OpenSees

Continuum Modeling With Emphasis on Geotechnical

Frank McKenna
UC Berkeley

OpenSees Days Shanghai 2011

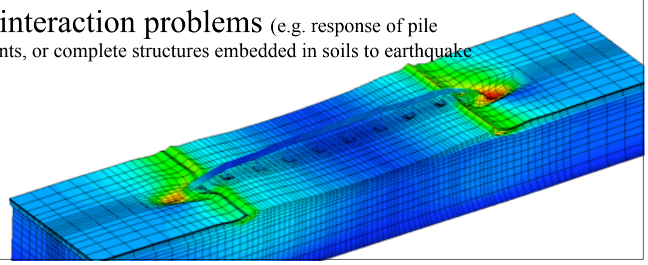
NEES NEEScomm

Outline of Presentation

- Why
- Elements for Continuum Modeling
- Mateials for Continuum Modeling
- Simple Examples

- Static Problems
  - Deformation analyses
  - Consolidation problems (diffusion problems)
  - Soil-structure interaction problems
    - Shallow foundations (e.g. bearing capacity, settlements)
    - Pile foundations (e.g. vertical and lateral capacity)

- Dynamic (earthquake problems)
  - Free-field analysis
  - Liquefaction induced problems
  - Soil structure interaction problems (e.g. response of pile foundations, bridge bents, or complete structures embedded in soils to earthquake excitations)



Single & Multiphase Models

- Single Phase Models
  - Structural Modeling
  - Dry Soils
- Multi Phase Models (Soil + Water)
  - Phase 1 for Soil Skeleton
  - Phase 2 for Water (Pore Pressure)

nD Materials

- Materials:
  - Elastic
  - DruckerPrager
  - J2 (VonMises)
  - Cam-Clay
  - PressureDependMultiYield (sand)
  - PressureIndependMultiYield (clay)
  - others

Additional commands for **multiyield** materials

- Help perform stage analysis

updateMaterialStage –material $matTag **–stage** $sNum

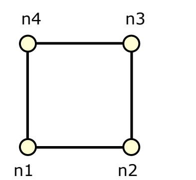
$MatTag → the tag of previously defined material
$sNum → (0 - elastic, 1-plastic, 2 – linear elastic constant **f(**σ_3**)**)

updateParameter –material $matTag **–refG** $newVal

$MatTag → the tag of previously defined material
$sNewVal → new parameter value

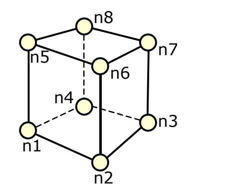
Single Phase Elements

- Quad (4,9 nodes)



quad (4 node)

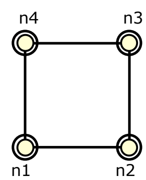
- Brick (8, 20 nodes)



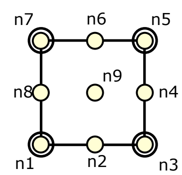
stdBrick (8 node)

Multi Phase Elements

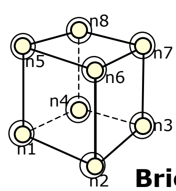
- Fully coupled u-p elements (2D & 3D)
- Fully coupled u-p-U elements (3D) for small deformations



quadUP



9_4_quadUP



BrickUP

Degrees of Freedom (DOFs) are:

- u → solid displacement, on ○
- P → pore fluid pressures, on ○
- U → pore fluid displacements, on ○

Simply Supported Beam

n.tcl

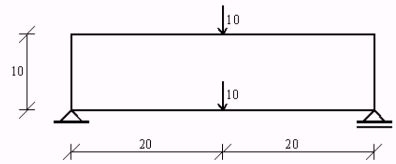


Fig. 1 Geometry and static loads

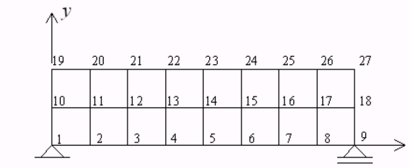


Fig. 2 Finite element mesh and node numbering

```
# some problem parameters
set L 40.0
set H 10.0
set thick 2.0
set P 10
set nX 9; # numNodes x dirn
set nY 3; # numNodes y dirn
# model builder
model Basic -ndm 2 -ndf 2
# create material
nDMaterial ElasticIsotropic 1 1000 0.25  3.0
```

```
# create nodes
set nodeTag 1
set yLoc 0.0;
for {set i 0} {$i < $nY} {incr i 1} {
    set xLoc 0.0;
    for {set j 0} {$j < $nX} {incr j 1} {
        node $nodeTag $xLoc $yLoc
        set xLoc [expr $xLoc+ $L/($nX-1.0)]
        incr nodeTag
    }
    set yLoc [expr $yLoc+ $H/($nY-1.0)]
}
# boundary conditions
fix  1  1 1
fix $nX 1 1
# create elements
set eleTag 1
for {set i 1} {$i < $nY} {incr i 1} {
    set iNode [expr 1+($i-1)*$nX];
    set jNode [expr $iNode+1];
    set kNode [expr $jNode+$nX]
    set lNode [expr $iNode+$nX]
    for {set j 1} {$j < $nX} {incr j 1}
        element quad $eleTag $iNode $jNode $kNode $lNode \
             $thick "PlaneStress" 1
        incr eleTag; incr iNode; incr jNode; incr kNode; incr lN
    }
}
# apply loads
set midNode [expr ($nX+1)/2]
timeSeries Linear 1
pattern Plain 1 1 {
    load $midNode 0 -$P
    load [expr $midNode + $nX*($nY-1)] 0 -$P
}
analysis Static;
analyze 1; print node $midNode
```

Simply Supported Beam

o.tcl

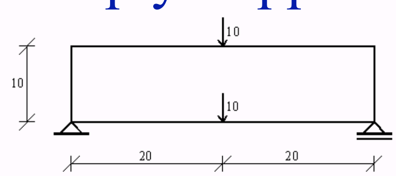


Fig. 1 Geometry and static loads

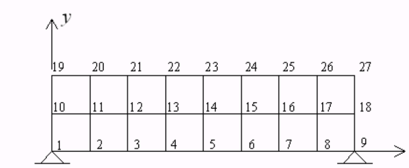


Fig. 2 Finite element mesh and node numbering

```
# some problem parameters
set L 40.0
set H 10.0
set thick 2.0
set P 10
set nX 9; # numNodes x dirn
set nY 3; # numNodes y dirn

# model builder
model Basic -ndm 2 -ndf 2
# create material
nDMaterial ElasticIsotropic 1 1000 0.25  3.0
```

```
# use block command
set cmd "block2D [expr $nX-1] [expr $nY-1] 1 1 \
    quad \" $thick PlaneStress 1\" {
        1   0   0
        2  $L   0
        3  $L  $H
        4   0  $H
}"

eval $cmd

# apply loads
set midNode [expr ($nX+1)/2]
timeSeries Linear 1
pattern Plain 1 1 {
    load $midNode 0 -$P
    load [expr $midNode + $nX*($nY-1)] 0 -$P
}
analysis Static;
analyze 1;
print node $midNode
```

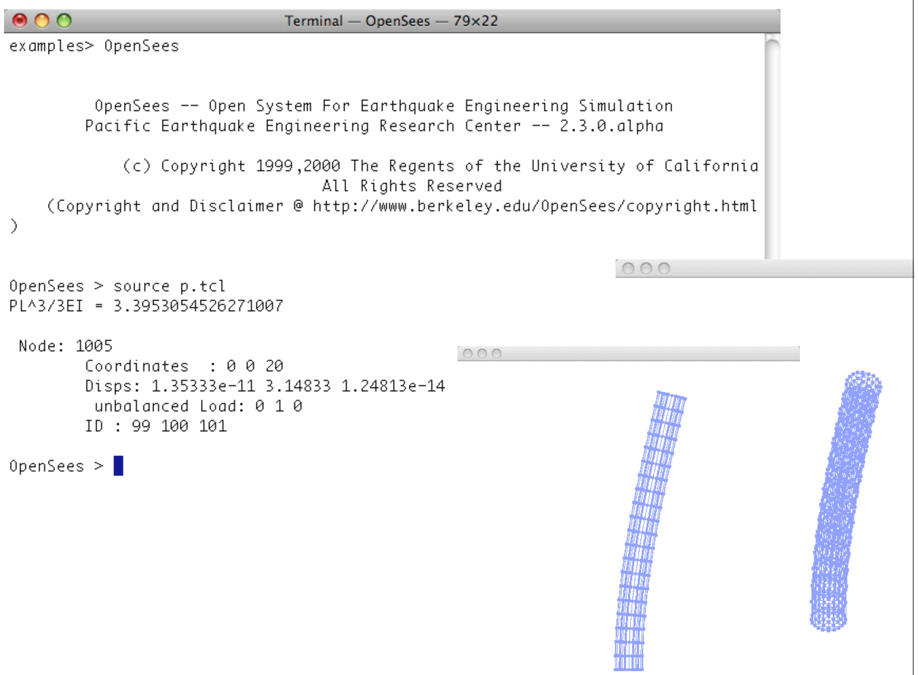
```
Terminal — bash — 85×37
examples> OpenSees n.tcl


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.3.0.alpha

            (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: 5
        Coordinates  : 20 0
        Disps: -1.37853e-16 -0.096041
         unbalanced Load: 0 -10
        ID : 26 27

examples> OpenSees o.tcl


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.3.0.alpha

            (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: 5
        Coordinates  : 20 0
        Disps: -1.37853e-16 -0.096041
         unbalanced Load: 0 -10
        ID : 26 27

examples> 
```

Cantilevered Circular Column

p.tcl

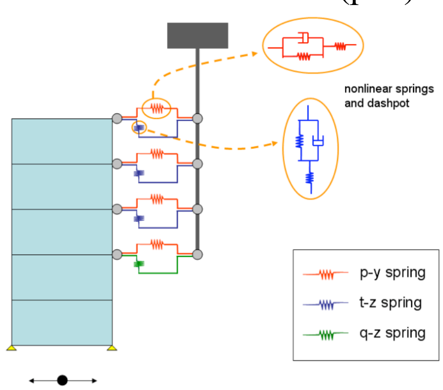
```
set P 1.0
set L 20.0
set R 1.0
set E 1000.0
set nz  20
set nx  6
set ny  6
set PI [expr 2.0 * asin(1.0)]
set I  [expr $PI*pow((2*$R),4)/64.0]
puts "PL^3/3EI = [expr $P*pow($L,3)/(3.0*$E*$I)

# Create ModelBuilder with 3 dimensions and 6 D
model Basic -ndm 3 -ndf 3

# create the material
nDMaterial ElasticIsotropic   1   $E   0.25  1.27

set eleArgs "1"
set element bbarBrick

set nn [expr ($nz)*($nx+1)*($ny+1) + (($nx+1)*(
set n1 [expr ($nz)*($nx+1)*($ny+1) +$nx]
```

```
# mesh generation
set sqrtR [expr sqrt($R/2.0)]
set cmd "block3D $nx $ny $nz   1 1  $el
     1   -$sqrtR    -$sqrtR     0
     2   $sqrtR     -$sqrtR     0
     3   $sqrtR     $sqrtR      0
     4   -$sqrtR    $sqrtR      0
     5   -$sqrtR    -$sqrtR     $L
     6   $sqrtR     -$sqrtR     $L
     7   $sqrtR     $sqrtR      $L
     8   -$sqrtR    $sqrtR      $L
    13    0      -$R       0
    14    $R      0        0
    15    0       $R       0
    16   -$R      0        0
    18    0      -$R       $L
    19    $R      0        $L
    20    0       $R       $L
    21   -$R      0        $L
    23    0      -$R       [expr $L/2.0]
    24    $R      0        [expr $L/2.0]
    25    0       $R       [expr $L/2.0]
    26   -$R      0        [expr $L/2.0]
  }"
eval $cmd

# boundary conditions
fixZ 0.0   1 1 1

# Constant point load
pattern Plain 1 Linear {
   load $nn  0.0 $P 0.0
}
```



Uniaxial models for Soil-Structure Interaction Models

- To capture interface response between solid (soil) and beam elements (pile)



Py Tz Qz Uniaxial Materials

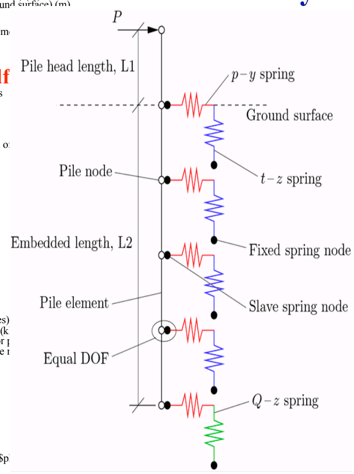
- PySimple1
- TzSimple1
- QzSimple1

- PyLiq1
- TzLiq1

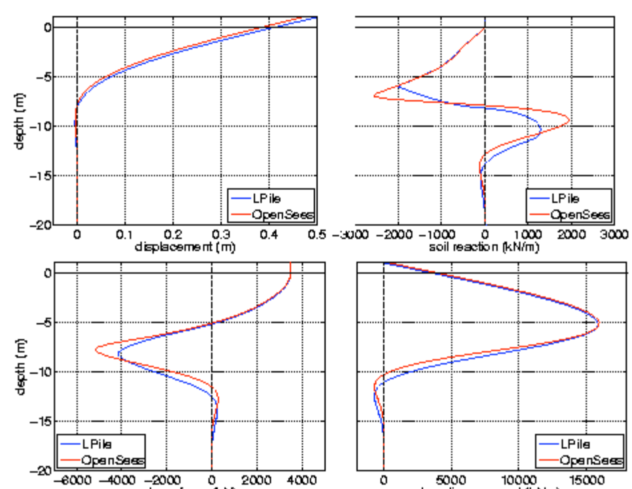
Lateral Pile Analysis

pile/pile.tcl

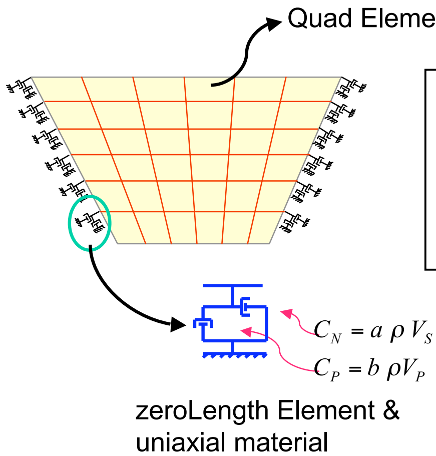
```
set L1 1.0; # length of pile head (above ground surface) (m)
set L2 20.0; # length of embedded pile (below ground surface) (m)
set diameter 1.0; # pile diameter
set nElePile 84; # number of pile elements
set eleSize [expr ($L1+$L2)/$nElePile]; # pile elem
# number of total pile nodes
set nNodePile [expr 1 + $nElePile]
# spring nodes created with 3 dim, 3 dof
model Basic -ndm 3 -ndf
# counter to determine number of embedded nodes
set count 0
# create spring nodes
for {set i 1} {$i <= $nNodePile} {incr i} {
   set zCoord [expr $eleSize*($i - 1)]
  # only create spring nodes over embedded length o
   if {$zCoord <= $L2} {
      node $i            0.00   0.00   $zCoord
      node [expr $i+100]  0.00   0.00   $zCoord
      set count [expr $count+1]
   }
}
# number of embedded nodes
set nNodeEmbed $count

# spring node fixities
for {set i 1} {$i <= $nNodeEmbed} {incr i} {
   fix $i            1   1   1
   fix [expr $i+100]  0   1   1
}
#  soil properties
set gamma  17.0; # soil unit weight (kN/m^3)
set phi    36.0; # soil internal friction angle (degrees)
set Gsoil  150000; # soil shear modulus at pile tip (k
set puSwitch 1 ; # select pult definition method for p
set kSwitch 1; # variation in coefficent of subgrade r

#  create spring material objects
source get_pyParam.tcl
source get_tzParam.tcl
source get_qzParam.tcl
# p-y spring material
for {set i 1} {$i <= $nNodeEmbed} {incr i} {
  # depth of current py node
   set pyDepth [expr $L2 - $eleSize*($i - 1)]
   set pyParam [get_pyParam $pyDepth $gamma $p
   set pult [lindex $pyParam 0]
   set y50  [lindex $pyParam 1]
   uniaxialMaterial PySimple1 $i  2  $pult  $y50  0.0
}
# t-z spring material
for {set i 2} {$i <= $nNodeEmbed} {incr i} {
  # depth of current tz node
     set pyDepth [expr $eleSize*($i - 1)]
  # vertical effective stress at current depth
   set sigV    [expr $gamma*$pyDepth]
   set tzParam  [get_tzParam $phi $diameter $sigV $eleSize]
   set tult [lindex $tzParam 0]
   set z50  [lindex $tzParam 1]
     uniaxialMaterial TzSimple1 [expr $i+100]  2  $tult  $z50  0.0
}
# q-z spring material
  # vertical effective stress at pile tip, no water table (depth is embedded pile length)
  set sigVq   [expr $gamma*$L2]
   set qzParam [get_qzParam $phi $diameter $sigVq $Gsoil]
   set qult [lindex $qzParam 0]
   set z50q [lindex $qzParam 1]
   uniaxialMaterial QzSimple1 101  2  $qult  $z50q  0.0  0.0
```




```
# element at the pile tip (has q-z spring)
element zeroLength 1001  1  101  -mat 1  101  -dir 1  3

# remaining elements
for {set i 2} {$i <= $nNodeEmbed} {incr i} {
   element zeroLength [expr $i+1000]  $i  [expr $i+100]  -mat $i [expr $i+100]  -
}
puts "Finished creating all zero-Length elements for springs..."

# pile nodes created with 3 dimensions, 6 degrees of freedom
model BasicBuilder -ndm 3 -ndf 6

# create pile nodes
for {set i 1} {$i <= $nNodePile} {incr i} {
  # z-coordinates of nodes depend on element length
   set zCoord [expr $eleSize*($i - 1)]

   node [expr $i+200] 0.00   0.00   $zCoord
}
puts "Finished creating all pile nodes..."

# create coordinate-transformation object
geomTransf Linear     1         0.0   -1.0  0.0

# create fixity at pile head (location of loading)
fix [expr 200+$nNodePile]     0    1    0    1    0    1

# create fixities for remaining pile nodes
for {set i 202} {$i < [expr 200+$nNodePile]} {incr i} {
   fix $i    0    1    0    1    0    1
}
puts "Finished creating all pile node fixities..."

#  define equal dof between pile and spring nodes
for {set i 1} {$i <= $nNodeEmbed} {incr i} {
   equalDOF [expr $i+200] [expr $i+100]   1
}
# elastic pile section
source elasticPileSection.tcl
for {set i 201} {$i <= [expr 200+$nElePile]} {incr i} {
   element dispBeamColumn  $i  $i  [expr $i+1]  $secTag3D   3   1
}
setTime 10.0

# apply point load at the uppermost pile node in the x-direction
pattern Plain 10 {Series -time {0 10 20 10000} -values {0 0 1 1} -factor 1} {
      load [expr 200+$nNodePile]  3500  0.0  0.0  0.0  0.0  0.0
}

#  create the analysis
integrator LoadControl  0.05
numberer RCM
system SparseGeneral
constraints Transformation
test NormDispIncr 1e-5      20      1
algorithm Newton
analysis Static
analyze         201
```

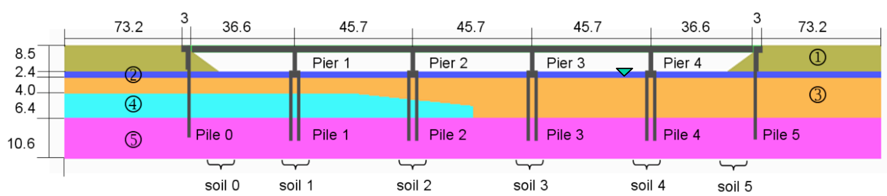


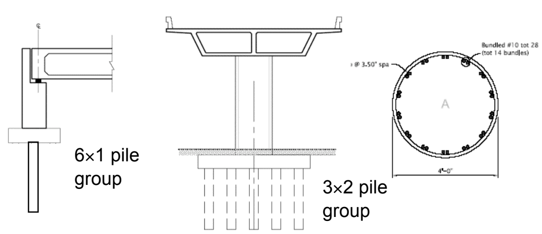
Absorbent Boundaries
Lysmer (1969)




```
1. set DampP 755
2. set DampN 1216
3. uniaxialMaterial Elastic  1  0 $DampP
4. uniaxialMaterial Elastic  2  0 $DampN
5. node 1 16.0 0.0
6. node 2 16.0 0.0
7. element zeroLength 1 1  2  -mat 1  2
    -dir 1 2 –orient 1 –2 0 2 1 0
```

Target bridge system



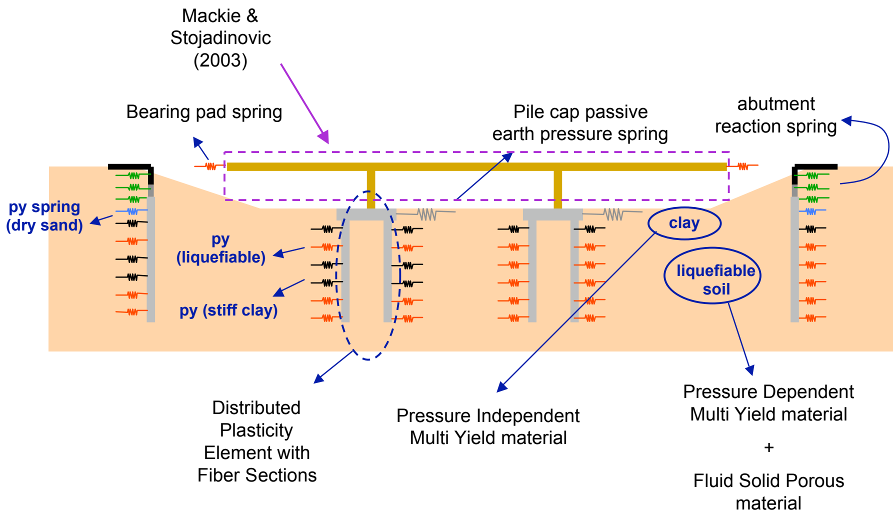


open-ended steel pipe pile

(ϕ = 2 ft, t = 0.5 in.)

① embankment
② med. stiff clay
③ loose-med sand
④ stiff clay
⑤ dense sand

Numerical modeling for target bridge system



Other useful **tcl** scripts @

- **`http://opensees.berkeley.edu/`**
- **`http://sokocalo.engr.ucdavis.edu/~jeremic`**
- **`http://cyclic.ucsd.edu/opensees/`**
- **`http://www.ce.washington.edu/~geotech/opensees/PEER/davis_meeting/`**