

**OpenSees User Workshop 2003****Setting up the model -- Example:  
Reinforced-Concrete Frame**

Silvia Mazzoni

*University of California, Berkeley*

21 August 2003

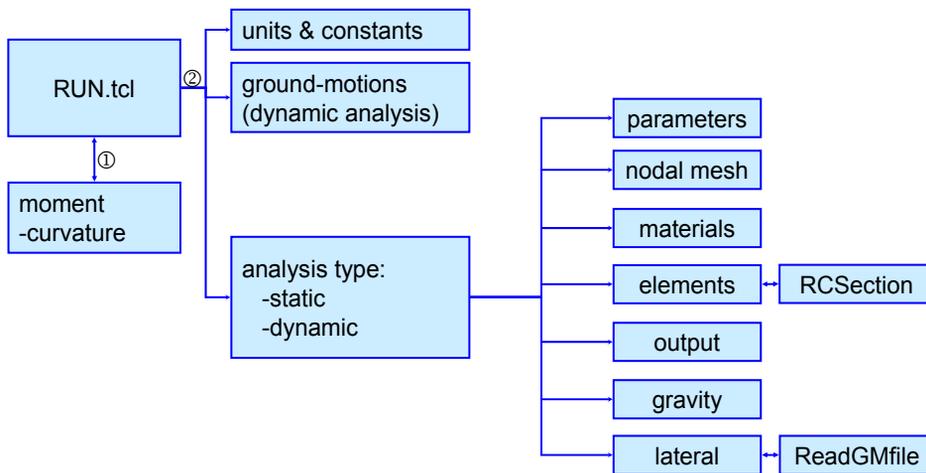
Sponsored by the National Science Foundation  
through the Pacific Earthquake Engineering Research Center**outline**

- advantages of tcl scripting language
- input-file architecture
- Example: Reinforced-Concrete Frame

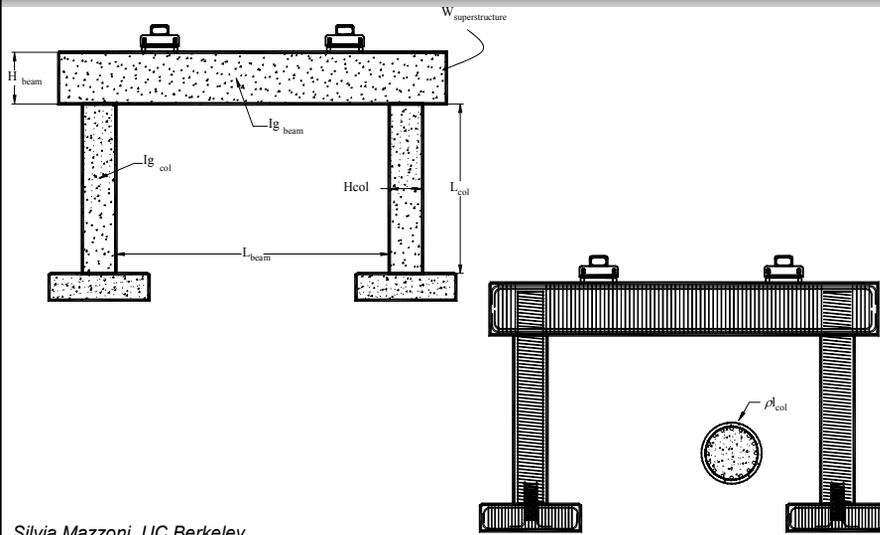
## Tcl Scripting language -- advantages

- ability to “source-in” files
  - input-file architecture
- variables
  - unit and constant definition
  - parameter definition

## input-file architecture



## Example: Reinforced-Concrete Frame



Silvia Mazzoni, UC Berkeley  
OpenSees User Workshop 2003

5

## RUN.tcl

```
wipe # clear memory  
source units.tcl; # units and constants
```

set up

```
set ANALYSIS "Static"; # static pushover analysis  
source Analysis.tcl
```

static analysis

```
set ANALYSIS "Dynamic"; # dynamic ground-motion analysis  
source Analysis.tcl
```

dynamic analysis

Silvia Mazzoni, UC Berkeley  
OpenSees User Workshop 2003

6

## units.tcl

1.	set in 1.;	
2.	set sec 1.;	← basic units
3.	set kip 1.;	
4.	set ksi [expr \$kip/pow(\$in,2)];	
5.	set psi [expr \$ksi/1000.];	← imperial units
6.	set ft [expr 12.*\$in];	
7.	set cm [expr 2.54*\$in];	
8.	set MPa [expr 145*\$psi];	← metric units
9.	set g [expr 981*\$cm/pow(\$sec,2)];	gravitational constant
10.	set PI [expr 2*asin(1.0)];	$\pi$
11.	set U 1.e10;	a really large number
12.	set u [expr 1./\$U];	a really small number

Silvia Mazzoni, UC Berkeley  
OpenSees User Workshop 2003

7

## analysis.tcl

1.	model basic -ndm 3 -ndf 6	← create model builder
2.	source units.tcl;	
3.	source parameters.tcl;	← set up parameters and variables
4.	source nodalmesh.tcl;	
5.	source materials.tcl;	← set up structural model
6.	source elements.tcl;	
7.	source output.tcl;	← specify data output
8.	source gravity.tcl;	
9.	source lateral.tcl;	← apply loading
10.	wipeanalysis	← clear memory

Silvia Mazzoni, UC Berkeley  
OpenSees User Workshop 2003

8

# parameters.tcl

## basic parameters

1.	<code>set Hcol [expr 6*\$ft];</code>	<code># column diameter</code>	column
2.	<code>set Lcol [expr 36*\$ft];</code>	<code># column length</code>	
3.	<code>set GrhoCol 0.0175;</code>	<code># column longitudinal-steel ratio</code>	
4.	<code>set Hbeam [expr 8*\$ft];</code>	<code># beam depth</code>	beam
5.	<code>set Lbeam [expr 42.*\$ft];</code>	<code># beam length</code>	
6.	<code>set Gblc 2;</code>	<code># ratio of beam-to-column moment of inertia</code>	
7.	<code>set Weight [expr 3000.*\$kip];</code>	<code># superstructure weight</code>	superstructure

# parameters.tcl

## GEOMETRY

1.	<code>set Rcol [expr \$Hcol/2];</code>	<code># COLUMN radius</code>	column
2.	<code>set Acol [expr \$PI*pow(\$Rcol,2)];</code>	<code># column cross-sectional area</code>	
3.	<code>set cover [expr \$Hcol/15];</code>	<code># column cover width</code>	
4.	<code>set IgCol [expr \$PI*pow(\$Rcol,4)/4];</code>	<code># column gross moment of inertia, uncracked</code>	
5.	<code>set IyCol \$IgCol;</code>	<code># elastic-column properties</code>	
6.	<code>set IzCol \$IgCol;</code>	<code># elastic-column properties</code>	
7.	<code>set IzBeam [expr \$Gblc*\$IgCol];</code>	<code># BEAM gross moment of inertia -- I</code>	beam
8.	<code>set Hbeam [expr 8*\$ft];</code>	<code># beam depth, not really used</code>	
9.	<code>set Bbeam [expr \$IzBeam*12/pow(\$Hbeam,3)];</code>	<code># beam width not used</code>	
10.	<code>set IyBeam [expr \$Hbeam*pow(\$Bbeam,3)/12];</code>	<code># beam gross moment of inertia--vert Y</code>	
11.	<code>set Abeam [expr \$Hbeam*\$Bbeam*10000];</code>	<code># beam cross-sectional area</code>	
12.	<code>set GLbLc [expr \$Lbeam/\$Lcol];</code>	<code># beam-to-column length ratio</code>	

# parameters.tcl

## MATERIAL PROPERTIES

1.	set	fc	[expr -5.5*\$ksi];	# CONCRETE Compressive Stren	concrete
2.	set	Ec	[expr 57*\$ksi*sqrt(-\$fc/\$psi)];	# Concrete Elastic Modulus	
3.	set	fc1C	[expr 1.26394*\$fc];	# CONFINED concrete max. stress (Mander)	
4.	set	eps1C	[expr 2.*\$fc1C/\$Ec];	# strain at maximum stress	
5.	set	fc2C	\$fc;	# ultimate stress	
6.	set	eps2C	[expr 2.*\$fc2C/\$Ec];	# strain at ultimate stress	
7.	set	fc1U	\$fc;	# UNCONFINED concrete maximum stress	
8.	set	eps1U	-0.003;	# strain at maximum stress	
9.	set	fc2U	[expr 0.1*\$fc];	# ultimate stress	
10.	set	eps2U	-0.006;	# strain at ultimate stress	
11.	set	Fy	[expr 70.*\$ksi];	# STEEL yield stress	steel
12.	set	epsY	0.002;	# steel yield strain	
13.	set	Fy1	[expr 75.*\$ksi];	# pre-strain-hardening stress	
14.	set	epsY1	[expr 3.*\$epsY];	# pre-strain-hardening strain	
15.	set	Fu	[expr 110.*\$ksi];	# ultimate stress of steel	
16.	set	epsU	0.1;	# ultimate strain of steel	
17.	set	Es	[expr \$Fy/\$epsY];	# Young's Modulus of steel	
18.	set	Bs	[expr (\$Fu-\$Fy)/(\$epsU-\$epsY)/\$Es];	# post-yield stiffness ratio of steel	

Silvia

OpenSees User Workshop 2003

11

# parameters.tcl

## HYSTERETIC MODEL

1.	set	pinchX	1.0;	# pinching parameter for hysteretic model	
2.	set	pinchY	1.0;	# pinching parameter for hysteretic model	
3.	set	damage1	0.0;	# damage parameter for hysteretic model	
4.	set	damage2	0.0;	# damage parameter for hysteretic model	
5.	set	betaMUsteel	0.0;	# degraded unloading stiffness for hysteretic material based on $MU^{(-beta)}$	
6.	set	betaMUjoint	0.0;	# degraded unloading stiffness for hysteretic material based on $MU^{(-beta)}$ -- timoshenko value of 0.5	
7.	set	betaMUph	0.0;	# degraded unloading stiffness for hysteretic material based on $MU^{(-beta)}$ -- timoshenko value of 0.5	
8.	set	G	\$U;	# Torsional stiffness Modulus	
9.	set	J	1.;	# Torsional stiffness of section	
10.	set	GJ	[expr \$G*\$J];	# Torsional stiffness	

Silvia Mazzoni, UC Berkeley  
OpenSees User Workshop 2003

12

# parameters.tcl

## Column&beam-model properties

```
1. # define COLUMN REINFORCEMENT parameters
2. set NbCol 20; # number of column longitudinal-reinf. bars
3. set AsCol [expr $GrhoCol*$Acol]; # total steel area in column section
4. set AbCol [expr $AsCol/$NbCol]; # bar area of column longitudinal reinforcement

5. # set up parameters for column section and element definition
6. set np 5; # Number of integration points
7. set riCol 0.0; # inner radius of column section
8. set roCol $Rcol; # outer radius of column section

9. set IDcore 1; # ID tag for core concrete
10. set IDcover 2; # ID tag for cover concrete
11. set IDsteel 3; # ID tag for steel

12. set nfCoreR 8; # number of radial fibers in core
13. set nfCoreT 16; # number of tangential fibers in core
14. set nfCoverR 2; # number of radial fibers in cover
15. set nfCoverT 16; # number of tangential fibers in cover

16. set IDcolFlex 2; # ID tag for column section in flexure, before aggreg. torsion
17. set IDcolTors 10; # ID tag for column section in torsion
18. set IDcolSec 1; # ID tag for column section

19. set IDcolTrans 1; # ID tag for column transformation, defining element normal
20. set IDbeamTrans 2; # ID tag for beam transformation, defining element normal
```

Silvia  
OpenSees

3

# parameters.tcl

## GRAVITY

```
1. set Pdl [expr $Weight/2]; # gravity axial load per column
2. set GPcol [expr -$Pdl/$Acol/$fc]; # gravity axial load per column
   as a fraction of compressive strength
3. set Wbeam [expr $Weight/$Lbeam]; # gravity dead load distributed
   along beam length
4. set Mdl [expr $Wbeam*pow($Lbeam,2)/12]; # nodal moment due
   to distributed dl
5. set Mass [expr $Weight/$g]; # mass of superstructure
6. set Mnode [expr $Mass/2]; # nodal mass for each column
   joint
```

# parameters.tcl

## DAMPING

1. # define DAMPING parameters from \$xDamp --SDOF system, use stiffness proportional damping only
2. set xDamp 0.02; # modal damping ratio
3. set lambda [eigen 1]; # eigenvalue analysis
4. set omega [expr sqrt(\$lambda)]; # natural frequency (rad/sec)
5. set Tperiod [expr 2\*\$PI/\$omega]; # period (sec.)
6. set alphaM 0; # mass-prop. RAYLEIGH damping parameter; D = alphaM\*M
7. set betaK 0; # stiffness proportional damping; +beatK\*KCurrent
8. set betaKcomm [expr 2\*\$xDamp/\$omega]; # +betaKcomm\*KlastCommitt
9. set betaKinit 0; # +beatKinit\*Kini

*NOTE: you actually have to move these commands to once the model has been set up, as you cannot perform an eigenvalue analysis at this point, yet*

# parameters.tcl

## ANALYSIS

1. set DxPush [expr 0.1\*\$in]; # Displacement increment for pushover a **static**
2. set DmaxPush [expr 20\*\$in]; # maximum displacement for pushover analysis
3. set DtAnalysis [expr 0.005\*\$sec]; # time-step Dt for lateral analysis **dynamic**
4. set DtGround [expr 0.02\*\$sec]; # time-step Dt for input ground motion
5. set TmaxGround [expr 50\*\$sec]; # maximum duration of ground-motion analysis
6. set gamma 0.5; # gamma value for newmark integration
7. set beta 0.25; # beta value for newmark integration

# nodalmesh.tcl

```
1. set IDctrlNode 3;
2. # Define nodes; # frame is in X-Y plane (X-horizonta
3. node 1 0.0 0.0 0.0
4. node 2 $Lbeam 0.0 0.0
5. node 3 0.0 $Lcol 0.0 -mass $Mnode 0.0 0.0 0.0 0.0
6. node 4 $Lbeam $Lcol 0.0 -mass $Mnode 0.0 0.0 0.0 0.0
7. # Boundary conditions; # node DX DY DZ RX RY RZ ! 1: fixed, 0: released
8. fix 1 1 1 1 1 1 1;
9. fix 2 1 1 1 1 1 1
10. fix 3 0 0 1 1 1 0
11. fix 4 0 0 1 1 1 0
12. #
13. #
14. #
15. #
```

coordinates

boundary conditions

Silvia Mazzoni, UC Berkeley  
OpenSees User Workshop 2003

17

# materials.tcl

concrete

```
1. set ConcreteMaterialType "inelastic" # options: "elastic","inelastic"
2. if {$ConcreteMaterialType == "elastic"} {
3. uniaxialMaterial Elastic $IDcore $Ec
4. uniaxialMaterial Elastic $IDcover $Ec
5. }
6. if {$ConcreteMaterialType == "inelastic"} {
7. # uniaxial Kent-Scott-Park concrete model w/ linear unload/reload, no T strength (-ve comp.)
8. uniaxialMaterial Concrete01 $IDcore $fc1C $seps1C $fc2C $seps2C; # Core
9. uniaxialMaterial Concrete01 $IDcover $fc1U $seps1U $fc2U $seps2U;
# Cover
10. }
```

Silvia Mazzoni, UC Berkeley  
OpenSees User Workshop 2003

18

# materials.tcl

## reinforcing steel

```
1. set SteelMaterialType "hysteretic"; # options: "elastic", "bilinear", "hysteretic"  
2. if {$SteelMaterialType == "elastic"} {  
3.   uniaxialMaterial Elastic $IDsteel $Es  
4. }  
5. if {$SteelMaterialType == "bilinear"} {  
6.   # Reinforcing steel, uniaxial bilinear steel model with isotropic hardening.  
7.   uniaxialMaterial Steel01 $IDsteel $Fy $Es $Bs  
8. }  
9. if {$SteelMaterialType == "hysteretic"} {  
10.  # Reinforcing steel, uniaxial hysteretic steel model with isotropic hardening.  
11.  uniaxialMaterial Hysteretic $IDsteel $Fy $sepsY $Fy1 $sepsY1 $Fu $sepsU -$Fy -  
    $sepsY -$Fy1 -$sepsY1 -$Fu -$sepsU $pinchX $pinchY $damage1 $damage2  
    $betaMUSTeel  
12. }
```

# tcl procedure: define fiber section

```
proc RCcircSection {id Ri Ro cover coreID coverID steelID Nbars Ab nfCoreR nfCoreT nfCoverR nfCoverT} {  
  section fiberSec $id {  
    CONCRETE  
    set Rc [expr $Ro-$cover]; # Core radius  
    patch circ $coreID $nfCoreT $nfCoreR 0 0 $Ri $Rc 0 360; # Define the core patch  
    patch circ $coverID $nfCoverT $nfCoverR 0 0 $Rc $Ro 0 360; # Define the cover patch  
  
    STEEL  
    if {$Nbars<= 0} { return }  
    set theta [expr 360.0/$Nbars]; # angle increment between bars  
    layer circ $steelID $Nbars $Ab 0 0 $Rc $theta 360; # Define the reinforcing layer  
  }  
}
```

The diagram illustrates the geometry of a circular fiber section. The left part shows the concrete section with an outer radius  $R_o$  and an inner core radius  $R_i$ . The concrete is divided into a central core patch and an outer cover patch. The right part shows the steel reinforcement, consisting of  $N_{bars}$  bars with a total area  $A_b$ , arranged in a circular pattern with an angle increment of  $360^\circ/N_{bars}$  between adjacent bars.

# elements.tcl

```

1. set ColumnType "inelastic";
2. source RCcircSection.tcl; # proc to define circular fiber section- flexure tcl procedure
3. RCcircSection $IDcolFlex $riCol $roCol $cover $IDcore $IDcover $IDsteel $NbCol $AbCol $nfCoreR $nfCoreT $nfCoverR $nfCoverT
4. uniaxialMaterial Elastic $IDcolTors $GJ; # Define torsion
5. section Aggregator $IDcolSec $IDcolTors T -section $IDcolFlex; # attach torsion & flex
6. geomTransf Linear $IDcolTrans 0 0 1; # no 2nd-order effects, define element normal
7.
8. if {$ColumnType == "elastic"}{
9.     element elasticBeamColumn 1 1 3 $Acol $Ec $G $J $lyCol $lzCol $IDcolTrans
10.    element elasticBeamColumn 2 2 4 $Acol $Ec $G $J $lyCol $lzCol $IDcolTrans }
11. if {$ColumnType == "inelastic"}{
12.    # element element type ID, node I, node J, no. int pts, section ID, transf. ID
13.    element nonlinearBeamColumn 1 1 3 $np $IDcolSec $IDcolTrans
14.    element nonlinearBeamColumn 2 2 4 $np $IDcolSec $IDcolTrans }
15. geomTransf Linear $IDbeamTrans 0 0 1; # BEAM transformation, define element normal
16. element elasticBeamColumn 3 3 4 $Abeam $Ec $G $J $lyBeam $lzBeam $IDbeamTrans

```

# output.tcl

```

1. if {$ANALYSIS == "Static"}{
2.     recorder Node DStat.out disp -time -node $IDctrlNode -dof 1
3.     for {set iel 1} {$iel <= 2} {incr iel 1}{
4.         recorder Element $iel -file EI[expr $iel]FStat.out force
5.     }
6. }
7. if {$ANALYSIS == "Dynamic"}{
8.     recorder Node Ddyna.out disp -time -node $IDctrlNode -dof 1
9.     for {set iel 1} {$iel <= 2} {incr iel 1}{
10.        recorder Element $iel -time -file EI[expr $iel]Fdyna.out force
11.    }
12. }
13. }

```

## conclusions

advantages of Tcl scripting language simplifies the following:

- simplify error check
- generate a new input file while using components of a previously-generated and tested input file