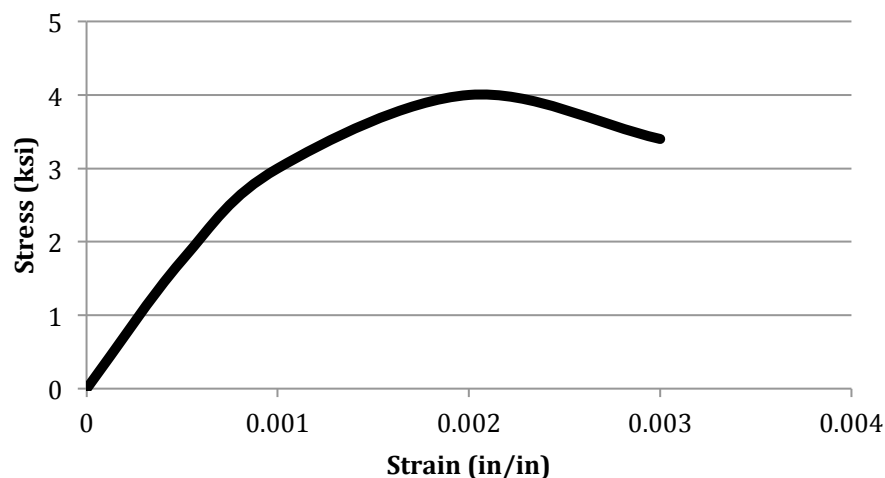


Figure 2-11 Concrete Stress Strain Plot

Another nonlinearity concrete exhibits is cracking. Cracking is an important nonlinear effect because it greatly reduces the bending stiffness of the member thereby causing the

deflection to increase. For a beam, this extra deflection is typically not an issue in design. However, for a beam-column, this additional deflection increases the $P-\delta$ effect. This especially becomes an issue with slender elements - such as tilt-up wall panels. Therefore, it is important to have accurate equations to use when determining the effective moment of inertia of the section.

In a reinforced concrete section, only the part of the concrete in compression is assumed to be part of the effective moment of inertia. The part in tension is assumed to have cracked and therefore not part of the effective stiffness. Of course, the reinforcing bars in tension must also be incorporated into these calculations.

While this method provides accurate results for most problems, an assumption was made that might not be correct. In the previous method, the whole section was assumed to be cracked all along the beam. This is not entirely true. Cracks in concrete are by no means consistent. There are often spaces of uncracked concrete in between the cracks. This tension stiffening effect results in the member actually being stiffer than calculated in the previous method. However, tension stiffening has its greatest effect under loads close to the cracking moment of the section. At loads close to the ultimate capacity of the member, tension stiffening is almost nonexistent (Jenkins, 2010). Therefore, since the ultimate capacity of the member is the main concern for the designer, tension stiffening can be safely ignored.

§2.3. Elements Used and their Properties

An almost unlimited amount of different finite elements exist. All of these different elements have varying properties and characteristics that make them suitable for some problems but not for others. For example, plates derived specifically for bending performance provide much better results than using brick elements connected together to form the same geometry. Additionally, some elements can behave overly stiff or exhibit a phenomenon known as shear locking. Because of this, it is critical to understand the basics of each type of element so they can be properly applied in real world problems. In the following sections, several of the most common types of elements and the elements used for analysis in this report will be discussed..

§2.3.1. Bar Element

The bar element (also known as the truss element or axial force element) is one of the most simple elements in a structural analysis problem. It consists of a straight element containing 2 DOFs (one on each end). This configuration limits this element to only axial

loading. This configuration is shown in Figure 2-4. It can be used to model several different types of construction; anywhere from concrete rebar to bracing elements. Additionally, it is extremely simple to implement into a finite element computer program because of its closed form solution of the global element stiffness matrix as shown below. Further details of this derivation have previously been covered in §2.1.3.

$$[k] = \frac{AE}{L} \begin{bmatrix} \cos^2 \phi & \sin \phi \cos \phi & -\cos^2 \phi & \sin \phi \cos \phi \\ \sin \phi \cos \phi & \sin^2 \phi & \sin \phi \cos \phi & -\sin^2 \phi \\ -\cos^2 \phi & \sin \phi \cos \phi & \cos^2 \phi & \sin \phi \cos \phi \\ \sin \phi \cos \phi & -\sin^2 \phi & \sin \phi \cos \phi & \sin^2 \phi \end{bmatrix}$$

§2.3.2. Bernoulli-Euler Plane Beam

The beam element is the next step up in element complexity. It consists of 2 DOFs per node (v and θ) as shown in Figure 2-12. It is useful for modeling bending in structures that can be idealized as lines. For example, beams and columns in structures can be modeled with this element (although, for columns, an axial degree of freedom must be introduced).

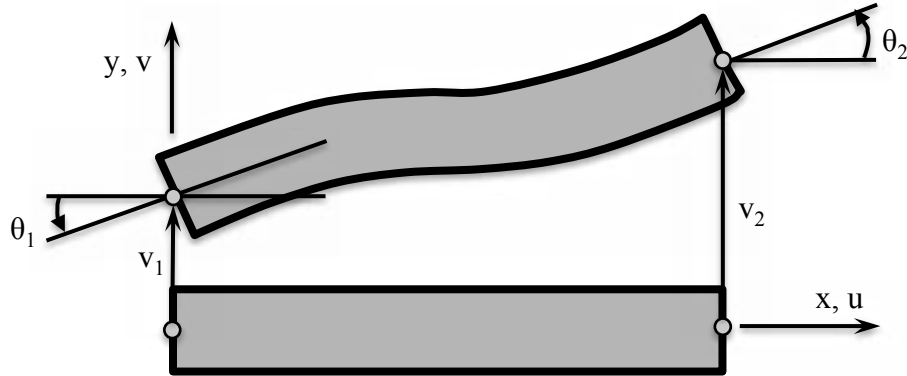


Figure 2-12 Euler Bernoulli Plane Beam Element

Because this element has four DOFs (as shown in Figure 2-12), its stiffness matrix will be 4x4 in size. In its most general form, this is shown as:

$$\begin{Bmatrix} V_1 \\ M_1 \\ V_2 \\ M_2 \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{Bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{Bmatrix}$$

To derive the coefficients of this matrix the bending formulas derived in Mechanics of Materials will be used. The first column of coefficients, $\begin{Bmatrix} k_{11} \\ k_{21} \\ k_{31} \\ k_{41} \end{Bmatrix}$, represents the reactions when

the first DOF, v_1 , has a unit displacement while all other DOFs have a displacement of zero.

The first column of the beam element is: $\begin{Bmatrix} \frac{12EI}{L^3} \\ \frac{6EI}{L^2} \\ -\frac{12EI}{L^3} \\ \frac{6EI}{L^2} \end{Bmatrix}$. The second, third, and fourth columns of the

beam stiffness matrix can be derived similarly by setting the DOF corresponding to that column to one while setting the others to zero.

Combining these in matrix form yields the following:

$$[k] = \begin{matrix} & \begin{matrix} v_1 & \theta_1 & v_2 & \theta_2 \end{matrix} \\ \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & 4EI & \frac{6EI}{L^2} & 2EI \\ -\frac{12EI}{L^3} & \frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & 2EI & -\frac{6EI}{L^2} & 4EI \end{bmatrix} \end{matrix}$$

This is the local stiffness matrix for the element. §2.1.3, mentioned that in order to perform an analysis on more than one member, the member had to be transformed into global coordinates before they could be assembled. This is not very practical for this element because it is assumed to be axially inextensible. As soon as this element is rotated and assembled with other elements that are not in the same coordinate system, the members are not allowed to deform as they should. To accommodate this issue, an axial DOF must be added. This axial DOF is borrowed from the truss or bar element and inserted into the proper place in the element stiffness matrix. The resulting element is often called a frame element.

Its local stiffness matrix is:

$$[k] = \begin{matrix} & \begin{matrix} u_1 & v_1 & \theta_1 & u_2 & v_2 & \theta_2 \end{matrix} \\ \begin{matrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{matrix} \end{matrix}$$

To analyze an entire structure, the local stiffness matrix must be transformed into the global coordinate system. This is done similarly to the truss element in §2.1.3 with:

$$[k] = [T]^T [k'] [T]$$

$$\text{where } [T] = \begin{bmatrix} \cos \Phi & \sin \Phi & 0 & 0 & 0 & 0 \\ -\sin \Phi & \cos \Phi & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \Phi & \sin \Phi & 0 \\ 0 & 0 & 0 & -\sin \Phi & \cos \Phi & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Calculating this symbolically yields:

$$[k] = \begin{matrix} & \begin{matrix} u_1 & v_1 & \theta_1 & u_2 & v_2 & \theta_2 \end{matrix} \\ \begin{matrix} \frac{AE}{L}C^2 + \frac{12EI}{L^3}S^2 & \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)SC & -\frac{6EI}{L^2}S & -\frac{6EI}{L^2}S & -\frac{6EI}{L^2}S & -\frac{6EI}{L^2}S \\ \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)SC & \frac{AE}{L}S^2 + \frac{12EI}{L^3}C^2 & \frac{6EI}{L^2}C & -\left(\frac{AE}{L} - \frac{12EI}{L^3}\right)SC & -\left(\frac{AE}{L}S^2 + \frac{12EI}{L^3}C^2\right) & \frac{6EI}{L^2}C \\ -\frac{6EI}{L^2}S & \frac{6EI}{L^2}C & \frac{4EI}{L} & \frac{6EI}{L^2}S & -\frac{6EI}{L^2}C & \frac{2EI}{L} \\ -\left(\frac{AE}{L}C^2 + \frac{12EI}{L^3}S^2\right) & -\left(\frac{AE}{L} - \frac{12EI}{L^3}\right)SC & \frac{6EI}{L^2}S & \frac{AE}{L}C^2 + \frac{12EI}{L^3}S^2 & \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)SC & \frac{6EI}{L^2}S \\ -\left(\frac{AE}{L} - \frac{12EI}{L^3}\right)SC & -\left(\frac{AE}{L}S^2 + \frac{12EI}{L^3}C^2\right) & -\frac{6EI}{L^2}C & \left(\frac{AE}{L} - \frac{12EI}{L^3}\right)SC & \frac{AE}{L}S^2 + \frac{12EI}{L^3}C^2 & -\frac{6EI}{L^2}C \\ -\frac{6EI}{L^2}S & \frac{6EI}{L^2}C & \frac{2EI}{L} & \frac{6EI}{L^2}S & -\frac{6EI}{L^2}C & \frac{4EI}{L} \end{matrix} \end{matrix}$$

where $C = \cos \Phi$ and $S = \sin \Phi$.

To obtain the forces at the element nodes in local coordinates by using the element displacements in global coordinates, the proper stiffness matrix must be used, as shown in the following:

$$\{P'\} = [K']\{U'\} = [K'][T]\{U\} = [\bar{K}]\{U\}$$

$$[\bar{K}] = \begin{bmatrix} \frac{AE}{L}C & \frac{AE}{L}S & 0 & -\frac{AE}{L}C & -\frac{AE}{L}S & 0 \\ -\frac{12EI}{L^3}S & \frac{12EI}{L^3}C & \frac{6EI}{L^2} & \frac{12EI}{L^3}S & -\frac{12EI}{L^3}C & \frac{6EI}{L^2} \\ -\frac{6EI}{L^2}S & \frac{6EI}{L^2}C & \frac{4EI}{L} & \frac{6EI}{L^2}S & -\frac{6EI}{L^2}C & \frac{2EI}{L} \\ -\frac{AE}{L}C & -\frac{AE}{L}S & 0 & \frac{AE}{L}C & \frac{AE}{L}S & 0 \\ \frac{12EI}{L^3}S & -\frac{12EI}{L^3}C & -\frac{6EI}{L^2} & -\frac{12EI}{L^3}S & \frac{12EI}{L^3}C & -\frac{6EI}{L^2} \\ -\frac{6EI}{L^2}S & \frac{6EI}{L^2}C & \frac{2EI}{L} & \frac{6EI}{L^2}S & -\frac{6EI}{L^2}C & \frac{4EI}{L} \end{bmatrix}$$

where $C = \cos \Phi$ and $S = \sin \Phi$.

§2.3.3. Two & Three Dimensional Elements

Up until this point, only one-dimensional elements have been discussed. The derivation of two-dimensional elements becomes much more complicated. The bending formulas of mechanics of materials are no longer applicable and we must resort to a much more general approach. One of these more general approaches is called displacement-based finite element method. The displacement-based method is similar to the derivation of beam and axial force elements described in the previous sections and is a good start for two and three dimensional elements. However, displacement based elements can exhibit some bad characteristics; shear locking being one of them. Therefore, other finite elements such as isoparametric elements will be described.

§2.3.3.1. Displacement Based Elements

The relationship between displacements within each element measured in a local coordinate system to the displacements at the element nodes must be defined. This is accomplished through the use of the following:

$$\mathbf{u}(x, y, z) = \mathbf{H}(x, y, z) \hat{\mathbf{U}}$$

“where \mathbf{H} is the displacement interpolation matrix and $\hat{\mathbf{U}}$ is a vector of the three global displacement components at all nodal points” (Bathe, 2006). The terms in \mathbf{H} are determined based on what order of displacement being used. For instance, the displacement interpolation functions can be linear which means that the element is assumed to have straight lines between the nodal points. Or, higher order interpolation functions can be defined and used to give more accurate results.

After defining the displacement interpolation functions, it is also possible to determine the corresponding strains:

$$\boldsymbol{\epsilon}(x, y, z) = \mathbf{B}(x, y, z) \hat{\mathbf{U}}$$

“where \mathbf{B} is the strain-displacement matrix obtained by appropriately differentiating and combining rows of the matrix, \mathbf{H} ” (Bathe, 2006).

To determine the stresses in a finite element, the following equation can be used:

$$\boldsymbol{\tau} = \mathbf{C} \boldsymbol{\epsilon} + \boldsymbol{\tau}'$$

“where \mathbf{C} is the elasticity matrix and $\boldsymbol{\tau}'$ are the given element initial stresses” (Bathe, 2006). For an isotropic linear elastic material, the stress-strain matrix, \mathbf{C} , for plane stress defined using Young’s modulus, E , and Poisson’s ratio, ν , is (Bathe, 2006):

$$\mathbf{C} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix}$$

In three dimensions, this looks like:

$$\mathbf{C} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$$

And for plate bending, this becomes:

$$\mathbf{C} = \frac{Eh^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

Using these relationships, the element stiffness matrix can be derived using the theory of virtual displacements to yield:

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV$$

The above information is only a basic overview of some of the concepts needed to comprehend the displacement based finite elements. The next section will provide an overview of some better performing elements.

§2.3.3.2. Isoparametric Elements

Isoparametric elements differ from displacement based elements in how the coordinates of the element are defined. In isoparametric elements, natural coordinates are used to determine the location of a point on the element. The bounds of the element range from -1 to 1 as shown in Figure 2-1. A third dimension can also be added in a similar manner for elements needing this extra DOF.

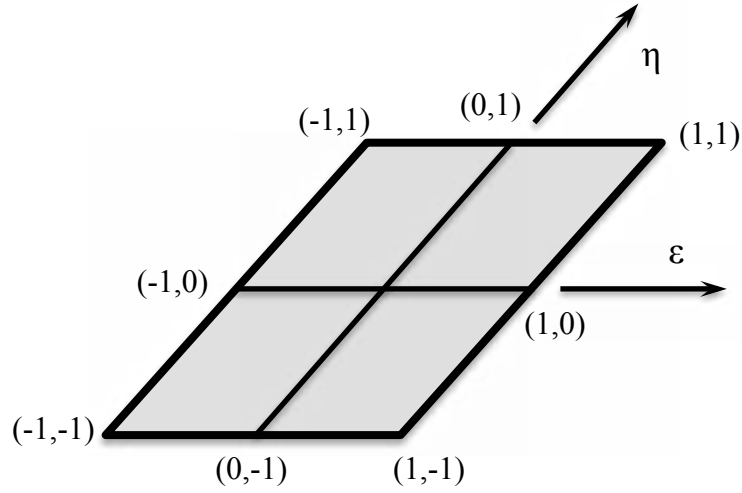


Figure 2-13 Natural Coordinates

This representation can be very beneficial in several different ways. First, using this method eliminates the need for other generalized coordinate systems or transformation matrices. Instead, we will make use of interpolation functions (also called shape functions). Secondly, these set of bounds will aid with the integration process. Finally, the isoparametric representation makes the evaluation by the computer very simple and efficient.

First, the coordinate interpolations for isoparametric elements must be defined. This is accomplished by the following:

$$x = \sum_{i=1}^q h_i x_i$$

$$y = \sum_{i=1}^q h_i y_i$$

$$z = \sum_{i=1}^q h_i z_i$$

“where x, y , and z are the coordinates at any point of the element (here local coordinates) and $x_i, y_i, z_i, i = 1, \dots, q$, are the coordinates of the q element nodes. The interpolation functions h_i are defined in the natural coordinate system of the element, which has variables r, s , and t ” (Bathe, 2006). Interpolation functions for a 4 node two-dimensional element might look like:

$$h_1 = \frac{1}{4}(1+r)(1+s)$$

$$h_2 = \frac{1}{4}(1-r)(1+s)$$

$$h_3 = \frac{1}{4}(1-r)(1-s)$$

$$h_4 = \frac{1}{4}(1+r)(1-s)$$

In order to calculate the element stiffness matrix, the x, y, z derivatives must be related to the r, s, t derivatives. To do this, the Jacobian operator is used. This is often denoted as \mathbf{J} .

$$\frac{\partial}{\partial \mathbf{r}} = \mathbf{J} \frac{\partial}{\partial \mathbf{x}}$$

$$\begin{Bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix}$$

This stiffness matrix can then be evaluated as:

$$\mathbf{K} = \int_V \mathbf{F} dr ds dt$$

where $\mathbf{F} = \mathbf{B}^T \mathbf{C} \mathbf{B} \det \mathbf{J}$. This form leads itself to numerical integration in the form of Gaussian quadrature. Gaussian quadrature (or integration) involves evaluating the function at a certain set of points and then applying specific weights to them to obtain accurate integration results. These formulas are much more efficient and computer programming friendly than explicit integration. So to carry out the integration of the stiffness matrix above, the following formula should be used with the appropriate sampling points and weights.

$$\mathbf{K} = \sum_{i,j,k} \alpha_{ijk} \mathbf{F}_{ijk}$$

where \mathbf{F}_{ijk} is the matrix \mathbf{F} evaluated at the point (r_i, s_j, t_k) and α_{ijk} is a constant that depends on the values of r_i , s_j , and t_k .

§2.3.4. Plane Stress Quad

One of the most basic isoparametric elements is the plane stress quadrilateral shown in Figure 2-14. It has two degrees of freedom at each node (one for the x direction and one for the y-direction). This element is useful for analyzing problems involving in plane forces and stresses. A real world example of this might be a steel plate in tension or shear.

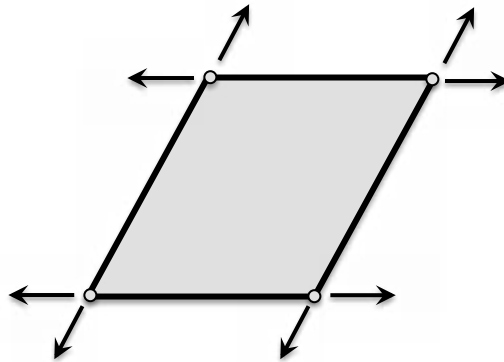


Figure 2-14 Plane Stress Quadrilateral

§2.3.5. Plate

Another isoparametric finite element is the quadrilateral plate element. This element consists of one translational DOF and two drilling (rotational) DOFs per node as illustrated in Figure 2-15. This element is useful for modeling two-dimensional bending elements such as concrete floor slabs or a non-structural façade. This element is not very practical for an analysis requiring axial degrees of freedom. For example, this element would not perform well for a load bearing wall with out of plane forces. In this case, a shell element would need to be used.

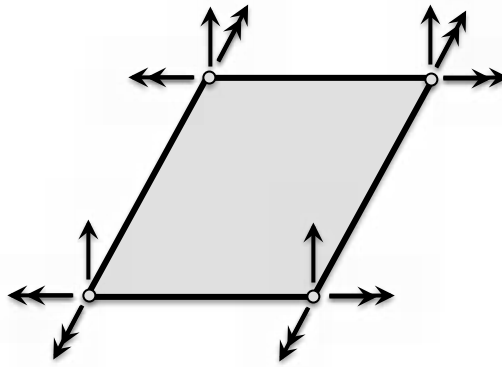


Figure 2-15 Quadrilateral Plate

§2.3.6. *Shell Element*

Shell elements at their root are a combination of the plane stress element and the plate bending element. They include three translational DOFs and two drilling (rotational) DOFs per node as illustrated in Figure 2-16. They are suitable for modeling elements requiring axial, shear, and bending DOFs. This would include load bearing walls such as tilt-up concrete walls. This is the type of element used for analysis in the next chapter.

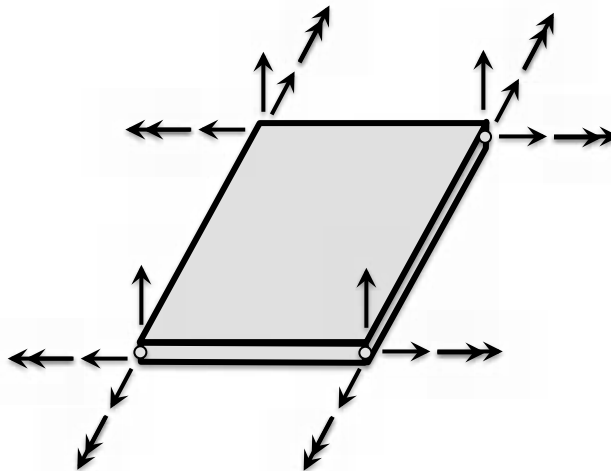


Figure 2-16 Shell Element

§2.3.7. *Hexahedron (Brick) Element*

The brick element is an eight-node element containing three translational DOFs per node resulting in 24 DOFs total as shown in Figure 2-17. This element is useful for modeling general three-dimensional problems where the shell element might not be appropriate. This might

include areas where the problem does not have much bending such as a beam bearing on concrete. This element would be useful in determining the stress in the surrounding area.

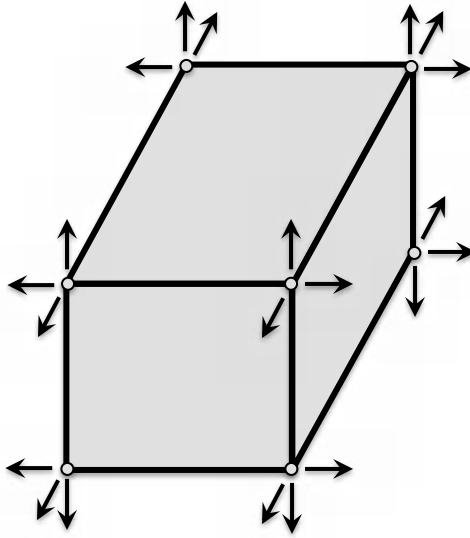


Figure 2-17 Hexahedron (Brick) Element

§3.0. Results

Four design philosophies or assumptions from Section 14.8 of the ACI 318-08 have been analyzed. These include two-way bending assumptions, constant bending stiffness along the member, a bending stiffness reduction factor, and the effect of axial load on the stiffness of the member.

§3.1. One-Way Bending Assumption Test

The first assumption made in Section 14.8 of the ACI 318-08 is that panels with openings behave with one-way bending. Several of the same setups used by Bartels (Bartels, 2010) were analyzed and are illustrated in Figure 3-1. The loading on the panels consists of out-of-plane wind pressure on the entire surface and eccentric vertical roof loads at the top of the panels. Specific loading values are included in Appendix A - Panel Loading . In this assumption, P- δ effects were not included because these effects will be tested separately in §3.2.

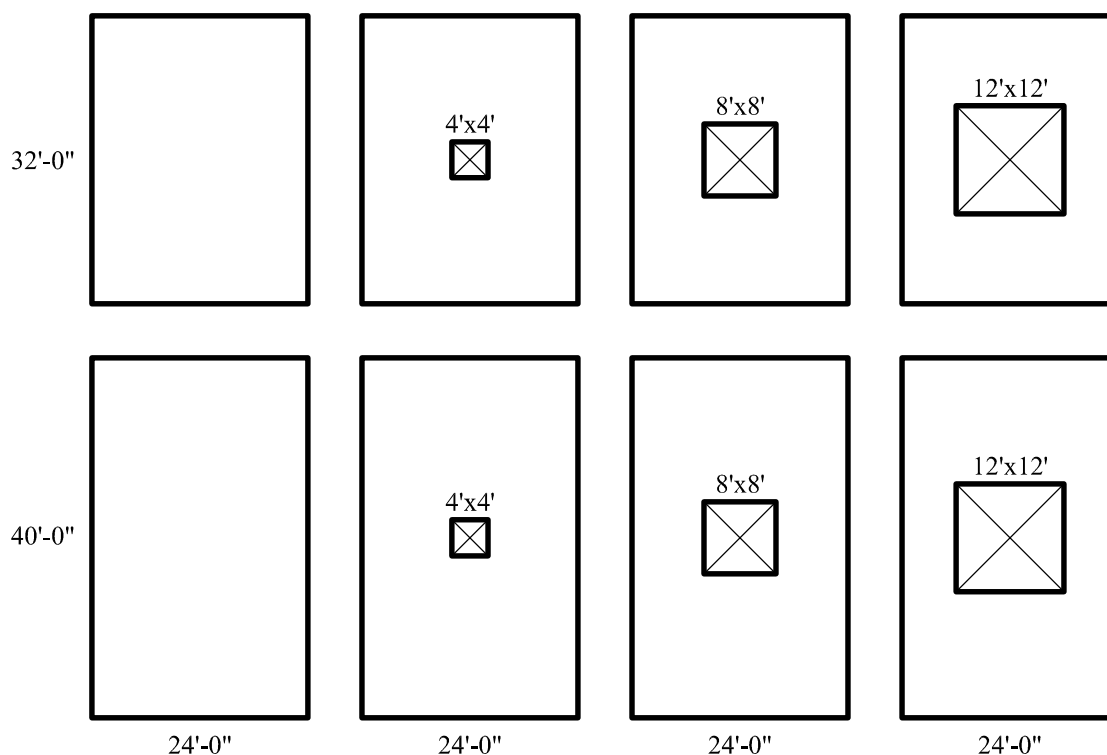


Figure 3-1 Panel Geometry

To test the one-way bending assumption, a computer model using shell elements was used. This allowed for the ability to handle axial loads, out of plane bending, and two-way behavior. Modeling was performed in SAP2000, a finite element program, using the size and spacing of reinforcement calculated by Bartels (Bartels, 2010) for the design strip of each panel case. The design strip is the full height portion of the wall on either side of the opening limited to 12 times the panel thickness. Temperature and shrinkage reinforcing was placed above and below the panel openings. Additionally, a “window” was modeled in the panel openings to apply the proper wind loads acting on the panel. The wind loads are transferred to the panel opening edges on all four sides. The edges of the “window” were pinned to the concrete panel to prevent moment transfer from the window to the panel. The validity of this assumption is beyond the scope of this paper.

The results of this analysis are within 3% of the results shown by Bartels (Bartels, 2010). Maximum stresses were found to occur near mid-height as shown in the control data. In addition, the two-way effects were very minimal in these panels. For instance, consider the 40’x20’ panel with a 12’x12’ opening shown in Figure 3-2; very little of the bending occurs outside of the leg strip of the panel. The bending that did occur outside of the leg strip was at a very low level - meaning it does not control the design of the vertical reinforcing. Additionally, this lack of two-way bending means that the panel design and analysis can be approximated by a frame element. This can prove advantageous for saving processing time and in the complexity of the analysis. The other panel configurations shown in Figure 3-1 provided similar results and are therefore not included here.

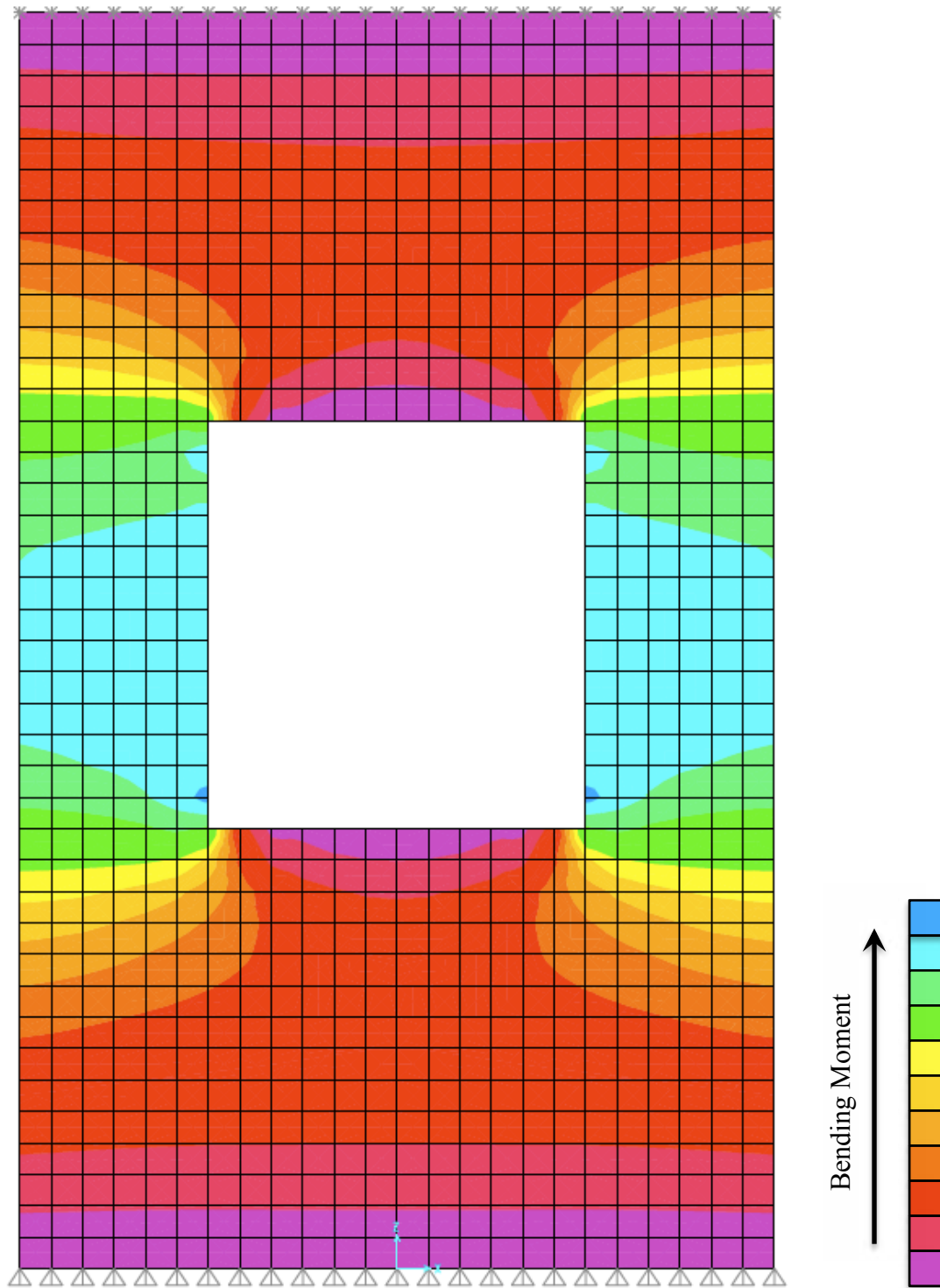


Figure 3-2 Stress Distribution in a 40' x 24' Panel with a 12' x 12' Opening

In conclusion, the assumption made by Bartels (Bartels, 2010) and Section 14.8 of ACI 318-08 (ACI Committee 318, 2008) about the panels behaving with one-way bending is a valid assumption. The panel can be idealized as a one-dimensional design strip.

§3.2. Constant Bending Stiffness Along Member Assumption Test

The second assumption made in Section 14.8 of the ACI 318-08 is that the bending stiffness is constant along the length of the panel (only the panel design strip is used for bending stiffness). However, the panel is not of constant stiffness. The concrete above and below the opening acts to stiffen the panel at those locations. To quantify the effect of this assumption, frame elements with varying stiffness were used to analyze a 40'-0" panel. For the middle section (adjacent to the opening) the stiffness is $I = I_{cr} = 711 \text{ in}^4$, reference (Bartels, 2010). For the other sections, the stiffness was varied by factors of 1.5, 2, 4, 6, 10, and 20 times the I_{cr} value. These factors are intentionally general to highlight the fact that these results can be applied to many different panel configurations. The results are shown in Table 3-1 and also illustrated in Figure 3-3.

Table 3-1 Constant Bending Stiffness Along Member Test Results

I_{cr} Along Entire Length of Panel		
Deflection	3.848	in
Max moment	480.69	kip-in

Portions Above & Below Opening 1.5x Stiffer		
Deflection	3.57	in
Max Moment	472.81	kip-in
Difference	1.64%	

Portions Above & Below Opening 2x Stiffer		
Deflection	3.379	in
Max Moment	467.29	kip-in
Difference	2.79%	

Portions Above & Below Opening 4x Stiffer		
Deflection	3.24	in
Max Moment	463.42	kip-in
Difference	3.59%	

Portions Above & Below Opening 6x Stiffer		
Deflection	3.18	in
Max Moment	461.61	kip-in
Difference	3.97%	

Portions Above & Below Opening 10x Stiffer		
Deflection	3.13	in
Max Moment	460.17	kip-in
Difference	4.27%	

Portions Above & Below Opening 20x Stiffer		
Deflection	3.09	in
Max Moment	459.09	kip-in
Difference	4.49%	

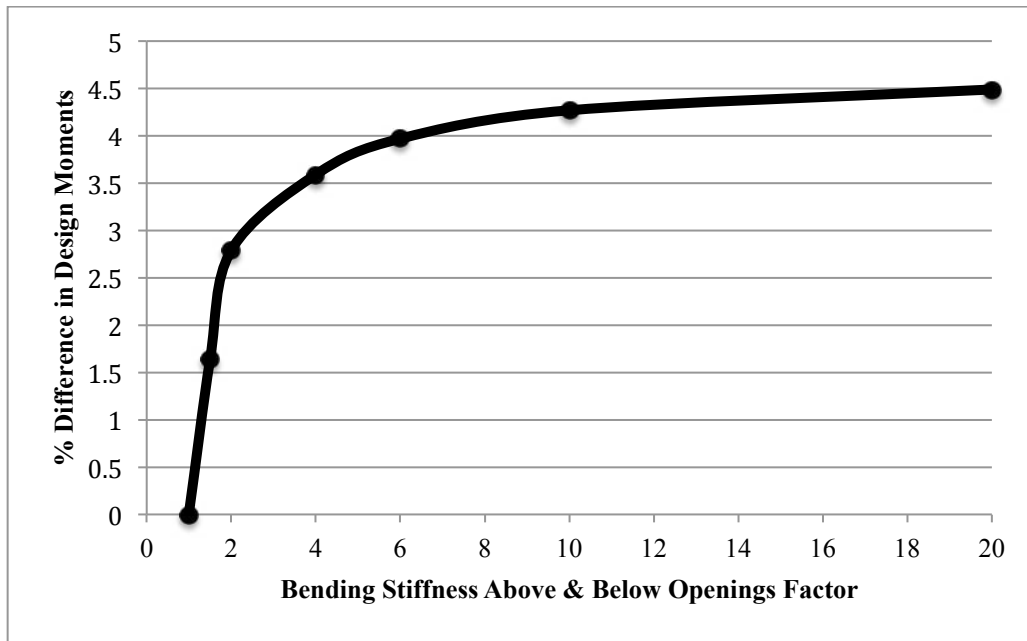


Figure 3-3 Constant Bending Stiffness Along Member Test Results

These results show that even when the stiffness of the panel above and below the opening is increased by an unrealistic amount (by a multiple of 20), the difference between the moments is less than 4.5%. By assuming a one-dimensional design strip and using I_{cr} as I_{eff} for analysis, the provisions of Section 14.8 of the ACI 318-08 are at most approximately 5% conservative. Further research might be to analyze the design strip with the column provisions of the ACI 318-08 for the case of very narrow design strips. For instance, when the stiffness of the panel above the openings is at the level of 10-20 times above the design strip stiffness, the column provisions of the ACI 318-08 might be more applicable.

§3.3. Bending Stiffness Reduction Factor Assumption Test

The third design philosophy or assumption analyzed surrounds a 0.75 factor that the code places on the bending stiffness of the member to account for variations in workmanship and reinforcing placement. This is analyzed by performing a moment curvature analysis on the section. Both the placement of the reinforcing and the thickness of the panel will be varied to the code allowed tolerances of the (Standard Specification for Tolerances for Concrete Construction & Materials (ACI 117-90)) (ACI Committee 117, 2002). The tolerances for reinforcing placement are $\pm 3/8''$ (Section 2.2.2 & 2.2.3) (ACI Committee 117, 2002). The tolerances for wall thicknesses are $+3/8''$ and $-1/4''$ (Section 4.4.1) (ACI Committee 117, 2002).

The specific panel being analyzed is a 40'x24' panel with a 12'x12' opening. The panel is 7.25" thick and has (2) layers of reinforcing ($A_s = 4.4 \text{ in}^2$ & $A'_s = 4.4 \text{ in}^2$) located 1.75" from the center of reinforcing to the edge of the panel. Without any variation in the reinforcing placement or panel thickness, the cracked moment of inertia, I_{cr} , is found to be 638 in^4 . Varying the placement of the reinforcing by 0.05" increments to the tolerance of 3/8" yields Figure 3-4. The largest reduction due to reinforcing placement variation to the allowable tolerance is approximately 0.85.

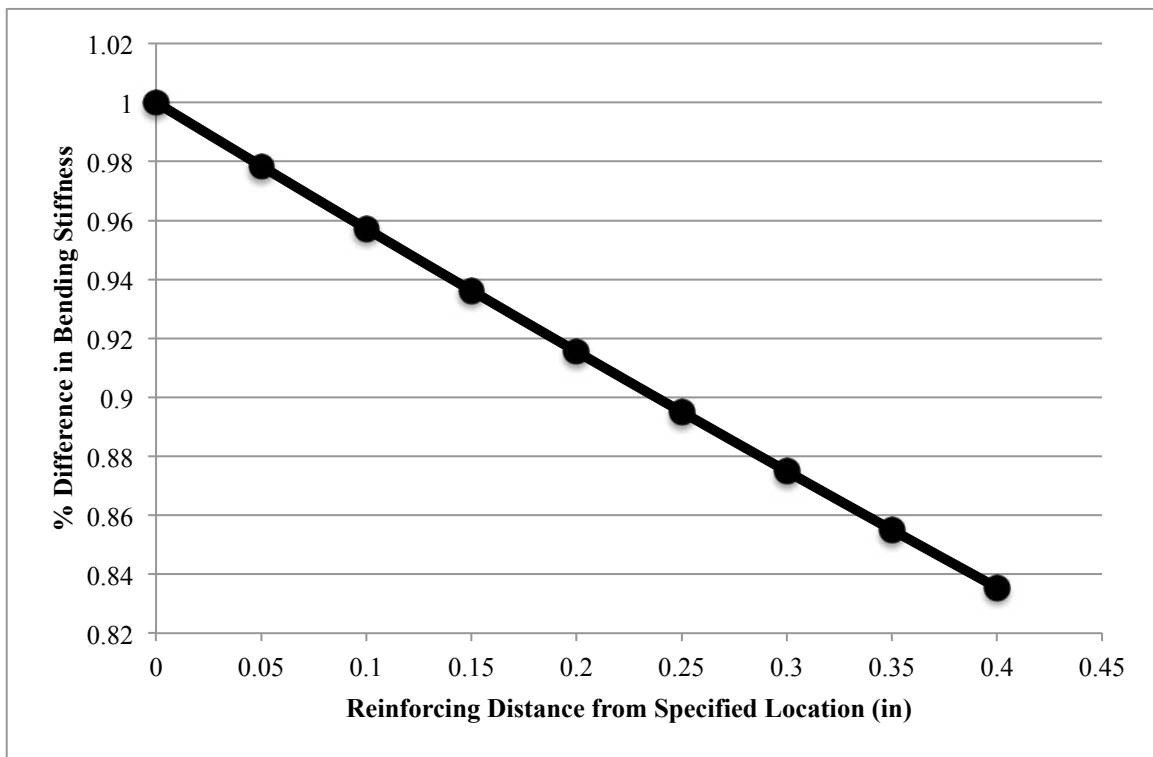


Figure 3-4 Bending Stiffness Workmanship Reduction Results

Varying the panel thickness to the allowable tolerance of 1/4" along with the placement of the reinforcing yields a cracked moment of inertia, I_{cr} , of 479 in^4 . This value is 75% reduction in bending stiffness compared to the specified panel dimensions and reinforcing placements. This is identical to the bending stiffness reduction factor that the code specifies. Further research might include performing this analysis on several different panel configurations to determine the applicability of this code specified bending stiffness reduction factor to a wide variety of designs.

§3.4. Effect of Axial Force on Bending Stiffness Test

The final design philosophy or assumption of Section 14.8 of the ACI 318-08 is how it accounts for the effects of axial force on the stiffness of the section. Section 14.8 specifies an equivalent area of reinforcing:

$$A_{se} = A_s + \frac{P_u}{f_y} \left(\frac{h}{2d} \right)$$

This equation was a modification from previous versions of the ACI 318 because the old version:

$$A_{se} = A_s + \frac{P_u}{f_y}$$

“overestimated the contribution of axial load in many cases where two layers of reinforcement were used in the slender wall” (ACI Committee 318, 2008). To determine the validity of this new equivalent area of reinforcing term, a moment curvature analysis was performed for varying axial loads up to the maximum specified by ACI 318-08 Section 14.8.2.6,

$$\frac{P_u}{A_g} \leq 0.06f'_c$$

In this case, the maximum axial force allowed by the code is $P_u = 125$ kips. The moment curvature analysis values are then compared with values obtained by using the design procedure outlined in Section 14.8 of the ACI 318-08. From Figure 3-5, it can be seen that the moment curvature analysis produced bending stiffness values that were lower than the equivalent reinforcing steel area approximation employed by Section 14.8 of the ACI 318-08. The maximum difference (21.5%) occurred at the greatest axial load as illustrated in Figure 3-5.

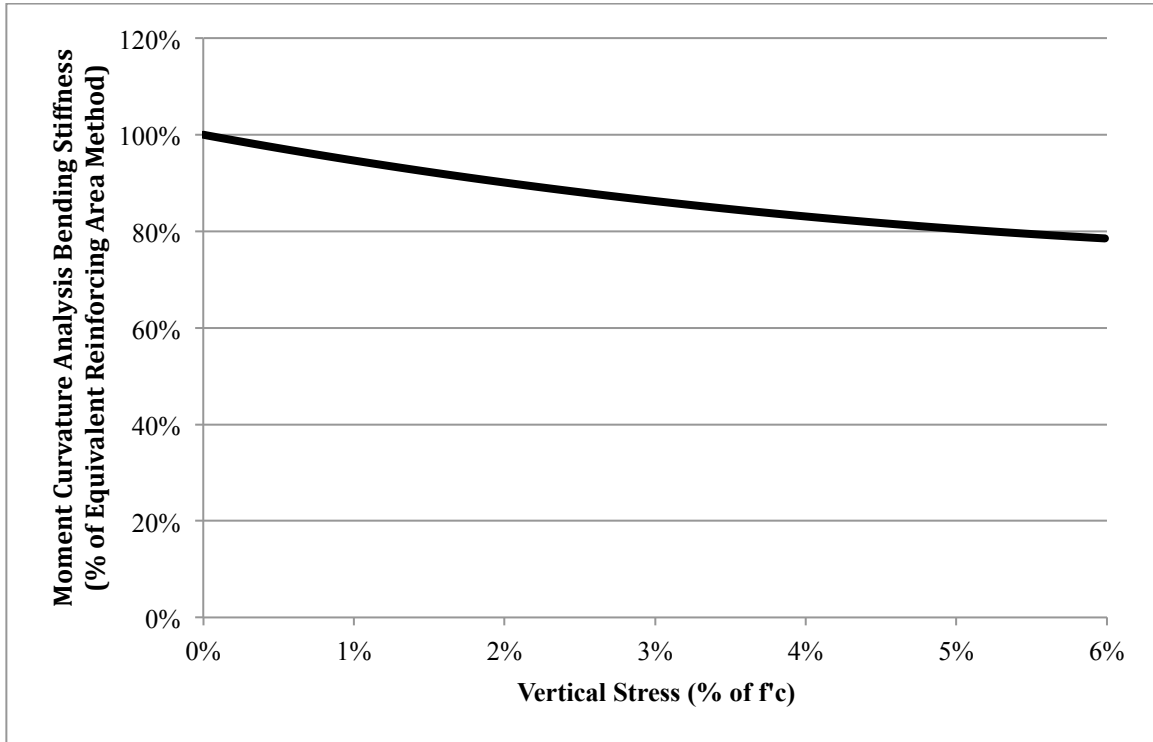


Figure 3-5 Effect of Axial Force on Bending Stiffness

The moment capacity of the section at these axial loads for the two different methods can also be compared. For this, the moment curvature analysis moment capacity is approximately 96% of the equivalent steel area approximation provided in Section 14.8 of the ACI 318-08. These two values are in better agreement than the bending stiffness comparison above. Further research in this area might include determining a better approximation method for the bending stiffness for inclusion in Section 14.8 of the ACI 318-08.

§4.0. Conclusion

This report tested the applicability of four of the design philosophies or assumptions made in Section 14.8 of the ACI 318-08 (ACI Committee 318, 2008). These include stress distribution around openings, the effect of varying stiffness of the member on the $P-\delta$ effect, stiffness variations due to workmanship and tolerances, and the effect of axial load on the stiffness of the member. First, the stress distributions around the panel openings were found to very closely resemble one-way bending. Therefore, beam elements can be used to model the behavior of the full height panel strip. Secondly, the portions of the panel above and below the openings were found to have little effect on the $P-\delta$ effect. Third, a 75% reduction in bending stiffness occurred when maximum allowable tolerances were employed which matches with the ACI 318-08 recommended bending stiffness reduction. Finally, the ACI 318-08 may overestimate the bending stiffness of doubly reinforced sections by up to 20%. This overestimation stems from the assumptions made in the calculation of the effective area of reinforcing defined in Section 14.8 of the ACI 318-08.

Bibliography

- ACI Committee 117. (2002). *Standard Specification for Tolerances for Concrete Construction & Materials (ACI 117-90)*. American Concrete Institute.
- ACI Committee 318. (2008). *Building Code Requirements For Structural Concrete (ACI 318-08) And Commentary*. American Concrete Institute.
- ACI Committee 551. (2010). *Design Guide for Tilt-Up Concrete Panels*. Farmington Hills, MI: American Concrete Institute.
- ASCE 7-05 *Minimum Design Loads for Buildings and Other Structures*. (2006). American Society of Civil Engineers.
- Bartels, B. (2010). *Analysis of Vertical Reinforcement in Slender Reinforced Concrete (Tilt-up) Panels with Openings & Subject to Varying Wind Pressures*. Kansas State University, Department of Architectural Engineering and Construction Science, Manhattan, KS.
- Bathe, K.-J. (2006). *Finite Element Procedures*. Prentice Hall, Pearson Education, Inc.
- Chen, W. F., & Lui, E. M. (1987). *Structural Stability - Theory and Implementation*. Prentice Hall.
- Felippa, C. A. (2004). *Introduction to Finite Element Methods*. Boulder, CO: University of Colorado.
- Hughes, T. J. (2000). *The Finite Element Method - Linear Static and Dynamic Finite Element Analysis*. Englewood Cliffs, N.J.: Prentice Hall, Inc.
- Jenkins, D. (2010, 03 12). *Tension Stiffening*. Retrieved from Newton Excel Bach, not (just) an Excel Blog: <http://newtonexcelbach.wordpress.com/2010/03/12/tension-stiffening/>

Appendix A - Panel Loading (Bartels, 2010)

Panel Height	Panel Thickness	$S_s \leq$	$0.4 \cdot I \cdot S_{DS} \cdot \text{Panel Weight}$	Wind Speed	Wind Pressure
32'-0"	7.25 in	1.410	23.9 psf	90 mph	24 psf
40'-0"	7.25 in	1.475	24.9 psf	90 mph	25 psf
32'-0"	7.25 in	2.125	35.9 psf	110 mph	36 psf
40'-0"	9.25 in	1.710	36.9 psf	110 mph	37 psf
32'-0"	7.25 in	2.960	49.8 psf	130 mph	50 psf
40'-0"	9.25 in	2.405	51.9 psf	130 mph	52 psf
32'-0"	9.25 in	3.055	65.9 psf	150 mph	66 psf
40'-0"	11.25 in	2.625	68.9 psf	150 mph	69 psf

Summary of Design Loads (Bartels, 2010)

32'-0" (90 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
7.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	66.20	0.46	31.70	100.00	12.76	13.86	0.28	199.65	63.09	111.64	1.88	2.56
	1.2D + 1.6W + 0.5(L _r +S)	57.70	0.92	27.65	100.00	12.76	13.72	0.27	197.88	120.26	194.43	1.90	
	0.9D + 1.6W	40.40	0.92	19.36	100.00	12.76	13.43	0.27	194.26	119.07	163.15	1.59	

32'-0" (110 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
7.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	66.20	0.69	31.70	100.00	8.80	9.53	0.13	223.28	92.58	116.91	2.49	2.56
	1.2D + 1.6W + 0.5(L _r +S)	57.70	1.38	27.65	100.00	8.80	9.43	0.12	221.23	179.24	219.37	2.52	
	0.9D + 1.6W	40.40	1.38	19.36	100.00	8.80	9.24	0.12	217.01	178.05	204.67	2.43	

32'-0" (130 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
7.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	66.20	0.96	31.70	100.00	16.40	17.13	0.22	383.46	126.98	147.64	2.55	2.56
	1.2D + 1.6W + 0.5(L _r +S)	57.70	1.92	27.65	100.00	16.40	17.03	0.22	381.59	248.06	282.67	2.56	
	0.9D + 1.6W	40.40	1.92	19.36	100.00	16.40	16.84	0.22	377.77	246.87	270.18	2.50	

32'-0" (150 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
9.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	79.20	1.27	29.71	162.00	10.80	11.61	0.11	373.36	167.11	187.12	1.97	2.56
	1.2D + 1.6W + 0.5(L _r +S)	70.70	2.53	26.54	162.00	10.80	11.53	0.11	370.71	327.15	361.93	1.99	
	0.9D + 1.6W	50.10	2.53	18.82	162.00	10.80	11.32	0.11	364.24	325.73	349.92	1.97	

Solid Panels - 32 ft (Bartels, 2010)

40'-0" (90 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
7.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	76.60	0.48	36.70	100.00	15.40	16.24	0.21	365.62	100.10	135.52	3.11	3.2
	1.2D + 1.6W + 0.5(L _r +S)	68.20	0.96	32.65	100.00	15.40	16.15	0.21	363.73	194.30	253.43	3.12	
	0.9D + 1.6W	48.20	0.96	23.11	100.00	15.40	15.93	0.21	359.28	193.11	231.69	3.00	

40'-0" (110 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
9.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	92.50	0.71	34.71	162.00	10.20	11.15	0.11	359.20	146.98	184.10	2.73	3.2
	1.2D + 1.6W + 0.5(L _r +S)	84.00	1.42	31.54	162.00	10.20	11.06	0.11	356.52	286.91	351.72	2.76	
	0.9D + 1.6W	60.10	1.42	22.57	162.00	10.20	10.82	1.00	348.97	285.48	329.75	2.71	

40'-0" (130 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
9.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	92.50	1.00	34.71	162.00	16.40	17.35	0.17	544.07	204.58	239.44	3.14	3.2
	1.2D + 1.6W + 0.5(L _r +S)	84.00	2.00	31.54	162.00	16.40	17.26	0.17	541.55	402.11	463.64	3.15	
	0.9D + 1.6W	60.10	2.00	22.57	162.00	16.40	17.02	0.16	534.43	400.68	443.19	3.12	

40'-0" (150 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
9.25" Panel	1.2D + 1.6(L _r +S) + 0.8W	92.50	1.33	34.71	162.00	28.60	29.55	0.28	876.94	269.86	301.69	3.17	3.2
	1.2D + 1.6W + 0.5(L _r +S)	84.00	2.65	31.54	162.00	28.60	29.46	0.28	874.71	532.67	589.22	3.17	
	0.9D + 1.6W	60.10	2.65	22.57	162.00	28.60	29.22	0.28	868.41	531.24	570.62	3.14	

Solid Panels - 40 ft (Bartels, 2010)

32'-0" (90 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
4' x 4' (7.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	32.70	0.23	37.54	42.00	3.2/face	3.56	0.11	83.90	31.54	43.10	1.82	2.56
	1.2D + 1.6W + 0.5(L _r +S)	28.40	0.46	32.68	42.00	3.2/face	3.51	0.11	82.85	60.13	78.68	1.84	
	0.9D + 1.6W	19.90	0.46	22.85	42.00	3.2/face	3.42	0.11	80.74	59.54	71.59	1.72	
8' x 8' (7.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	31.40	0.23	45.05	33.00	3.2/face	3.54	0.14	82.53	31.54	43.37	2.45	2.56
	1.2D + 1.6W + 0.5(L _r +S)	27.10	0.46	38.98	33.00	3.2/face	3.50	0.14	81.51	60.13	78.93	2.48	
	0.9D + 1.6W	18.90	0.46	27.16	33.00	3.2/face	3.41	0.13	79.54	59.54	71.66	2.35	
12' x 12' (7.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	29.20	0.23	55.91	25.00	4.4/face	4.72	0.25	104.55	31.54	41.13	2.49	2.56
	1.2D + 1.6W + 0.5(L _r +S)	25.00	0.46	47.80	25.00	4.4/face	4.67	0.25	103.64	60.13	75.21	0.25	
	0.9D + 1.6W	17.30	0.46	33.09	25.00	4.4/face	4.59	0.24	101.98	59.54	69.25	2.39	
16' x 16' (9.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	30.70	0.23	69.14	27.00	2.2/face	2.52	0.14	79.66	31.94	40.16	2.05	2.56
	1.2D + 1.6W + 0.5(L _r +S)	26.50	0.46	59.61	27.00	2.2/face	2.47	0.14	78.38	60.36	73.48	2.08	
	0.9D + 1.6W	18.40	0.46	41.46	27.00	2.2/face	2.39	0.14	75.91	59.64	68.35	2.01	

Panel with Openings - 32 ft (Bartels, 2010)

40'-0" (90 mph)	Load Combination	P _{um} (k)	w _u (klf)	P _u /A _g (psi)	M _{cr} (k-ft)	A _s (in ²)	A _{se} (in ²)	c/d	ΦM _n (k-ft)	M _{ua} (k-ft)	M _u (k-ft)	Δ _s (in)	Δ _s < Δ _{allow}
4' x 4' (9.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	37.90	0.24	43.54	42.00	10.6/face	11.02	0.35	232.50	50.05	65.56	3.09	3.2
	1.2D + 1.6W + 0.5(L _r +S)	33.70	0.48	38.68	42.00	10.6/face	10.97	0.35	231.68	97.15	123.07	3.10	
	0.9D + 1.6W	23.80	0.48	27.35	42.00	10.6/face	10.86	0.34	229.80	96.55	113.55	2.99	
8' x 8' (9.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	44.00	0.24	49.57	54.00	3.8/face	4.25	0.12	136.04	50.45	68.13	2.65	3.2
	1.2D + 1.6W + 0.5(L _r +S)	39.80	0.48	44.81	54.00	3.8/face	4.21	0.12	134.72	97.37	127.51	2.68	
	0.9D + 1.6W	28.40	0.48	31.98	54.00	3.8/face	4.09	0.12	131.18	96.66	116.79	2.61	
12' x 12' (9.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	41.20	0.24	61.93	41.00	4.2/face	4.62	0.18	144.27	50.45	67.16	3.14	3.2
	1.2D + 1.6W + 0.5(L _r +S)	37.00	0.48	55.58	41.00	4.2/face	4.58	0.18	143.02	97.37	125.60	3.16	
	0.9D + 1.6W	26.30	0.48	39.52	41.00	4.2/face	4.47	0.17	139.86	96.66	115.40	3.08	
16' x 16' (9.25" Panel)	1.2D + 1.6(L _r +S) + 0.8W	43.40	0.24	80.29	40.00	2.8/face	3.23	0.15	129.37	50.85	65.20	2.65	3.2
	1.2D + 1.6W + 0.5(L _r +S)	39.10	0.48	72.46	40.00	2.8/face	3.19	0.15	127.81	97.60	122.07	2.68	
	0.9D + 1.6W	27.90	0.48	51.68	40.00	2.8/face	3.08	0.14	123.65	96.77	113.40	2.64	

Panel with Openings - 40 ft (Bartels, 2010)

Appendix B - Stiffness Matrix Assembly MS Excel VBA Code

```
Public Function SMAssemble() As Double
```

```
Dim i As Double 'Counter
```

```
Dim j As Double 'Counter
```

```
Dim g As Double 'Counter
```

```
Dim EFT() As Variant
```

```
Dim nodfat() As Variant
```

```
Dim NumDOF As Double
```

```
Dim KmgsM() As Double 'Master Global Stiffness Matrix
```

```
Dim ncoor() As Double 'Node Coordinates for each element
```

```
Dim E As Double 'E for each element
```

```
Dim Izz As Double 'Izz for each element
```

```
Dim A As Double 'A for each element
```

```
NumNode = Range("Input!A1")
```

```
NumElement = Range("Input!K1")
```

```
NodeList = Range("Input!A3:A" & CStr(NumNode + 2))
```

```
NodeCoord = Range("Input!B3:D" & CStr(NumNode + 2))
```

```
NodeBC = Range("Input!E3:J" & CStr(NumNode + 2))
```

```
ElementList = Range("Input!K3:K" & CStr(NumElement + 2))
```

```
ElementType = Range("Input!L3:L" & CStr(NumElement + 2))
```

```
ElementNodes = Range("Input!M3:N" & CStr(NumElement + 2))
```

```
ElementProps = Range("Input!O3:Q" & CStr(NumElement + 2))
```

```
ElementRelease = Range("Input!R3:S" & CStr(NumElement + 2))
```

```
ReDim EFT(1 To NumElement) As Variant 'Element Freedom Table
```

```
ReDim nodfat(1 To NumNode) As Variant
```

```
EFTAssemble EFT, nodfat
```

```
'Count Number of DOF's
```

```
NumDOF = 0
```

```
For i = 1 To NumNode
```

```
    For j = 1 To 6
```

```
        NumDOF = NumDOF + nodfat(i)(j)
```

```
    Next j
```

```
Next i
```

```
ReDim KmgsM(1 To NumDOF, 1 To NumDOF) As Double 'Master Global Stiffness Matrix
```

```
EFTAssemble EFT, nodfat
```

```

'Main Assembly Loop
Dim k As Variant 'Select which SM to use
ReDim ncoor(1 To 2, 1 To 2) As Double
For g = LBound(EFT, 1) To UBound(EFT, 1)

    ncoor(1, 1) = NodeCoord(ElementNodes(g, 1), 1)
    ncoor(1, 2) = NodeCoord(ElementNodes(g, 1), 2)
    ncoor(2, 1) = NodeCoord(ElementNodes(g, 2), 1)
    ncoor(2, 2) = NodeCoord(ElementNodes(g, 2), 2)
    E = ElementProps(g, 1)
    Izz = ElementProps(g, 2)
    A = ElementProps(g, 3)
    Select Case ElementType(g, 1)
        Case "PlaneBar"
            k = PlaneBar(ncoor, E, A)
        Case "PlaneBeamColumn"
            k = PlaneBeamColumn(ncoor, E, A, Izz, 0)
    End Select

'Inner Assembly Loop
For i = LBound(k, 1) To UBound(k, 1)
    For j = LBound(k, 2) To UBound(k, 2)
        Kmgsm(EFT(g)(i), EFT(g)(j)) = Kmgsm(EFT(g)(i), EFT(g)(j)) + k(i, j)
    Next j
Next i

Next g

Range("Sheet2!H34: R44").Value = Kmgsm

'SkyMatrix Assembly Section
Dim GSDLT As Variant 'Global Skyline Diagonal Location Table
Dim NumEntriesSkyMatrix As Double
Dim GlobalSkyMatrix() As Double

ConstructGSDLT EFT, GSDLT, NumDOF, NumEntriesSkyMatrix

Range("Sheet2!H58:S58").Value = GSDLT
Range("Sheet2!T58").Value = NumEntriesSkyMatrix

ReDim GlobalSkyMatrix(1 To NumEntriesSkyMatrix)

'Main SkyMatrix Assembly Loop
'Dim k As Variant 'Select which SM to use
ReDim ncoor(1 To 2, 1 To 2) As Double
For g = LBound(EFT, 1) To UBound(EFT, 1)

```

```

ncoor(1, 1) = NodeCoord(ElementNodes(g, 1), 1)
ncoor(1, 2) = NodeCoord(ElementNodes(g, 1), 2)
ncoor(2, 1) = NodeCoord(ElementNodes(g, 2), 1)
ncoor(2, 2) = NodeCoord(ElementNodes(g, 2), 2)
E = ElementProps(g, 1)
Izz = ElementProps(g, 2)
A = ElementProps(g, 3)
Select Case ElementType(g, 1)
    Case "PlaneBar"
        k = PlaneBar(ncoor, E, A)
    Case "PlaneBeamColumn"
        k = PlaneBeamColumn(ncoor, E, A, Izz, 0)
End Select

'Inner Assembly Loop
For i = LBound(k, 1) To UBound(k, 1)
    For j = LBound(k, 2) To UBound(k, 2)
        If j >= i Then
            GlobalSkyMatrix(GSDLT(EFT(g)(j)) + EFT(g)(j) - EFT(g)(i)) =
GlobalSkyMatrix(GSDLT(EFT(g)(j)) + EFT(g)(j) - EFT(g)(i)) + k(i, j)
            'Kmgsm(EFT(g)(i), EFT(g)(j)) = Kmgsm(EFT(g)(i), EFT(g)(j)) + k(i, j)
        End If
    Next j
Next i

Next g

Range("Sheet2!H60:R60") = GlobalSkyMatrix

End Function

```

```

Public Function EFTAssemble(EFT As Variant, nodfat As Variant) As Double
Dim g As Double 'Counter
Dim i As Double 'Counter
Dim j As Double 'Counter
Dim nfs() As Double

ReDim nodfmt(1 To NumNode) As Integer
ReDim efs(1 To NumElement, 1 To 2) As Variant

For g = 1 To NumElement
    'Construct EFS

```

For i = 1 To 2 'For looping for first and second node... if more than 2 nodes this value must be increased

```
ReDim nfs(1 To 6) As Double
Select Case ElementType(g, 1)
    Case "PlaneBeamColumn"
        nfs(1) = 1
        nfs(2) = 1
        nfs(6) = 1
    Case "PlaneBar"
        nfs(1) = 1
        nfs(2) = 1
End Select
```

efs(g, i) = nfs

```
'Construct nodfat
If IsEmpty(nodfat(ElementNodes(g, i))) Then
    nodfat(ElementNodes(g, i)) = nfs
Else
    For j = 1 To 6 'Loop through nfs and add to existing
        If nodfat(ElementNodes(g, i))(j) = 0 Then
            nodfat(ElementNodes(g, i))(j) = nfs(j)
        End If
    Next j
End If

Next i
```

Next g

```
'Fill in Empty spots in nodfat
For i = LBound(nodfat, 1) To UBound(nodfat, 1)
    If IsEmpty(nodfat(i)) Then
        nfs(1) = 0
        nfs(2) = 0
        nfs(3) = 0
        nfs(4) = 0
        nfs(5) = 0
        nfs(6) = 0
        nodfat(i) = nfs
    End If
Next i
```

'Construct nodfmt

```

nodfmt(1) = 0 'Starts with 0
For i = 1 To NumNode - 1
    nodfmt(i + 1) = nodfmt(i)
    For j = 1 To 6
        If Not IsEmpty(nodfat(i)) Then
            nodfmt(i + 1) = nodfmt(i + 1) + nodfat(i)(j)
        End If
    Next j
Next i

'Construct EFT - Element Freedom Table
Dim k As Integer 'ief Counter
Dim PreviousNode As Integer 'Node Counter
Dim ief() As Integer

For g = 1 To NumElement
    PreviousNode = 0
    k = 1
    ReDim ief(1 To k)
    For i = 1 To 2
        For j = 1 To 6
            If efs(g, i)(j) = 1 Then
                ReDim Preserve ief(1 To k)

                If ElementNodes(g, i) <> PreviousNode Then
                    ief(k) = nodfmt(ElementNodes(g, i)) + 1
                    k = k + 1
                    PreviousNode = ElementNodes(g, i)
                Else
                    ief(k) = ief(k - 1) + 1
                    k = k + 1
                End If
            End If
        Next j
    Next i
    EFT(g) = ief
Next g

```

```
Public Function ConstructGSDLT(EFT As Variant, GSDLT As Variant, NumDOF As Double,
NumEntriesSkyMatrix As Double) As Variant
'Construct GSDLT Global Skyline Diagonal Location Table
```

```
'Get Skyline of matrix
Dim i As Integer 'Counter
Dim j As Integer 'Counter
Dim Skyline As Variant 'keeps track of skyline for each column
ReDim skylinetable(1 To NumDOF) As Variant
ReDim GSDLT(1 To NumDOF + 1)
```

```
'Loop through SkylineTable to set skyline on diagonals - to be modified later
For i = LBound(skylinetable) To UBound(skylinetable)
    skylinetable(i) = i
Next i
```

```
'Loop through EFT to get skyline table
For i = LBound(EFT) To UBound(EFT)
    For j = LBound(EFT(i)) To UBound(EFT(i))
        If j = 1 Then
            Skyline = EFT(i)(j)
        Else
            Skyline = Min(CDbl(Skyline), CDbl(EFT(i)(j))) 'Get minimum to use for all columns
in this element of the EFT
        End If
    Next j
```

```
    For j = LBound(EFT(i)) To UBound(EFT(i))
        skylinetable(EFT(i)(j)) = Min(CDbl(skylinetable(EFT(i)(j))), CDbl(Skyline))
    Next j
Next i
```

```
'Loop through skyline table to generate GSDLT
GSDLT(1) = 1
For i = 2 To NumDOF + 1
    GSDLT(i) = GSDLT(i - 1) + i - skylinetable(i - 1)
Next i
```

```
NumEntriesSkyMatrix = GSDLT(NumDOF) + NumDOF + 1 - skylinetable(NumDOF) - 1
```

```
End Function
```


Appendix C - Stiffness Matrix Formulation MS Excel VBA Code

```
Public Function PlaneBar(ncoor As Variant, E As Variant, A As Variant) As Variant
Dim x1 As Double
Dim x2 As Double
Dim y1 As Double
Dim y2 As Double

x1 = ncoor(1, 1)
y1 = ncoor(1, 2)
x2 = ncoor(2, 1)
y2 = ncoor(2, 2)

Dim x21 As Double
Dim y21 As Double

x21 = x2 - x1
y21 = y2 - y1

Dim L As Double
L = Sqr(x21 ^ 2 + y21 ^ 2)

Dim Ke(1 To 4, 1 To 4) As Double

Ke(1, 1) = x21 * x21 * E * A / L ^ 3
Ke(1, 2) = x21 * y21 * E * A / L ^ 3
Ke(1, 3) = -x21 * x21 * E * A / L ^ 3
Ke(1, 4) = -x21 * y21 * E * A / L ^ 3
Ke(2, 2) = y21 * y21 * E * A / L ^ 3
Ke(2, 3) = -x21 * y21 * E * A / L ^ 3
Ke(2, 4) = -y21 * y21 * E * A / L ^ 3
Ke(3, 3) = x21 * x21 * E * A / L ^ 3
Ke(3, 4) = x21 * y21 * E * A / L ^ 3
Ke(4, 4) = y21 * y21 * E * A / L ^ 3

MakeSymmetrical Ke

PlaneBar = Ke
End Function

Public Function PlaneBeamColumn(ncoor As Variant, E As Double, A As Double, Izz As
Double, P As Double) As Variant
Dim i As Double 'Counter
Dim j As Double 'Counter
Dim g As Double 'Counter
```

```
Dim x1 As Double
Dim x2 As Double
Dim y1 As Double
Dim y2 As Double
```

```
x1 = ncoor(1, 1)
y1 = ncoor(1, 2)
x2 = ncoor(2, 1)
y2 = ncoor(2, 2)
```

```
Dim x21 As Double
Dim y21 As Double
```

```
x21 = x2 - x1
y21 = y2 - y1
```

```
Dim L As Double
L = Sqr(x21 ^ 2 + y21 ^ 2)
```

```
'Variables for geometric stiffness
Dim k As Double 'k factor
Dim Phic As Double 'Compressive Phi
Dim Phit As Double 'Tensile Phi
Dim Phi1 As Double 'Phi1 Factor
Dim Phi2 As Double 'Phi2 Factor
Dim Phi3 As Double 'Phi3 Factor
Dim Phi4 As Double 'Phi4 Factor
```

```
k = Sqr(Abs(P) / (E * Izz))
```

```
'If not considered a column, only a beam, revert to regular stiffness matrix
If P = 0 Then
```

```
    Phi1 = 1
    Phi2 = 1
    Phi3 = 1
    Phi4 = 1
```

```
End If
```

```
'If compressive, modify stiffness matrix
```

```
If P < 0 Then
```

```
    Phic = 2 - 2 * Cos(k * L) - k * L * Sin(k * L)
    Phi1 = (k * L) ^ 3 * Sin(k * L) / (12 * Phic)
    Phi2 = (k * L) ^ 2 * (1 - Cos(k * L)) / (6 * Phic)
    Phi3 = (k * L) * (Sin(k * L) - k * L * Cos(k * L)) / (4 * Phic)
    Phi4 = (k * L) * (k * L - Sin(k * L)) / (2 * Phic)
```

```
End If
```

'If compressive, modify stiffness matrix

If $P > 0$ Then

$$\text{Phit} = 2 - 2 * \cosh(k * L) - k * L * \sinh(k * L)$$

$$\text{Phi1} = (k * L)^3 * \sinh(k * L) / (12 * \text{Phit})$$

$$\text{Phi2} = (k * L)^2 * (\cosh(k * L) - 1) / (6 * \text{Phit})$$

$$\text{Phi3} = (k * L) * (k * L * \cosh(k * L) - \sinh(k * L)) / (4 * \text{Phit})$$

$$\text{Phi4} = (k * L) * (\sinh(k * L) - k * L) / (2 * \text{Phit})$$

End If

Dim Ke(1 To 6, 1 To 6) As Double

$$\text{Ke}(1, 1) = E * A / L$$

$$\text{Ke}(1, 4) = -E * A / L$$

$$\text{Ke}(4, 4) = E * A / L$$

$$\text{Ke}(2, 2) = 12 * E * I_{zz} * \text{Phi1} / L^3$$

$$\text{Ke}(2, 5) = -12 * E * I_{zz} * \text{Phi1} / L^3$$

$$\text{Ke}(5, 5) = 12 * E * I_{zz} * \text{Phi1} / L^3$$

$$\text{Ke}(2, 3) = 6 * E * I_{zz} * \text{Phi2} / L^2$$

$$\text{Ke}(2, 6) = 6 * E * I_{zz} * \text{Phi2} / L^2$$

$$\text{Ke}(3, 5) = -6 * E * I_{zz} * \text{Phi2} / L^2$$

$$\text{Ke}(5, 6) = -6 * E * I_{zz} * \text{Phi2} / L^2$$

$$\text{Ke}(3, 3) = 4 * E * I_{zz} * \text{Phi3} / L$$

$$\text{Ke}(6, 6) = 4 * E * I_{zz} * \text{Phi3} / L$$

$$\text{Ke}(3, 6) = 2 * E * I_{zz} * \text{Phi4} / L$$

MakeSymmetrical Ke

'Make Transformation Matrix

Dim Te(1 To 6, 1 To 6) As Double

$$\text{Te}(1, 1) = x21 / L$$

$$\text{Te}(1, 2) = y21 / L$$

$$\text{Te}(2, 1) = -y21 / L$$

$$\text{Te}(2, 2) = x21 / L$$

$$\text{Te}(3, 3) = 1$$

$$\text{Te}(4, 4) = x21 / L$$

$$\text{Te}(4, 5) = y21 / L$$

$$\text{Te}(5, 4) = -y21 / L$$

$$\text{Te}(5, 5) = x21 / L$$

$Te(6, 6) = 1$

'Apply transformation matrix to local matrix to make global matrix

PlaneBeamColumn = MatMultiply(MatMultiply(MatTranspose(Te), Ke), Te)

End Function

Appendix D - Skymatrix Solver MS Excel VBA Code

Public Function SkySolve(Mat() As Double, GSDLT() As Double, Vect() As Double) As Variant

'Mat(), GSDLT(), and Vect() input must be in one dimensional arrays

Dim j As Integer

Dim i As Integer

Dim r As Integer

Dim Skyline As Double 'First nonzero element in column

'Decompose Stiffness Matrix

'Refer to Pg 709 of "Finite Element Procedures" for details

For j = LBound(GSDLT) + 1 To UBound(GSDLT) - 1 'Loop over all columns

Skyline = j + 1 - (GSDLT(j + 1) - GSDLT(j)) 'Diagonal +1 - Height of column

'(First Loop over height of column from top down to get in place factors)

For i = Skyline + 1 To j - 1

For r = Skyline To i - 1

If SkyLookup(r, i, GSDLT) <> -1 Then 'Check to see if this term is inside the skyline

Mat(SkyLookup(i, j, GSDLT)) = Mat(SkyLookup(i, j, GSDLT)) -
Mat(SkyLookup(r, i, GSDLT)) * Mat(SkyLookup(r, j, GSDLT))

End If

Next r

Next i

'Divide gij by dii to get lij

For i = Skyline To j - 1

Mat(SkyLookup(i, j, GSDLT)) = Mat(SkyLookup(i, j, GSDLT)) / Mat(SkyLookup(i, i,
GSDLT))

Next i

'Diagonal Elements

For r = Skyline To j - 1

Mat(SkyLookup(j, j, GSDLT)) = Mat(SkyLookup(j, j, GSDLT)) - Mat(SkyLookup(r, j,
GSDLT)) * Mat(SkyLookup(r, j, GSDLT)) * Mat(SkyLookup(r, r, GSDLT))

Next r

Next j

'Reduce vect

For i = LBound(Vect) + 1 To UBound(Vect)

Skyline = i + 1 - (GSDLT(i + 1) - GSDLT(i)) 'Diagonal +1 - Height of column

For r = Skyline To i - 1

```

    Vect(i) = Vect(i) - Mat(SkyLookup(r, i, GSDLT)) * Vect(r)

Next r
Next i

'Back Substitution
For i = LBound(Vect) To UBound(Vect)
    Vect(i) = Vect(i) / Mat(SkyLookup(i, i, GSDLT))
Next i

For i = UBound(Vect) To LBound(Vect) + 1 Step -1
    Skyline = i + 1 - (GSDLT(i + 1) - GSDLT(i)) 'Get skyline of current element

    For r = Skyline To i - 1
        Vect(r) = Vect(r) - Mat(SkyLookup(r, i, GSDLT)) * Vect(i)
    Next r
Next i

'Test Output
For i = LBound(Vect) To UBound(Vect)
    MsgBox Vect(i)
Next i

For i = LBound(Mat) To UBound(Mat)
    MsgBox Mat(i)
Next i

End Function
Public Function SkyLookup(Row As Integer, Column As Integer, GSDLT As Variant) As Variant
If Column - Row + 1 > GSDLT(Column + 1) - GSDLT(Column) Then 'Check to see if the
element falls outside of the skyline
    SkyLookup = -1 'If it does, set skyloop to -1 as a flag that it is outside the skyline
Else
    SkyLookup = GSDLT(Column) + Column - Row
End If
End Function

```

Appendix E - Gaussian Elimination Solver with Pivoting MS Excel

VBA Code

```
Public Function GESolve(Mat As Variant, Vect As Variant, Optional DetSign As Variant) As Variant
    Dim NumEq As Long, MBound As Long, i As Long, j As Long, k As Long
    Dim Df As Double, SSum As Double, ResultA() As Double

    DetSign = 1 'Sets initial Determinant Sign

    NumEq = UBound(Mat) - LBound(Mat) + 1
    If NumEq <= 0 Then Exit Function
    MBound = NumEq
    ReDim ResultA(1 To MBound, 1 To 1)

    For k = 1 To MBound - 1
        For i = k + 1 To MBound
            If Abs(Mat(k, k)) < Abs(Mat(i, k)) Then
                Call swapRowMat(i, k, MBound, Mat)
                Call swapRowMat(i, k, 1, Vect)

                DetSign = DetSign * -1 'Switches sign of determinant when a row is swapped
            End If
            If Mat(k, k) <> 0 Then
                Df = Mat(i, k) / Mat(k, k)
                For j = k To MBound
                    Mat(i, j) = Mat(i, j) - Mat(k, j) * Df
                Next j
                Vect(i, 1) = Vect(i, 1) - Vect(k, 1) * Df
            End If
        Next i
    Next k

    'Multiplies the diagonal elements to return the sign of the determinant
    For i = 1 To MBound
        DetSign = DetSign * Sgn(Mat(i, i))
    Next i

    ResultA(MBound, 1) = Vect(MBound, 1) / Mat(MBound, MBound)
    For i = MBound - 1 To 1 Step -1
        SSum = 0
        For j = i + 1 To MBound
            SSum = SSum + Mat(i, j) * ResultA(j, 1)
        Next j
```

```

    If Mat(i, i) <> 0 Then ResultA(i, 1) = (Vect(i, 1) - SSum) / Mat(i, i)
Next i
GESolve = ResultA

```

End Function

```

Private Sub swapRowMat(ByVal r1 As Long, ByVal r2 As Long, MBound As Long, Mat)
    Dim i As Long, mB As Long, dVal As Double

```

```

    For i = 1 To MBound
        dVal = Mat(r1, i)
        Mat(r1, i) = Mat(r2, i)
        Mat(r2, i) = dVal
    Next i
End Sub

```

```

Public Function GetArray(Arrayname As Variant) As Variant
    Dim TempA() As Variant, LBound1 As Long, UBound1 As Long, UBound2 As Long
    Dim i As Long, j As Long

```

```

' This function converts a single or multi-cell range into an array, and
' converts a 1D array into single row or single column 2D base 1 array

```

```

' If Arrayname is not an array, convert it into one.
' IsArray is true for multi-cell ranges, but not for a single cell range
    If Not IsArray(Arrayname) Then
        Arrayname = Array(Arrayname)
    End If

```

```

' If Arrayname is a range, convert it into an array containing the range cell values
    If TypeName(Arrayname) = "Range" Then
        GetArray = Arrayname.Value2
' Otherwise simply allocate the array to GetArray
    Else
        GetArray = Arrayname
    End If

```

```

' Check for a 1D array, or a base 0 array
    On Error Resume Next
    UBound2 = UBound(GetArray, 2)
' Convert to base 1
    If UBound2 = 0 Then
        LBound1 = LBound(GetArray)
        UBound1 = UBound(GetArray)

```



```
ReDim TempA(1 To 1, 1 To UBound1 - LBound1 + 1)
j = 1
For i = LBound1 To UBound1
    TempA(1, j) = Arrayname(i)
    j = j + 1
Next i

    GetArray = TempA
End If

End Function
```