OpenSees

OpenSees Analysis

Frank McKenna
UC Berkeley

OpenSeesDays 2010

NEES NEEScomm

Why Nonlinear Analysis

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

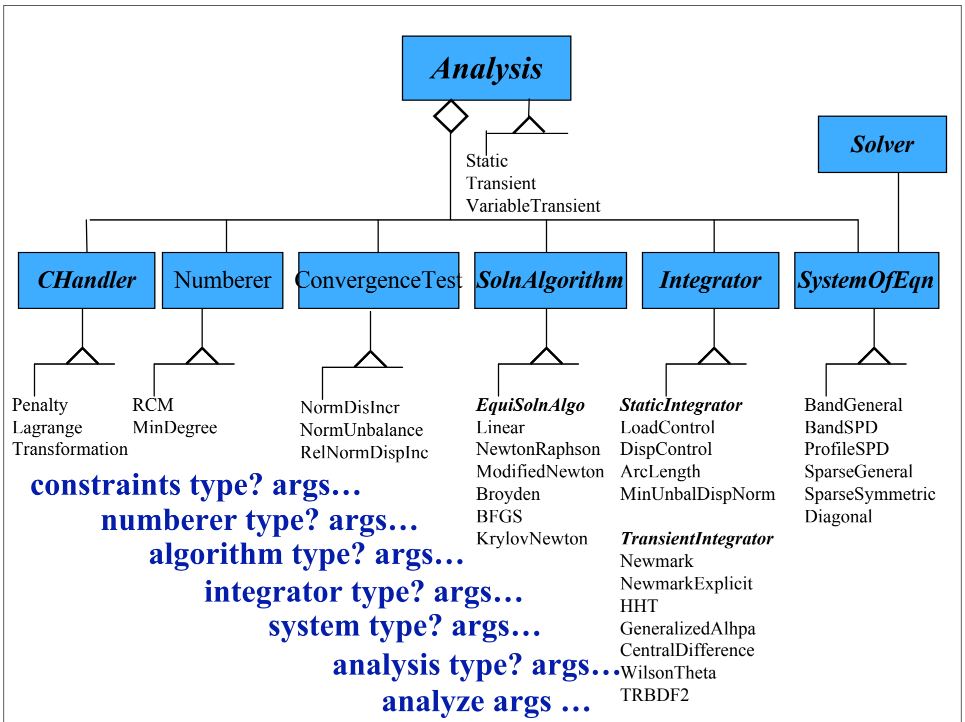
•**Material nonlinearities** - nonlinearities occur when material stress-strain relationship depends on load history (plasticity problems), load duration (creep problems), temperature (thermoplasticity), or combo of all.

•**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

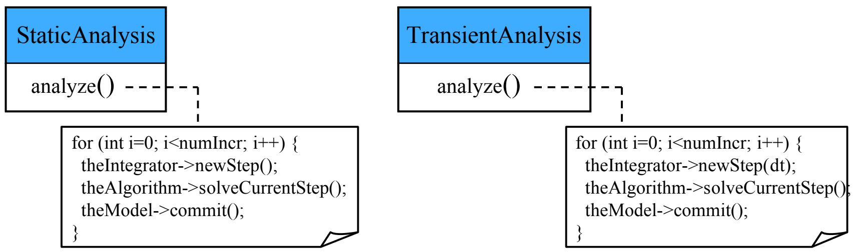


analysis command:

•Static Analysis — *analysis Static*

•Transient Analysis — *analysis Transient*

- both incremental solution strategies



•Eigenvalue

▪general eigenvalue problem

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

eigen numModes? -general

▪standard eigenvalue problem

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

eigen numModes? -standard

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>

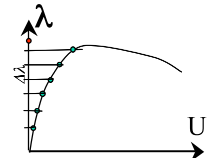
▪Generalized Alpha Method

integrator GeneralizedAlpha α_M α_F *<γ β>*

•Static Integrators for Use in Static Analysis

Nonlinear equation of the form:

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

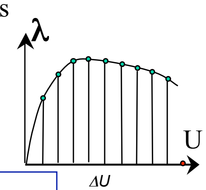


▪Load Control

$\lambda_n = \lambda_{n-1} + \Delta\lambda$

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

$U_{jn} = U_{j\, n-1} + \Delta U_j$

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

▪Minimum Unbalance Displacement Norm

$$\frac{d}{d\Delta\lambda}(\Delta U_n \hat{} \Delta U_n) = 0$$

integrator MinUnbalanceDispNorm Δλ

algorithm command:

- to specify the steps taken to solve the nonlinear equation

•Linear Algorithm

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

algorithm Linear

•Newton-Raphson Algorithm

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

algorithm Newton

•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

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

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

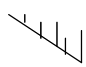
$[\mathbf{K} + \mathbf{C}^\wedge \boldsymbol{\alpha} \mathbf{C}]\,\mathbf{U} = [\mathbf{R} + \mathbf{C}^\wedge \boldsymbol{\alpha} \mathbf{Q}]$

constraints Penalty α_{sp}? α_{mp}?

system command:

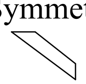
- to specify how matrix equation KU = R is stored and solved

•Profile Symmetric Positive Definite (SPD)



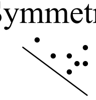
system ProfileSPD

•Banded Symmetric Positive Definite



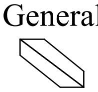
system BandSPD

•Sparse Symmetric Positive Definite



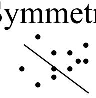
system SparseSPD

•Banded General



system BandGeneral

•Sparse Symmetric



system SparseGeneral

system Umfpack

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

•AMDNumberer

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.

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

$\frac{1}{2}\,(\mathbf{U^\wedge R}) < \mathbf{tol}$

test NormEnergyIncr tol? numIter? <flag?>

•Relative Tests

test RelativeNormUnbalance tol? numIter? <flag?>

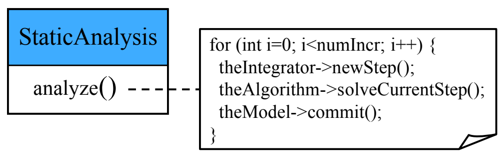
test RelativeNormDispIncr tol? numIter? <flag?>

test RelativeNormEnergyIncr tol? numIter? <flag?>

analyze command:

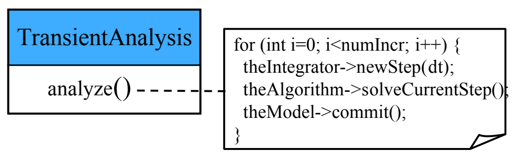
- to perform the static/transient analysis

•Static Analysis



analyze numIter?

•Transient Analysis

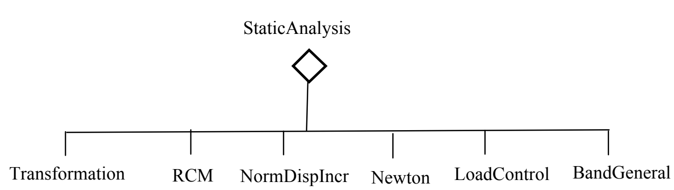


analyze numIter? Δt?

Example Static Analysis:

•Static Nonlinear Analysis with LoadControl

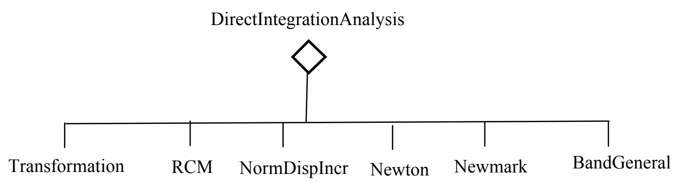
```
constraints Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator LoadControl 0.1
analysis Static
analyze 10
```



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.

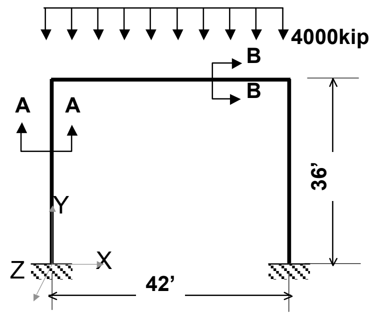
Segmentation Faults, etc:

•Email: fmckenna@ce.berkeley.edu

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

Model

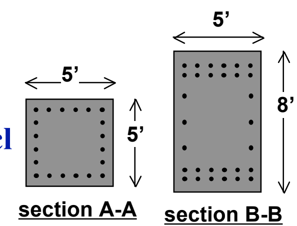
```
# ---------------------------------------------------------------------------------------
# Example4. 2D Portal Frame--  Build Model
# nonlinearBeamColumn element, uniaxial inelastic section
#         Silvia Mazzoni & Frank McKenna, 2006
#
#    ^Y
#    |
#    3__________(3)_________4     __
#    |                      |     |
#    |                      |     |
#    |                      |     |
#   (1)                    (2)   LCol
#    |                      |     |
#    |                      |     |
#    |                      |     |
#   =1=                    =2=   _|_  -------->X
#    |----------LBeam-----------|
#

# SET UP ---------------------------------------------------------------------
```



Ex4.Portal2D.build.InelasticSection.tcl

```
source LibUnits.tcl;          # define basic and system units

# define GEOMETRY -------------------------------------------------------------
set LCol [expr 36*$ft];        # column length
set LBeam [expr 42*$ft];       # beam length
set Weight [expr 4000.*$kip];      # superstructure weight
# define section geometry
```

source Ex4.Portal2D.build.InelasticSection.tcl



Gravity Load Analysis

\# first source in the model
source Ex4.Portal2D.build.InelasticSection.tcl

\# create analysis & perform analysis
constraints transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator LoadControl 0.1
analysis Static
analyze 10

\# look at what happened to node 3
print node 3

* We will call this file example1.tcl for future examples

Terminal — bash — 136×34

```
fmk:~/Desktop/Workshops/OpenSeesDays2008/Examples/NonlinearFrame$ OpenSees example1.tcl


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.0.0

            (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
        commitDisps: 0.000657854 -0.0404395 -0.000957325
         unbalanced Load: 0 0 0
        Mass :
2.58799 0 0
0 0 0
0 0 0

        ID : 4 5 6

fmk:~/Desktop/Workshops/OpenSeesDays2008/Examples/NonlinearFrame$
```

Cyclic Lateral Load Analysis

```
# first source in the model and do gravity load analysis
source example1.tcl

# set gravity loads constant & reset time in domain
loadConst -time 0.0

# create load pattern for lateral loads
pattern Plain 2 Linear {
    load 3 200.0 0.0 0.0
    load 4 200.0 0.0 0.0
}

# do some cyclic analysis
foreach {numSteps stepSize} {10 0.1 10 -0.1 10 -0.1 10 0.1 10 0.1} {
    integrator LoadControl $stepSize
    analyze $numSteps

    set time [getTime]
    set disp [nodeDisp 3 1]
    puts “Time: $time Displacement $disp”
}
```

```
Terminal — bash — 136×44
fmk:~/Desktop/Workshops/OpenSeesDays2008/Examples/NonlinearFrame$ OpenSees example2.tcl


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.0.0

            (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
        commitDisps: 0.000657854 -0.0404395 -0.000957325
         unbalanced Load: 0 0 0
        Mass :
2.58799 0 0
0 0 0
0 0 0

        ID : 4 5 6

Time: 1.000000  Displacement:             2.40694285656669615392
Time: 0.000000  Displacement:             0.11429315708905971039
Time: -1.000000         Displacement:            -2.43518598042165068662
Time: -0.000000         Displacement:            -0.00944903994495972094
Time: 1.000000  Displacement:             2.43795164845030987166

 Node: 3
        Coordinates  : 0 432
        commitDisps: 2.43795 -0.0113628 -0.00158989
         unbalanced Load: 200 0 0
        Mass :
2.58799 0 0
0 0 0
```

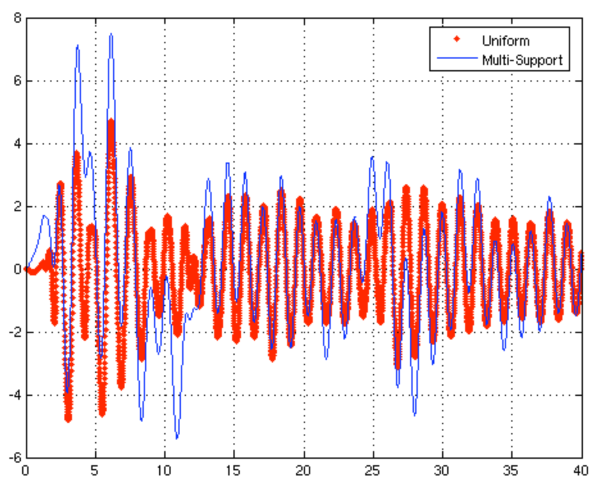
Transient Analysis - Uniform Excitation

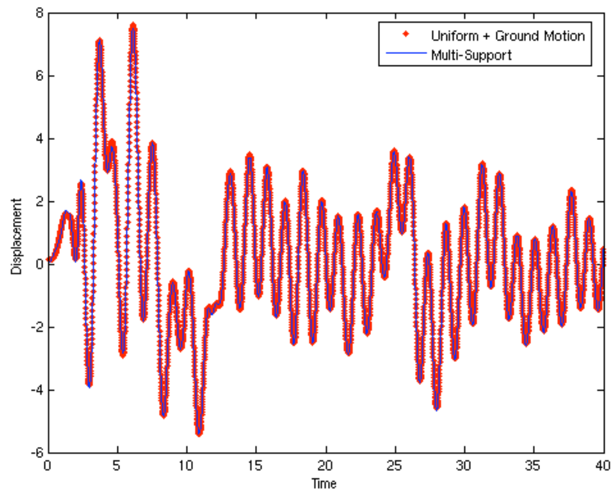
```
# first source in the model and do gravity load analysis
source example1.tcl

# set gravity loads constant & reset time in domain
loadConst -time 0.0

# create load pattern
set motion IELC180
source READSMDFile.tcl
ReadSMDFile $motion.AT2 $motion.acc dt
set accelSeries “Path -filePath $motion.acc -dt $dt -factor 386.4”
pattern UniformExcitation 2 1 -accel $accelSeries

# set damping factors
rayleigh 0.0 0.0 0.0 0.0 0.0

# create the analysis
wipeAnalysis
system ProfileSPD
test NormDispIncr 1.0e-14 10
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient

# create a recorder
recorder Node -time -file example3.out -node 3 -dof 1 disp

analyze 4000 $dt
```

Transient Analysis - MultiSupport Excitation

```
# first source in the model and do gravity load analysis
source example1.tcl

# set gravity loads constant & reset time in domain
loadConst -time 0.0

# remove some sp constraints
remove sp 0
remove sp 3


# create load pattern
set motion IELC180
source READSMDFile.tcl
ReadSMDFile $motion.DT2 $motion.disp dt
set dispSeries "Path -filePath $motion.disp -dt $dt -factor 0.3937
pattern MultiSupport 2 {
    groundMotion 5 Plain -disp $dispSeries
    imposedMotion 1 1 5
    imposedMotion 2 1 5
}

# set damping factors
rayleigh 0.0 0.0 0.0 0.0 0.0


# create a recorder
recorder Node -time -file example4.out -node 3 -dof 1 disp
```

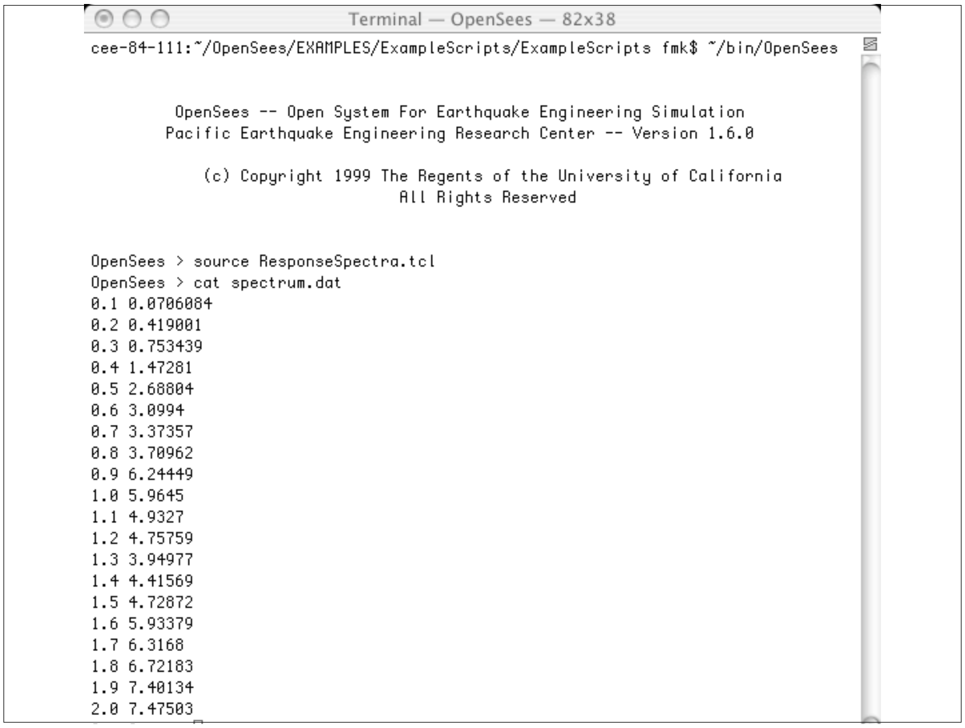
```
# set damping factors
rayleigh 0.0 0.0 0.0 0.0 0.0

# create the analysis
wipeAnalysis
system ProfileSPD
test NormDispIncr 1.0e-14 10
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient

analyze 4000 $dt
```





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
```