Parameter Studies Using OpenSees

Silvia Mazzoni
University of California, Berkeley

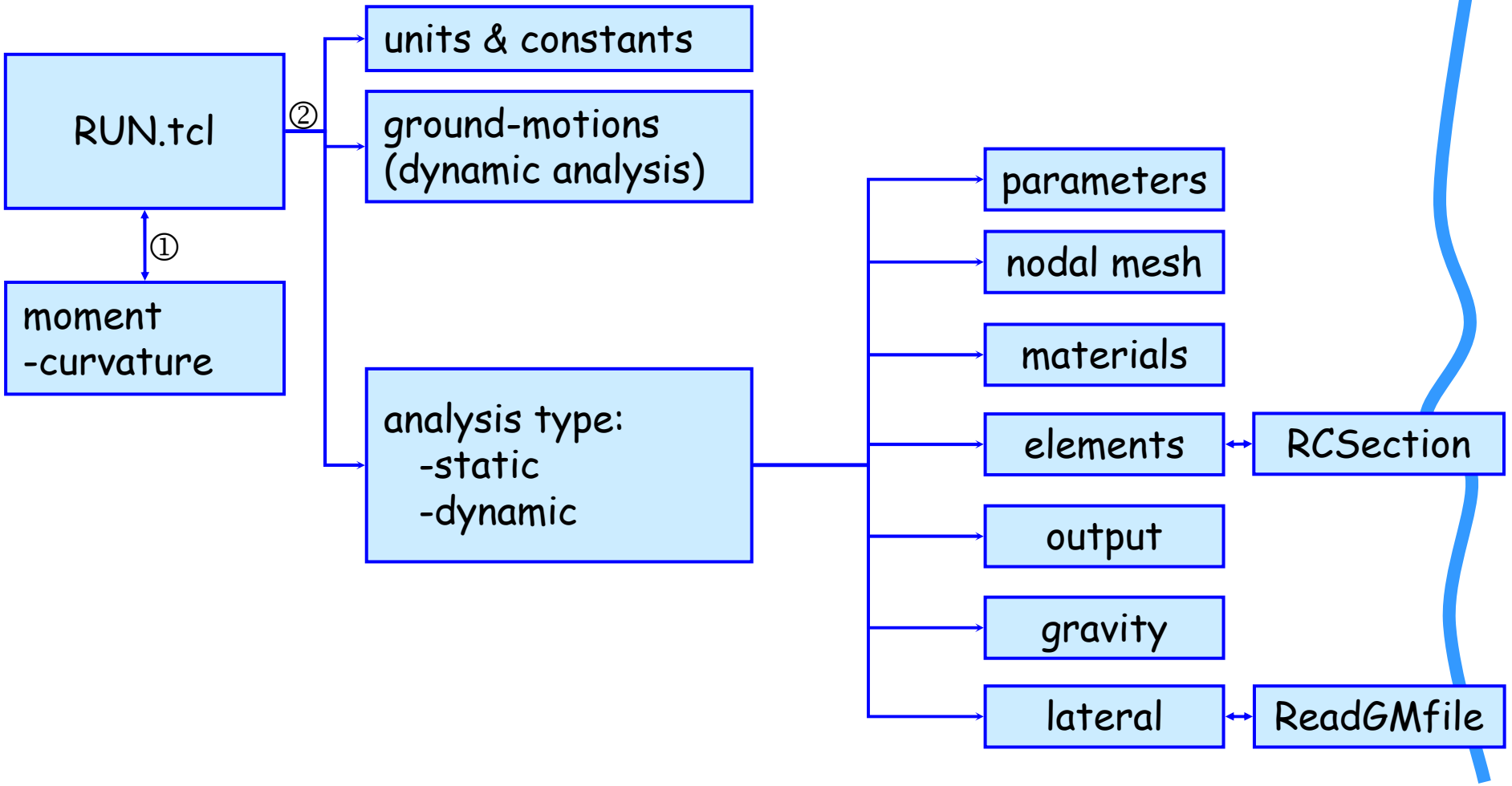
OpenSees User Workshop

14 August 2006

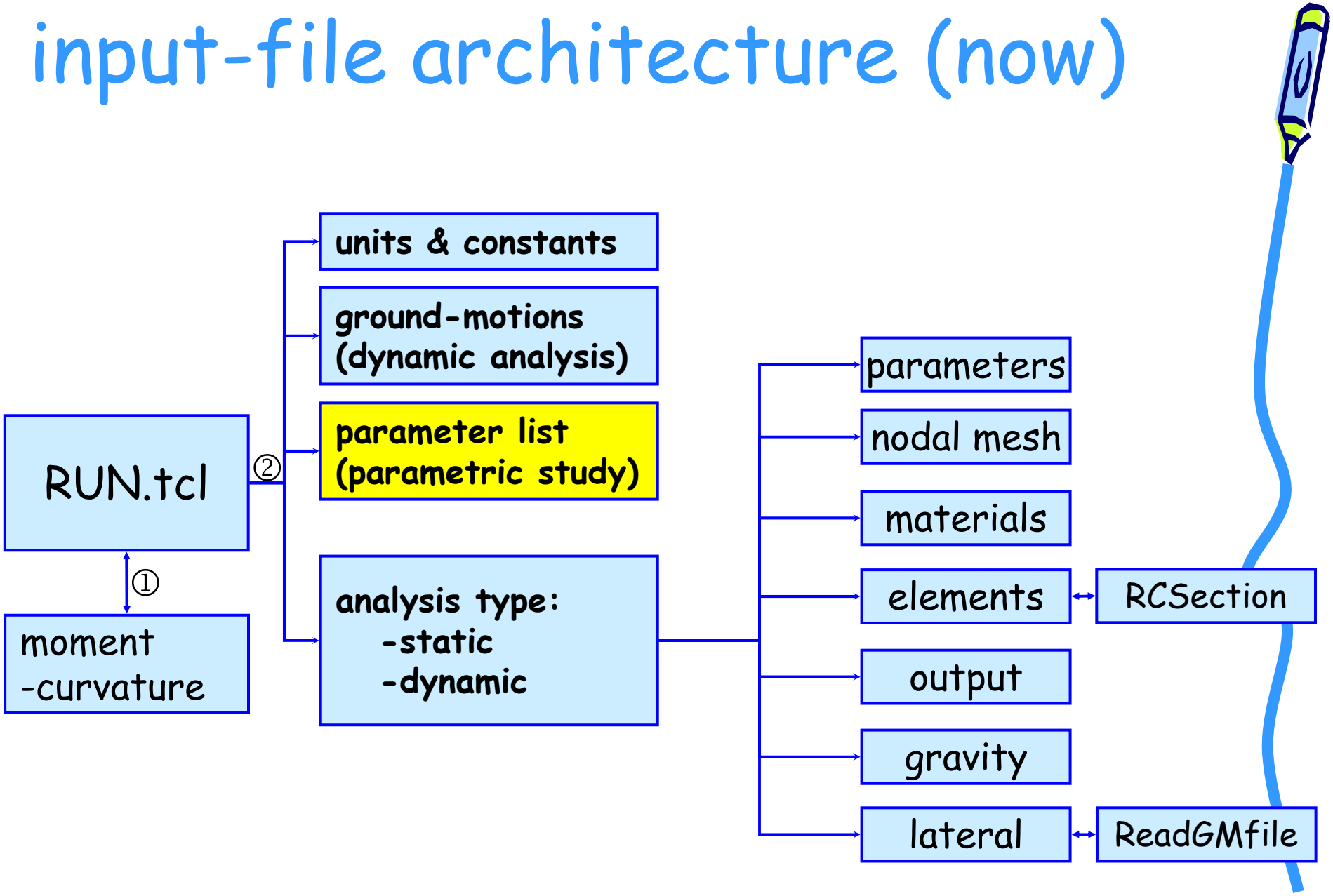
Tcl Scripting language -- advantages

- ability to "source-in" files
  - input-file architecture
- variables
  - unit and constant definition
  - parameter definition
- array management
  - set up parameter matrix
  - logical operations
  - for/while loops

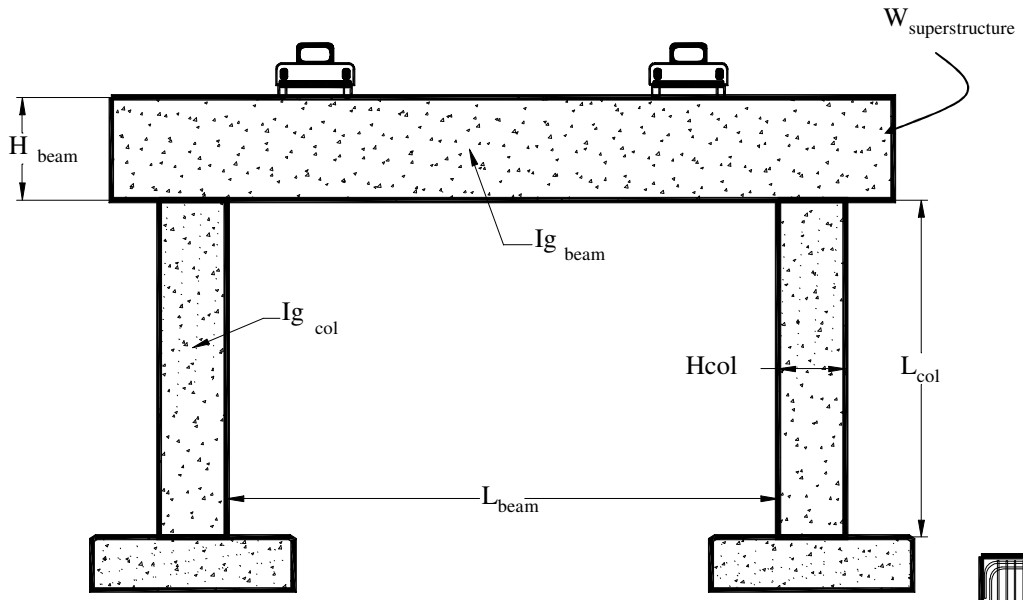
input-file architecture (before)

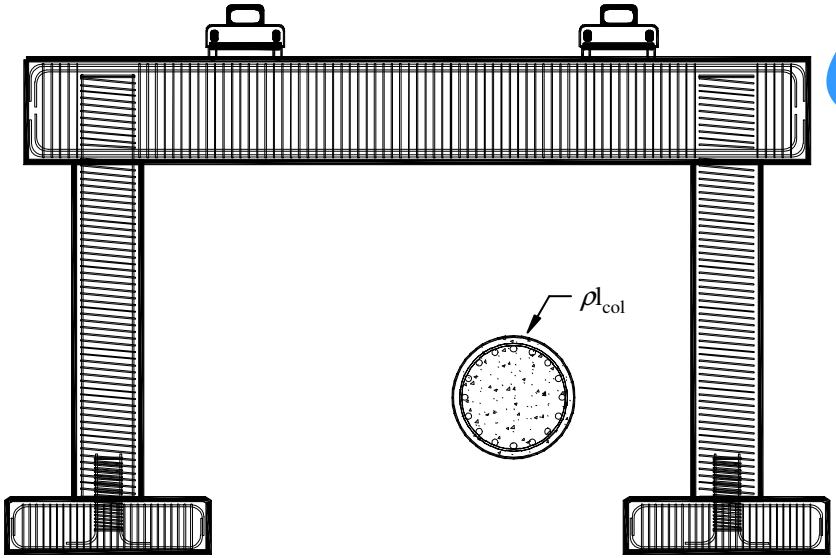


input-file architecture (now)



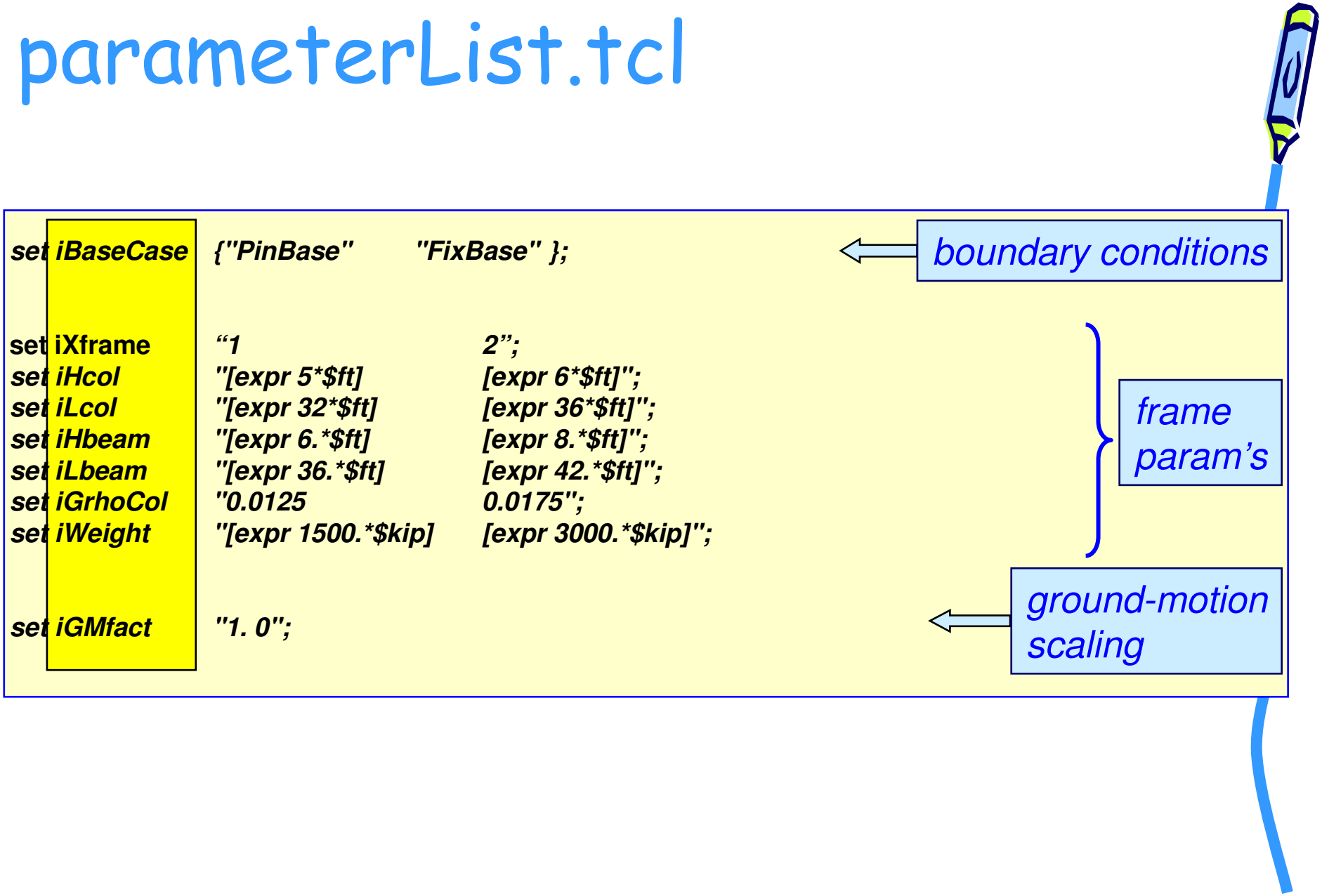
application I:
parametric study of bridge-bent model





OpenSees NEESit

parameterList.tcl



RUN.tcl (type 1)

```
1. source Units.tcl;                    # define units
2. source ParamList.tcl;                # load up parameter values
3. source GMFiles.tcl;                  # load up ground-motion filenames
4. foreach   BaseCase $iBaseCase {
     foreach Xframe $iXframe Hcol $iHcol Lcol $iLcol Lbeam $iLbeam GrhoCol $iGrhoCol
          Weight $iWeight GMfact $iGMfact Hbeam $iHbeam Bbeam $iBbeam  {        FRAME
          set ANALYSIS "Static";                                                STATIC
          source Analysis.tcl

          set ANALYSIS "Dynamic";                                               DYNAMIC
          foreach GroundFile $iGroundFile
               source Analysis.tcl                                              GROUND MOTION
          }
     }
}
```

RUN.tcl (type 2)

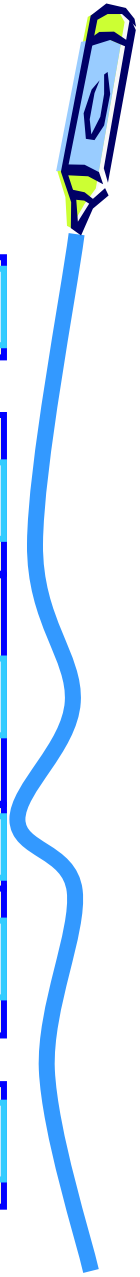
```
source Units.tcl;          source ParamList.tcl;    source GMFiles.tcl;

foreach BaseCase $iBaseCase {
    set Xframe 0;
    foreach Hcol $iHcol {
     foreach Lcol $iLcol {
      foreach Hbeam $iHbeam {
       foreach Lbeam $iLbeam {
        foreach GrhoCol $iGrhoCol {
         foreach Weight $iWeight {
          foreach GMfact $iGMfact {
                set Xframe [expr $Xframe+1];

                set ANALYSIS "Static";
                source Analysis.tcl

                set ANALYSIS "Dynamic";
                   foreach GroundFile $iGroundFile
                      source Analysis.tcl
                   }
    }; }; }; }; }; };};};
```

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
```

parameters.tcl

GEOMETRY

column

```
1.  set  Rcol     [expr $Hcol/2];                  # COLUMN radius
2.  set  Acol     [expr $PI*pow($Rcol,2)];         # column cross-sectional area
3.  set  cover    [expr $Hcol/15];                 # column cover width
4.  set  IgCol    [expr $PI*pow($Rcol,4)/4];       # column gross moment of inertia, uncracked
5.  set  IyCol    $IgCol;                          # elastic-column properties
6.  set  IzCol    $IgCol;                          # elastic-column properties
```

beam

```
7.  set  IzBeam   [expr $GIbIc*$IgCol];            # BEAM gross moment of inertia --
8.  set  Hbeam    [expr 8*$ft];                    # beam depth, not really used
9.  set  Bbeam    [expr $IzBeam*12/pow($Hbeam,3)]; # beam width not used
10. set  IyBeam   [expr $Hbeam*pow($Bbeam,3)/12];  # beam gross moment of inertia--vert Y
11. set  Abeam    [expr $Hbeam*$Bbeam*10000];      # beam cross-sectional area
12. set  GLbLc    [expr $Lbeam/$Lcol];             # beam-to-column length ratio
```

output.tcl

```
# Record nodal displacements -NODAL DISPLACEMENTS
set filename0 "data/$BaseCase/“
set filename1 DStatFrame[expr $Xframe]
set filename2 GM$GroundFile
set iNode "3 4";
foreach xNode $iNode {
        set filename3 Node$xNode
        set filename $filename0$filename3$filename1$filename2
        recorder Node $filename.out disp -node $xNode -dof 1 2  6;
};              # end of xNode
```

directory

frame ID

ground motion

node no.

example filename: data/Pinbase/Node3DStatFrame1ElCentro.out

generating matlab input

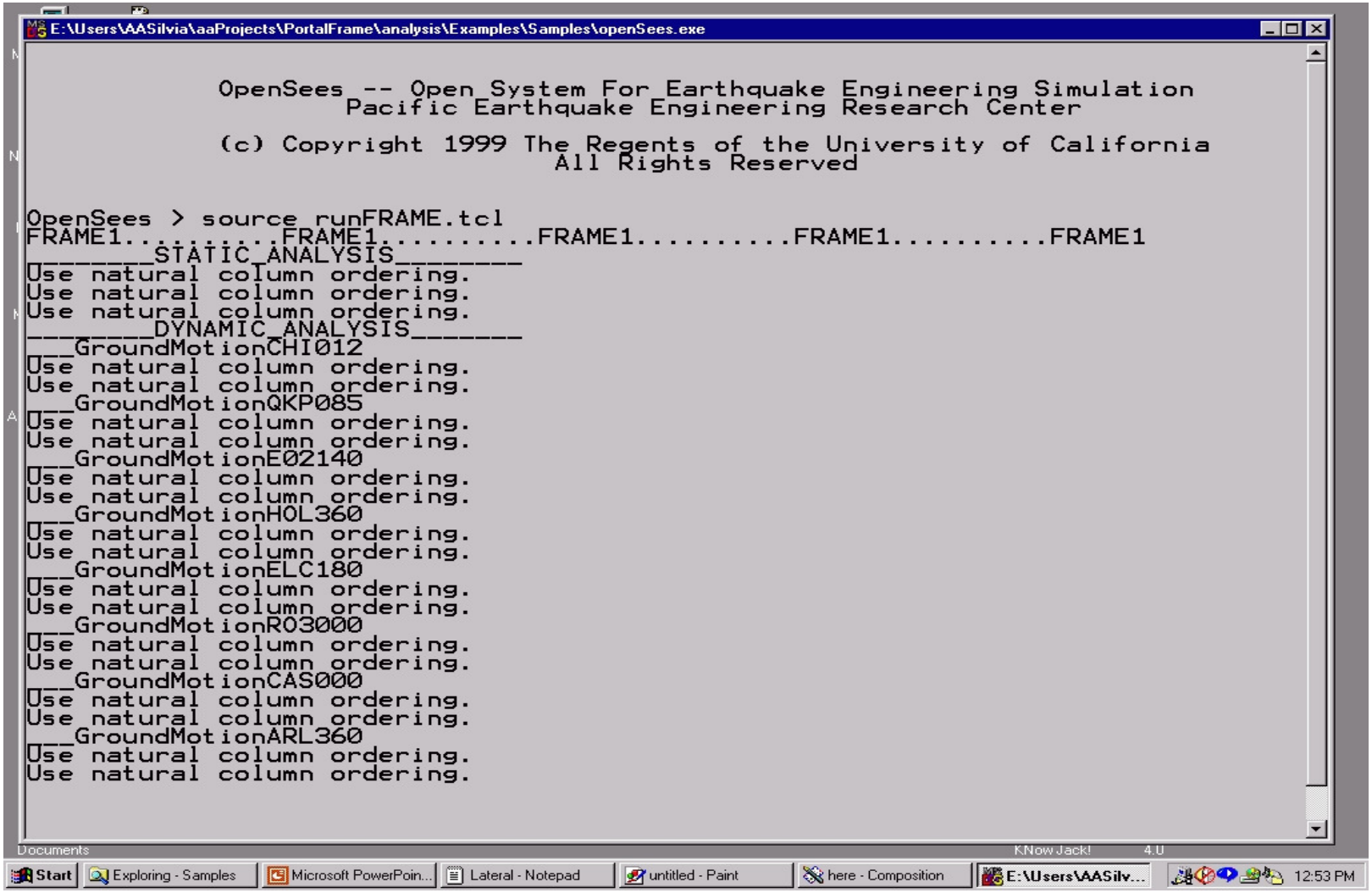
```
set datadir "Data/"
# Open output file for writing
set outFileID [open Data/DataFrame$Xframe.m w]

puts $outFileID   "Xframe($Xframe) = $Xframe;";     # frame ID
puts $outFileID   "Hcol($Xframe) = $Hcol;";         # column diameter
puts $outFileID   "Lcol($Xframe) = $Lcol;" ;        # column length
puts $outFileID   "Lbeam($Xframe) = $Lbeam;";       # beam length
puts $outFileID   "Hbeam($Xframe) = $Hbeam;";       # beam depth
puts $outFileID   "Bbeam($Xframe) = $Bbeam;";       # beam width
puts $outFileID   "GrhoCol($Xframe) = $GrhoCol;";# column long.-steel ratio
puts $outFileID   "GPcol($Xframe) = $GPcol;";       # Col.axial load:strength
puts $outFileID   "GMfact($Xframe) = $GMfact;";     # ground-mot. scaling fact
puts $outFileID   "Acol($Xframe) = $Acol;";         # column cross-sect. area
puts $outFileID   "Weight($Xframe) = $Weight;";     # superstructure weight
```

dataframe3.m

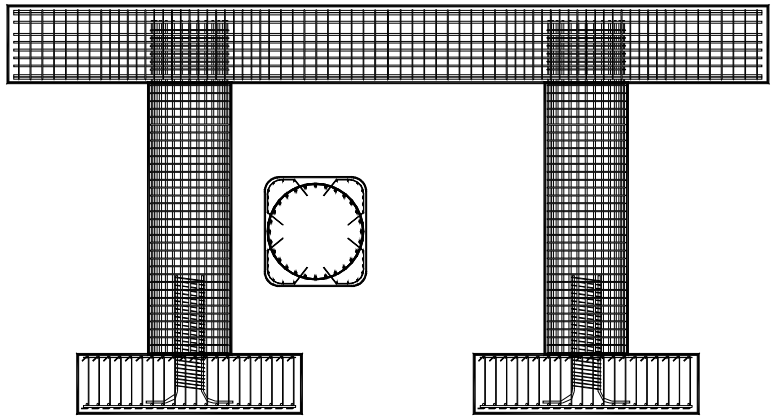
- Xframe(3) = 3;
- Hcol(3) = 78.0;
- Lcol(3) = 432.0;
- Lbeam(3) = 432.0;
- Hbeam(3) = 96.0;
- Bbeam(3) = 78.0;
- GrhoCol(3) = 0.0125;
- GPcol(3) = 0.0570754682058;
- GMfact(3) = 1.5;
- Acol(3) = 4778.36242611;
- Weight(3) = 3000.0;

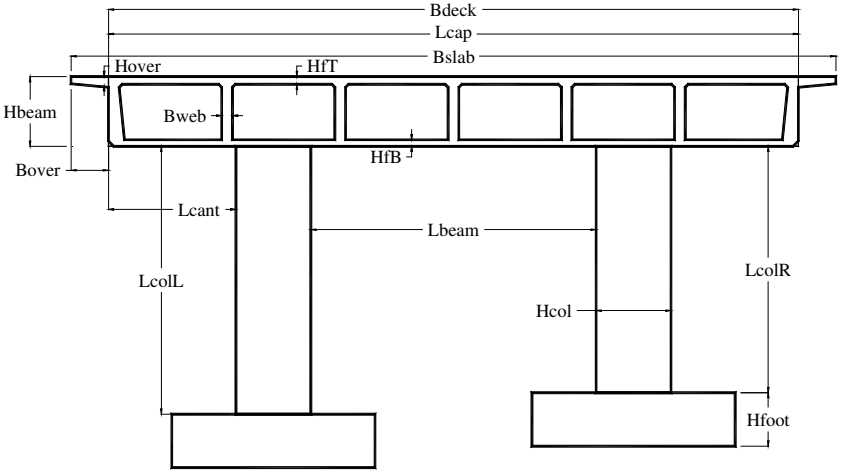
analyses running



application II:
3D model of bridge frame

Cypress-Street Viaduct Replacement Structure





bent detail

frame detail

OpenSees NEESit

nodalmesh.tcl – a number of bents

```
set iNode0   "100   200    300"
set iZbent "[expr 0*$ft] [expr 200*$ft] [expr 400*$ft]"
```

define nodes of each bent

```
foreach Node0 $iNode0 Zoffset $iZbent {
        # Define nodes  ------ frame is in X-Y plane (X-horizontal, Y-vertical)
        #       tag          X                  Y              Z     <-mass MX MY MZ RX  RY  RZ> (nodal masses)
        node [expr $Node0 + 1]      [expr -$Lbeam/2.]     [expr -$LcolL]   $Zoffset
        node [expr $Node0 +  2]     [expr  $Lbeam/2.]     [expr -$LcolR]   $Zoffset
        node [expr $Node0 +  3]     [expr -$Lbeam/2.]     0                $Zoffset   -mass $Mnode 0.0 0.0 0.0 0.0 0.0
        node [expr $Node0 +  4]     [expr  $Lbeam/2.]     0                $Zoffset   -mass $Mnode 0.0 0.0 0.0 0.0 0.0
        node [expr $Node0 +  5]     [expr -$Lbeam/2.- $Lcant]       0      $Zoffset;  #overhang
        node [expr $Node0 +  6]     [expr   $Lbeam/2.+$Lcant]       0      $Zoffset;
        node [expr $Node0 +  13]    [expr -$Lbeam/2.]     0                $Zoffset;
        node [expr $Node0 +  14]    [expr  $Lbeam/2.]     0                $Zoffset
        node [expr $Node0 +  23]    [expr -$Lbeam/2.]     0                $Zoffset;
        node [expr $Node0 +  24]    [expr  $Lbeam/2.]     0                $Zoffset


        # Boundary conditions   # node DX DY DZ RX RY RZ      # 1: fixed, 0: released
        fix [expr $Node0 + 1]              1  1  1  0  1  0;              # pin support
.........
        #            5-----4-----3-----------3-----------4------5-----6
        #                        |                       |
        #                        |                       |
        #                      --1--                   --2--              ^  Y
        #                        |                       |                |
        #                        |                       |                |
        #                        1                       2                ----------> X
}
```

elements.tcl

```
set iNode0   "100   200   300";
set iElem0   "100   200   300"
```

elem. #

node i node j

define elements of each bent

```
foreach Node0 $iNode0 Elem0 $iElem0 {
    # Define COLUMNS
    element nonlinearBeamColumn [expr $Elem0 + 1]  [expr $Node0 + 1]  [expr $Node0 +3]  $np  $IDcolSec  $ZZ;
    element nonlinearBeamColumn [expr $Elem0 + 2]  [expr $Node0 + 2]  [expr $Node0 +4]  $np  $IDcolSec  $ZZ;

    #IDphtMat  has been defined in parameters.tcl and materials.tcl  -- TOP PLASTIC HINGE, connects column to joint
    rotSpringDOF6 [expr $Elem0 + 11]  [expr $Node0 +3]  [expr $Node0 +13]  $IDphtMat  [expr $Node0 +3];
    rotSpringDOF6 [expr $Elem0 + 12]  [expr $Node0 +4]  [expr $Node0 +14]  $IDphtMat  [expr $Node0 +4]

    #IDjointMat  has been defined in parameters.tcl and materials.tcl -- JOINT, connects hinge to beam
    rotSpringDOF6 [expr $Elem0 + 21]  [expr $Node0 +13]  [expr $Node0 +23]  $IDjointMat  [expr $Node0 +3];
    rotSpringDOF6 [expr $Elem0 + 22]  [expr $Node0 +14]  [expr $Node0 +24]  $IDjointMat  [expr $Node0 +4]

    # Define BEAM
    element elasticBeamColumn [expr $Elem0 + 3]  [expr $Node0 +23]  [expr $Node0 +24]  $Abeam $Ec $GJ 1.0 $IyBeam $IzBeam $ZZ
    element elasticBeamColumn [expr $Elem0 + 4]  [expr $Node0 +5]   [expr $Node0 +23]  $Abeam $Ec $GJ 1.0 $IyBeam $IzBeam $ZZ
    element elasticBeamColumn [expr $Elem0 + 5]  [expr $Node0 +24]  [expr $Node0 +6]   $Abeam $Ec $GJ 1.0 $IyBeam $IzBeam $ZZ

};  # end foreach node and element in a bent
```

connect bents

```
element elasticBeamColumn   501   [expr $Node0Bent1 +3]   [expr $Node0Bent2 +3]   $Abeam $Ec $GJ 1.0 $U $U $YY
element elasticBeamColumn   502   [expr $Node0Bent2 +3]   [expr $Node0Bent3 +3]   $Abeam $Ec $GJ 1.0 $U $U $YY
element elasticBeamColumn   503   [expr $Node0Bent1 +4]   [expr $Node0Bent2 +4]   $Abeam $Ec $GJ 1.0 $U $U $YY
element elasticBeamColumn   504   [expr $Node0Bent2 +4]   [expr $Node0Bent3 +4]   $Abeam $Ec $GJ 1.0 $U $U $YY
```

conclusions

- advantages of Tcl scripting language simplifies the following:
  - parameter studies
  - simplify error check
  - generate a new input file while using components of a previously-generated and tested input file