Graphical User Interface and Teaching Aid for Moment Curvature Analysis

By John E. Warren

B.S. in Civil Engineering, December 1998, VPI&SU

A Theses Submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Masters of Science

May 20, 2012

Theses directed by

Pedro Silva
Professor of Engineering and Applied Science
Dedication

I dedicate this Thesis to my wife, Linh Nguyen Warren for her constant support over the period of time while writing my Thesis.
Acknowledgements

I would like to thank my advisor, Dr. Pedro F. Silva for his help and encouragement on the development of MCAP. Also to the other professors who have helped me learn and advance in an academic level at GWU. I would also like to thank my parents, J. Fred and Virginia Warren for all of their support especially in my education.
Abstract

Graphical User Interface and Teaching Aid for Moment Curvature Analysis

Moment Curvature Analysis Program (MCAP) has been developed for establishing moment-curvature analyses using an interface between a MATLAB® Graphical User Interface (GUI) and the finite element program Open System for Earthquake Engineering Simulation (OpenSees). The MATLAB® GUI was developed with the main goal of facilitating the use and interpretation of OpenSees. The combination of these two software programs provides a user friendly interface that can be used to supplement classroom instruction and enhance learning the response of sections using accurate nonlinear material stress-strain relationships.

MCAP was primarily developed within a framework that guides the user through a logical sequence of steps for determining the moment and rotation capacity of commonly used structural sections. The simplicity of its use and its framework makes it particularly suitable as a teaching aid for graduates and undergraduates to analyze irregular and commonly used structural sections. Moment-curvature relationships are of primary importance in evaluating the nonlinear behavior of sections, members and structural systems.

Some features of MCAP can be summarized as follows: (1) MCAP has been extended to analyze rectangular, circular and user defined arbitrary shape sections. (2) The development of a GUI interface for a user defined arbitrary shape section is significant as a teaching aid since most Moment-Curvature
programs do not allow the user to define and analyze sections other than rectangular and circular sections. (3) MCAP also has the capability to perform a Moment-Curvature analysis on a rotated rectangular section.

To further enhance classroom instruction a supplementary moment-curvature program was also implemented using MATLAB®. In its present form, this supplementary program can be used by users in tracing the programming steps required to develop a moment curvature analysis and subsequent load deformation response of rectangular reinforced concrete sections. One of the innovative features in the programming of this supplementary program is the implementation of a linear relationship between the curvature and the strains in the extreme steel fibers. To the best knowledge of the author, this linear relationship has not been previously used in encoding the moment curvature program. The main implication of this relationship is a significant decrease in the number of iterations required for convergence.

Future developments for MCAP include a pile analysis module and expanding the program to perform a pushover analyses for reinforced concrete members. The implementation of a FRP confined section module is also being developed with an immediate application to the retrofit of circular and rectangular concrete sections. Further research on the relationship between the curvature and the strains in the extreme steel fibers is a future development that MCAP can help accomplish.
# Table of Contents

Dedication ................................................................................................................................. ii

Acknowledgements .................................................................................................................... iii

Abstract ........................................................................................................................................ iv

Table of Contents ....................................................................................................................... vi

List of Figures ............................................................................................................................. ix

List of Symbols ........................................................................................................................ xii

List of Equations ....................................................................................................................... xv

Chapter 1 - Introduction ............................................................................................................ 1

1.1. Objective ............................................................................................................................ 2

1.2. Results ............................................................................................................................... 3

1.3. Organization ....................................................................................................................... 3

Chapter 2 - Literature Review: Moment-Curvature Analysis ................................................. 4

2.1. Moment-Curvature Procedure .......................................................................................... 4

2.2. Nonlinear Material Models ............................................................................................. 9

2.2.1. Unconfined Concrete Models ...................................................................................... 9

2.2.1.1. Tsai’s Model (Chang and Mander 1994, 3-9) .................................................... 10

2.2.1.2. Popovic’s Model (Chang and Mander 1994, 3-5) ........................................... 10

2.2.1.3. Young’s Models (Chang and Mander 1994, 3-3) ......................................... 10

2.2.1.4. Mirza & Hsu Model (Chang and Mander 1994, 3-8) ................................. 11

2.2.1.5. Chang and Mander Model (Chang and Mander 1994, 3-12) ...................... 11

2.2.2. Confined Concrete ..................................................................................................... 12

2.2.2.1. Chang and Mander Model .............................................................................. 12
2.2.2.2. Piecewise Models ................................................................. 15
2.2.3. Maximum Concrete Strain ........................................................... 15
2.2.4. Reinforcing Steel ........................................................................ 16
  2.2.4.1. Kent and Park Model (Kent and Park, 1973, 98-103).......... 17
  2.2.4.2. Park Model (Park and Paulay 1975, 229) ..................... 17
  2.2.4.3. Chang and Mander Model (Chang and Mander 1994, 2-1 to 2-
  2) ........................................................................................... 18
2.4. Available Moment-Curvature Computer Programs ....................... 20
  2.4.1. OpenSees (Mazzoni et al., 2006) ............................................. 20
    2.4.1.1. Modeling Commands ................................................. 21
    2.4.1.2. Analysis Commands .................................................. 22
    2.4.1.3. Recorder Commands ................................................ 23
    2.4.1.4. Miscellaneous Commands ....................................... 23
  2.4.2. SeismoStruct (Seismosoft 2011) ............................................ 23
  2.4.3. Response 2000 (Bentz 2011) .............................................. 24
  2.4.4. BENT (Silva et al.) ......................................................... 24
  2.4.5. CONSEC (Matthews 2009) .............................................. 25
2.5. MATLAB® Interface Used for MCAP ........................................... 25
  2.5.1. Variables ............................................................................. 26
  2.5.2. Functions ........................................................................... 26
  2.5.3. GUIDE and GUI Interfaces ............................................... 27
  2.5.4. Creating and Running OpenSees Files ................................. 28
  2.5.5. Data Analysis and Plotting ................................................. 28
Chapter 3 - MCAP User’s Manual and Development Information .......... 30

3.1. Installation Instructions ............................................................................. 30
3.2. Running MCAP ...................................................................................... 30
3.3. Section Analysis Title Screens ............................................................... 32
3.4. Defining Sections ................................................................................... 33
3.5. Material Properties ................................................................................. 44
3.6. Create *.Tcl File ..................................................................................... 46
3.7. Analyzing the Section ............................................................................. 50
3.8. Moment-Curvature Plot (and Data) ........................................................ 50
3.9. Results for Test Sections ....................................................................... 51
3.10. Testing & Verification of MCAP ............................................................ 53

Chapter 4 - MCAP Applications ....................................................................... 58

4.1. MCAP as a Teaching Aid ....................................................................... 58
4.2. MCAP as a basis to find Force-Deformation Relationship ...................... 59
4.3. MCAP as an analysis Tool ..................................................................... 60
4.4. Relationship Between Reinforcing Strain and Curvature ...................... 60
4.5. Using the Reinforcing Strain – Curvature Relationship .......................... 66

Chapter 5 - Conclusions in the Development of MCAP ............................... 70

References ........................................................................................................ 72

Appendix A – Rectangular Section Example *.tcl File ................................... 74
Appendix B – Circular Section Example *.tcl File .......................................... 79
Appendix C – User Defined Section Example *.tcl File ................................. 84
Appendix D – Rectangular Practice Program ................................................ 91
List of Figures

Figure 2-1: Typical Section, Strain Distribution and Actual Force Distribution.... 5
Figure 2-2: Discretization of a Concrete Section in Fibers.............................. 6
Figure 2-3: Unconfined Concrete Model Comparison........................................ 12
Figure 2-4: Park, Priestley and Gill Concrete Model (A), Shiekh and Uzumeri Concrete Model (B) ........................................................................................................ 15
Figure 2-5: Reinforcing Steel Model Comparison............................................. 19
Figure 2-6: Bent Application ........................................................................... 25
Figure 3-1: Command to Run MCAP Once Installed ........................................ 31
Figure 3-2: MCAP Title Screen ....................................................................... 31
Figure 3-3: Choose Section Analysis Type....................................................... 32
Figure 3-4: Module Title Screen ..................................................................... 33
Figure 3-5: Define Section Button ................................................................... 35
Figure 3-6: Rectangular Section Definition ..................................................... 36
Figure 3-7: Circular Section Definition ............................................................ 36
Figure 3-8: User Defined Section Definition .................................................. 37
Figure 3-9: Rotated Test Section Results ....................................................... 38
Figure 3-10: Draw Rectangular Section............................................................ 39
Figure 3-11: Draw Circular Section................................................................. 39
Figure 3-12: User Defined Section Drawing Canvas .................................... 40
Figure 3-13: User Defined Section ................................................................. 41
Figure 3-14: Define the Transverse Reinforcement....................................... 42
Figure 3-15: Define Longitudinal Reinforcement ......................................... 42
List of Tables

Table 3- 1: Rectangular Example Section ................................................................. 34
Table 3- 2: Circular Example Section ................................................................. 35
Table 3- 3: “I” Section Properties for Verification ................................................. 55
Table 4- 1: Strain vs. Curvature Test Sections ....................................................... 61
List of Symbols

$A_{sp}$ – Area Transverse Reinforcement

$A_{sx}$ – Area Transverse Reinforcement Running in X Direction

$A_{sy}$ – Area Transverse Reinforcement Running in Y Direction

$D$ – Diameter

$E$ – Modulus of Elasticity

$E_c$ – Modulus of Elasticity of Concrete

$E_s$ – Modulus of Elasticity of Steel

$E_{sec}$ – Tangent Modulus of Elasticity - Concrete

$E_{sh}$ – Modulus of Elasticity of Steel at Onset of Strain Hardening

$F_c$ – Internal Compressive Force

$F_y$ – Internal Steel Tensile Force

$F_{si}$ – Internal at $i^{th}$ Steel Force

$I$ – Second Moment of Inertia

$K_e$ – Confinement Effectiveness Coefficient

$M$ – Moment

$P$ – Axial Force

$b$ – Section Width

$b_c$ – Core Dimension to Center Line of Perimeter Reinforcement in X Direction

$c$ – Distance from Compressive Extreme Layer to Neutral Axis

$d_c$ – Core Dimension to Center Line of Perimeter Reinforcement in Y Direction

$d_{ci}$ – Distance to $i^{th}$ concrete fiber

$d_s$ – Diameter to Center Line of Spiral or Hoop
dsi – Distance to ith steel fiber

f – Stress

f_{ch} – Characteristic Stress of Steel

f_u – Ultimate Stress of Steel

f_y – Yield Stress of Steel

f_{yh} – Yield Stress of Transverse Reinforcement

f_c’ – Peak Stress of Unconfined Concrete, 28 Day Strength

f_{cc} – Peak Stress of Confined Concrete

f’_i – Effective Confining Stress

h – Section Height

r – Factor to Control Descending Branch of Conc. Stress – Strain Relationship

s – Center Line to Center Line Spacing of Spiral or Hoop Reinforcement

s’ – Clear Space Between Spiral or Hoop Reinforcement

w’_i – ith Clear Distance Between Longitudinal Bars

\varepsilon – Strain

\varepsilon_{cc} – Peak Strain Confined Concrete

\varepsilon_{ci} – Strain at ith Concrete Fiber

\varepsilon_{cu} – Ultimate Strain of Concrete

\varepsilon_{ip} – Plastic Strain of Previous Loading

\varepsilon_{sh} – Strain at Onset of Strain Hardening

\varepsilon_{si} – Strain at ith Steel Fiber

\varphi – Curvature

\rho_{cc} – Ratio of Area of Longitudinal Reinforcement to Concrete Core Area
$\rho_s$ – Volumetric Ratio of Confining Steel

$\rho_x$ – Ratio of Trans. Reinf. Area to Volume of Confined Concrete Core X Direction

$\rho_y$ – Ratio of Trans. Reinf. Area to Volume of Confined Concrete Core Y Direction
List of Equations

(2. 1) .......................................................................................................................... 5
(2. 2) .......................................................................................................................... 5
(2. 3) .......................................................................................................................... 7
(2. 4) .......................................................................................................................... 7
(2. 5) .......................................................................................................................... 8
(2. 6) .......................................................................................................................... 10
(2. 7) .......................................................................................................................... 10
(2. 8) .......................................................................................................................... 10
(2. 9) .......................................................................................................................... 10
(2. 10) ......................................................................................................................... 11
(2. 11) ......................................................................................................................... 11
(2. 12) ......................................................................................................................... 11
(2. 13) ......................................................................................................................... 12
(2. 14) ......................................................................................................................... 13
(2. 15) ......................................................................................................................... 13
(2. 16) ......................................................................................................................... 14
(2. 17) ......................................................................................................................... 14
(2. 18) ......................................................................................................................... 15
(2. 19) ......................................................................................................................... 16
(2. 20) ......................................................................................................................... 17
(2. 21) ......................................................................................................................... 17
(2. 22) ......................................................................................................................... 17
Chapter 1 - Introduction

Moment Curvature Analysis Program (MCAP) serves primarily as an interface between a MATLAB® Graphical User Interface (GUI) and the finite element program Open System for Earthquake Engineering Simulation (OpenSees). MCAP was developed with the main goal of facilitating the use and interpretation of OpenSees while providing undergraduates with an easy-to-use and friendly educational computational tool for evaluating: (1) the moment and rotation capacity of a section, (2) the effect that nonlinear material relationships play on a section moment-curvature (M-φ) capacity. As such, MCAP provides a framework that guides the user through a logical sequence of steps for determining the moment and rotation capacity of commonly used structural sections.

The overall goal of MCAP is that it can be used as a teaching aid for undergraduates learning material properties, it can serve as an introduction to the design of reinforced concrete sections and it can aid graduates as an analysis tool for studying non-linear design of reinforced concrete (RC) structures. The simplicity of MCAP usage and its framework makes it particularly suitable as a teaching aid for graduates and undergraduates to analyze irregular and commonly used structural sections. As such, users should find MCAP to be relatively intuitive and simple to use.
1.1. Objective

The main objective of this work was to develop an easy-to-use and friendly user interface educational computational tool for obtaining the moment and curvature relationships of commonly used structural sections. Some features of MCAP can be summarized as follows: (1) MCAP has been extended to analyze rectangular, circular and user defined arbitrary shape sections. (2) The development of a GUI interface for a user defined arbitrary shape section is significant as a teaching aid since most Moment-Curvature programs do not allow the user to define and analyze sections other than rectangular and circular sections. (3) MCAP also has the capability to perform a Moment-Curvature analysis on a rotated rectangular section.

MCAP is of primary interest to those users first learning the nonlinear stress-strain relationships of commonly used materials in the construction of structures. For instance, users may evaluate how changes in a reinforcing steel or concrete material property parameter can affect a reinforced concrete section’s strength. In other cases, users designing reinforced concrete sections may use this program to check their hand calculations by computing the ultimate moment capacity of a section. Finally, more advanced users can use this program to compute the ultimate moment capacity for a section with an arbitrary cross sectional shape and reinforcement configuration, such as I-sections and other more complex configurations.
1.2. Results

The results of a successful program will be an effective teaching aid which combines the Graphical User Interface of MATLAB® and the power and efficiency of OpenSees to create Moment-Curvature analyses.

The relationship between curvature and the strain in the extreme steel tension layer is shown to be linear. This simple finding provides for faster convergence in M-\( \phi \) programs by being able to predict strain levels at the different steps in the programs.

1.3. Organization

This thesis consists of 5 chapters. Chapter 1 provides an introduction and main objectives of the thesis. Chapter 2 provides a succinct review of the literature and software programs that are commonly used for conducting an M-\( \phi \) analysis as well as the interfaces used to create MCAP. A description of the MCAP Modules and User’s Manual is provided in Chapter 3. Chapter 4 discusses the applications of MCAP and some findings concerning the relationship between curvature and the tensile strain in the extreme reinforcing layer. Chapter 5 includes conclusions about MCAP.
Chapter 2 - Literature Review: Moment-Curvature Analysis

MCAP is mainly an interface program that uses graphics building subroutines from the standard MATLAB® library and subsequently creates the input files that are necessary to run the OpenSees procedures and commands required to perform M-φ analyses. In order to expedite the use and/or understanding of MCAP, this section presents an overview of typical information necessary to develop an M-φ analyses program. Salient points of an M-φ analysis program are: (1) geometry of the section and the respective modeling parameters necessary to model the cross section of a member, (2) the nonlinear stress-strain relationships to model the material properties used in the section, (3) the ultimate or limiting strains for each of the materials, and (4) mathematical formulation of the program. Understanding many of these items was an essential part in creating the MCAP graphical user interface program.

Certainly, there are several other programs which can perform M-φ analysis and each program has its merits and drawbacks. This section includes an overview of programs current available in the literature for use in moment-curvature analysis and concludes with an overview of MATLAB®, which served as the GUI for MCAP.

2.1. Moment-Curvature Procedure

There exists an extensive body of information in the literature that details the steps involved in performing a Moment-Curvature analysis for either using simple
hand calculations or for computer implementation. Figure 2-1a shows typical section information required to establish an \( M-\phi \) analysis.

![Diagram of section, strain, and force distributions](image)

**Figure 2-1: Typical Section, Strain Distribution and Actual Force Distribution**

From its sectional geometry and for the given curvature, \( \phi \), Figure 2-1b depicts the linear variation of strains along the section height by ensuring that strains at a constant depth, \( d_{bi} \), are identical. According to the Euler-Bernoulli theory of slender beams the linear variation of strains satisfies the principle that ‘plane sections’ before loading ‘remain plane’ after loading. This infers that any deformations within the section caused by shear are not accounted for in the analysis. As such, the strains in the steel bars at any depth, \( d_{si} \), are obtained based on the strain compatibility equation 2.1:

\[
\varepsilon_{si} = \varepsilon_{cc} - \phi d_{si} \tag{2.1}
\]

Likewise, the strains in the concrete at any depth, \( d_{ci} \), are obtained based on the strain compatibility equation 2.2:

\[
\varepsilon_{ci} = \varepsilon_{cc} - \phi d_{ci} \tag{2.2}
\]
This linear variation of strains makes it possible to create the stress distribution of Figure 2-1c, and subsequently the internal forces equilibrium necessary to obtain the moment capacity of the section.

The first step in an M-\(\phi\) analysis program is the division of the section in fibers or segments, which is schematically shown in Figure 2-2. Each of these fibers or segments are properly defined in terms of their center position to a reference line, which in Figure 2-2 is the top of the section, area, and as importantly the nonlinear stress-strain model.

![Figure 2-2: Discretization of a Concrete Section in Fibers](image)

The next step involves assigning the curvature, \(\phi\), and estimating the strains in the top fibers, \(\varepsilon_c\), which in conjunction with the fibers position, \(d_{si}\) or \(d_{ct}\), are used to develop the strain compatibility equations.

To further enhance classroom instruction a supplementary moment-curvature program was also implemented using MATLAB®. This supplementary program was created to allow users unfamiliar with a moment curvature analysis to understand the steps required to such an analysis. The code for this
supplementary program is in Appendix D. One of the innovative features in the programming of this supplementary program is the implementation of a linear relationship between the curvature and the strains in the extreme steel fibers, which was used to reduce the iterations required to convergence, which is discussed next. The rationale to establish the curvature increments involved in MCAP are outlined later within this section.

Once the strain distribution is determined along the depth of the section, the stresses in each of the fibers or segments is formulated from the respective nonlinear stress-strain models.

The next step involves computing the internal forces in each of the fibers or segments. In conjunction with the externally applied load, \( P_U \), the internal forces in the concrete, \( F_c = \sum \sigma_{ci} A_{ci} \), and in the reinforcing steel, \( F_s = \sum \sigma_{si} A_{si} \), are used in equation 2.3 to determine the equilibrium conditions of the section.

\[
\sum \sigma_{ci} A_{ci} + \sum \sigma_{si} A_{si} - P_U = 0 \tag{2.3}
\]

If the equality in the equation above is satisfied the program proceeds to the next step, otherwise the strains in the top fibers, \( \varepsilon_c \), are estimated once again and the procedure repeats itself until the equality is satisfied.

The next step involves computing the moment capacity, \( M_\phi \), for the assigned curvature in terms of equation 2.4:

\[
M_\phi = \sum \sigma_{ci} A_{ci} (c - d_{ci}) + \sum \sigma_{si} A_{si} (c - d_{si}) \tag{2.4}
\]
This stepwise procedure in a moment curvature program repeats itself for a range of curvatures up to the maximum curvature. The maximum curvature is determined by the curvature at which the maximum tensile strain or compressive strain in the extreme fibers of the section occurs. A similar procedure can be found in “Earthquake Engineering From Engineering Seismology to Performance-Based Engineering” by Bozorgnia & Bertero (Bozorgnia & Bertero 2004, 13-7 to 13-13).

The selection of the curvatures for MCAP was based on an estimated yield curvature, \( \phi_y \), using the following relationships developed by Priestley (Priestley Seible and Calvi 1996, 555). The User Defined Sections use the same formula as the rectangular section with the height of the maximum height of the section.

\[
\phi_y = 2.1 \frac{\varepsilon_y}{h} \quad \text{(Rectangular)}
\]

\[
\phi_y = 2.1 \frac{\varepsilon_y}{D} \quad \text{(Circular)}
\]  

(2.5)

This relationship for estimating the yield curvature is subsequently used to establish a maximum analysis point of 50\( \phi_y \) for the ultimate curvature. The increment for the curvature to obtain the complete envelop for the moment curvature analysis is chosen as 0.05 \( \phi_y \) and the analysis iterates through all these curvatures. The data is later truncated by applying the maximum compressive or tensile strain condition at the extreme fiber.

The Structware® website also lists an article detailing a similar simplified process. The article was written by Robert Matthews. This article is the basis of the calculation for the CONSEC Program (Matthews 2001).
An analysis routine performed by integration of basic equations can be found in “Seismic Design Aids for Nonlinear Analysis of Reinforced Concrete Structures” by Chandrasekaran et al. The procedure outlined includes calculation of the moment for a given curvature by integration and evaluation of constants of integration (Chandrasekaran et al. 2010, 45-88). This process lends itself more readily to spreadsheets.

2.2. Nonlinear Material Models

It is critical in any Moment-Curvature analysis to apply an appropriate nonlinear material model for either the concrete and/or the steel. Confined concrete, unconfined concrete and steel models were used for MCAP. Much work has been done in the past concerning models for confined concrete, unconfined concrete and reinforcing steel, and the models used in MCAP are discussed in the next sections.

2.2.1. Unconfined Concrete Models

Chang and Mander presented a review of concrete models and equations (Eq 2.6-2.13) for these models are shown below (Chang and Mander 1994, 3-1 to 3-21).
2.2.1.1. Tsai’s Model (Chang and Mander 1994, 3-9)

Tsai’s Model, which is of the form of Equation 2.6 can describe confined and unconfined concrete.

\[ y = \frac{nx}{1 + \left( n - \frac{r}{r-1} \right) + \frac{x^r}{r-1}} \quad (2.6) \]

Where, \( y = \frac{f}{f_{ucc}} \) and \( x = \frac{e}{e_{ucc}} \) (for all equations presented)

2.2.1.2. Popovic’s Model (Chang and Mander 1994, 3-5)

Tsai’s model is a generalization of Popovic’s Model shown in Equation 2.7:

\[ y = \frac{rx}{r - 1 + x^r} \quad (2.7) \]

2.2.1.3. Young’s Models (Chang and Mander 1994, 3-3)

Young developed at three different relationships (Equations 2.8, 2.9, 2.10) for the stress strain relationship of unconfined concrete:

\[ y = x[(n - 2)x^2 - (2n - 3)x + n] \quad (2.8) \]

\[ y = xe^{(1-x)} \quad (2.9) \]
\[ y = \sin\left(\frac{\pi}{2} x\right) \] \hspace{1cm} (2. 10)

Where, \( n = \frac{E_{c}e^{1/\epsilon}}{f'_{cc}} \)

2.2.1.4. Mirza & Hsu Model (Chang and Mander 1994, 3-8)

Mirza and Hsu proposed a stress-strain relationship for unconfined concrete detailed by equations 2.11 and 2.12:

\[ y = \sin\left(\frac{\pi}{2} x\right) + 0.2x(x - 1)(e^{(1-x)} - 1) \] \hspace{1cm} (2. 11)

Where \( x \in [0, 1] \)

\[ y = 0.226 + 2.157x - 1.91x^2 + 0.596x^3 - 0.064x^4 \] \hspace{1cm} (2. 12)

Where \( x \in [1, 3.4] \)

2.2.1.5. Chang and Mander Model (Chang and Mander 1994, 3-12)

The Mander and Chang model used by OpenSees modeling command “Concrete07” for confined and unconfined concrete is a variation of the Tsai Model. It is essentially the Popovic Model with parameters (\( r \) and \( n \)) described by Chang and Mander. Chang and Mander found this model to be the most representative of high strength confined concrete (Chang & Mander 1994, 3-12). The model is shown in Equation 2.13. MCAP uses this model.
\[ y = \frac{nx}{1 + \left(n - \frac{r}{r-1}\right) + \frac{x^r}{r-1}} \]  

(2.13)

Where, \( n = \frac{r}{r-1} \); \( y = \frac{f}{f_{cc}} \); \( x = \frac{\varepsilon}{\varepsilon_{cc}} \).

Figure 2-3 shows a graphical representation of Equations 2.6 through 2.13. Equation 2.13 (the Chang and Mander Model) is the same as Equation 2.7 when it is reduced down and the appropriate parameters are applied.

\[ Y = \frac{f}{f_{cc}} \]

\[ X = \frac{\varepsilon}{\varepsilon_{cc}} \]

2.2.2. Confined Concrete

2.2.2.1. Chang and Mander Model

The Chang and Mander Model can describe both confined concrete and unconfined concrete. It is the model used to define concrete in the OpenSees...
model used by MCAP. The Chang and Mander model is a well known concrete model that has general agreement with experimental results for confined concrete (Penelis and Kappos 1997, 194). To obtain stress values for confined concrete, Equation 2.13 is simply multiplied by \( f'_{cc} \). The parameters for the Chang and Mander Model are shown in Equations 2.14 & 2.15.

The strain at peak stress and peak stress are given by Mander, Priestley and Park in Equations 2.14 and 2.15 respectively (Mander, Priestley and Park 1988).

\[ \varepsilon_{cc} = \varepsilon_{co} \left[ 1 + 5 \left( \frac{f'_{cc}}{f'_c} - 1 \right) \right] \]  
(2.14)

\[ f'_{cc} = f'_c \left[ -1.254 + 2.254 \sqrt{\left( 1 + \frac{7.94f'_l}{f'_c} \right) - 2 \frac{f'_l}{f'_c}} \right] \]  
(2.15)

For Rectangular Sections:

\[ f'_l = K_e \rho_x f_yh \]

\[ f'_l = K_e \rho_y f_yh \]

\[ K_e = \frac{1 - \sum_{w}^{n \left( \frac{w}{d_c} \right)^2} \left( \frac{1 - s'}{2b_c} \right) \left( \frac{1 - s'}{2d_c} \right)}{1 - \rho_{cc}} \]

\[ \rho_x = \frac{A_{sx}}{d_c s} \quad \text{and} \quad \rho_y = \frac{A_{sy}}{b_c s} \]

For Circular Sections:

\[ f'_l = \frac{1}{2} K_e \rho_s f_yh \]

\[ K_e = \frac{\left( \frac{1 - s'}{2d_s} \right)^2}{1 - \rho_{cc}} \quad (Hoops) \]
The Mander Model also uses values to define the variable “r” in Equation 2.13. The values are found by using Equation 2.16:

\[ r = \frac{E_c}{E_c - E_{sec}} \]  \hspace{1cm} (2.16)

Where, \[ E_{sec} = \frac{f'_{cc}}{e_{cc}} \]

For rectangular sections, the following relationship in Equation 2.17 for confinement index, K, was developed by Chang and Mander (Chang and Mander 1994, 3-32).

\[ K = \frac{f'_{cc}}{f'_{c}} = 1 + A\bar{x} \left( 0.1 + \frac{0.9}{1 + B\bar{x}} \right) \]  \hspace{1cm} (2.17)

Where, \[ \bar{x} = \frac{f'_{t1} + f'_{t2}}{2f'_{c}} \]

\[ r = \frac{f'_{t1}}{f'_{t2}}, \ f'_{t2} \geq f'_{t1} \]

\[ A = 6.8886 - (0.6069 + 17.275r)e^{-4.989r} \]

\[ B = \left( \frac{4.5}{5} \right) \left( \frac{0.9849 - 0.6306e^{-3.8939r}}{0.1} \right) - 5 \]

When the confining pressure in both directions is the same, a triaxial state of stress is achieved and the confining pressure boils down to the following relationship in equation 2.18 (Chang and Mander 1994, 3-32):
\[ f'_t = \frac{f'_t1 + f'_t2}{2} \]  

\[ (2.18) \]

### 2.2.2.2. Piecewise Models

Park, Priestley and Gill proposed a model that is a piecewise function (Penelis and Kappos 1997, 183-184). Another commonly accepted piecewise model is the Sheikh and Uzumeri model (Penelis and Kappos 1997, 183-184). The models are detailed in Figure 2-4. Both of the models are empirically determined and rely on a confinement index, \( K \), to determine \( f'_{cc} \).

![Figure 2-4: Park, Priestley and Gill Concrete Model (A), Sheikh and Uzumeri Concrete Model (B)](image)

#### 2.2.3. Maximum Concrete Strain

One of the relationships used for evaluation of the \( M-\phi \) curve is the ultimate strain of unconfined concrete and confined concrete. Since the \( M-\phi \) relationship
lends itself to dynamic analysis of RC structures, a limit from seismic engineering literature were chosen.

For unconfined concrete $\varepsilon_{cu}$ was taken as 0.004 in/in. This value represents the spalling strain of the concrete and is generally accepted in sections with confined concrete cores and unconfined concrete cover (Penelis and Kappos 1997, 178). The equations for $\varepsilon_{cu}$ given by Priestley and Paulay are shown in Equation 2.19 for circular and rectangular sections (Priestley and Paulay 1992, 103). The expression in Equation 2.19 is the expression used in MCAP.

$$\varepsilon_{cu} = 0.004 + 1.4\rho_s f_{yu} \varepsilon_u / f'_{cc}$$  \hfill (2.19)

Where, $\rho_s = \rho_x + \rho_y$ for rectangular sections

2.2.4. Reinforcing Steel

There are several different representations of the reinforcing steel stress-strain relationship. The overall shape of the Moment-Curvature relationship is as much determined by the reinforcing steel as the concrete stress-strain model. It is in fact the reinforcing steel with gives tensile capacity to the section and allows for moment capacity above the cracking moment of the section. The yield plateau of the steel and plastic region of reinforcing steel stress-strain behavior defines the Moment-Curvature above the yield curvature. Therefore both the elastic range and plastic range of the reinforcing steel stress-strain relationship are of interest in selecting a steel model for use by the analysis program.
2.2.4.1. Kent and Park Model (Kent and Park, 1973, 98-103)

The model researched by Kent and Park and shown below in Equations 2.20-2.23 (Kent and Park, 1973) uses the Ramberg-Osgood function and experimentally determined parameters to develop a cyclical stress strain curve for reinforcing steel. The equations for the stress-strain curve and experimentally derived parameters are indicated below.

\[ \varepsilon = \frac{f}{E_s} \left[ 1 + \left( \frac{f}{f_{ch}} \right)^{(r-1)} \right] \]  
(2.20)

\[ f_{ch} = f_y \left[ \frac{0.744}{ln(1+1000\varepsilon_{ip})} - \frac{0.071}{1-e^{1000\varepsilon_{ip}}} + 0.241 \right] \]  
(2.21)

When \( n \) is odd:
\[ r = \left[ \frac{4.49}{ln(1 + n)} - \frac{6.03}{(e^n - 1)} + 0.297 \right] \]  
(2.22)

When \( n \) is even:
\[ r = \left[ \frac{2.20}{ln(1 + n)} - \frac{0.469}{(e^n - 1)} + 3.04 \right] \]  
(2.23)

2.2.4.2. Park Model (Park and Paulay 1975, 229)

The Park Model for the plastic region of steel for Monotonic loading of reinforcing steel is given by equation 2.24 (Park and Paulay 1975, 229). This is the model used by CONSEC and BENT for modeling of the reinforcing steel.

\[ f = f_y \left[ \frac{m(\varepsilon_s - \varepsilon_{sh}) + 2}{60(\varepsilon_s - \varepsilon_{sh}) + 2} + \frac{(\varepsilon_s - \varepsilon_{sh})(60 - m)}{2(30r + 1)^2} \right] \]  
(2.24)
Where, \( m = \frac{f_u(30r+1)^2 - 60r - 1}{15r^2} \) and \( r^* = \varepsilon_{su} - \varepsilon_{sh} \)

The elastic region of the relationship (\( \varepsilon \leq \varepsilon_y \)) and yield plateau (\( \varepsilon_y < \varepsilon \leq \varepsilon_{sh} \)) are given by equations 2.25 and 2.26 respectively:

\[
f = \varepsilon E_s \quad \text{(2. 25)}
\]

\[
f = f_y \quad \text{(2. 26)}
\]

**2.2.4.3. Chang and Mander Model (Chang and Mander 1994, 2-1 to 2-2)**

The Chang and Mander stress-strain relationship for the plastic region of reinforcing steel is given in equation 2.27 (Chang and Mander 1994, 2-1 to 2-2). The monotonic portion of the plastic region is shown. The elastic portion and yield plateau portion are given by equations 2.25 and 2.26 respectively. The full text of the report also gives rules for cyclic loading that have the curve match well with the Kent and Park model for cyclic loading (Chang and Mander 1994, 2-1 to 2-58).

\[
f = \left[ f_u + (f_y - f_u) \frac{\varepsilon_u - \varepsilon}{\varepsilon_u - \varepsilon_{sh}} \right]^p \quad \text{(2. 27)}
\]

Where, \( p = \frac{E_{sh} \varepsilon_u - \varepsilon_{sh}}{f_u - f_y} \)

A comparison of the Park monotonic reinforcing steel model, Kent and Park monotonic reinforcing steel model and the Chang and Mander monotonic
reinforcing steel model are shown in Figure 2-5. All three models have a significant difference in the representation of the plastic region of the stress-strain curve.

![Figure 2-5: Reinforcing Steel Model Comparison](image)

The reinforcing and concrete models provide the stress-strain basis for the materials used in the determination of the moment and curvature values in the moment-curvature plot. Different models will have an effect on the shape and maximum values in the M-φ relationship. If more models become available for use by OpenSees, different M-φ relationships can be developed.
2.4. Available Moment-Curvature Computer Programs

The previous sections outlined the procedure used in developing the monotonic moment-curvature envelope and the material models used in developing the M-φ envelope. This section outlines a few M-φ programs for use in obtaining the moment-curvature relationship of reinforced concrete sections. This literature review is not meant to cover all programs used nowadays in academia and/or industry but simply a brief outline of the main differences between MCAP and other software programs.

2.4.1. OpenSees (Mazzoni et al., 2006)

OpenSees is a robust Finite Element Modeling program used for earthquake simulation. It can perform Moment-Curvature analysis and is used for the analysis portion of MCAP. It is free to anyone who registers. OpenSees does not however currently have a Graphical User Interface specifically used for Moment-Curvature analysis. This means that users need to manipulate scripts in Notepad applications and run the script in a DOS prompt.

This process is overly complicated for first time users learning the important relationships between material properties and section analysis. First time users in an undergraduate setting may not be acclimated to programming in scripts. Most beginning users in the undergraduate setting are used to a Graphical User Interface to work with their software applications. Instead of learning the subject matter of stress-strain and M-φ relationships, the typical undergraduate would be
first compelled to learn the programming involved in writing and running a script in OpenSees. This would involve learning the syntax associated with OpenSees and how to analyze the data resulting from the successful operation.

The complications for first time users using OpenSees prompted the development of MCAP. The simplifications by MCAP to the process of entering the material properties and performing the section analysis can increase attention to the subject matter for undergraduates in engineering materials classes.

OpenSees is the backbone of MCAP and performs the numerical analysis for MCAP. OpenSees uses the Tcl (pronounced “tickle”) language. OpenSees consists of 4 command groups that create and analyze the model. These four groups are Modeling, Analysis, Recorder and Miscellaneous commands. Tcl Scripts are created with the four command groups and then are run at the OpenSees command prompt using the source command. The commands used for creation of the moment-curvature analysis in MCAP are explained below.

2.4.1.1. Modeling Commands

OpenSees is a Finite Element modeling program. The modeling commands allow the user to assemble a series of nodes and elements with all appropriate material properties. Details for the modeling commands that are used in MCAP are discussed next.

The “model BasicBuilder” command establishes the number of dimensions and number of degrees of freedom for the model. The “node” command
establishes each node. For MCAP, 2 nodes are created for the section. A zero length element is defined between the nodes to model the section using the “element zerolengthSection” command. The “fix” command establishes the restraints at each node. The material properties are defined by using the “uniaxialMaterial” command. The section is divided up into fibers using the “fiber” command with various methods to define the various fibers. In general, the more fibers, the more precise the results are. The load pattern is defined using the “pattern” command. A constant load pattern is used for MCAP.

2.4.1.2. Analysis Commands

Analysis parameters are established using “constraints” command (constraint handler command), “integrator” command (load control, displacement control, etc. for next time step), “numberer” command which is a DOF Numberer command that maps the Degrees Of Freedom to the equations, “algorithm” command which decides the solution techniques to be used, “system” command which solves the equations within the solution algorithm, “test” command which determines the convergence test and the “analysis” command which determines which type of analysis is used (ie: Static). The analysis is completed using the “analyze” command, which runs the analysis for the time step. One of the useful features of MCAP is analysis parameters that can be switched if the program has convergence issues. The subroutine for switching from one analysis parameter to another was found in the examples manual on the OpenSees website (Mazzoni and Mckenna 2006).
2.4.1.3. Recorder Commands

MCAP uses recorders to find the results of the analysis. A “node” recorder records the nodal moment and curvature at the unrestrained node. An “element” recorder records the stress and strain at a specific location within the section of the zero-length element. The extreme fibers of compression or tension are selected automatically to establish the ultimate limit state of the moment-curvature analysis. Once the ultimate compression or tensile stress is reached for the concrete or steel respectively, the data is later truncated by using MATLAB®.

2.4.1.4. Miscellaneous Commands

Miscellaneous Commands are used to monitor the analysis during computational steps. The analysis subroutine that will switch analysis parameters in MCAP uses a “nodeDisp” command to monitor the current nodal displacement of the unrestrained node. The “nodeDisp” Command also controls the while loop in the previously mentioned subroutine.

2.4.2. SeismoStruct (Seismosoft 2011)

SeismoStruct is a Finite Element Modeling software program that allows the user to perform Time History analysis and Push Over Analysis. It is available for download from Seismosoft (Seismosoft 2011). It has a GUI and is a free program for registered users. SeismoStruct allows users to choose between several material models and element types. However, a more advanced
knowledge of Finite Elements is required to use the program. Also, one must generate a complete, working Finite Element Model to generate a $M-\varphi$ relationship. The Program also restricts the user to several pre-defined shapes and defining the reinforcement in a reinforced concrete member is more tedious than with MCAP.

### 2.4.3. Response 2000 (Bentz 2011)

Response 2000 is cross-section and beam modeling software that models reinforced and prestressed concrete beams and columns. It is available for download for users who register at the Response 2000 website (Bentz 2011). Response 2000 requires more time to master than MCAP and is meant for more advanced non-linear analysis than MCAP. It is a useful tool and relatively easy to learn, but has a few differences with MCAP. Response 2000 does not internally compute the confinement of concrete materials but a user defined model must be defined. Response 2000 also has several other types of pre-defined sections.

### 2.4.4. BENT (Silva et al.)

BENT was created by Silva and Seible from UCSD and is a useful Moment-Curvature analysis program for circular and rectangular sections. The program does not have a Graphical User Interface and uses command prompts. MCAP could serve as a replacement for BENT, as today’s users are accustomed to computer software with GUI’s. A screenshot of BENT is shown in Figure 2-6.
BENT uses the same confinement models for concrete as MCAP but different reinforcing steel models (Silva et al., 1999). The alternative reinforcing steel models in BENT make it difficult to complete a direct comparison between BENT and MCAP.

![Bent Application](image)

**Figure 2-6: Bent Application**

### 2.4.5. CONSEC (Matthews 2009)

CONSEC was developed by Robert Matthews from Structware®. CONSEC is free to download (Matthews 2009). It performs M-\(\phi\) analysis and interaction diagrams. The user interface is a Windows tab system. The sections are developed with confined concrete and the Park model for reinforcing steel. It performs moment curvature analysis on circular and rectangular sections.

### 2.5. MATLAB® Interface Used for MCAP

MATLAB® is a versatile programming language that has many predefined functions. It is useful for numerical calculations, control of lab devices and Graphical User Interfaces through predefined functions and “GUIDE”. MATLAB®
is based on the C++ language, but has been made easier to use through the preprogrammed functions. Users also have the option of defining their own sub-functions. The following sections describe the level of programming used for MCAP.

2.5.1. Variables

There are several variable types used by MATLAB® and most of the variable types were used by MCAP. Global variables are available for use by all of the functions in the workspace (programming environment) one is running. Persistent variables work in the individual script (or a series of MATLAB® statements in a file denoted by a *.m extension or M-file) that one is running. A sub-function may use a script and only the persistent variables called to be transferred outside of the function will be available for use by other scripts. The global variables will be available for all scripts used in the workspace.

2.5.2. Functions

A user-defined function is a script that starts with the function statement and includes input values and output values. A function may have nested sub-functions that may be used by the function. Functions are useful for repetitive calculations that are used by the overall script. The use of functions allows one to reduce the amount of overall coding required by the program and has the added benefit of allowing one to troubleshoot any errors in a more simple fashion.
MATLAB® also includes functions that are available for use and are preprogrammed into the code. Examples of these predefined functions are the sine function with angle in degrees. The format for input and output of these functions is found in the MATLAB® help section.

2.5.3. GUIDE and GUI Interfaces

MATLAB® uses the GUIDE command to create Graphic User Interfaces. The GUIDE command opens a graphic user interface layout editor which allows one to create the graphics for a GUI. The graphics are called figures. The scripts used to run the figures are automatically created with the layout of the graphics. For MCAP programming, the figure and script were created using the graphic user interface layout editor. Once the script was created, different graphic user interface options (for example a button) called subroutines created in other MATLAB® scripts. This process was used to create each screen or figure in MCAP. The buttons can call scripts containing other figures and graphics screens.

The representations of shapes were plotted out into a set of axes embedded into the GUI. Lists boxes and radio buttons were also used as necessary. The drafting portion of the user defined shapes was created under a separate subroutine which had its basis in an example code in the MATLAB® help menu.
2.5.4. Creating and Running OpenSees Files

OpenSees Tcl files are created by writing preformatted text and the variables into a script Tcl file. The files are text files with a *.tcl extension. Two files are actually simultaneously created in MCAP, one is “current.tcl” and the other is a file named as the user has defined. The reason for this is the running of the OpenSees File using MATLAB® commands. The command in MATLAB® that allows the OpenSees file to execute is the bang command, which simply puts an exclamation character (i.e.: “!”) in front of the executable file name and then an ampersand and the command to be run. MATLAB® is not able to write the bang command with a variable command name, so a static file to be rewritten and rerun each time was created (the static file is “current.tcl”).

2.5.5. Data Analysis and Plotting

Some data analysis and plotting is used in MCAP. The results generated by OpenSees are generally not in a readily usable form. MCAP uses MATLAB® to process the data generated in the output files in OpenSees. Generally, MATLAB® reads the text files generated by OpenSees and rewrites the data to a text file formatted by MATLAB® that has headers and can be easily imported into a spreadsheet program. Finally, mathematic functions (ie: square roots, sums, exponential, etc.) for calculating the various input for the materials models used in the OpenSees Tcl scripts are also used.
MCAP also plots the data into the GUI’s created in GUIDE using plot functions in MATLAB®. Various plot functions such as the plot and line function are used to create the graphics depicting the sections in MCAP.
Chapter 3 - MCAP User’s Manual and Development Information

The following section outlines a User’s manual and the development information for MCAP. The interface is intuitive, but some simple installation instructions are followed by a guide on the operation of MCAP. It should be noted that MCAP requires that OpenSees first be installed on your computer. OpenSees is available for download for those who register from the OpenSees Website. MCAP is initiated in MATLAB® through typing “Main” at the command prompt.

3.1. Installation Instructions

Step 1: Copy MCAP Version 1.5 Folder to a convenient directory on your computer.

Step 2: Install OpenSees on your computer (see http://opensees.berkeley.edu/ for instructions).

Step 3: Make sure OpenSees.exe and LibUnits.tcl are in the MCAP Folder.

3.2. Running MCAP

Step 1: Open MATLAB®. Set Screen Resolution to 1200x800.

Step 2: Browse to the MCAP Version 1.5 Directory.

Step 3: Type “Main” at the command prompt.
Step 4: The MCAP main menu appears.
Step 5: Choose section type (ie: Rectangular, Circular or User Defined)

![Choose Section Analysis Type](image)

**Figure 3-3: Choose Section Analysis Type**

### 3.3. Section Analysis Title Screens

The section analysis title screens (see Figure 3-4) are all the same with the exception of the graphic showing the direction of the moment for the section. The analysis process for running an analysis in MCAP is the same for all three modules. First we define the section. This is followed by the material properties of the section elements (ie: steel and concrete). Then the *.TCL file for OpenSees analysis is created. Next the section is analyzed. Then the Moment-Curvature plot is displayed. Finally, we can exit the module if we want using “Back to Title Screen”.

32
3.4. Defining Sections

Once the “Define Section” button (see Figure 3-5) is clicked the section definition screens are shown. Figures 3-6, 3-7 & 3-8 show the section definition screens for each module. Test sections will be used for this chapter. The desired properties are shown in Tables 3-1 & 3-2 for rectangular and circular sections respectively. The User Defined section will be 24” high, 16” wide, a 6” top and bottom flange, a web 6” wide and a 2” taper from the flange to connection with the web. There will be 4 - #6 Bars top and bottom with a #4 transverse reinforcement layer. The cover will be 1”. The confining pressure will be 0.15 ksi. For all sections the axial load will be 10 kips in compression.
Table 3-1: Rectangular Example Section

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>16 in</td>
</tr>
<tr>
<td>Width</td>
<td>12 in</td>
</tr>
<tr>
<td>Cover</td>
<td>1 in</td>
</tr>
<tr>
<td>Number of Top &amp; Bottom Bars</td>
<td>4</td>
</tr>
<tr>
<td>Number of Side Bars</td>
<td>0</td>
</tr>
<tr>
<td>Size Top &amp; Bottom Bars</td>
<td>#4</td>
</tr>
<tr>
<td>Size Transverse Reinforcement</td>
<td>#3</td>
</tr>
<tr>
<td>Spacing Transverse Reinforcement</td>
<td>4 in</td>
</tr>
<tr>
<td>Number of Legs in X Direction</td>
<td>2</td>
</tr>
<tr>
<td>Number of Legs in Y Direction</td>
<td>2</td>
</tr>
<tr>
<td>Rotation Angle</td>
<td>0 Degrees</td>
</tr>
<tr>
<td>( f'c )</td>
<td>5 ksi</td>
</tr>
<tr>
<td>( f_y )</td>
<td>60 ksi</td>
</tr>
<tr>
<td>( f_u )</td>
<td>90 ksi</td>
</tr>
<tr>
<td>( \varepsilon_u )</td>
<td>0.12 in/in</td>
</tr>
<tr>
<td>( \varepsilon_{sh} )</td>
<td>0.00672 in/in</td>
</tr>
<tr>
<td>( E_{sh} )</td>
<td>2200 Ksi</td>
</tr>
<tr>
<td>( E )</td>
<td>29000 Ksi</td>
</tr>
</tbody>
</table>
Table 3-2: Circular Example Section

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter</td>
<td>16 in</td>
</tr>
<tr>
<td>Cover</td>
<td>1 in</td>
</tr>
<tr>
<td>Number Bars</td>
<td>8</td>
</tr>
<tr>
<td>Size Bars</td>
<td>#5</td>
</tr>
<tr>
<td>Size Transverse Reinforcement</td>
<td>#3</td>
</tr>
<tr>
<td>Spiral Spacing</td>
<td>3 in</td>
</tr>
<tr>
<td>Hoops/Spirals</td>
<td>Spiral</td>
</tr>
<tr>
<td>$f_c$</td>
<td>5 ksi</td>
</tr>
<tr>
<td>$f_y$</td>
<td>60 ksi</td>
</tr>
<tr>
<td>$f_u$</td>
<td>90 ksi</td>
</tr>
<tr>
<td>$\varepsilon_{ul}$</td>
<td>0.12 in/in</td>
</tr>
<tr>
<td>$\varepsilon_{sh}$</td>
<td>0.00672 in/in</td>
</tr>
<tr>
<td>$E_{sh}$</td>
<td>2200 Ksi</td>
</tr>
<tr>
<td>$E$</td>
<td>29000 Ksi</td>
</tr>
</tbody>
</table>

Figure 3-5: Define Section Button
Figure 3-6: Rectangular Section Definition

Figure 3-7: Circular Section Definition
For the rectangular and circular sections, define the section dimensions shown in the boxes. The rectangular module allows the user to rotate the section if a rotated analysis is desired. Click on “Draw Section” as shown in Figures 3-10 & 3-11. An image of the section will appear. Press return/exit to go back to the module title screen and move on.

The rotation angle in the rectangular section lets the user rotate the section counter clockwise from zero degrees to 90 degrees. This gives the user a larger range of configurations to choose from and this option is not available through most Moment-Curvature programs. The section is rotated by using a transformation matrix to define each patch area. The test section was rotated zero degrees, 45 degrees and 90 degrees CCW. The results of the rotation of the section are shown in Figure 3-9. As expected, the section capacity is diminished with increased rotation through 90 degrees.
For the circular section, the geometric orientation of the longitudinal bars assumes that the bars start at the three o’clock position and are distributed evenly counterclockwise. Spirals or hoops can be selected for confinement.
Figure 3-10: Draw Rectangular Section

Figure 3-11: Draw Circular Section

The section definition for the user defined section is more complicated. The section must be drawn. The User Defined module allows the user to draw the section down to the sixteenth of an inch. Subroutines (windows_motion_test.m and draw_lines.m) in the MATLAB® examples manual section were modified
significantly to create the drawing application in MCAP (The MathWorks 2007). The user defined section definition title screen has several settings for grid, snap and drawing canvas size. Set these boxes to what you will need for your section. For the example, the settings will be a snap of $\frac{1}{2}''$, a grid of 1” and a drawing canvas size of 24” (which is 24" x 24"). Next press the “Draft Section” Button. A drawing grid or canvas will pop up (see Figure3-12). Left click on all corners of your section. Right Click on the last point to close the section and window. It is important that all subsequent points for the transverse steel and core definition be defined in the same order from start point location to end point location.

![User Defined Section Drawing Canvas](image)

Figure 3-12: User Defined Section Drawing Canvas
Continue through the user defined section drawing buttons to define the transverse reinforcement (with the proper cover requirements), longitudinal reinforcement and to define the core area. Keep in mind that you start with a left click and end with a right click. You can define the longitudinal steel all at one time. Always start and follow the same order for the section extents, transverse reinforcement and core area definition. Try to define the core area as close to the Transverse Reinforcement center line as possible.
Figure 3-14: Define the Transverse Reinforcement

Figure 3-15: Define Longitudinal Reinforcement
After the core area is defined, click on draw section. At this point, the section definition can also be saved (see Figure 3-17). Click on Exit to return to the user defined module title screen.
3.5. Material Properties

The material properties screen is essentially the same for all three modules. The OpenSees models used for MCAP are “concrete07” for concrete and “reinforcingsteel” for the reinforcement. The user defined module requires that the confining pressure, $f'_c$ be defined for the concrete (see Figure3-19). The steel properties menu allows the user to load a lab test data and graphically match the lab data with the steel model used. The lab data file should be a tab delimited text file with the strain values in the first column and stress values in the second column. To load the lab tested values click on the “Load” button, select the file and then click on plot steel model (see Figure3-20). Also, plot the concrete model after the steel model has been defined. As the steel properties have an effect on the confinement in the Chang-Mander Model.

For the circular section, Equations 2.14, 2.15 and 2.16 are used to detail the confined concrete. The Chang and Mander model is used to detail the reinforcing steel. The OpenSees material models are again “concrete07” and “reinforcingsteel”

The confinement of the concrete for rectangular sections is controlled by the number of legs of reinforcing steel in each of the x and y directions of the section. The confinement in each direction may be different, so the relationship described by Chang and Mander for determining the peak confined concrete stress was used. This relationship is given by equation 2.17 (Chang and Mander 1994, 3-32).
The confining pressure for the user defined module is determined by the user and the peak confined stress is determined using the Chang and Mander model in Equation 2.15. Estimating the user defined confining pressures will result in a learning process for the user.

Figure 3-18: Material Properties Screen - Rectangular and Circular Sections

Figure 3-19: Confining Pressure for User Defined Material Properties
3.6. Create *.Tcl File

After definition of the section and definition of the material properties, the *.Tcl file needs to be created. This is the file that OpenSees uses to perform the analysis. Sample files created by MCAP are shown in Appendices A, B & C for the rectangular, circular and user defined test sections respectively. Figure 3-21 shows the screen for the rectangular and circular modules. Type in the file name without extension and enter the axial load in kips. The axial force is assumed to be in compression (ie: a column). Therefore, a positive value of 10 kips indicates 10 kips in compression. Click on “Create .Tcl File” and “Return” to get back to the module title screen.
The user defined module has an extra step. The patches required for the section must be defined. Enter the number of patches in the X and Y direction and then click on “create patches” (see Figure3-22). After the patches are displayed, enter the file name, axial load and then click on “Create .Tcl File”. Click on “Return” to return to the module title screen.
The main challenge in defining a model in OpenSees for the User Defined Module is that the shape of the section was not known. In order to address this unknown, a method for dividing the section into patches for implementation into OpenSees was created. Moreover, the region of the section bounding the confined concrete and the cover concrete was treated separately. To do this task, the main cross section was divided up into points on a grid. If the points were within the confines of the sub-section, they were included in the sub-section and material properties and areas were assigned to the coordinates. This was not a trivial task. It required significant and challenging programming in MATLAB®. After a few cycles of programming, a technique was developed that can be implemented for most any common RC sections. The area, say a rectangle is confined by 4 lines (ie: the transverse reinforcement). As part of the input of the section, the confined area is described by 4 lines interior to the transverse reinforcement lines describing the side of the confined concrete section. Each point that is on the same side of the section boundary line as the line describing the confined area has a value added to its count. By doing this for each line in the rectangle, the points on the interior have a count value of 4 and the points to the exterior have a count value less than 2. Through trial and error and some manipulation of the values added to each point for various cases, it was found that the interior point count greater than 2.5 to 3.33 were indeed interior points and the points less than this value were exterior points. A sample graphic of this technique is shown by the interior (confined) rectangle in Figure 3-23.
The cover concrete patches were easier to determine. If the points were between the exterior line of the section and the transverse reinforcement line, they were kept and material properties were assigned. Careful attention had to be made to the statements describing the points in between the lines. But through a series of 4 switch cases, this was accomplished.

It should be noted in the analysis of the sections that OpenSees does not determine the end point of the M-\(\phi\) analysis. The analysis is carried to 50\(\phi_y\) and then the analysis is truncated by either the extreme compressive strain or extreme tensile strain reaching its ultimate strain. This process occurs for each module of MCAP.
3.7. Analyzing the Section

The section analysis is done using the “Analyze Section” Button. Once the section, material properties and *.Tcl file are defined, Click on “Analyze Section” (see Figure3-24). When the analysis is complete a pop-up box will appear showing that the analysis is complete. The analysis is completed using the “bang” command in MATLAB® to run the “current.tcl” file created by MATLAB® in OpenSees.

![Figure 3-24: Analyze Section](image)

3.8. Moment-Curvature Plot (and Data)

The Moment Curvature Plot is created after the analysis is complete. Simply click on “Current Mom.-Curv. Plot” (see Figure3-25) after the analysis is complete to display the plot. Upon plotting of the results, a file is created which has the M-ϕ data from the analysis as well as the neutral axis position, steel stress and concrete stress. This information can be plotted by the user in a spreadsheet.
program for using the results outside of MCAP. This data file is in a folder where MCAP is located with the same name as the file name previously entered in section 3.6. A screen shot of the sample plot is shown in Figure 3-25. Click on the Back to Title Screen button to switch between modules.

![Moment-Curvature Plot](image)

**Figure 3-25:** Moment-Curvature Plot

### 3.9. Results for Test Sections

The results for the test sections previously described are shown in Figures 3-26 to 3-28.
Figure 3-26: Rectangular Test Section M-C Plot

Figure 3-27: Circular Test Section M-C Plot
3.10. Testing & Verification of MCAP

Testing of MCAP was accomplished through the creation of another program in MATLAB®. The test program consisted of code that used the material stress-strain relationships for confined concrete, unconfined concrete and reinforcing steel. The test program has the peak concrete strengths, section geometry and steel material properties given to it. The test program effectively breaks the section up into small increments vertically and then iterates through a predefined curvature to determine the strain at each increment of the section. The stresses are determined for each increment of the area and the forces in tension and compression computed based on the areas of the increments. Simpson’s rule is used to calculate the concrete forces and moments. The tensile and compressive forces are compared to the tolerance in the program. If the sum of the compressive and tensile forces is above the tolerance, the neutral axis is
moved and the process starts again for the given curvature. Once the tensile and compressive forces are in balance the moment about the plastic center is taken and the moment and curvature are recorded. The same technique was used to calculate the moment-curvature relationship for rectangular and circular sections. As shown in Figures 3-29 and 3-30, the results for both circular and rectangular sections show the OpenSees analysis and test program analysis to be consistent with each other. The tables 3-1 & 3-2 show the section properties of the test program comparison.

To test the user defined sections, a rectangle was created through the user defined module with the confining pressure matching the confining pressure for a rectangular section of the same configuration. The analysis showed that the rectangular section analysis, user-defined section analysis and test program analysis all matched closely which seems to indicate that each analysis is accurate as the same results are obtained using different processes.

MCAP was also compared versus Response 2000. The plots in Figures 3-29 to 3-31 show that the overall capacity of the section was similar for all three sections reviewed. Table 3-3 shows the properties for the “I” section. There were some differences on the non-linear portions of the curve and the extreme curvature values. The differences in the plots were caused by different reinforcing steel material models and the lack of a confined concrete model in Response 2000. The plastic region of the steel models was very different as shown in Figure 2-5. The continued value of stress with increased strains with a confined concrete model would allow the moment values for the Response 2000
curves to be higher at larger curvature values if a confined concrete model was used. With these two adjustments the plots would be very similar.

Table 3-3: “I” Section Properties for Verification

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>24</td>
</tr>
<tr>
<td>Width</td>
<td>24</td>
</tr>
<tr>
<td>Cover</td>
<td>1 in</td>
</tr>
<tr>
<td>Web Thickness</td>
<td>12 in</td>
</tr>
<tr>
<td>Flange Thickness (Top &amp; Bottom)</td>
<td>8 in</td>
</tr>
<tr>
<td>Number of Top &amp; Bottom Bars</td>
<td>6</td>
</tr>
<tr>
<td>Number of Side Bars</td>
<td>0</td>
</tr>
<tr>
<td>Size Top &amp; Bottom Bars</td>
<td>#6</td>
</tr>
<tr>
<td>Size Transverse Reinforcement</td>
<td>#3</td>
</tr>
<tr>
<td>Spacing Transverse Reinforcement</td>
<td>4 in</td>
</tr>
<tr>
<td>Number of Legs in X Direction</td>
<td>2</td>
</tr>
<tr>
<td>Number of Legs in Y Direction</td>
<td>2</td>
</tr>
<tr>
<td>Rotation Angle</td>
<td>0 Degrees</td>
</tr>
<tr>
<td>$f'_{c}$</td>
<td>5 ksi</td>
</tr>
<tr>
<td>$f_y$</td>
<td>60 ksi</td>
</tr>
<tr>
<td>$f_u$</td>
<td>90 ksi</td>
</tr>
<tr>
<td>$\epsilon_u$</td>
<td>0.12 in/in</td>
</tr>
<tr>
<td>$\epsilon_{sh}$</td>
<td>0.00672 in/in</td>
</tr>
<tr>
<td>$E_{sh}$</td>
<td>2200 Ksi</td>
</tr>
<tr>
<td>$E$</td>
<td>29000 Ksi</td>
</tr>
</tbody>
</table>
Figure 3-29: Comparison between Rectangular Section Programs

Figure 3-30: Comparison between Circular Section Programs
Figure 3-31: Comparison between "I" Section Programs
Chapter 4 - MCAP Applications

The primary purpose of MCAP is to serve mainly as a teaching aid. There are many teaching aid applications and a few are listed in the following sections. Material properties, introductions to the design of structures, testing of theoretical predictions and many other concepts can be explained with the help of MCAP. Most of these ideas consist of determining moment capacity of a section. Both elastic and inelastic behavior of a section is determined by MCAP.

4.1. MCAP as a Teaching Aid

The underlying OpenSees Code for MCAP was used in the GWU Fall 2009 CE Materials Laboratory class to predict the failure moment for a simply supported RC beam under load. Undergraduates were asked to predict the load and ultimate moment at which the beam would fail. After the test, when the load and moment were known, undergraduates were then able to alter material model assumptions to see how the material properties affected the strength of the beam. This exercise also provided a glimpse into design of beams to see how spacing and size of reinforcement as well as material properties affect the strength of a RC Section.

Material testing and strength is critical to structural design. MCAP users will be able to learn how reinforcing confinement alters the strength of a concrete section. A module could be added to allow users to create confined concrete specimens, test them and compare the results to the predicted model.
Users can also currently test reinforcing steel in the lab and compare the actual stress-strain relationship to the theoretical relationship given by the Chang and Mander Model by reading the data in MCAP. The theoretical model can be altered to match the actual test results and the material properties of the tested reinforcing steel can be graphically determined. Once the steel is tested in the lab, it can be placed in the model for a beam section analysis. Users can then find and predict the ultimate moment of a simply supported beam. If the beam is then created and tested in the lab, users can test their predictions with actual members in the lab. By performing this process, users can be taught what aspects of the material properties lead to beam failure, be it the tensile strength of the steel or compressive strength of the concrete, etc. Users can also predict and observe the failure mode.

4.2. MCAP as a basis to find Force-Deformation Relationship

If one knows the Moment-Curvature relationship of a section and also knows the structural configuration (ie: cantilever, simply supported beam, etc.) of a member with the section, one can determine the force-deformation relationship of the member using the Moment-Curvature Relationship. There are several methods of doing this including the Euler Method and Runge-Kutta Method. For either case, MCAP can provide the Moment-Curvature relationship that can then be used to determine the force-deformation relationship.
4.3. MCAP as an analysis Tool

MCAP determines the complete moment-curvature relationship for a RC section. The M-\(\phi\) relationship generated by MCAP can be used to determine yield moment and yield curvature (at first yield of the reinforcing steel). For a given curvature ductility, the plastic moment value can determined from the relationship generated by MCAP. The plastic moment can then be used to find the location of the plastic hinge and the maximum loading for a member before it becomes unstable. The yield curvature would also allow one to determine the plastic hinge length for a member.

4.4. Relationship Between Reinforcing Strain and Curvature

Through comparing results in MCAP, it was found that there is a relationship between the curvature of a section and the steel strain in the extreme tensile reinforcing layer. The relationship is independent of the number of bars in the extreme layer and independent of the shape of the section.

The relationship was found by analyzing 18 different sections with different axial loads, geometry and material properties. Table 4-1 shows the different test scenarios examined. The results of the analysis are shown in Figures 4-1 to 4-5. For various sections, compressive axial force, no axial force and tensile axial forces were examined. In each case the relationship of curvature vs. steel strain was linear or very close to linear.
This relationship is useful for programmers of a Moment-Curvature software program. Most programs iterate through many neutral axis positions to determine the neutral axis position of the section where the compressive forces and tensile forces are in balance. If the neutral axis is nearly known, the number of iterations of the neutral axis position is greatly reduced, which results in a program that is accurate and fast.

By performing a first iteration and finding the neutral axis position and the steel strain at that position as well as knowing the two curvatures (current and next iteration), one can predict the position of the neutral axis of the next iteration closely. By using a linear relationship between the curvature and steel strain, computational times are greatly reduced. This idea was used in the test program for MCAP and was highly effective in reducing the computational time. In fact, precision of the test program was increased because of the savings in the iterations required to determine the neutral axis.

### Table 4-1: Strain vs. Curvature Test Sections

<table>
<thead>
<tr>
<th>Test #</th>
<th>Section</th>
<th>Size</th>
<th>Reinforcement</th>
<th>Spacing Trans. Reinf.</th>
<th>Axial Load (Kips)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rect.</td>
<td>16&quot;x12&quot;</td>
<td>4-#4 T&amp;B - #3 Trans, 2x2 legs</td>
<td>4&quot;</td>
<td>-5</td>
</tr>
<tr>
<td>2</td>
<td>Rect.</td>
<td>16&quot;x12&quot;</td>
<td>4-#4 T&amp;B - #3 Trans, 2x2 legs</td>
<td>4&quot;</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Rect.</td>
<td>16&quot;x12&quot;</td>
<td>4-#4 T&amp;B - #3 Trans, 2x2 legs</td>
<td>4&quot;</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Rect.</td>
<td>24&quot;x24&quot;</td>
<td>6-#6 T&amp;B, 6-#6 SB, #4 Trans, 3x4 legs</td>
<td>6&quot;</td>
<td>-25</td>
</tr>
<tr>
<td>5</td>
<td>Rect.</td>
<td>24&quot;x24&quot;</td>
<td>6-#6 T&amp;B, 6-#6 SB, #4 Trans, 3x4 legs</td>
<td>6&quot;</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Rect.</td>
<td>24&quot;x24&quot;</td>
<td>6-#6 T&amp;B, 6-#6 SB, #4 Trans, 3x4 legs</td>
<td>6&quot;</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>Rect.</td>
<td>30&quot;x36&quot;</td>
<td>8-#8 T&amp;B, 10-#8 SB, #5 Trans, 4x5 legs</td>
<td>8&quot;</td>
<td>-50</td>
</tr>
<tr>
<td>8</td>
<td>Rect.</td>
<td>30&quot;x36&quot;</td>
<td>8-#8 T&amp;B, 10-#8 SB, #5 Trans, 4x5 legs</td>
<td>8&quot;</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>Rect.</td>
<td>30&quot;x36&quot;</td>
<td>8-#8 T&amp;B, 10-#8 SB, #5 Trans, 4x5 legs</td>
<td>8&quot;</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>Circ.</td>
<td>16&quot; Dia</td>
<td>16-#4 Long, #3 Sprals</td>
<td>3&quot;</td>
<td>-10</td>
</tr>
<tr>
<td>11</td>
<td>Circ.</td>
<td>16&quot; Dia</td>
<td>16-#4 Long, #3 Sprals</td>
<td>3&quot;</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>Circ.</td>
<td>16&quot; Dia</td>
<td>16-#4 Long, #3 Sprals</td>
<td>3&quot;</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>Circ.</td>
<td>24&quot; Dia</td>
<td>12-#8, #4 Hoops</td>
<td>2&quot;</td>
<td>-25</td>
</tr>
<tr>
<td>14</td>
<td>Circ.</td>
<td>24&quot; Dia</td>
<td>12-#8, #4 Hoops</td>
<td>2&quot;</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>Circ.</td>
<td>24&quot; Dia</td>
<td>12-#8, #4 Hoops</td>
<td>2&quot;</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>W</td>
<td>16&quot;x15&quot;</td>
<td>5-#8 Top, 4-#8 Bottom, #4 Trans</td>
<td>4&quot;</td>
<td>-5</td>
</tr>
<tr>
<td>17</td>
<td>W</td>
<td>16&quot;x15&quot;</td>
<td>5-#8 Top, 4-#8 Bottom, #4 Trans</td>
<td>4&quot;</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>w</td>
<td>16&quot;x15&quot;</td>
<td>5-#8 Top, 4-#8 Bottom, #4 Trans</td>
<td>4&quot;</td>
<td>5</td>
</tr>
</tbody>
</table>
Figure 4-1: Test Sections 1-4
Figure 4-2: Test Sections 5-8
Figure 4-3: Test Sections 9-12
Figure 4-4: Test Sections 13-16
4.5. Using the Reinforcing Strain – Curvature Relationship

As a way to verify how the reinforcing strain and curvature relationship might be used another M-ϕ program was created in MATLAB® as a learning exercise. The sample code for the rectangular section is shown in Appendix D. The program consisted of code that used the same material stress-strain relationships for confined concrete, unconfined concrete and reinforcing steel as MCAP. The program has the peak concrete strengths, section geometry and steel material properties given to it. Following the procedures outlined in Section 2, the program effectively breaks the section up into small increments (see Figure 2-2) vertically and then iterates through a predefined curvature to
determine the strain at each increment of the section. The initial placement of the neutral axis is determined by using the linear relationship between the tensile strain in the extreme reinforcing layer and the curvature from the previous iteration. The neutral axis is then fine tuned by moving it with an algorithm that assumes a linear change in neutral axis position based on the amount of change in tensile and compressive forces in the last iteration. The stresses are determined for each increment of the area and the forces in tension and compression computed based on the areas of the increments. Simpson’s rule is used to calculate the concrete forces and moments. The tensile and compressive forces are compared to the tolerance in the program. If the sum of the compressive and tensile forces is above the tolerance, the neutral axis is moved and the process starts again for the given curvature. Once the tensile and compressive forces are in balance the moment about the plastic center is taken and the moment and curvature are recorded. The same technique was used to calculate the moment-curvature relationship for rectangular and circular sections. As shown in Figures 4-6 and 4-7, the results for both circular and rectangular sections show the OpenSees analysis and “practice” program analysis to be consistent with each other. Tables 3-1 & 3-2 show the section properties of the practice program comparison.

The process of finding the neutral axis position using the relationship between tensile strain in the extreme layer and curvature assures the smallest amount of iterations to find the right neutral axis position. In writing this program,
the time for the analysis was significantly reduced by incorporating a good initial estimate for the neutral axis position.

Figure 4-6: Rectangular Section Comparison
Figure 4-7: Circular Section Comparison
Chapter 5 - Conclusions in the Development of MCAP

MCAP is an effective program for generating M-φ relationships for RC Sections. The program provides a useful interface for an OpenSees analysis backbone and is user friendly. As a teaching aid, MCAP is accurate, uses conventional material models and MCAP provides an advantage on modeling user defined sections and rotated sections that were not possible with other Moment-Curvature programs. There are a range of instructional applications including learning about materials and structural analysis. The program is versatile in this respect since it can be used equally well by undergraduates and graduates. The development of MCAP has also provided insight as to how to increase computational accuracy through the relationship between curvature and the strain in the extreme reinforcing steel layer and decrease computational times for all programs that are developed for M-φ analysis.

Future developments for MCAP include a pile analysis module and expanding the program to perform a pushover analyses for reinforced concrete members. The implementation of a FRP confined section module is also being developed with an immediate application to the retrofit of circular and rectangular concrete sections. Further research on the relationship between the curvature and the strains in the extreme steel fibers is a future development that MCAP can help accomplish. This relationship is valuable in Moment-Curvature analysis. If one could define a relationship between curvature and the strain in the extreme steel fiber for any section, the neutral axis could be easily defined for the analysis. This would then result in Moment-Curvature analysis that could be performed by
hand calculations instead of an iterative process adapted to computer analysis. A simplified hand analysis technique would be a great tool for engineers needing the Moment-Curvature relationship.
References


Appendix A – Rectangular Section Example *.tcl File

# Units: Kips, Inches, Seconds
#-------------------------------------------------------------

wipe;
file mkdir TEST_Rect_1;
model BasicBuilder -ndm 3 -ndf 6;
source LibUnits.tcl
# define GEOMETRY
#-------------------------------------------------------------

set LCol [expr 120*$in];
set Weight [expr -10*$kip];

# calculated parameters
set PCol $Weight;             # nodal dead-load weight per column

# nodal coordinates:
node 1 0 0 0 0 0 0;                     # Node#, X, Y
node 3 0 0 0 0 0 0;

set SecTag 1;

# Single point constraints -- Boundary Conditions
fix 1 1 1 1 1 1 1;    # node DX DY RZ
fix 3 0 1 1 1 0 1;

# MATERIAL parameters
#-------------------------------------------------------------

#------------Concrete--------------------
set IDconcCore 1;
set IDconcCover 2;
set fc   [expr -5*$ksi];
set Ec  [expr 57*$ksi*pow((-$fc/$psi),0.5)];

# confined concrete
set Kfc   1.121;             # ratio of confined to unconfined
concrete strength
set fc1C   [expr $Kfc*$fc];  # CONFINED concrete (mander model),
maximum stress
set eps1C  -0.0032099;      # strain at maximum stress
set xncon   7.0431;  # Non-dimensional term that defines the
strain at which the straight line descent begins in compression
set rrcon   1.7644;   # Parameter that controls the nonlinear
descending branch

# unconfined concrete
set fc1U   $fc;             # UNCONFINED concrete (todeschini
parabolic model), maximum stress
set eps1U -0.002;    # strain at maximum strength of unconfined
concrete
set xnunc 2.3;  # Non-dimensional term that defines the
strain at which the straight line descent begins in compression
set rrunc 4.7667;  # Parameter that controls the
nonlinear descending branch

# tensile-strength properties
set ftC 0.60458;  # tensile strength +tension
set ftU 0.60458;  # tensile strength +tension
set et 0.00015;  # tensile strain at maximum tension
set xp 2;  # Non-dimensional term that defines the
strain at which the straight line descent begins in tension

# Steel Reinforcement------------------
set IDreinf 3;  # material ID tag -- reinforcement

# Steel properties
set Fy [expr 60*$ksi];  # STEEL yield stress
set Fu [expr 90*$ksi];  # Ultimate STEEL yield stress
set esh 0.00672;    # strain at hardening
set eult 0.12;    # strain at ultimate
set Es 29000;  # modulus of steel
set Esh 2200; # modulus of steel after hardening

# build uniaxial material
uniaxialMaterial ReinforcingSteel $IDreinf $Fy $Fu $Es $Esh $esh $eult;
uniaxialMaterial Concrete07 $IDconcCore $fc1C $eps1C $Ec $ftC $et $xp
$xncon $rrcon;  # build core concrete (confined)
uniaxialMaterial Concrete07 $IDconcCover $fc1U $eps1U $Ec $ftC $et $xp
$xnunc $rrunc;  # build cover concrete (unconfined)

# FIBER SECTION properties
#---------------------------------------

# RC Rectangular Section
# Column section geometry

set Depth [expr 16*$in];    # Column Depth
set Width [expr 12*$in];    # Column Width
set cover [expr 1*$in];    # Column cover to reinforcing steel
set numBarsTop 4;     # number of longitudinal-reinforcement TOP
set numBarsBot 4; # number of longitudinal-reinforcement BOTTOM
set numBarsIntTot 0;  # Number of Side Bars (Total)
set barAreaTop [expr 0.19635*$in2]; # area of longitudinal-
reinforcement bars -- top
set barAreaBot [expr 0.19635*$in2]; # area of longitudinal-
reinforcement bars -- bot
set barAreaInt [expr 0.7854*$in2]; # area of longitudinal-
reinforcement bars -- intermediate skin reinf
set TransDia [expr 0.375*$in];    # Dia of Trans-reinforcement bars
set LongRad [expr 0.25*$in];    # Dia of Trans-reinforcement bars
set SBSpacing [expr 12.75*$in];   # Dia of Trans-reinforcement bars
set SideRad [expr 0.5*$in];  # Dia of Trans-reinforcement bars
set coverZ [expr $Depth/2.0]; # The distance from the section z-axis to
the edge of the cover concrete -- outer edge of cover concrete
set coverY [expr $Width/2.0]; # The distance from the section x-axis to
the edge of the cover concrete -- outer edge of cover concrete
set coreZ [expr $coverZ-$cover];    # The distance from the section z-axis to the edge of the core concrete/inner edge of cover concrete
set coreY [expr $coverY-$cover];    # The distance from the section x-axis to the edge of the core concrete/inner edge of cover concrete
set coreY2 [expr $coverY-$cover-$TransDia/2-$LongRad];
set coreZ2 [expr $coverZ-$cover-$TransDia/2-$LongRad];
set coreZ3 [expr $coverZ-$cover-$TransDia/2-$SBspacing];
set coreY3 [expr $coverY-$cover-$TransDia/2-$SideRad];
set nfY 64;   # number of fibers for concrete in Y-direction
set nfZ 64;    # number of fibers for concrete in Z-direction
set numBarsInt [expr $numBarsIntTot/2];   # number of intermediate bars per side

section Fiber $SecTag     {;  
 patch quadr $IDconcCore $nfY $nfZ -5 -7 5 7 -5 7;  # Define the core patch
 patch quadr $IDconcCover 1 $nfZ -6 -8 -5 -8 -5 8 -6 8; # Define the four cover patches
 patch quadr $IDconcCover 1 $nfZ 5 -8 6 -8 6 8 5 8;
 patch quadr $IDconcCover $nfY 1 -5 7 5 7 5 8 -5 8;
 layer straight $IDreinf $numBarsTop $barAreaTop -4.5625 -6.5625 4.5625 6.5625; # top layer reinforcement
 layer straight $IDreinf $numBarsBot $barAreaBot -4.5625 6.5625 4.5625 6.5625; # bottom layer reinforcement
}; # end of fibersection definition

# Define element
# tag ndI ndJ secTag
element zeroLengthSection 1 1 3 $SecTag

# Create recorder
recorder Node -file TEST_Rect_1/TEST_Rect_1.out -time -node 3 -dof 5 disp; # output moment (col 1) & curvature (col 2)
recorder Element -file TEST_Rect_1/TEST_Rect_12.out -time -ele 1 section fiber $coreY2 $coreZ2 $IDreinf stressStrain;

recorder Element -file TEST_Rect_1/TEST_Rect_13.out -time -ele 1 section fiber [expr $coreY-$1.5*2*$coreY/64] [expr $coreZ+$0.5*2*$coreZ/64] $IDconcCore stressStrain;

# Define constant axial load
pattern Plain 1 "Constant" {
 load 3 $PCol 0.0 0.0 0.0 0.0 0.0
}

# Define analysis parameters
integrator LoadControl 0 1 0 0
system SparseGeneral -piv; # Overkill, but may need the pivoting!
test NormUnbalance 1.0e-9 10
numberer RCM
constraints Transformation
algorithm NewtonLineSearch
analysis Static

# Do the section analysis
analyze 1
# Define reference moment pattern
pattern Plain 2 "Linear" {
  load 3 0.0 0.0 0.0 0.0 1.0 0.0
}
set epsY [expr $Fy/$Es]
set phiYest [expr ($epsY*2.10)/($Depth)]; # estimate yield curvature
set maxK [expr 50*$phiYest]; # maximum curvature reached during analysis

# Compute curvature increment
set numIncr 1000
set dK [expr $maxK/$numIncr]

# Use displacement control at node 3 for section analysis
integrator DisplacementControl 3 5 $dK 1 $dK $dK
# Do the section analysis
set ok [analyze $numIncr]

# ----------------------------------if convergence failure-----------------------------
set IDctrlNode 3
set IDctrlDOF 5
set Dmax $maxK
set Dincr $dK
set TolStatic 1.e-9;
set testTypeStatic EnergyIncr
set maxNumIterStatic 6
set algorithmTypeStatic Newton
if {$ok != 0} {
  # if analysis fails, we try some other stuff, performance is slower inside this loop
  set Dstep 0.0;
  set ok 0
  while {$Dstep <= 1.0 && $ok == 0} {
    set controlDisp [nodeDisp $IDctrlNode $IDctrlDOF ]
    set Dstep [expr $controlDisp/$Dmax]
    set ok [analyze 1];                  # this will return zero if no convergence problems were encountered
    if {$ok != 0} {    # reduce step size if still fails to converge
      # if analysis fails, we try some other stuff
      # performance is slower inside this loop
      global maxNumIterStatic;     # max no. of iterations performed before "failure to converge" is returned
      puts "Trying Newton with Initial Tangent .."
test NormDispIncr $TolStatic 2000 0
algorithm Newton -initial
set ok [analyze 1]
test $testTypeStatic $TolStatic $maxNumIterStatic 0
algorithm $algorithmTypeStatic
} if {$ok != 0} {
puts "Trying Broyden .."
algorithm Broyden 8
set ok [analyze 1]
algorithm $algorithmTypeStatic
} if {$ok != 0} {
puts "Trying NewtonWithLineSearch"
algorithm NewtonLineSearch 0.8
set ok [analyze 1]
algorithm $algorithmTypeStatic
} if {$ok != 0} {
# stop if still fails to converge
puts [format $fmt1 "PROBLEM" $IDctrlNode $IDctrlDOF [nodeDisp $IDctrlNode $IDctrlDOF] $LunitTXT]
return -1
}; # end if
} # end for
integrator DisplacementControl $IDctrlNode $IDctrlDOF $Dincr; # bring back to original increment
}; # end if
}; # end while loop
}; # end if ok !0
# --------------------------------------------------------------------
global LunitTXT; # load time-unit text
if { ![info exists LunitTXT] != 1} {set LunitTXT "Length"}; # set blank if it has not been defined previously.
set fmt1 "%s Pushover analysis: CtrlNode %.3i, dof %.1i, Curv=%.4f /%s"; # format for screen/file output of DONE/PROBLEM analysis
if {$ok != 0} {
puts [format $fmt1 "PROBLEM" $IDctrlNode $IDctrlDOF [nodeDisp $IDctrlNode $IDctrlDOF] $LunitTXT]
} else {
puts [format $fmt1 "DONE" $IDctrlNode $IDctrlDOF [nodeDisp $IDctrlNode $IDctrlDOF] $LunitTXT]
}
Appendix B – Circular Section Example *.tcl File

#Units: Kips, Inches, Seconds
#------------------------------------------------------------------
wipe;
file mkdir Test_Circ_1;
model BasicBuilder -ndm 3 -ndf 6;
source LibUnits.tcl
# define GEOMETRY
#------------------------------------------------------------------
set LCol [expr 120*$in];
set Weight [expr -10*$kip];

# calculated parameters
set PCol $Weight;             # nodal dead-load weight per column

# nodal coordinates:
node 1 0 0 0 0 0 0;                     # Node#, X, Y
node 3 0 0 0 0 0 0;

set SecTag 1;

# Single point constraints -- Boundary Conditions
fix 1 1 1 1 1 1 1;    # node DX DY RZ
fix 3 0 1 1 1 0 1;

# MATERIAL parameters
#----------------Concrete-----------------
set IDconcCore 1;
set IDconcCover 2;

# nominal concrete compressive strength
set fc   [expr -5*$ksi];
set Ec  [expr 57*$ksi*pow((-$fc/$psi),0.5)];

# confined concrete
set Kfc   1.3627;   # ratio of confined to unconfined concrete strength
set fc1C   [expr $Kfc*$fc];  # CONFINED concrete (mander model), maximum stress
set eps1C  -0.0056266; # strain at maximum stress
set xncon   3.5529;  # Non-dimensional term that defines the strain at which the straight line descent begins in compression
set rrcon   1.4295;   # Parameter that controls the nonlinear descending branch

# unconfined concrete
set fc1U   $fc;  # UNCONFINED concrete (todeschini parabolic model), maximum stress
set eps1U  -0.002; # strain at maximum strength of unconfined concrete
set xnunc 2.3;  # Non-dimensional term that defines the
strain at which the straight line descent begins in compression
set rrunc 4.7667;  # Parameter that controls the
nonlinear descending branch

# tensile-strength properties
set ftC 0.60458;  # tensile strength +tension
set ftU 0.60458;  # tensile strength +tension
set et 0.00015;  # tensile strain at maximum tension
stress
set xp 2;  # Non-dimensional term that defines the
strain at which the straight line descent begins in tension

#----------------Steel Reinforcement------------------
set IDreinf 3;  # material ID tag -- reinforcement

# Steel properties
set Fy [expr 60*$ksi];  # STEEL yield stress
set Fu [expr 90*$ksi];  # Ultimate STEEL yield stress

set esh 0.00672;    # strain at hardening
set eult 0.12;    # strain at ultimate
set Es 29000;  # modulus of steel
set Esh 2200; # modulus of steel after hardening

# build uniaxial material
uniaxialMaterial ReinforcingSteel $IDreinf $Fy $Fu $Es $Esh $esh $eult;
uniaxialMaterial Concrete07 $IDconcCore $fc1C $eps1C $Ec $ftC $et $xp
$xncon $rrcon;  # build core concrete (confined)
uniaxialMaterial Concrete07 $IDconcCover $fc1U $eps1U $Ec $ftC $et $xp
$xnunc $rrunc;  # build cover concrete (unconfined)

# FIBER SECTION properties
#-----------------------------------------

#RC Circular Section
# Column section geometry

set cover [expr 1*$in];  # Column cover to reinforcing steel
set numBars 8;  # number of longitudinal-reinforcement
set barArea [expr 0.3068*$in2];  # area of longitudinal-
reinforcement bars -- top
set TransDia [expr 0.375*$in];  # Dia of Trans-reinforcement bars
set SteelRad [expr 6.5*$in];  # Rad of Long-reinforcement bars Layer
set CoreRad [expr 7*$in];  # Rad of Conc. Core
set OuterRad [expr 8*$in];  # Rad of Column
set OuterDia [expr 16*$in];  # DIA of Column
set nfr 48;  # number of fibers for concrete in radius
set nfc 32;  # number of fibers for concrete in
 circulated direction
section Fiber $SecTag {;  # Define the fiber section
patch circ $IDconcCore $nfc $nfr 0 0 0 $CoreRad 0 360;  # Define
the core patch
patch circ $IDconcCover $nfc $nfr 0 0 $CoreRad $OuterRad 180 472.5;
# Define the cover patch
layer circ $IDreinf $numBars $barArea 0 0 $SteelRad;  # Steel Layer
};  # end of fibersection definition
# Define element
tag ndI ndJ secTag
element zeroLengthSection 1 1 3 $SecTag

# Create recorder
recorder Node -file Test_Circ_1/Test_Circ_1.out -time -node 3 -dof 5
disp; # output moment (col 1) & curvature (col 2)

recorder Element -file Test_Circ_1/Test_Circ_12.out -time -ele 1
section fiber 6.5 0 $IDreinf stressStrain;

recorder Element -file Test_Circ_1/Test_Circ_13.out -time -ele 1
section fiber -0.67897 -6.8937 $IDconcCore stressStrain

# Define constant axial load
pattern Plain 1 "Constant" {
load 3 $PCol 0.0 0.0 0.0 0.0 0.0
}

# Define analysis parameters
integrator LoadControl 0 1 0 0
system SparseGeneral -piv; # Overkill, but may need the pivoting!
test NormUnbalance 1.0e-9 10
numberer RCM
constraints Transformation
algorithm NewtonLineSearch
analysis Static

# Do the section analysis
analyze 1
# Define reference moment pattern
pattern Plain 2 "Linear" {
load 3 0.0 0.0 0.0 0.0 1.0 0.0
}
set epsY [expr $Fy/$Es]
set phiYest [expr ($epsY*2.10)/($OuterDia)]; # estimate yield curvature
set maxK [expr 100*$phiYest]; # maximum curvature reached during analysis

# Compute curvature increment
set numIncr 1000
set dK [expr $maxK/$numIncr]

# Use displacement control at node 3 for section analysis
integrator DisplacementControl 3 5 $dK 1 $dK $dK
# Do the section analysis
set ok [analyze $numIncr]

# -------------------------------if convergence failure------------------------
set IDctrlNode 3
set IDctrlDOF 5
set Dmax $maxK
set Dincr $dK
set TolStatic 1.e-9;
set testTypeStatic EnergyIncr
set maxNumIterStatic 6
set algorithmTypeStatic Newton
if {$ok != 0} {
    # if analysis fails, we try some other stuff, performance is slower
    inside this loop
    set Dstep 0.0;
    set ok 0
    while {$Dstep <= 1.0 && $ok == 0} {
        set controlDisp [nodeDisp $IDctrlNode $IDctrlDOF ]
        set Dstep [expr $controlDisp/$Dmax]  # this will return zero if no
        convergence problems were encountered
        if {$ok != 0} {
            # reduce step size if still fails to converge
            set NK 4;  # reduce step size
            set DincrReduced [expr $Dincr/$NK];
            integrator DisplacementControl $IDctrlNode $IDctrlDOF $DincrReduced
            for {set ik 1} {$ik <=$NK} {incr ik 1} {
                set ok [analyze 1];
                if {$ok != 0} {
                    # if analysis fails, we try some other stuff
                    # performance is slower inside this loop
                    global maxNumIterStatic;  # max no. of iterations performed
                    before "failure to converge" is returned
                    puts "Trying Newton with Initial Tangent .."
                    test NormDispIncr $TolStatic $maxNumIterStatic 0
                    algorithm Newton -initial
                    set ok [analyze 1]
                    test $testTypeStatic $TolStatic $maxNumIterStatic 0
                    algorithm $algorithmTypeStatic
                } 
            }
            if {$ok != 0} {
                puts "Trying Broyden .."
                algorithm Broyden 8
                set ok [analyze 1]
                algorithm $algorithmTypeStatic
            } 
            if {$ok != 0} {
                puts "Trying NewtonWithLineSearch"
                algorithm NewtonLineSearch 0.8
                set ok [analyze 1]
                algorithm $algorithmTypeStatic
            } 
            if {$ok != 0} {
                # stop if still fails to converge
                puts [format $fmt1 "PROBLEM" $IDctrlNode $IDctrlDOF [nodeDisp
                $IDctrlNode $IDctrlDOF] $LunitTXT]
                return -1
            };
        };
    };
};
# end if
};
# end while loop
};
# end if ok !0
# --------------------------------------------------------------------
global LunitTXT;  # load time-unit text
if { [info exists LunitTXT] != 1} {set LunitTXT "Length"};  # set blank if it has not been defined previously.
set fmt1 "%s Pushover analysis: CtrlNode %.3i, dof %.1i, Curv=%.4f /
%s"; # format for screen/file output of DONE/PROBLEM analysis
if {$ok != 0 } {
    puts [format $fmt1 "PROBLEM" $IDctrlNode $IDctrlDOF [nodeDisp
$IDctrlNode $IDctrlDOF] $LunitTXT]
} else {
    puts [format $fmt1 "DONE"  $IDctrlNode $IDctrlDOF [nodeDisp $IDctrlNode
$IDctrlDOF] $LunitTXT]
}
# Units: Kips, Inches, Seconds

wipe;
file mkdir UD_Example;
model BasicBuilder -ndm 3 -ndf 6;
source LibUnits.tcl
# define geometry

## Calculated parameters
set LCol [expr 120*$in];
set Weight [expr -10*$kip];

## Nodal coordinates:
node 1 0 0 0 0 0 0;                # Node#, X, Y
node 3 0 0 0 0 0 0;

set SecTag 1;

# Single point constraints -- Boundary Conditions
fix 1 1 1 1 1 1 1;        # node DX DY RZ
fix 3 0 1 1 1 0 1;

# MATERIAL parameters
#------------Concrete-----------------
set IDconcCore 1;
set IDconcCover 2;
set fc   [expr -5*$ksi];
set Ec  [expr 57*$ksi*pow((-fc/$psi),0.5)];
set Kfc   1.1941; # ratio of confined to unconfined concrete strength
set fc1C   [expr $Kfc*$fc]; # CONFINED concrete (mander model), maximum stress
set eps1C  -0.0039413; # strain at maximum stress
set xncon  7.8225;  # Non-dimensional term that defines the strain at which the straight line descent begins in compression
set rrcon  1.6022;  # Parameter that controls the nonlinear descending branch

# unconfined concrete
set fc1U   $fc;    # UNCONFINED concrete (todeschini parabolic model), maximum stress
set eps1U -0.002; # strain at maximum strength of unconfined concrete
set xnunc 2.3;     # Non-dimensional term that defines the strain at which the straight line descent begins in compression
set rrunc 4.7667;  # Parameter that controls the nonlinear descending branch

# tensile-strength properties
set ftC 0.60458;  # tensile strength +tension
set ftU 0.60458;  # tensile strength +tension
set et 0.00015;  # tensile strain at maximum tension stress
set xp 2;        # Non-dimensional term that defines the strain at which the straight line descent begins in tension

#----------------Steel Reinforcement------------------
set IDreinf 3;   # material ID tag -- reinforcement

# Steel properties
set Fy [expr 60*$ksi];  # STEEL yield stress
set Fu [expr 90*$ksi];  # Ultimate STEEL yield stress
set esh 0.00672;    # strain at hardening
set eult 0.12;    # strain at ultimate
set Es 29000;  # modulus of steel
set Esh 2200; # modulus of steel after hardening

# build uniaxial material
uniaxialMaterial ReinforcingSteel $IDreinf $Fy $Fu $Es $Esh $esh $eult;
uniaxialMaterial Concrete07 $IDconcCore $fc1C $eps1C $Ec $ftC $et $xp $xncon $rrcon; # build core concrete (confined)
uniaxialMaterial Concrete07 $IDconcCover $fc1U $eps1U $Ec $ftC $et $xp $xnunc $rrunc; # build cover concrete (unconfined)

# FIBER SECTION properties
#--------------------------------------------------------------

#RC Rectangular Section
# Column section geometry

section Fiber $SecTag   ; # Define the fiber section
fiber 7.7867 11.68 0.1536 $IDconcCover
fiber 7.7867 11.36 0.068267 $IDconcCover
fiber 7.7867 11.04 0.068267 $IDconcCover
fiber 7.7867 10.72 0.068267 $IDconcCover
fiber 7.7867 10.4 0.068267 $IDconcCover
fiber 7.7867 10.08 0.068267 $IDconcCover
fiber 7.7867 9.76 0.068267 $IDconcCover
fiber 7.7867 9.44 0.068267 $IDconcCover
fiber 7.7867 9.12 0.068267 $IDconcCover
fiber 7.7867 8.8 0.068267 $IDconcCover
fiber 7.7867 8.48 0.068267 $IDconcCover
fiber 7.7867 8.16 0.068267 $IDconcCover
fiber 7.7867 7.84 0.068267 $IDconcCover
fiber 7.7867 7.52 0.068267 $IDconcCover
fiber 7.7867 7.2 0.068267 $IDconcCover
fiber 7.7867 6.88 0.068267 $IDconcCover
fiber 7.7867 6.56 0.068267 $IDconcCover
<table>
<thead>
<tr>
<th>fiber</th>
<th>6.24</th>
<th>0.068267</th>
<th>IDconcCover</th>
</tr>
</thead>
<tbody>
<tr>
<td>fiber</td>
<td>5.92</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-5.92</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-6.24</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-6.56</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-6.88</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-7.2</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-7.52</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-7.84</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-8.16</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-8.48</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-8.8</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-9.12</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-9.44</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-9.76</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-10.08</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-10.4</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-10.72</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-11.04</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fiber</th>
<th>11.68</th>
<th>0.1536</th>
<th>IDconcCover</th>
</tr>
</thead>
<tbody>
<tr>
<td>fiber</td>
<td>11.68</td>
<td>0.1024</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>11.36</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>11.04</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>10.72</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>10.4</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>10.08</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>9.76</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>9.44</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>9.12</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>8.8</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>8.48</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>8.16</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>7.84</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>7.52</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>7.2</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>6.88</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>6.56</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>6.24</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>5.92</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-5.92</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-6.24</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-6.56</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-6.88</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-7.2</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-7.52</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-7.84</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-8.16</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-8.48</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-8.8</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-9.12</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-9.44</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-9.76</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-10.08</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-10.4</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-10.72</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>fiber</td>
<td>-11.04</td>
<td>0.068267</td>
<td>IDconcCover</td>
</tr>
<tr>
<td>Fiber</td>
<td>X Position</td>
<td>Y Position</td>
<td>IDconcCore</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>10.7067</td>
<td>0.1232</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>10.4133</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>10.12</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>9.8267</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>9.5333</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>9.24</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>8.9467</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>8.6533</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>8.36</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.8133</td>
<td>8.0667</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>10.7067</td>
<td>0.082133</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>10.4133</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>10.12</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>9.8267</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>9.5333</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>9.24</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>8.9467</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>8.6533</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>8.36</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>8.0667</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>7.7733</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>7.48</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>7.1867</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>6.8933</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>6.5333</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>6.24</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>5.9467</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>5.6533</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>5.36</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>5.0667</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>4.7733</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>4.48</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>4.1867</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>3.9067</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>3.6133</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>3.3267</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>3.0393</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>2.7526</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>2.4659</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>2.1792</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>1.8925</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>1.6058</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>1.3190</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>1.0323</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>0.7456</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>0.4589</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>0.1722</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-0.2041</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-0.4978</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-0.7115</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-0.9252</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-1.1389</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-1.3525</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-1.5662</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-1.7799</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-2.0035</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-2.2172</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-2.4308</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-2.6445</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-2.8581</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-3.0718</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-3.2855</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-3.5091</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-3.7227</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-3.9364</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-4.1501</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-4.3637</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-4.5774</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-4.7910</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-5.0046</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-5.2183</td>
<td>0.054756</td>
</tr>
<tr>
<td></td>
<td>6.6267</td>
<td>-5.4319</td>
<td>0.054756</td>
</tr>
</tbody>
</table>
------ Pages of the Fiber Definition have been omitted for brevity------

fiber 6 10.5 0.44179 $IDreinf
fiber 2 10.5 0.44179 $IDreinf
fiber -2 10.5 0.44179 $IDreinf
fiber -6 10.5 0.44179 $IDreinf
fiber -6 -10.5 0.44179 $IDreinf
fiber -2 -10.5 0.44179 $IDreinf
fiber 2 -10.5 0.44179 $IDreinf
fiber 6 -10.5 0.44179 $IDreinf
}; # end of fibersection definition

# Define element
# tag ndI ndJ secTag
element zeroLengthSection 1 1 3 $SecTag

# Create recorder
recorder Node -file UD_Example/UD_Example.out -time -node 3 -dof 5 disp; # output moment (col 1) & curvature (col 2)

recorder Element -file UD_Example/UD_Example2.out -time -ele 1 section fiber 6 10.5 $IDreinf stressStrain;

recorder Element -file UD_Example/UD_Example3.out -time -ele 1 section fiber 6.8133 -10.7067 $IDconcCore stressStrain

# Define constant axial load
pattern Plain 1 "Constant" {
load 3 $PCol 0.0 0.0 0.0 0.0 0.0
}

# Define analysis parameters
integrator LoadControl 0 1 0 0
system SparseGeneral -piv; # Overkill, but may need the pivoting!
test NormUnbalance 1.0e-9 10
numberer RCM
constraints Transformation
algorithm NewtonLineSearch
analysis Static

# Do the section analysis
analyze 1
# Define reference moment pattern
pattern Plain 2 "Linear" {
load 3 0.0 0.0 0.0 0.0 1.0 0.0
}
set epsY [expr $Fy/$Es]
set phiYest [expr ($epsY*2.10)/($Depth)]; # estimate yield curvature
set maxK [expr 50*$phiYest]; # maximum curvature reached during analysis

# Compute curvature increment
set numIncr 1000
set dK [expr $maxK/$numIncr]
# Use displacement control at node 3 for section analysis
integrator DisplacementControl 3 5 $dK 1 $dK $dK

# Do the section analysis
set ok [analyze $numIncr]

# -------------------------------if convergence failure---------------------
set IDctrlNode 3
set IDctrlDOF 5
set Dmax $maxK
set Dincr $dK
set TolStatic 1.e-9;
set testTypeStatic EnergyIncr
set maxNumIterStatic 6
set algorithmTypeStatic Newton
if {$ok != 0} {
    # if analysis fails, we try some other stuff, performance is slower
    inside this loop
    set Dstep 0.0;
    set ok 0
    while {$Dstep <= 1.0 && $ok == 0} {
        set controlDisp [nodeDisp $IDctrlNode $IDctrlDOF ]
        set Dstep [expr $controlDisp/$Dmax]
        set ok [analyze 1];                  # this will return zero if no
        # this will return zero if no convergence problems were encountered
        if {$ok != 0} {
            # reduce step size if still fails to converge
            set Nk 4;   # reduce step size
            set DincrReduced [expr $Dincr/$Nk];
            integrator DisplacementControl $IDctrlNode $IDctrlDOF $DincrReduced
            for {set ik 1} {$ik <=$Nk} {incr ik 1} {
                set ok [analyze 1];
                if {$ok != 0} {
                    # if analysis fails, we try some other stuff
                    # performance is slower inside this loop
                    global maxNumIterStatic;     # max no. of iterations performed
                    before "failure to converge" is returned
                    puts "Trying Newton with Initial Tangent .."
                    test NormDispIncr $TolStatic 2000 0
                    algorithm Newton -initial
                    set ok [analyze 1]
                    test $testTypeStatic $TolStatic $maxNumIterStatic 0
                    algorithm $algorithmTypeStatic
                }
            }
            if {$ok != 0} {
                puts "Trying Broyden .."
                algorithm Broyden 8
                set ok [analyze 1 ]
                algorithm $algorithmTypeStatic
            }
            if {$ok != 0} {
                puts "Trying NewtonWithLineSearch "
                algorithm NewtonLineSearch 0.8
                set ok [analyze 1]
                algorithm $algorithmTypeStatic
            }
        }
    }
}
if {$ok != 0} {;   # stop if still fails to converge
puts [format $fmt1 "PROBLEM" $IDctrlNode $IDctrl1DOF [nodeDisp $IDctrlNode $IDctrl1DOF] $LunitTXT]
return -1
}; # end if
}; # end for
integrator DisplacementControl $IDctrlNode $IDctrl1DOF $Dincr; # bring back to original increment
}; # end if
}; # end while loop
}; # end if ok !0
# --------------------------------------------------------------------
global LunitTXT;     # load time-unit text
if {  [info exists LunitTXT] != 1} {set LunitTXT "Length"};  # set blank if it has not been defined previously.
set fmt1 "%s Pushover analysis: CtrlNode %.3i, dof %.1i, Curv=%.4f /%s"; # format for screen/file output of DONE/PROBLEM analysis
if {$ok != 0 } {
puts [format $fmt1 "PROBLEM" $IDctrlNode $IDctrl1DOF [nodeDisp $IDctrlNode $IDctrl1DOF] $LunitTXT]
} else {
puts [format $fmt1 "DONE" $IDctrlNode $IDctrl1DOF [nodeDisp $IDctrlNode $IDctrl1DOF] $LunitTXT]
}

Appendix D – Rectangular Practice Program

The following sections detail the functions required for the rectangular practice program. The circular program is similar, but takes into account the circular geometry.

“main function”

```matlab
function main()

global fy Es Height inc Numinc dx1 dx2 Width cover NUMTB NUMBB Rowside S2TB S2BB DIVCT DIVCB DIVCC TB BB SB phi na DIVTB SZTRANS DIVBB Strain
coordinates(); % Call Subroutine

phiy=(fy/Es)*2.10/Height; % Estimate Yield Curvature
phiinc=phiy/10; % Increment of Phi, Curvature

dx3=1/300; % # Divisions
TOL=2.5;
phi=0; % Initial Value of Phi

%--------Set Initial Values of Compression, Tension and Moment to Zero
C1=0;
C2=0;
C3=0;
C4=0;
C5=0;
C6=0;
C7=0;
T1=0;
T2=0;
T3=0;
T4=0;
T5=0;
T6=0;
T7=0;
C1x=0;
C2x=0;
C3x=0;
C4x=0;
C5x=0;
C6x=0;
C7x=0;
T1x=0;
T2x=0;
T3x=0;
T4x=0;
T5x=0;
```

91
T6x=0;
T7x=0;

Cx=0;
Tx=0;

for j=1:147  % Number of increments = 147
    if j==1
        Moment(1,1)=0; % Start Point for M, Phi and Max Strain in Tensile Layer
        Curvature(1,1)=0;
        phi=phi+phiinc;
        MaxTen(1,1)=0;
    end

    if j==1 || j==2   % Adjust Tolerance for first two steps
        TOL=1.5;
    end

    if j==1                                     %Define Neutral Axis
        na=-Height/4;
    end

    if j==2
        na=0;
    end

    if j==3
        na=2;
    end

    if j==4
        na=2.5;
    end

    if j>2
        XYZ=MaxTen((j-1),1)*phi/Curvature((j-1),1);   % Use Linear Relationship Between Strain and
dist=-XYZ/phi;                                    %curvature to define Neutral Axis for Current Step.
        na=dist-6.6733-.18;
    end

    if j==5
        na=4;
    end
end
% Find initial Forces for Given Phi

[st1, st2, st3, st4, st5, st6] = strain(phi, na); % Find Strains for Discretization
[f1, f2, f3, f4, f5, f6, f7] = stress(st1, st2, st3, st4, st5, st6); % Find Stresses for Discretization

for i=1:inc
    forcecovT(i, 1) = Width * dx2 * (f1(i, 1) + f1((i+1), 1)) / 2; % Force in Cover Layers Top
    forcecovB(i, 1) = Width * dx2 * (f2(i, 1) + f2((i+1), 1)) / 2; % Force in Cover Layers Bottom
    xbarT(i, 1) = (DIVCT(i, 1) + DIVCT((i+1), 1)) / 2; % Fiber Centroid Top cover
    xbarB(i, 1) = (DIVCB(i, 1) + DIVCB((i+1), 1)) / 2; % Fiber Centroid Bottom Cover
    if forcecovT(i, 1) > 0
        C1 = C1 + forcecovT(i, 1); % Force in Cover Top if Compression
        C1x = C1x + forcecovT(i, 1) * xbarT(i, 1); % Moment in Cover Top if Compression
    end
    if forcecovT(i, 1) <= 0
        T1 = T1 + forcecovT(i, 1); % Force in Cover Top if Tension
        T1x = T1x + forcecovT(i, 1) * xbarT(i, 1); % Moment in Cover Top if Tension
    end
end

%-------------------Forces and Moments in Core Concrete Divisions

for i=1:Numinc
    forceconf(i, 1) = (Width - 2 * cover) * dx1 * (f3(i, 1) + f3((i+1), 1)) / 2;
    forceunconf(i, 1) = (2 * cover) * dx1 * (f4(i, 1) + f4((i+1), 1)) / 2;
    xbarConf(i, 1) = (DIVCC(i, 1) + DIVCC((i+1), 1)) / 2;
    xbarUnConf(i, 1) = (DIVCC(i, 1) + DIVCC((i+1), 1)) / 2;
    if forceconf(i, 1) > 0
        C3 = C3 + forceconf(i, 1);
        C3x = C3x + forceconf(i, 1) * xbarConf(i, 1);
    end
    if forceconf(i, 1) <= 0
        T3 = T3 + forceconf(i, 1);
    end
end
T3x = T3x + forceconf(i, 1) * xbarConf(i, 1);
end
if forceunconf(i, 1) > 0
    C4 = C4 + forceunconf(i, 1);
    C4x = C4x + forceunconf(i, 1) * xbarUnConf(i, 1);
end
if forceunconf(i, 1) <= 0
    T4 = T4 + forceunconf(i, 1);
    T4x = T4x + forceunconf(i, 1) * xbarUnConf(i, 1);
end

end

%-------Forces and Moments in Reinforcing
%Layers--Bars are divided up into fibers---
for i = 1:Numinc
    L = abs(TB(1, 2) - DIVTB(i, 1));
    r = SZTB/16;
    CHDLGN = 2*abs((r^2-L^2)^0.5);

    % Top Bars
    forctb(i, 1) = NUMTB*(f5(i, 1)+f5((i+1), 1))/2*dx1*CHDLGN;
    xbarTB = (DIVTB(i, 1)+DIVTB((i+1), 1))/2;
    if forctb(i, 1) > 0
        C5 = C5 + forctb(i, 1);
        C5x = C5x + forctb(i, 1) * xbarTB;
    end
    if forctb <= 0
        T5 = T5 + forctb(i, 1);
        T5x = T5x + forctb(i, 1) * xbarTB;
    end
end

for i = 1:Numinc
    L = abs(BB(1, 2) - DIVBB(i, 1));
    r = SZBB/16;
    CHDLGN = 2*abs((r^2-L^2)^0.5);

    % Bottom Bars
    forcebb(i, 1) = NUMBB*(f6(i, 1)+f6((i+1), 1))/2*dx1*CHDLGN;
    xbarBB = (DIVBB(i, 1)+DIVBB((i+1), 1))/2;
    if forcebb(i, 1) > 0
        C6 = C6 + forcebb(i, 1);
        C6x = C6x + forcebb(i, 1) * xbarBB;
    end
    if forcebb <= 0
        T6 = T6 + forcebb(i, 1);
        T6x = T6x + forcebb(i, 1) * xbarBB;
    end
end

if Rowside~=0
for i=1:Rowside
  % Side Bars
  forcesb(i,1)=2*f7(i,1)*(SZSB/16)^2*pi();
  if forcesb(i,1)>0
    C7=C7+forcesb(i,1);
    C7x=C7x+forcesb(i,1)*(SB(i,3));
  end
  if forcesb(i,1)<=0
    T7=T7+forcesb(i,1);
    T7x=T7x+forcesb(i,1)*(SB(i,3));
  end
end
else forcesb=0;
end
end

% Iterate until Compression and Tension Internal Forces are in Balance
for k=1:((Height/2-na)/dx3-(cover+SZTRANS/16)/dx3)%Height/(2*dx3)
  Delta(k,1)=(C1+C2+C3+C4+C5+C6+C7+T1+T2+T3+T4+T5+T6+T7-10);
  C=C1+C2+C3+C4+C5+C6+C7-10;
  T=T1+T2+T3+T4+T5+T6+T7;
  if abs(C+T)>=TOL  % if Greater than Tolerance of Balance
    dna(k,1)=dx3;  % Incremental Change of Neutral Axis
  else if abs(Delta(k,1))<TOL  % if Less than Tolerance
    dna(k,1)=0;  % Stop increment
    break  % Leave loop
  end
  na=na+dna(k,1);  % Increment neutral Axis
end
%-------- Reinitialize variables
C1=0;
C2=0;
C3=0;
C4=0;
C5=0;
C6=0;
C7=0;
T1=0;
T2=0;
T3=0;
T4=0;
T5=0;
T6=0;
T7=0;
C1x=0;
C2x=0;
C3x=0;
C4x=0;
C5x=0;
C6x=0;
C7x=0;
T1x=0;
T2x=0;
T3x=0;
T4x=0;
T5x=0;
T6x=0;
T7x=0;

%------ Find Stress and Strains, Forces and Moments

[st1, st2, st3, st4, st5, st6]=strain(phi,na);
[f1, f2, f3, f4, f5, f6, f7] = stress(st1, st2, st3, st4, st5, st6);

for i=1:inc
    forcecovT(i,1)=Width*dx2*(f1(i,1)+f1((i+1),1))/2;
    forcecovB(i,1)=Width*dx2*(f2(i,1)+f2((i+1),1))/2;
    xbarT(i,1)=(DIVCT(i,1)+DIVCT((i+1),1))/2;
    xbarB(i,1)=(DIVCB(i,1)+DIVCB((i+1),1))/2;
    if forcecovT(i,1)>0
        C1=C1+forcecovT(i,1);
        C1x=C1x+forcecovT(i,1)*xbarT(i,1);
    end
    if forcecovT(i,1)<=0
        T1=T1+forcecovT(i,1);
        T1x=T1x+forcecovT(i,1)*xbarT(i,1);
    end
    if forcecovB(i,1)>0
        C2=C2+forcecovB(i,1);
        C2x=C2x+forcecovB(i,1)*xbarB(i,1);
    end
    if forcecovB(i,1)<=0
        T2=T2+forcecovB(i,1);
        T2x=T2x+forcecovB(i,1)*xbarB(i,1);
    end
end

for i=1:Numinc
    forceconf(i,1)=(Width-2*cover)*dx1*(f3(i,1)+f3((i+1),1))/2;
    forceunconf(i,1)=(2*cover)*dx1*(f4(i,1)+f4((i+1),1))/2;
    xbarConf(i,1)=(DIVCC(i,1)+DIVCC((i+1),1))/2;
    xbarUnConf(i,1)=(DIVCC(i,1)+DIVCC((i+1),1))/2;
end
if forceconf(i,1)>0
    C3=C3+forceconf(i,1);
    C3x=C3x+forceconf(i,1)*xbarConf(i,1);
end
if forceconf(i,1)<=0
    T3=T3+forceconf(i,1);
    T3x=T3x+forceconf(i,1)*xbarConf(i,1);
end
if forceunconf(i,1)>0
    C4=C4+forceunconf(i,1);
    C4x=C4x+forceunconf(i,1)*xbarUnConf(i,1);
end
if forceunconf(i,1)<=0
    T4=T4+forceunconf(i,1);
    T4x=T4x+forceunconf(i,1)*xbarUnConf(i,1);
end

for i=1:Numinc
    L=abs(TB(1,2)-DIVTB(i,1));
    r=SZTB/16;
    CHDLGN=2*abs((r^2-L^2)^0.5);
    forcetb(i,1)=NUMTB*(f5(i,1)+f5((i+1),1))/2*dx1*CHDLGN;
    xbarTB=(DIVTB(i,1)+DIVTB((i+1),1))/2;
    if forcetb(i,1)>0
        C5=C5+forcetb(i,1);
        C5x=C5x+forcetb(i,1)*xbarTB;
    end
    if forcetb<=0
        T5=T5+forcetb(i,1);
        T5x=T5x+forcetb(i,1)*xbarTB;
    end
end
for i=1:Numinc
    L=abs(BB(1,2)-DIVBB(i,1));
    r=SZBB/16;
    CHDLGN=2*abs((r^2-L^2)^0.5);
    forcebb(i,1)=NUMBB*(f6(i,1)+f6((i+1),1))/2*dx1*CHDLGN;
    xbarBB=(DIVBB(i,1)+DIVBB((i+1),1))/2;
    if forcebb(i,1)>0
        C6=C6+forcebb(i,1);
        C6x=C6x+forcebb(i,1)*xbarBB;
    end
    if forcebb<=0
        T6=T6+forcebb(i,1);
        T6x=T6x+forcebb(i,1)*xbarBB;
    end
end
C6=C6+forcebb(i,1);
C6x=C6x+forcebb(i,1)*xbarBB;
end
if forcebb<=0
    T6=T6+forcebb(i,1);
    T6x=T6x+forcebb(i,1)*xbarBB;
end
end

if Rowside~=0
    for i=1:Rowside
        forcesb(i,1)=2*f7(i,1)*(SZSB/16)^2*pi();
        if forcesb(i,1)>0
            C7=C7+forcesb(i,1);
            C7x=C7x+forcesb(i,1)*(SB(i,3));
        end
        if forcesb(i,1)<=0
            T7=T7+forcesb(i,1);
            T7x=T7x+forcesb(i,1)*(SB(i,3));
        end
    end
else forcesb=0;
end
C=C1+C2+C3+C4+C5+C6+C7-10;
T=T1+T2+T3+T4+T5+T6+T7;

display(C);
display(T);
%display(C1);
display(T6);
%display(T4);
%display(T2);
%display(T3);
display(na);
end

Moment((j+1),1)=C1x+T1x+C2x+C3x+C4x+C5x+C6x+C7x+T2x+T3x+T4x+T5x+T6x+T7x;
% Record Moment
Curvature((j+1),1)=phi; % Record Curvature
NeutralAxis(j,1)=na; % Record Neutral Axis
MaxTen((j+1),1)=Strain; % Record Max Tension for use in other parts of program

if j>20
    phiinc=phiy/4;  % Increment if outside of yield curvature
end
phi=phi+phiinc;  % increment Curvature (phi)

C=0;
T=0;
Cx=0;
Tx=0;

%Set NA for Next Loop;

if j==1
    na=-Height/4;
end

if j==2
    na=0;
end

if j>2
    XYZ=MaxTen((j-1),1)*phi/Curvature((j-1),1);
    dist=-XYZ/phi
    na=dist-6.6733-.18;

end

%initialize for next loop (increment j)
C1=0;
C2=0;
C3=0;
C4=0;
C5=0;
C6=0;
C7=0;
T1=0;
T2=0;
T3=0;
T4=0;
T5=0;
T6=0;
T7=0;
C1x=0;
C2x=0;
C3x=0;
C4x=0;
C5x=0;
C6x=0;
C7x=0;
T1x=0;
T2x=0;
T3x=0;
T4x=0;
T5x=0;
T6x=0;
T7x=0;

end

plot(Curvature(:,1),Moment(:,1)); % Plot Moment and Curvature

display(Moment); % Display Values
display(Curvature);
display(NeutralAxis);
display(MaxTen);
“coordinates function”

function coordinates()

global NUMTB NUMBB SZTB SZBB SZSB Width Height cover SZTRANS Numx Numy Rowside SP fc DIVCC DIVCT DIVCB Numinc inc TB BB SB dx1 dx2 DIVTB DIVBB ip();

%clear global TB BB SB;
%global TB BB SB

for i=1:NUMTB
    ytb=(Height/2-cover-SZTRANS/16-SZTB/16);  % Top Bar position in Y
    spx=(Width-2*cover-SZTRANS/8-SZTB/8)/((NUMTB-1)); % Spacing in x

    x=(Width/2-cover-SZTRANS/16-SZTB/16); % Start of X direction
    TB(i,1)=0-x+(i-1)*spx;  % Top Bar X Spacing Placement
    TB(i,2)=ytb; % Top Bar Y position
    %display(ytb);
    display(TB);
end

for i=1:NUMBB
    % Bottom Bar Spacing Similar to Top Bar Spacing
    y=-(Height/2-cover-SZTRANS/16-SZBB/16);
    spx=(Width-2*cover-SZTRANS/8-SZBB/8)/((NUMBB-1));
    x=(Width/2-cover-SZTRANS/16-SZBB/16);
    BB(i,1)=0-x+(i-1)*spx;
    BB(i,2)=y;
    %display(BB(1,2));
end

if Rowside~=0
    % Spacial coordinates of Side Bars

    for i=1:Rowside
        y=(Height-2*cover-2*SZTRANS/16-SZBB/16-SZTB/16);
        spy=y/(Rowside+1);
        x=(Width-2*cover-SZTRANS/8-SZSB/8)/2;
        SB(i,1)=0-x;
        SB(i,2)=0+x;
        SB(i,3)=(y/2)-spy*i;
    end
end

Numinc=300;
inc=15;
dx1=(Height-2*cover)/Numinc; % Core Incremental change
dx2=cover/inc; % Cover incremental change

for i=1:inc+1
DIVCT(i,1)=Height/2-(i-1)*dx2; % Divide Cover Top Section into fibers
DIVCB(i,1)=-Height/2+(i-1)*dx2; % divide Cover Bottom Section into Fibers
end

for i=1:Numinc+1
    DIVCC(i,1)=((Height-2*cover)/2)-(i-1)*dx1; % Divide Core into Fibers
end
%display(DIVCT);
%display(DIVCB);
%display(DIVCC);

for i=1:Numinc+1
    DIVTB(i,1)=((Height-2*cover)/2)-(i-1)*dx1; % Divide top Bars into fibers
end

for i=1:Numinc+1
    DIVBB(i,1)=((Height-2*cover)/2)-(i-1)*dx1; % Divide Bottom Bars into Fibers
end

end
“ip (input) function”

function ip()

global NUMTB NUMBB SZTB SZBB SZSB Width Height cover SZTRANS Numx Numy Rowside SP fc fprime rhox rhoy fy esh Es eu fu Esh fprimecc

fy=60; % Yield Strength stee.
esh=.00672; % Onset of Strain hardening
Es=29000; % Mod. Elasticity Steel
eu=0.12; % Steel Ultimate Strain
fu=90; % Steel ultimate Stress
Esh=2200; % Steel Mod. at Strain hardening

rhox=0.0041; % Confined Concrete
rhoy=0.0057; % Confined Concrete

fprimecc=5604.9; % Peak Stress Confined Concrete

NUMTB=4; % number top bars
NUMBB=4; % Number Bottom Bars
SZTB=4; % Size Top Bars
SZBB=4; % Size Bottom Bars
SZSB=4; % Size Side Bars
Width=12;
Height=16;
cover=1;
SZTRANS=3; % Size Transverse Reing.
Numx=2; % Number legs x
Numy=2; % Number Legs y
Rowside=0; % rows of Side Bars
SP=4; % Spacing of Trans. Reinf
fc=5; % Peak Stress unconfined Conc.

end
function [f]=ManderConf(e)
% Mander Model for Confined Concrete
global flprime fc rhox rhoy fy fprimecc

%fprimecc=fc*1000*((2.254*sqrt((1+(7.94*flprime)/(fc*1000))))-
(2*flprime/(fc*1000))-1.254);
ecc=0.002*(1+5*((fprimecc/(fc*1000))-1));

Ec=57000*sqrt(fc*1000);
Esec=fprimecc/ecc;
r=Ec/(Ec-Esec);

ecu=0.005+1.4*(rhox+rhoy)*0.12*fy*1000/fprimecc;  % ultimate Strain
et=.00015; % Tensile Strain Ult.
ft=et*Ec; % Tensile Stress ult.

if e>-2.2*et && e<0
    x=abs(e/et);
    f=-(ft*x*r/(r-1+x^r))/1000; % Tensile Stress
end

if e>0 && e<=ecu
    x=abs(e/ecc);
    n=abs(Ec*ecc/fprimecc);
    f=(fprimecc/1000)*x*r/(r-1+x^r);  % Compressive Stress
end

if e>ecu
    f=0; % Compression > Ultimate.
end

if e<-2.2*et
    f=0;   %Tension lower than -2.2 et
end

if e==0
    f=0;
end
end
"ManderUnConf function"

```matlab
function [f]= ManderUnConf(e)
    % Mander Model unconfined (see confined model for nomenclature)
    global fc

    ec=0.002;
    Ec=57000*sqrt(fc*1000);
    Esec=fc*1000/ec;
    r=Ec/(Ec-Esec);

    ecu=0.004;
    et=.00015;
    ft=et*Ec;

    if e>-2.2*et && e<=0
        x=abs(e/et);
        %n=Ec*ft/et;
        f=-(ft/1000)*x*r/(r-1+x^r);
    end

    if e>0 && e<=ecu
        x=abs(e/ec);
        %n=abs(Ec*ec/(fc*1000));
        f=fc*(1000/1000)*x*r/(r-1+x^r);
    end

    if e>ecu
        f=0;
    end

    if e<-2.2*et
        f=0;
    end

    end
```
"steel function"

function [f]=steel(e)
% Steel Stress Function
global fy Es esh fu eu Esh

ey=fy/Es;

if abs(e)<=ey
  f=abs(e)*Es*sign(e);  % Between Yield in Tension & Compression
end

if abs(e)<=esh && abs(e)>ey
  f=sign(e)*fy;  % More than yield, but before esh in Tension and Compression
end

if abs(e)>esh && abs(e)<=eu

  p=Esh*(eu-esh)/(fu-fy);
  f=(fu+(fy-fu)*(abs((eu-abs(e)))/(eu-esh))^p)*sign(e);  % More than esh in Tension and Compression
end
if abs(e)>eu
  f=0;  % Greater than ultimate in Tension and Compression
end

end
"strain function"

function [st1, st2, st3, st4, st5, st6]=strain(curv, na)

%----------------- Find Strains in fibers throughout section-------------------

global DIVCC DIVCT DIVCB inc Numinc TB BB SB Rowside Height DIVTB SZTB
DIVBB SZBB Strain
% display(curv);

st=curv*(Height/2-na);
% display(st);
for i=1:inc+1
  % Top layer Cover
  if DIVCT(i,1)>na % Greater than Neutral Axis
    st1(i,1)=st/(Height/2-na)*(DIVCT(i,1)-na);
  end
  if DIVCT(i,1)<na % Less than Neutral axis
    st1(i,1)=-st/(Height/2-na)*abs(DIVCT(i,1)-na);
  end
  if DIVCT(i,1)== na % Neutral Axis
    st1(i,1)=0;
  end
end

for i=1:inc+1
  % Bottom Layer Cover
  if DIVCB(i,1)>na
    st2(i,1)=st/(Height/2-na)*(DIVCB(i,1)-na);
  end
  if DIVCB(i,1)<na
    st2(i,1)=-st/(Height/2-na)*abs(DIVCB(i,1)-na);
  end
  if DIVCB(i,1)== na
    st2(i,1)=0;
  end
end

for i=1:Numinc+1
  % Confined Concrete
  if DIVCC(i,1)>na
    st3(i,1)=st/(Height/2-na)*(DIVCC(i,1)-na);
  end
  if DIVCC(i,1)<na
    st3(i,1)=-st/(Height/2-na)*abs(DIVCC(i,1)-na);
  end
  if DIVCC(i,1)== na
    st3(i,1)=0;
  end
end

end
for i=1:Numinc+1
  % Top Bars
  if DIVTB(i,1)>na && DIVTB(i,1)>=(TB(1,2)-SZTB/16) &&
  DIVTB(i,1)<=(TB(1,2)+SZTB/16)
      st4(i,1)=st/(Height/2-na)*(DIVTB(i,1)-na);
  end
  if DIVTB(i,1)<na && DIVTB(i,1)>=(TB(1,2)-SZTB/16) &&
  DIVTB(i,1)<=(TB(1,2)+SZTB/16)
      st4(i,1)=-st/(Height/2-na)*abs(DIVTB(i,1)-na);
  end
  if DIVTB(i,1)== na
      st4(i,1)=0;
  end
  if DIVTB(i,1)<(TB(1,2)-SZTB/16) || DIVTB(i,1)>(TB(1,2)+SZTB/16)
      st4(i,1)=0;
  end
end
for i=1:Numinc+1
  % Bottom Bars
  if DIVBB(i,1)>na && DIVBB(i,1)>=(BB(1,2)-SZBB/16) &&
  DIVBB(i,1)<=(BB(1,2)+SZBB/16)
      st5(i,1)=st/(Height/2-na)*(DIVBB(i,1)-na);
  end
  if DIVBB(i,1)<na && DIVBB(i,1)>=(BB(1,2)-SZBB/16) &&
  DIVBB(i,1)<=(BB(1,2)+SZBB/16)
      st5(i,1)=-st/(Height/2-na)*abs(DIVBB(i,1)-na);
  end
  if DIVBB(i,1)== na
      st5(i,1)=0;
  end
  if DIVBB(i,1)<(BB(1,2)-SZBB/16) || DIVBB(i,1)>(BB(1,2)+SZBB/16)
      st5(i,1)=0;
  end
end
if Rowside~=0
  % Side Bars
  for i=1:Rowside
    if SB(i,3)>na
      st6=st/(Height/2-na)*(SB(i,3)-na);
    end
    if SB(i,3)<na
      st6=-st/(Height/2-na)*abs(SB(i,3)-na);
    end
    if SB(i,3)== na
      st6=0;
    end
  end
else st6=0;
end
Strain=min(st5(:,1)); % Max Tensile Strain for Rectangular section
end
“stress function”

function [f1, f2, f3, f4, f5, f6, f7] = stress(st1, st2, st3, st4, st5, st6)

% -------------------------Stresses for Section fibers-----------------

global Numinc inc Rowside

for i=1:inc+1
    f1(i,1)=ManderUnConf(st1(i,1));
    f2(i,1)=ManderUnConf(st2(i,1));
end

for i=1:Numinc+1
    f3(i,1)=ManderConf(st3(i,1));
    f4(i,1)=ManderUnConf(st3(i,1));
end

for i=1:Numinc+1
    f5(i,1)=steel(st4(i,1));
end

for i=1:Numinc+1
    f6(i,1)=steel(st5(i,1));
end

if Rowside~=0
    for i=1:Rowside
        f7(i,1)=steel(st6(i,1));
    end
else f7=0;
end

end