OpenSees

Nonlinear Analysis
With Simple Examples

Frank McKenna
UC Berkeley

OpenSees Days Shanghai 2011

NEES NEEScomm

Outline of Presentation

- Why Nonlinear Analysis
- OpenSees Analysis Options in More Depth
- Nonlinear Beam-Column Modeling & Examples

Why Nonlinear Analysis

•**Geometric Nonlinearities** - occur in model when applied load causes large displacement and/or rotation, large strain, or a combo of both

•**Material nonlinearities** - nonlinearities occur when material stress-strain relationship depends on load history (plasticity problems), load duration (creep problems), temperature (thermoplasticity), or combo of all.

•**Contact nonlinearities** - occur when structure boundary conditions change because of applied load.

Nonlinear Analysis is Harder

•It requires **much** more thought when setting up the model

•It requires more thought when setting up the analysis

•It takes more computational time.

•It does not always converge.

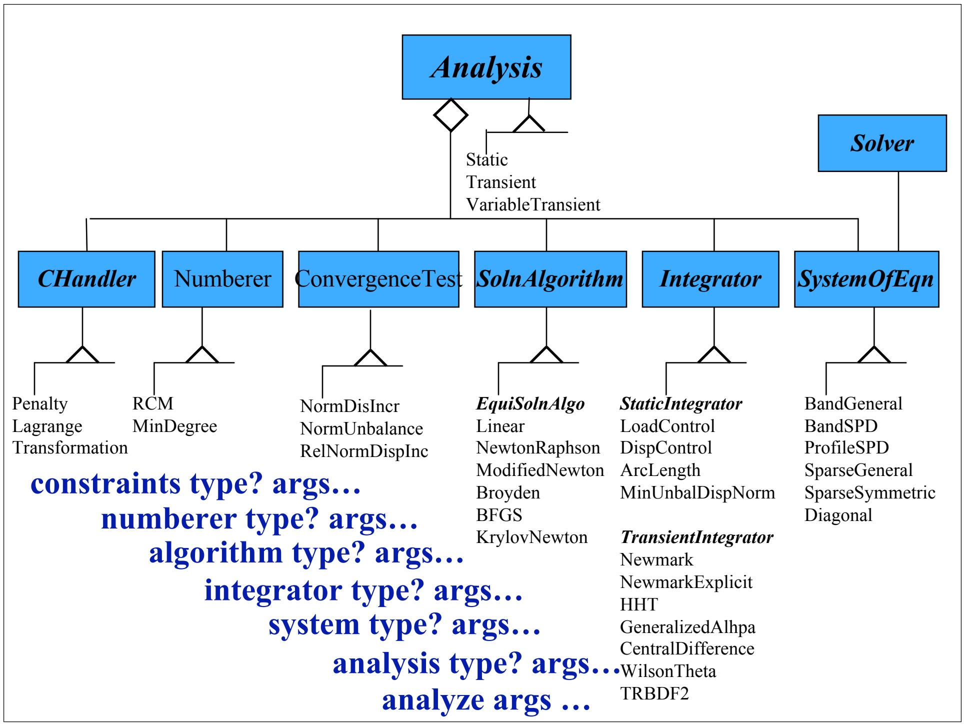
•It does not always converge to the correct solution.

BUT Most Problems Require Nonlinear Analysis

Remember that nonlinear analysis does not always converge

CHECK YOUR MODEL

CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL



test command:

- to specify when convergence has been achieved

all look at system: **KU = R**

•Norm Unbalance

$\sqrt{\mathbf{R^\wedge R}} < \mathbf{tol}$

test NormUnbalance tol? numIter? <flag?>

•Norm Displacement Increment

$\sqrt{\mathbf{U^\wedge U}} < \mathbf{tol}$

test NormDispIncr tol? numIter? <flag?>

•Norm Energy Increment

½ (U^R) < tol

test NormEnergyIncr tol? numIter? <flag?>

•Relative Tests

test RelativeNormUnbalance tol? numIter? <flag?>

test RelativeNormDispIncr tol? numIter? <flag?>

test RelativeNormEnergyIncr tol? numIter? <flag?>

numberer command:

- to specify how the degrees of freedom are numbered

•Plain Numberer

nodes are assigned dof arbitrarily

numberer Plain

•RCM Numberer

nodes are assigned dof using the Reverse Cuthill-McKee algorithm

numberer RCM

•AMD Numberer

nodes are assigned dof using the Approx. MinDegree algorithm

numberer AMD

• numbering has an impact on performance of banded and profile solvers. The sparse solvers all use their own optimal numbering schemes.

integrator command:

-determines the predictive step for time t+δt

-specifies the tangent matrix and residual vector at any iteration

-determines the corrective step based on ΔU

•Transient Integrators for Use in Transient Analysis

Nonlinear equation of the form:

$$\mathbf{R}(\mathbf{U}, \dot{\mathbf{U}}, \ddot{\mathbf{U}}) = \mathbf{P}(t) - \mathbf{F}_I(\ddot{\mathbf{U}}) - \mathbf{F}_R(\mathbf{U}, \dot{\mathbf{U}})$$

▪CentralDifference

integrator CentralDifference

▪Newmark Method

integrator Newmark γ β

▪Hilbert-Hughes-Taylor Method (alpha between 0.5 and 1.0)

integrator HHT α <γ β>

▪Alpha Operator Splitting Method

integrator AlphaOS α

•Static Integrators for Use in Static Analysis

Nonlinear equation of the form:

$$\mathbf{R(U, \lambda) = \lambda P^* - FR(U)}$$

▪Load Control

$\lambda_n = \lambda_{n-1} + \Delta\lambda$ *integrator LoadControl Δλ*

*does not require a reference load, i.e. loads in load patterns with Linear series and all other loads constant.

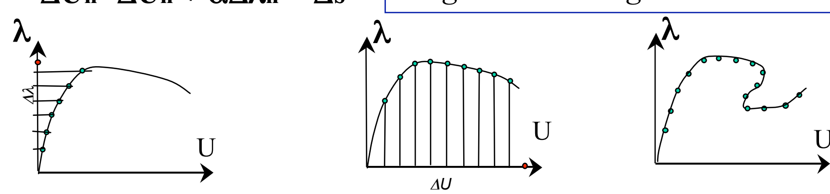
▪Displacement Control

$Uj_n = Uj_{n-1} + \Delta Uj$

integrator DisplacementControl node dof ΔU

▪Arc Length

$\Delta U_n \hat{} \Delta U_n + \alpha^2 \Delta\lambda_n = \Delta s^2$ *integrator ArcLength α Δs*



algorithm command:

- to specify the steps taken to solve the nonlinear equation

•Linear Algorithm

algorithm Linear

```
theIntegrator->formUnbalance();
theIntegrator->formTangent();
theSOE->solve()
theIntegrator->update(theSOE->getX());
```

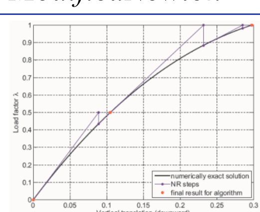
•Newton-Raphson Algorithm

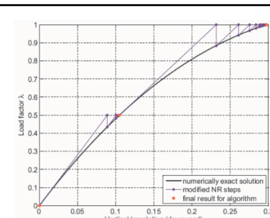
algorithm Newton

```
theIntegrator->formUnbalance();
do {
  theIntegrator->formTangent();
  theSOE->solve()
  theIntegrator->update(theSOE->getX());
  theIntegrator->formUnbalance();
} while (theTest->test() == fail)
```

•Modified Newton Algorithm

algorithm ModifiedNewton <-initial>





•Accelerated Modified Newton Algorithm

algorithm KrylovNewton <-initial>

constraints command:

- to specify how the constraints are enforced

$$\mathbf{U}_c = \mathbf{C}_{rc}\,\mathbf{U}_r$$

$$\mathbf{C}\,\mathbf{U} = \mathbf{0} \qquad [\mathbf{C}_r \;\; \mathbf{C}_c]^{\wedge}[\mathbf{U}_r \;\; \mathbf{U}_c] = 0$$

$$\mathbf{T}\,\mathbf{U}_r = [\mathbf{U}_r \;\; \mathbf{U}_c]^{\wedge}$$

•Transformation Handler

$$K^* \, U_r = R^* \qquad K^* = T^{\wedge}KT \qquad R^* = T^{\wedge}R$$

constraints Transformation

in OpenSees currently don't allow retained node in one constraint to be a constrained node in another constraint

•Lagrange Handler

$$\begin{bmatrix} \mathbf{K} & \mathbf{C}^{\wedge} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{Q} \end{bmatrix}$$

constraints Lagrange

•Penalty Handler

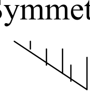
$$[\mathbf{K} + \mathbf{C}^{\wedge}\boldsymbol{\alpha}\mathbf{C}]\,\mathbf{U} = [\mathbf{R} + \mathbf{C}^{\wedge}\boldsymbol{\alpha}\mathbf{Q}]$$

constraints Penalty α_{sp}? α_{mp}?

system command:

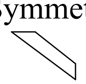
- to specify how matrix equation KU = R is stored and solved

•Profile Symmetric Positive Definite (SPD)



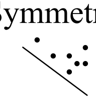
system ProfileSPD

•Banded Symmetric Positive Definite



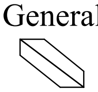
system BandSPD

•Sparse Symmetric Positive Definite



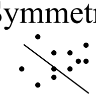
system SparseSPD

•Banded General



system BandGeneral

•Sparse Symmetric



system SparseGeneral

system Umfpack

analysis command:

•Static Analysis — *analysis Static*

•Transient Analysis — *analysis Transient*

- both incremental solution strategies

StaticAnalysis
analyze()

```
for (int i=0; i<numIncr; i++) {
  theIntegrator->newStep();
  theAlgorithm->solveCurrentStep();
  theModel->commit();
}
```

TransientAnalysis
analyze()

```
for (int i=0; i<numIncr; i++) {
  theIntegrator->newStep(dt);
  theAlgorithm->solveCurrentStep();
  theModel->commit();
}
```

•Eigenvalue

▪general eigenvalue problem

$$(\mathbf{K}-\lambda\mathbf{M})\Phi=0$$

eigen numModes? -general

▪standard eigenvalue problem

$$(\mathbf{K}-\lambda)\Phi=0$$

eigen numModes? -standard

analyze command:

- to perform the static/transient analysis

•Static Analysis

StaticAnalysis
analyze()

```
for (int i=0; i<numIncr; i++) {
  theIntegrator->newStep();
  theAlgorithm->solveCurrentStep();
  theModel->commit();
}
```

analyze numIter?

•Transient Analysis

TransientAnalysis
analyze()

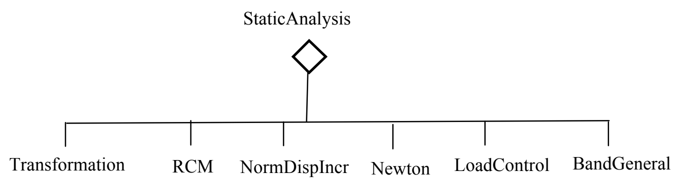
```
for (int i=0; i<numIncr; i++) {
  theIntegrator->newStep(dt);
  theAlgorithm->solveCurrentStep();
  theModel->commit();
}
```

analyze numIter? Δt?

Example Static Analysis:

•Static Nonlinear Analysis with LoadControl

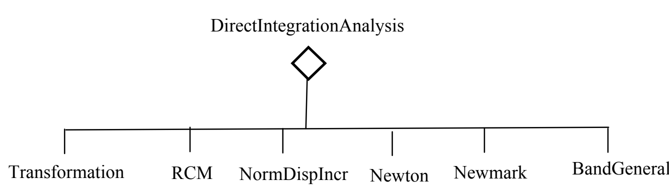
```
constraints Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator LoadControl 0.1
analysis Static
analyze 10
```



Example Dynamic Analysis:

•Transient Nonlinear Analysis with Newmark

```
constraints Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient
analyze 2000 0.01
```



Remember that nonlinear analysis does not always converge

CHECK YOUR MODEL

CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL

Commands that Return Values

•analyze command

The analyze command returns 0 if successful.
It returns a negative number if not

set ok [analyze numIter <Δt>]

•getTime command

The getTime command returns pseudo time in Domain.

set currentTime [getTime]

•nodeDisp command

The nodeDisp command returns a nodal displacement.

set disp [nodeDisp node dof]

Example Usage – Displacement Control

```
set maxU 15.0; set dU 0.1
constraints transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator DispControl  3 1 $dU
analysis Static
set ok 0
set currentDisp 0.0
while {$ok == 0 && $currentDisp < $maxU} {
        set ok [analyze 1]
        if {$ok != 0} {
           test NormDispIncr 1.0e-6 1000 1
           algorithm ModifiedNewton –initial
           set ok [anal;yze 1]
           test NormDispIncr 1.0e-6 6 2
           algorithm Newton
        }
        set currentDisp [nodeDisp 3 1]
}
```

Example Usage – Transient Analysis

```
set tFinal 15.0;
constraints  Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient
set ok 0
set currentTime 0.0
while {$ok == 0 && $currentTime < $tFinal} {
        set ok [analyze 1 0.01]
        if {$ok != 0} {
          test NormDispIncr 1.0e-6 1000 1
          algorithm ModifiedNewton –initial
          set ok [analyze 1 0.01]
          test NormDispIncr 1.0e-6 6 2
          algorithm Newton
        }
        set currentTime [getTime]
}
```

Still Not Working!

1. Search the Message Board
2. Post Problem on the Message Board

To check which scale of elcentro earthquake makes the SDF inelastic, the following file was used. By trial and error, scale 10 was found in OpenSees, which is inconsistent with the results(scale 4) from using other programs.

Is there anything wrong in this file?

```
# create ModelBuilder (with two-dimensions and 2 DOF/node)
model BasicBuilder -ndm 1 -ndf 1


# Define geometry for model
# ------------------------
puts "Define geometry for model"
set k1 2.75
set uy 1.35
```

i suggest you check your other input files .. if you have a look at chopra's book he plots the respone spectrum for this e.q. .. for a period of 0.1, D for an elastic system is with 0% damping is about .11 (fig 6.8.1 in my version) .. so you need a scale factor of about 12 [1.35/.11] to reach the ultimate.
(note using Newmark 0.5 0.25 you get .11)

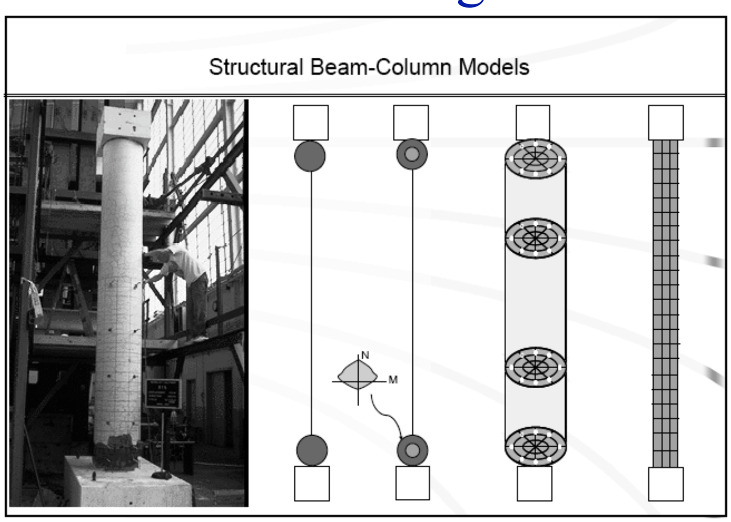
to compute the scale factor for yield i suggest you also stop playing with trying to predict the scale factor & just divide yield disp by the max response from elastic system.

Segmentation Faults, etc:

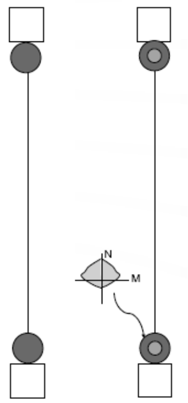
•Email: fmckenna@ce.berkeley.edu

NOTE: Zip up your files in 1 directory and send them to us

Nonlinear Beam Columm Modeling

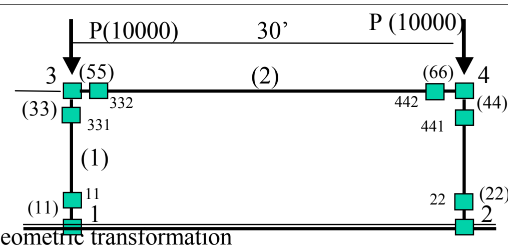


Concentrated Plasticity Models



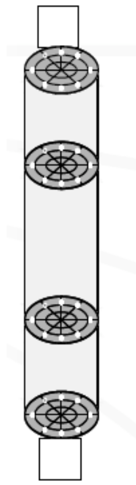
Advantages:
•Simple
•Good for Interface Effects (bar pullout, shear sliding)

Disadvantages:
•Properties of springs depend on geometry &
Moment distribution
•Force-Displacement relationship of element needs
to be related to Plastic Hinge Length & elastic elemen
•Properties of elastic element




```
# create modelbuilder
model BasicBuilder -ndm 2 -ndf 3;
# create 10 nodes
node 1 0.0   0.0
node 2 360.0 0.0
node 3 0.0   180.0
node 4 360.0 180.0
node 11  0.0   0.0
node 22  360.0 0.0
node 331 0.0   180.0
node 332 0.0   180.0
node 441 360.0 180.0
node 442 360.0 180.0
# Single point constraints
fix  1 1 1 1
fix  2 1 1 1
fix 11 1 1 0
fix 22 1 1 0
# Multi point constraints
equalDOF 3 331 1 2
equalDOF 3 332 1 2
equalDOF 4 441 1 2
equalDOF 4 442 1 2

# Define Materials
uniaxialMaterial Hysteretic 1 ...
uniaxialMaterial Hysteretic 2...

# define geometric transformation
geomTransf Linear 1

# Define Elements -----------------------------------------------------
# elastic elements:
element elasticBeamColumn 1 11 331  109.0 29000.0 10880.0 1;
element elasticBeamColumn 2 22 441  109.0 29000.0 10880.0 1;
element elasticBeamColumn 3 332 442 41.6  29000.0 14900.0 1;

# rotational hinges:
# columns
element zeroLength 11 1 11  -mat 1 -dir 6
element zeroLength 22 2 22  -mat 1 -dir 6
element zeroLength 33 3 331 -mat 1 -dir 6
element zeroLength 44 4 441 -mat 1 -dir 6
# beam
element zeroLength 55 3 332 -mat 2 -dir 6
element zeroLength 66 4 442 -mat 2 -dir 6
```

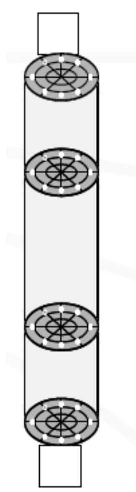
Distributed Plasticity



- Permits the spread of plasticity along the element
  - Four or five Gauss-Lobatto points is usually sufficient
  - If no strength degradation, converges to unique solution as N_p increases
- Allows yielding to occur at any section along the element, which is important in the presence of distributed element loads
  - Girders with high gravity loads

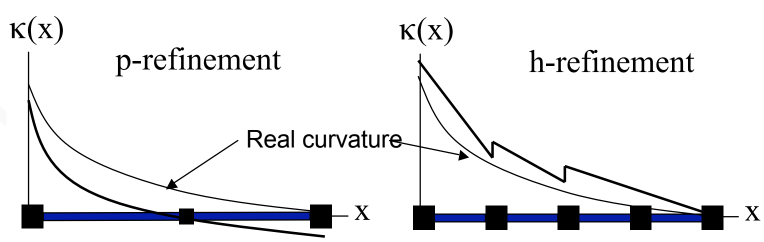
- Integration weights from optimality constraints for the integration of high order polynomials
  - Do not reflect plastic hinge lengths from experiments or observed structural damage
  - Non-objective localized response when N_p changes (force based)
- Computation and memory overhead when yielding does not occur on the element interior
  - Typical in frame structures where columns are in double curvature (fix - beamWithHinges)

Distributed Plasticity Displacement Based Formulation

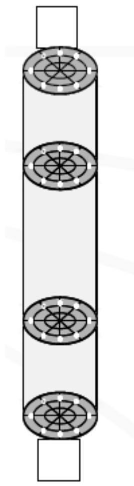


element dispBeamColumn $eleTag …

- Constant axial deformation and linear curvature distribution enforced along the element length
  - Exact only for prismatic, linear elastic elements (Hermitian)
- Weak equilibrium leads to errors in force boundary conditions … internal forces not in equilibrium with external reactions
- Requires *p*-refinement or *h*-refinement to represent higher order distributions of deformations



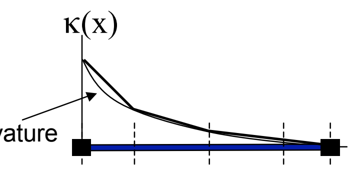
Distributed Plasticity
Force Based Formulation



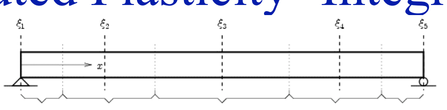
element forceBeamColumn $eleTag …

- Section forces determined from basic forces by interpolation within the basic system
  - Interpolation from static equilibrium
  - Constant axial force, linear distribution of bending moment in the absence of distributed element loads
- Equilibrium between element and section forces is exact, which holds in the range of constitutive nonlinearity
- Use Principle of Virtual Forces & Integrate section deformations along the element to get the element deformations

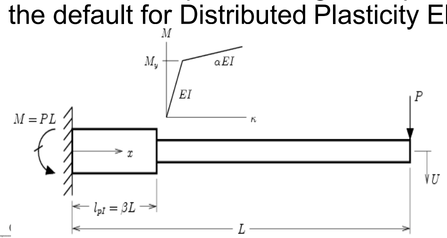
$$\mathbf{v} = \int_0^L \mathbf{b}^T \mathbf{e}\, dx$$

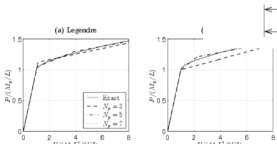


Distributed Plasticity -Integration Methods

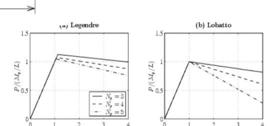


- Behaviour of element depends on choice of integration method
- Number of options: Trapezoidal, Mid-Point, and others based on gauss-quadrature (Gauss-Legendre,Gauss-Lobotto, Gauss-radau)
- Of these, only Gauss-Lobotto places integration points at ends of the element. This is the default for Distributed Plasticity Elements.





Increasing Strength



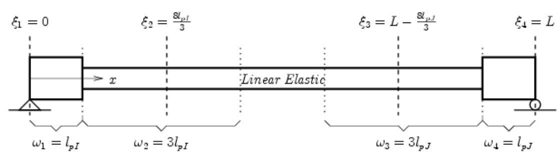
Decreasing Strength

NON-OBJECTIVE

BeamWithHinges Element

```
element beamWithHinges $eleTag …
```

- The force based distributed plasticity element with inelastic sections at ends and elastic sections elsewhere, uses Gauss-Radau
- User specified hinge length.
- It is Objective even for stiffness degredation

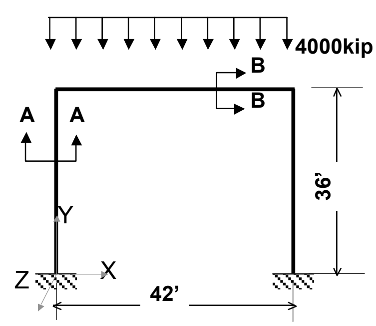


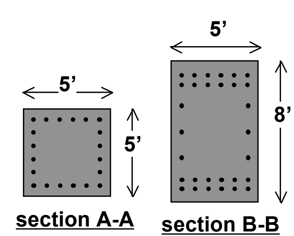
- BUT elastic in middle

Model

RCFrame.tcl

```
model Basic -ndm 2 -ndf 3
set ft 12.0
set in 1.0
set cm 0.3937
# Set parameters for overall model geometry
set width    [expr 42.0*$ft]
set height   [expr 36.0*$ft]
# Create nodes
node  1      0.0     0.0
node  2   $width     0.0
node  3      0.0 $height
node  4   $width $height
# set boundary conditions
fix  1    1   1   1
fix  2    1   1   1
# create materials for sections
uniaxialMaterial Concrete01  1  -6.0 -0.004 -5.0 -0.014; # core
niaxialMaterial Concrete01  2  -5.0 -0.002  0.0 -0.006; # cover
uniaxialMaterial Steel01  3 60.0 30000.0 0.01

# create sections
set bWidth  [expr 5.0*$ft]; set bDepth  [expr 5.0*$ft]; set cover  1.5
set As    0.60;    # area of no. 7 bars
source RCsection2D.tcl
RCsection2D 1 $bWidth $bDepth $cover 1 2 3 6 3 \
   $As 10 10 2
#create geometric transformations
geomTransf PDelta 1
geomTransf Linear 2
# Create the coulumns usind distributed plasticity  elements
set np 5
set eleType forceBeamColumn
element $eleType  1   1   3   $np   1      1
element $eleType  2   2   4   $np   1      1
# Create the beam element using elastic element
set d [expr 8.0*$ft];          # Beam Depth
set b [expr 5.0*$ft];          # Beam Width
set Eb 3600.0; set Ab [expr $b*$d];; set Izb [expr ($b*pow($d,3))/12.0];
section Elastic 2   $Eb $Ab $Izb;      # elastic beam section
#element elasticBeamColumn  3   3   4   $Ab   $Eb  $Izb    2
element $eleType  3   3   4   $np   2      2
```





RCFrameGravity,tcl

Gravity Load Analysis

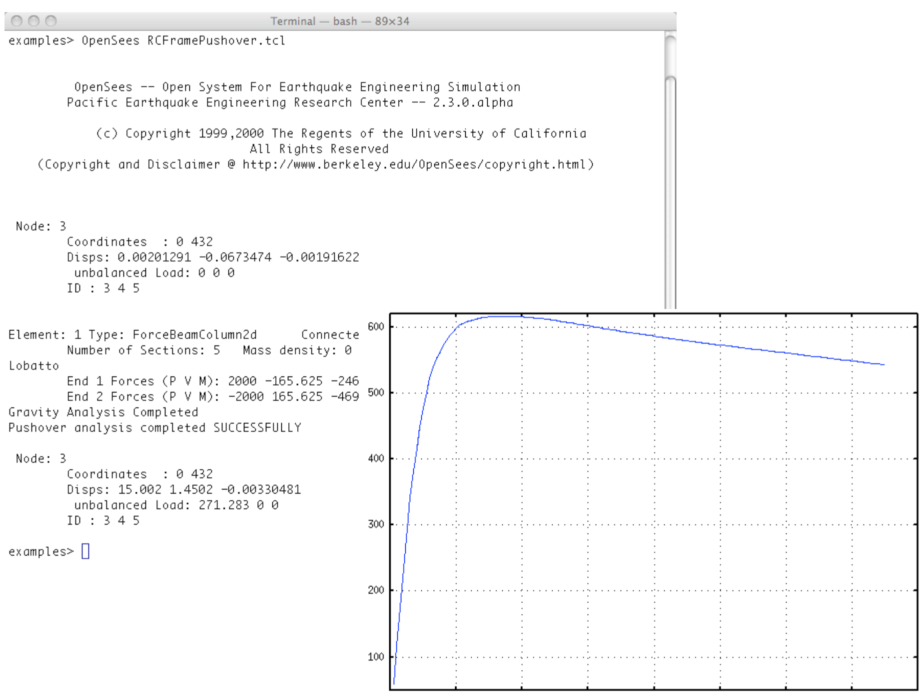
```
# first source in the model
source RCFrame.tcl

# Create the gravity loads
set W 4000.0;
timeSeries Linear 1
pattern Plain 1 1 {
   eleLoad -ele 3 -type -beamUniform [expr -$W/$width]
}

# create the analysis
system BandGeneral
constraints Transformation
numberer RCM
test NormDispIncr 1.0e-12  10 3
algorithm Newton
integrator LoadControl 0.1
analysis Static

# perform the analysis
analyze 10
```

Terminal — bash — 87×22

```
         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.3.0.alpha

            (c) Copyright 1999,2000 The Regents of the University of California
                                 All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)


 Node: 3
        Coordinates  : 0 432
        Disps: 0.00201291 -0.0673474 -0.00191622
         unbalanced Load: 0 0 0
        ID : 3 4 5


Element: 1 Type: ForceBeamColumn2d      Connected Nodes: 1 3
        Number of Sections: 5   Mass density: 0
Lobatto
        End 1 Forces (P V M): 2000 -165.625 -24646.3
        End 2 Forces (P V M): -2000 165.625 -46903.7
examples>
```

Pushover Analysis

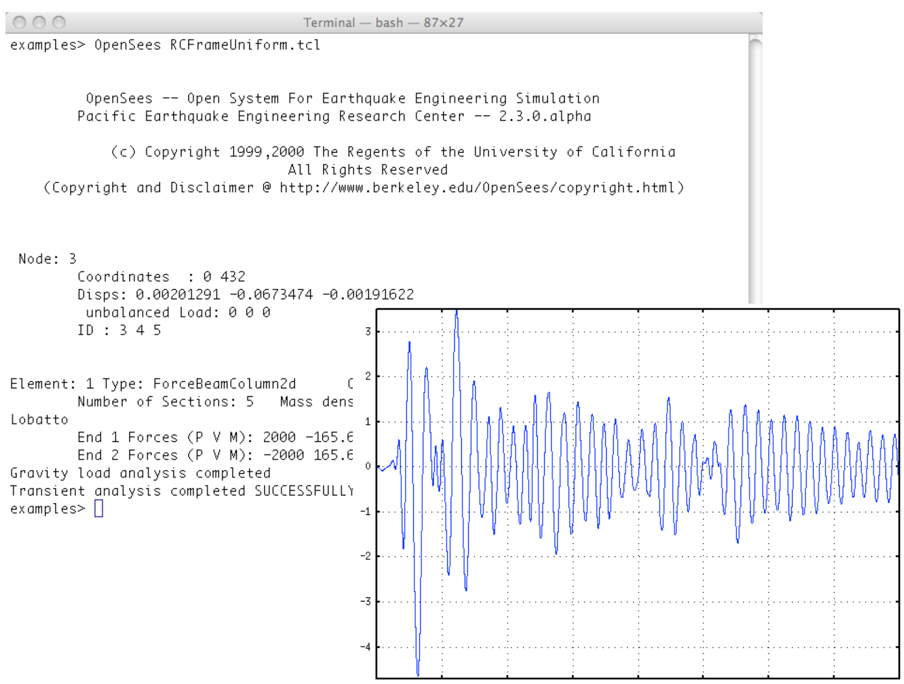
```
source RCFrameGravity.tcl
puts "Gravity Analysis Completed"

# Set the gravity loads to be constant & reset the time in the domain
loadConst -time 0.0

# Define Pattern for Lateral Reference loads
set H 10.0;
pattern Plain 2 1 {
    load 3 $H 0.0 0.0
    load 4 $H 0.0 0.0
}

set dU 0.1;
integrator DisplacementControl  3   1   $dU  1 $dU $dU

# Set some parameters
set maxU 15.0;          # Max displacement
set currentDisp 0.0;
set ok 0
while {$ok == 0 && $currentDisp < $maxU} {
      set ok [analyze 1]

      # if the analysis fails try initial tangent iteration
      if {$ok != 0} {
         puts "regular newton failed .. lets try an initail stiffness for this step"
         test NormDispIncr 1.0e-12  1000
         algorithm ModifiedNewton -initial
         set ok [analyze 1]
         test NormDispIncr 1.0e-12  10
         algorithm Newton
      }
      set currentDisp [nodeDisp 3 1]
}
if {$ok == 0} {
  puts "Pushover analysis completed SUCCESSFULLY";
```



Transient Analysis - Uniform Excitation

```
source RCFrameGravity.tcl
puts "Gravity load analysis completed"

# Set the gravity loads to be constant
#   & reset the time in the domain
loadConst -time 0.0

# Define nodal mass
set g 386.4
set m [expr ($W/2.0)/$g];

#    tag   MX    MY    RZ
mass  3    $m    $m    1.0e-16
mass  4    $m    $m    1.0e-16

# Define dynamic loads
set record IELC180
source ReadRecord.tcl
ReadRecord $record.AT2  $record.dat dT nPts
timeSeries Path 2 -filePath $record.dat -dt $dT
pattern UniformExcitation  2 1 -accel 2
rayleigh 0.0 0.0 0.0 0.0

#create a recorder
recorder Node -time -file disp.out -node 3 4 -dof 1 2 3 disp

# remove old analysis
wipeAnalysis
```

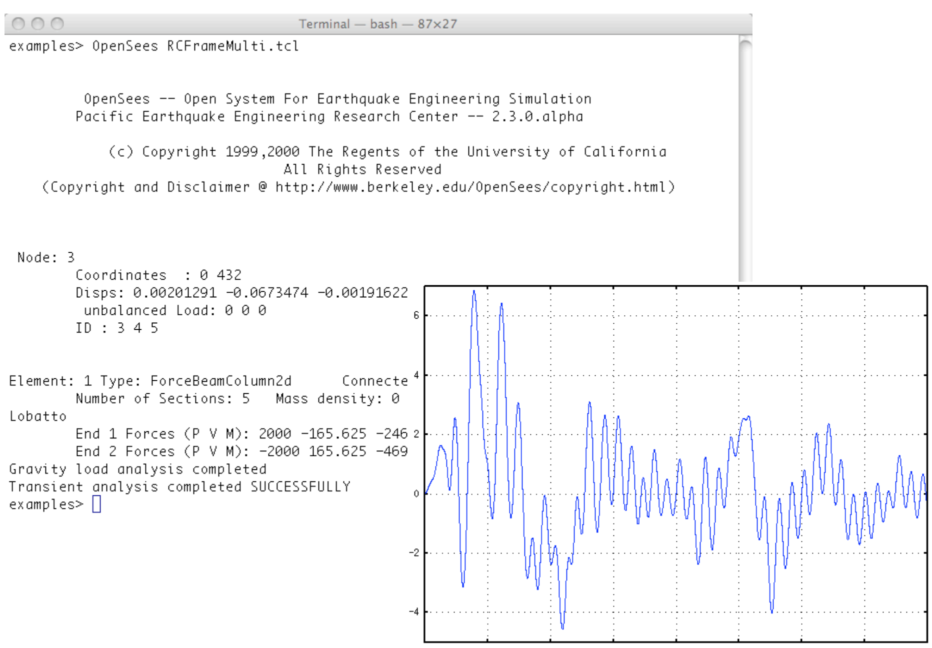
```
#create the analysis
system BandGeneral
constraints Plain
test NormDispIncr 1.0e-8  10
algorithm Newton
numberer RCM
integrator Newmark  0.5  0.25
analysis Transient

set tFinal [expr $nPts * $dT]
set tCurrent [getTime]
set ok 0
# perofrm the analysis
while {$ok == 0 && $tCurrent < $tFinal} {
   set ok [analyze 1 $dT]
   # if the analysis fails try initial tangent iteratio
   if {$ok != 0} {
      puts "regular newton failed .. lets try anothe
      test NormDispIncr 1.0e-8  1000 1
      algorithm ModifiedNewton -initial
      set ok [analyze 1 $dT]
      test NormDispIncr 1.0e-12  10
      algorithm Newton
   }
   set tCurrent [getTime]
}
# Print a message to indicate if analysis succesfu
if {$ok == 0} {
   puts "Transient analysis completed SUCCESSI
} else {
   puts "Transient analysis completed FAILED";
}
```

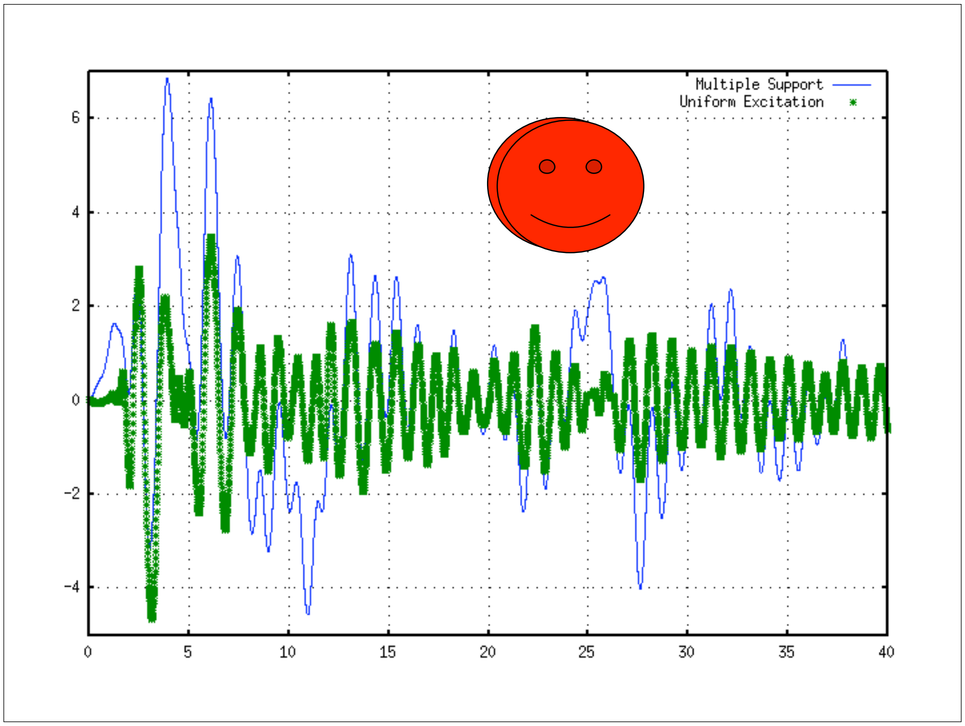


Transient Analysis - MultiSupport Excitation

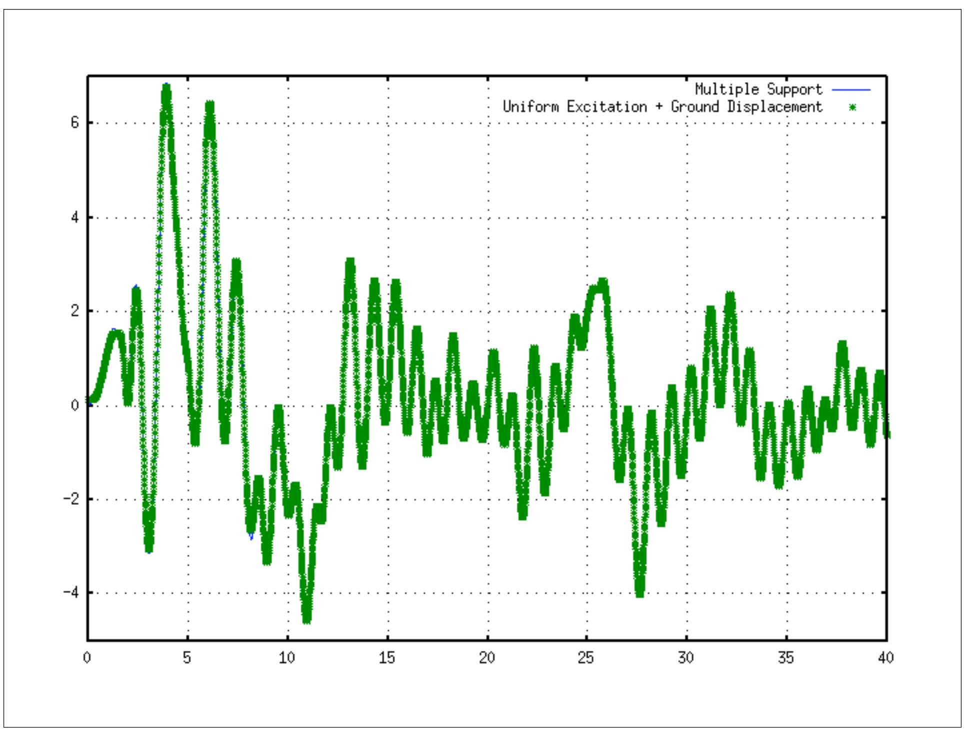
```
source RCFrameGravity.tcl

# Set the gravity loads to be constant
#  & reset the time in the domain
loadConst -time 0.0

# Define nodal mass
set m [expr ($W/2.0)/$g];

#    tag   MX   MY   RZ
mass  3    $m   $m    1.0e-16
mass  4    $m   $m    1.0e-16

# Define dynamic loads
# Set some parameters
set record IELC180
# Source in TCL proc to read PEER SMD record
source ReadRecord.tcl
ReadRecord $record.DT2  $record.dat dT nPts
timeSeries Path 2 -filePath $record.dat -dt $dT -factor $cm
pattern MultiSupport  2   {
    groundMotion 5 Plain -disp 2
    imposedMotion 1 1 5
    imposedMotion 2 1 5
}
recorder Node -time -file multi.out -node 1 3 -dof 1 disp
```

```
rayleigh 0.0 0.0 0.0 0.0
remove sp 1 1
remove sp 2 1

wipeAnalysis
#create the analysis
system BandGeneral
constraints Plain
test NormDispIncr 1.0e-8  10
algorithm Newton
numberer RCM
integrator Newmark  0.5  0.25
analysis Transient

set tFinal [expr $nPts * $dT]
set tCurrent [getTime]
set ok 0
# perofrm the analysis
while {$ok == 0 && $tCurrent < $tFinal} {
   set ok [analyze 1 $dT]
   # if the analysis fails try initial tangent iterati
   if {$ok != 0} {
      puts "regular newton failed .. lets try anoth
      test NormDispIncr 1.0e-8  1000 1
      algorithm ModifiedNewton -initial
      set ok [analyze 1 $dT]
      test NormDispIncr 1.0e-12  10
      algorithm Newton
   }
   set tCurrent [getTime]
}
# Print a message to indicate if analysis succesfu
if {$ok == 0} {
   puts "Transient analysis completed SUCCESS
} else {
   puts "Transient analysis completed FAILED"
}
```





Multiple Support
Uniform Excitation
6
4
2
0
-2
-4
0
5
10
15
20
25
30
35
40



Multiple Support
Uniform Excitation + Ground Displacement
6
4
2
0
-2
-4
0
5
10
15
20
25
30
35
40

Parameter Study - Response Spectra

```
source READSMDFile.tcl
modelBuilder BasicBuilder -ndm 1 -ndf 1

# set a bunch of parameters
set PI 3.14159265
set g 386.4
set TnMin 0.1;  #min period
set TnMax 2.0; #max period
set TnIncr 0.1; #period incr
set M 1.0;       #mass
set A 1.0;       #area
set L 1.0;       #length
set motion ELCENTRO
set outFilename spectrum.dat

# open output file
Set outFileID [open $outFilename w]

#create accel series
ReadSMDFIle $motion.AT2 $motion.acc dt
Set accelSeries “Path -filePath $motion.acc \
      -dt $dt -factor $g”

# loop over period range
Set Tn $TnMin
while {$Tn <= $TnMax} {
   wipe
    set w [expr 2.0 * $PI / $Tn]
    set K [expr $w * $w * $M]
    set E [expr $k * $l / $A
    node 1 0.0
    node 2 $l -mass $M
    fix 1 1
    uniaxialMaterial Elastic 1 $E
    element truss 1 1 2 $A 1
    pattern UniformExcitation 2 1 -accel $accelSeries
    rayleigh 0.0 0.0 0.0 0.0 0.0

    recorder EnvelopeNode -file envelope.out -node 2 -dof 1 disp
    system ProfileSPD
    test NormDispIncr 1.0e-16 10
   algorithm Newton
   integrator Newmark 0.5 0.25
   analysis Transient
    analyze 2000 $dt

   if [catch {open envelope.out r} inFileID]
      puts puts “ERROR - could not open file”

  set min [gets $inFileID]
  set max [gets $inFileID]
  set absMax [gets $inFileID]
  close $inFileID
  puts $outFileID “$Tn $absmax”
  set Tn [expr $Tn + $TnIncr]
}
close $outFileID
```

Terminal — OpenSees — 82x38

```
cee-84-111:~/OpenSees/EXAMPLES/ExampleScripts/ExampleScripts fmk$ ~/bin/OpenSees


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- Version 1.6.0

            (c) Copyright 1999 The Regents of the University of California
                              All Rights Reserved


OpenSees > source ResponseSpectra.tcl
OpenSees > cat spectrum.dat
0.1 0.0706084
0.2 0.419001
0.3 0.753439
0.4 1.47281
0.5 2.68804
0.6 3.0994
0.7 3.37357
0.8 3.70962
0.9 6.24449
1.0 5.9645
1.1 4.9327
1.2 4.75759
1.3 3.94977
1.4 4.41569
1.5 4.72872
1.6 5.93379
1.7 6.3168
1.8 6.72183
1.9 7.40134
2.0 7.47503
```