Open System for Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center

NEES

OpenSees & Output

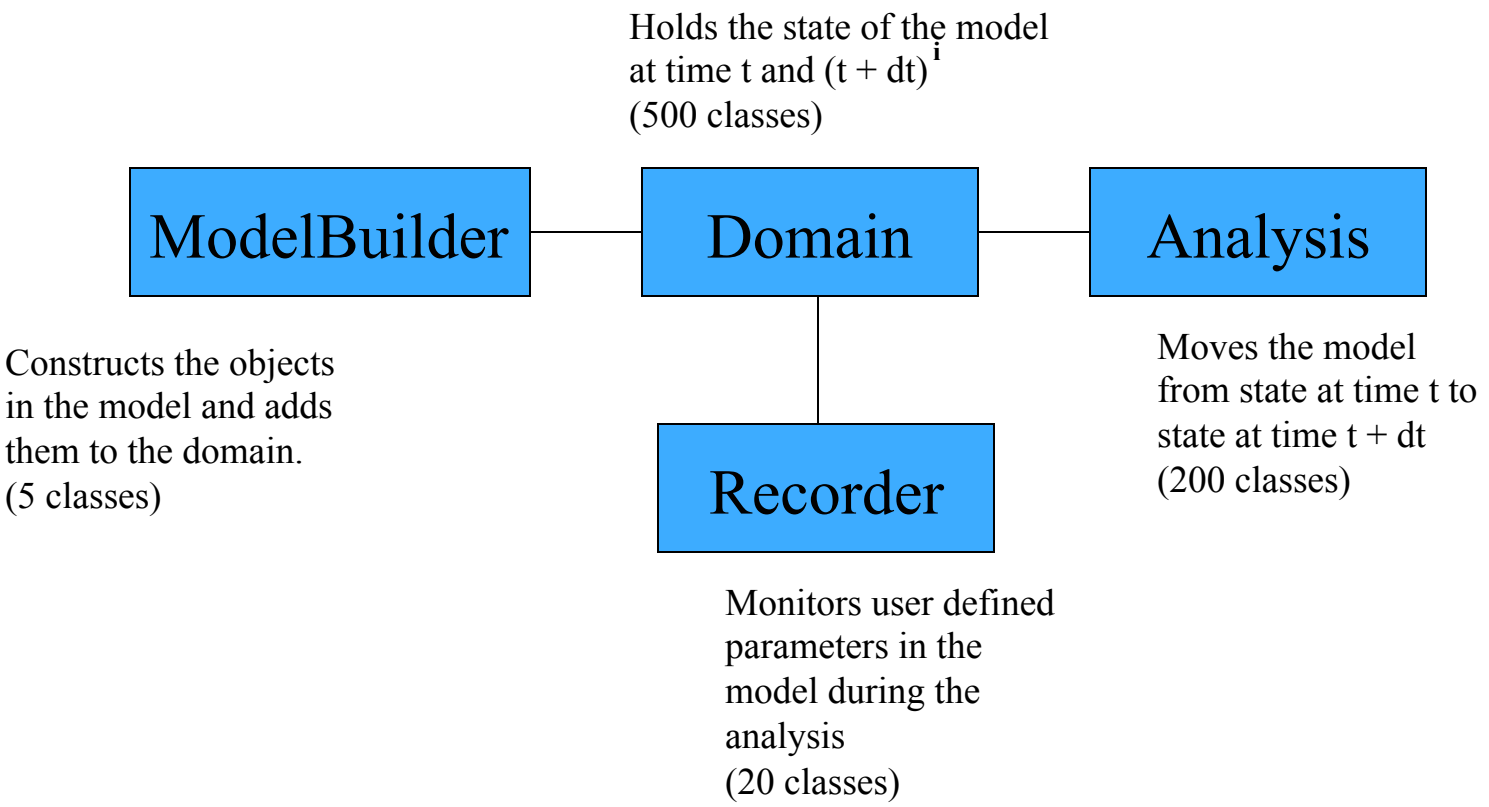
Vesna Terzic
UC Berkeley

November 2011

Agenda

- Introduction to Output options
- OpenSees commands for creating the output: recorder, print, recorder display, commands that return values (e.g., nodeDisp) - demonstrated by examples
- Tcl “puts” commands for creating the output - demonstrated by example
- Summary
- Q & A with web participants

Main Abstractions in OpenSees Framework



Output Options

When you run OpenSees **THERE IS NO OUTPUT PROVIDED UNLESS YOU REQUEST IT**

The commands for creating the output:

1. **recorder** command

   recorder $type $arg1 $arg2 …

2. **puts** command

   puts <$fileID> $string

3. **print** command

   print <-file $fileName> <-node $nd1 $nd2 ..> <-ele $ele1 $ele2 …>

4. OpenSees commands that return values (e.g., nodeDisp)

5. **recorder display** command

Recorder Command

recorder $type $arg1 $arg2 $arg3 ….

http://opensees.berkeley.edu/wiki/index.php/Recorder_Command

Recorder types:

Node:

- Node
- EnvelopeNode
- Drift

Element/section/fiber:

- Element
- EnvelopeElement

Graphics:

- plot

Node/EnvelopeNode Recorders

- To monitor what's happening at the Nodes.

recorder Node <-file $fileName><-timeSeries $tsTag> <-time> <-node $tg1 $tg2 …> -dof $d1 $d2 .. disp
<-xml $fileName> <-nodeRange $tgS $tgE> vel
<-binary $fileName> <-region $rTag> accel
<-tcp $inetAddr> incrDisp
"eigen i"
reaction

Example:

recorder Node -file nodeD.out -node 2 -dof 1 2 3 disp

recorder Node -file nodeA.out -timeSeries 1 -node 2 -dof 1 accel

- The EnvelopeNode takes exactly same args as Node

recorder EnvelopeNode <-file $fileName> <-timeSeries $tsTag> <-time> <-node $tg1 $tg2 …> -dof $d1 $d2 .. disp
<-xml $fileName> <-nodeRange $tgS $tgE> vel
<-binary $fileName> <-region $rTag> accel
<-tcp $inetAddr> incrDisp
reaction

Element/EnvelopeElement Recorders

- To monitor what's happening in the elements/sections/fiber:

```
recorder Element <-file $fileName>    <-time>   <-ele $tg1 $tg2 …>       $arg1 $arg2 …
                 <-xml $fileName>               <-eleRange $tgS $tgE>
                 <-binary $fileName>            <-region $rTag>
                 <-tcp $inetAddr>
```

- The response you can ask vary from element to element. There are arguments that are same for all elements, e.g. forces.

```
recorder Element -file ele.out -ele 1 2 forces
```

- The EnvelopeElement takes exactly same args

```
recorder EnvelopeElement <-file $fileName>    <-time> <-ele $tg1 $tg2 …>   $arg1 $arg2 …
                         <-xml $fileName>             <-eleRange $tgS $tgE>
                         <-binary $fileName>          <-region $rTag>
                         <-tcp $inetAddr>
```

The valid args for different elements

Elastic BCE:

force

Force BCE and BWHE:

force
globalForce
localForce
basicForce
section $secTag $arg1 $arg2
basicDeformation
plasticDeformation
inflectionPoint
tangentDrift
integrationPoints
integrationWeights

Displacement BCE:

force
section $secTag $arg1 $arg2

The valid args for different elements

ZeroLength Element:

force
deformation
stiff
material $matTag $arg1 $arg2 ...

ZeroLengthSection Element:

force
deformation
stiff
section $arg1 $arg2…

Truss element:

axialForce
forces
localForce
deformations
section $arg1 $arg2…
material $arg1 $arg2 ...

The valid args for different sections

Valid args to any section type are: *force* and *deformation*

Fiber Section:

forces
deformations
forceAndDeforamtion
fiber $fiberNum $matArg1 $matArg2 ...
fiber $yLoc $zLoc $matTag $matArg1 $matArg2 ...

The valid args for different materials

Valid args to any material are: *strain*, *stress*, and *tangent*

Fatigue Material:

stressStrain

damage

Examples

recorder Element <-file $fileName> <-time> <-ele $tg1 $tg2 …> $arg1 $arg2 …
<-xml $fileName> <-eleRange $tgS $tgE>
<-binary $fileName> <-region $rTag>
<-tcp $inetAddr>

Force-based beam-column element with fiber sections:

Element forces in global coordinate system:

recorder Element -file ele1force.out -ele 1 force

Sectional deformation (axial strain and curvature):

recorder Element -file ele1sect1def.out -ele 1 section 1 deformations

Stress in a fiber at a specific location:

recorder Element -file ele1sect1fiber00.out -ele 1 section 1 fiber 0. 0. 1 stress

Force BCE:

force
globalForce
localForce
basicForce
basicStiffness
plasticDeformation
inflectionPoint
tangentDrift
integrationPoints
integrationWeights

Fiber Section:

section $secTag $arg1 $arg2 …
fiber $yLoc $zLoc $matTag $matArg1 $matArg2

Fiber Section:

section $secTag $arg1 $arg2 …
fiber $fiberNum $matArg1 $matArg2 ...
fiber $yLoc $zLoc $matTag $matArg1 $matArg2

forces
deformations
forceAndDeformation
fiber $fiberNum $matArg1 $matArg2 ...
fiber $yLoc $zLoc $matTag $matArg1 $matArg2

Eigen analysis of a two-storey one-bay frame

Example Provided by: Vesna Terzic, UC Berkeley

This example demonstrates how to perform eigenvalue analysis and plot mode shapes. A two-storey one-bay frame (Example 10.5 from "Dynamic of Structures" book by Professor Anil K. Chopra) is Figure 1. Node and element numbering is given in Figure 2.

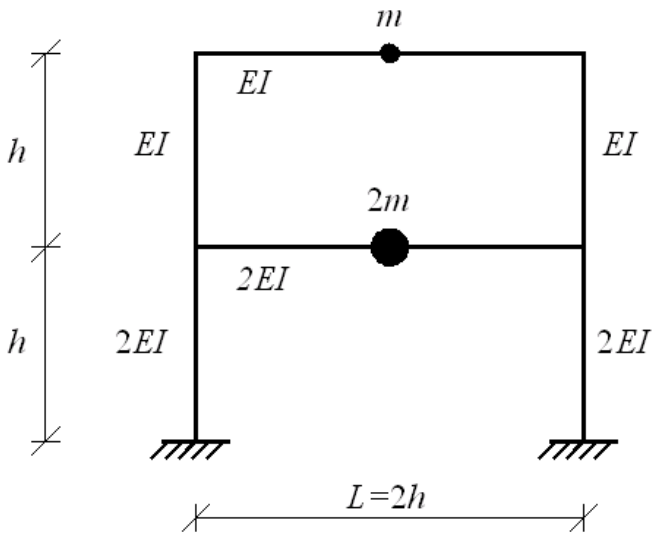


Figure 1

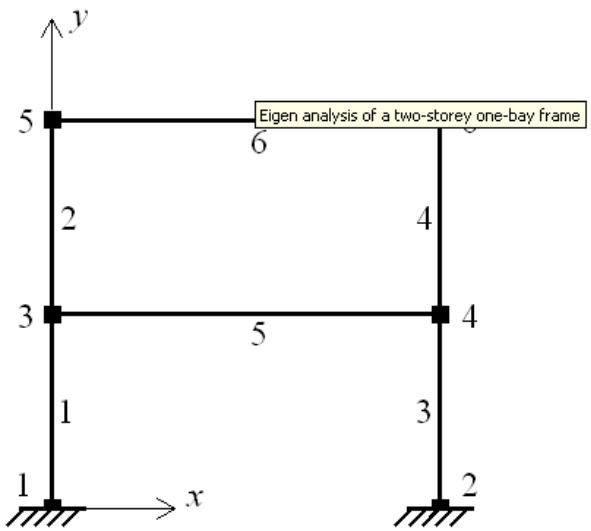


Figure 2

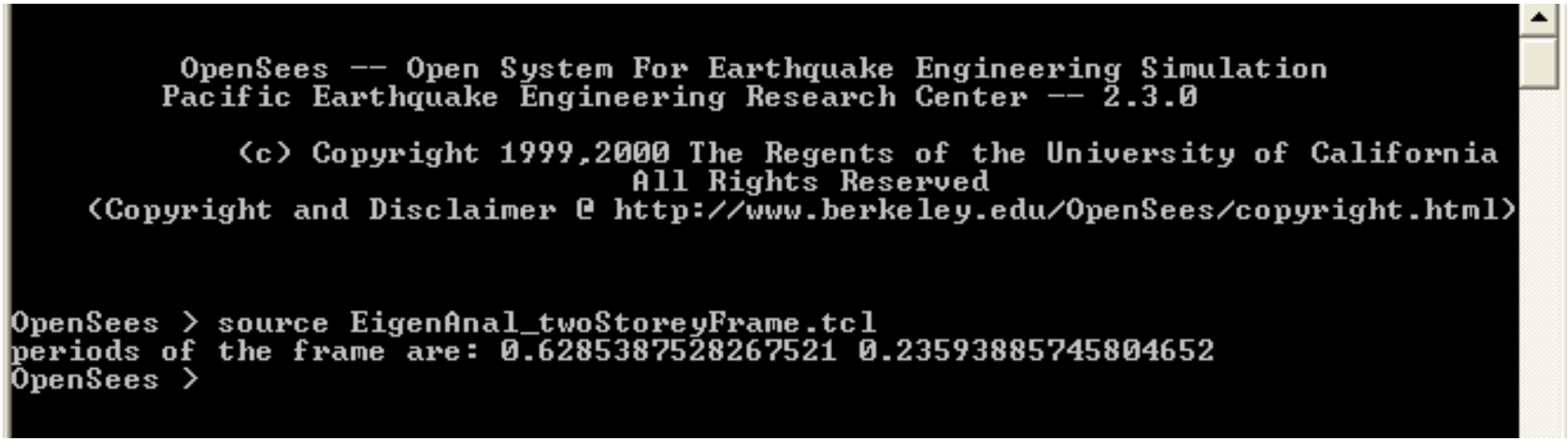
“puts” command used to print data to the screen

In addition to storing periods into a file we can also print it to the terminal using **puts** command:

puts “$string”

Example (EigenAnal_twoStoreyFrame.tcl): printing periods on terminal

```
# print periods to terminal
puts “periods of the frame are: $T”
```



Print command

http://opensees.berkeley.edu/wiki/index.php/Print_Command

To print all objects of the domain:

print <-file $fileName>

To print node information:

print <-file $fileName> -node <$node1 $node2 ...>

To print element information :

print <-file $fileName> -ele <$ele1 $ele2 ...>

Example (EigenAnal_twoStoreyFrame.tcl):

print -node 3

```
OpenSees > print -node 3

 Node: 3
        Coordinates  : 0 120
        Disps: 0 0 0
        Velocities   : 0 0 0
        commitAccels: 0 0 0
        Mass :
0.259067 0 0
0 0 0
0 0 0

         Rayleigh Factor: alphaM: 0
         Eigenvectors:
0.666844 1.21874
0.00224274 -0.00138942
-0.00599676 -0.00256026
```

Commands That Return Values

http://opensees.berkeley.edu/wiki/index.php/Misc_Commands

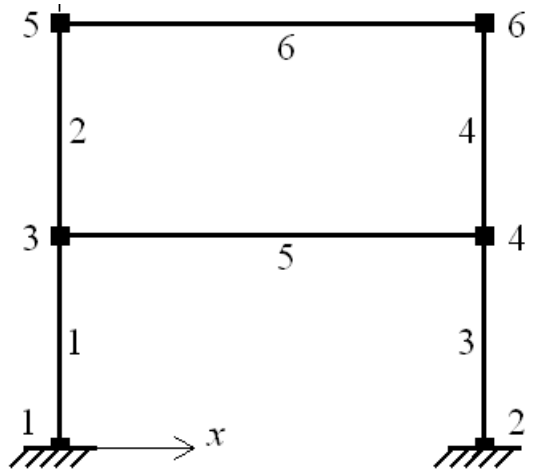
- analyze command — *set ok [analyze numIter <Δt>]*
- getTime command — *set currentTime [getTime]*
- nodeDisp command — *set disp [nodeDisp $node <$dof>]*
- nodeVel command — *set vel [nodeVel $node <$dof>]*
- nodeAccel command — *set acc [nodeAccel $node <$dof>]*
- nodeEigenvector command — *set eig [nodeEigenvector $node $eigenvector <$dof>]*
- eleResponse command — *set resp [eleResponse $eleTag $arg1 $arg2 ...]*

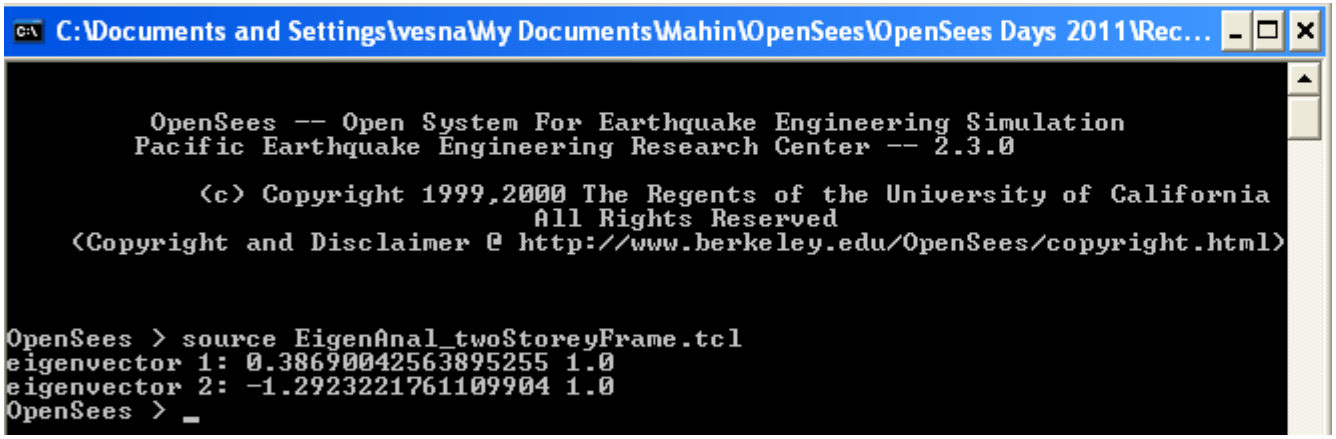
Example: eigenvectors for translational DOFs

- nodeEigenvector command

set eig [nodeEigenvector $node $eigenvector <$dof>]

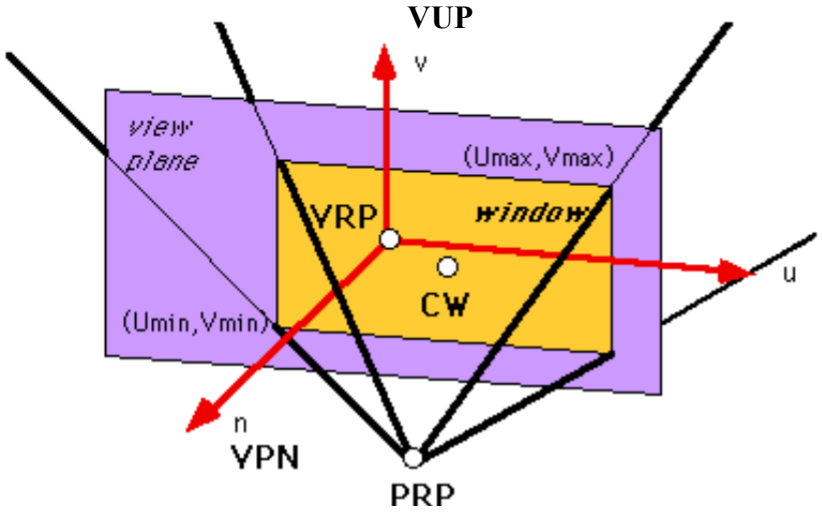
```
# get 2 eigenvectors for nodes 3 and 5
set f11 [nodeEigenvector 3 1 1]
set f21 [nodeEigenvector 5 1 1]
set f12 [nodeEigenvector 3 2 1]
set f22 [nodeEigenvector 5 2 1]

# print them on terminal in a normalized form
puts "eigenvector 1: [list [expr {$f11/$f21}] [expr {$f21/$f21}] ]"
puts "eigenvector 2: [list [expr {$f12/$f22}] [expr {$f22/$f22}] ]"
```





Display command

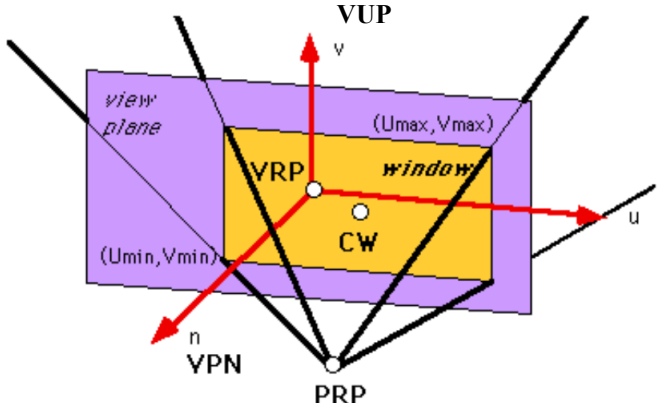
recorder display $windowTitle $xLoc $yLoc $xPixels $yPixels
prp $x $y $z
vup x_v y_v z_v
vpn x_n y_n z_n
viewWindow $x_{prp,n}$ $x_{prp,p}$ $y_{prp,n}$ $y_{prp,p}$
display $arg1 $arg2 $arg3

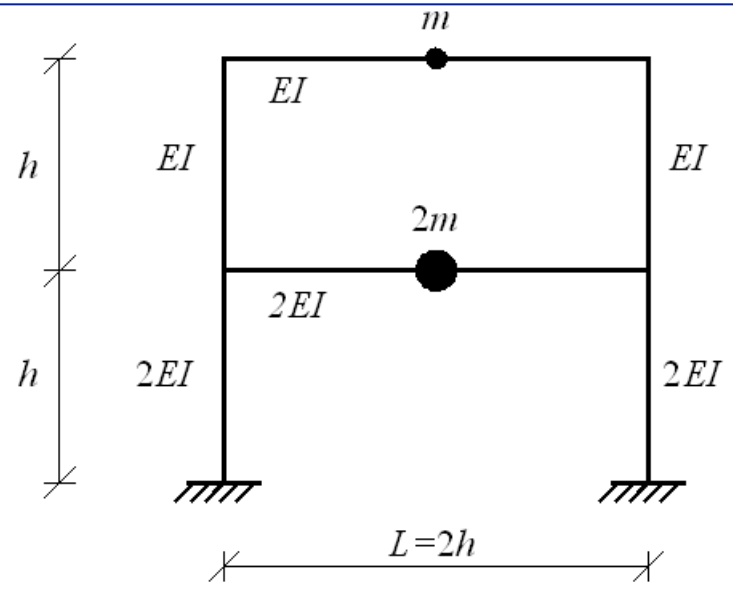


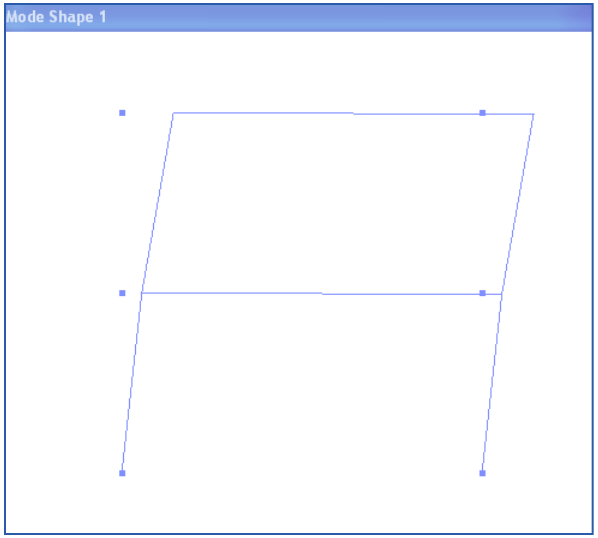
http://www.cs.uic.edu/~jbell/CourseNotes/ComputerGraphics/Projections_Viewpoints.html

Display command: example

```
set h 120
recorder display "Mode Shape 1" 10   10   500   500
prp $h $h 1
vup  0 1 0
vpn  0 0 1
viewWindow -200 200 -200 200
display -1 5 20
```







Procedures for graphical visualization

- For 2D and 3D frame structures the procedures for graphical visualization can be downloaded from the example manual
- For 2D frames:

  http://opensees.berkeley.edu/wiki/index.php/OpenSees_Example_6._generic_2D_Frame,_N-story_N-bay,_Reinforced-Concrete_Section_%26_Steel_W-Section
- For 3D frames:

  http://opensees.berkeley.edu/wiki/index.php/OpenSees_Example_7._3D_Frame,_3-story_3-bayX_3-bayZ,_Reinforced-Concrete_Section_%26_Steel_W-Section
- The proc files are: DisplayPlane.tcl, DisplayModel2D.tcl, DisplayModel3D.tcl

Summary

- Use RECORDER command to monitor response quantities of interest during analysis
- Use PUTS command to:
  - create output files of quantities that RECORDER does not support
  - print data onto the screen
- Use PRINT command to get information about all or some objects of the domain
- Use commands that return values (e.g. nodeDisp) to:
  - check your model
  - change parameters of the model or analysis in the course of analysis
- Use “recorder display” command or OpenSees procedures for creating graphical images of OpenSees models

Questions?