

## ***ShallowFoundationGen* OpenSees Documentation**

Prishati Raychowdhury<sup>1</sup> and Tara C. Hutchinson<sup>2</sup>

January, 2008

<sup>1</sup>Ph.D. Candidate, University of California, San Diego; prishati@gmail.com

<sup>2</sup>Associate Professor, University of California, San Diego; tara@ucsd.edu

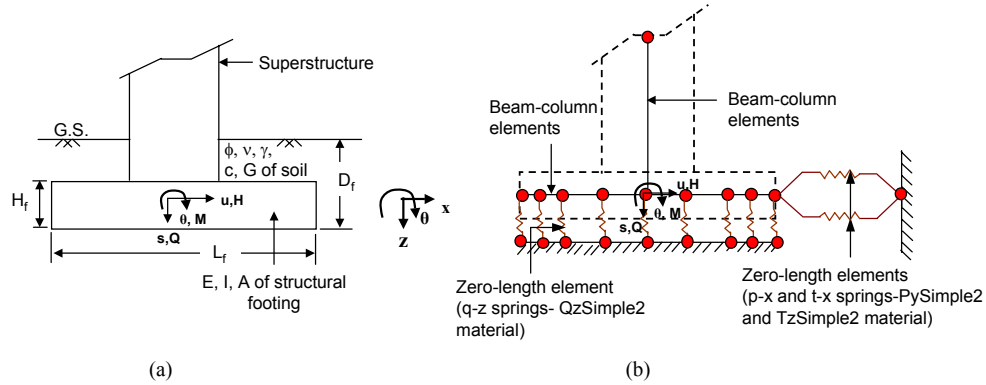
Modeling of the 2-dimensional rocking, sliding and settling response of a shallow foundation using a Beam-on-Nonlinear-Winkler-Foundation (BNWF) concept (Figure 1) is a complicated and cumbersome task. This is because the modeling requires defining (i) the node numbers and corresponding coordinates for foundation and spring elements (p-x, t-x and q-z), (ii) material type and properties of each created element and (iii) nodal fixity. Within OpenSees, this is currently achieved by writing many lines of Tcl script. With multiple shallow foundations, the procedure becomes even more cumbersome and confusing as one has to keep track of node and element numbering manually or needs to employ a systematic numbering scheme for this.

The *ShallowFoundationGen* command, which is integrated with the OpenSees program, overcomes the above-mentioned difficulties in a very simple and elegant way, while adding some extra modeling options. The basis of selection of model parameters hardwired within *ShallowFoundationGen* have been calibrated against centrifuge and other experiments (Raychowdhury and Hutchinson, 2008). *ShallowFoundationGen* assumes the user has defined the superstructure nodes and elements, their associated material properties, the system loading and solution strategies needed to complete the problem analysis. In addition, the materials used within the *ShallowFoundationGen* restrict the problem to foundations supported on non-strain-softening materials. The command returns important variables such as the element numbers of middle and end springs and node numbers of end nodes, which are required to determine general footing response. This documentation provides a roadmap for users modeling input and output format, limitations of the input parameters and examples for the users. The first example deals with a shearwall supported on a strip footing system (single foundation) and the second example demonstrates the use multiple foundations. The *ShallowFoundationGen* command, which requires four arguments, is as follows:

***ShallowFoundationGen* \$FoundationTag \$ConnectNode \$InputFileName \$FootingCondition**

The first argument is a tag associated with the foundation under consideration. The second argument is the node tag of the structural node that is to be connected with the middle node of the foundation. The third argument is the name of the input file which contains soil and foundation properties that are required to define the springs of the Winkler mesh. The fourth argument, which provides some extra

modeling options, is the foundation condition tag. The foundation condition tag specifies whether the foundation is modeled as fixed base, elastic or nonlinear. The command *ShallowFoundationGen* creates a Tcl output file with all information needed to describe the foundation, including the nodes, footing elements, spring zerolength elements and necessary materials to define those elements. This is then sourced in the next line of the master Tcl source file.



Axis	Force	Displacement
x	H	u
z	Q	s
$\theta$	M	$\theta$

Notes:

1. Superstructure elements are not generated by the **ShallowFoundationGen** command. The user must specify the mid node (node in the schematic).
2. p-x and t-x springs are connected to identical end nodes with zero distance between them, however, are they independent.

**FIG. 1. Schematic diagram of (a) superstructure-foundation system and (b) BNWF showing springs with their orientations**

**Argument 1: \$FoundationTag**

The foundation tag is an integer number denoting the foundation number. The output file generated by the *ShallowFoundationGen* command will be named as per the foundation tag specified. For example, if a structure has three shallow foundations, and the foundations are numbered as 1, 2 and 3, then the foundation tags for those three foundations would be 1, 2 and 3, respectively, and the OpenSees generated output files will automatically be named as “*Foundation\_1.tcl*”, “*Foundation\_2.tcl*” and “*Foundation\_3.tcl*”, respectively.

**Argument 2: \$ConnectNode**

The analyst has to specify the node of the structure that is to be connected with the middle node of the foundation. Note that the code assumes that the structure is supported at the center of a symmetric footing. For example, if the user specifies the *ConnectNode* to be 1, node 1 will be connected to the mid node of the shallow foundation, and the degrees of freedom of these two nodes will be the same (i.e. same generalized displacements for these two nodes).

**Argument 3: \$InputFileName**

The analyst must provide an input file containing the soil and footing properties required to calculate the strength and stiffness parameters for the BNWF

springs. The name of the file can be arbitrary. The first uncommented line (lines that are not starting with #) of this file should contain parameters describing the soil in the following sequence:

*SoilProp \$SoilType \$c \$φ \$γ \$G \$ν \$c<sub>rad</sub> \$TP*

where, Soil type = type of soil, i.e., sand or clay should be denoted by number 1 or 2. Soil type = 1 represents clay and Soil type = 2 represents sand;  $c$  = Cohesion,  $φ$  = Friction angle (in degree),  $Γ$  = Unit weight,  $G$  = Shear modulus,  $ν$  = Poisson's ratio,  $c_{rad}$  = Radiation damping (in fraction),  $TP$  = Tension capacity (in fraction).

Using the above defined parameters, capacity and stiffness of the foundation calculated. For vertical QzSimple2 springs, the ultimate bearing capacity is calculated based on bearing capacity equation after Terzaghi (1943) (Equation 1). The bearing capacity factors along with depth, shape and inclination factors are calculated after Meyerhof (1963).

$$q_{ult} = cN_cF_{cs}F_{cd}F_{ci} + \gamma D_f N_q F_{qs} F_{qd} F_{qi} + 0.5\gamma B N_\gamma F_{\gamma s} F_{\gamma d} F_{\gamma i} \quad (1)$$

where  $q_{ult}$  = ultimate bearing capacity per unit area of footing,  $c$  = cohesion,  $\gamma$  = unit weight of soil,  $D_f$  = depth of embedment,  $B$  = width of footing;  $N_c$ ,  $N_q$  and  $N_\gamma$  are bearing capacity factors,  $F_{cs}$ ,  $F_{qs}$  and  $F_{\gamma s}$  are shape factors,  $F_{cd}$ ,  $F_{qd}$  and  $F_{\gamma d}$  are depth factors and  $F_{ci}$ ,  $F_{qi}$  and  $F_{\gamma i}$  are inclination factors.

For the PySimple2 material, the ultimate lateral load capacity is determined as the total passive resisting force acting on the front side of the embedded footing. For homogeneous backfill against the footing, the passive resisting force can be calculated using a linearly varying pressure distribution resulting in the following expression:

$$p_{ult} = 0.5\gamma K_p D_f^2 \quad (2)$$

where  $p_{ult}$  = passive earth pressure per unit length of footing,  $\gamma$  = unit weight of soil,  $D_f$  = depth of embedment,  $K_p$  = passive earth pressure coefficient are calculated using Coulomb (1776).

For the TzSimple2 material, the lateral capacity is the total sliding resistance. The frictional resistance is determined using the general equation for shear strength of footing soil interface after considering a reasonable base friction angle between soil and the footing base. The equation used to calculate the sliding capacity of a foundation resting on cohesionless soil is calculated as follows:

$$t_{ult} = W_g \tan \delta + A_b c \quad (3)$$

where  $t_{ult}$  = frictional resistance per unit area of foundation,  $W_g$  = weight on the foundation from the structure,  $\delta$  = angle of friction between foundation and soil, which typically varies from  $1/3\phi$  to  $2/3\phi$ ,  $A_b$  = the area of the base of footing in contact with the soil (=L x B).

The vertical and lateral stiffness are calculated using expressions given by

Gazetas (1991) (also suggested in ATC-40 (1996)). The rotational stiffness of the foundation is accounted implicitly for the differential movement of the vertical springs. Table 1 shows the equations to calculate stiffness of footing in different modes such as vertical, lateral and rotational for both surface and embedded footing.

Alternatively, the command CapSoil can be involved, signifying that the user can input strength and stiffness directly. The command requires following input in the given sequence:

*CapSoil* \$Q\_{ult}\$ \$P\_{ult}\$ \$T\_{ult}\$ \$K\_v\$ \$K\_h\$

where,  $Q_{ult}$  = Vertical bearing capacity,  $P_{ult}$  = Passive resistance capacity,  $T_{ult}$  = Sliding resistance capacity,  $K_v$  = Vertical stiffness of footing and  $K_h$  = Lateral stiffness of footing.

The next line of the input file contains essential footing dimensions and properties given in the following sequence:

*FootProp* \$L\_f\$ \$B\_f\$ \$H\_f\$ \$D\_f\$ \$E\_f\$ \$W\_g\$ \$\beta\$

where  $L_f$  = Length of footing,  $B_f$  = Width of footing,  $H_f$  = Height of footing,  $D_f$  = Depth of embedment,  $E_f$  = Young's modulus for footing material,  $\beta$  = Inclination of the load on the foundation with respect to vertical, (in degree).

The final line of the input file requires some important parameters regarding the mesh generation that are to be specified by the user in the following sequence:

*MeshProp* \$R\_k\$ \$R\_e\$ \$S\_e\$

where,  $R_k$  = Stiffness intensity ratio ( $k_{end}/k_{mid}$ ),  $R_e$  = End length ratio ( $L_{end}/L_f$ ) and  $S_e$  = vertical spring spacing as a fraction of total footing length ( $l_e/L_f$ ).

Note that a variable stiffness distribution over the length of the foundation is used in this model to account for the higher reaction at the ends of the footing. that develops as the footing rocks under lateral loads (e.g., seismically induced inertial loads). ATC-40 (1996) suggests that a higher intensity of springs should be placed at the end region.

The end region  $L_{end}$  is defined as the length of the edge portion having a higher reaction. ATC-40 (1996) suggests this region has a larger stiffness that extends to a length of B/6 from each end of the footing. However, Harden et al. (2005) showed that the end length ratio is a function of the footing aspect ratio (B/L).

The length of each element is also a user-defined parameter. The user has to specify the fraction of the element length with respect to the total footing length.

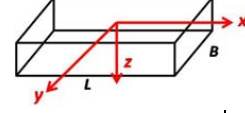
#### **Argument 4: \$FootingCondition**

The user has the option of defining a variety of foundation conditions by providing an integer value from 1 to 5 (Figure 3), where:

- \$FootingCondition = 1: fixed in x, z and  $\theta$  degree of freedom

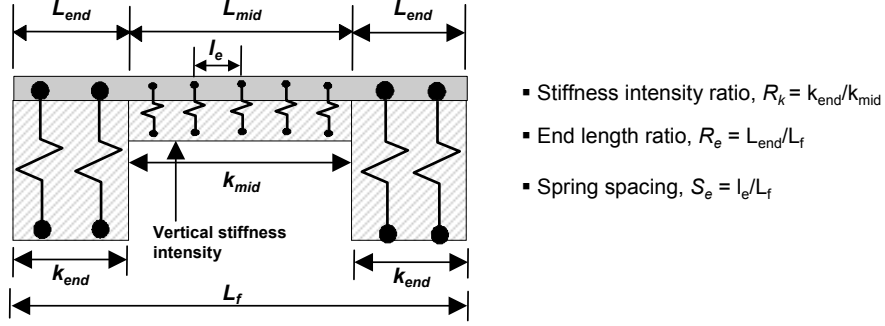
**TABLE 1. Gazetas equations for shallow foundation stiffness after Gazetas (1991) as summarized in ATC-40 (1996).**

Stiffness	Equation
Surface Stiffness	
Vertical Translation	$K_z' = \frac{GL}{1-\nu} \left[ 0.73 + 1.54 \left( \frac{B}{L} \right)^{0.75} \right]$
Horizontal Translation (toward long side)	$K_y' = \frac{GL}{2-\nu} \left[ 2 + 2.5 \left( \frac{B}{L} \right)^{0.85} \right]$
Horizontal Translation (toward short side)	$K_x' = \frac{GL}{2-\nu} \left[ 2 + 2.5 \left( \frac{B}{L} \right)^{0.85} \right] + \frac{GL}{0.75-\nu} \left[ 0.1 \left( 1 - \frac{B}{L} \right) \right]$
Rotation about x-axis	$K_{\alpha_x}' = \frac{G}{1-\nu} I_x^{0.75} \left( \frac{L}{B} \right)^{0.25} \left( 2.4 + 0.5 \frac{B}{L} \right)$
Rotation about y-axis	$K_{\alpha_y}' = \frac{G}{1-\nu} I_y^{0.75} \left[ 3 \left( \frac{L}{B} \right)^{0.15} \right]$
Stiffness Embedment Factors	
Embedment Factor, Vertical Translation	$e_z = \left[ 1 + 0.095 \frac{D_f}{B} \left( 1 + 1.3 \frac{B}{L} \right) \right] \left[ 1 + 0.2 \left( \frac{2L + 2B}{LB} H \right)^{0.67} \right]$
Embedment Factor, Horizontal Translation (toward long side)	$e_y = \left[ 1 + 0.15 \left( \frac{2D_f}{B} \right)^{0.5} \right] \left\{ 1 + 0.52 \left[ \frac{\left( D_f - \frac{H}{2} \right) 16(L+B)H}{BL^2} \right]^{0.4} \right\}$
Embedment Factor, Horizontal Translation (toward short side)	$e_x = \left[ 1 + 0.15 \left( \frac{2D_f}{L} \right)^{0.5} \right] \left\{ 1 + 0.52 \left[ \frac{\left( D_f - \frac{H}{2} \right) 16(L+B)H}{LB^2} \right]^{0.4} \right\}$
Embedment Factor, Rotation about x axis	$e_{\alpha_x} = 1 + 2.52 \frac{H}{B} \left( 1 + \frac{2H}{B} \left( \frac{d}{D_f} \right)^{-0.2} \left( \frac{B}{L} \right)^{0.5} \right)$
Embedment Factor, Rotation about y axis	$e_{\alpha_y} = 1 + 0.92 \left( \frac{2H}{L} \right)^{0.60} \left( 1.5 + \left( \frac{2H}{L} \right)^{1.9} \left( \frac{H}{D_f} \right)^{-0.60} \right)$



Where,  $K_i$  = Uncoupled Total Surface Stiffness for a rigid plate on a semi-infinite homogeneous elastic half-space,  $e_i$  = Stiffness Embedment Factor for a rigid plate on a semi-infinite homogeneous elastic half-space,  $L$  = foundation length,  $B$  = foundation width,  $D_f$  = depth of embedment,  $H$  = foundation thickness,  $G$  = Shearing Modulus,  $\nu$  = Poisson's ratio.

- \$FootingCondition = 2: linear elastic in  $z$  and  $\theta$ , fixed in  $x$  degree of freedom
- \$FootingCondition = 3: linear elastic in  $x$ ,  $z$  and  $\theta$  degree of freedom



**FIG. 2. BNWF Mesh discretization with variable stiffness intensity**

- \$FootingCondition = 4: nonlinear, inelastic in  $z$  and  $\theta$  degree of freedom, fixed in  $x$  degree of freedom
- \$FootingCondition = 5: nonlinear, inelastic in  $x$ ,  $z$  and  $\theta$  degree of freedom

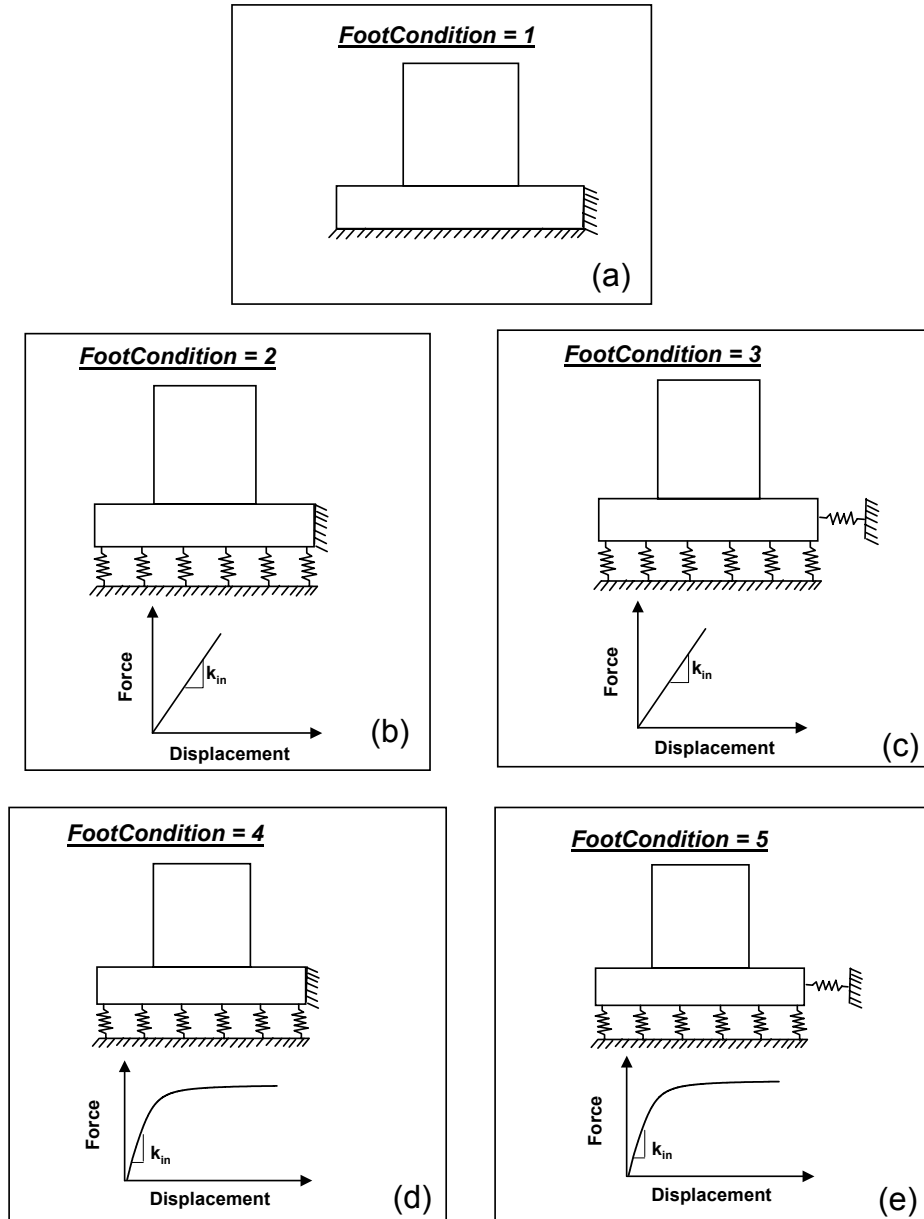
Condition 1 means the foundation will be rigid, and therefore no displacement at the base. Condition 2 will generate a mesh of vertical linear, elastic springs interconnected with linear elastic beam-column elements to represent the footing. However, the footing will be restrained from lateral movement. Condition 3 is similar to condition 2, with the addition of linear elastic response in the lateral direction. Condition 4 will use a modified version of the nonlinear springs (QzSimple2 material) introduced by Boulanger (2000) and modified by Raychowdhury and Hutchinson (2008) in the vertical direction to account for axial and rocking movement of the footing. The footing will be restrained against sliding in the case. The fifth condition (Condition 5) generates a mesh of nonlinear inelastic springs in both the vertical and horizontal directions as described in Figure 1, where nonlinear inelastic spring response is considered, the nonlinear inelastic spring models QzSimple2, PySimple2 and TzSimple2, which are modified forms of those used for piles, are used.

## IMPORTANT IMPLEMENTATION DETAILS

To make balance between simplicity and numbering needed, a systematic node, element and material type numbering scheme is used in the *ShallowFoundationGen* command. The details of the numbering scheme are as follows:

### Node Numbering:

If the foundation tag is 1, then the footing node numbers will start from 1001 and the connecting spring node numbers will start from 100001. Similarly, for the second foundation (foundation tag 2), the footing node numbers will start from 2001 and connecting spring node numbers will start from 200001. Note that by following this numbering scheme, it has been assumed that the last node tag used in the structure is less than 1001 and the maximum number of nodes in a single foundation should not exceed 99000.

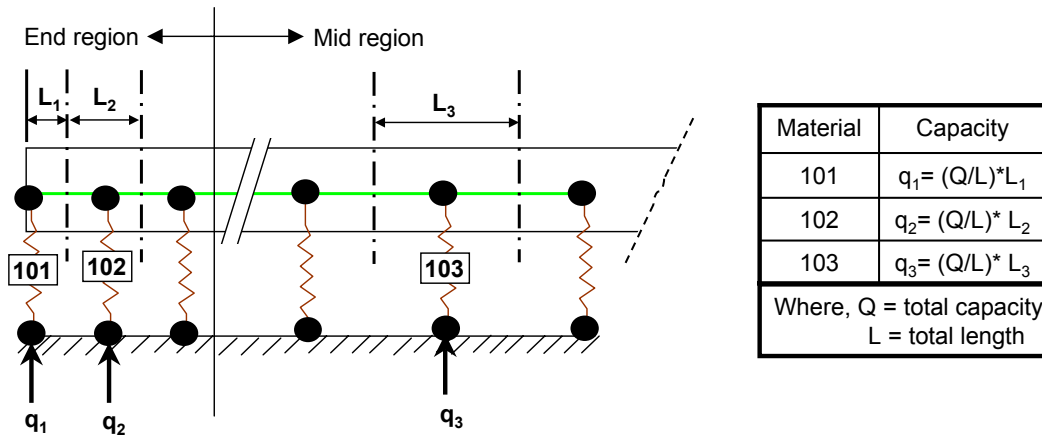


**FIG. 3. Different footing conditions: (a) Fixed base, (b) Elastic base sliding restricted, (c) Elastic base with sliding allowed, (d) Nonlinear base sliding restricted and (e) Nonlinear base with sliding allowed**

#### Material Numbering:

The material tags in the *ShallowFoundationGen* starts from 101, which implies that the material tags used in the structure must be less than (or equal to) 100 in order to avoid confusion with the material tags used for the foundation springs. If there is more than one foundation to be analyzed, then for the foundation with tag 1, the material tags will start from 101, while the second foundation (foundation

tag 2), the material tag will automatically start from 201. The material tags for different footing conditions are specified in the following fashion. For the fixed base case (FootingCondition = 1), no additional elements will be created, and the foundation will be fixed against vertical, horizontal and rotational movements. For FootingCondition = 2, three types of foundation materials will be created, namely material tags for the two extreme end springs, other end zone springs and the mid zone springs defined as 101, 102 and 103, respectively (Figure 4). Creating the three types of vertical springs allows the generator to account for variation in spring spacing (a corresponding tributary area variation in within the different regions). In addition, the extreme end springs will have one-half of the tributary area of other end springs and thus half the stiffness and capacity.



**FIG. 4. Tributary capacity calculation**

For the FootingCondition = 3, in addition to the vertical elastic springs, one horizontal elastic spring is introduced (with material tag =104). The material tag assignment for the FootingCondition = 4 case is similar to FootingCondition = 2 case, except the springs assigned in this case are nonlinear QzSimple2 springs. FootingCondition = 5 case is similar to FootingCondition = 4 case, with the addition of two nonlinear horizontal springs provided at the ends of the footing: one is a TzSimple2 spring to account for sliding behavior and the another one PySimple2 spring to account for passive pressure for embedded footings (Figure 1). The material tag for PySimple2 is assigned as 105 and that for TzSimple2 is assigned as 106. Table 1 summarizes the material tag assignments for the various footing conditions.

### Element Numbering:

The elements are numbered following a method similar to the node numbering scheme (Figure 5). The footing elements (elastic beam-column elements) are numbered starting with 1001 for a foundation with FoundationTag = 1. The numbering is assigned from left to right. The zerolength springs are numbered starting from 100001 from left to right for a foundation with FoundationTag = 1.



**TABLE 2. Material Tag assignment for the various footing conditions**

Material Tag	Spring Description	\$FootCondition applicable
101	Vertical springs (Elastic/ QzSimple2) at two extreme ends	2, 3, 4, 5
102	Vertical springs (Elastic/ QzSimple2) at end region	2, 3, 4, 5
103	Vertical springs (Elastic/ QzSimple2) at mid region	2, 3, 4, 5
104	Lateral spring (Elastic)	3
105	Lateral spring (PySimple2)	5
106	Lateral spring (TzSimple2)	5

**Geometric Transfer Tag:**

To avoid the confusion between the transfer tags provided in the structure with those provided in the foundation, the geometric transfer tags in the foundation begins with 10, assuming that the maximum number of transfer tags given in the structure will not exceed 9. The geometric transfer tag of a foundation with FoundationTag = 1 will be assigned a geometric transfer tag = 10 automatically. Similarly foundations with FoundationTag = 2 and 3 will be assigned the geometric transfer tag = 20 and 30, respectively. Herein, the geometric transfer tag is assumed to be a linear transfer tag.

**Some Useful Variables:**

The command generates some useful variables that are usually required for evaluating foundation response. These are as follows:

1. endFootNodeL\_\$FoundationTag: extreme left end node for the foundation associated with any FoundationTag
2. endFootNodeR\_\$FoundationTag: extreme right end node for the foundation associated with any FoundationTag
3. endSprEleL\_\$FoundationTag: extreme left end vertical spring element for the foundation associated with any FoundationTag
4. endSprEleR\_\$FoundationTag: extreme right end vertical spring element for the foundation associated with any FoundationTag
5. midSprEle\_\$FoundationTag: middle vertical spring element for the foundation associated with any FoundationTag

## EXAMPLE PROBLEMS

Two example problems are described to illustrate the use of the *ShallowFoundationGen* command. One is a shearwall building supported by a single strip footing. The second example is of a single-story single-bay frame, the columns of which are supported by two square footings. The following subsections discuss the example problems showing the input files, OpenSees generated output files, and analysis results.

### Example 1: Shearwall Supported by a Strip Footing

Figure 5 schematically shows an example single shearwall resting on a strip footing. The footing condition = 5 case is used in this example. Element numbers are shown in circles, while node numbers are not. The foundation tag given here is 1, so the foundation node numbers start from 1001 automatically. The footing element numbers will also be be designated from 1001, while the spring end nodes and spring elements will be numbered from 100001. The footing input parameters are shown in File-1. The master tcl file to describe the structure and call the *ShallowFoundationGen* command is shown in File-2. File-3 shows the tcl file that is generated by the *ShallowFoundationGen* command. The shearwall-footing system is subjected to and analyzed for a simple monotonic lateral push and the results obtained are shown in Figure 6.

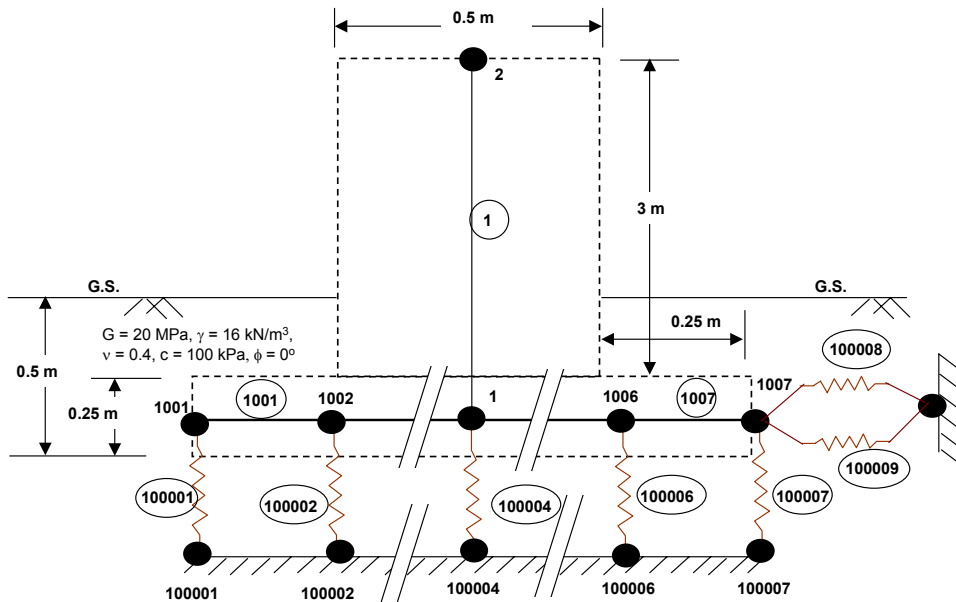
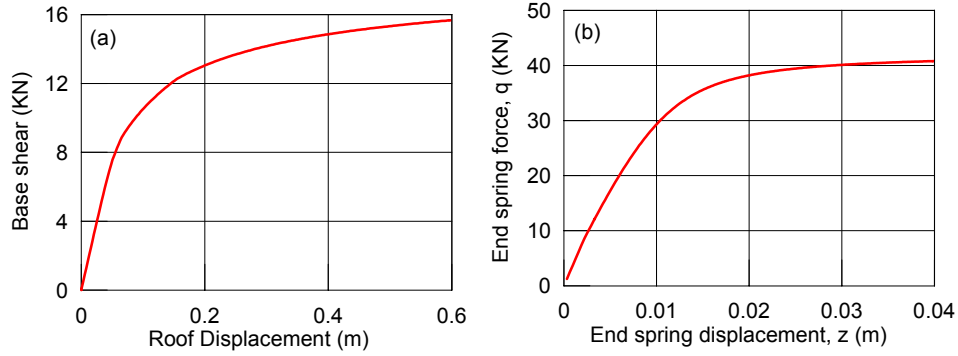


FIG. 5. Example 1: Shearwall resting on a strip footing

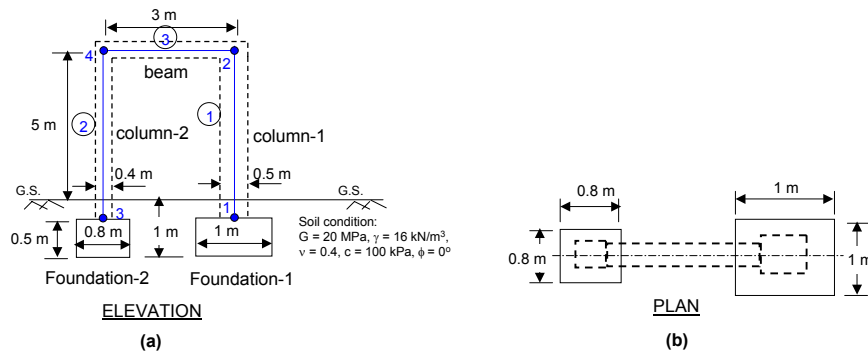
### Example 2: Frame Supported by Two Square Footings

In this example, a multi-foundation system is analyzed. The system consists of a single-story single-bay frame. The two columns of the frame are supported by two isolated footings of different sizes. The foundation tags for the two foundations are 1 and 2 (Figure 7 and Figure 8). Here, the *ShallowFoundationGen*



**FIG. 6. Results for (a) Wall force-displacement response (b) Extreme end vertical spring response (Example 1)**

command is used twice within the main code to create two different output files “*Foundation.1.tcl*” and “*Foundation.2.tcl*” for the two shallow foundations, respectively. In this example, two different footing conditions are used for two footings to demonstrate the different spring responses. Footing-1 is assigned `$FootingCondition = 5`, while Footing-2 is assigned with `$FootingCondition = 4`, i.e. that sliding restrained case assuming that this foundation is attached to a larger foundation system. Figure 9 shows the results of half cycle of push-pull applied to the frame structure.



**FIG. 7. Example 2:(a) Elevation and (b) Plan of the single-bay frame for Example 2**

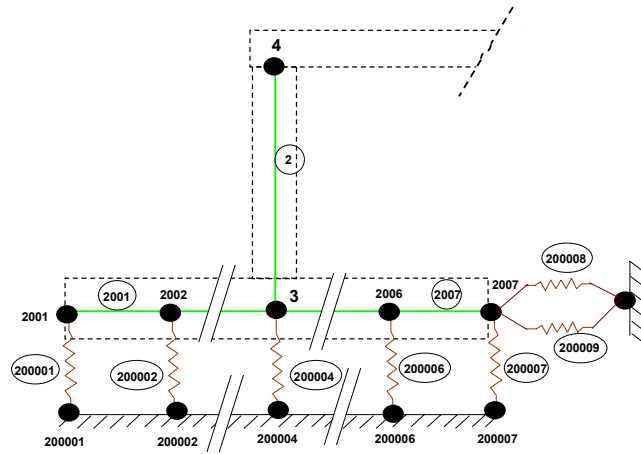


FIG. 8. Node and element configuration for foundation-2 and column-2 of Example 2

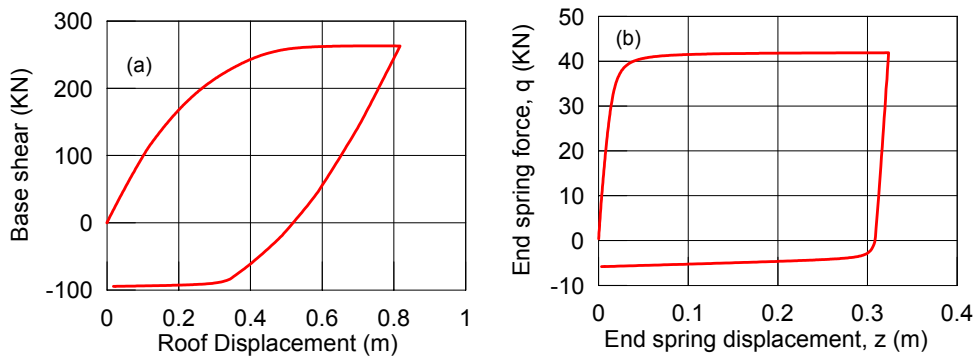


FIG. 9. (a) Force-displacement response (b) Extreme end vertical spring response of Example 2

## REFERENCES

- ATC-40 (1996). *Seismic Evaluation and Retrofit of Concrete Buildings*. Applied Technology Council (ATC).
- Boulangier, R. W. (2000). “The pysimple1, qzsimple1 and tzsimple1 material documentation.” *Documentation for the OpenSees platform available at: <http://opensees.berkeley.edu/>*.
- Coulomb, C. A. (1776). “Essai sur une application des rgles des maximis et minimis quelques problmes de statique relatifs l’architectute.” *Mm. acad. roy. prs. Divers savants, Paris*, 3, 38.
- Gazetas, G. (1991). *Foundation Engineering Handbook*. Fang, H.Y. edit. Van Nostrand Rienhold.
- Harden, C. W., Hutchinson, T., Martin, G. R., and Kutter, B. L. (2005). “Numerical modeling of the nonlinear cyclic response of shallow foundations.” *Report for Pacific Earthquake Engineering Research Center, PEER*, 2005/04.
- Meyerhof, G. G. (1963). “Some recent research on the bearing capacity of foundations.” *Canadian Geotechnical Journal*, 1(1), 16–26.
- Raychowdhury, P. (2008). “Numerical modeling, calibration and performance evaluation of nonlinear winkler-based shallow foundations.” *PhD Dissertation, University of California, San Diego*.
- Raychowdhury, P. and Hutchinson, T. (2008). “Nonlinear material models for winkler-based shallow foundation response evaluation.” *In the proceeding of GeoCongress 2008*.
- Terzaghi, K. (1943). *Theoretical Soil Mechanics*. J. Wiley, New York.

## APPENDIX 1: INPUT AND OUTPUT FILES FOR EXAMPLE 1

### File 1: Input file ("Input1.txt") for soil and footing properties

```
#####  
## INPUT FILE FOR SHALLOWFOUNDATIONGEN  
#Input data for foundation 1  
  
#SoilProp $SoilType $c $Phi $Gamma $G $Nu $Crad $Tp  
SoilProp 1 100000.0 0.0 16000.0 20000000.0 0.4 0.05 0.1  
  
#CapSoil $Qult $Pult $Tult $Kv $Kx  
#CapSoil 1800.0 1000.0 1000.0 2000.00 3000.0  
  
#FootProp $Lf $Bf $Hf $Df $Ef $Wg $beta  
FootProp 1.0 1.0 0.25 0.5 21500000000.0 120000.0 0.0  
  
#MeshProp $Rk $Re $le/L  
MeshProp 5.0 0.2 0.2  
  
#End of input data  
#####
```

```
#####
```

### File 2: Master Tcl file

```
##--Example # 1 - "A shear wall supported by a strip footing"  
##--Written by Prishati Raychowdhury (UCSD)  
##--units: N,m  
wipe  
wipeAnalysis  
  
# Create ModelBuilder  
model BasicBuilder -ndm 2 -ndf 3  
  
# Set wall and footing dimensions  
set LengthWall 0.5;  
set WidthWall 0.2;  
set HeightWall 5.0;  
  
# Set structural nodes  
node 1 0. 0.  
node 2 0. $HeightWall  
  
# set wall properties  
set AWall [expr $WidthWall*$LengthWall]  
set EWall [expr 2.15*pow(10,10)]; #----[N/m^2] concrete  
set IWall [expr $WidthWall*pow($LengthWall,3)/12.]
```

```

# set wall material
uniaxialMaterial Elastic 1 $EWall

# set geometric transformation
geomTransf Linear 1

# set wall element
#element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $transfTag
element elasticBeamColumn 1 1 2 $AWall $EWall $IWall 1

# set wall mass
set MWall 1200.0; #---mass of structure (kg)
mass 2 $MWall $MWall 1
#
#
#-----
# Use ShallowFoundationGen command to
# attach shallow foundation with Foundation tag=1
# at node 1
#-----
#
#
set FoundationTag 1
#ShallowFoundationGen $FoundationTag $ConectNode $InputFileName $FootCondition
ShallowFoundationGen $FoundationTag 1 "Input1.txt" 5
source Foundation $FoundationTag.tcl

set MassFooting 1200.0
mass 1 $MassFooting $MassFooting 1
#-----
# Eigen Value Analysis
#-----
set PI 3.1415926
set lambda [eigen 1]
set lambda [lindex $lambda 0]
set omega [expr pow($lambda,0.5)]
set Tn [expr 2*$PI/$omega]
set fn [expr 1/$Tn]
puts "1st mode, Tn=$Tn sec, fn=$fn Hz"
#-----
# Recorder
#-----
###-wall
recorder Node -time -file WallRoofdisp.dat -node 2 -dof 1 disp
recorder Element -file WallElementforce.dat -time -ele 1 localForce
###-Spring
recorder Node -time -file EndSprL.disp.dat -node $endFootNodeL_1 -dof 1 2 3 disp
recorder Element -file EndSpringL.force.dat -time -ele $endSprEleL_1 force
#-----
# Gravity LOAD PATTERNS
#-----
set gacc 9.87
set FSv 5.0
set deadLoad [expr ($MassFooting+$MWall)*$gacc*$FSv]; #---total gravity load on footing (footing+wall)
#puts $deadLoad
pattern Plain 1 "Linear" {

```

```

    load 2 0. [expr -$deadLoad] 0.
}
#-----
# gravity analysis
#-----
system UmfPack
constraints Plain
test NormDispIncr 1.0e-8 40 0
algorithm Newton
numberer RCM
integrator LoadControl 0.1
analysis Static
analyze 10; #use 10 analysis steps
#-----
# Pushover analysis
#-----
#loadConst
loadConst -time 0.0
#-----
# Start of Load Control
#-----
pattern Plain 2 "Linear" {
    load 2 -10.0 0.0 0.
}
analyze 100
##-----
## End of Load Control
##-----

#-----
# Start of Displacement Control
#-----
#Set some parameters
set dU -0.005; # Displacement increment
##integrator DisplacementControl node dof init Jd min max
integrator DisplacementControl 2 1 $dU 3 [expr $dU/2.0] $dU
analyze 300
#-----
# End of Displacement Control
#-----

```

**File 3: Output tcl file**

```

#####
# This is an intermediate file generated by the command ShallowFoundationGen. #
# Source it after the ShallowFoundationGen command. #
# Use this file to check shallow foundation nodes, elements, fixity details #
# ShallowFoundationGen.cpp is developed by Prishati Raychowdhury (UCSD) #
#####

# Foundation Tag =1
# Foundation Base Condition Tag =5

```



```

#node $NodeTag $Xcoord $Ycoord
node 1001 -0.5 0
node 100001 -0.5 0
node 1002 -0.4 0
node 100002 -0.4 0
node 1003 -0.3 0
node 100003 -0.3 0
node 1004 0 0
node 100004 0 0
node 1005 0.3 0
node 100005 0.3 0
node 1006 0.4 0
node 100006 0.4 0
node 1007 0.5 0
node 100007 0.5 0
node 100008 0.5 0
node 100009 0.5 0

#equalDOF $rNodeTag $cNodeTag $dof1 $dof2 $dof3
equalDOF 1 1004 1 2 3

#Materials for shallow foundation

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-end-extreme $z50-end <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 101 1 38020 0.00274347 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-end $z50-end <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 102 1 76040 0.00274347 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-mid $z50-mid <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 103 1 228120 0.0137173 0.1 0.05

#uniaxialMaterial PySimple2 $matTag $SoilType $Pp $xp50 Cd <CradSoil>
uniaxialMaterial PySimple2 105 1 102000 0.0145067 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $SoilType $Tult $xt50 <CradSoil>
uniaxialMaterial TzSimple2 106 1 100000 0.0142222 0.1 0.05

#Vertical spring element connectivity
#element zeroLength $eleTag $iNode $jNode -mat$matTag -dir $dir
element zeroLength 100001 100001 1001 -mat 101 -dir 2
element zeroLength 100002 100002 1002 -mat 102 -dir 2
element zeroLength 100003 100003 1003 -mat 103 -dir 2
element zeroLength 100004 100004 1004 -mat 103 -dir 2
element zeroLength 100005 100005 1005 -mat 103 -dir 2
element zeroLength 100006 100006 1006 -mat 102 -dir 2
element zeroLength 100007 100007 1007 -mat 101 -dir 2

#Horizontal spring element connectivity
#element zeroLength $eleTag $iNode $jNode -mat$matTag -dir $dir
element zeroLength 100008 1007 100008 -mat 105 -dir 1
element zeroLength 100009 1007 100009 -mat 106 -dir 1

# geomTransf Linear $stransfTag <-jntOffset $dXi $dYi $dXj $dYj>
geomTransf Linear 10

```

```

#foundation element connectivity
#element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $stransfTag
element elasticBeamColumn 1001 1001 1002 0.25 2.15e+010 0.00130208 10
element elasticBeamColumn 1002 1002 1003 0.25 2.15e+010 0.00130208 10
element elasticBeamColumn 1003 1003 1004 0.25 2.15e+010 0.00130208 10
element elasticBeamColumn 1004 1004 1005 0.25 2.15e+010 0.00130208 10
element elasticBeamColumn 1005 1005 1006 0.25 2.15e+010 0.00130208 10
element elasticBeamColumn 1006 1006 1007 0.25 2.15e+010 0.00130208 10

#fixity
fix 100001 1 1 1
fix 100002 1 1 1
fix 100003 1 1 1
fix 100004 1 1 1
fix 100005 1 1 1
fix 100006 1 1 1
fix 100007 1 1 1
fix 100008 1 1 1
fix 100009 1 1 1

set endFootNodeL_1 1001
set endFootNodeR_1 1007
set endSprEleL_1 100001
set endSprEleR_1 100007
set midSprEle_1 100004

#####

```

## APPENDIX 2: INPUT AND OUTPUT FILES FOR EXAMPLE 2

### File 1: Input file "Input1.txt" for Foundation-1

```
#####  
#Input data for foundation 1  
#SoilProp $SoilType $c $Phi $Gamma $G $Nu $Crad $Tp  
SoilProp 1 100000.0 0.0 16000.0 30000000.0 0.4 0.05 0.1  
  
#CapSoil $Qult $Pult $Tult $Kv $Kx  
#CapSoil 1800.0 1000.0 1000.0 2000.00 3000.0  
  
#FootProp $Lf $Bf $Hf $Df $Ef $Wg $beta  
FootProp 1.0 1.0 0.5 1.0 21500000000.0 120000.0 0.0  
  
#MeshProp $Rk $Re $le/L  
MeshProp 2.0 0.2 0.2  
  
#End of input data  
#####
```

### File 2: Input file "Input2.txt" for Foundation-2

```
#####  
#Input data for foundation 1  
  
#SoilProp $SoilType $c $Phi $Gamma $G $Nu $Crad $Tp  
SoilProp 1 100000.0 0.0 16000.0 30000000.0 0.4 0.05 0.1  
  
#CapSoil $Qult $Pult $Tult $Kv $Kx  
#CapSoil 1800.0 1000.0 1000.0 2000.00 3000.0  
  
#FootProp $Lf $Bf $Hf $Df $Ef $Wg $beta  
FootProp 0.8 0.8 0.4 1.0 21500000000.0 120000.0 0.0  
  
#MeshProp $Rk $Re $le/L  
MeshProp 2.0 0.2 0.2  
  
#End of input data  
#####
```

**File 3: Master Tcl file: MultiFoundation.tcl**

```
#####
#--Example # 1 - "A single bay single story Frame supported by two square footings"
#--Written by Prishati Raychowdhury (UCSD)
#--units: N,m
#####
wipe
wipeAnalysis

# Create ModelBuilder
model BasicBuilder -ndm 2 -ndf 3

# Set column dimensions
set LengthCol1 0.5; #Column 1
set WidthCol1 0.5;

set LengthCol2 0.4; #Column 2
set WidthCol2 0.4;

set HeightCol 5.0; #Height of each column
set ColSpacing 3.0; #@c/c column spacing

set HeightBeam 0.4; #Beam
set WidthBeam 0.4;

# Set structural nodes
#node Tag X Y
node 1 0 0.
node 2 0. $HeightCol
node 3 $ColSpacing 0.
node 4 $ColSpacing $HeightCol

# set wall properties
set ACol1 [expr $WidthCol1*$LengthCol1]
set ICol1 [expr $WidthCol1*pow($LengthCol1,3)/12.]
set ACol2 [expr $WidthCol2*$LengthCol2]
set ICol2 [expr $WidthCol2*pow($LengthCol2,3)/12.]
set ABeam [expr $WidthBeam*$HeightBeam]
set IBeam [expr $WidthBeam*pow($HeightBeam,3)/12.]
set E [expr 2.15*pow(10,10)]; #----[N/m^2] concrete

# set wall material
uniaxialMaterial Elastic 1 $E

# set geometric transformation
geomTransf Linear 1
```

```

geomTransf Linear 2

# set column and beam elements
#element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $tranfTag
element elasticBeamColumn 1 1 2 $ACol1 $E $ICol1 1
element elasticBeamColumn 2 3 4 $ACol2 $E $ICol2 1
element elasticBeamColumn 3 2 4 $ABeam $E $IBeam 2
# set wall mass
set Mass 1200.0; #---nodal mass of structure (kg)
mass 2 $Mass $Mass 1.0
mass 4 $Mass $Mass 1.0
#
#
#-----
# Use ShallowFoundationGen command to
# attach shallow foundation with Foundation tag=1
# connected at node 1
#-----
#
set FoundationTag 1
#ShallowFoundationGen $FoundationTag $ConectNode $inpufile $FootCondition
ShallowFoundationGen $FoundationTag 1 "Input1.txt" 5
source Foundation_ $FoundationTag.tcl
#
#
#-----
# Use ShallowFoundationGen command to
# attach shallow foundation with Foundation tag=2
# connected at node 3
#-----
#
set FoundationTag 2
#ShallowFoundationGen $FoundationTag $ConectNode "Input.dat" $FootCondition
ShallowFoundationGen $FoundationTag 3 "Input2.txt" 4
source Foundation_ $FoundationTag.tcl

#-----
# Eigen Value Analysis
#-----
set PI 3.1415926
set lambdax [eigen 1]
set lambda [lindex $lambdax 0]
set omega [expr pow($lambda,0.5)]
set Tn [expr 2*$PI/$omega]
set fn [expr 1/$Tn]
puts "1st mode, Tn=$Tn sec, fn=$fn Hz"

```

```

#-----
# Recorder
#-----
# Column
recorder Node -time -file Coldisp1.dat -node 2 -dof 1 disp
recorder Element -file Colforce1.dat -time -ele 1 localForce
#Foundation 1: End Spring
recorder Node -time -file EndSprLdisp1.dat -node $endFootNodeL_1 -dof 1 2 3 disp
recorder Element -file EndSpringLforce1.dat -time -ele $endSprEleL_1 force
#Foundation 2: End Spring
recorder Node -time -file EndSprLdisp2.dat -node $endFootNodeL_2 -dof 1 2 3 disp
recorder Element -file EndSpringLforce2.dat -time -ele $endSprEleL_2 force

#-----
# Gravity LOAD PATTERNS
#-----
set gacc 9.87
set FSv 5.0
set deadLoad [expr $Mass*$gacc*$FSv]; #---total gravity load on footing (footing+wall)
#puts $deadLoad
pattern Plain 1 "Linear" {
    load 2 0. [expr -$deadLoad] 0.
    load 4 0. [expr -$deadLoad] 0.
}

#-----
# gravity analysis
#-----
system UmfPack
constraints Plain
test NormDispIncr 1.0e-8 40 0
algorithm Newton
numberer RCM
integrator LoadControl 0.1
analysis Static
analyze 10; #use 10 analysis steps

#-----
# Pushover analysis
#-----
#loadConst
loadConst -time 0.0
#-----
#puts $deadLoad
pattern Plain 2 "Linear" {
    load 2 -10.0 0.0 0.
}
analyze 100

```

```

# -----
# Start of Displacement Control
# -----
#Set some parameters
set dU -0.005;      # Displacement increment
#integrator DisplacementControl node dof init Jd min max
integrator DisplacementControl 2 1 $dU 3 [expr $dU/2.0] $dU
analyze 300

set dU 0.005;      # Displacement increment
##integrator DisplacementControl node dof init Jd min max
integrator DisplacementControl 2 1 $dU 3 [expr $dU/10.0] $dU
#analyze 160
# End of the analysis

#####

```

**File 4: Output Tcl file: "Foundation\_1.tcl"**

```

#####
#
# This is an intermediate file generated by the command ShallowFoundationGen.
# Source it after the ShallowFoundationGen command.
# Use this file to check shallow foundation nodes, elements, fixity details
# ShallowFoundationGen.cpp is developed by Prishati Raychowdhury (UCSD)
#
#####

# Foundation Tag =1
# Foundation Base Condition Tag =5

#node $NodeTag $Xcoord $Ycoord
node 1001 -0.5 0
node 100001 -0.5 0
node 1002 -0.4 0
node 100002 -0.4 0
node 1003 -0.3 0
node 100003 -0.3 0
node 1004 0 0
node 100004 0 0
node 1005 0.3 0
node 100005 0.3 0
node 1006 0.4 0
node 100006 0.4 0
node 1007 0.5 0
node 100007 0.5 0
node 100008 0.5 0
node 100009 0.5 0

```

```

#equalDOF $rNodeTag $cNodeTag $dof1 $dof2 $dof3
equalDOF 1 1004 1 2 3

#Materials for shallow foundation

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-end-extreme $z50-end <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 101 1 41840 0.00270946 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-end $z50-end <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 102 1 83680 0.00270946 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-mid $z50-mid <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 103 1 251040 0.00541893 0.1 0.05

#uniaxialMaterial PySimple2 $matTag $SoilType $Pp $xp50 Cd <CradSoil>
uniaxialMaterial PySimple2 105 1 208000 0.0197215 0.1 0.05

#uniaxialMaterial TzSimple2 $matTag $SoilType $Tult $xt50 <CradSoil>
uniaxialMaterial TzSimple2 106 1 100000 0.00948148 0.1 0.05

#Vertical spring element connectivity
#element zeroLength $eleTag $iNode $jNode -mat$matTag -dir $dir
element zeroLength 100001 100001 1001 -mat 101 -dir 2
element zeroLength 100002 100002 1002 -mat 102 -dir 2
element zeroLength 100003 100003 1003 -mat 103 -dir 2
element zeroLength 100004 100004 1004 -mat 103 -dir 2
element zeroLength 100005 100005 1005 -mat 103 -dir 2
element zeroLength 100006 100006 1006 -mat 102 -dir 2
element zeroLength 100007 100007 1007 -mat 101 -dir 2

#Horizontal spring element connectivity
#element zeroLength $eleTag $iNode $jNode -mat$matTag -dir $dir
element zeroLength 100008 1007 100008 -mat 105 -dir 1
element zeroLength 100009 1007 100009 -mat 106 -dir 1

# geomTransf Linear $transfTag <-jntOffset $dXi $dYi $dXj $dYj>
geomTransf Linear 10

#foundation element connectivity
#element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $transfTag
element elasticBeamColumn 1001 1001 1002 0.5 2.15e+010 0.0104167 10
element elasticBeamColumn 1002 1002 1003 0.5 2.15e+010 0.0104167 10
element elasticBeamColumn 1003 1003 1004 0.5 2.15e+010 0.0104167 10
element elasticBeamColumn 1004 1004 1005 0.5 2.15e+010 0.0104167 10
element elasticBeamColumn 1005 1005 1006 0.5 2.15e+010 0.0104167 10
element elasticBeamColumn 1006 1006 1007 0.5 2.15e+010 0.0104167 10

#fixity
fix 100001 1 1 1
fix 100002 1 1 1
fix 100003 1 1 1

```



```

fix 100004 1 1 1
fix 100005 1 1 1
fix 100006 1 1 1
fix 100007 1 1 1
fix 100008 1 1 1
fix 100009 1 1 1

set endFootNodeL_1 1001
set endFootNodeR_1 1007
set endSprEleL_1 100001
set endSprEleR_1 100007
set midSprEle_1 100004
#####

```

**File 5: Output Tcl file: "Foundation\_2.tcl"**

```

#####
#
# This is an intermediate file generated by the command ShallowFoundationGen.
# Source it after the ShallowFoundationGen command.
# Use this file to check shallow foundation nodes, elements, fixity details
# ShallowFoundationGen.cpp is developed by Prishati Raychowdhury (UCSD)
#####

# Foundation Tag =2
# Foundation Base Condition Tag =4

#node $NodeTag $Xcoord $Ycoord
node 2001 -0.4 0
node 200001 -0.4 0
node 2002 -0.32 0
node 200002 -0.32 0
node 2003 -0.24 0
node 200003 -0.24 0
node 2004 0 0
node 200004 0 0
node 2005 0.24 0
node 200005 0.24 0
node 2006 0.32 0
node 200006 0.32 0
node 2007 0.4 0
node 200007 0.4 0

#equalDOF $rNodeTag $cNodeTag $dof1 $dof2 $dof3
equalDOF 3 2004 1 2 3

#Materials for shallow foundation

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-end-extreme $z50-end <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 201 1 27872 0.00225616 0.1 0.05

```

```

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-end $z50-end <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 202 1 55744 0.00225616 0.1 0.05

#uniaxialMaterial QzSimple2 $matTag $SoilType $Qult-mid $z50-mid <TpSoil> <CradSoil>
uniaxialMaterial QzSimple2 203 1 167232 0.00451232 0.1 0.05

#fix $midNode 1 0 0
fix 3 1 0 0

#Vertical spring element connectivity
#element zeroLength $eleTag $iNode $jNode -mat$matTag -dir $dir
element zeroLength 200001 200001 2001 -mat 201 -dir 2
element zeroLength 200002 200002 2002 -mat 202 -dir 2
element zeroLength 200003 200003 2003 -mat 203 -dir 2
element zeroLength 200004 200004 2004 -mat 203 -dir 2
element zeroLength 200005 200005 2005 -mat 203 -dir 2
element zeroLength 200006 200006 2006 -mat 202 -dir 2
element zeroLength 200007 200007 2007 -mat 201 -dir 2

# geomTransf Linear $stransfTag <-jntOffset $dXi $dYi $dXj $dYj>
geomTransf Linear 20

#foundation element connectivity
#element elasticBeamColumn $eleTag $iNode $jNode $A $E $Iz $stransfTag
element elasticBeamColumn 2001 2001 2002 0.32 2.15e+010 0.00426667 20
element elasticBeamColumn 2002 2002 2003 0.32 2.15e+010 0.00426667 20
element elasticBeamColumn 2003 2003 2004 0.32 2.15e+010 0.00426667 20
element elasticBeamColumn 2004 2004 2005 0.32 2.15e+010 0.00426667 20
element elasticBeamColumn 2005 2005 2006 0.32 2.15e+010 0.00426667 20
element elasticBeamColumn 2006 2006 2007 0.32 2.15e+010 0.00426667 20

#fixity
fix 200001 1 1 1
fix 200002 1 1 1
fix 200003 1 1 1
fix 200004 1 1 1
fix 200005 1 1 1
fix 200006 1 1 1
fix 200007 1 1 1

set endFootNodeL_2 2001
set endFootNodeR_2 2007
set endSprEleL_2 200001
set endSprEleR_2 200007
set midSprEle_2 200004
#####

```