Homework 3

Louis Lin

March 3, 2021

University of California, San Diego

Prof. Joel Conte

SE 201B Nonlinear Analysis

Table of Contents

List of Figure

List of Tables

Introduction

This report examines the effects of fiber level discretization and constitutive models on monotonic and cyclic, uniaxial moment-curvature diagrams and biaxial moment interaction diagrams. In nonlinear analysis, discretization of the section and the parameters used for the constitutive modeling plays a large role in the overall behavior of the element. Discretization affects the quality of the measured response: whether or not cracking or yielding is well captured. The constitutive model is the stress-strain relationship of the section which dictates the amount of force the fiber is able to resist.

The effects of inadequate discretization are explored in this paper as well as the different constitutive models available in Opensees. There are four constitutive models used for each of the three materials assumed in the section of this paper, which is part of a reinforced concrete column. In Opensees, these four constitutive models are CONCRETE01, CONCRETE02, STEEL01, and STEEL02. The three materials considered in this column are the core concrete, the cover concrete, and the steel reinforcement. The material properties used in the model for the confined concrete and unconfined concrete are shown below.

Table 1 Parameters Used in CONCRETE01 and CONCRETE02 for Confined and Unconfined Concrete

Opensees Parameters	Units	Unconfined Concrete		Confined Concrete	
$fpc	[MPa]	f_c'	-32.5	f_{cc}'	-47.9
$E	[MPa]	E_c	$27{,}000$	E_c	$27{,}000$
$epsc0	[-]	ϵ'_c	$2f_c'/E_c$	ϵ'_{cc}	$2f_{cc}'/E_{cc}$
$ft	[MPa]	f_{cr}'	1.9	f_{cr}'	1.9
$lambda	[-]	λ	0.25	λ	0.25
$Ets	[Mpa]		$0.1E_c$		$0.1E_c$
$fpcU	[Mpa]	f_{cu}'	$0.2f_c'$	f_{ccu}'	$0.2f_{cc}'$
$epsU	[-]	ϵ_{cu}'	-0.004	ϵ_{ccu}'	-0.0276

Table 2 Parameters Used in STEEL01 and STEEL02

Opensees Parameters	Units	Reinforcing Steel	
$Fy	[MPa]	F_y	455
$E	[MPa]	E_s	215,000.
$b	[-]	b	0.01
$R0	[-]	R_0	20
$cR1	[-]	cR_1	0.925
$cR2	[-]	cR_2	0.15
$a1	[-]	a_1	0
$a2	[-]	a_2	1
$a3	[-]	a_3	0
$a4	[-]	a_4	1
$sigInit		-	0

In the case of the analysis, a zero-length section element was used in generating the moment curvature diagrams in this report. This takes away the geometric nonlinearity of the system when computing the capacity of the section.

Part (a) Examining the Hysteresis Loops of Constitutive Models for Concrete and Steel

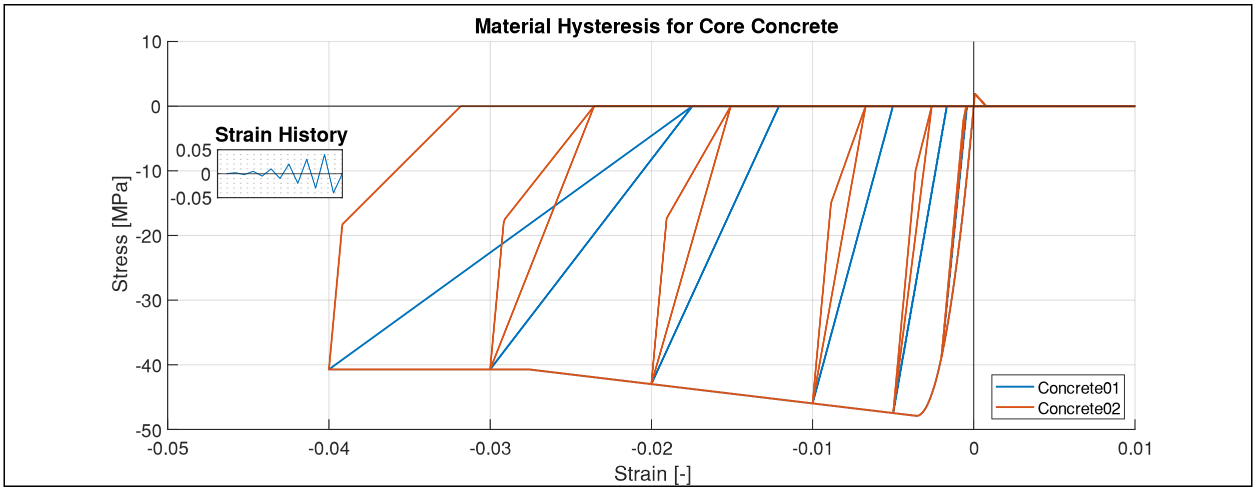


Figure 1 Material Hysteresis Loops for Core Concrete Using CONCRETE01 and CONCRETE02

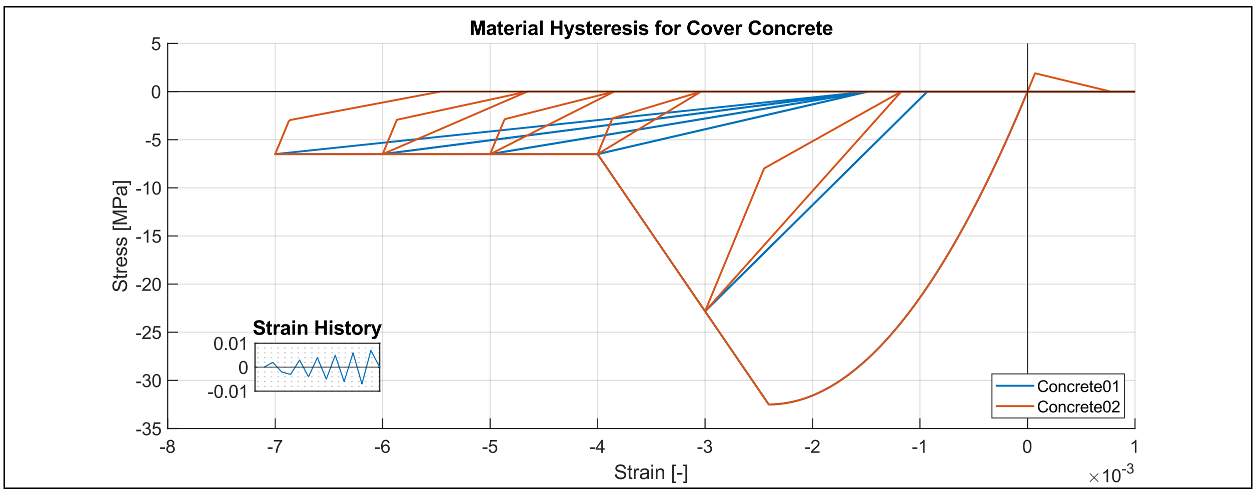


Figure 2 Material Hysteresis Loops for Cover Concrete Using CONCRETE01 and CONCRETE02

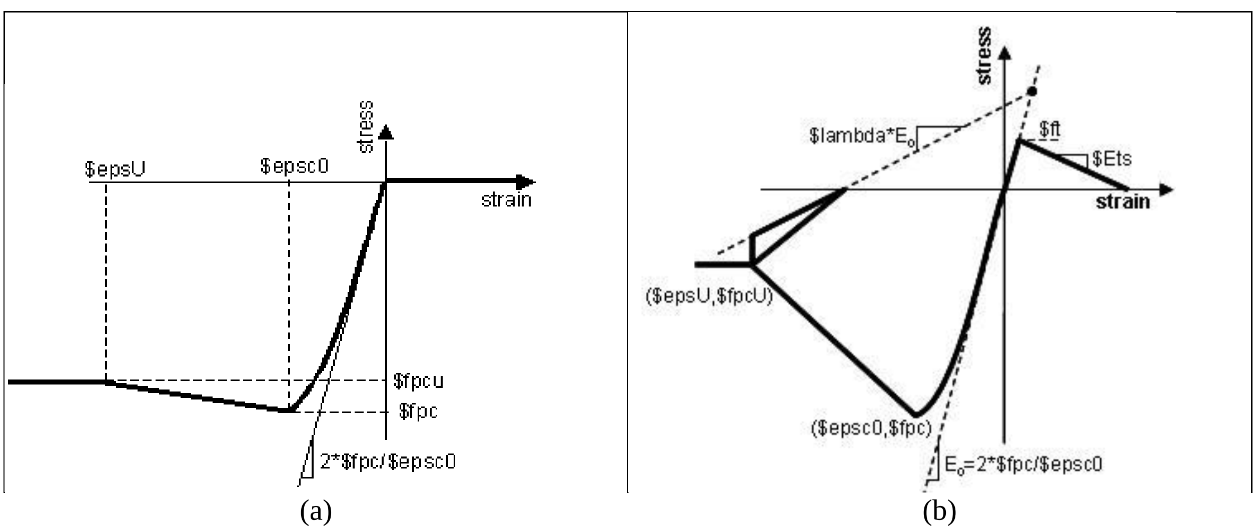


Figure 3 Parameters used in Opensees CONCRETE01 and CONCRETE02

As shown in the Figure 1 and Figure 2, there are two constitutive models for the concrete as well as two types of concrete identified in the column. The core concrete is defined as the concrete confined by the transverse reinforcement while the cover concrete is the portion not contained by the transverse reinforcement. The core concrete has an increased strength due to the confining pressure produced by the hoop reinforcements. This restraint delays the onset of cracking of the concrete which allows the concrete to resist higher forces. There is a significant relation between confinement and increased strength which warrants the modeling of these two concretes as separate materials.

CONCRETE01 is based on the Kent-Scott-Park concrete material model which ignores the tensile strength of concrete and assumes linear unloading/reloading of concrete. It is defined in the code by the following command. In Figure 3 (a), the parameters used to define the stress-strain curve is shown.

uniaxialMaterial Concrete01 $matTag $fpc $epsc0 $fpcu $epsU

$matTag	integer tag identifying material
$fpc	concrete compressive strength at 28 days (compression is negative)
$epsc0	concrete strain at maximum strength*
$fpcu	concrete crushing strength
$epsU	concrete strain at crushing strength*

CONCRETE02 is based on a similar model as CONCRETE01 however it considers that the concrete has tensile strength as well as a linear tension softening region after it reaches its tensile strength. The other behavior CONCRETE02 captures is the softening of the concrete as it unloads. The concrete will unload at a slope parallel to its original elastic modulus for a bit but will then soften by a factor lambda, which is more reflective of experimental data. Modeling these two phenomena is more rigorous than CONCRETE01 and has a higher computational requirement. This increase in accuracy would be more appropriate for matching experimental data. The input parameters commands are shown below and also shown graphically in Figure 3 (b).

uniaxialMaterial Concrete02 $matTag $fpc $epsc0 $fpcu $epsU $lambda $ft $Ets

$matTag	integer tag identifying material
$fpc	concrete compressive strength at 28 days (compression is negative)
$epsc0	concrete strain at maximum strength
$fpcu	concrete crushing strength
$epsU	concrete strain at crushing strength
$lambda	ratio between unloading slope at $epscu and initial slope
$ft	tensile strength
$Ets	tension softening stiffness (absolute value) (slope of the linear tension softening branch)

When comparing the core concrete and the cover concrete, the cover concrete has a huge drop off in strength after peak-strength is reached. This large change in slope could lead to convergence issues when using iterative-solvers that might overshoot the correct converged state like the original Newton-Raphson Method.

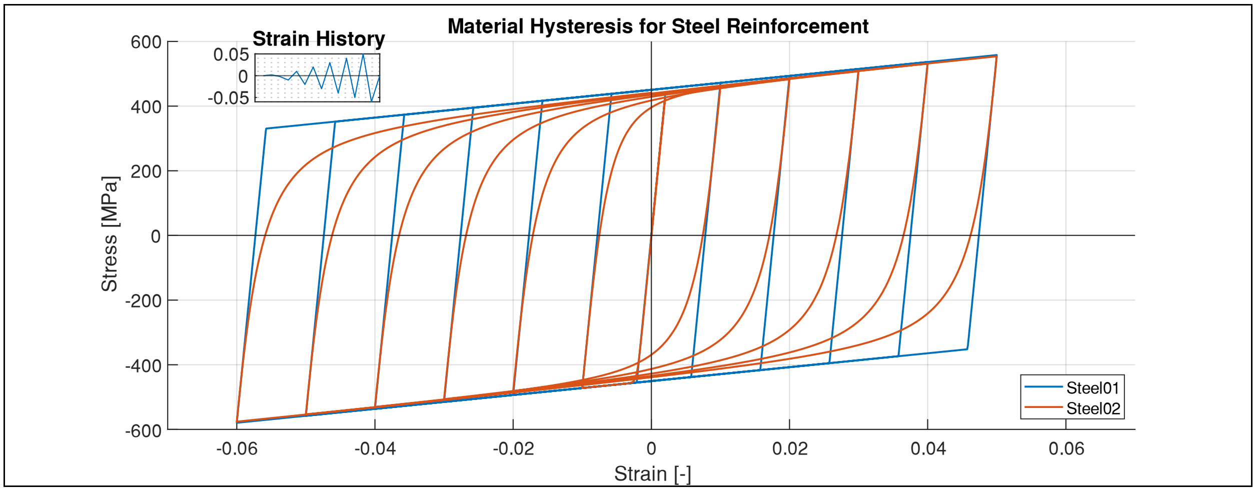
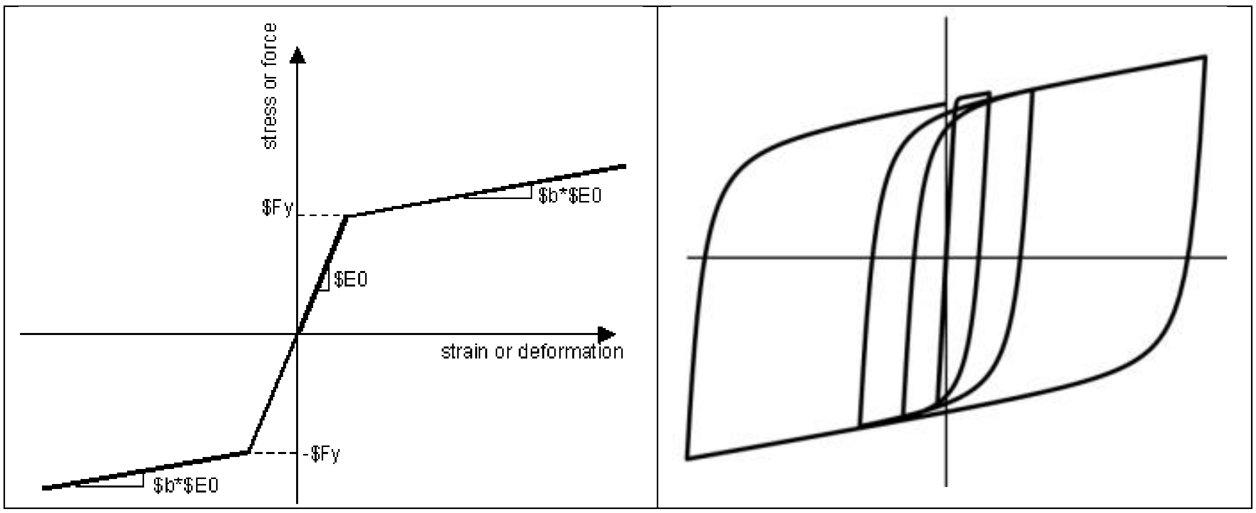


Figure 4 Material Hysteresis Loops for Steel Reinforcement Using STEEL01 and STEEL02 in Opensees



(a) STEEL01 (b) STEEL02

Figure 5 Parameters used in Opensees STEEL01 and STEEL02

STEEL01 and STEEL02 are two steel constitutive models based on the Giuffré-Menegotto-Pinto steel model, as shown in Figure 5. Both models assume isotropic strain hardening of the steel during loading and unloading. STEEL01 is considered bilinear with an elastic region of slope E_o and a strain hardening region with a modified slope of $b \cdot E_o$, where b is the strain-hardening ratio. In STEEL02, there are radius parameters that transitions between the elastic and strain hardening region, evident in that the curve being rounded around yielding. This is to capture the Bauschinger effect which is the phenomenon of gradual softening of the steel after the first yielding and progressive softening during cyclic loading. The growth of the strain-hardening region can be controlled as well as the development of the Bauschinger effect by certain parameters in Opensees. Common to both models is the lack of consideration for an ultimate strength of the steel; the stress-strain relationship is assumed to go on indefinitely which is not reflective of real material behavior. Both models also don't consider buckling of the steel when the reinforcing steel is exposed and no longer braced by the concrete. This behavior is important when considering the cyclic loading of a structure after the crushing of concrete, however most structures are not designed to allow concrete to crush. The commands for the two constitutive models as used in Opensees are given below.

uniaxialMaterial Steel01 $matTag $Fy $E0 $b <$a1 $a2 $a3 $a4>

$matTag	integer tag identifying material
$Fy	yield strength
$E0	initial elastic tangent
$b	strain-hardening ratio
$a1	isotropic hardening parameter (optional)
$a2	isotropic hardening parameter (optional).
$a3	isotropic hardening parameter (optional)
$a4	isotropic hardening parameter (optional)

uniaxialMaterial Steel02 $matTag $Fy $E $b $R0 $cR1 $cR2 <$a1 $a2 $a3 $a4 $sigInit>

$matTag	integer tag identifying material
$Fy	yield strength
$E0	initial elastic tangent
$b	strain-hardening ratio
$R0 $CR1 $CR2	parameters to control the transition from elastic to plastic branches. Recommended values: **$R0**=between 10 and 20, **$cR1**=0.925, **$cR2**=0.15
$a1	isotropic hardening parameter, (optional)
$a2	isotropic hardening parameter, (optional default = 1.0).
$a3	isotropic hardening parameter. (optional default = 0.0)
$a4	isotropic hardening parameter (optional default = 1.0)
$sigInit	Initial Stress Value (optional, default: 0.0)

Part (b) Monotonic Moment Curvature Analysis

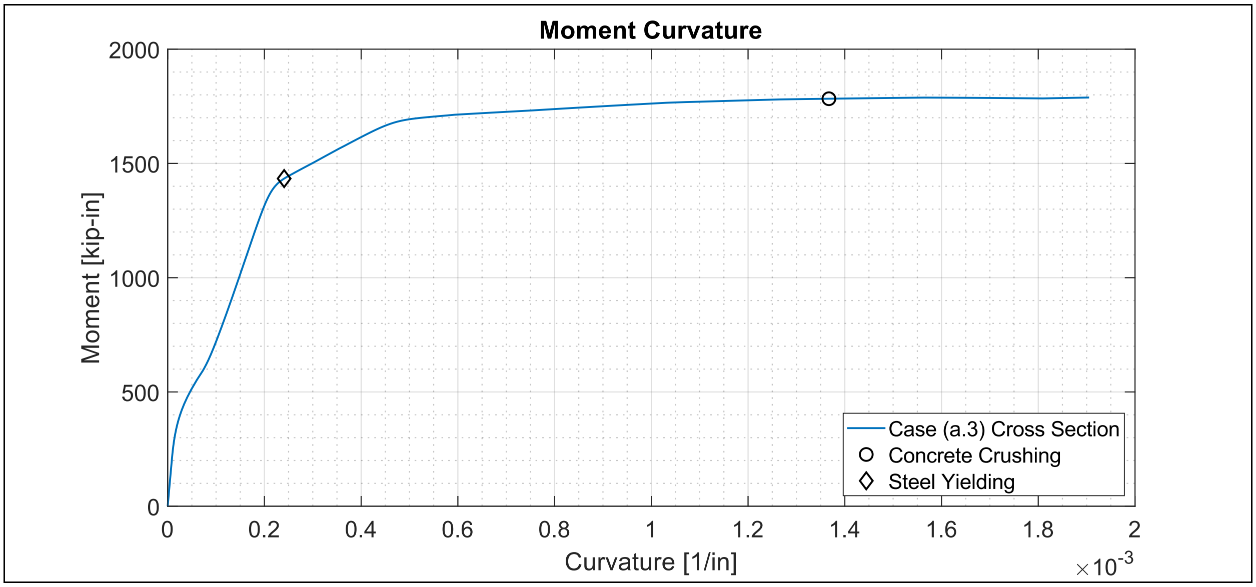


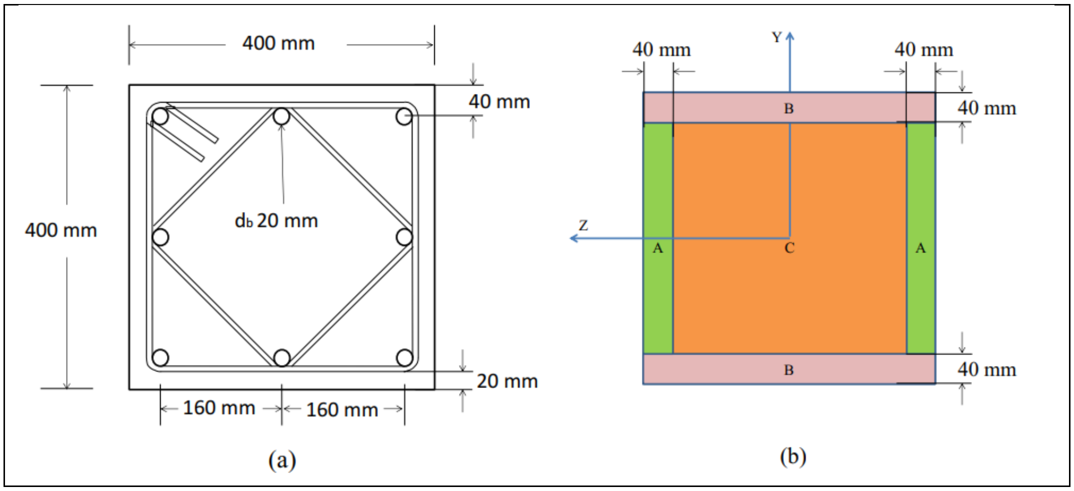
Figure 6 Monotonic Moment Curvature Analysis

A moment curvature analysis is performed for the column given the cross section shown in Figure 7. A moment diagram is a tool which allows designers to predict the moment capacity of a section given the curvature, the amount of bending, experience by the section. A monotonic moment curvature analysis is given in Figure 6 for curvature values of $\kappa = [0.0013, 0.0040, 0.0080, 0.0130, 0.0200]/H$, where H was the height of the cross section. It is also shown where the concrete crushes, the point where the strain in extreme compressive fiber of the concrete reaches -0.003, and it is also shown where the steel yields, where the extreme tensile steel fiber stress equals 65.9 ksi. Comparing these two points in design is important because when either of these two states are reached, there will be changes in structural performance as shown by the decrease in slope after both points. When concrete crushes first, the structural strength will decrease abruptly as the concrete will no longer be able to carry any capacity. This brittle failure is best avoided by designing the cross section's capacity to be tensioned-dominated, which means yielding of the tensile steel reinforcement first.

Other things to note on the moment curvature analysis are the points where the curve changes slope which signifies a chape in stiffness of the section. The initial kink is due to the cracking of the concrete. The cracked concrete is unable to contribute to the stiffness of the section and thus the curve has a shallower slope. The second major change in slope occurs, for Figure 6, is the yielding of the steel reinforcement; subsequent slope changes before the crushing of concrete can be attributed to the other layers of reinforcement also yielding. When yielding occurs, the constitutive models of steel depict the steel as softening and plastically deforming along a smaller slope which means a smaller increase in resistance for the same amount of deformation.

Part (c) Section Discretization

The cross section of the column is given in Figure 7(a). The concrete was discretized into three zones which represent the core concrete (Section C) and the cover concrete (Sections A and B). The steel rebars were modeled as circular cross sections with area of 0.487 in^2, spaced about their real centers.



(a) Cross Section (b) Concrete Discretization Zones

Figure 7 Column Cross Section and Concrete Discretization Zones

The three discretization explored in this report are shown in Figure 8. Each zone was discretized based on the numbers of fibers as prescribed in Table 3.

Table 3 Analysis Cases for Effects of Section Discretization

Case	Zone A (# of layers)	Zone A (# of layers)	Zone A (# of layers)	Total number of fibers
(a.1)	3	1	3	11
(a.2)	8	2	8	28
(a.3)	20	5	20	70

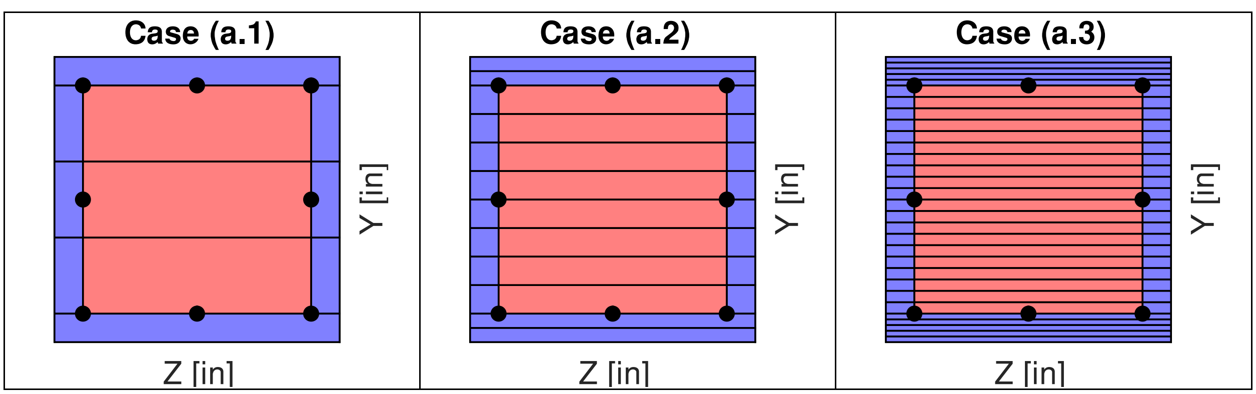


Figure 8 Depiction of Section Discretizations

In Figure 8, the purple color is for the cover concrete while the red is the core concrete. Since the cross section was only bent about one axis, the Y-axis, there was no need to discretize the section in the other direction.

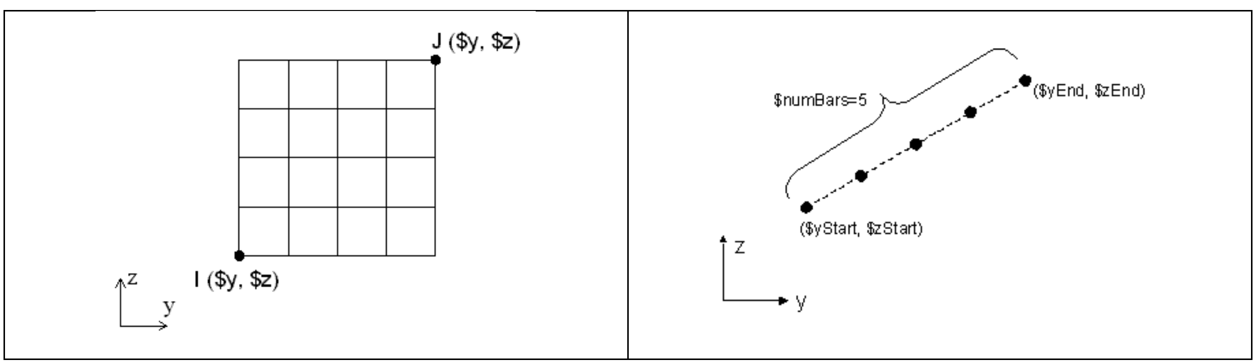
The section was discretized using the 'patch' and 'layer' commands in Opensees. In the above cases, the patch command is used to define rectangular regions and the layer command is used to generate a number of fibers along a line.

patch rect $matTag $numSubdivY $numSubdivZ $yI $zI $yJ $zJ

$matTag	tag of previously defined material (UniaxialMaterial tag for a FiberSection or NDMaterial tag for use in an NDFiberSection)
$numSubdivY	number of subdivisions (fibers) in the local y direction.
$numSubdivZ	number of subdivisions (fibers) in the local z direction.
$yI $zI	y & z-coordinates of vertex I (local coordinate system)
$yJ $zJ	y & z-coordinates of vertex J (local coordinate system)

layer straight $matTag $numFiber $areaFiber $yStart $zStart $yEnd $zEnd

$matTag	material tag of previously created material (UniaxialMaterial tag for a FiberSection or NDMaterial tag for use in an NDFiberSection)
$numFibers	number of fibers along line
$areaFiber	area of each fiber
$yStart $zEnd	y and z-coordinates of first fiber in line (local coordinate system)
$yEnd $zEnd	y and z-coordinates of last fiber in line (local coordinate system)



(a) Patch Command (b) Layer Command

Figure 9 Patch and Layer Commands in Opensees

In the patch command, using the rectangular option, the bottom leftmost corner and then the uppermost right corners are defined on the cross section plane. Then the amount of discretization in the two axes of the cross section can be specified. The layer command can also be used to generate a number of fibers in an arc, but not for this report. The number of fibers needs to be specified as well as the area for each fiber.

Part (d) Effects of Section Discretization on Moment Curvature Diagram

Discretization plays a large part in the convergence of model as well as the results of the moment-curvature diagram. With more sections, the model is able to better capture the behavior of the section as it undergoes changes in its constitutive model. In Figure 10, it is observed that the cross section (a.1) shows a huge decrease in moment capacity when the concrete crushes. This happens when a large fiber of the cross section changes its stress-strain relationship. The drop in strength comes from the entire fiber of concrete losing capacity to resist force. This is not reflective of the real behavior of concrete as the fibers will not crush as a whole fiber but rather in smaller pieces. The dip in the moment curvature shown by section (a.1) is not real but rather an effect of inadequate discretization of the cross section. For section (a.1), after the concrete crushes, the steel will take on an increase in strain which stiffens the structure again after the huge drop.

In terms of convergence, all three cross section converge to similar behaviors for the moment-curvature diagram up to the crushing of concrete. Before cracking, the moment capacities of all three discretized sections have the same moment capacity with minor differences. Afterwards, section (a.1) does not develop the same amount of moment capacity as the other two sections even though they undergo the same strain. The fibers will develop less force in the larger discretization of the section, because it will approximate the stress of the section more when calculating the force developed by each fiber. After the reinforcing steel yields, section (a.1) has a slightly higher moment capacity because the assumed stress in the concrete fibers is higher than the ones with smaller discretization. This is again due to the approximation of the stress per fiber, as the larger fibers will linearize more of the stress.

The difference between discretizing the section into (a.2) and (a.3) is minimal and should be noted. The need for highly discretize section is not necessary for capturing the behavior of the section as shown.

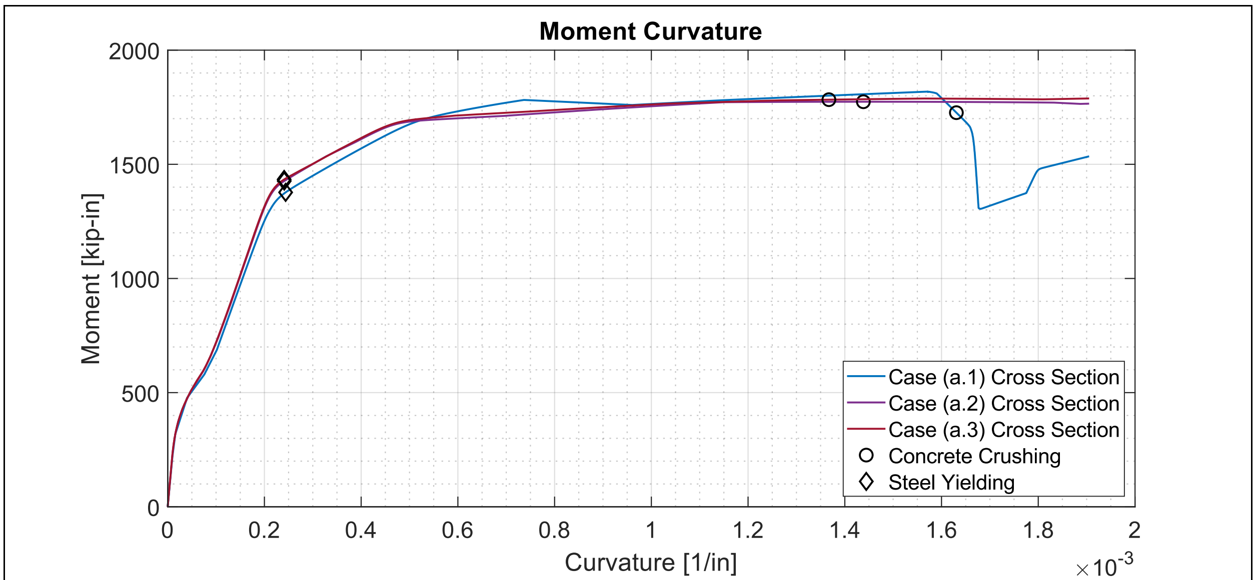


Figure 10 Comparison of the Effects Section Discretization on Moment Curvature

Part (e) Moment Curvature Under Various Axial Load Ratios

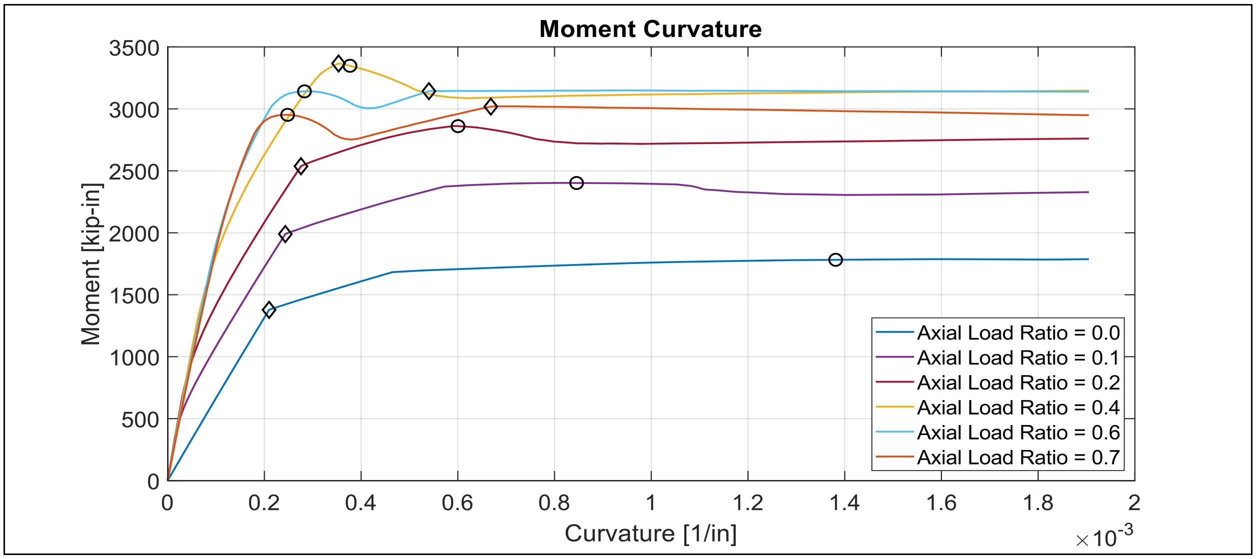


Figure 11 Comparison of Moment Curvature Under Various Axial Loads

In this section, the moment curvature diagram is examined for various axial load ratios. The axial load ratio describes the amount of axial load to axial capacity of the section as calculated as $\frac{P}{f_c' \cdot A_g}$, where P is the axial load and $f_c' \cdot A_g$ is the axial capacity of the section. In this analysis, f_c' was assumed to be the compressive strength of the cover concrete. The effect of axial load on the moment curvature is necessary since sections, as used in real, load-carrying elements, will experience a combination of axial load and moments.

In Figure 11, moment curvature diagrams for axial load ratios of $[0.0, 0.1, 0.2, 0.4, 0.6, 0.7]$ are shown. The axial load ratio increases the moment capacity of the section up to a certain axial load ratio. This is evident in the higher peaks of the moment curvature diagrams as the axial load ratio increases but decreases after the load ratio of 0.4. This phenomenon is caused by shifting the failure mechanism of the cross section from tensile steel reinforcement yielding to concrete crushing. The strain profile of the cross section with higher axial load ratio has higher compressive strains, which means the concrete will crush earlier while the tensile steel reinforcement has not yielded. The crushing of concrete will lead to a sudden drop in section capacity before the section regains some strength due to the reinforcement carrying more strain. The load ratio of 0.4 is relevant to this section because it represents the balance failure point which is where the crushing of concrete and yielding of tensile steel reinforcement happens simultaneously. This is evident on the 0.4 axial load ratio curve in that the points where yielding and crushing occur right next to each other. If the axial load per moment capacity where the concrete crushes was plotted, as later shown in Figure 19, called a moment-axial interaction diagram, the axial load to moment capacity relationship is even more clear.

For smaller axial load ratios, the yielding of the tensile steel reinforcement occurs first which is described as tension-controlled failure, while at higher axial load, the section will crush first, which is described as compression-controlled failure. Tensioned-controlled failure is more ductile since the capacity of the yielded reinforcement is still presented while the compression-controlled failure is brittle because the crushed concrete cannot resist any force. This is why for higher axial loads, there is a large drop in strength after one of the crushing of concrete. The balanced point is found to occur when the axial load reaches approximately 460 lbf. Given the section capacity of approximately 1170 lbf, the balanced axial load ratio is 0.393 which is around 0.4 found by inspection. It is also noted the increase axial force delays the onset of cracking in the section as the strain of the concrete sections reaches the cracking strain much later.

$\kappa = 8.255e-5$ $\kappa = 0.000254$ $\kappa = 0.000508$ $\kappa = 0.0008255$ $\kappa = 0.00127$

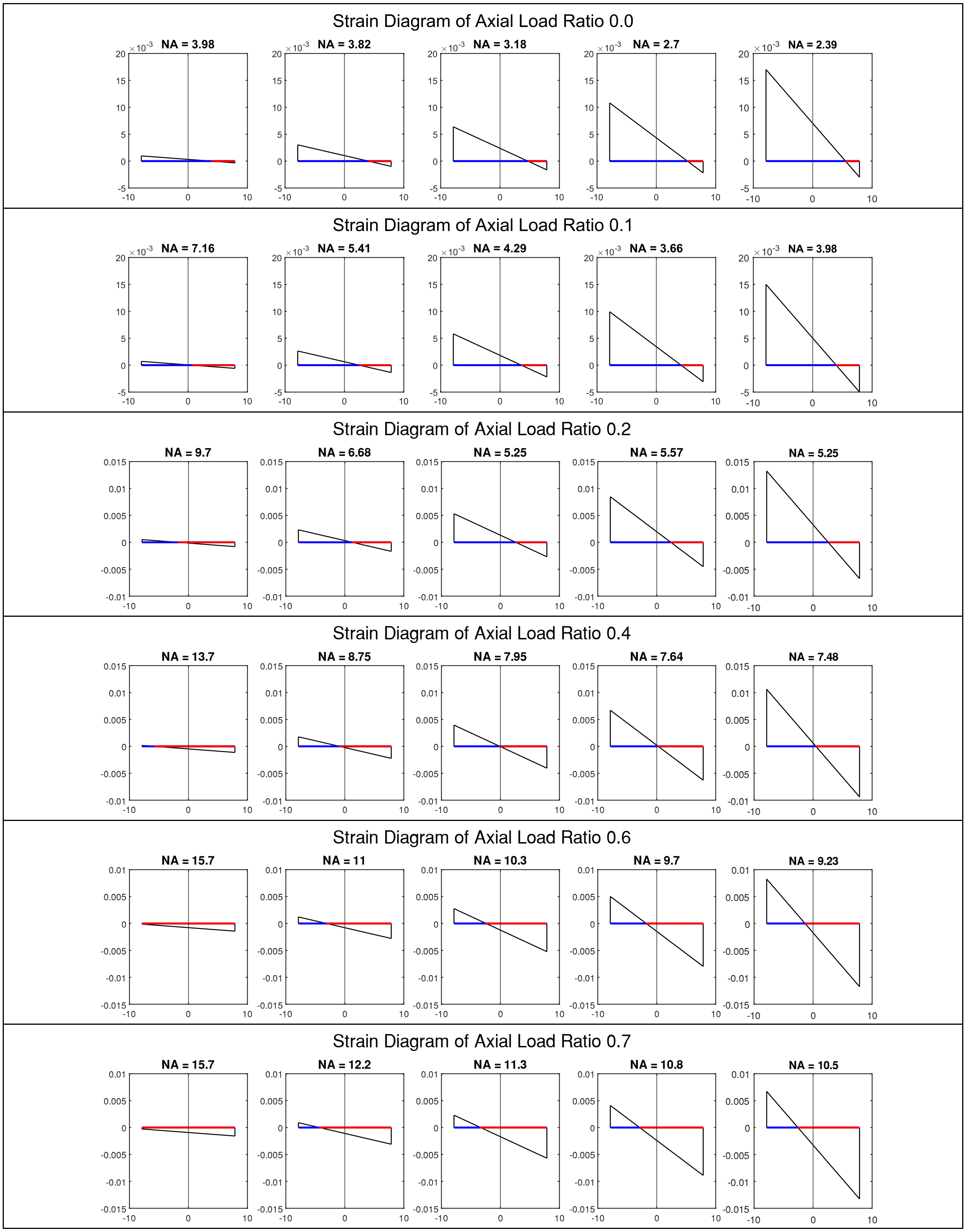


Figure 12 Various Strain Profiles for Various Curvatures for Various Axial Loads

The strain profiles were found for curvatures equal to $[0.0013, 0.0040, 0.0080, 0.0130, 0.0200]/H$ where H is the height of the section. The right side of the strain profile is the top of the section and experiences compression while the bottom is in tension due to convention of the positive moment being applied. The neutral axis's position is defined by the fiber in the cross section for which the strain remains zero and separates the compressive strain and tensile strain regions. Its location is reported as the distance from the top of the section.

The first note from the above figures is the progression of the strain profile as the curvature of the section increases. The slope of the strain profile for a section under bending, assuming that the section remained plane, is the curvature. Thus, as the curvature increases, the overall maximums and minimums for the strain profile will also increase. Across all the axial loads, the slope of the strain profile for a given curvature is consistent with one another.

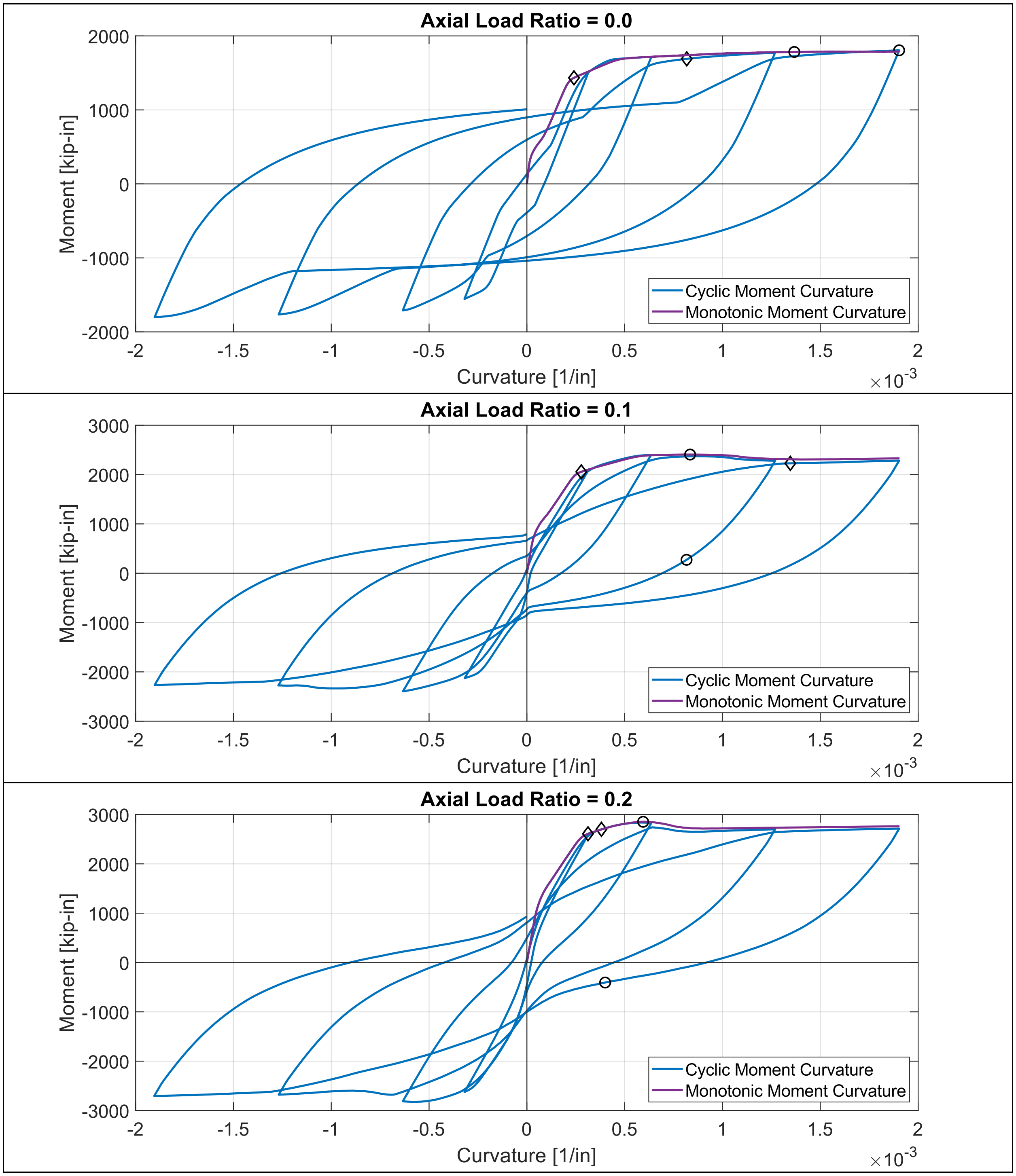
For higher axial loads, the neutral axis will shift towards the bottom as more of the section is in compression. The axial load applies a uniform compressive strain on the section which is evident in the downward shift in the strain profile. This strain profile develops a higher compressive stress and force in order to balance the applied compressive axial force.

For compression-controlled sections, the migration of the neutral axis will always be towards the bottom of the section. This is because the strain in the extreme compression fibers of the concrete will reach its crushing strain faster and thus there is a huge decrease in strength of the concrete. The concrete won't be able to carry any stress and thus more of the section will need to be in compression in order to balance the applied load, shifting the neutral axis downwards.

For tension-controlled sections, the migration of the neutral axis will rise towards the top of the section. As the concrete reaches its tensile cracking strain, less of the concrete is able to develop stress and thus the section resists less force. For no or low axial loads, the section needs progressively less compressive force to balance the tensile force even as the curvature increases because the force developed in the steel slows down after it yields. As the axial loads grows though, more of the section needs to be in compression to reach equilibrium and that is why the neutral axis shifts towards the bottom again.

Part (f) Cyclic Moment Curvature Analysis

A cyclic moment curvature analysis shows the behavior of a section as it undergoes curvature loading and unloading which is useful in exploring the effects of the nonlinear behavior of the materials on the moment curvature diagram. A cyclic moment curvature analysis is performed with a curvature time history equal to $\kappa = [0.0, 0.005, 0.005, 0.01, 0.01, 0.02, 0.02, 0.03, 0.03, 0.0]/H$ where H is the height of the section. The section used the discretization case (a.3) and the constitutive models of CONCRETE02 and STEEL02.



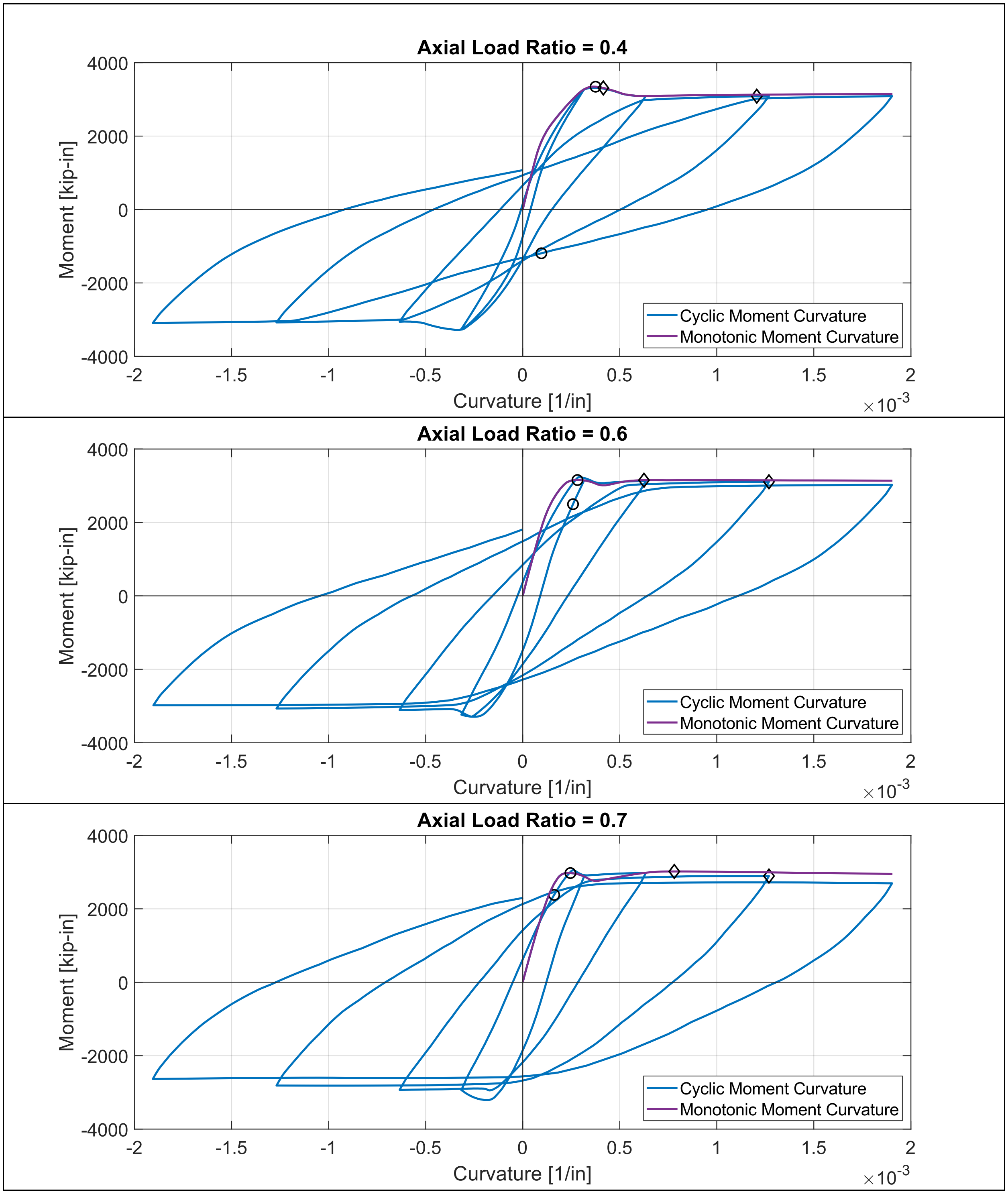


Figure 13 Cyclic Moment Curvature Diagrams for Various Axial Loads

The monotonic moment curvature is shown to be a good envelope for the cyclic moment curvature response for low axial loads but diverges a bit at higher axial. There is degradation in the stiffness for the cyclic moment curvature which is caused in part by the steel yielding. One phenomenon that can be observed is pinching of the curve on curvature reversals. This jump in stiffness is caused by cracks in the section closing and regaining strength. At higher axial loads, pinching is not as apparent because there is less cracking.

The axial strain of the section is measured at the centroid of the section. It is shown to vary linearly with the curvature as evident in Figure 14. This response is expected as the curvature of a section is the slope of the strain profile. As the slope increases, the axial strain at the centroid of the section will also increase. Residual axial strain does not occur at low axial load ratios but will after the crushing or cracking of concrete. When the concrete crushes, the centroidal axial strain will persist and grow in compressive strain to develop the force needed to balance the applied load. When the concrete cracks, the axial strain will grow in tensile strain as more of the section loses capacity and further cracks. For the zero axial load case, the flatter regions are where the concrete cracks are closing, and the strain doesn't change much because these regions need to be loaded again. This phenomenon contributes to plastic hinge elongation and permanent axial strain.

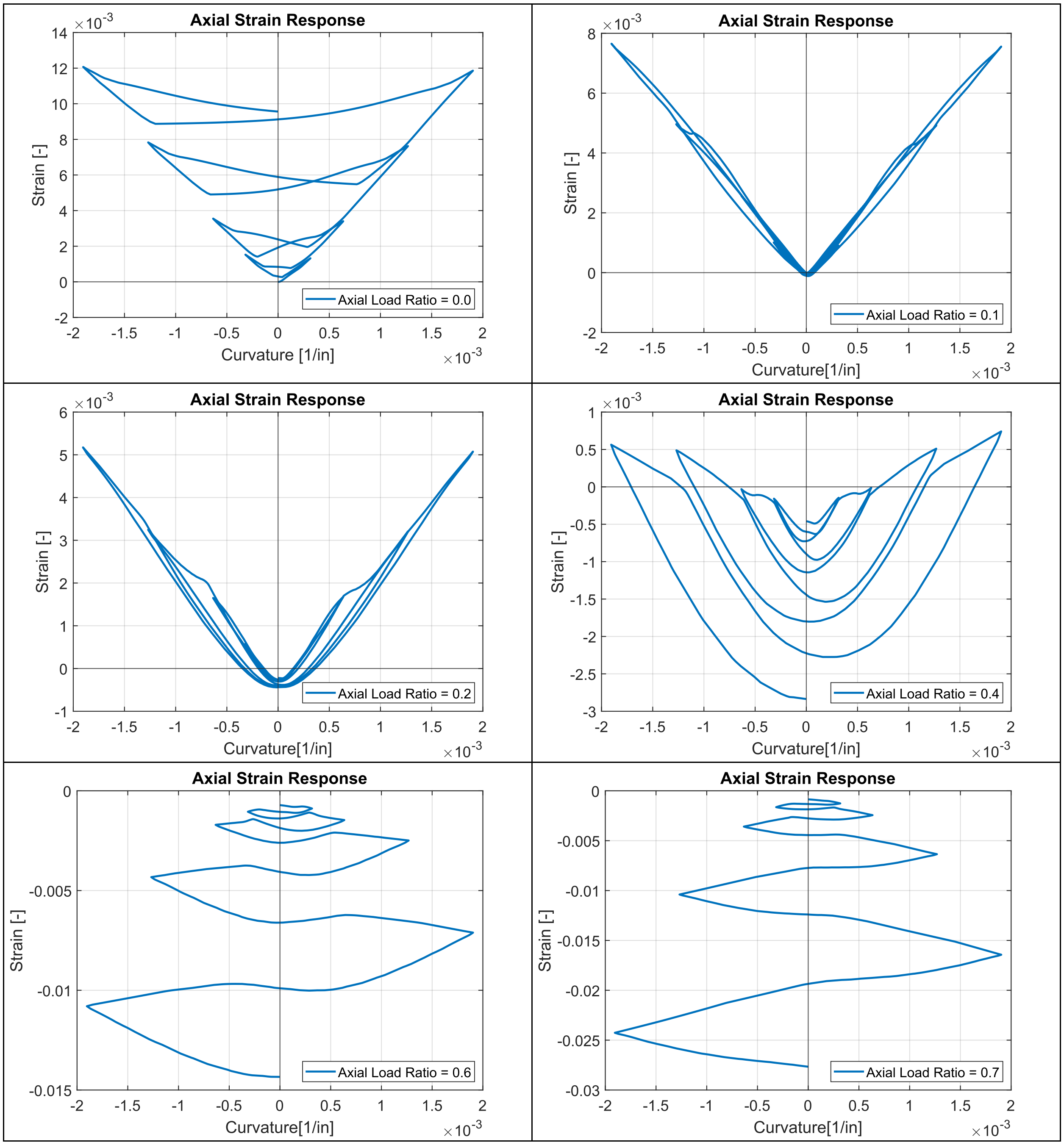
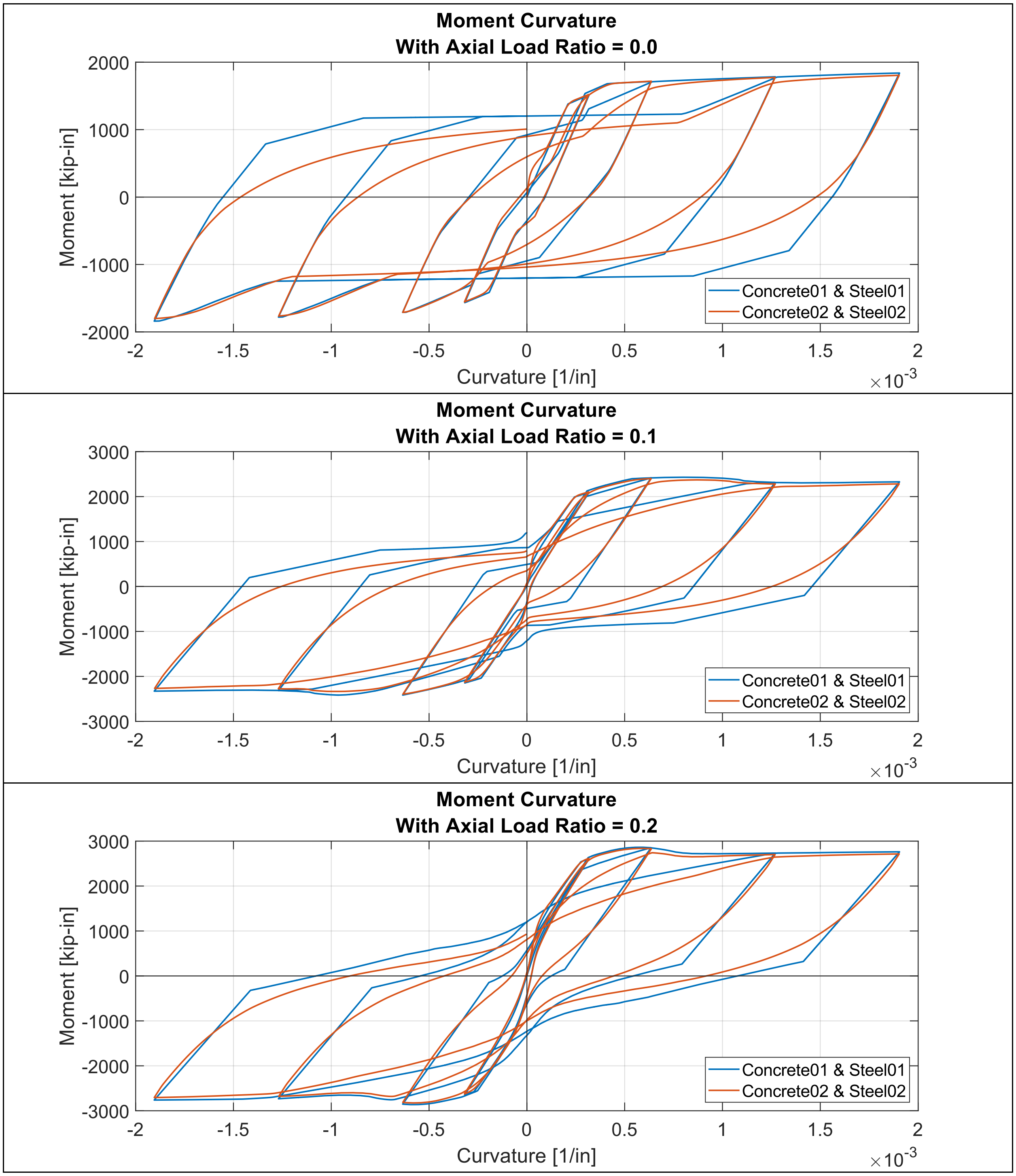


Figure 14 Axial Strain Response for Various Axial Loads

Part (g) Examining the use of Different Constitutive Models on Cyclic Moment Curvature

The two sets of constitutive models used in this report show different cyclic moment curvature behavior, but still converge to the same capacities for curvatures as shown in Figure 15. One clear difference is the yielding of section is modeled better with STEEL02 and the unloading of concrete in CONCRETE02. The pinching of the curve is evident in both plots as the discretization is the same and the concrete still cracks.



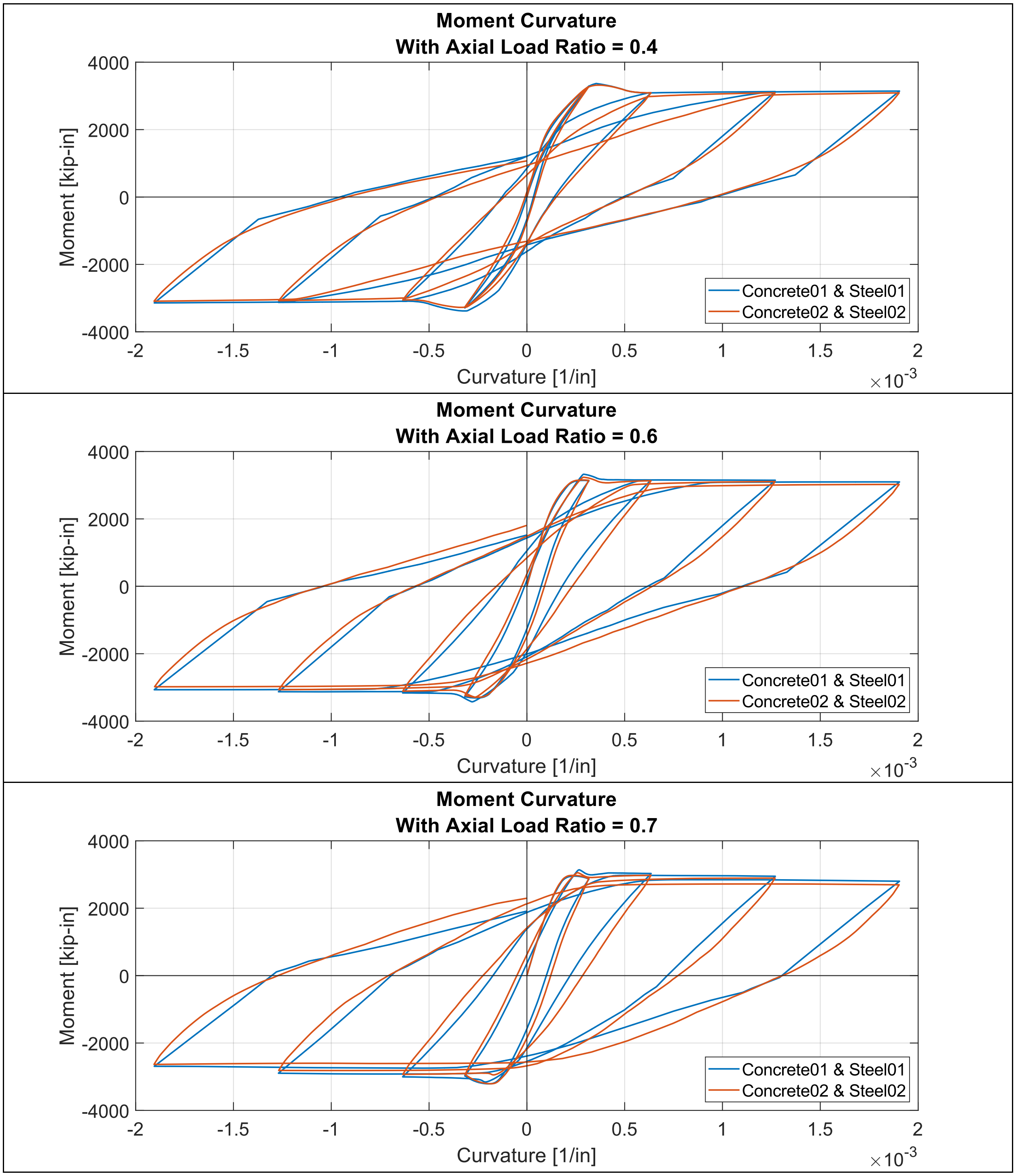


Figure 15 Comparing Cyclic Moment Curvature of Various Constitutive Models

While it is thought that the computational time of the second set of constitutive model may be higher, the first set actually took longer to converged. This may be due to the sharp changes in the constitutive model of CONCRETE01 and STEEL01. The curvature steps had to be significantly lowered in order to for the model to converge with the first set of models. It is noted that while the behavior of the section is better captured with the second set of constitutive models, since the capacity is the sole focus of design, CONCRETE01 and STEEL01 will suffice in most applications.

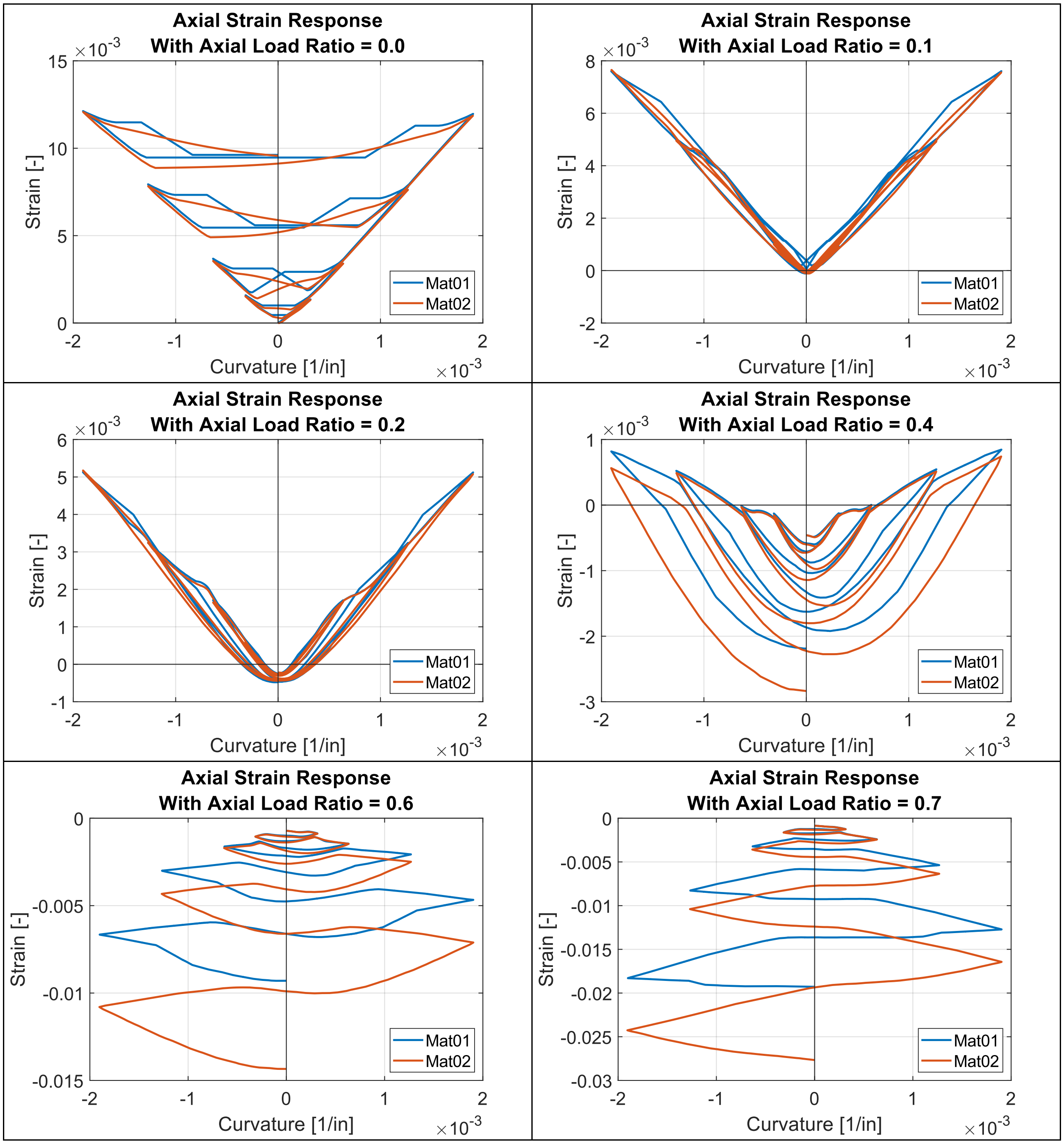


Figure 16 Axial Strain Response of Various Constitutive Models

In Figure 16 the axial strain of the centroidal fiber is shown, where Mat01 represents the model that used CONCRETE01 and STEEL01 constitutive models, while Mat02 used CONCRETE02 and STEEL02. The axial strain responses for the two models are nearly identical except for high axial load cases. This could be caused by the unloading schemes adopted by each concrete constitutive model. While CONCRETE01 unloads linearly, CONCRETE02 unloads bilinearly which means that there will be higher residual axial strain as the model undergoes more cyclic strain time histories. The difference is that at higher axial loads, the concrete would have crushed which means that the constitutive model is unloading from a higher compressive strain which results in higher discrepancy in unloading behavior. This residual axial strain is then propagated through subsequent cycles.

Part (h) Comparison with Experimental Data

The experimental moment curvature of a column cross section from X. Li's PhD thesis was provided and a model was created in Opensees in order to model its behavior. In the experiment a constant force was applied to the column while a curvature time history was applied as shown in Figure 17. The minimum and maximum curvatures applied to the real column were matched in the model, but each curvature was applied as single iteration steps as shown rather than the steps as measured.

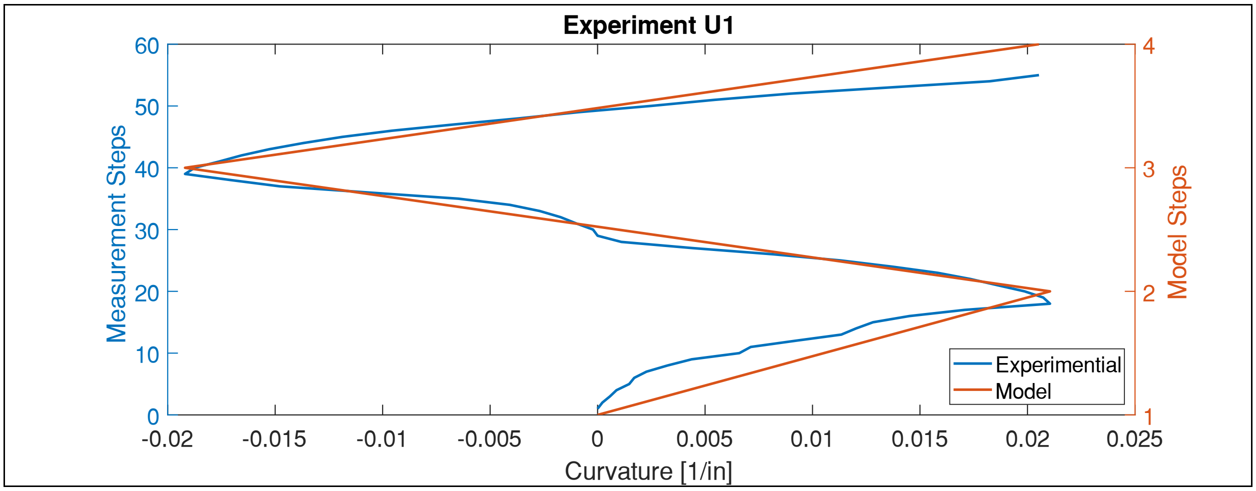


Figure 17 Curvatures used in Modeling X. Li's Cyclic Moment Curvature Diagram

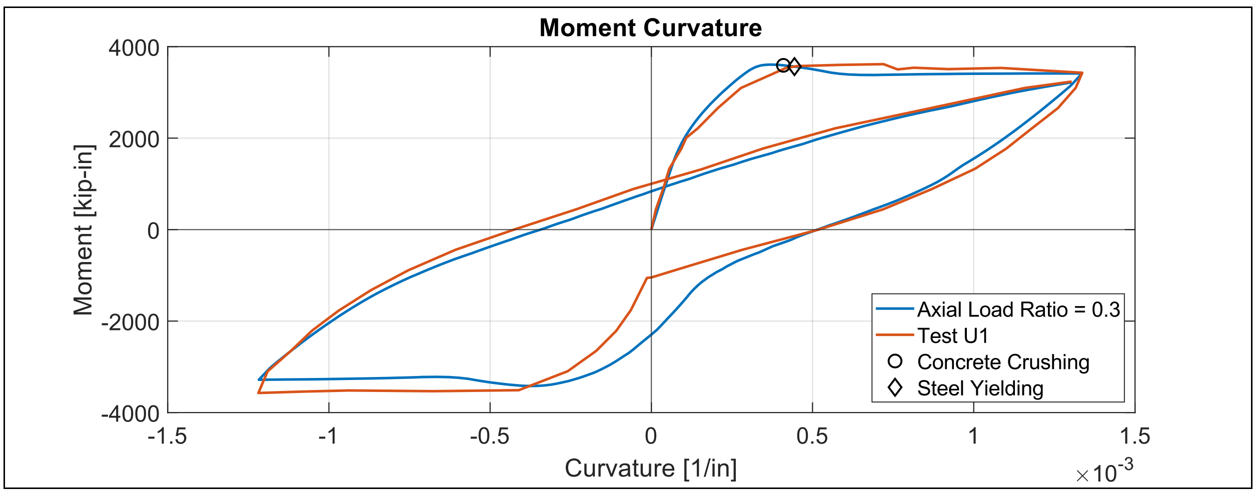


Figure 18 Comparing Experimental and Model Cyclic Moment Curvature

The model's behavior is shown to be remarkably close to the actual test results in Figure 18, differing slightly in the ultimate capacity reached in either direction as well as the behavior of pinching. There were assumptions made in modeling the material plasticity thus the failure mechanism like yielding and cracking of the actual concrete and steel reinforcement may be different from that of the model. The material properties used in the model were matched with those measured in the actual thesis as shown in Table 4.The modulus of the concrete was not reported and was assumed to be $4700\sqrt{f_c'}$ in MPa. Other parameters like the ultimate capacity and strain were also assumed from Mander et al. (1988) and Scott et al. (1982). The thesis did report the compressive strength of the base of the column, and this was used as the compressive strength of the confined concrete. The tensile elastic modulus was taken as $0.01E_c$. The isotropic hardening parameter for STEEl02 were also assumed as well as the softening parameters from Filippou (1983).

After accounting for differences due to material modeling, the elastic region matched the experimental results extremely well. The slopes are nearly identical, however differed after the steel yielded which is most likely due to the improper parameters used to model the yielding of steel. The unloading of the structure shows a larger pinching as the cracks in the real structure make have been larger than the one in predicted in the model. This could be that the tensile capacity of the concrete is less than what is in the model and thus a larger crack formed than expected. Additionally, the bilinear assuming of concrete unloading may play a role in this pinching behavior. The capacity at the end of the cyclic moment curvature diagram reached by the model was 3217.81 kip-in while the experimentally the capacity ended at 3236.56 kip-in which gives a 0.5% error. Given all of the other factors that could have taken place this model is fairly accurate.

Table 4 Parameters Used in CONCRETE02 for Modeling the Confined and Unconfined Concrete

Opensees Parameters	Units	Unconfined Concrete		Confined Concrete	
$fpc	[MPa]	f_c'	-33.2	f_{cc}'	-53.7
$E	[MPa]	E_c	27,000	E_c	27,000
$epsc0	[-]	ϵ'_c	$2f_c'/E_c$	ϵ'_{cc}	$2f_{cc}'/E_{cc}$
$ft	[MPa]	f_{cr}'	4.8	f_{cr}'	4.8
$lambda	[-]	λ	0.25	λ	0.25
$Ets	[Mpa]		$0.01E_c$		$0.01E_c$
$fpcU	[Mpa]	f_{cu}'	$0.2f_c'$	f_{ccu}'	$0.2f_{cc}'$
$epsU	[-]	ϵ_{cu}'	-0.004	ϵ_{ccu}'	-0.0276

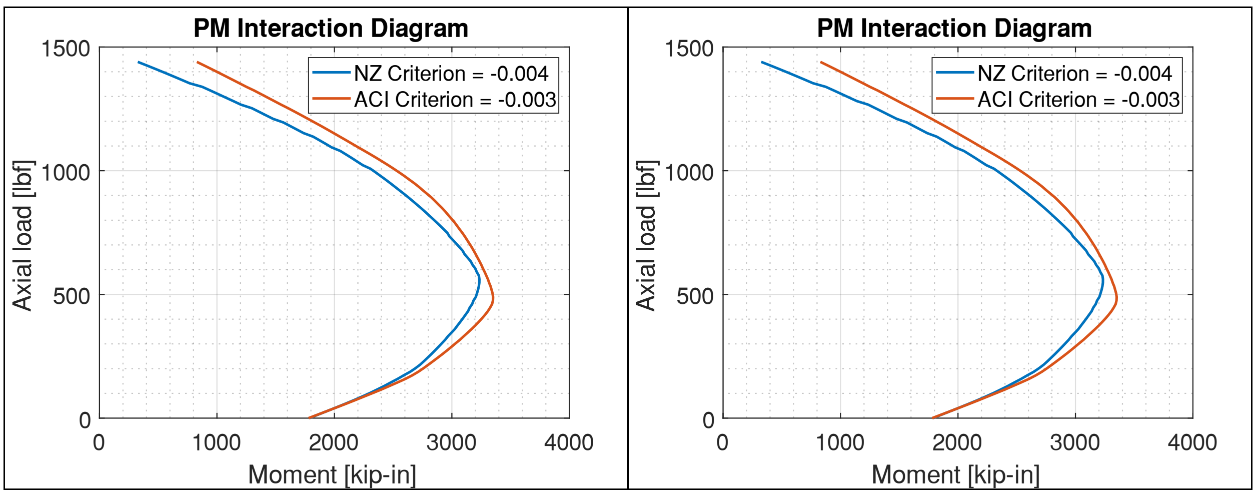
Table 5 Parameters Used in STEEL02 for Modeling the Steel Reinforcement in X.Li's Experiment

Opensees Parameters	Units	Reinforcing Steel	
$Fy	[MPa]	F_y	455
$E	[MPa]	E_s	215,000.
$b	[-]	b	0.01
$R0	[-]	R_0	20
$cR1	[-]	cR_1	0.925
$cR2	[-]	cR_2	0.15
$a1	[-]	a_1	0
$a2	[-]	a_2	1
$a3	[-]	a_3	0
$a4	[-]	a_4	1
$sigInit		-	0

Part (i) Axial Moment Interaction Diagram

An axial-moment interaction diagram is generated by taking the moment capacity from the moment curvature diagram where the extreme compressive concrete fiber reaches a certain strain criterion and plotting it against the applied axial load. This strain criterion is prescribed as -0.003 by the American Concrete Institute, while this criterion is $-.004$ for the New Zealand Standards. Ss shown in Figure 19 the difference is small for tension-controlled failure and higher for compression-controlled failures. This is due to the effect of the crushing of concrete on the moment capacity for compression-controlled sections. The sudden decrease in strength is better captured with the higher strain criterion. There is a capacity penalty for designing a beam that is compression controlled.

The shape of the PM interaction diagram is explained by the movement of the neutral axis. Although the axial load is applied across the entire cross section, the resultant of the force acts at the centroid of the section. While the neutral axis is above the centroid, the axial load gives the cross section additional moment capacity since the axial load induces a negative moment on the section. But as the axial load increases and the neutral axis shifts downwards, the capacity of the beam decreases because now the axial load is adding positive moment, reducing the capacity of the section. Up the balanced point, the axial load adds to the section's moment capacity, and then subtracts from it afterwards.



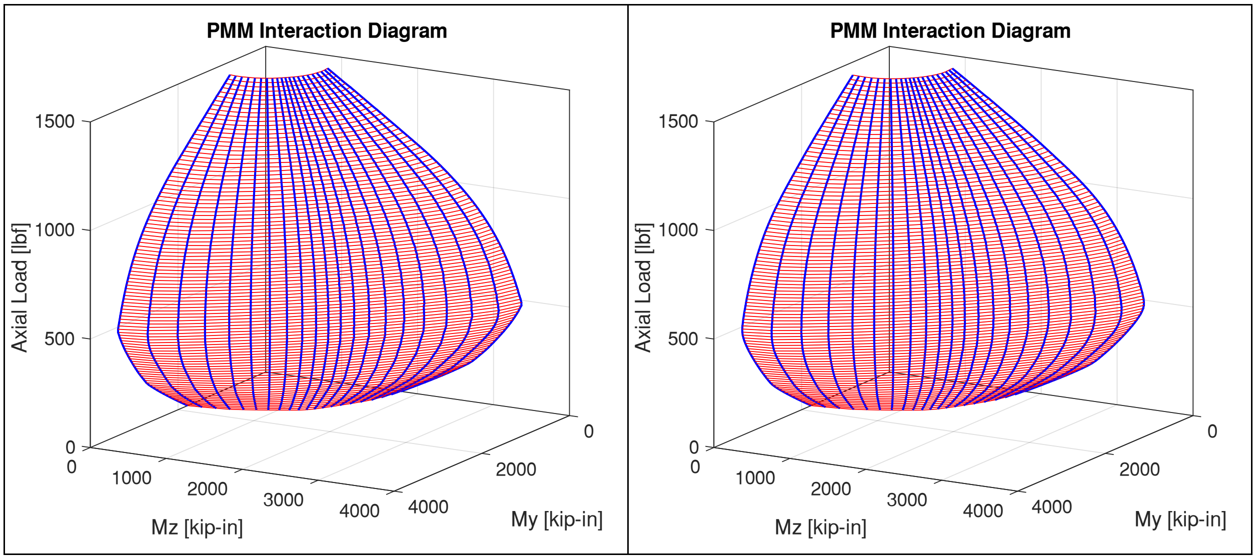
(a) Using CONCRETE01 and STEEL01 (b) Using CONCRETE02 and STEEL02

Figure 19 Comprison of Axial-Moment Interaction Diagrams for Various Constitutive Models

The usage of CONCRETE01 and STEEL01 versus CONCRETE02 and STEEL02 is minimal. The ultimate capacity reached by either constitutive model is the same for the ACI criterion, $M = 3{,}349.86$ kip-in and for the NZS criterion, $M = 3234.25$ kip-in.

A section may undergo bending about both its axes and thus a biaxial-moment interaction diagram is necessary. This is generated by applying a combination of moments in both directions on the cross-section and then plotting the moment developed by the section for the point reached of a certain criterion, as before. This is shown in Figure 20 for the section given previously. The main difference for this analysis is the need to discretize the section in both directions. This is shown in Figure 21, in total there being 50 fibers in each direction.

The difference between using CONCRETE01 and STEEL01 versus CONCRETE02 and STEEL02 is minimal.



(a) Using CONCRETE01 and STEEL01 (b) Using CONCRETE02 and STEEL02

Figure 20 Bi-axial Moment Interaction Diagram for Various Constitutive Models

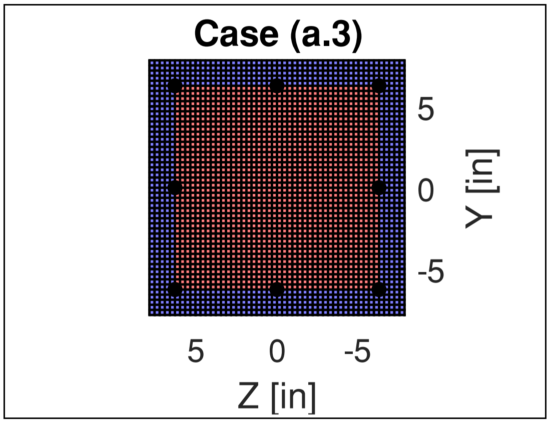


Figure 21 Discretization of the Cross Section for Bi-axial Moment Analysis

References

Li, X. (1994). *Reinforced Concrete Columns Under Seismic Lateral Force and Varying Axial Load* (Ph.D.). University of Canterbury.

Mander, J. (1988). Observed Stress-Strain Behavior of Confined Concrete. *Structural Engineering*, *114*(8). Retrieved 4 March 2021

Appendix A. MATLAB Code

```
openseesPath = "C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 201B\Opensees\bin\OpenSees.exe";

%% Monotonic with Single Axial Load
clc; close all;
RunName = '.\run.tcl';
loadControlName = 'loadControlStaticAnalysis.tcl';
analysisName = 'dispControlAnalysis.tcl';
modelName = 'model02.tcl';
curvatures = 0.03;
axialLoad = 0.0;

setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures)
runTCLfile(RunName)
plotMK("Case (a.3) Cross Section"); grid minor;
[s1, s2] = addYield();
s1.HandleVisibility = 'on'; s1.DisplayName = 'Concrete Crushing';
s2.HandleVisibility = 'on'; s2.DisplayName = 'Steel Yielding';
print_figure("Part B MK", [13, 6], 18)

%% Monotonic with Varying Axial Load
clc; close all;
RunName = '.\run.tcl';
loadControlName = 'loadControlStaticAnalysis.tcl';
analysisName = 'dispControlAnalysis.tcl';
modelName = 'model01.tcl';
curvatures = 0.03;

for axialLoad = [0.0, 0.1, 0.2, 0.4, 0.6, 0.7]
    close all;
    setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures)
    runTCLfile(RunName)
    plotMK("Axial Load Ratio = "+sprintf('%1.1f',axialLoad) );
    addYield();
    H = 15.748;
    K = [0.0013, 0.004, 0.008, 0.0130 0.0200]/H;
    plotStrainProfile(H, K)
    sgtitle("Strain Diagram of Axial Load Ratio "+num2str(axialLoad,'%1.1f'),'FontSize',18);
    print_figure("Part E iv Strain Diagram of Axial Load Ratio "+num2str(10*axialLoad), [13, 2.75], 10)
end
figure(1);
print_figure("Part E Moment Curvature Varying Axial Load", [13, 6], 18)
%% Monotonic & Cyclic Envelope
clc; close all;
RunName = '.\run.tcl';
loadControlName = 'loadControlStaticAnalysis.tcl';
analysisName = 'dispControlAnalysis.tcl';
modelName = 'model02.tcl';

for axialLoad = [0.0, 0.1, 0.2, 0.4, 0.6, 0.7]
    curvatures = [0.0, 0.005, -0.005, 0.01, -0.01, 0.02, -0.02, 0.03, -0.03, 0.0];
    setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures, 0.001)
    runTCLfile(RunName)

    plotMK('Cyclic Moment Curvature', "Axial Load Ratio = "+sprintf('%1.1f',axialLoad) );
    addYield();
    hold on;
    curvatures = 0.03;
```

```
    setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures, 0.001)
    runTCLfile(RunName)
    plotMK('Monotonic Moment Curvature',"Axial Load Ratio = "+sprintf('%1.1f',axialLoad) );
    addYield();

    figure(1);
    print_figure("Part L Moment Curvature Envelope Axial Load Ratio = "+10*axialLoad , [13,5], 18)
    close all;
end

%% Cyclic Moment Curvautre With Material 02
clc; close all;
RunName = '.\run.tcl';
loadControlName = 'loadControlStaticAnalysis.tcl';
analysisName = 'dispControlAnalysis.tcl';
modelName = 'model02.tcl';
curvatures = [0.0, 0.005, -0.005, 0.01, -0.01, 0.02, -0.02, 0.03, -0.03, 0.0];

for axialLoad = [0.0, 0.1, 0.2, 0.4, 0.6, 0.7]
    setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures, 0.001)
    runTCLfile(RunName)
    close all;
    plotMK("Axial Load Ratio = "+sprintf('%1.1f',axialLoad) );
    print_figure("Part F i Cyclic Moment Curvature"+num2str(10*axialLoad), [13, 5], 18)

    plotEoK("Axial Load Ratio = "+sprintf('%1.1f',axialLoad),"Axial Strain Response");
    figure(2); print_figure("Part F ii Axial Strain Axial Load Ratio "+num2str(10*axialLoad), [7,5], 15)
end


%% Cyclic Moment Curvautre With Material 01
clc; close all;
RunName = '.\run.tcl';
loadControlName = 'loadControlStaticAnalysis.tcl';
analysisName = 'dispControlAnalysis.tcl';
% modelName = 'model01.tcl';
curvatures = [0.0, 0.005, -0.005, 0.01, -0.01, 0.02, -0.02, 0.03, -0.03, 0.0];
tic
for axialLoad = [0.0]
      close all;
      modelName = 'model01.tcl';
      setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures, 0.00001)
       runTCLfile(RunName)
       plotMK('Concrete01 & Steel01',["Moment Curvature", "With Axial Load Ratio = "+sprintf('%1.1f',axialLoad)]);
       plotEoK('Mat01',["Axial Strain Response","With Axial Load Ratio = "+num2str(axialLoad,'%1.1f')]);

       modelName = 'model02.tcl';
       setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures, 0.001)
       runTCLfile(RunName)
       plotMK('Concrete02 & Steel02',["Moment Curvature", "With Axial Load Ratio = "+sprintf('%1.1f',axialLoad)]);
       plotEoK('Mat02',["Axial Strain Response","With Axial Load Ratio = "+num2str(axialLoad,'%1.1f')]);

       figure(1); print_figure("Part G Cyclic MK Axial Load Ratio "+num2str(10*axialLoad), [13,5], 18);
       savefig(1,"../figures/"+"Part G Cyclic MK Axial Load Ratio "+num2str(10*axialLoad),'compact')
       figure(2); print_figure("Part G Axial Strain Axial Load "+num2str(10*axialLoad), [7,5], 18)
       savefig(2,"../figures/"+"Part G Axial Strain Axial Load "+num2str(10*axialLoad),'compact')
end
```

```matlab
toc
%% X. Li Test 1
clc; close all;
[K1, M1] = getXLisheet('Response U1', "B10:F64");

RunName = '.\run.tcl';
loadControlName = 'loadControlStaticAnalysis.tcl';
analysisName = 'dispControlAnalysis.tcl';
modelName = 'modelLI1.tcl';

curvatures1 = [0.0, 0.0210457, -0.019193, 0.0205331];
for axialLoad = .2953
    setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures1,0.001)
    runTCLfile(RunName);
    plt = plotMK("4 Step Curvatures"); plt.LineWidth =2;
end

plot(K1/15.748,M1*12,'DisplayName','Test U1','LineWidth',2);
[s1, s2] = addYield();
s1.HandleVisibility = 'on'; s1.DisplayName = 'Concrete Crushing';
s2.HandleVisibility = 'on'; s2.DisplayName = 'Steel Yielding';
set(gcf,'Position',[0,0,1300,500])
figure(1); print_figure("Part H X Li Test 1", [13,5], 18);
%% Curvature of X. Li
clc; close;

figure(5); hold on; box on;
K = K1;
yyaxis left;
plot(K,1:length(K),'LineWidth',2,'DisplayName','Experimential');
ylabel('Measurement Steps'); xlabel('Curvature [1/in]');
yyaxis right; ylabel('Model Steps'); yticks(0:1:length(curvatures1));
plot(curvatures1,1:length(curvatures1),'LineWidth',2,'DisplayName','Model');
title('Experiment U1'); legend('Location','southeast');
set(gca,'FontSize',18);

print_figure("Part H Curvature", [13,5], 18);
%% PM Diagrams for Concrete01 and Steel01
close all; clc;

runTCLfile('.\runPM_NZ_01.tcl');
PlotPM(6, 'NZ Criterion = -0.004' );
runTCLfile('.\runPM_ACI_01.tcl');
PlotPM(6, 'ACI Criterion = -0.003' );
grid minor
print_figure("Part I i PM Mat01", [6.5,5], 18);
%% PM Diagrams for Concrete02 and Steel02
close all; clc;
runTCLfile('.\runPM_NZ_02.tcl');
PlotPM(7, 'NZ Criterion = -0.004' );
runTCLfile('.\runPM_ACI_02.tcl');
PlotPM(7, 'ACI Criterion = -0.003' );
grid minor
print_figure("Part I i PM Mat02", [6.5,5], 18);
%% PMM Diagram
close all; clc;
hold on;
```

```
runTCLfile('.\runPMM.tcl')
plotPMM(8)
print_figure("Part I ii PMM", [8,7], 18);

%% FUNCTION Runs a TCL File
function runTCLfile(filename)
  openseesPath = "C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 201B\Opensees\bin\OpenSees.exe";
  [filepath,name,ext] = fileparts(filename);
  [filepath,name,ext,openseesPath] = convertStringsToChars(filepath,name,ext,openseesPath);
  currpath = pwd;
  cd(filepath)
  system(['"',openseesPath,'" "' name, ext, '"']);
  cd(currpath);
end
%% FUNCTION Plot Moment Curvature
function plt = plotMK(legName, titleName)
  arguments
    legName = 'Moment Curvature';
    titleName = 'Moment Curvature'
  end
  figure(1); hold on;
  load('./AnalysisResults/MK.txt');
  plt = plot(MK(:,3),MK(:,1),'LineWidth',2,"DisplayName",legName);
  grid on; box on;
  title(titleName); ylabel('Moment [kip-in]'); xlabel('Curvature [1/in]');
  legend("Location","Southeast"); xline(0,"HandleVisibility",'off'); yline(0,"HandleVisibility",'off');
end

%%  FUNCTION Adds in Crushing and Yielding of Steel
function [scat1, scat2] = addYield(~)
  fy = 65.9;
  ecu = -0.003;
  load('./AnalysisResults/ConcFib2_SS.txt');
  load('./AnalysisResults/SteelFib1_SS.txt');
  load('./AnalysisResults/MK.txt');
  [~, yield] = min(abs(SteelFib1_SS(:,2) - fy));
  [~, crush] = min(abs(ConcFib2_SS(:,3) - ecu));

  scat1 = plot(MK(crush,3), MK(crush,1),'ko','MarkerSize',10, 'LineWidth', 1.5,'HandleVisibility','off');
  scat2 = plot(MK(yield,3), MK(yield,1),'kd','MarkerSize',10,'LineWidth',1.5,'HandleVisibility','off');
end

%% FUNCTION  Plot Strain Profile
function [] = plotStrainProfile(H, K)
  close;
  figure(4); set(gcf,'DefaultAxesFontSize', 10)
  load('./AnalysisResults/MK.txt');
  sectionCoord = linspace(-H/2,H/2,100);
  plotYLim = [0,0];
  for i = 1:length(K)
    subplot(1,length(K),i); hold on; box on
    [~, indK] = min(abs(MK(:,3) - K(i)));
    eps = MK(indK,2) - sectionCoord*K(i);
    plot(sectionCoord,eps,'LineWidth',1.,'Color','k')
    plot(sectionCoord,zeros(size(sectionCoord)),'b-','LineWidth',2);
    plot([sectionCoord(1),sectionCoord(1)], [0, eps(1)], 'LineWidth',1.,'Color','k')
    plot([sectionCoord(end),sectionCoord(end)], [0, eps(end)], 'LineWidth',1.,'Color','k')
```

```
        yLim = get(gca,'yLim');
        if yLim(1) < plotYLim(1)
            plotYLim(1) = yLim(1);
        end
        if yLim(2) > plotYLim(2)
            plotYLim(2) = yLim(2);
        end
        [~,indNA] = min(abs(eps - 0));
        depthNA = H/2 - sectionCoord(indNA);
        plot([sectionCoord(indNA),sectionCoord(end)],[0,0],'r-','LineWidth',2);
        title(["NA = " + num2str(depthNA,3)],'FontSize',12)
    end
    figure(4)
    for i = 1:length(K)
        subplot(1,length(K),i)
        set(gca,'yLim',plotYLim)
        xline(0);
    end
end

%% FUNCTION Plot Axial Strain Curvature
function plt = plotEoK(displayName, titleName)
    arguments
        displayName = 'Axial Strain Response';
        titleName = 'Axial Strain Response';
    end
    figure(2); hold on;
    load('./AnalysisResults/MK.txt');
    plt = plot(MK(:,3),MK(:,2),'LineWidth',2,'DisplayName',displayName);
    grid on; box on;
    title(titleName); ylabel('Strain [-]'); xlabel('Curvature [1/in]'); legend('Location','southeast');
     xline(0,"HandleVisibility",'off'); yline(0,"HandleVisibility",'off');
end
%% FUNCTION Get X. Li's
function [K,M] = getXLisheet(sheet, range)
    opts = spreadsheetImportOptions("NumVariables", 5);
    opts.Sheet = sheet;
    opts.DataRange = range;
    opts.VariableNames = ["K", " "," "," ", "M"];
    opts.SelectedVariableNames = ["K", "M"];
    opts.VariableTypes = ["double", "char", "char", "char", "double"];
    tbl = readtable("C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 201B\HW\HW3\files\XLi Units 1 and 2.xlsx", opts, "UseExcel", false);
    K = tbl.K;
    M = tbl.M;
end
%% FUNCTION Plot PM Diagram
function plt = PlotPM(figNum, displayName)
    load 'AnalysisResults/PM_Results.txt';
    figure(figNum);  hold on;
    plt = plot(PM_Results(:,2),PM_Results(:,1),'LineWidth',2,'DisplayName',displayName);
    grid on; box on; legend();
    xlabel('Moment [kip-in]')
    ylabel('Axial load [lbf]')
    title('PM Interaction Diagram')
    set(gcf,'Position',[0,0,600,500])
end
%% FUNCTION Plot PMM Diagram
```

```
function plotPMM(figNum)
    name='AnalysisResults/PM';

    numFiles = 21;
    for iFile = 1:numFiles
        fileLoad = [name num2str(iFile) '.txt'];
        mp{iFile} = load(fileLoad);
    end

    figure(figNum)
    for iFile = 1:21
        if iFile == 1
            h1 = plot3(mp{iFile}(:,11),mp{iFile}(:,12),mp{iFile}(:,1),'b','LineWidth',2);hold on
        else
            plot3(mp{iFile}(:,11),mp{iFile}(:,12),mp{iFile}(:,1),'b','LineWidth',2);hold on
        end
    end
    for ii = 1:size(mp{iFile}(:,11),1)
        for iFile = 1:20
            plot3(...
                [mp{iFile}(ii,11) mp{iFile+1}(ii,11)], ...
                [mp{iFile}(ii,12) mp{iFile+1}(ii,12)],...
                [mp{iFile}(ii,1) mp{iFile+1}(ii,1)],...
                'r','LineWidth',0.5);hold on
        end
    end

    grid on
    box on

    title('PMM Interaction Diagram')
    xlabel('My [kip-in]')
    ylabel('Mz [kip-in]')
    zlabel('Axial Load [lbf]')
    view([120 15]);
end
%% FUNCTION Print Section
function printSectionTag(tag, name)
    secDefFilePath = "./Model/modelData.txt";
    grid off;
    figNum = 3;
    secTag = tag;
    camroll(90)
    fibColor = [1 0 0 1 0.5; 2 1 0 0 0.5; 3 0 0 0 1];
    plot_fiberSection(secDefFilePath, secTag, figNum, fibColor)
    camroll(90)
    title(["Case (a." + secTag+")"]);
    ylabel('Y [in]');
    zlabel('Z [in]');
end
```

```matlab
function [ ] = setupRun(RunName, loadControlName, analysisName, modelName, axialLoad, curvatures, stepScale)
% Creates the TCL files parametrically for the moment curvature analysis
arguments
    RunName = 'run.tcl';
    loadControlName = 'loadControlStaticAnalysis.tcl';
    analysisName = 'dispControlAnalysis.tcl'
    modelName = 'model01.tcl';
    axialLoad = 0.0;
    curvatures = 0.03;
     stepScale = 0.0001;
end

 FID = fopen(RunName, 'w');
 fprintf(FID,[...
    'set analysisResultsDirectory "AnalysisResults";\n',...
    'file mkdir $analysisResultsDirectory;\n',...
    'set modelDirectory "Model";\n',...
    'file mkdir $modelDirectory;\n',...
    'set modelExportFileID [open "$modelDirectory/modelData.txt" "w"];\n',...
    'source units.tcl;\n',...
    'source ' modelName ';\n',...
    'source ' loadControlName ';\n',...
    'source ' analysisName ';\n',...
    'remove recorders\n',...
    'wipe;\n']);
 fclose(FID);

 FID = fopen(loadControlName, 'w');
    fprintf(FID,...
    ['set axialLoadTag 1;\n',...
    'set axialLoadRatio ' num2str(axialLoad,'%1.2f') ';\n',...
    'set P [expr abs($fpc)*$colArea*$axialLoadRatio];\n',...
    'pattern Plain $axialLoadTag "Linear" { load $controlNode -$P 0.0 0.0;};\n',...
    'set numAnalysisSteps 1 ;\n',...
    'integrator LoadControl [expr 1./$numAnalysisSteps];\n',...
    'system BandGeneral ;\n',...
    'test NormUnbalance 1e-6 100;\n',...
    'numberer Plain ;\n',...
    'constraints Plain ;\n',...
    'algorithm KrylovNewton ;\n',...
    'analysis Static;\n',...
    'set ok [analyze $numAnalysisSteps];\n',...
    'if {$ok == 0} {puts "Axial load applied and analyzed"};\n',...
    'loadConst -time 0.0']);
    fclose(FID);


    appendage = [];
    for k = curvatures
        appendage = [appendage, 'lappend peakDisp [expr ', num2str(k,"%1.5f"), '/$colDepth]' , newline];
    end

    FID = fopen(analysisName, 'w');
    fprintf(FID,[...
    'set controlDOF 3',newline,...
    'set dispControlLoadTag 2 ', newline,...
    '# Define reference moment', newline,...
```

```
'pattern Plain $dispControlLoadTag "Linear" {load $controlNode 0.0 0.0 1.0}', newline,...
'set peakDisp {};', newline,...
appendage,...
'set numCycles [llength $peakDisp]', newline,...
'set cyclelabel {};', newline,...
'for {set i 1} {$i <= $numCycles} {incr i 1} {', newline,...
'    lappend cyclelabel $i', newline...
'}', newline,...
'set maxDisp [expr 0.03/$colDepth];', newline,...
'set du [expr ' num2str(stepScale) '*$maxDisp];', newline,...
'set ok 0; ', newline,...
'set currentDisp 0.; # This is the current value of the displacement at the control DOF.', newline,...
'set tol 1e-6;', newline,...
'set iter 250', newline,...
'recorder Node -file $analysisResultsDirectory/MK.txt -time -node $controlNode -dof 1 $controlDOF disp', newline,...
'recorder Element -file $analysisResultsDirectory/ConcFib1_SS.txt  -time -ele 1 section fiber -$y1  0. $matTagConcCover stressStrain', newline,...
'recorder Element -file $analysisResultsDirectory/ConcFib2_SS.txt  -time -ele 1 section fiber  $y1  0. $matTagConcCover stressStrain', newline,...
'recorder Element -file $analysisResultsDirectory/SteelFib1_SS.txt -time -ele 1 section fiber -$y1  0. $matTagSteel     stressStrain', newline,...
'recorder Element -file $analysisResultsDirectory/SteelFib2_SS.txt -time -ele 1 section fiber  $y1  0. $matTagSteel     stressStrain', newline,...
'record; # This is to record the state before the analysis starts', newline,...
'for {set ii 1} {$ii<=[llength $peakDisp]} {incr ii} {', newline,...
    '# Convergence check', newline,...
    'if {$ok == 0} { ', newline,...
        'set cycleDisp [expr [lindex $peakDisp [expr $ii-1]] - $currentDisp]; # the total deformation of the loading cycle', newline,...
        '# determine the sign of loading:', newline,...
        'if {$cycleDisp>0} {', newline,...
        '    set sign 1.;', newline,...
        '} else {', newline,...
        '    set sign -1.;', newline,...
        '};', newline,...
        'set dU [expr $du*$sign];', newline,...
        '# General analysis properties', newline,...
        'constraints Transformation;        ', newline,...
        'numberer Plain;           ', newline,...
        'system BandGeneral;', newline,...
        'integrator DisplacementControl $controlNode $controlDOF $dU;', newline,...
        'test RelativeNormDispIncr $tol $iter;', newline,...
        'algorithm KrylovNewton;', newline,...
        'analysis Static;', newline,...
        'set NSteps [expr int(abs($cycleDisp/$dU))];', newline,...
        'puts "";', newline,...
        'puts "Starting Cycle # [lindex $cyclelabel [expr $ii-1]] with target displacement of [expr [lindex $peakDisp [expr $ii-1]]]"', newline,...
        'puts "======================================================"', newline,...
        'puts "---> Running $NSteps steps with step size = $dU in. to go from displ. = $currentDisp to displ. = [expr [lindex $peakDisp [expr $ii-1]]]"', newline,...
        'set ok1 [analyze $NSteps];', newline,...
        'set currentDisp [nodeDisp $controlNode $controlDOF];', newline,...
        '#If it does not converge, change strategies', newline,...
        'if {$ok1 !=0 } {', newline,...
            'set ok 0;', newline,...
            'puts "    Try stuff, peak disp = [expr [lindex $peakDisp [expr $ii-1]]]";', newline,...
            'puts "    Current disp = $currentDisp";', newline,...
            'puts "    Cycle disp  = $cycleDisp";', newline,...
```

```
        'set counter 1;', newline,...
        'while {( ( ([expr $currentDisp] <= [expr [lindex $peakDisp [expr $ii-1]]]) && ($sign == 1)  ) || (  ([expr $currentDisp] >= [expr [lindex $peakDisp [expr $ii-1]]]) && ($sign == -1)   ) )&&($ok==0)} {', newline,...
        '   set ok 1;', newline,...
        '   while {$ok!=0} {', newline,...
        '       if {$counter == 0} {', newline,...
        '           # return to initial conditions', newline,...
        '           set dU [expr $du*$sign*1.00];', newline,...
        '           test NormDispIncr $tol $iter 0;', newline,...
        '           set counter 1;', newline,...
        '       } elseif {$counter == 1} {', newline,...
        '           # increase load stepsize', newline,...
        '           set dU [expr $du*$sign*1.5];', newline,...
        '           #puts "dU = $du*$sign*1.5 = $dU";', newline,...
        '           set counter 2;', newline,...
        '       } elseif {$counter == 2} {', newline,...
        '           # increase load stepsize', newline,...
        '           set dU [expr $du*$sign*2.00];', newline,...
        '           #puts "dU = $du*$sign*2.0 = $dU";', newline,...
        '           set counter 3;', newline,...
        '       } elseif {$counter == 3} {', newline,...
        '           # decrease load stepsize', newline,...
        '           set dU [expr $du*$sign*0.5];', newline,...
        '           #puts "dU = $du*$sign*0.5 = $dU";', newline,...
        '           set counter 4;', newline,...
        '       } elseif {$counter == 4} {', newline,...
        '           # decrease load stepsize', newline,...
        '           set dU [expr $du*$sign*0.1];', newline,...
        '           #puts "dU = $du*$sign*0.1 = $dU";', newline,...
        '           set counter 5;', newline,...
        '       } elseif {$counter == 5} {', newline,...
        '           # decrease load stepsize', newline,...
        '           set dU [expr $du*$sign*0.05];', newline,...
        '           #puts "dU = $du*$sign*0.05 = $dU";', newline,...
        '           set counter 6;', newline,...
        '       } elseif {$counter == 6} {', newline,...
        '           # decrease load stepsize', newline,...
        '           set dU [expr $du*$sign*0.01];', newline,...
        '           #puts "dU = $du*$sign*0.01 = $dU";', newline,...
        '           set counter 7;', newline,...
        '       } elseif {$counter == 7} {', newline,...
        '           # decrease load stepsize', newline,...
        '           set dU [expr $du*$sign*0.001];', newline,...
        '           #puts "dU = $du*$sign*0.001 = $dU";', newline,...
        '           set counter 8;', newline,...
        '       };', newline,...
        '       integrator DisplacementControl $controlNode  $controlDOF  $dU;', newline,...
        '       set ok [analyze 1]', newline,...
        '       if {$ok != 0} {', newline,...
        '           puts "Try Newton Initial"', newline,...
        '           algorithm Newton -initial', newline,...
        '           test NormDispIncr $tol $iter 0;', newline,...
        '           set ok [analyze 1];', newline,...
        '       };', newline,...
        '       if {$ok != 0} {', newline,...
        '           puts "Test Relative Displacement"', newline,...
        '           test RelativeNormDispIncr $tol $iter 0;', newline,...
```

```
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "ModifiedNewton"', newline,...
        '            algorithm ModifiedNewton', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "Test Relative Energy"', newline,...
        '            algorithm Newton -initial;', newline,...
        '            test RelativeEnergyIncr $tol $iter 0;', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "Newton Modified -initial"', newline,...
        '            algorithm ModifiedNewton -initial', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "Test Relative Force"', newline,...
        '            algorithm Newton -initial;', newline,...
        '            test RelativeNormUnbalance $tol $iter 0', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "Test Relative Displ"', newline,...
        '            test RelativeNormDispIncr $tol $iter 0', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "Broyden"', newline,...
        '            algorithm Broyden 8', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "Newton Line Search"', newline,...
        '            algorithm NewtonLineSearch .8', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '        if {$ok != 0} {', newline,...
        '            puts "BFGS"', newline,...
        '            algorithm BFGS', newline,...
        '            set ok [analyze 1];', newline,...
        '        };', newline,...
        '    };', newline,...
        '    set counter 0;', newline,...
        '    set currentDisp [nodeDisp $controlNode $controlDOF];', newline,...
        '    #puts $currentDisp', newline,...
        '};', newline,...
        'puts "Cycle # [lindex $cyclelabel [expr $ii-1]] successfully finished!"', newline,...
        'puts "target displ.  = [expr [lindex $peakDisp [expr $ii-1]]]"', newline,...
        'puts "current displ. = [nodeDisp $controlNode $controlDOF]"', newline,...
        'puts "------------------------------x------------------------------------";', newline,...
    '} else {', newline,...
        'puts "Cycle # [lindex $cyclelabel [expr $ii-1]] successfully finished!"', newline,...
        'puts "target displ.  = [expr [lindex $peakDisp [expr $ii-1]]]"', newline,...
        'puts "current displ. = [nodeDisp $controlNode $controlDOF]"', newline,...
```

```
            'puts "--------------------------------x--------------------------------------";', newline,...
          '};', newline,...
        '};', newline,...
    '};', newline,...
    'if { $ok<0 } {', newline,...
    '   puts "FAILED TO CONVERGE!!" ', newline,...
    '} else {', newline,...
    '   puts "";', newline,...
    '   puts "ALL SUCCESSFUL!!"', newline,...
    '   puts $modelnum', newline,...
    '}', newline]);
    fclose(FID);

end
```

```matlab
%% Define Constitutive Properties
% Steel01
% uniaxialMaterial Steel01 matTag Fy E0 b a1 a2 a3 a4
Fy = 455.0;
E = 215000;
b = 0.01;
matDef_Steel01 = join(["uniaxialMaterial Steel01 1",Fy, E, b],' ');

% Steel02
% uniaxialMaterial Steel02 matTag Fy E b RO cR1 cR2 a1 a2 a3 a4
Fy = 455.0;
E = 215000;
b = 0.01;
ROm = 20.0;
cR1 = 0.925;
cR2 = 0.15;
a1 = 0;
a2 = 1;
a3 = 0;
a4 = 1;
matDef_Steel02 = join(["uniaxialMaterial Steel02 1",Fy, E, b, ROm cR1, cR2, a1, a2, a3, a4],' ');

% Concrete01 Cover
% uniaxialMaterial Concrete01 matTag fpc epsc0 fpcu epsU
fpc = -32.5;
epsc0 = -0.0024074074;
fpcu = -6.5;
epsU = -0.004;
matDef_Concrete01_Cover = join(["uniaxialMaterial Concrete01 1",fpc, epsc0, fpcu, epsU],' ');

% Concrete01
% uniaxialMaterial Concrete01 matTag fpc epsc0 fpcu epsU
fpc = -47.9;
epsc0 = -0.003548148;
fpcu = -40.715;
epsU = -0.0276;
matDef_Concrete01_Core = join(["uniaxialMaterial Concrete01 1",fpc, epsc0, fpcu, epsU],' ');

% Concrete02
% uniaxialMaterial Concrete02 matTag fpc epsc0 fpcu epsU lambda ft Ets
fpc = -32.5;
epsc0 = -0.0024074074;
fpcu = -6.5;
epsU = -0.004;
lambda = 0.25;
ft = 1.9;
Ets = 2700.0;
matDef_Concrete02_Cover = join(["uniaxialMaterial Concrete02 1",fpc, epsc0, fpcu, epsU, lambda, ft, Ets],' ');

% Concrete02
% uniaxialMaterial Concrete02 matTag fpc epsc0 fpcu epsU lambda ft Ets
fpc = -47.9;
epsc0 = -0.003548148;
fpcu = -40.715;
epsU = -0.0276;
lambda = 0.25;
ft = 1.9;
```

```
Ets = 2700.0;
matDef_Concrete02_Core = join(["uniaxialMaterial Concrete02 1",fpc, epsc0, fpcu, epsU, lambda, ft, Ets],' ');
%%
close all; clc;
localOpenSeesPath = "C:\Users\Louis Lin\Workspace\Academic\UCSD\SE 201B\Opensees\bin\OpenSees.exe";
i = 1;
numIncr = 1000;
%%  Steel
Steel_strain_history = [0, 0.002, -0.002, -0.01, 0.01,-0.02,0.02,-0.03,0.03,-0.04,0.04,-0.05,0.05,-0.06,0];

close all;
for matDef = [matDef_Steel01, matDef_Steel02]
    inputData = Steel_strain_history;
    figure(3); hold on;
    inputs = strsplit(matDef);
    out = get_materialHysteresis(matDef, inputData, numIncr, localOpenSeesPath);
    plot(out(:,1), out(:,2), 'linewidth', 1.5,'DisplayName',inputs(2)); grid on;
    xline(0,"HandleVisibility",'off'); yline(0,"HandleVisibility",'off');
    title("Material Hysteresis for Steel Reinforcement");
    ylabel('Stress [MPa]'); xlabel('Strain [-]'); legend('Location','southeast');
    xlim([-0.07, 0.07]);
end
set(gca,'FontSize',14);
ax2 = axes('Position',[.2 .8 .1 .1],'FontSize',14);
plot(Steel_strain_history); xline(0); yline(0);
title('Strain History'); box on; grid minor;
xticks('');
print_figure("Part A Steel ", [13,5], 14)
%% Cover
Cover_Concrete_strain_history = [0, 0.002, -0.002, -0.003, 0.003,-0.004,0.004,-0.005,0.005,-0.006,0.006,-0.007,0.007,0];
close all;
for matDef = [ matDef_Concrete01_Cover, matDef_Concrete02_Cover]
    inputData = Cover_Concrete_strain_history;
    figure(3); hold on;
    inputs = strsplit(matDef);
    out = get_materialHysteresis(matDef, inputData, numIncr, localOpenSeesPath);
    plot(out(:,1), out(:,2), 'linewidth', 1.5,'DisplayName',inputs(2)); grid on;
    xline(0,"HandleVisibility",'off'); yline(0,"HandleVisibility",'off');
    title("Material Hysteresis for Cover Concrete");
    ylabel('Stress [MPa]'); xlabel('Strain [-]'); legend('Location','southeast');
    xlim([-0.008, 0.001]);
end
set(gca,'FontSize',14);
ax2 = axes('Position',[.2 .2 .1 .1],'FontSize',14);
plot(Cover_Concrete_strain_history); xline(0); yline(0);
title('Strain History'); box on; grid minor;
xticks('');
print_figure("Part A Cover Concrete ", [13, 5], 14)

%% CORE
Core_Concrete_strain_history = [0,0.002, -0.002, 0.005,-0.005, 0.01, -0.01, 0.02, -0.02,0.03,-0.03, 0.04,-0.04, 0];
close all;
for matDef = [ matDef_Concrete01_Core, matDef_Concrete02_Core]
    inputData = Core_Concrete_strain_history;
    figure(3); hold on;
    out = get_materialHysteresis(matDef, inputData, numIncr, localOpenSeesPath);
    inputs = strsplit(matDef);
```

```
    plot(out(:,1), out(:,2), 'linewidth', 1.5,'DisplayName',inputs(2)); grid on;
    xline(0,"HandleVisibility",'off'); yline(0,"HandleVisibility",'off');
     title("Material Hysteresis for Core Concrete");
    ylabel('Stress [MPa]'); xlabel('Strain [-]'); legend('Location','southeast');
    xlim([-0.05, 0.01]);
end
set(gca,'FontSize',14);
ax2 = axes('Position',[.17 .6 .1 .1],'FontSize',14);
plot(Core_Concrete_strain_history); xline(0); yline(0);
title('Strain History'); box on; grid minor;
xticks('');
print_figure("Part A Core Concrete ", [13, 5], 14)

%%
close all;
figure();hold on;
set(gca,'DefaultLineLineWidth',2)
plot(Steel_strain_history)
plot(Cover_Concrete_strain_history)
plot(Core_Concrete_strain_history)
legend('Steel','Cover Concrete','Core Concrete','Location','Southwest')
grid on; grid minor;
xticks('')
ylabel('Strain [-]')
title('Strain History for Material Hysteris')
print_figure("Part A Strain History", [13, 5], 14)

%%
function  [ out ] = get_materialHysteresis( matDef, inputData, numIncr, localOpenSeesPath )

%% Function
[matDef, localOpenSeesPath] = convertStringsToChars(matDef, localOpenSeesPath);
inputData = arrayfun(@(x) num2str(x), inputData, 'UniformOutput', 0);
temp = strsplit(matDef);
matTag = temp{3};

materialTesterFid = fopen('matTest.tcl','w+');
fprintf(materialTesterFid,['wipe;\n']);
fprintf(materialTesterFid,['model testUniaxial;\n']);
fprintf(materialTesterFid,['set matTag ', num2str(matTag,'%u'), ';\n']);
fprintf(materialTesterFid,['set strainHistory {', strjoin(inputData), '};\n']);
fprintf(materialTesterFid,['set fileOut "hysteresis_matTag_$matTag.txt";\n']);
fprintf(materialTesterFid,['set out [open $fileOut w];\n']);
fprintf(materialTesterFid,[matDef '\n']);
fprintf(materialTesterFid,...
    ['uniaxialTest $matTag;\n',...
    'set strain 0.0;\n',...
    'set count 1;\n',...
    'set iTime 0;\n',...
    'set strain [expr $strain];\n',...
    'strainUniaxialTest $strain;\n',...
    'set stress [stressUniaxialTest];\n',...
    'set tangent [tangUniaxialTest];\n',...
    'set iTime [expr $iTime+1];\n',...
    'puts $out "$strain $stress";\n',...
    'foreach {strainExtremeVal} $strainHistory {\n',...
    '       set numIncr ' num2str(numIncr,'%u') ';\n',...
```

```
'       set strainIncr [expr ($strainExtremeVal - $strain)/$numIncr];\n',...
'       for {set i 0} {$i < $numIncr} {incr i 1} {\n',...
'          set strain [expr $strain+$strainIncr];\n',...
'          strainUniaxialTest $strain;\n',...
'          set stress [stressUniaxialTest];\n',...
'          set tangent [tangUniaxialTest];\n',...
'          set iTime [expr $iTime+1];\n',...
'          puts $out "$strain $stress";\n',...
'       }\n',...
'}\n',...
'close $out;\n',...
'puts "MATERIAL TESTER RAN SUCCESSFULLY!";\n',...
'wipe;\n'...
]);
fclose(materialTesterFid);

[~, ~] = system(['"',localOpenSeesPath,'" "matTest.tcl"']);

fid = fopen(['hysteresis_matTag_' num2str(matTag,'%u') '.txt'],'r');
dataRead = textscan(fid, repmat('%f ',1,2), 'CollectOutput',true);
out = dataRead{1};
fclose(fid);
delete(['hysteresis_matTag_' num2str(matTag,'%u') '.txt']);
delete('matTest.tcl');

end
```

Appendix B. Opensees Files

```
# ##########################################################################
# SE 201B: NONLINEAR STRUCTURAL ANALYSIS
# NONLINEAR FIBER SECTION ANALYSIS
# ##########################################################################

#Always start with
wipe; # Clear memory of all past model definitions
model BasicBuilder -ndm 2 -ndf 3; # Define the model builder, ndm=#dimension, ndf=#dofs

# --------------------------------------------------------------------------
# DEFINE NODES
# --------------------------------------------------------------------------
set nodeTag1 1;
set nodeTag2 2;

node $nodeTag1       0. 0.;
node $nodeTag2       0. 0.;

puts $modelExportFileID "node $nodeTag1       0. 0.;"
puts $modelExportFileID "node $nodeTag2       0. 0.;"

# --------------------------------------------------------------------------
# DEFINE CONSTRAINTS
# --------------------------------------------------------------------------
fix $nodeTag1        1 1 1; # Pin
fix $nodeTag2        0 1 0; # Roller

# --------------------------------------------------------------------------
# DEFINE MATERIAL
# --------------------------------------------------------------------------

# Define unconfined concrete material parameters
set fpc              [expr -32.2*$MPa]
set Ec               [expr 27100.0*$MPa]
set epsc0            [expr 2.0*$fpc/$Ec]
set ft               [expr 4.8*$MPa]
set lambda           0.25
set Ets              [expr 0.01*$Ec]
set fpcU             [expr 0.2*$fpc]
set epsU             -0.004

# Define confined concrete material parameters
set fpcc             [expr -53.7*$MPa]
set Ecc              [expr 27100.0*$MPa]
set epscc0           [expr 2.0*$fpcc/$Ecc]
set ftc              [expr 4.8*$MPa]
set lambdac          0.25
set Etsc             [expr 0.01*$Ecc]
set fpccU            [expr 0.85*$fpcc]
set epscU            -0.0276

# Define steel material parameters
set fy               [expr 450.0*$MPa]
set Es               [expr 218300.0*$MPa]
set b                0.0219
set R0               20.0
set cR1              0.925
set cR2              0.15
set a1               0.0
set a2               1.0
set a3               0.0
set a4               1.0
set sigInit          0.0

set matTagConcCover 1
set matTagConcCore  2
set matTagSteel     3
set modelnum 1.0

```

```
# Unconfined concrete:
uniaxialMaterial Concrete02 $matTagConcCover  $fpc $epsc0 $fpcU $epsU $lambda $ft $Ets

# Confined concrete:
uniaxialMaterial Concrete02 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU $lambdac $ftc $Etsc

# Reinforcing steel:
uniaxialMaterial Steel02    $matTagSteel      $fy $Es $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4 $sigInit

puts $modelExportFileID "uniaxialMaterial Concrete02 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU $lambdac $ftc $Etsc"
puts $modelExportFileID "uniaxialMaterial Concrete02 $matTagConcCover  $fpc $epsc0 $fpcU $epsU $lambda $ft $Ets"
puts $modelExportFileID "uniaxialMaterial Steel02    $matTagSteel      $fy $Es $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4 $sigInit"

# ------------------------------------------------------------------------------------
# DEFINE SECTION
# ------------------------------------------------------------------------------------
set colWidth    [expr 400.*$mm]
set colDepth    [expr 400.*$mm]
set colArea     [expr $colWidth * $colDepth]
set cover       [expr 40.*$mm]
set dB          [expr 20.*$mm]
set As          [expr 314.159*$mm2]
set y1          [expr $colDepth/2.0]
set z1          [expr $colWidth/2.0]
set totNumBars  8

set secTag 3
set fiberA 20
set fiberB 5
set fiberC 20

section Fiber $secTag -GJ $Ubig {
    # ------------------------------------------------------------------------------------
    # Create rectangular patches
    # ------------------------------------------------------------------------------------
    # Cover concrete
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # Core concrete
    patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # ------------------------------------------------------------------------------------
    # Create straight layers
    # ------------------------------------------------------------------------------------
    # Reinforcing steel
    layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
}


puts $modelExportFileID "section Fiber $secTag -GJ $Ubig {
```

```
	# ------------------------------------------------------------------------------------
	# Create rectangular patches
	# ------------------------------------------------------------------------------------
	# Cover concrete
	patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
	patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
	patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
	patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
	# Core concrete
	patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
	# ------------------------------------------------------------------------------------
	# Create straight layers
	# ------------------------------------------------------------------------------------
	# Reinforcing steel
	layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
	layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
	layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
}"

# ------------------------------------------------------------------------------------
# DEFINE ELEMENT
# ------------------------------------------------------------------------------------
set eleTag 1
set secTag 3
element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0
puts $modelExportFileID "element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0"
close $modelExportFileID

set controlNode $nodeTag2
```

```
# ##################################################################################
# SE 201B: NONLINEAR STRUCTURAL ANALYSIS
# NONLINEAR FIBER SECTION ANALYSIS
# ##################################################################################

#Always start with
wipe; # Clear memory of all past model definitions
model BasicBuilder -ndm 2 -ndf 3; # Define the model builder, ndm=#dimension, ndf=#dofs

# ----------------------------------------------------------------------------------
# DEFINE NODES
# ----------------------------------------------------------------------------------
set nodeTag1 1;
set nodeTag2 2;

node $nodeTag1       0. 0.;
node $nodeTag2       0. 0.;

#puts $modelExportFileID "node $nodeTag1       0. 0.;"
#puts $modelExportFileID "node $nodeTag2       0. 0.;"

# ----------------------------------------------------------------------------------
# DEFINE CONSTRAINTS
# ----------------------------------------------------------------------------------
fix $nodeTag1        1 1 1; # Pin
fix $nodeTag2        0 1 0; # Roller

# ----------------------------------------------------------------------------------
# DEFINE MATERIAL
# ----------------------------------------------------------------------------------

# Define unconfined concrete material parameters
set fpc             [expr -32.5*$MPa]
set Ec              [expr 27000.0*$MPa]
set epsc0           [expr 2.0*$fpc/$Ec]
set ft              [expr 1.9*$MPa]
set lambda          0.25
set Ets             [expr 0.1*$Ec]
set fpcU            [expr 0.2*$fpc]
set epsU            -0.004

# Define confined concrete material parameters
set fpcc            [expr -47.9*$MPa]
set Ecc             [expr 27000.0*$MPa]
set epscc0          [expr 2.0*$fpcc/$Ecc]
set ftc             [expr 1.9*$MPa]
set lambdac         0.25
set Etsc            [expr 0.1*$Ecc]
set fpccU           [expr 0.85*$fpcc]
set epscU           -0.0276

# Define steel material parameters
set fy              [expr 455.0*$MPa]
set Es              [expr 215000.0*$MPa]
set b               0.01
set R0              20.0
set cR1             0.925
set cR2             0.15
set a1              0.0
set a2              1.0
set a3              0.0
set a4              1.0
set sigInit         0.0

set matTagConcCover 1
set matTagConcCore  2
set matTagSteel     3
set modelnum 1.0

```

```
# Unconfined concrete:
uniaxialMaterial Concrete02 $matTagConcCover  $fpc $epsc0 $fpcU $epsU $lambda $ft $Ets

# Confined concrete:
uniaxialMaterial Concrete02 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU $lambdac $ftc $Etsc

# Reinforcing steel:
uniaxialMaterial Steel02    $matTagSteel      $fy $Es $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4 $sigInit

#puts $modelExportFileID "uniaxialMaterial Concrete02 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU $lambdac $ftc $Etsc"
#puts $modelExportFileID "uniaxialMaterial Concrete02 $matTagConcCover  $fpc $epsc0 $fpcU $epsU $lambda $ft $Ets"
#puts $modelExportFileID "uniaxialMaterial Steel02    $matTagSteel      $fy $Es $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4 $sigInit"

# ------------------------------------------------------------------------------------
# DEFINE SECTION
# ------------------------------------------------------------------------------------
set colWidth    [expr 400.*$mm]
set colDepth    [expr 400.*$mm]
set colArea     [expr $colWidth * $colDepth]
set cover       [expr 40.*$mm]
set dB          [expr 20.*$mm]
set As          [expr 314.159*$mm2]
set y1          [expr $colDepth/2.0]
set z1          [expr $colWidth/2.0]
set totNumBars  8

set secTag 1
set fiberA 3
set fiberB 1
set fiberC 3

section Fiber $secTag -GJ $Ubig {
    # ----------------------------------------------------------------------------------
    # Create rectangular patches
    # ----------------------------------------------------------------------------------
    # Cover concrete
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # Core concrete
    patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # ----------------------------------------------------------------------------------
    # Create straight layers
    # ----------------------------------------------------------------------------------
    # Reinforcing steel
    layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
}


#puts $modelExportFileID "section Fiber $secTag -GJ $Ubig {
```

```
    # ----------------------------------------------------------------------------------------
    # Create rectangular patches
    # ----------------------------------------------------------------------------------------
    # Cover concrete
    #patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    #patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    #patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    #patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # Core concrete
    #patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # ----------------------------------------------------------------------------------------
    # Create straight layers
    # ----------------------------------------------------------------------------------------
    # Reinforcing steel
    #layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    #layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    #layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
#}"

set secTag 2
set fiberA 8
set fiberB 2
set fiberC 8

section Fiber $secTag -GJ $Ubig {
    # ----------------------------------------------------------------------------------------
    # Create rectangular patches
    # ----------------------------------------------------------------------------------------
    # Cover concrete
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # Core concrete
    patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # ----------------------------------------------------------------------------------------
    # Create straight layers
    # ----------------------------------------------------------------------------------------
    # Reinforcing steel
    layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
}


# puts $modelExportFileID "section Fiber $secTag -GJ $Ubig {
    # #
```

```
    -------------------------------------------------------------------------------------
    # # Create rectangular patches
    # # -------------------------------------------------------------------------------------
    # # Cover concrete
    # patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    # patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    # patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    # patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # # Core concrete
    # patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # # -------------------------------------------------------------------------------------
    # # Create straight layers
    # # -------------------------------------------------------------------------------------
    # # Reinforcing steel
    # layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    # layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    # layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
# }"

set secTag 3
set fiberA 20
set fiberB 5
set fiberC 20

section Fiber $secTag -GJ $Ubig {
    # -------------------------------------------------------------------------------------
    # Create rectangular patches
    # -------------------------------------------------------------------------------------
    # Cover concrete
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # Core concrete
    patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # -------------------------------------------------------------------------------------
    # Create straight layers
    # -------------------------------------------------------------------------------------
    # Reinforcing steel
    layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
}


# puts $modelExportFileID "section Fiber $secTag -GJ $Ubig {
    # # -------------------------------------------------------------------------------------
```

```
    # # Create rectangular patches
    # # ------------------------------------------------------------------------------------
    # # Cover concrete
    # patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    # patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    # patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    # patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # # Core concrete
    # patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # # ------------------------------------------------------------------------------------
    # # Create straight layers
    # # ------------------------------------------------------------------------------------
    # # Reinforcing steel
    # layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    # layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    # layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
# }"

# ------------------------------------------------------------------------------------
# DEFINE ELEMENT
# ------------------------------------------------------------------------------------
set eleTag 1
set secTag 3
element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0
# puts $modelExportFileID "element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0"
# close $modelExportFileID

set controlNode $nodeTag2
```

```
# ##################################################################################
# SE 201B: NONLINEAR STRUCTURAL ANALYSIS
# NONLINEAR FIBER SECTION ANALYSIS
# ##################################################################################

#Always start with
wipe; # Clear memory of all past model definitions
model BasicBuilder -ndm 2 -ndf 3; # Define the model builder, ndm=#dimension, ndf=#dofs

# ----------------------------------------------------------------------------------
# DEFINE NODES
# ----------------------------------------------------------------------------------
set nodeTag1 1;
set nodeTag2 2;

node $nodeTag1        0. 0.;
node $nodeTag2        0. 0.;

# puts $modelExportFileID "node $nodeTag1        0. 0.;"
# puts $modelExportFileID "node $nodeTag2        0. 0.;"

# ----------------------------------------------------------------------------------
# DEFINE CONSTRAINTS
# ----------------------------------------------------------------------------------
fix $nodeTag1         1 1 1; # Pin
fix $nodeTag2         0 1 0; # Roller

# ----------------------------------------------------------------------------------
# DEFINE MATERIAL
# ----------------------------------------------------------------------------------

# Define unconfined concrete material parameters
set fpc              [expr -32.5*$MPa]
set Ec               [expr 27000.0*$MPa]
set epsc0            [expr 2.0*$fpc/$Ec]
set ft               [expr 1.9*$MPa]
set lambda           0.25
set Ets              [expr 0.1*$Ec]
set fpcU             [expr 0.2*$fpc]
set epsU             -0.004

# Define confined concrete material parameters
set fpcc             [expr -47.9*$MPa]
set Ecc              [expr 27000.0*$MPa]
set epscc0           [expr 2.0*$fpcc/$Ecc]
set ftc              [expr 1.9*$MPa]
set lambdac          0.25
set Etsc             [expr 0.1*$Ecc]
set fpccU            [expr 0.85*$fpcc]
set epscU            -0.0276

# Define steel material parameters
set fy               [expr 455.0*$MPa]
set Es               [expr 215000.0*$MPa]
set b                0.01
set R0               20.0
set cR1              0.925
set cR2              0.15
set a1               0.0
set a2               1.0
set a3               0.0
set a4               1.0
set sigInit          0.0

set matTagConcCover 1
set matTagConcCore  2
set matTagSteel     3
set modelnum 2.0
```

```
# Unconfined concrete:
uniaxialMaterial Concrete01 $matTagConcCover  $fpc $epsc0 $fpcU $epsU

# Confined concrete:
uniaxialMaterial Concrete01 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU

# Reinforcing steel:
uniaxialMaterial Steel01    $matTagSteel      $fy $Es $b $a1 $a2 $a3 $a4

# puts $modelExportFileID "uniaxialMaterial Concrete01 $matTagConcCover  $fpc $epsc0 $fpcU $epsU"
# puts $modelExportFileID "uniaxialMaterial Concrete01 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU"
# puts $modelExportFileID "uniaxialMaterial Steel01    $matTagSteel      $fy $Es $b $a1 $a2 $a3 $a4"


# ------------------------------------------------------------------------------------
# DEFINE SECTION
# ------------------------------------------------------------------------------------
set colWidth    [expr 400.*$mm]
set colDepth    [expr 400.*$mm]
set colArea     [expr $colWidth * $colDepth]
set cover       [expr 40.*$mm]
set dB          [expr 20.*$mm]
set As          [expr 314.*$mm2]
set y1          [expr $colDepth/2.0]
set z1          [expr $colWidth/2.0]
set totNumBars  8

set secTag 3
set fiberA 20
set fiberB 5
set fiberC 20

section Fiber $secTag -GJ $Ubig {
    # ------------------------------------------------------------------------------------
    # Create rectangular patches
    # ------------------------------------------------------------------------------------
    # Cover concrete
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # Core concrete
    patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # ------------------------------------------------------------------------------------
    # Create straight layers
    # ------------------------------------------------------------------------------------
    # Reinforcing steel
    layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
}


# puts $modelExportFileID "section Fiber $secTag -GJ $Ubig {
    # #
```

```
    # # Create rectangular patches
    # # ------------------------------------------------------------------------------------
    # # Cover concrete
    # patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    # patch rect $matTagConcCover $fiberA 1 [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    # patch rect $matTagConcCover $fiberB 1 [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    # patch rect $matTagConcCover $fiberB 1 [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # # Core concrete
    # patch rect $matTagConcCore $fiberC 1 [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # # ------------------------------------------------------------------------------------
    # # Create straight layers
    # # ------------------------------------------------------------------------------------
    # # Reinforcing steel
    # layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    # layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    # layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
# }"

# ------------------------------------------------------------------------------------
# DEFINE ELEMENT
# ------------------------------------------------------------------------------------
set eleTag 1
set secTag 3
element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0
# puts $modelExportFileID "element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0"
# close $modelExportFileID

set controlNode $nodeTag2
```

```
# ##################################################################################
# SE 201B: NONLINEAR STRUCTURAL ANALYSIS
# NONLINEAR FIBER SECTION ANALYSIS
# ##################################################################################

#Always start with
wipe; # Clear memory of all past model definitions
model BasicBuilder -ndm 3 -ndf 6; # Define the model builder, ndm=#dimension, ndf=#dofs

# ----------------------------------------------------------------------------------
# DEFINE NODES
# ----------------------------------------------------------------------------------
set nodeTag1 1;
set nodeTag2 2;

node $nodeTag1        0. 0. 0.;
node $nodeTag2        0. 0. 0.;

# ----------------------------------------------------------------------------------
# DEFINE CONSTRAINTS
# ----------------------------------------------------------------------------------
fix $nodeTag1         1 1 1 1 1 1; # Pin
fix $nodeTag2         0 1 1 1 0 0; # Roller

# ----------------------------------------------------------------------------------
# DEFINE MATERIAL
# ----------------------------------------------------------------------------------

# Define unconfined concrete material parameters
set fpc              [expr -32.5*$MPa]
set Ec               [expr 27000.0*$MPa]
set epsc0            [expr 2.0*$fpc/$Ec]
set ft               [expr 1.9*$MPa]
set lambda           0.25
set Ets              [expr 0.1*$Ec]
set fpcU             [expr 0.2*$fpc]
set epsU             -0.004

# Define confined concrete material parameters
set fpcc             [expr -47.9*$MPa]
set Ecc              [expr 27000.0*$MPa]
set epscc0           [expr 2.0*$fpcc/$Ecc]
set ftc              [expr 1.9*$MPa]
set lambdac          0.25
set Etsc             [expr 0.1*$Ecc]
set fpccU            [expr 0.85*$fpcc]
set epscU            -0.0276

# Define steel material parameters
set fy               [expr 455.0*$MPa]
set Es               [expr 215000.0*$MPa]
set b                0.01
set R0               20.0
set cR1              0.925
set cR2              0.15
set a1               0.0
set a2               1.0
set a3               0.0
set a4               1.0
set sigInit          0.0

set matTagConcCover 1
set matTagConcCore  2
set matTagSteel     3
set modelnum 1.0

# Unconfined concrete:
uniaxialMaterial Concrete02 $matTagConcCover  $fpc $epsc0 $fpcU $epsU $lambda $ft $Ets

```

```
# Confined concrete:
uniaxialMaterial Concrete02 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU $lambdac $ftc $Etsc

# Reinforcing steel:
uniaxialMaterial Steel02    $matTagSteel      $fy $Es $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4 $sigInit

#puts $modelExportFileID "uniaxialMaterial Concrete02 $matTagConcCore   $fpcc $epscc0 $fpccU $epscU $lambdac $ftc $Etsc"
#puts $modelExportFileID "uniaxialMaterial Concrete02 $matTagConcCover  $fpc $epsc0 $fpcU $epsU $lambda $ft $Ets"
#puts $modelExportFileID "uniaxialMaterial Steel02    $matTagSteel      $fy $Es $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4 $sigInit"

# ------------------------------------------------------------------------------------
# DEFINE SECTION
# ------------------------------------------------------------------------------------
set colWidth    [expr 400.*$mm]
set colDepth    [expr 400.*$mm]
set colArea     [expr $colWidth * $colDepth]
set cover       [expr 40.*$mm]
set dB          [expr 20.*$mm]
set As          [expr 314.159*$mm2]
set y1          [expr $colDepth/2.0]
set z1          [expr $colWidth/2.0]
set totNumBars  8

set secTag 4
set fiberA 40
set fiberB 5
set fiberC 40

section Fiber $secTag -GJ $Ubig {
    # ------------------------------------------------------------------------------------
    # Create rectangular patches
    # ------------------------------------------------------------------------------------
    # Cover concrete
    patch rect $matTagConcCover $fiberA $fiberB [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    patch rect $matTagConcCover $fiberA $fiberB [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    patch rect $matTagConcCover $fiberB [expr $fiberA+2*$fiberB] [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    patch rect $matTagConcCover $fiberB [expr $fiberA+2*$fiberB] [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # Core concrete
    patch rect $matTagConcCore $fiberC $fiberC [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # ------------------------------------------------------------------------------------
    # Create straight layers
    # ------------------------------------------------------------------------------------
    # Reinforcing steel
    layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
}


# puts $modelExportFileID "section Fiber $secTag -GJ $Ubig {
    # # ------------------------------------------------------------------------------------
    # # Create rectangular patches
```

```
    # # ------------------------------------------------------------------------------------------
    # # Cover concrete
    # patch rect $matTagConcCover $fiberA $fiberB [expr $cover - $y1] [expr -$z1] [expr $y1 - $cover] [expr $cover - $z1]
    # patch rect $matTagConcCover $fiberA $fiberB [expr $cover - $y1] [expr $z1 - $cover] [expr $y1 - $cover] [expr $z1]
    # patch rect $matTagConcCover $fiberB $fiberA [expr -$y1] [expr -$z1] [expr $cover - $y1] [expr $z1]
    # patch rect $matTagConcCover $fiberB $fiberA [expr $y1 - $cover] [expr -$z1] [expr $y1] [expr $z1]
    # # Core concrete
    # patch rect $matTagConcCore $fiberC $fiberC [expr $cover - $y1] [expr $cover - $z1] [expr $y1 - $cover] [expr $z1 - $cover]
    # # ------------------------------------------------------------------------------------------
    # # Create straight layers
    # # ------------------------------------------------------------------------------------------
    # # Reinforcing steel
    # layer straight $matTagSteel 3 $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    # layer straight $matTagSteel 2 $As 0 [expr $cover - $z1] 0 [expr $z1 - $cover]
    # layer straight $matTagSteel 3 $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]
# }"


# ------------------------------------------------------------------------------------------
# DEFINE ELEMENT
# ------------------------------------------------------------------------------------------
set eleTag 1
set secTag 4
element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0
# puts $modelExportFileID "element zeroLengthSection $eleTag $nodeTag1 $nodeTag2 $secTag -orient 1 0 0 0 1 0"
# close $modelExportFileID

set controlNode $nodeTag2
```

```
# ------------------------------------------------------------------------------------
# UNITS
# ------------------------------------------------------------------------------------
set in 1.;               # define basic units -- output units
set kip 1.;              # define basic units -- output units
set sec 1.;              # define basic units -- output units
set LunitTXT "inch";         # define basic-unit text for output
set FunitTXT "kip";          # define basic-unit text for output
set TunitTXT "sec";          # define basic-unit text for output
set ft  [expr 12.*$in];              # feet
set ksi [expr $kip/pow($in,2)];      # kips per square inch
set psi [expr $ksi/1000.];           # pounds per square inch
set lbf [expr $psi*$in*$in];         # pounds force
set pcf [expr $lbf/pow($ft,3)];      # pounds per cubic foot
set psf [expr $lbf/pow($ft,2)];      # pounds per square foot
set in2 [expr $in*$in];              # inch^2
set in4 [expr $in*$in*$in*$in];      # inch^4
set cm  [expr $in/2.54];             # centimeter
set cmsec2 [expr $cm/pow($sec,2)];   # cm/sec^2
set m [expr $cm*100];                # meter
set mm [expr $cm/10];                # millimeter
set mm2 [expr $mm*$mm];              # millimeter^2
set kN [expr 0.2247*$kip];           # kilo newton
set N [expr 1.e-3*$kN];              # newton
set MN [expr 1.e6*$N];               # mega newton
set MPa [expr 0.1450*$ksi];          # mega pascal
set GPa [expr 1000*$MPa];            # giga pascal
set pi [expr 2*asin(1.0)];           # define constants
set g [expr 32.2*$ft/pow($sec,2)];   # gravitational acceleration
set Ubig 1.e12;                      # a really large number
set Usmall [expr 1/$Ubig];           # a really small number
puts "Basic Units - $LunitTXT, $FunitTXT, $TunitTXT"
```