Structural Example – Reinforced-Concrete Frame: Building the Model

Silvia Mazzoni
University of California, Berkeley

OpenSees User Workshop

14 August 2006

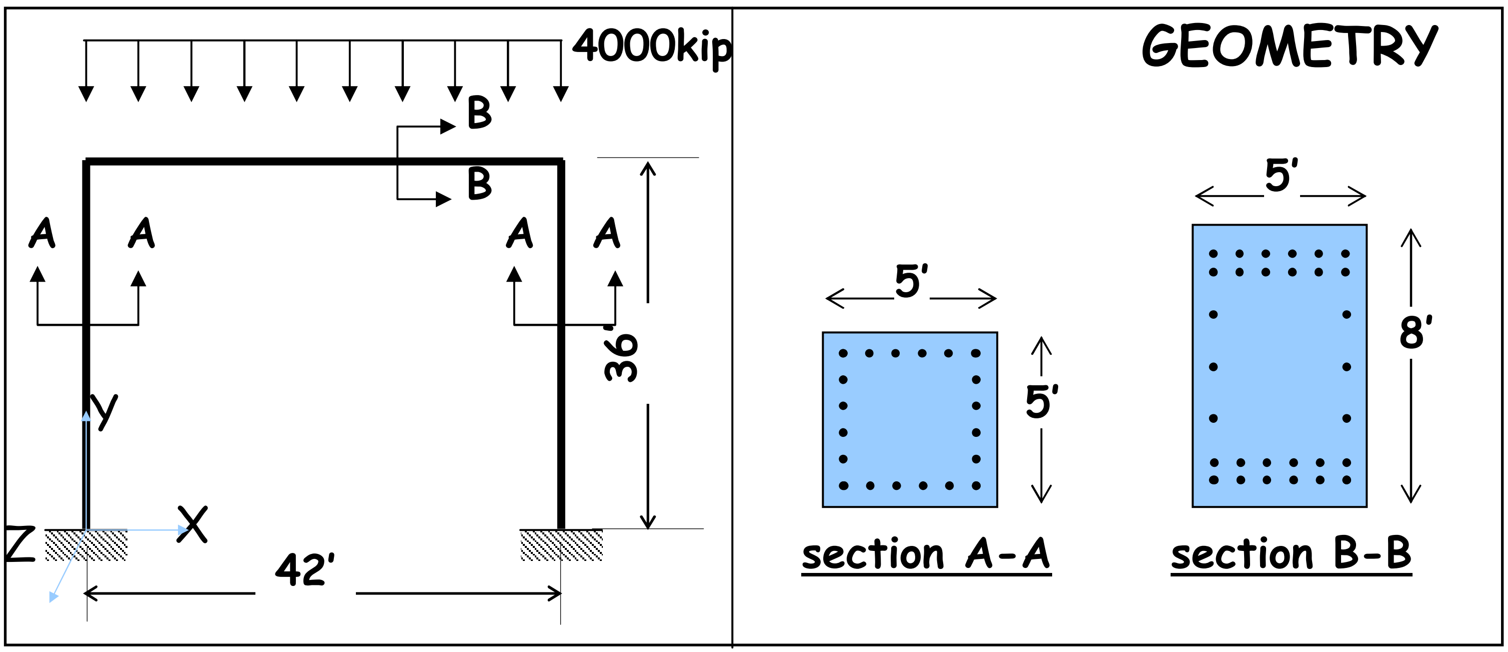
OpenSees

PACIFIC EARTHQUAKE ENGINEERING RESEARCH

NEESit

problem statement

- Reinforced-Concrete Portal Frame
- start with ALL elastic elements (At a more advanced level, these elements can be replaced by more refined element models)
- use kip, inch and sec as basic units



OpenSees NEESit

Model Builder command

- Defining the model builder expands the Tcl command library to include OpenSees-specific commands, such as node and element definition, etc. Currently, there is only one model builder available, basic model builder, this is the model builder that includes all the commands presented in this library.
- The model builder also defines the number of dimensions (ndm) and degrees of freedom per node (ndf).
- For a 2-D problem, you really only need three degrees of freedom at each node, the two translations in the plane and the rotation about the plane's normal:

model basic -ndm 2 -ndf 3

Nodes

- nodal coordinates:

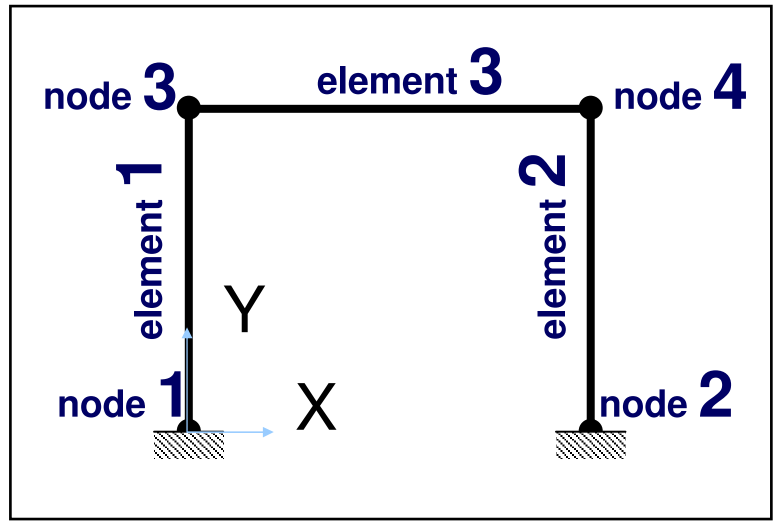
```
node 1 0 0
node 2 504 0
node 3 0 432
node 4 504 432
```

- bondary conditions:

```
fix 1 1 1 1
fix 2 1 1 1
fix 3 0 0 0
fix 4 0 0 0
```

- nodal masses:

```
mass 3 5.18 0. 0.
mass 4 5.18 0. 0.
```



$$\text{mass} = \frac{\frac{4000\,\text{kip}}{2}}{\left(32.2\cdot\frac{\text{ft}}{\text{sec}}\right)\cdot\left(\frac{12\cdot\text{inch}}{1\cdot\text{ft}}\right)} = 5.18$$

Elements -- properties

	columns	beam
area	3600 $\left[(5\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]\cdot\left[(5\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]$	5760 $\left[(5\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]\cdot\left[(8\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]$
moment of inertia Iz	1080000 $\frac{1}{12}\cdot\left[(5\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]\cdot\left[(5\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]^3$	4423680 $\frac{1}{12}\cdot\left[(5\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]\cdot\left[(8\cdot \text{ft})\cdot\left(12\cdot\frac{\text{inch}}{\text{ft}}\right)\right]^3$

Elements – orientation and connectivity

- transformation:
  - local element coordinates ➔ global model coordinates. In a 2D problem, element orientation does not need to be considered, and same for all elements

    geomTransf Linear 1
- connectivity:
  - arguments: $eleTag $iNode $jNode $A $E $Iz $transfTag

  element elasticBeamColumn 1 1 3 3600 4227 1080000 1

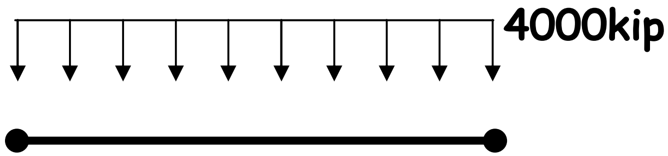
  element elasticBeamColumn 2 2 4 3600 4227 1080000 1

  element elasticBeamColumn 3 3 4 5760 4227 4423680 1

Gravity Loads - member-end forces

- Gravity loads are independent of the type of lateral loading and here they are considered part of the structural model.
- Model gravity load as distributed uniform load along beam

$$\text{DistributedLoad} = \frac{4000 \cdot \text{kip}}{(42 \cdot \text{ft}) \cdot \left(12 \cdot \frac{\text{inch}}{\text{ft}}\right)} = 7.94 \cdot \frac{\text{kip}}{\text{inch}}$$



Define load pattern:

```
pattern Plain 1 Linear {
        eleLoad -ele 3 -type -beamUniform -7.94
}
```

Recorders

- horizontal and vertical displacements at node 3 into a file named Node3.out:

  recorder Node -file Node3.out -time -node 3 -dof 1 2 disp

- local element forces for element 1 into file Element1.out:

  recorder Element -file Element1.out -time -ele 1 force

Summary: example.tcl

```
model basic -ndm 2 -
   ndf 3
# nodal coordinates:
node 1 0 0
node 2 504 0
node 3 0 432
node 4 504 432
# bondary conditions:
fix 1 1 1 1
fix 2 1 1 1
fix 3 0 0 0
fix 4 0 0 0
# nodal masses:
mass 3 5.18 0. 0.
mass 4 5.18 0. 0.
```

```
# transformation:
geomTransf Linear 1
# connectivity:
element elasticBeamColumn 1 1 3 3600 4227
   1080000 1
element elasticBeamColumn 2 2 4 3600 4227
   1080000 1
element elasticBeamColumn 3 3 4 5760 4227
   4423680 1
# Define gravity load pattern:
pattern Plain 1 Linear {
    eleLoad -ele 3 -type -beamUniform -7.94
}
# recorders
recorder Node -file Node3.out -time -node 3 -
   dof 1 2 disp
recorder Element -file Element1.out -time -ele
   1 force
```

execute: line commands

C:\WINDOWS\system32\cmd.exe - opensees172

```
          OpenSees -- Open System For Earthquake Engineering Simulation
         Pacific Earthquake Engineering Research Center -- 1.7.2

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



OpenSees > model basic -ndm 2 -ndf 3
OpenSees > # nodal coordinates:
OpenSees > node 1 0 0
OpenSees > node 2 504 0
OpenSees > node 3 0 432
OpenSees > node 4 504 432
OpenSees > # bondary conditions:
OpenSees > fix 1 1 1 1
OpenSees > fix 2 1 1 1
OpenSees > fix 3 0 0 0
OpenSees > fix 4 0 0 0
OpenSees > # nodal masses:
OpenSees > mass 3 5.18 0. 0.
OpenSees > mass 4 5.18 0. 0.
OpenSees > _
```

execute: source input file

Portal2D.Ex1a.Push.tcl - Notepad

File Edit Format View Help

```
# --------------------------------------------------------------------------------------------
# Example 1. portal frame in 2D
# static pushover analysis of Portal Frame, with gravity.
# all units are in kip, inch, second
# elasticBeamColumn ELEMENT
#                                   Silvia Mazzoni, 2006
#
#    ^Y
#    |
#    3_________(3)________4       __
#    |                    |        |
#    |                    |        |
#    |                    |        |
#   (1)                  (2)       LC
#    |                    |        |
#    |                    |        |
#    |                    |        |
#   =1=                  =2=      _|_
#    |----------LBeam------------|
#

# SET UP ----------------------------------
wipe;
model basic -ndm 2 -ndf 3;

# define GEOMETRY -------------------
# nodal coordinates:
node 1 0 0;
node 2 504 0
node 3 0 432
node 4 504 432
```

C:\WINDOWS\system32\cmd.exe - opensees172

```
C:\Users\AAsilvia\AAProjects\OpenSees\ExamplesManualFinal\Portal2D>opensees172


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 1.7.2

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



OpenSees > source Portal2D.Ex1a.Push.tcl
Done!
OpenSees >
OpenSees >
OpenSees >
OpenSees >
OpenSees >
OpenSees >
OpenSees >
OpenSees >
OpenSees >
OpenSees >
```

Let's redo the example

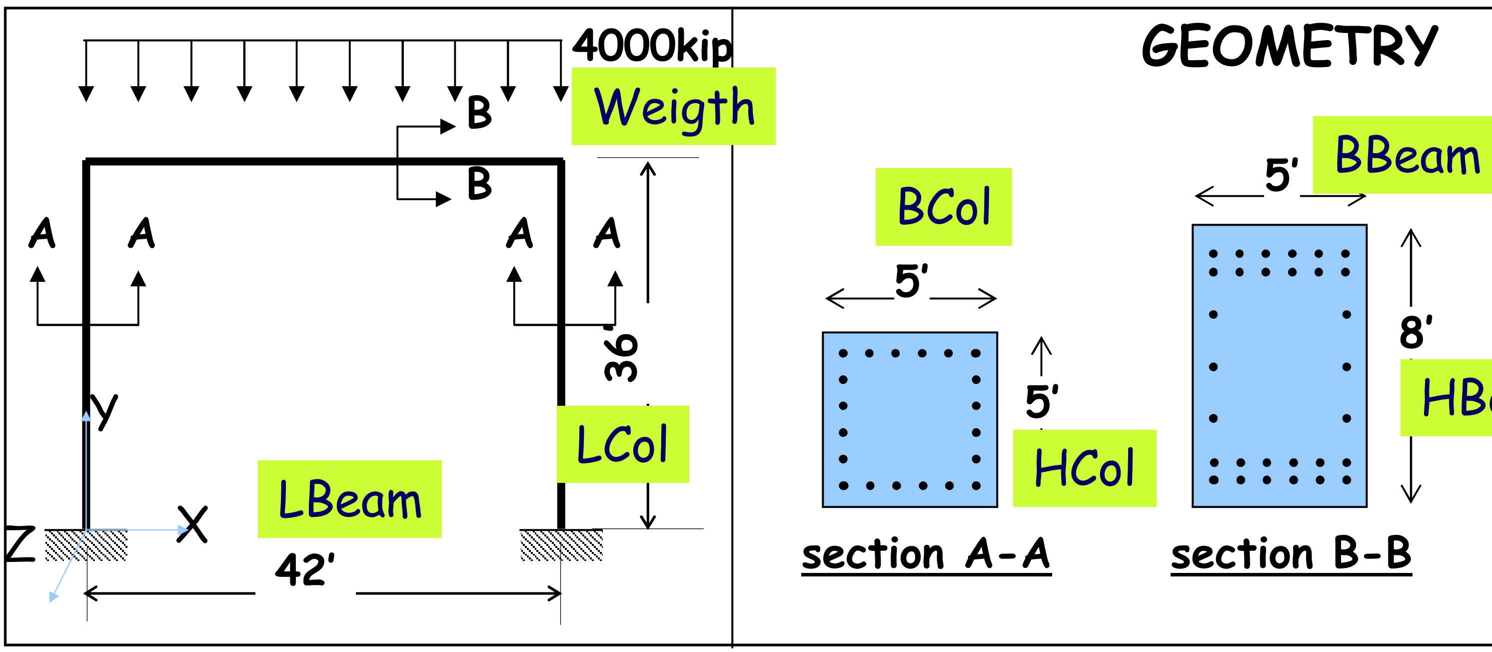
.....my way!!

…..remember what I told you about Tcl?

- Tcl is a string based scripting language
- enables variables and variable substitution (use variables to define units!!!)
- Expression evaluation
- Array management
- Basic control structures (if , while, for, foreach)
- Procedures
- File manipulation

problem statement

- Reinforced-Concrete Portal Frame
- start with ALL elastic elements (At a more advanced level, these elements can be replaced by more refined element models)
- use kip, inch and sec as basic units



Model Builder command -- same

- Defining the model builder expands the Tcl command library to include OpenSees-specific commands, such as node and element definition, etc. Currently, there is only one model builder available, basic model builder, this is the model builder that includes all the commands presented in this library.
- The model builder also defines the number of dimensions (ndm) and degrees of freedom per node (ndf).
- For a 2-D problem, you really only need three degrees of freedom at each node, the two translations in the plane and the rotation about the plane's normal:

model basic -ndm 2 -ndf 3

Now: Units/constants

```
set in      1.;                              # basic units
set sec     1.;                              # basic units
set kip     1.;                              # basic units
set ft      [expr 12.*$in];                  # engineering units
set ksi     [expr $kip/pow($in,2)];
set psi     [expr $ksi/1000.];
set in2     [expr $in*$in];                  # inch^2
set in4     [expr $in*$in*$in*$in];          # inch^4
set PI      [expr 2*asin(1.0)];              # define constants
set g       [expr 32.2*$ft/pow($sec,2)];     # grav. acc.
set Ubig    1.e10;                           # a large number
set Usmall  [expr 1/$Ubig];                  # a small number
set cm      [expr $in/2.54];                 # SI unit
```

Now: Define variables

```
set LCol [expr 36.*$ft];            # column length
set LBeam [expr 42.*$ft];           # beam length
set BCol [expr 5.*$ft];             # column width
set HCol [expr 5.*$ft];             # column depth
set BBeam [expr 5. *$ft];           # beam width
set HBeam [expr 8.*$ft];            # beam depth
set Dmax [expr 15.*$in];            # max displacement
set Weight [expr 4000.*$kip];       # Weight
set Ec [expr 4227*$ksi];            # Young's Modulus

set Wnode [expr $Weight/2];         # node Weight
set Mnode [expr $Wnode/$g];         # node Mass
```

Nodes

- # nodal coordinates:

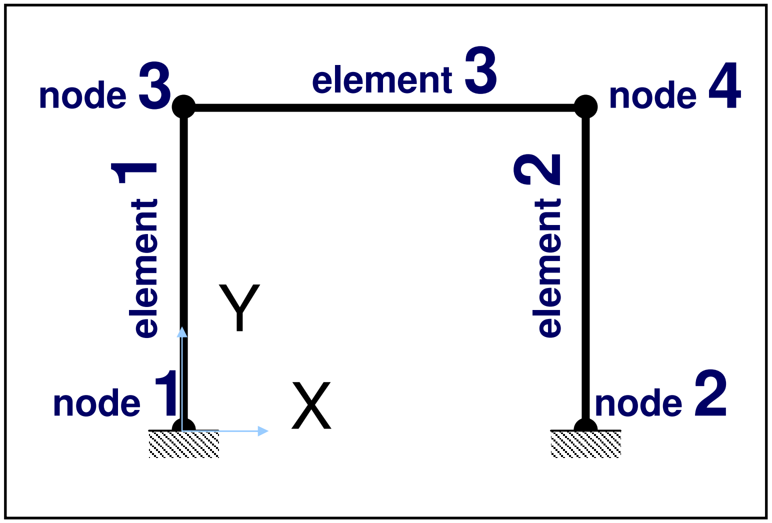
```
node 1 0 0
node 2 $LBeam 0
node 3 0 $LCol
node 4 $LBeam $LCol
```

- # bondary conditions:

```
fix 1 1 1 1
fix 2 1 1 1
fix 3 0 0 0
fix 4 0 0 0
```

- # nodal masses:

```
mass 3 $Mnode 0. 0.
mass 4 $Mnode 0. 0.
```



Elements -- properties

	columns	beam
area	3600 BCol*HCol	5760 BBeam*HBeam
moment of inertia Iz	1080000 1/12*BCol*HCol³	4423680 1/12*BBeam*HBeam³

Elements orientation & connectivity

- transformation:
  - local element coordinates ➔ global model coordinates.

set IDtransf 1

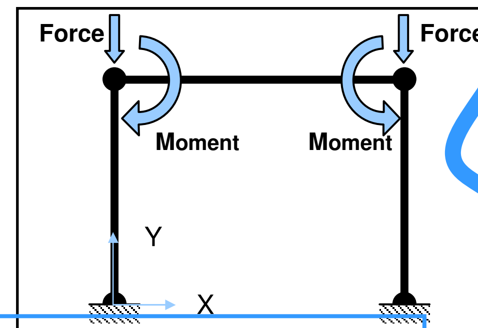
geomTransf Linear $IDtransf

- connectivity:
  - arguments: $eleTag $iNode $jNode $A $E $Iz $transfTag

```
set ACol [expr $BCol*$HCol];
set ABeam [expr $BBeam*$HBeam];
set ICol [expr 1/12*$BCol*pow($HCol,3)];
set IBeam [expr 1/12*$BBeam*pow($HBeam,3)];
element elasticBeamColumn 1 1 3 $ACol $Ec $ICol $IDtransf
element elasticBeamColumn 2 2 4 $ACol $Ec $ICol $IDtransf
element elasticBeamColumn 3 3 4 $ABeam $Ec $IBeam $IDtransf
```

Gravity Loads – member-end forces

- Gravity loads are independent of the type of lateral loading and here they are considered part of the structural model.
- Equivalend member-end forces for distributed loads along an elastic element:



$$DistributedLoad = \frac{4000 \cdot kip}{(42 \cdot ft) \cdot \left(12 \cdot \frac{}{ft}\right)} \quad kip$$

w=Weight/LBeam

Define load pattern:

```
Set w [expr $Weight/$LBeam]
pattern Plain 1 Linear {
                eleLoad -ele 3 -type -beamUniform -$w
        }
```

Recorders

- horizontal and vertical displacements at node 3 into a file named Node3.out:

set Analysis "pushover"

recorder Node -file Node3$Analysis.out -time -node 3 -dof 1 2 disp

- local element forces for element 1 into file Element1.out:

recorder Element -file Element1.out -time -ele 1 force

summary example.tcl

```
set ft      [expr 12.*$in];                    # engineering units
set ksi     [expr $kip/pow($in,2)];
set psi     [expr $ksi/1000.];
set in2     [expr $in*$in];                    # inch^2
set in4     [expr $in*$in*$in*$in];            # inch^4
set PI      [expr 2*asin(1.0)];                # define constants
set g       [expr 32.2*$ft/pow($sec,2)];       # grav. acc.
set Ubig    1.e10;                             # a large number
set Usmall          [expr 1/$Ubig];            # a small number
set cm      [expr $in/2.54];                   # SI unit

set LCol [expr 36*$ft];                        # column length
set LBeam [expr 42*$ft];                       # beam length
set Weight [expr 4000*$kip];                   # Weight
set BCol [expr 5*$ft];                         # column width
set HCol [expr 5*$ft];                         # column depth
set BBeam [expr 5 *$ft];                       # beam width
set HBeam [expr 8*$ft];                        # beam depth

set Wnode [expr $Weight/2];                    # node Weight
set Mnode [expr $Wnode/$g];                    # node Mass

node 1 0 0
node 2 $LBeam 0
node 3 0 $LCol
node 4 $LBeam $LCol
```

summary example.tcl (cont)

```
fix 1 1 1 1; fix 2 1 1 1; fix 3 0 0 0; fix 4 0 0 0
mass 3 $Mnode 0. 0.
mass 4 $Mnode 0. 0.
set IDtransf 1
geomTransf Linear $IDtransf
set ACol [expr $BCol*$HCol];
set ABeam [expr $BBeam*$HBeam];
set ICol [expr 1/12*$BCol*pow($HCol,3)];
set IBeam [expr 1/12*$BBeam*pow($HBeam,3)];
set Ec [expr 4227*$ksi];
element elasticBeamColumn 1 1 3 $ACol $Ec $ICol $IDtransf
element elasticBeamColumn 2 2 4 $ACol $Ec $ICol $IDtransf
element elasticBeamColumn 3 3 4 $ABeam $Ec $IBeam $IDtransf
set w [expr $Weight/$LBeam]
pattern Plain 1 Linear {
    eleLoad -ele 3 -type -beamUniform -$w
}
recorder Node -file Node3.out -time -node 3     -dof 1 2 disp
recorder Element -file Element1.out -time -ele 1 force
```