Nonlinear Fiber Section Analysis: *Displacement Control, Moment Curvature and Interaction Diagrams*

SE201B Nonlinear Structural Analysis

OpenSees Tutorial Session # 4

Instructor: Joel P. Conte, Professor

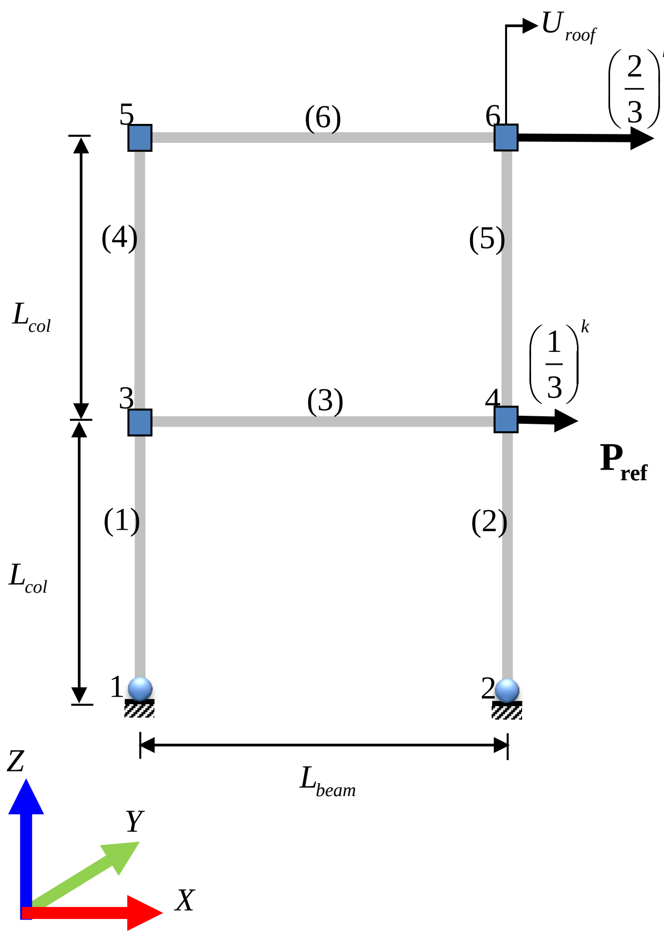
Tutorial By: Angshuman Deb, Ph.D. Student

Feb 19 2021

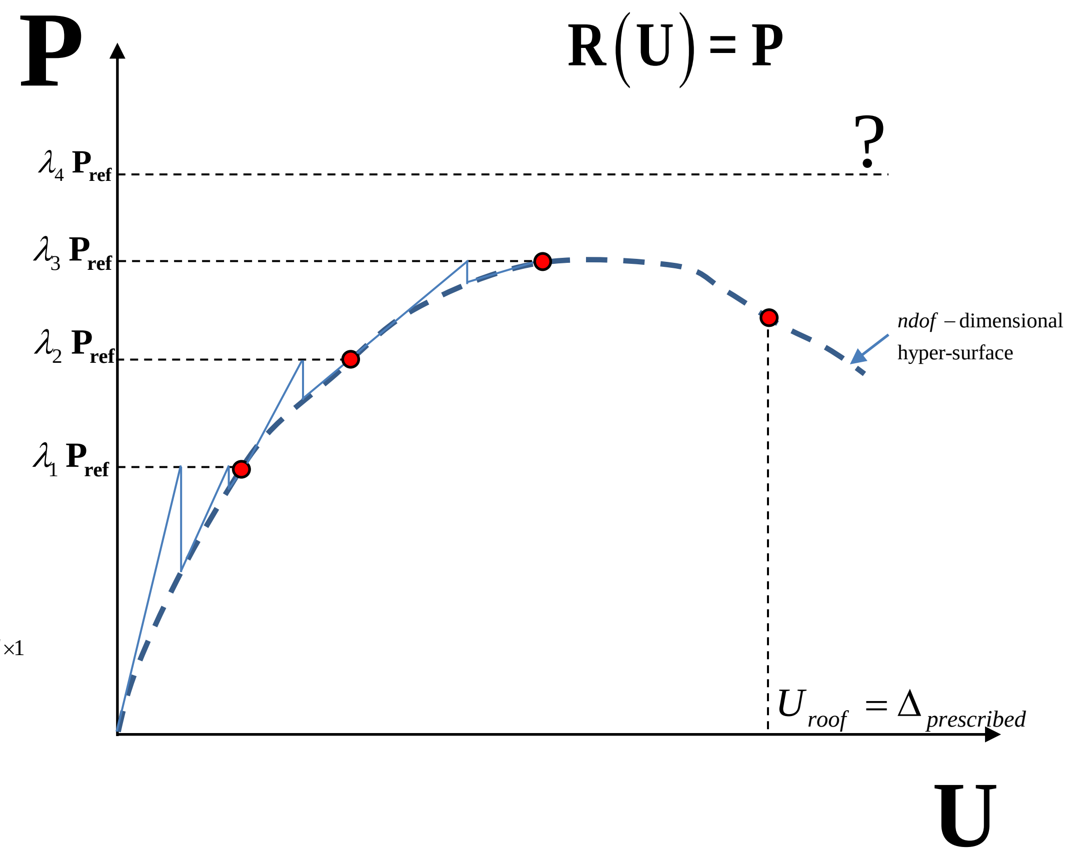
Outline

- Static Analysis: Force-control vs. displacement-control
- Hierarchical levels of a structure
- Material and fiber section definition in OpenSees
- Nonlinear fiber section analysis
- ***zeroLengthSection*** element
- PM/PMM interaction diagrams

Static Analysis Integrator: Load Control vs Displacement Control

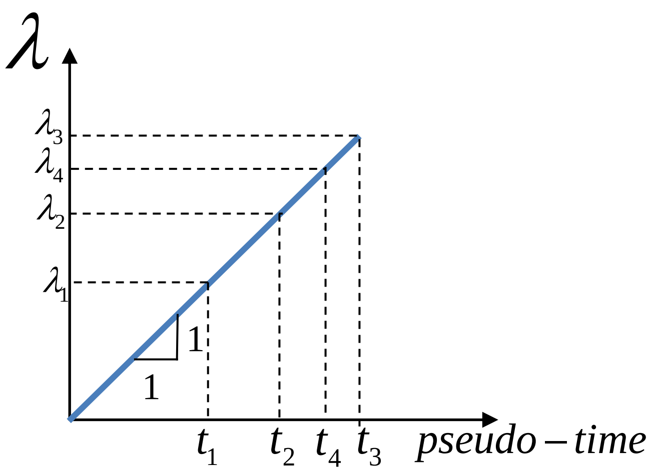


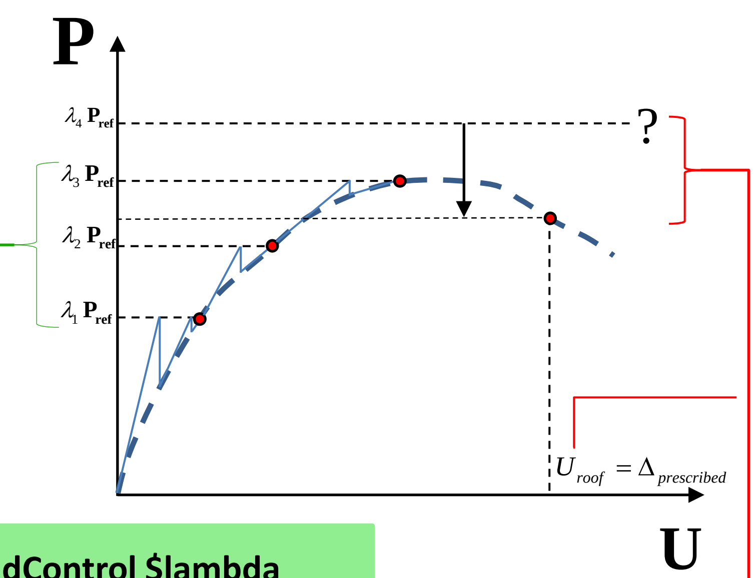
$$\mathbf{P_{ref}} = \begin{Bmatrix} \mathbf{0} \\ \frac{1}{3} \\ \mathbf{0} \\ \frac{2}{3} \end{Bmatrix}_{ndof \times 1}$$



Load Factor and Pseudo Time

By default, the load factor λ is varied with respect to pseudo-time, in a linear fashion, with a slope of 1.





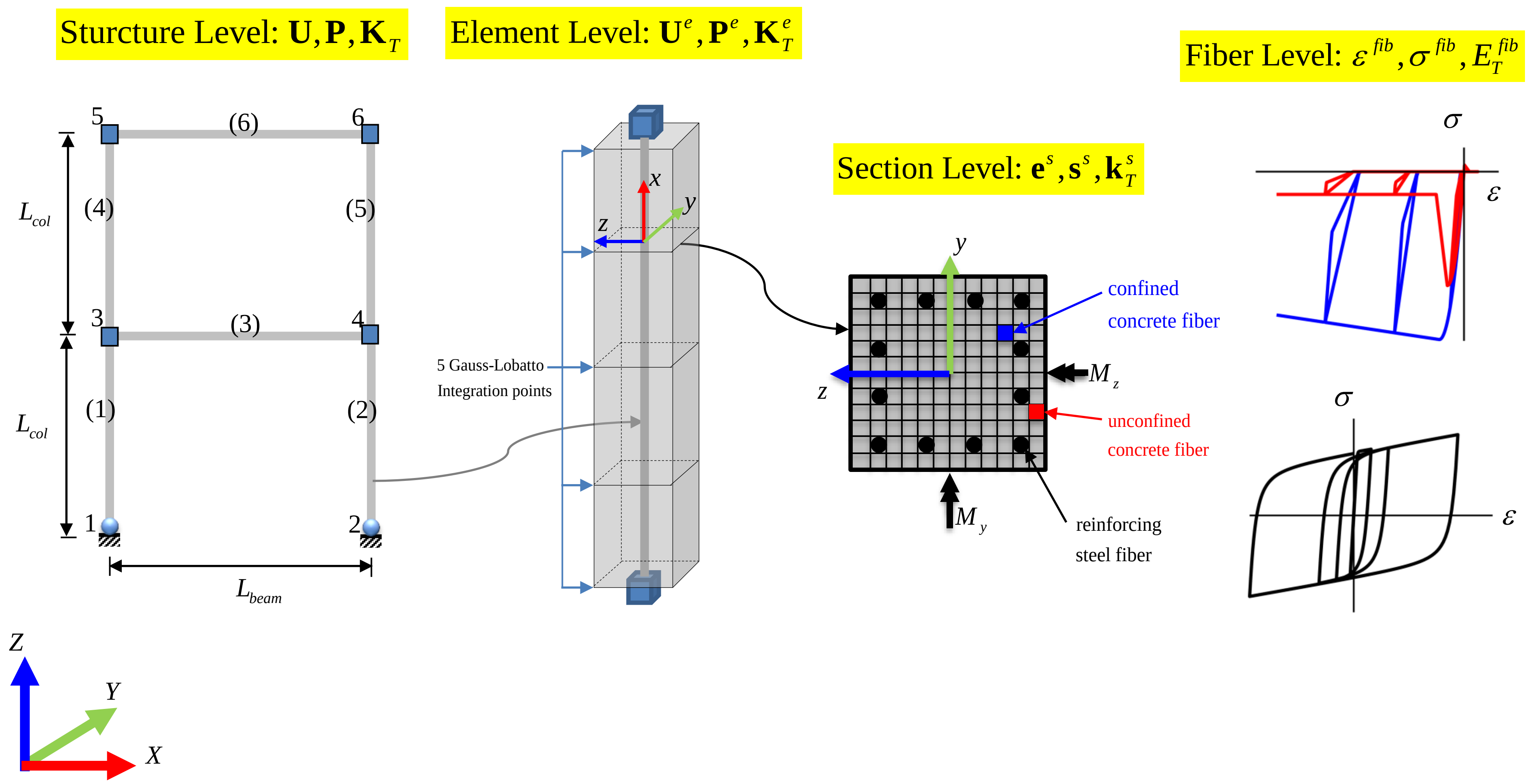
integrator LoadControl $lambda

User defined λ

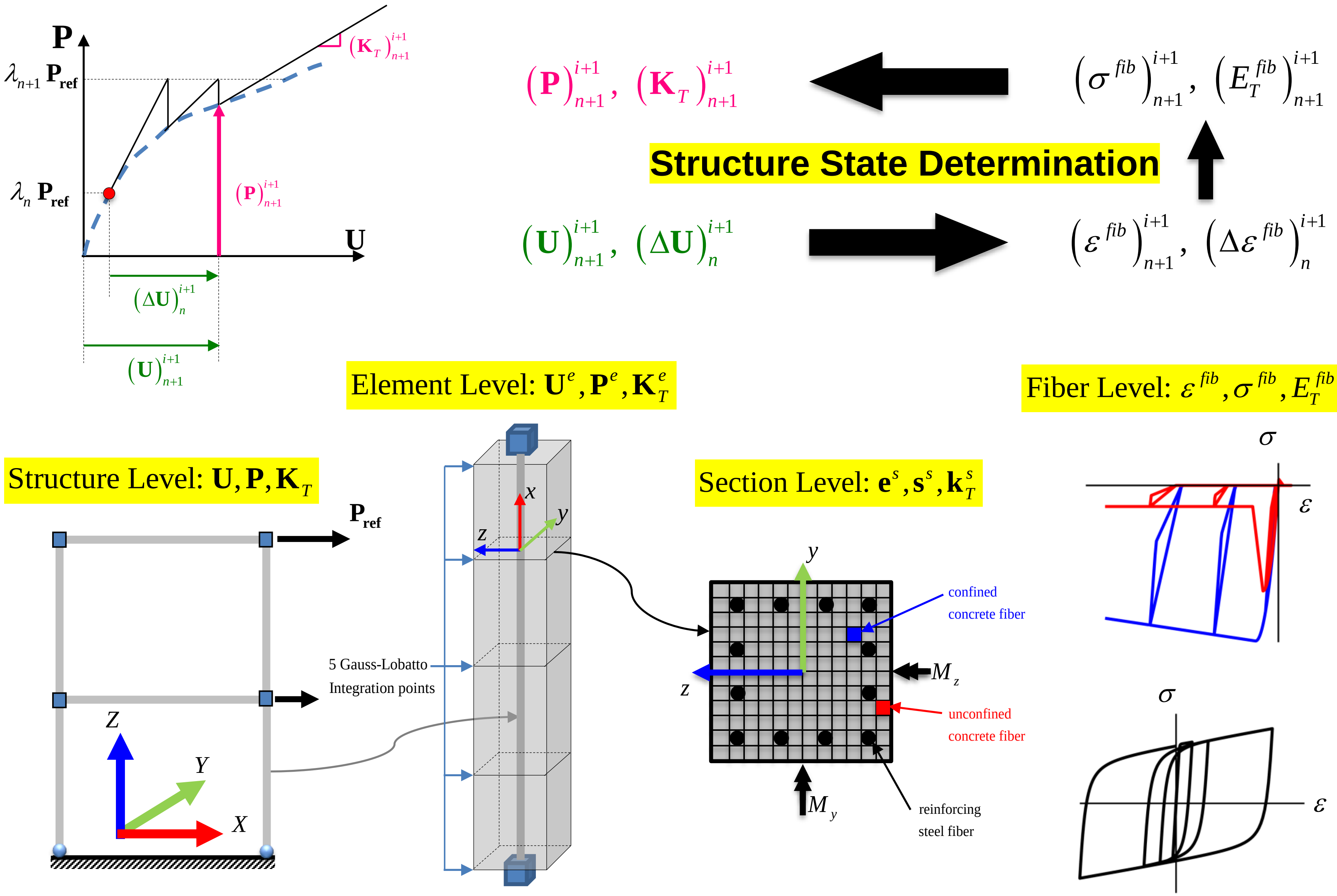
integrator DisplacementControl $node $dof $incr

Find a λ such that a prescribed displacement is achieved at **one control node** along **one control DOF**

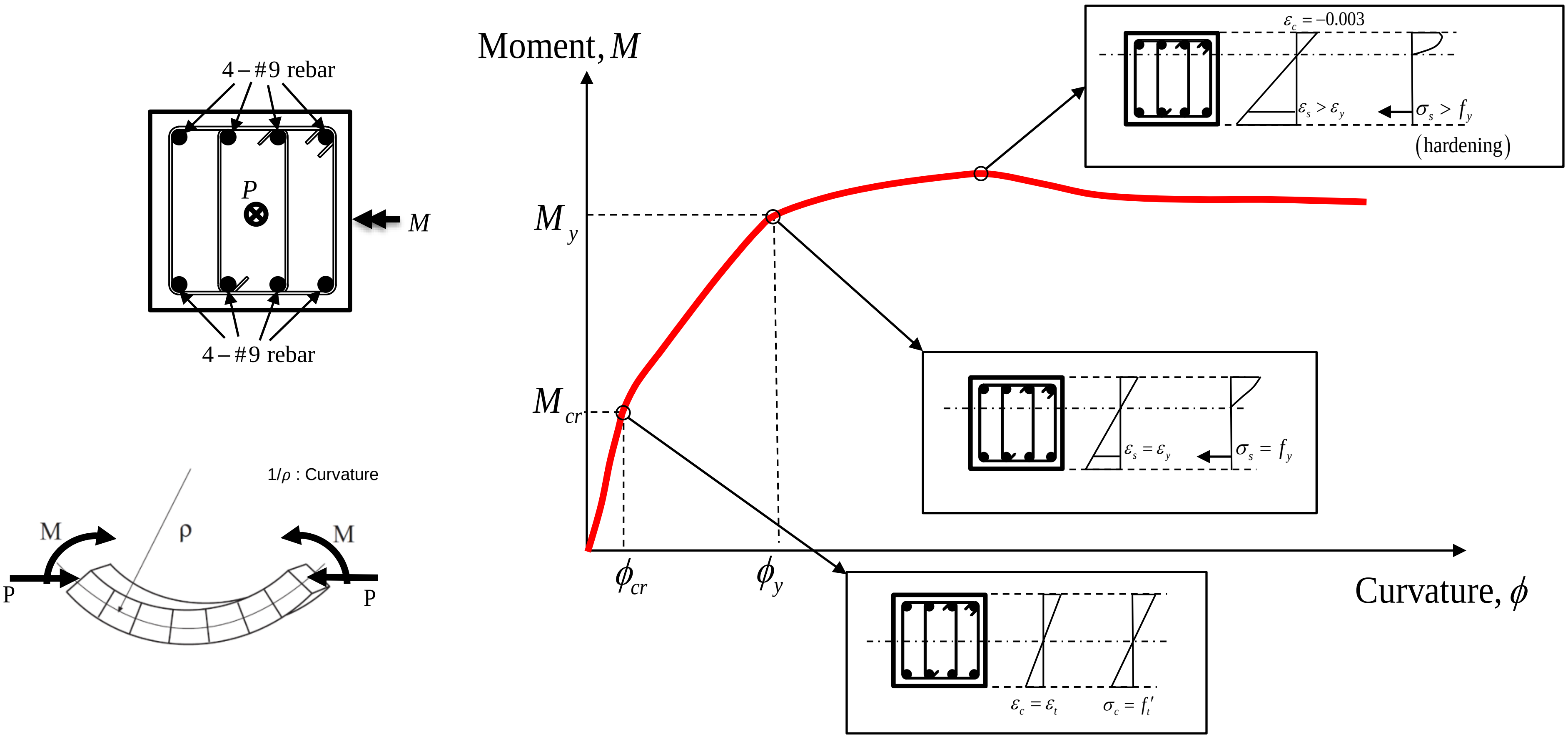
Hierarchical levels in an MDOF structure



Hierarchical levels in an MDOF structure

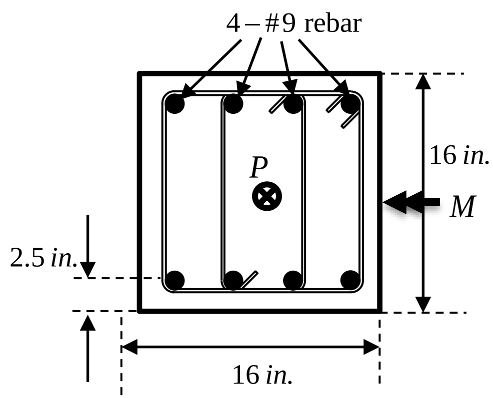


Moment Curvature Analysis

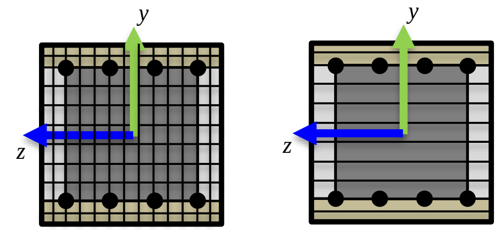


Problem Statement

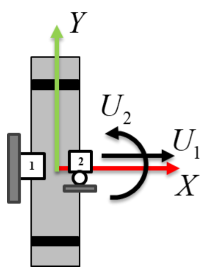
Problem Statement:
Develop a tcl code to compute the Moment-Curvature relationship of following column section:



1. Define materials
2. Define layer or fiber discretized section (based on uniaxial or biaxial bending) and attach materials to layers or fibers



3. Define nodes and ***zeroLengthSection*** **element.** Attach Fiber Section to element



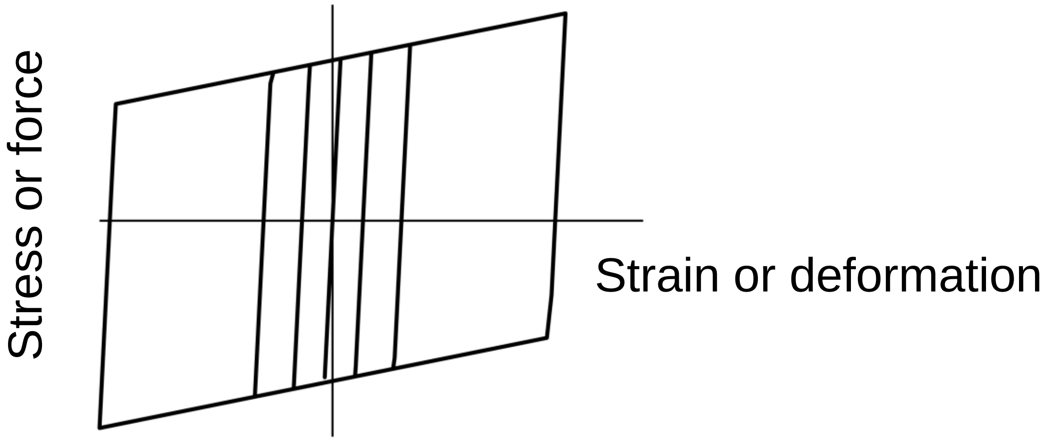
4. Apply a hybrid (force-controlled for axial load and displacement-controlled for moment/curvature) loading scheme.

Material Definition in OpenSees

Steel01:

- Uniaxial bilinear steel material
- With kinematic hardening and optional isotropic hardening described by a non-linear evolution equation.

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



Material Definition in OpenSees

Steel02:

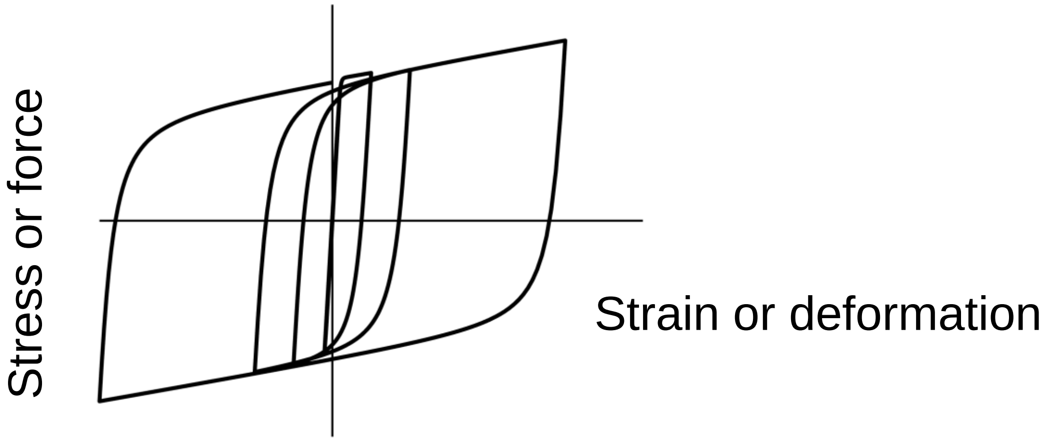
- Uniaxial Giuffre-Menegotto-Pinto steel material object with isotropic strain hardening.
- Same as Mate25 code provided in class.

$\frac{a_1}{R_0}$ of Prof. Conte's notes

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

parameters for Isotropic Hardening

a_2 of Prof. Conte's notes

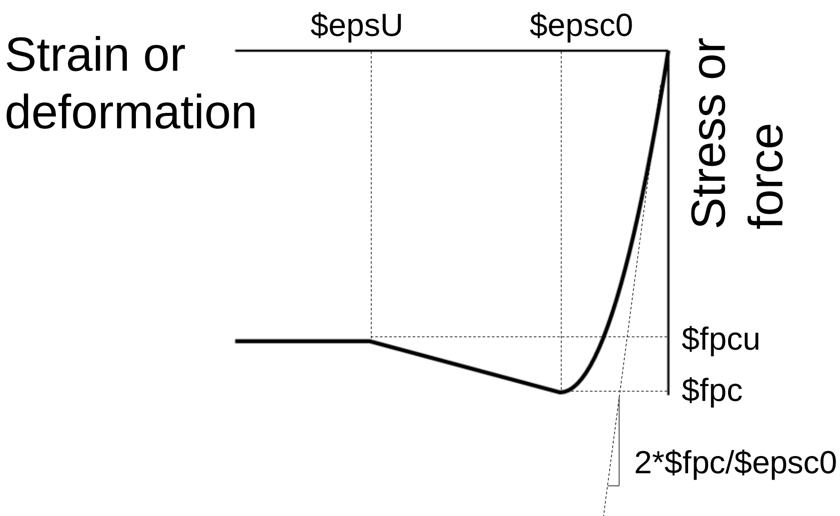


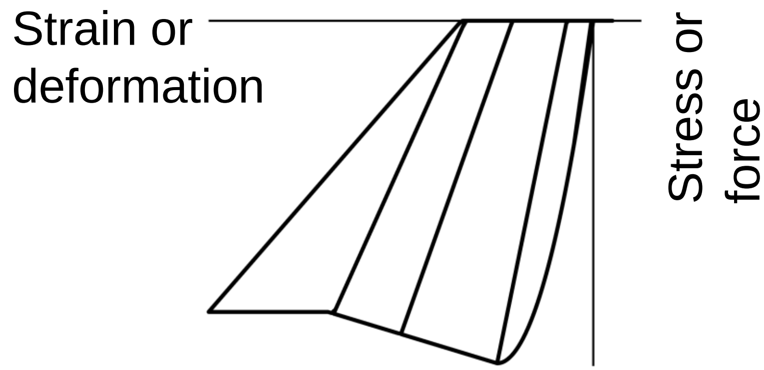
Material Definition in OpenSees

Concrete01:

- Kent-Scott-Park concrete.
- Degraded linear unloading/reloading stiffness according to the work of Karsan-Jirsa.
- No tensile strength.

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





Notes:

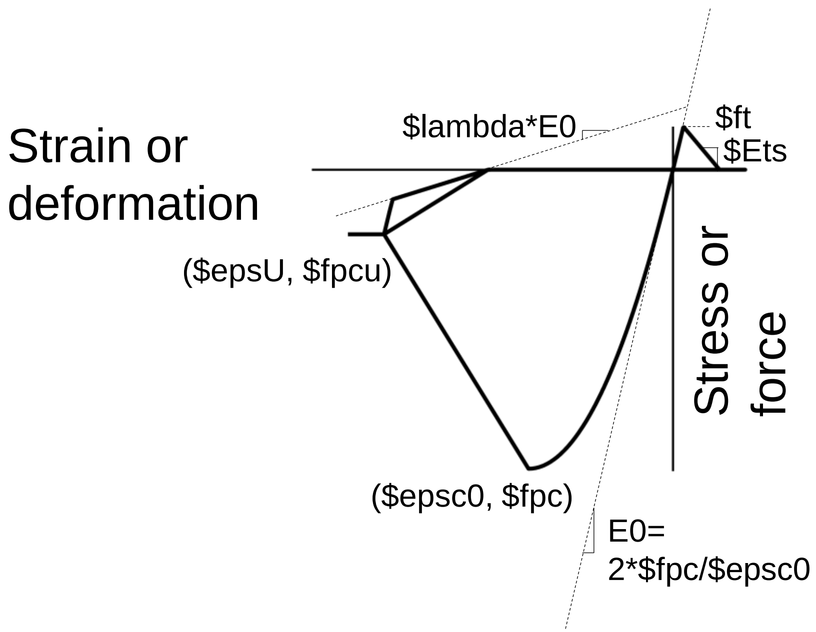
- *Compressive concrete parameters should be input as negative values.*
- *The initial slope for this model is* 2*$fpc/$epsc0

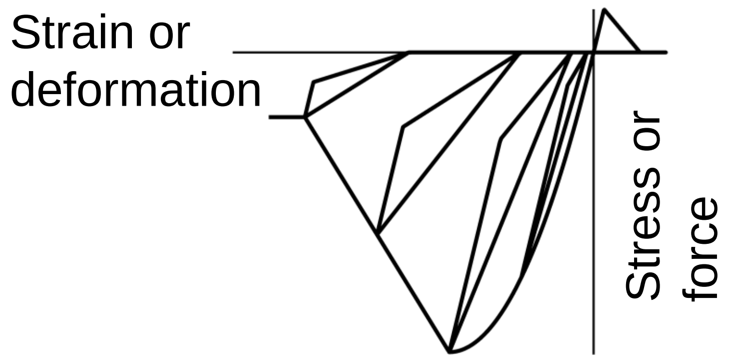
Material Definition in OpenSees

Concrete02:

- With tensile strength.
- Linear tension softening.

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





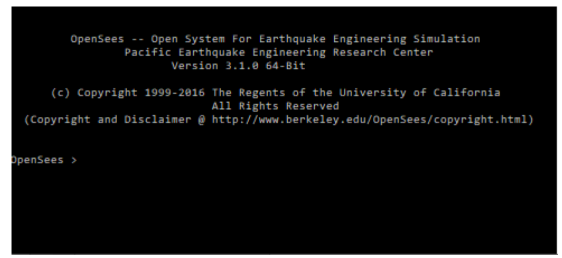
Notes:

- *Compressive concrete parameters should be input as negative values.*
- *The initial slope for this model is* 2*$fpc/$epsc0

Material Tester in OpenSees

```
output = get_materialHysteresis(matDef, inputData, numIncr, localOpenSeesPath)
```

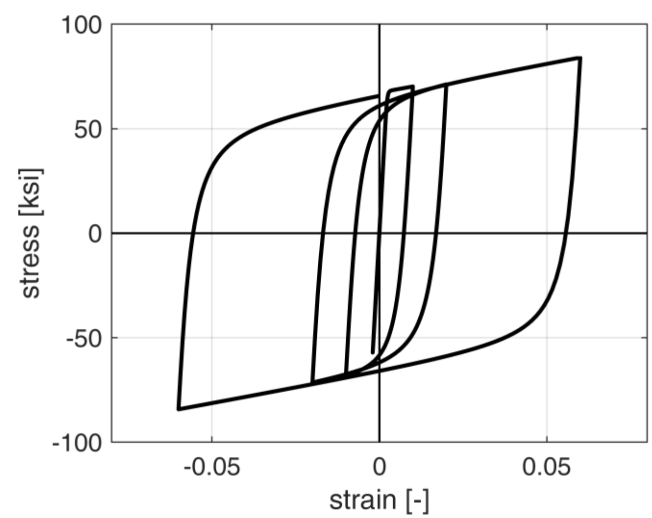
calls OpenSees executable located in localOpenSeesPath



get_materialHysteresis.m

Example MATLAB driver

```
figure();
matDef = 'uniaxialMaterial Steel02 1 68.0 29000.0 0.01 20.0 0.925 0.15';
inputData = [0, 0.002, -0.002, 0.01, -0.01, 0.02, -0.02, 0.06, -0.06, 0];
numIncr = 100;

% Use your own for this. Shift + Right-click (Option + Right-click in Mac)
% to copy OpenSees executable as path. Paste the path here.
localOpenSeesPath = "…/OpenSees.exe";

out = get_materialHysteresis(matDef, inputData, numIncr, localOpenSeesPath);
plot(out(:,1), out(:,2), 'k', 'linewidth', 1.5); grid on;
xline(0);
yline(0);
set(gca,'xlim',[-0.08, 0.08])
set(gca,'ylim',[-100, 100])
```



Fiber Section in OpenSees

- Fiber section formed by sub regions of simpler, regular shapes (e.g. quadrilateral, circular and triangular regions) called patches.
- Layers of reinforcement bars can be specified.
- Individual fibers, can also be defined.

Note: Each fiber is associated with an "uniaxialMaterial"!

https://opensees.berkeley.edu/wiki/index.php/Fiber_Section

Fiber Section in OpenSees

```
section Fiber $secTag -GJ $GJ {
fiber <fiber arguments>
patch <patch arguments>
layer <layer arguments>
}
```

Linear elastic torsional stiffness

Define one single fiber

Define rectangular, quadrilateral, or circular patches

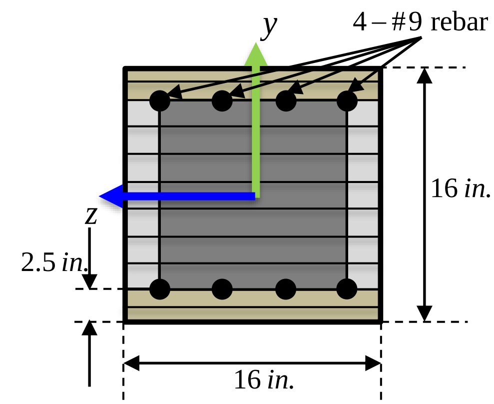
Define a number of fibers along a line or a circular arc

Note:

fiber, patch, and layer arguments require material tag of previously defined uniaxial material

Fiber Section in OpenSees - Example

```
# ---------------------------------------------------------------------------------
# DEFINE SECTION
# ---------------------------------------------------------------------------------
set colWidth     [expr 16.*$in]
set colDepth     [expr 16.*$in]
set colArea      [expr $colWidth * $colDepth]
set cover        [expr 2.5*$in]
set dB           [expr 1.128*$in]
set As           [expr 1.*$in2]
set y1           [expr $colDepth/2.0]
set z1           [expr $colWidth/2.0]
set secTag 1

section Fiber $secTag -GJ $Ubig {

    # ---------------------------------------------------------------------------------
    # Create rectangular patches
    # ---------------------------------------------------------------------------------

    # Cover concrete
    patch rect $matTagConcCover 7   1   [expr $cover - $y1]     [expr -$z1]             [expr $y1 - $cover]     [expr $cover - $z1]
    patch rect $matTagConcCover 7   1   [expr $cover - $y1]     [expr $z1 - $cover]     [expr $y1 - $cover]     [expr $z1]
    patch rect $matTagConcCover 2   1   [expr -$y1]             [expr -$z1]             [expr $cover - $y1]     [expr $z1]
    patch rect $matTagConcCover 2   1   [expr $y1 - $cover]     [expr -$z1]             [expr $y1]              [expr $z1]

    # Core concrete
    patch rect $matTagConcCore  7   1   [expr $cover - $y1]     [expr $cover - $z1]     [expr $y1 - $cover]     [expr $z1 - $cover]

    # ---------------------------------------------------------------------------------
    # Create straight layers
    # ---------------------------------------------------------------------------------

    # Reinforcing steel
    layer straight $matTagSteel  4  $As [expr $y1 - $cover] [expr $z1 - $cover] [expr $y1 - $cover] [expr $cover - $z1]
    layer straight $matTagSteel  4  $As [expr $cover - $y1] [expr $cover - $z1] [expr $cover - $y1] [expr $z1 - $cover]

}
```



Fiber Section Plotter

optional

```
[ ] = plot_fiberSection(secDefFilePath, secTag, figNum, fibColor)
```

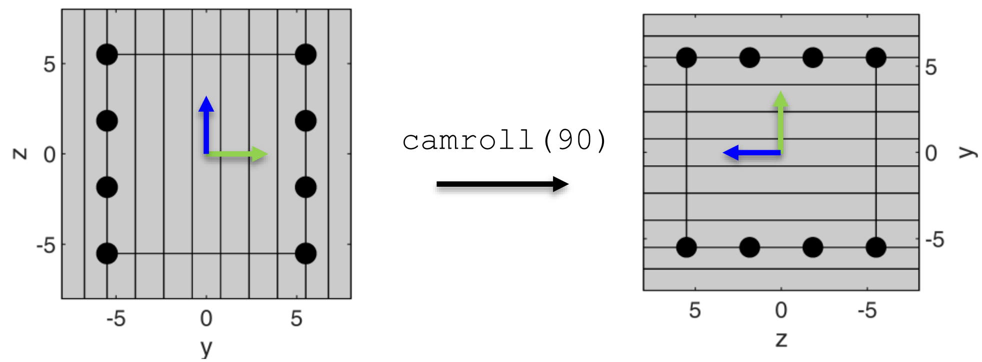
plot_fiberSection.m

modelData.txt

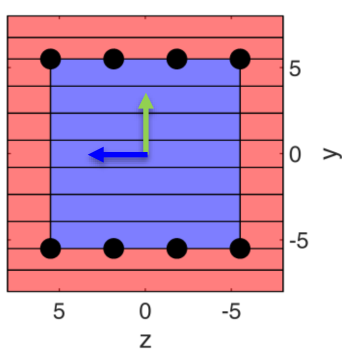
```
section Fiber 1 -GJ 1.e12 {

  # ------------------------------------------------------------------------------------------
  # Create rectangular patches
  # ------------------------------------------------------------------------------------------

  # Cover concrete
  patch rect 1   7   1   -5.5    -8.0     5.5    -5.5
  patch rect 1   7   1   -5.5     5.5     5.5     8.0
  patch rect 1   2   1   -8.0    -8.0    -5.5     8.0
  patch rect 1   2   1    5.5    -8.0     8.0     8.0

  # Core concrete
  patch rect 2   7   1   -5.5    -5.5     5.5     5.5

  # ------------------------------------------------------------------------------------------
  # Create straight layers
  # ------------------------------------------------------------------------------------------

  # Reinforcing steel
  layer straight 3  4  1.0    5.5   5.5   5.5   -5.5
  layer straight 3  4  1.0   -5.5  -5.5  -5.5    5.5

}
```

Example MATLAB driver

```
secDefFilePath = "…/modelData.txt";
secTag = 1;
figNum = 1;
plot_fiberSection(secDefFilePath, secTag, figNum)
grid off;
camroll(90)
```




```
fibColor = [1 1 0 0 0.5;
            2 0 0 1 0.5;
            3 0 0 0 1];
plot_fiberSection(secDefFilePath, secTag, figNum, fibColor)
grid off;
camroll(90)
```



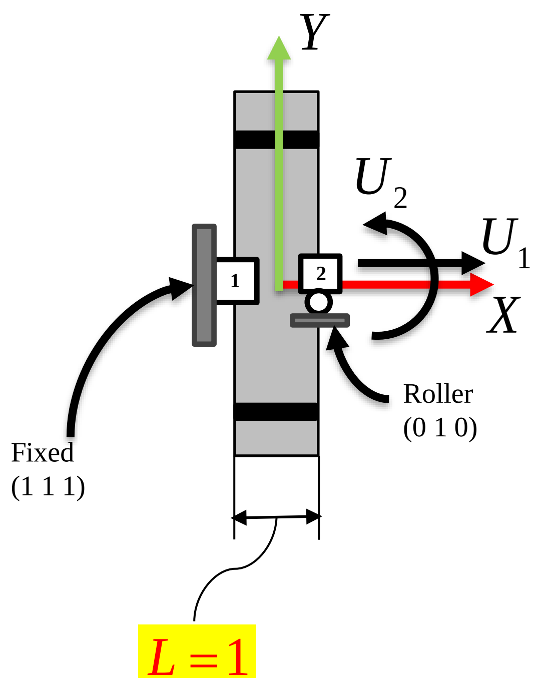
zeroLengthSection Element in OpenSees

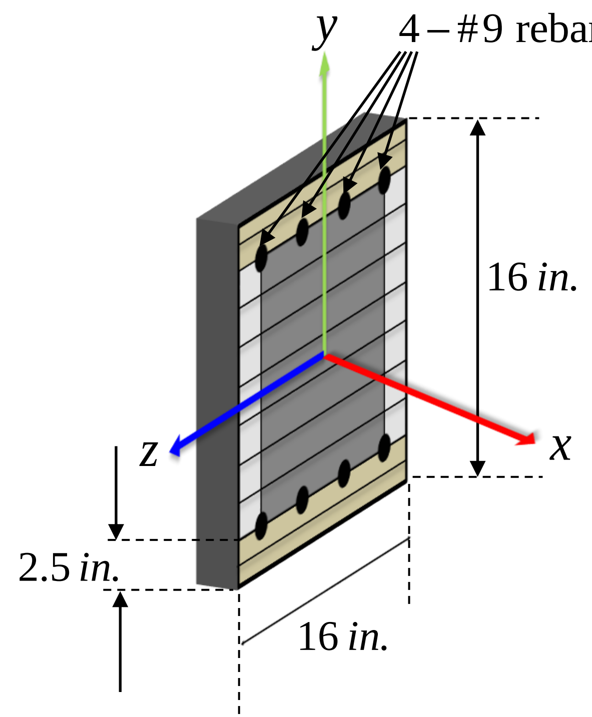
```
element zeroLengthSection $eleTag $iNode $jNode $secTag <-orient $x1 $x2 $x3 $yp1 $yp2 $yp3>
```

- If the optional orientation vectors are not specified, the local element axes coincide with the global axes.

$$\boldsymbol{U} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} L\,\varepsilon_o \\ L\,\phi_z \end{bmatrix} = \begin{bmatrix} \varepsilon_o \\ \phi_z \end{bmatrix}$$

$$\boldsymbol{P} = \begin{bmatrix} N \\ M \end{bmatrix}$$





Hybrid Loading Scheme – Load Control Application of Axial Load

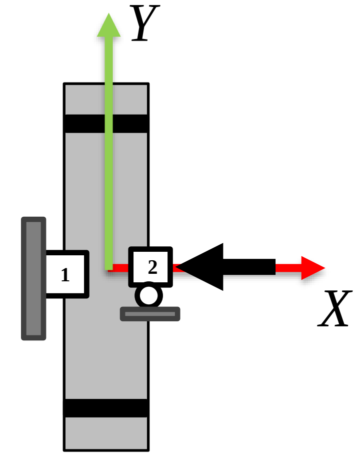
```
# Define axial load
set axialLoadTag 1
set axialLoadRatio 0.1
set P [expr abs($fpc)*$colArea*$axialLoadRatio]

pattern Plain $axialLoadTag "Linear" {
    load $controlNode -$P 0.0 0.0;
}
```

```
# Define load control integrator
set numAnalysisSteps 1
integrator LoadControl [expr 1./$numAnalysisSteps]; # Note the use of 1.
```

```
# Analyze
system BandGeneral
test NormUnbalance 1e-6 100
numberer Plain
constraints Plain
algorithm KrylovNewton
analysis Static

set ok [analyze $numAnalysisSteps]

if {$ok == 0} {
    puts "Axial load applied and analyzed"
}
```

```
# Very important to set
loadConst -time 0.0
```

$$\mathbf{R}(\mathbf{U}) = \begin{bmatrix} P \\ 0 \end{bmatrix}$$



Load control application of pattern defined above

Maintain axial load and reset pseudo-time $(= \text{load factor}, \lambda)$ to zero

Hybrid Loading Scheme – Displacement Control Application of Curvature

```
set controlNode 2
set controlDOF 3

# Define reference moment
set dispControlLoadTag 2
pattern Plain $dispControlLoadTag "Linear" {
    load $controlNode 0.0 0.0 1.0
}

# Define displacement control integrator
set peakDisp {}
lappend peakDisp [expr 0.01/$colDepth]
lappend peakDisp [expr -0.01/$colDepth]
lappend peakDisp [expr 0.03/$colDepth]
lappend peakDisp [expr -0.03/$colDepth]
lappend peakDisp [expr 0.0/$colDepth]

set maxDisp [expr 0.03/$colDepth]
set du [expr 0.01*$maxDisp]
integrator DisplacementControl $controlNode $controlDOF $du

recorder Node -file $analysisResultsDirectory/MK.txt -time -node $controlNode -dof 1 $controlDOF disp

set numAnalysisSteps [expr int(abs($maxDisp/$du))]
analyze $numAnalysisSteps
```

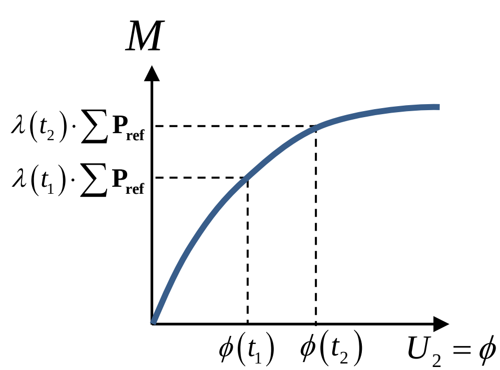
$$\mathbf{R}(\mathbf{U}) = \begin{bmatrix} P \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

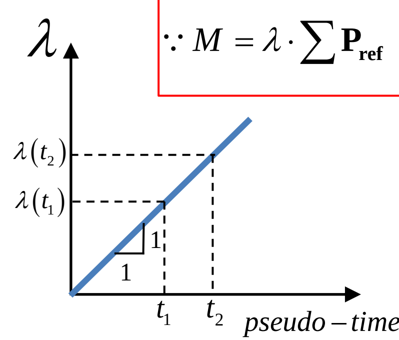
Apply a displacement increment $du at DOF #3 of node #2

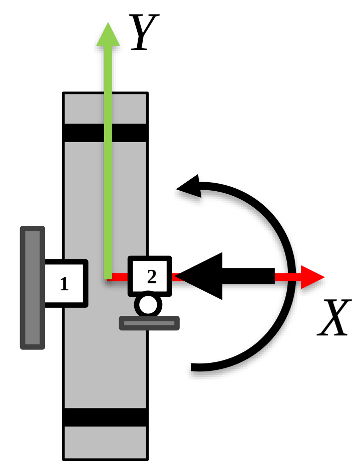
$= \lambda$

$= M$

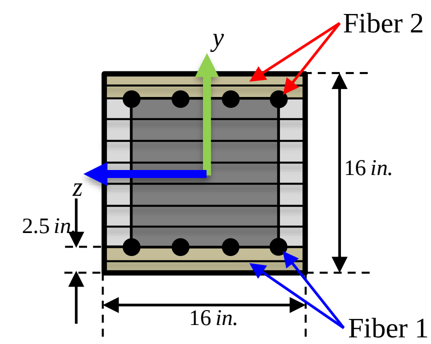
$\because M = \lambda \cdot \sum \mathbf{P}_{\text{ref}}$ and $\mathbf{P}_{\text{ref}} = \begin{bmatrix} N \\ M \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$



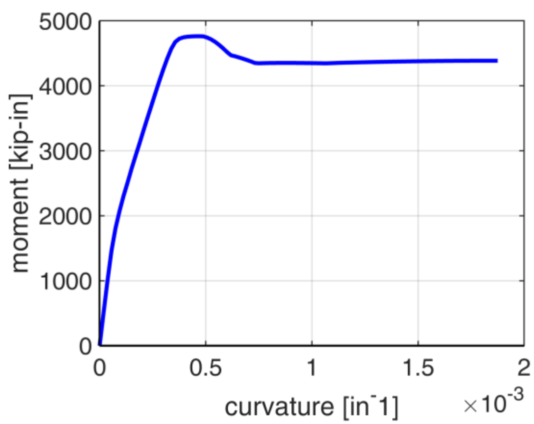




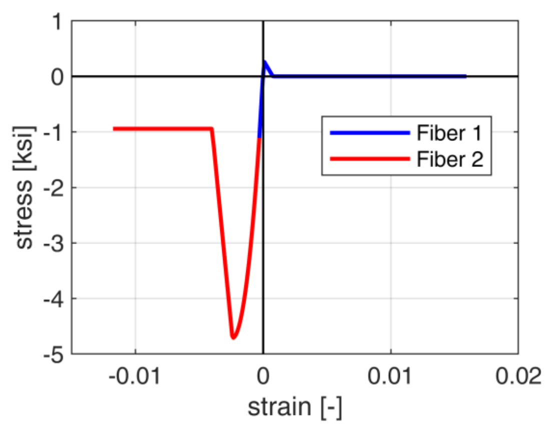
Results



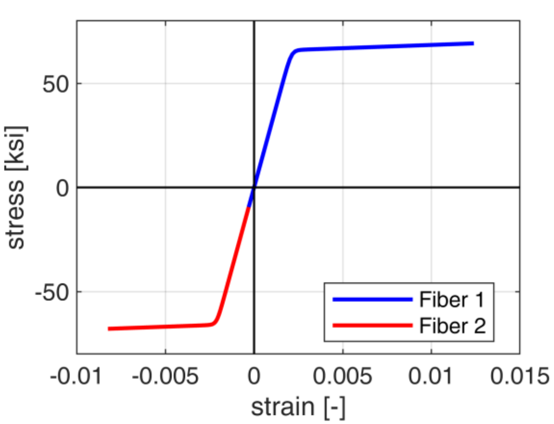

```
recorder Node -file $analysisResultsDirectory/MK.txt -time -node $controlNode -dof 1 $controlDOF disp

recorder Element -file $analysisResultsDirectory/ConcFib1_SS.txt  -time -ele 1 section fiber -$y1  0. $matTagConcCover stressStrain
recorder Element -file $analysisResultsDirectory/ConcFib2_SS.txt  -time -ele 1 section fiber  $y1  0. $matTagConcCover stressStrain
recorder Element -file $analysisResultsDirectory/SteelFib1_SS.txt -time -ele 1 section fiber -$y1  0. $matTagSteel      stressStrain
recorder Element -file $analysisResultsDirectory/SteelFib2_SS.txt -time -ele 1 section fiber  $y1  0. $matTagSteel      stressStrain

record; # This is to record the state before the analysis starts
```




```
plot(MK(:,3),MK(:,1),'b-')
```

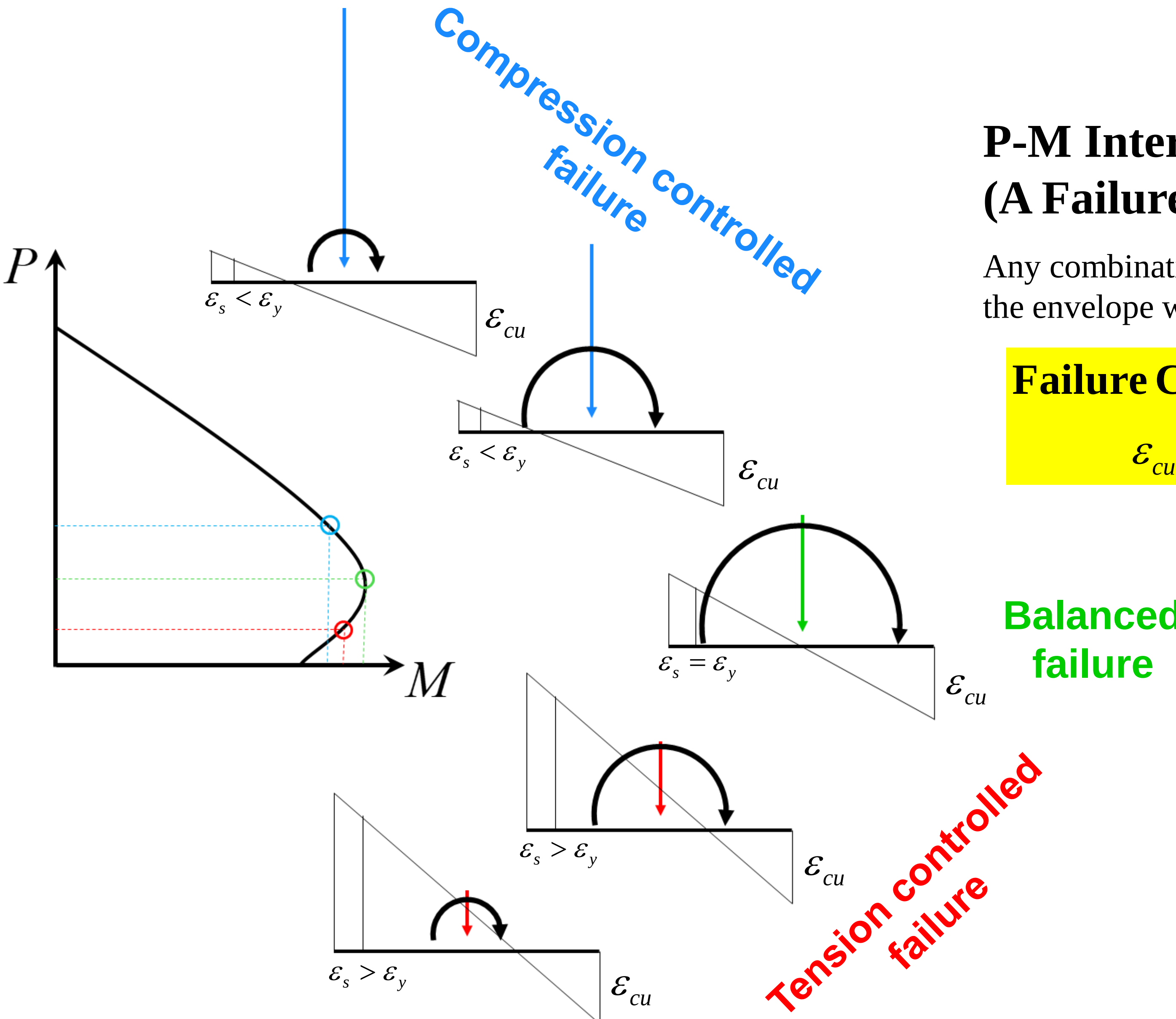



```
plot(ConcFib1_SS(:,3),ConcFib1_SS(:,2),'b-')
plot(ConcFib2_SS(:,3),ConcFib2_SS(:,2),'r-')
```




```
plot(SteelFib1_SS(:,3),SteelFib1_SS(:,2),'b-')
plot(SteelFib2_SS(:,3),SteelFib2_SS(:,2),'r-')
```

PM Interaction Diagrams



P-M Interaction Diagram (A Failure Envelope)

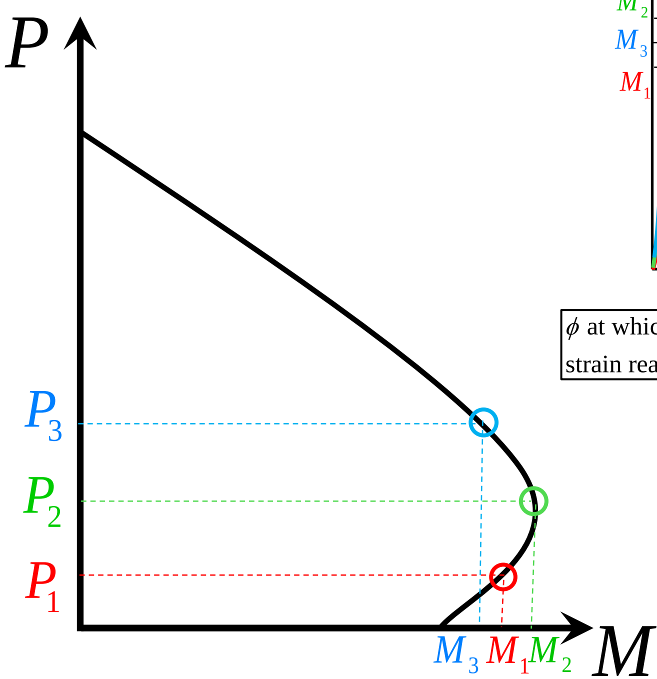
Any combination of P and M outside the envelope will cause failure.

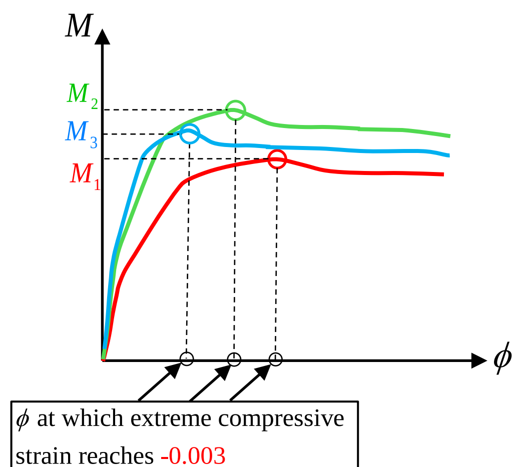
Failure Criterion (ACI):

$$\varepsilon_{cu} = -0.003$$

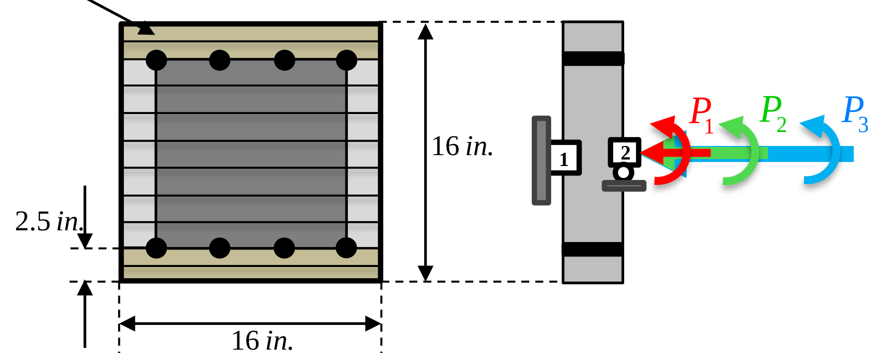
Development of PM Diagrams in OpenSees – Analysis Procedure

For various levels of axial load held constant, increase curvature of the section until a concrete strain of **-0.003** is reached

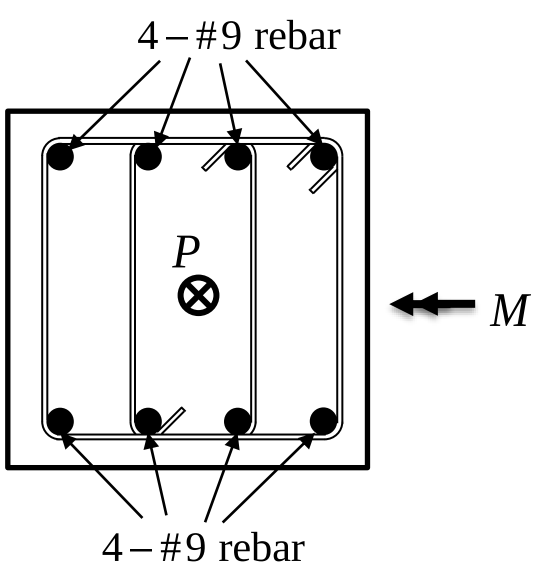


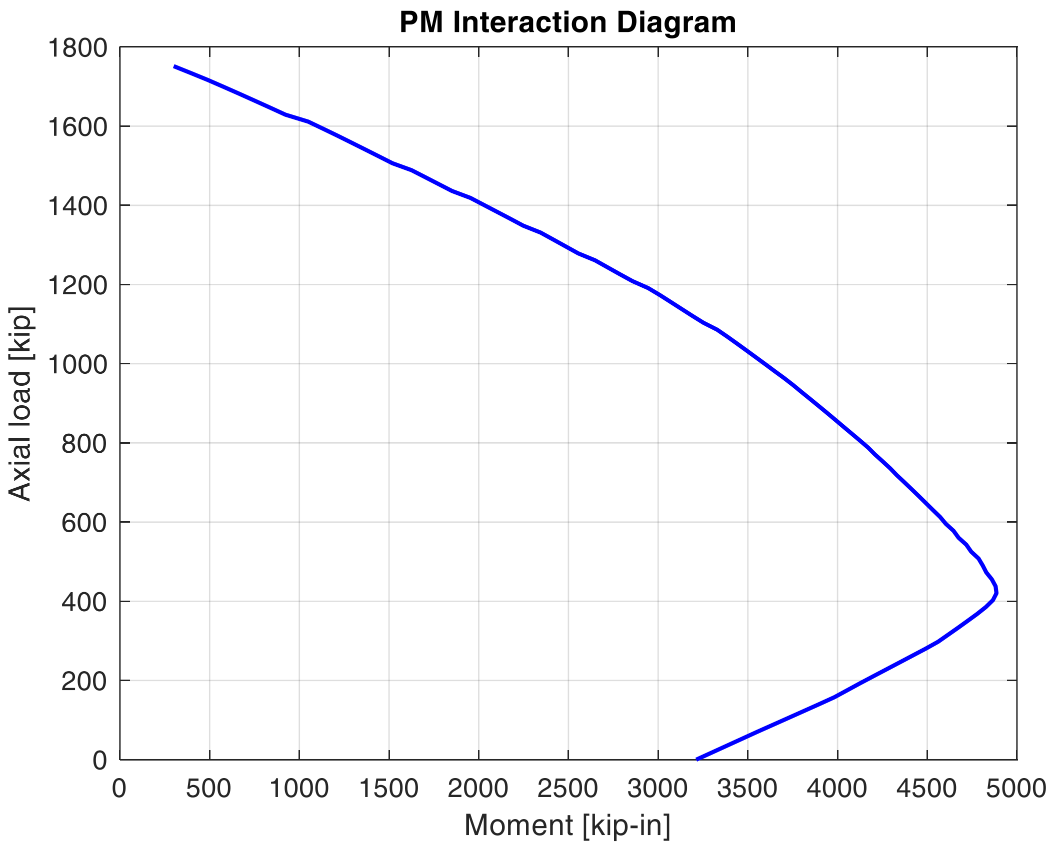


For a given P, apply ϕ and calculate M i.e., perform a displacement-controlled $M-\phi$ analysis



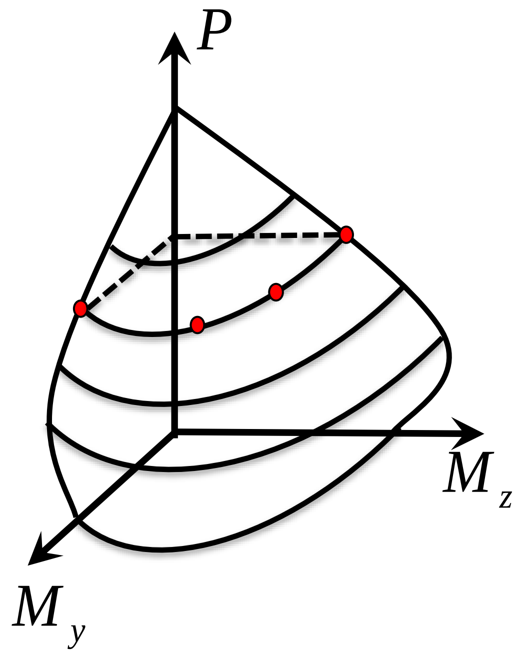
PM Interaction Diagram

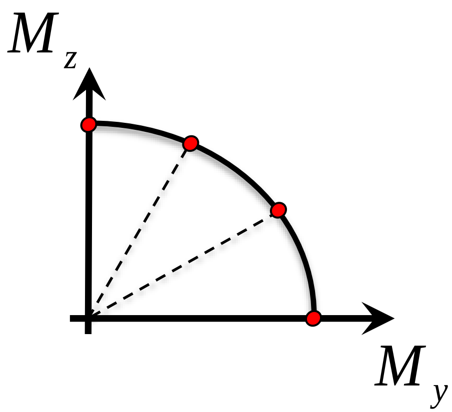




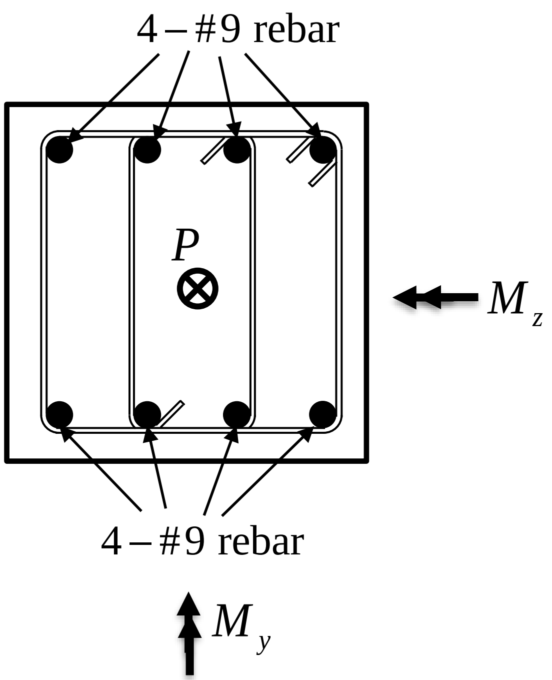
PMM - Same Concept Extended to 3D

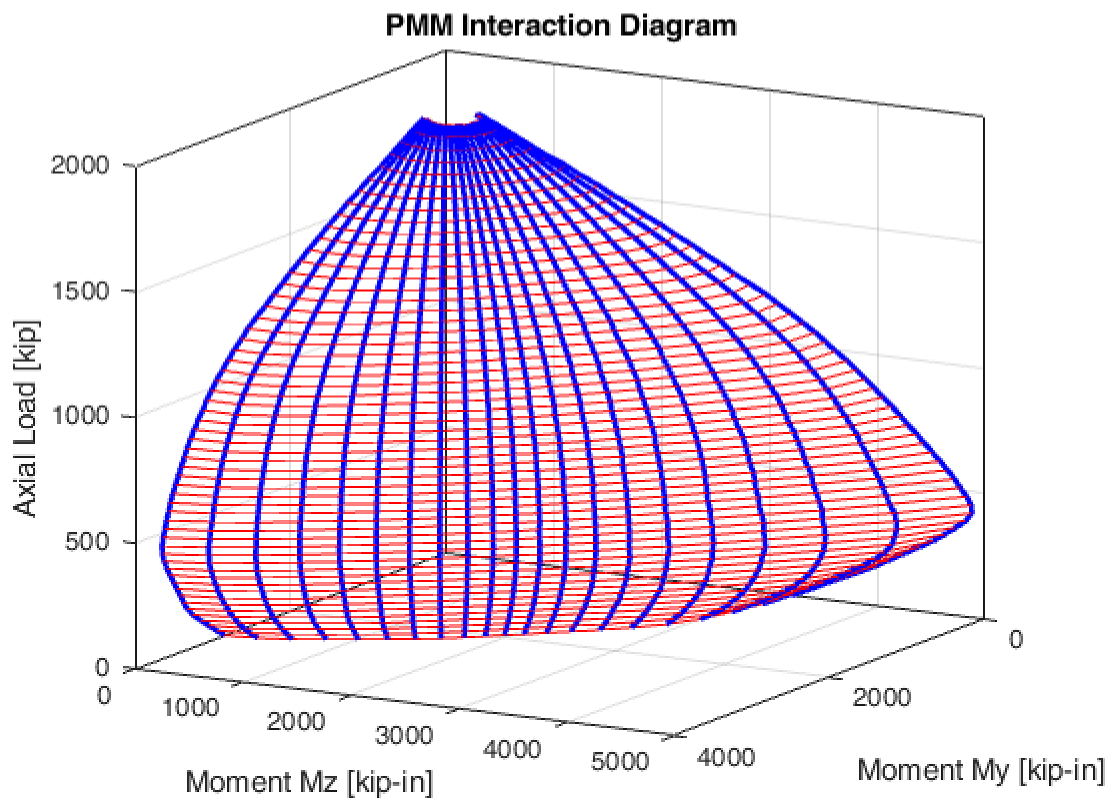
– For various levels of axial load, increase curvature along radial line to reach failure surface.





PMM - Same Concept Extended to 3D





Questions?