11 Shotcrete and Cable Support

11.1 Problem Statement

This example application demonstrates the use of the structural element logic in *UDEC* to simulate the support of a circular excavation provided by combining a shotcrete lining and cable bolts.

The geometry for this example is illustrated in Figure 11.1. A circular tunnel is excavated in a rock containing a high-angle continuous joint set oriented at 50° dip, with an average spacing of 7 m. The excavation also intersects a vertical fault, creating a triangular wedge above the crown of the tunnel.

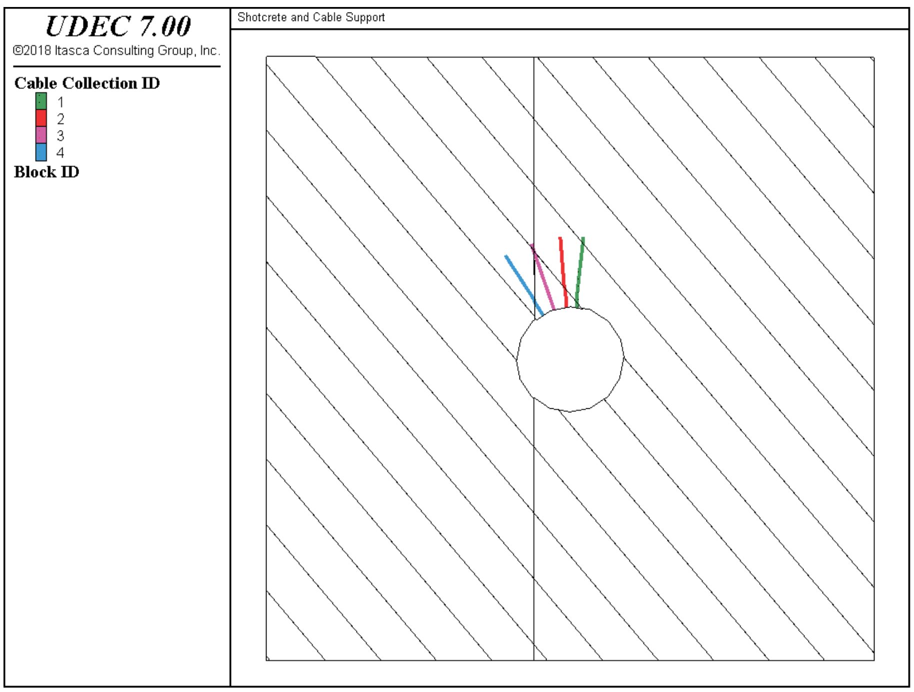

Figure 11.1 UDEC model of a circular excavation in a rock mass containing a high-angle joint set and a vertical fault – tunnel support with a shotcrete lining and cable bolts

The properties of the rock, joints and fault are summarized:

Intact Rock	
bulk modulus	1.5 GPa
shear modulus	0.6 GPa
density	2500 kg/m^3
Joints and Fault	
normal stiffness	2 GPa/m
shear stiffness	2 GPa/m
friction angle	10°
cohesion	100 Pa
tensile strength	100 Pa

Several properties are used for the shotcrete lining and cable bolts:

Shotcrete:	
thickness	0.1 m
density	2500 kg / m^3
Young's modulus	21 GPa
Poisson's ratio	0.15
tensile yield strength	2 MPa
residual yield strength	1 MPa
compressive yield strength	4 MPa
Rock/Shotcrete Interface:	
normal stiffness	1 GPa / m
shear stiffness	1 GPa / m
friction	45°
cohesive strength	1 MPa
tensile strength	1 MPa
Cable Bolts:	
cable area	10^{-3} m^2
cable length	20 m
cable modulus (E)	100 GPa
cable ultimate tensile strength	10 MN
grout bond stiffness	10^9 N/m/m
grout cohesive strength	10^6 N/m

For demonstration purposes, the tunnel is excavated and support installed instantaneously. Two support analyses are presented. In the first, only a shotcrete lining is applied. In the second, both shotcrete and cable bolts are used to provide support.

The parameters selected for this example are not intended to represent typical cable and shotcrete properties; they are chosen to provide a clear demonstration of the response of the cable and structural-element support models. For example, the compressive strength of the shotcrete is set to a very low value so that the liner fails in the first analysis; the support provided by the cables is then clearly seen in the second analysis.

11.2 *UDEC* Analysis

In the first analysis, the shotcrete lining is installed in the upper half of the tunnel. The model is brought to an equilibrium state prior to excavating the tunnel. Then the tunnel blocks are deleted, the lining is installed and the calculation is continued. A y-displacement history is recorded at one point in the tunnel crown. The history plot in Figure 11.2 indicates that the tunnel roof is collapsing. Figure 11.3 plots a close-up view of the tunnel region and lining; the wedge is falling and the lining is failing.

In the second analysis, cables are installed in the region of the roof wedge along with the shotcrete lining, as shown in Figure 11.1. In order to combine the support provided by both structural element types, the cables must be connected to the lining. This is accomplished by specifying the **connect** keyword with the **block struct cable create** command. This will position the cable node to coincide with the nearest structural element node on a lining.

The *FISH* function **place_cables** is used to place the cables around the lining, and position the starting node of each cable to coincide with a lining node. The cable pattern is defined by an origin location for the cable pattern (**xOrigin**,**yOrigin**), a radius to the remote end of the first cable in the pattern (**radius1**), a radius to the remote end of the last cable in the pattern (**radius2**), a starting angle for the cable pattern (**theta1**) and an ending angle for the cable pattern (**theta2**). See the *FISH* data file “SUP_CAB.FIS” in Example 11.2 for the listing of this function.

The following **fish set** commands define the cable pattern to span the region of the roof wedge block.

```
fish set @xOrigin=0.0 @yOrigin=0.0
fish set @radius1=20  @radius2=20
fish set @theta1=80.0 @theta2=130.0
```

When the analysis is repeated with the shotcrete and cable bolts, the roof wedge is stabilized after the wedge has moved approximately 0.45 m. Figure 11.4 plots the y-displacement history in the crown; the wedge stabilizes after approximately 0.45 m movement. The axial forces in the cables and lining are shown in the plot in Figure 11.5.

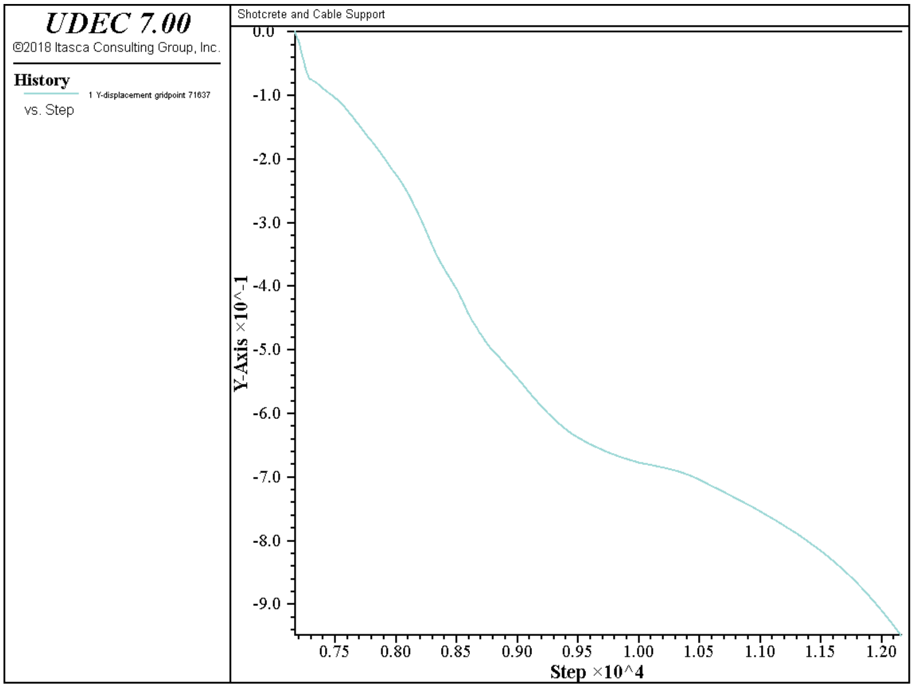

Figure 11.2 y-displacement history recorded at a location on the roof wedge – unstable movement

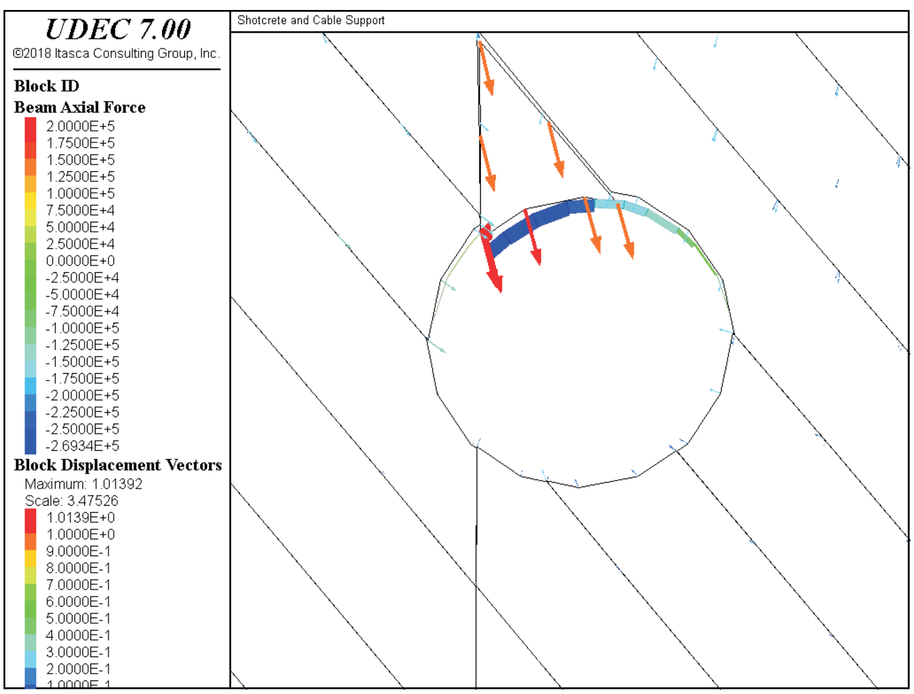

Figure 11.3 Close-up view of tunnel showing sliding of roof wedge and failure of shotcrete lining

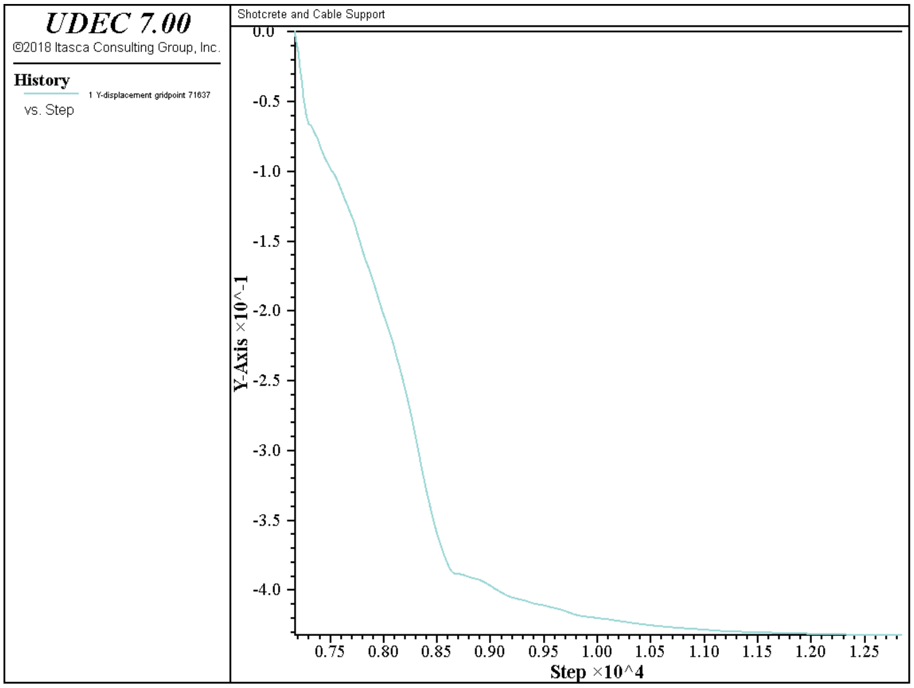

Figure 11.4 ***y-displacement history recorded at a location on the roof wedge – movement stops***

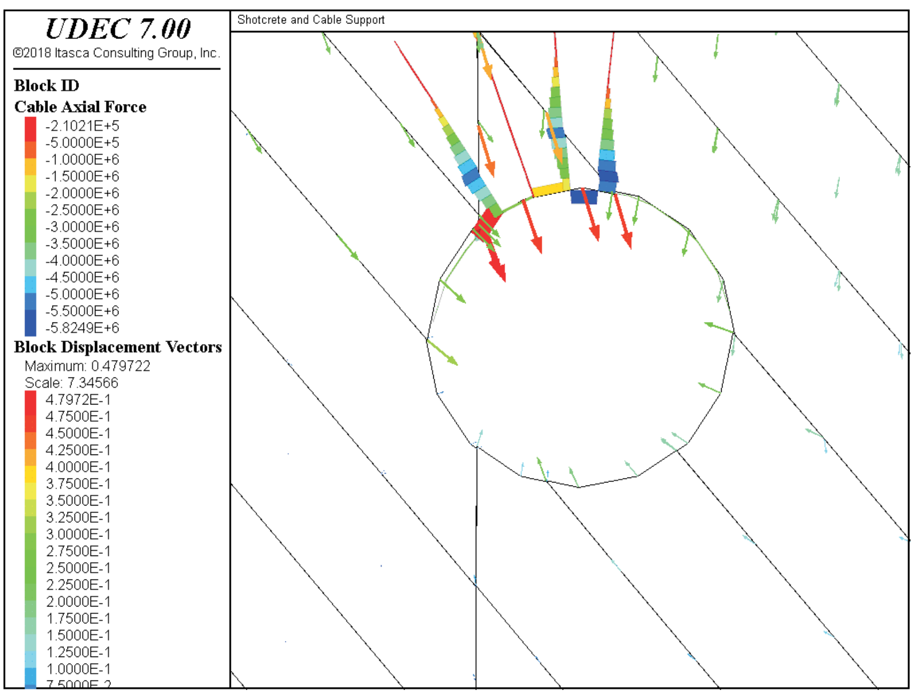

Figure 11.5 ***Close-up view of tunnel showing the stable roof wedge and axial forces in the cables and shotcrete lining***

11.3 Listing of Data Files

Example 11.1 SUPPORT.DAT

```
;File:support.dat
model new
model title 'Shotcrete and Cable Support'
block config
block tolerance corner-round-length 0.1
block tolerance minimum-edge-length 0.2
block create polygon -50 -50 -50 50 50 50 50 -50
block cut joint-set angle 310 spacing 7 origin 0 0
block cut crack -6 -50 -6 50
block cut arc 0 0 9 0 360 16 join
block zone gen edge 10.0
;
block zone group 'rock'
block zone cmodel assign elastic density 2.5E3 bulk 1.5E9 shear 6E8 ...
  range group 'rock'
block contact group 'joint'
block contact cmodel assign area stiffness-shear 1E9 ...
  stiffness-normal 2E9 friction 10 cohesion 100 tension 100 ...
  range group 'joint'
; new contact default
block contact cmodel default area stiffness-shear 1E9 ...
  stiffness-normal 2E9 friction 10 cohesion 100 tension 100
;
block edge apply stress 0.0 0.0 -1.0E7 range pos-x -50 50 pos-y 49 50
block gridpoint apply velocity-x 0 range pos-x -51 -49 pos-y -51 51
block gridpoint apply velocity-x 0 range pos-x 49 51 pos-y -51 51
block gridpoint apply velocity-y 0 range pos-x -51 51 pos-y -51 -49
model gravity 0.0 -10.0
block solve ratio 1.0E-5 elastic
model save 'supp1.sav'
;
; shotcrete liner only
block gridpoint init displacement-x 0
block gridpoint init displacement-y 0
hist reset
block mechanical time 0
block gridpoint history displacement-y 0.0 9.0
block delete range annulus center 0 0 radius 0 9
block structure liner create by-end-points ...
  begin 8.6183 1.4905 end -8.6471 1.2842 ...
  length-maximum 100.0 length-minimum 0.2 material-beam 1
block structure beam property material 1 density 2.5E3 poisson 0.15 ...
```

```
  yield-compression 4E6 yield-tension 2E6 young 2E10 ...
  yield-tension-residual 1E6 cross-sectional-area 0.2 moi 2.5E-2 ...
  shape-factor 0.8333 spacing 1 thickness 0.1 width 1 ...
  coupling-friction 45 coupling-cohesion 1E6 coupling-tension 1E6 ...
  coupling-stiffness-normal 1E9 coupling-stiffness-shear 1E9
block cycle 5000
model save 'supp2.sav'
;
; shotcrete liner and cable bolts
model restore 'supp1.sav'
block gridpoint init displacement-x 0
block gridpoint init displacement-y 0
hist reset
block mechanical time 0
block gridpoint history displacement-y 0.0 9.0
block delete range annulus center 0 0 radius 0 9
block structure liner create by-end-points ...
  begin 8.6183 1.4905 end -8.6471 1.2842 ...
  length-maximum 100.0 length-minimum 0.2 material-beam 1
block structure beam property material 1 density 2.5E3 poisson 0.15 ...
  yield-compression 4E6 yield-tension 2E6 young 2E10 ...
  yield-tension-residual 1E6 cross-sectional-area 0.2 moi 2.5E-2 ...
  shape-factor 0.8333 spacing 1 thickness 0.1 width 1 ...
  coupling-friction 45 coupling-cohesion 1E6 coupling-tension 1E6 ...
  coupling-stiffness-normal 1E9 coupling-stiffness-shear 1E9
call 'sup_cab2.fis'
fish set @radius1=20 @radius2=20 @xOrigin=0.0 @yOrigin=0.0 @theta1=80.0
fish set @theta2=130
@place_cables
block struct cable property material 3 cross-sectional-area 1E-3 ...
  density 7.5E3 rupture-tension-strain 1E30 yield-compression 1E10 ...
  yield-tension 1E7 young 1E11 grout-stiffness 1E9 grout-strength 1E6 ...
  spacing 1
block solve ratio 1.0E-5
model save 'supp3.sav'
```

Example 11.2 SUP_CAB.FIS

```
;Name:place_cables
;Input:radius1/float/20/remote radius for first cable
;Input:radius2/float/20/remote radius for last cable
;Input:xOrigin/float/0.0/x-origin of cable pattern
;Input:yOrigin/float/0.0/y-origin of cable pattern
;Input:theta1/float/80.0/starting angle of cable pattern
;Input:theta2/float/130/ending angle of cable pattern
fish def setup
; Create vars for later use
  xOrigin = 0.0      ; x-coord of cable radial centroid
  yOrigin = 0.0      ; x-coord of cable radial centroid
  theta1  = 0.0      ; starting angle for cables
  theta2  = 180.0    ; ending angle for cables
  radius1 = 0.0      ; starting radius for remote end of cables
  radius2 = 0.0      ; ending radius for remote end of cables
end
@setup

def place_cables
; This example places cable elements at structural nodes along
; a given arc of tunnel.
  ; clean out any existing tables that are in our way...
  command
    table 95 delete
    table 96 delete
    table 97 delete
    table 98 delete
  endcommand
  ; start struct node address
  node = bl.str.node.head
  counter  = 0

  ; collect relevant nodes and sort according to increasing theta
  loop while node # 0
    _x = bl.str.liner.node.pos.x(node) - xOrigin
    _y = bl.str.liner.node.pos.y(node) - yOrigin
    _t = math.atan2(_y, _x)
    if _t < 0
      _t = 2 * math.pi + _t
    endif
    th1 = math.degrad * theta1
    th2 = math.degrad * theta2
    if _t >= th1 then
```

```
      if _t <= th2 then
        counter = counter + 1
        table(95, _t) = int(node)
      endif
    endif
    node = bl.str.liner.node.next(node)
  endloop

  ; we have an ordered table of indices, so...
  loop ii (1, counter)
    _x = bl.str.liner.node.pos.x(index(int(table.y(95, ii)))) - xOrigin
    _y = bl.str.liner.node.pos.y(index(int(table.y(95, ii)))) - yOrigin
    _r = math.sqrt(_x^2 + _y^2)

    table.x(96, ii) = _x + xOrigin
    table.y(96, ii) = _y + yOrigin
    table.x(97, ii) = _r

    ; create other endpoint for cables and place
    _r = radius1+(radius2-radius1)*float((ii-1))/float((counter-1))
    if table.x(97, ii) < _r
      table.y(97, ii) = _r
      ; calculate (x, y) for cable end
      table.x(98, ii) = table.y(97, ii) * math.cos(table.x(95, ii))+xOrigin
      table.y(98, ii) = table.y(97, ii) * math.sin(table.x(95, ii))+yOrigin

      ; place the cable and connect to liner
      _x1 = table.x(96, ii)
      _y1 = table.y(96, ii)
      _x2 = table.x(98, ii)
      _y2 = table.y(98, ii)
      command
        bl struct cable create begin @_x1 @_y1  end @_x2 @_y2 ...
        segments 20 mat-steel 3 mat-grout 3 connect
      endcommand

    endif
  endloop

end
```