11 Analysis of a Concrete Diaphragm Wall

11.1 Problem Statement

A concrete diaphragm wall is a continuous concrete wall built into the ground from the ground surface. The wall may consist of precast or cast-in-place concrete panels, or contiguous bored concrete piles. The most common type of wall is a tremie-concrete diaphragm wall cast within a slurry stabilized trench. Trenches are 0.6 to 1.0 meter wide and are excavated in 3 to 6 meter lengths. After the individual panels are excavated, end-stops and reinforcing are installed. Concrete is placed and the end-stops are removed. Once the concrete has set, the neighboring panel can be excavated. After all panels have been constructed, the major excavation can begin, with internal bracing installed as the excavation progresses. In saturated ground, dewatering must also be performed.

The various stages of construction of a diaphragm wall can be simulated with FLAC. In this example, we model the stages following the construction of the panels. Dewatering, excavation and installation of struts are simulated in five excavation stages.

A cross section through the diaphragm wall, given in Figure 11.1, shows the subsurface conditions, the depth at each of the five excavation stages, and the location of the support struts. The properties of the soil layers are summarized in Table 11.1. A porosity of 0.3 is assumed for all soil layers.

A thick aquifer of dense gravel is located below a depth of 34 meters. The pore pressure is assumed to be a constant value of 317 kPa at the bottom of the diaphragm wall. The wall is 1 meter thick and is assumed to behave as an elastic material with a Young’s modulus of 19.2 GPa and Poisson’s ratio of 0.2. The struts are spaced at 2 meter intervals and are pre-loaded. The excavation and dewatering depths are listed in Table 11.2. The locations and properties of the struts are summarized in Table 11.3. The Young’s modulus for the struts is 200 GPa.

Of interest in this analysis are the distribution of shear force and bending moment in the wall, the axial force in the struts, and the displacements of the soil behind the wall.
### Example Applications

**Figure 11.1 Cross section through diaphragm wall**

**Table 11.1 Soil properties**

<table>
<thead>
<tr>
<th>Soil</th>
<th>Dry Density (kg/m³)</th>
<th>Bulk Modulus (MPa)</th>
<th>Shear Modulus (MPa)</th>
<th>Friction Angle (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soil 1</td>
<td>1880</td>
<td>16.67</td>
<td>10.17</td>
<td>30</td>
</tr>
<tr>
<td>Soil 2</td>
<td>1980</td>
<td>30.30</td>
<td>25.60</td>
<td>32</td>
</tr>
<tr>
<td>Soil 3</td>
<td>1910</td>
<td>33.33</td>
<td>20.25</td>
<td>30</td>
</tr>
<tr>
<td>Soil 4</td>
<td>1970</td>
<td>41.67</td>
<td>32.50</td>
<td>34</td>
</tr>
<tr>
<td>Soil 5</td>
<td>1980</td>
<td>77.78</td>
<td>47.25</td>
<td>32</td>
</tr>
<tr>
<td>Soil 6</td>
<td>1700</td>
<td>78.43</td>
<td>53.20</td>
<td>38</td>
</tr>
</tbody>
</table>

**Table 11.2 Excavation and dewatering depths**

<table>
<thead>
<tr>
<th>Excavation Depth (m)</th>
<th>Dewatering Depth (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>5.0</td>
</tr>
<tr>
<td>8.0</td>
<td>9.0</td>
</tr>
<tr>
<td>12.0</td>
<td>13.0</td>
</tr>
<tr>
<td>15.0</td>
<td>16.0</td>
</tr>
<tr>
<td>18.0</td>
<td>19.0</td>
</tr>
</tbody>
</table>
### Table 11.3 Strut properties

<table>
<thead>
<tr>
<th>Strut Number</th>
<th>Depth (m)</th>
<th>Area (\text{m}^2)</th>
<th>Moment of Inertia (\text{m}^4)</th>
<th>Pre-Load (kN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>(173.9 \times 10^{-4})</td>
<td>(4.03 \times 10^{-4})</td>
<td>200.0</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>(218.7 \times 10^{-4})</td>
<td>(6.60 \times 10^{-4})</td>
<td>400.0</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>(218.7 \times 10^{-4})</td>
<td>(6.60 \times 10^{-4})</td>
<td>450.0</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>(218.7 \times 10^{-4})</td>
<td>(6.60 \times 10^{-4})</td>
<td>400.0</td>
</tr>
</tbody>
</table>

### 11.2 Modeling Procedure

The numerical analysis for this problem provides a solution at each of the five excavation depths. At each depth, the modeling sequence includes dewatering, excavation and strut installation steps. A vertical line of symmetry is taken through the center of the excavation, and only one wall is modeled. The data files for this problem are listed in Sections 11.5 through 11.8.

The groundwater flow option, adjust total stress option and structural elements are activated from the Model Options dialog for this problem, as shown in Figure 11.2. The automatic adjustment of total stresses is selected to facilitate the simulation of dewatering.

![Figure 11.2 Settings in Model Options dialog for concrete diaphragm wall example](FLAC Version 6.0)
The grid for this analysis is created using the Build/Generate/Block tool. The model dimensions selected for the grid are assigned in the dialog shown in Figure 11.3. The grid is shown in Figure 11.4. The right boundary of the grid is the line of symmetry for the excavation. The left boundary is located approximately 15 times the excavation width away from the excavation, in order to minimize boundary effects.

Figure 11.3 Build/Generate/Block dialog

The diaphragm wall is also in place in the model before the initial equilibrium calculation is performed, as shown in Figure 11.4. The wall is modeled by 28 beam elements. The upper 18 elements are each 1 meter in height and are positioned so that beam nodes coincide with the depths of the struts. (See Figure 11.5.) The structural nodes with numbers 27, 23, 19 and 16 will be
attached to the struts during the excavation stages. The beam is connected to the soil grid via interface elements attached on both sides of the beam elements. Note that structural element logic is a plane-stress formulation, so the value specified for the Young’s modulus, $E$, is divided by $(1 − \nu^2)$ to correspond to the plane-strain model (see Section 1.2.2 in Structural Elements).

![Figure 11.5](image)

Figure 11.5 Structural beam nodes in upper part of wall

The soil/wall interface properties selected for this example are for demonstration purposes: actual values for wall friction and adhesion should be determined from physical testing or from the literature (e.g., see Clayton et al. 1993). The interface nodes are assigned low shear strength (approximately $2/3$ of the soil friction angle) to simulate a relatively smooth concrete-to-soil interface. (The influence of the interface resistance can be easily investigated by varying the properties of the interface nodes.) Interface stiffness values were selected to approximate the results for the case that the wall is rigidly attached to the grid.

The analysis begins at an initial equilibrium state prior to excavation. The initial stress state is found for the given soil conditions, assuming that the ratio of effective horizontal stress to effective vertical stress is 0.5. A series of FISH functions contained in “ININV.FIS” (see the FISH Library in Section 3 in the FISH volume) are used to establish the initial stress state and pore pressures for the horizontally layered media, with the groundwater table located 2 meters below the ground surface. This analysis is performed as a coupled groundwater flow and mechanical calculation (CONFIG gw) in order for “ININV.FIS” to be applied. However, the groundwater flow is inhibited (SET flow off) and the water bulk modulus is set to zero, so that the specified pore pressures are maintained. The recommended procedure to initialize a stress state in a medium with a phreatic surface is discussed in Section 3.4.8 in the User’s Guide.
Although stresses and pore pressures are initialized in the grid, some stepping is required to bring the model to equilibrium. This is because the additional weight of the beam elements representing the concrete wall produces an imbalance that necessitates some stepping to equilibrate the model again. Note that the grid is pulled down slightly as the wall settles due to gravity. This problem is run in large-strain mode. The model at the initial equilibrium stage is saved to the file “DW_EQUIL.SAV.”

The structural forces in the wall arising from the presence of groundwater will depend on the details of the connection, the fluid boundary conditions, and when the support is installed. Section 1.9.5 in Fluid-Mechanical Interaction contains a discussion about the various possibilities. It is important to recognize the conditions that the model is to represent before setting up model conditions. For this example, we assume the following conditions apply:

1. The structure exists independent of the grid. Forces are transmitted through the interface elements.
2. The wall provides an impermeable barrier to the groundwater, which acts directly on the surface of the wall.
3. The wall is dewatered instantaneously; no movements or transient flow effects are considered. (See Section 1.8.2 in Fluid-Mechanical Interaction for a recommended procedure to determine if this assumption is appropriate.)

The pore pressure distribution in the model is changed at each stage to represent dewatering. The distribution approximates the steady-state flow condition for that excavation stage. (It is not necessary to perform a flow calculation for this analysis, because of the constant pore pressure at the base of the wall imposed by the underlying aquifer.) Note that stresses are also changed automatically by specifying CONFIG ats, because total stresses change if pore pressure is changed by some external method. (See Section 1.9.7 in Fluid-Mechanical Interaction.) The effect of dewatering for the first excavation stage is illustrated in Figure 11.6. This plot shows the settlement within the trench region, as well as the moments in the wall, induced by the dewatering.

The struts are modeled with beam elements. One long element segment represents each strut. The Young’s modulus of the strut is automatically scaled by the 2 meter strut spacing when the spacing property is given. The pre-load applied with the STRUCT node n load command is scaled manually by dividing the actual pre-load value by the 2 meter spacing.

The struts should be installed so that moments cannot develop at their connection with the wall. This is accomplished by slaving the strut node to the wall node using the STRUCTURE node n slave x y m command, in which n corresponds to the strut node and m to the wall node. The strut node is slaved in the x- and y-directions to the wall node, but is free to rotate. In this way, the wall and strut can move without moments developing between the strut and the wall.
A pre-load is applied in the x-direction to the strut node located at the centerline of the excavation. This node is also prevented from rotating or translating in the y-direction. Although the fixity condition in the y-direction is not required for this problem, it may become necessary to minimize any effect of drift for excavations requiring longer struts. The fixity condition does not affect the solution, provided moments in the struts can be neglected. (Check moments in the struts to ensure that this is the case.) The pre-load is applied in two steps. First, the force is applied to the strut node with the **STRUCT node n load fx fy mn** command, and the model is stepped to an equilibrium state. After the equilibrium state is achieved with the pre-load, the pre-load force is removed, the x-velocity of the node is set to zero, and the strut node is fixed so that the load in the strut can change during subsequent excavation stages.

This problem is run for two cases: (1) pre-loading in the struts; and (2) no pre-loading. This will allow us to assess the effect of pre-loading on the displacement of the soil and wall, and on the loads in the wall and struts.
11.3 Results

The deflections of the diaphragm wall at each excavation stage are indicated by the plot of $x$-displacement of the wall structure versus wall depth in Figure 11.7 for the pre-loaded struts, and in Figure 11.8 for no pre-loading. These plots are table plots generated using the FISH function “WALL_DISP.FIS” listed in Section 11.6. The $x$-displacement and the $y$-position of each node along the wall are stored in five tables corresponding to each excavation stage. The maximum deformation is approximately 62 mm at 20 m depth for pre-loaded struts (shown in Figure 11.7) and 70 mm for no pre-load (shown in Figure 11.8) at the final stage.

The actual axial loads in the struts are calculated by the FISH functions in “STRUT_AX_LOAD.FIS” listed in Section 11.8, after the model has come to equilibrium for each excavation stage. Note that the axial loads accessed by FISH are scaled values and must be multiplied by the spacing to determine the actual values. The actual axial load values are then stored in tables for comparison at the end of the calculation. The results are shown in Figure 11.9 for pre-loaded struts, and in Figure 11.10 for no pre-load.

As shown in Figure 11.9 for the pre-load case, the axial load in strut 1 (stored in Table 1) is initially at a pre-load value of 200 kN for the first stage, increases at the second stage, and then decreases for the later stages. The axial load in strut 2 (Table 2) increases until the third stage and then decreases. The load in strut 3 (Table 3) increases until the fourth stage, while strut 4 (Table 4) increases through the fifth stage.

The results are similar for the no pre-load case, as shown in Figure 11.10.

The change in axial load is related to the movement of the wall during excavation (indicated by the plot in Figure 11.11 for pre-loaded struts, and in Figure 11.12 for no pre-loading). These figures show the evolution of the horizontal displacement at five elevations on the wall (at the top of the wall, and at the location of each strut). The top of the wall is shown to move away from the excavation (i.e., in the negative $x$-direction) after an initial inward movement, while most of the wall moves into the excavation (i.e., in the positive $x$-direction). The movements increase with depth, and coincide with the increase in axial forces for the struts.

Figures 11.13 and 11.14 plot the $x$-displacement contours and axial forces in the struts after the final excavation stage, for pre-loading and no pre-loading, respectively. Again, as shown in these figures, the movements correspond to the increase in loads in the struts. Note that actual values for the axial forces are plotted directly for these plots.

The moment distribution and shear forces in the wall, for the analysis with pre-loaded struts, are shown in Figures 11.15 and 11.17, and for the no pre-loading case in Figures 11.16 and 11.18. In both cases, a large bending moment is shown to develop in the wall at the bottom of the excavation.
Figure 11.7  x-displacement of diaphragm wall at the end of each excavation stage for pre-loaded struts

Figure 11.8  x-displacement of diaphragm wall at the end of each excavation stage for no pre-loading in struts
Figure 11.9 Actual axial forces in struts at the end of each excavation stage for pre-loaded struts

Figure 11.10 Actual axial forces in struts at the end of each excavation stage for no pre-loading in struts
**Figure 11.11** x-displacement histories at five elevations along the wall for pre-loaded struts

**Figure 11.12** x-displacement histories at five elevations along the wall for no pre-loading in struts
Figure 11.13 x-displacement contours in the grid and axial forces in the struts at the final excavation stage for pre-loaded struts

Figure 11.14 x-displacement contours in the grid and axial forces in the struts at the final excavation stage for no pre-loading in struts
Figure 11.15  Moments in the diaphragm wall at the final excavation stage (with pre-loading)

Figure 11.16  Moments in the diaphragm wall at the final excavation stage (no pre-loading)
Figure 11.17 Shear forces in the diaphragm wall at the final excavation stage (with pre-loading)

Figure 11.18 Shear forces in the diaphragm wall at the final excavation stage (no pre-loading)
11.4 Reference

11.5 Data File “DIAP.DAT”

;Project Record Tree export
;Title:Concrete diaphragm wall

;... State: dw.ini.sav ....
config gwflow ats
grid 37,35
gen (-150.0,-60.0) (-150.0,-18.0) (-4.0,-18.0) (-4.0,-60.0) ratio 0.9,0.9 &
i 1 24 j 1 18
gen (-150.0,0.0) (-150.0,0.0) (-4.0,0.0) (-4.0,-18.0) ratio 0.9,1.0 &
i 1 24 j 18 36
gen (-4.0,-60.0) (-4.0,-18.0) (10.0,-18.0) (10.0,-60.0) ratio 1.0,0.9 &
i 24 38 j 1 18
gen (-4.0,-18.0) (-4.0,0.0) (10.0,0.0) (10.0,-18.0) i 24 38 j 18 36
model elastic i=1,37 j=1,35
group 'Soil 6' j 1 7
model mohr group 'Soil 6'
prop density=1700.0 bulk=7.843E7 shear=5.32E7 cohesion=0.0 friction=38.0 &
dilation=0.0 tension=0.0 group 'Soil 6'
group 'Soil 5' j 8
model mohr group 'Soil 5'
prop density=1980.0 bulk=7.779992E7 shear=4.725E7 cohesion=0.0 &
friction=32.0 dilation=0.0 tension=0.0 group 'Soil 5'
group 'Soil 4' j 9 14
model mohr group 'Soil 4'
prop density=1970.0 bulk=4.1670004E7 shear=3.25E7 cohesion=0.0 &
friction=34.0 dilation=0.0 tension=0.0 group 'Soil 4'
group 'Soil 3' j 15 18
model mohr group 'Soil 3'
prop density=1910.0 bulk=3.3330002E7 shear=2.025E7 cohesion=0.0 &
friction=30.0 dilation=0.0 tension=0.0 group 'Soil 3'
group 'Soil 2' j 19 26
model mohr group 'Soil 2'
prop density=1980.0 bulk=3.03E7 shear=2.56E7 cohesion=0.0 friction=32.0 &
dilation=0.0 tension=0.0 group 'Soil 2'
group 'Soil 1' j 27 35
model mohr group 'Soil 1'
prop density=1880.0 bulk=1.6670001E7 shear=1.017E7 cohesion=0.0 &
friction=30.0 dilation=0.0 tension=0.0 group 'Soil 1'
model null i 28
group 'null' i 28
group delete 'null'
prop por=0.3 notnull
struct node 1 0.0,-33.7
struct node 2 0.0,-18.0

FLAC Version 6.0
struct node 3 0.0,0.0
struct beam begin node 1 end node 2 seg 10 prop 1001
struct beam begin node 2 end node 3 seg 18 prop 1001
struct prop 1001
struct prop 2001
struct prop 3001
struct prop 4001
struct prop 5001
struct prop 6001
struct prop 1001 density 2000.0 spacing 1.0 e 2.0E10 height 1.0 width 1.0
interface 1 aside from 28,8 to 28,36 bside from node 1,4 to node 3
interface 1 unglued kn=1.0E8 ks=1.0E8 cohesion=0.0 dilation=0.0 &
friction=20.0 tbond=0.0 bslip=0ff
interface 2 aside from 29,36 to 29,8 bsie from node 3,29 to node 1
interface 2 unglued kn=1.0E8 ks=1.0E8 cohesion=0.0 dilation=0.0 &
friction=20.0 tbond=0.0 bslip=0ff
ini x add -1.0 y add 0.0 nmregion 29 1
attach aside from 28,1 to 28,6 bsie from 29,1 to 29,6
interface 3 aside from 28,7 to 28,8 bsie from 29,7 to 29,8
interface 3 glued kn=1.0E8 ks=1.0E8
fix y j 1
fix x i 38
fix x i 1
set gravity=10.0
set flow=off
water density=1000.0
set echo off
call Ininv.fis
set wth=-2.0 k0x=0.5 k0z=0.5
ininv
save dw_ini.sav

;... State: dw_equil.sav ....
history 1 syy i=37, j=28
history 2 esyy i=37, j=28
history 3 pp i=37, j=28
set large
history 999 unbalanced
solve elastic
save dw_equil.sav

;... State: dw_w1.sav ....
initial xdisp 0 ydisp 0
initial xvel 0 yvel 0
initial pp 0.0 i 29 38 j 32 36
initial saturation 0.0 i 29 38 j 32 36
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 32
cscline reset
cscline 1 (4.5,-60.0) (4.5,0.0)
solve
save dw_wl.sav

;... State: dw_e1.sav ....
history reset
history 1 xdisp i=28, j=36
history 2 xdisp i=28, j=33
history 3 xdisp i=28, j=29
history 4 xdisp i=28, j=25
history 5 xdisp i=28, j=22
model null i 29 37 j 32 35
group 'null' i 29 37 j 32 35
group delete 'null'
solve
save dw_e1.sav

;*** Branch: Preload ****

;... State: dw_s1.sav ....
struct node 30 9.0,-3.0
struct node 31 0.0,-3.0 pin slave x y 27
struct beam begin node 30 end node 31 prop 1002
struct prop 1002
struct node 30 fix y r load -1e5,0.0 0.0
struct prop 1002 spacing 2.0 e 1.99999996E11 area 0.01739 I 4.03E-4
solve
struct node 30 fix x initial xvel=0.0 load 0.0,0.0 0.0
solve
save dw_s1.sav

;... State: dw_r1.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=1
wall_disp
set echo off
call init_strut_table.fis
init_strut_table
set echo off
call strut_ax_load.fis
set nstage=1 nstrut=11 el_num=29
strut_ax_load
save dw_r1.sav

FLAC Version 6.0
Analysis of a Concrete Diaphragm Wall

;... State: dw_w2.sav ....
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 28
initial pp 0.0 i 29 38 j 28 32
initial saturation 0.0 i 29 38 j 28 32
solve
save dw_w2.sav

;... State: dw_e2.sav ....
model null i 29 37 j 28 31
group ‘null’ i 29 37 j 28 31
group delete ‘null’
solve
save dw_e2.sav

;... State: dw_s2.sav ....
struct node 32 9.0,-7.0
struct node 33 0.0,-7.0 pin slave x y 23
struct beam begin node 32 end node 33 prop 1003
struct prop 1003
struct node 32 fix y r load -200000.0,0.0 0.0
struct prop 1001 height 0.0 width 0.0
struct prop 1003 spacing 2.0 e 2e11 area 0.02187 i 6.6E-4
solve
struct node 32 fix x initial xvel 0.0 load 0.0,0.0 0.0
solve
save dw_s2.sav

;... State: dw_r2.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=2
wall_disp
set echo off
call strut_ax_load.fis
set nstage=2 nstrut=11 el_num=29
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=2 nstrut=12 el_num=30
strut_ax_load
save dw_r2.sav

;... State: dw_w3.sav ....
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 24
initial saturation 0.0 i 29 38 j 24 28
initial pp 0.0 i 29 38 j 24 28
solve
save dw_w3.sav

;... State: dw_e3.sav ....
model null i 29 37 j 24 27
group 'null' i 29 37 j 24 27
group delete 'null'
solve
save dw_e3.sav

;... State: dw_s3.sav ....
struct node 34 9.0,-11.0
struct node 35 0.0,-11.0 pin slave x y 19
struct beam begin node 34 end node 35 prop 1003
struct node 34 fix y r load -225000.0,0.0 0.0
solve
struct node 34 fix x initial xvel 0.0 load 0.0,0.0 0.0
solve
save dw_s3.sav

;... State: dw_r3.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=3
wall_disp
set echo off
call strut_ax_load.fis
set nstage=3 nstrut=11 el_num=29
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=3 nstrut=12 el_num=30
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=3 nstrut=13 el_num=31
strut_ax_load
save dw_r3.sav

;... State: dw_w4.sav ....
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 21
initial pp 0.0 i 29 38 j 21 24
initial saturation 0.0 i 29 38 j 21 24
solve
save dw_w4.sav
Analysis of a Concrete Diaphragm Wall

Model null i 29 37 j 21 23
group ‘null’ i 29 37 j 21 23
group delete ‘null’
solve
save dw_e4.sav

model null i 29 37 j 21 23
group ‘null’ i 29 37 j 21 23
group delete ‘null’
solve
save dw_s4.sav

model null i 29 37 j 21 23
group ‘null’ i 29 37 j 21 23
group delete ‘null’
solve
save dw_r4.sav

initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 18
initial pp 0.0 i 29 38 j 18 21
initial saturation 0.0 i 29 38 j 18 21
solve
save dw_w5.sav

;... State: dw_e5.sav ....
model null i 29 37 j 18 20
group 'null' i 29 37 j 18 20
group delete 'null'
solve
save dw_e5.sav

;... State: dw_r5.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=5
wall_disp
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=11 el_num=29
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=12 el_num=30
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=13 el_num=31
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=14 el_num=32
strut_ax_load
save dw_r5.sav

;*** Branch: NoPreload ****
restore dw_e1.sav

;... State: noload_dw_s1.sav ....
struct node 30 9.0,-3.0
struct node 31 0.0,-3.0 pin slave x y 27
struct beam begin node 30 end node 31 prop 1002
struct prop 1002
struct prop 1002 spacing 2.0 e 1.99999996E11 area 0.01739 I 4.03E-4
struct node 30 fix x y r
solve
save noload_dw_s1.sav

;... State: noload_dw_r1.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=1
wall_disp
set echo off
call init_strut_table.fis
init_strut_table
set echo off
call strut_ax_load.fis
set nstage=1 nstrut=11 el_num=29
strut_ax_load
save noload_dw_r1.sav

;... State: noload_dw_w2.sav ....
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 28
initial pp 0.0 i 29 38 j 28 32
initial saturation 0.0 i 29 38 j 28 32
solve
save noload_dw_w2.sav

;... State: noload_dw_e2.sav ....
model null i 29 37 j 28 31
group 'null' i 29 37 j 28 31
group delete 'null'
solve
save noload_dw_e2.sav

;... State: noload_dw_s2.sav ....
struct node 32 9.0,-7.0
struct node 33 0.0,-7.0 pin slave x y 23
struct beam begin node 32 end node 33 prop 1003
struct prop 1003
struct prop 1001 height 0.0 width 0.0
struct prop 1003 spacing 2.0 e 2e11 area 0.02187 I 6.6E-4
struct node 32 fix x y r
solve
save noload_dw_s2.sav

;... State: noload_dw_r2.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=2
wall_disp
set echo off
call strut_ax_load.fis
set nstage=2 nstrut=11 el_num=29
Example Applications

strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=2 nstrut=12 el_num=30
strut_ax_load
save noload_dw_r2.sav

;... State: noload_dw_w3.sav ....
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 24
initial saturation 0.0 i 29 38 j 24 28
initial pp 0.0 i 29 38 j 24 28
solve
save noload_dw_w3.sav

;... State: noload_dw_e3.sav ....
model null i 29 37 j 24 27
group 'null' i 29 37 j 24 27
group delete 'null'
solve
save noload_dw_e3.sav

;... State: noload_dw_s3.sav ....
struct node 34 9.0,-11.0
struct node 35 0.0,-11.0 pin slave x y 19
struct beam begin node 34 end node 35 prop 1003
struct node 34 fix x y r
solve
save noload_dw_s3.sav

;... State: noload_dw_r3.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=3
wall_disp
set echo off
call strut_ax_load.fis
set nstage=3 nstrut=11 el_num=29
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=3 nstrut=12 el_num=30
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=3 nstrut=13 el_num=31
strut_ax_load
save noload_dw_r3.sav

;... State: noload_dw_w4.sav ....
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 21
initial pp 0.0 i 29 38 j 21 24
initial saturation 0.0 i 29 38 j 21 24
solve
save noload_dw_w4.sav

;... State: noload_dw_e4.sav ....
model null i 29 37 j 21 23
group 'null' i 29 37 j 21 23
group delete 'null'
solve
save noload_dw_e4.sav

;... State: noload_dw_s4.sav ....
struct node 36 9.0,-14.0
struct node 37 0.0,-14.0 pin slave x y 16
struct beam begin node 36 end node 37 prop 1003
struct node 36 fix x y r
solve
save noload_dw_s4.sav

;... State: noload_dw_r4.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=4
wall_disp
set echo off
call strut_ax_load.fis
set nstage=4 nstrut=11 el_num=29
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=4 nstrut=12 el_num=30
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=4 nstrut=13 el_num=31
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=4 nstrut=14 el_num=32
strut_ax_load
save noload_dw_r4.sav

FLAC Version 6.0
Example Applications

... State: noload_dw_w5.sav ....
initial pp 317000.0 var 0.0,-317000.0 i 29 38 j 8 18
initial pp 0.0 i 29 38 j 18 21
initial saturation 0.0 i 29 38 j 18 21
solve
save noload_dw_w5.sav

... State: noload_dw_e5.sav ....
model null i 29 37 j 18 20
group ‘null’ i 29 37 j 18 20
group delete ‘null’
solve
save noload_dw_e5.sav

... State: noload_dw_r5.sav ....
set echo off
call wall_disp.fis
set nodes_in_wall=29 n_table=5
wall_disp
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=11 el_num=29
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=12 el_num=30
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=13 el_num=31
strut_ax_load
set echo off
call strut_ax_load.fis
set nstage=5 nstrut=14 el_num=32
strut_ax_load
save noload_dw_r5.sav

;*** plot commands ****
;plot name: ydisp-pp-moments
plot hold ydisp fill int 2.5E-4 zero inv pp int 50000.0 white struct beam &
moment fill white
;plot name: xdisp histories
plot hold history 1 line 2 line 3 4 5
;plot name: wall moments
plot hold beam bound struct beam moment 1 fill max 5000000.0

FLAC Version 6.0
;plot name: strut loads
plot hold bound struct beam axial 2 5 fill max 8000000.0 lcyan xdisp fill &
   int 0.02 beam lmagenta
;plot name: wall shear
plot hold struct beam shear 1 fill max 2000000.0 lcyan beam bound
;plot name: grid
plot hold grid beam lmagenta
;plot name: Strut load vs. stage
label table 11
Strut 1
label table 12
Strut 2
label table 13
Strut 3
label table 14
Strut 4
plot hold table 14 line 13 line 12 line 11 line
;plot name: x-disp along wall
label table 1
Stage 1
label table 2
Stage 2
label table 3
Stage 3
label table 4
Stage 4
label table 5
Stage 5
plot hold table 5 line 4 line 3 line 2 line 1 line
11.6 Data File “WALL_DISP.FIS”

; Name: wall_disp
; Diagram:
; Input: nodes_in_wall/int/29/number of structural nodes along wall
; Input: n_table/int/10/table number to store vertical displacement

call str.fin

def wall_disp
  ip = imem(str_pnt+$ksnode)
  mt = 0
  loop while ip # 0
    id_node = imem(ip+$kndid)
    if id_node <= nodes_in_wall then
      mt = mt + 1
      xn_disp = fmem(ip + $kndu1)
      yn_pos = fmem(ip + $kndy)
      table(n_table,yn_pos) = xn_disp
    endif
    ip = imem(ip)
  endloop
  loop m (1,mt)
    x_value = ytable(n_table,m)
    y_value = xtable(n_table,m)
    xtable(n_table,m) = x_value
    ytable(n_table,m) = y_value
  endloop
end

FLAC Version 6.0
11.7 Data File "INIT_STRUT_TABLE.FIS"

```plaintext
;Name: init_strut_table
;Diagram:
def init_strut_table
nstage = 0
nstrut = 11
   table(nstrut,nstage) = 0
nstrut = 12
   table(nstrut,nstage) = 0
nstrut = 13
   table(nstrut,nstage) = 0
nstrut = 14
   table(nstrut,nstage) = 0
;
   nstage = 1
   nstrut = 12
   table(nstrut,nstage) = 0
nstrut = 13
   table(nstrut,nstage) = 0
nstrut = 14
   table(nstrut,nstage) = 0
;
   nstage = 2
   nstrut = 13
   table(nstrut,nstage) = 0
nstrut = 14
   table(nstrut,nstage) = 0
;
   nstage = 3
   nstrut = 14
   table(nstrut,nstage) = 0
end
```
11.8 Data File “STRUT_AX_LOAD.FIS”

;Name:strut_ax_load
;Diagram:
;Input:nstage/int/1/excavation stage number
;Input:nstrut/int/11/strut number
;Input:el_num/int/29/beam element ID number
call str.fin

def get_el_num
    ip = imem(str_pnt+$ksels)
    loop while ip # 0
        id_num = imem(ip + $kelid)
        if id_num = el_num then
            el_addr = ip
            exit
        endif
        ip = imem(ip)
    endloop
end
def get_ax_load
    ax_load = 2.0 * fmem(el_addr+$kelfax)
end
def store_ax_load
    mstrut = nstrut
    mstage = nstage
    command
        table mstrut insert mstage,ax_load
    endcommand
end
def strut_ax_load
    get_el_num
    get_ax_load
    store_ax_load
end