# 6 Elastic Behavior of a Jointed Medium

## 6.1 Problem Statement

This verification analysis demonstrates the capability of *UDEC* to simulate the elastic response of a jointed rock mass. The analytical solution given by Singh (1973) relates a transversely isotropic continuum characterization of the medium to the properties of the joints and the intact material in the medium. The representation of this characterization is given in Figure 6.1.

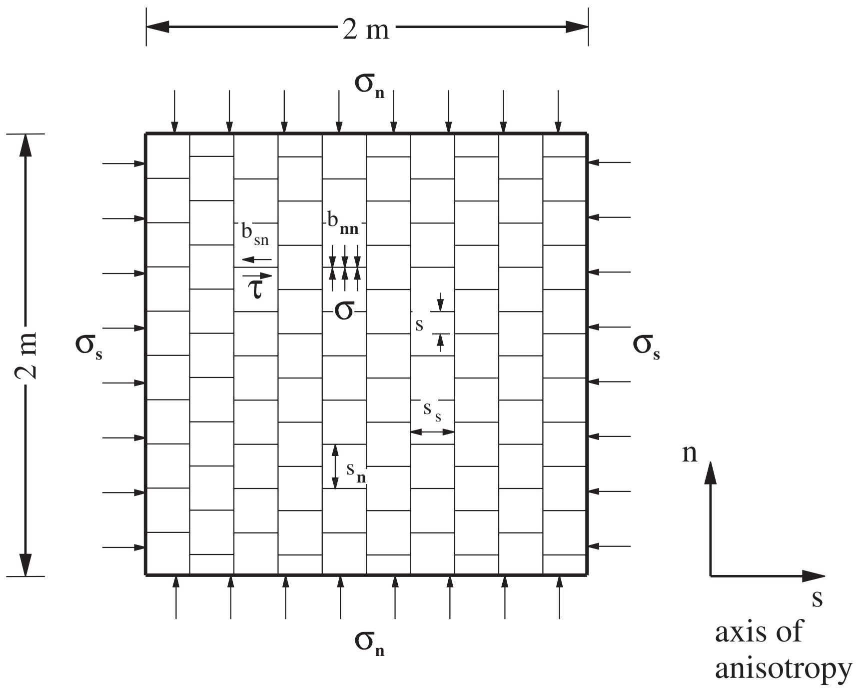


***Figure 6.1 Rock mass containing staggered joints***

Transversely isotropic, elastic rock-mass moduli can be derived in terms of joint normal and shear stiffness and intact rock moduli for two orthogonal joint sets with staggered joints. The relations for rock mass moduli defined with respect to the axes of anisotropy, $n$ and $s$, shown in Figure 6.1, are given by the equations

$$E_n = \frac{E_i}{1 + \frac{b_{nn}}{k_{nh}}\frac{E_i}{s_n}} \tag{6.1}$$

$$E_s = \frac{E_i}{1 + \frac{E_i}{k_{nv}\ s_s}} \tag{6.2}$$

$$G_{ns} = \frac{G_i\, s_s\, s_n\, k_{sh}\, k_{sv}}{s_s\, s_n\, k_{sh}\, k_{sv} + G_i\, b_{sn}\, s_s\, k_{sv} + G_i\, s_n\, k_{sh}} \tag{6.3}$$

$$\nu_{ns} = \nu_i\, \frac{k_{nh}\, s_n}{k_{nh}\, s_n + b_{nn}\, E_i} \tag{6.4}$$

where $E_n$ is the modulus of elasticity in the $n$-direction (i.e., normal to the plane of isotropy);

$E_s$ is the modulus of elasticity in the $s$-direction (i.e., in the plane of isotropy);

$G_{ns}$ is the shear modulus;

$\nu_{ns}$ is the Poisson's ratio, which relates strain in the $s$-direction to strain in the $n$-direction for loading in the $n$-direction;

$E_i$ is the intact rock elastic modulus;

$\nu_i$ is the intact rock Poisson's ratio;

$G_i$ is the intact rock shear modulus;

$k_{nv}$, $k_{sv}$ are the normal and shear stiffnesses, respectively, of the subvertical joints;

$k_{nh}$, $k_{sh}$ are the normal and shear stiffnesses, respectively, of the subhorizontal joints;

$s_s$ is the average spacing of the subvertical joints in the $s$-direction;

$s_n$ is the average spacing of the subhorizontal joints in the $n$-direction;

$s$ is the joint offset of the staggered subhorizontal joints; and

$$b_{nn} = \left[\, 1 + \frac{k_{sv}\, s}{k_{nh}\, s_s}\, (1 - \frac{s}{s_n})\, \right]^{-1} \tag{6.5}$$

$$b_{sn} = \left[\, 1 + \frac{k_{nv}\, s}{k_{sh}\, s_s}\, (1 - \frac{s}{s_n})\, \right]^{-1} \tag{6.6}$$

which are stress concentration factors for staggered joints.

The stress concentration factor is defined by Singh (1973) as the ratio of average normal and shear stresses along the joint ($\sigma$ and $\tau$ in Figure 6.1) to the corresponding overall stresses on a plane parallel to that joint within the rock mass. Eqs. (6.5) and (6.6) are derived for rigid rock blocks. Approximations for $b_{nn}$ and $b_{sn}$, which include elasticity of the rock, are also given by Singh (1973).

Properties used in this analysis represent the behavior of jointed basalt. The average spacing of the joints is assumed to be 20 cm (i.e., $s_n$ = $s_s$ = 20 cm, $s$ = 10 cm). Joint properties, estimated by Hart et al. (1985), are

$k_{nv}$ = 395.9 GPa/m

$k_{sv}$ = 438.8 GPa/m

$$k_{nh} = 531.9 \text{ GPa/m}$$
$$k_{sh} = 438.8 \text{ GPa/m}$$

The selected intact rock properties are

$$E_i = 89 \text{ GPa}$$
$$\nu_i = 0.26$$

## 6.2 *UDEC* Model

The analytical solution using Eqs. (6.1) through (6.6) is compared to the calculated results using *UDEC* for the problem setting described in Figure 6.1. The *UDEC* model for this test is shown in Figure 6.2.

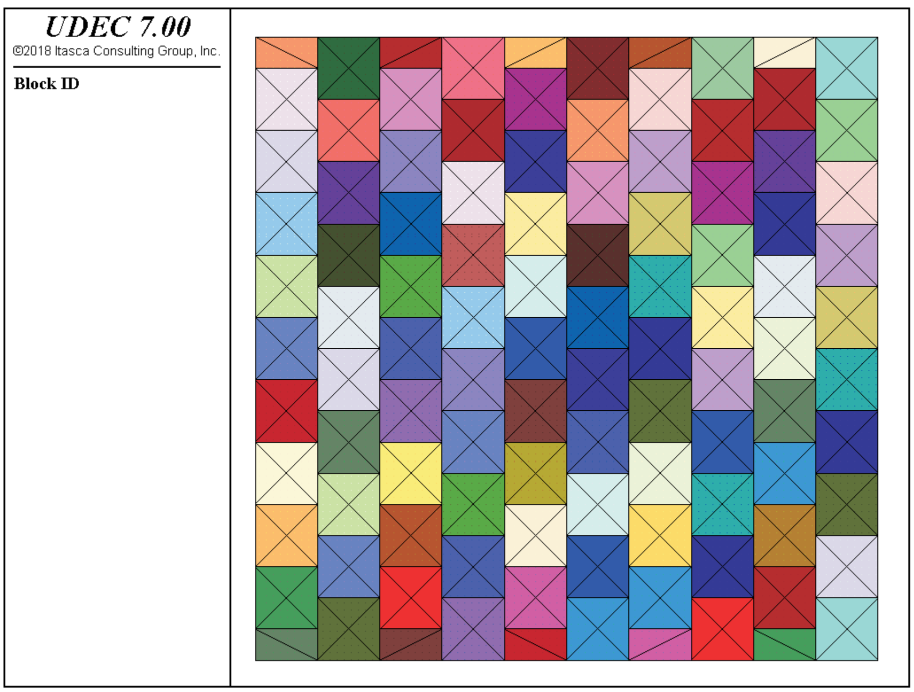


***Figure 6.2** UDEC model containing two orthogonal joint sets with staggered joints; zoning is also shown*

Two tests are performed: a load cycling in the $n$-direction while keeping the load in the $s$-direction constant; and a load cycling in the $s$-direction while keeping the $n$-direction load constant. In both tests, the cycled load is changed from 2 MPa to 10 MPa, and back to 2 MPa, while the other load is fixed at 5 MPa. The *UDEC* data file for the $n$-direction loading test is listed in "N_LOAD.DAT" in Example 6.1, and the file for the $s$-direction loading test is listed in "S_LOAD.DAT" in Example 6.2.

For the problem conditions shown in Figure 6.1, the joint spacings must be adjusted to account for the effect of the model boundaries. Therefore, the spacings used in the analytical solution are

$$s_n = 2 \text{ m}/9.5 \text{ joints} = 0.21 \text{ m/joint}$$
$$s_s = 2 \text{ m}/9 \text{ joints} = 0.22 \text{ m/joint}$$

Using the joint properties and spacings noted above, the transversely isotropic constants calculated from Eqs. (6.1) through (6.6) are

$$E_n = 53.4 \text{ GPa}$$
$$E_s = 44.0 \text{ GPa}$$
$$G_{ns} = 21.0 \text{ GPa}$$
$$\nu_{ns} = 0.16$$

The *FISH* function **trans_iso** is provided to calculate these constants. See file "SINGH.FIS" in Example 6.3.

The constants are used in stress-strain relations for transversely isotropic material behavior (e.g., see Lekhnitskii 1981) to calculate strain in the $n$-direction and $s$-direction for the two load-cycle tests. Two-dimensional plane strain conditions are imposed on these relations to make the calculations compatible with *UDEC* model conditions. *FISH* functions **trans_iso_nload** and **trans_iso_sload** calculate the strains for the $n$-direction loading and $s$-direction loading, respectively. (See file "N_LOAD.FIS" in Example 6.4, and file "S_LOAD.FIS" in Example 6.5.)

## 6.3 Results

The results from the analytical solution and those from *UDEC* are compared in Figures 6.3 and 6.4. The agreement is considered good for the two validation tests. The comparison can be further improved by expanding the model to include more joints.

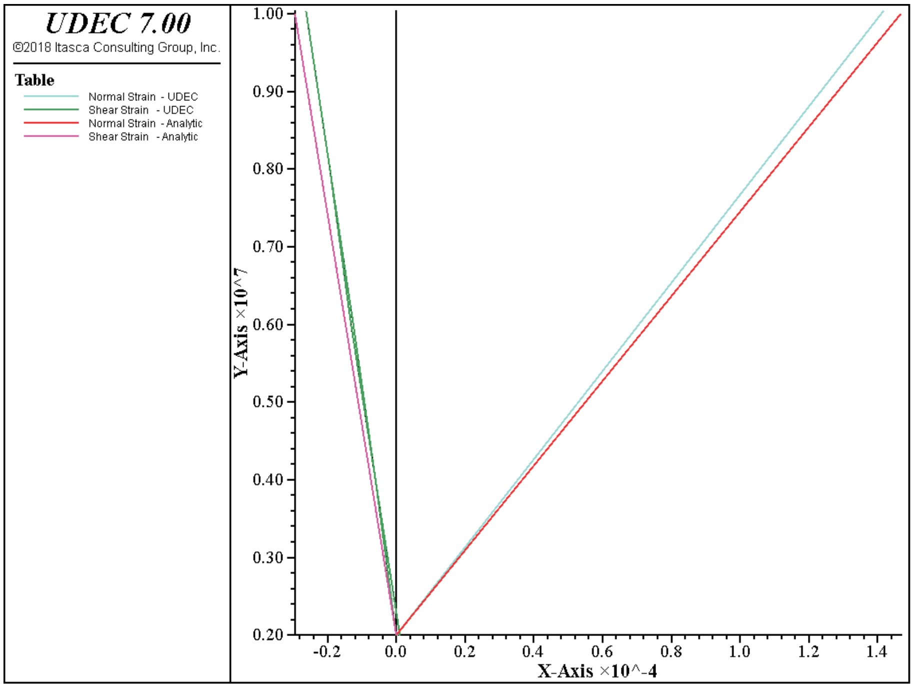


*Figure 6.3 Comparison of results for n-direction load cycling*

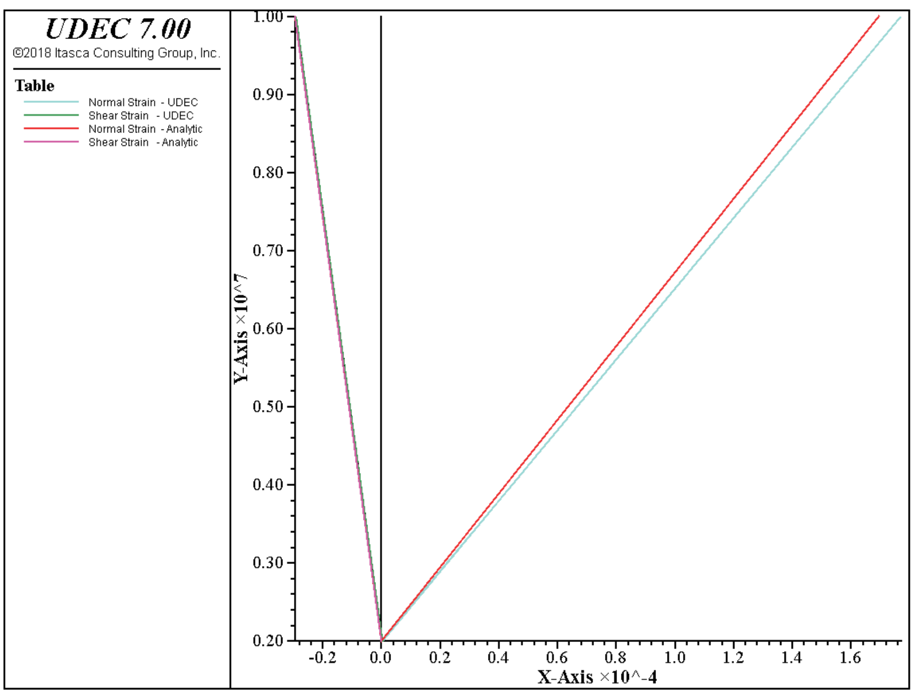


*Figure 6.4 Comparison of results for s-direction load cycling*

## 6.4 References

Hart, R. D., P. A. Cundall and M. L. Cramer. "Analysis of a Loading Test on a Large Basalt Block," in ***Research and Engineering Applications in Rock Masses (Proceedings of the 26th U.S. Symposium on Rock Mechanics)***, Vol. 2, pp. 759-768. Eileen Ashworth, ed. Boston: A. A. Balkema (1985).

Lekhnitskii, S. G. ***Theory of Elasticity of an Anisotropic Body***. Moscow: Mir Publishers (1981).

Singh, B. "Continuum Characterization of Jointed Rock Masses: Part I – The Constitutive Equations," *Int. J. Rock Mech. Min. Sci. & Geomech. Abst.*, ***10***, 337-345 (1973).

## 6.5 Listing of Data Files

### *Example 6.1 N_LOAD.DAT*

```
model new
fish define constants
   bl_k = 61.8e9
   bl_g = 35.3e9
   bl_e = (9.0 * bl_k * bl_g) / (3.0 * bl_k + bl_g)
   bl_p = (3.0 * bl_k - 2.0 * bl_g) / (2.0 * (3.0 * bl_k + bl_g))
;
   j_kn_hor = 531.9e9
   j_ks_hor = 438.8e9
   j_sp_n   = 0.21
   j_kn_ver = 395.9e9
   j_ks_ver = 438.8e9
   j_sp_s   = 0.22
   j_sp_stg = 0.1
end
@constants
bl tolerance corner-round-length 0.001
bl create polygon 0 0 0 2.0 2.0 2.0 2.0 0
bl cut j-set angle 90.0 trace 5.0 gap 0 spac 0.2
bl cut j-set angle 0.0 trace 0.21 gap 0.19 spac 0.2 or -0.20 0.0
bl cut j-set angle 0.0 trace 0.21 gap 0.19 spac 0.2 or 0.0 0.1
bl zone gen edge 0.28
;  intact bl properties
bl prop mat 1 density 2850 bulk @bl_k shear @bl_g
;  horizontal joint properties
bl contact prop mat 1 st-n @j_kn_hor st-s @j_ks_hor fric 100 tens 1e10
;  vertical joint properties
bl contact prop mat 2 st-n @j_kn_ver st-s @j_ks_ver fric 100 tens 1e10
bl contact change mat 2 range angle 90 tol 2
; stress state for loading in n-direction
bl edge apply stress -5.0e6 0.0 -2.0e6
bl insitu  stress -5.0e6 0.0 -2.0e6
;  apply load cycle
fish define load_bl
   loop n_step (1,8)
      if n_step <= 4 then
         command
            bl edge apply stress 0.0 0.0 -2.0e6 ...
              range pos-x -.1 2.1 pos-y -.1 .1
            bl edge apply stress 0.0 0.0 -2.0e6 ...
              range pos-x -.1 2.1 pos-y 1.9 2.1
            bl solve force 100
```

```
            endcommand
         endif
         if n_step > 4 then
            command
               bl edge apply stress 0.0 0.0 2.0e6 ...
                 range pos-x -.1 2.1 pos-y -.1 .1
               bl edge apply stress 0.0 0.0 2.0e6 ...
                 range pos-x -.1 2.1 pos-y 1.9 2.1
               bl solve force 100
            endcommand
         endif
         ntab = ntab + 1
         tot_sstr = 0.0
         tot_nstr = 0.0
         n_nz = 0
         loop n (1,10)
            x_z = 0.2 * float(n)
            iz = bl.zone.near(x_z,1.0)
            tot_nstr = tot_nstr + bl.zone.stress.yy(iz)
            n_nz = n_nz + 1
         endloop
         p_s1 = bl.gp.near(0.0,1.0)
         p_s2 = bl.gp.near(2.0,1.0)
         p_n1 = bl.gp.near(1.0,0.0)
         p_n2 = bl.gp.near(1.0,2.0)
         s_strn = (bl.gp.disp.x(p_s1)-bl.gp.disp.x(p_s2))/2.0
         n_strn = (bl.gp.disp.y(p_n1)-bl.gp.disp.y(p_n2))/2.0
         n_strs = - tot_nstr / n_nz
         table.x(1,ntab) = n_strn
         table.y(1,ntab) = n_strs
         table.x(2,ntab) = s_strn
         table.y(2,ntab) = n_strs
      endloop
   end
   table '1' add 0 2e6
   table '2' add 0 2e6
   fish set @ntab = 1
   @load_bl
   ;
   call 'singh.fis'
   call 'n_load.fis'
   model save 'n_load.sav'

   return
```

***Example 6.2 S_LOAD.DAT***

```
model new
fish define constants
   bl_k = 61.8e9
   bl_g = 35.3e9
   bl_e = (9.0 * bl_k * bl_g) / (3.0 * bl_k + bl_g)
   bl_p = (3.0 * bl_k - 2.0 * bl_g) / (2.0 * (3.0 * bl_k + bl_g))
;
   j_kn_hor = 531.9e9
   j_ks_hor = 438.8e9
   j_sp_n   = 0.21
   j_kn_ver = 395.9e9
   j_ks_ver = 438.8e9
   j_sp_s   = 0.22
   j_sp_stg = 0.1
end
@constants
bl tolerance corner-round-length 0.001
bl create polygon 0 0 0 2.0 2.0 2.0 2.0 0
bl cut j-set angle 90.0 trace 5.0 gap 0 spac 0.2
bl cut j-set angle 0.0 trace 0.21 gap 0.19 spac 0.2 origin -0.20 0.0
bl cut j-set angle 0.0 trace 0.21 gap 0.19 spac 0.2 origin 0.0 0.1
bl zone gen edge 0.28
;  intact bl properties
bl prop mat 1 density 2850 bulk @bl_k shear @bl_g
;  horizontal joint properties
bl contact prop mat 1 st-n @j_kn_hor st-s @j_kn_hor fric 100 tens 1e10
;  vertical joint properties
bl contact prop mat 2 st-n @j_kn_ver st-s @j_ks_ver fric 100 tens 1e10
bl contact change mat 2 range angle 90 tol 2
; stress state for loading in s-direction
bl edge apply stress -2.0e6 0.0 -5.0e6
bl insitu  stress -2.0e6 0.0 -5.0e6
;  apply load cycle
fish define load_bl
   loop n_step (1,8)
      if n_step <= 4 then
         command
            bl edge apply stress -2.0e6 0.0 0.0 ...
              range pos-x -.1 .1 pos-y -.1 2.1
            bl edge apply stress -2.0e6 0.0 0.0 ...
              range pos-x 1.9 2.1 pos-y -.1 2.1
            bl solve force 100
         endcommand
```

```
      endif
      if n_step > 4 then
         command
            bl edge apply stress 2.0e6 0.0 0.0 ...
              range pos-x -.1 .1 pos-y -.1 2.1
            bl edge apply stress 2.0e6 0.0 0.0 ...
              range pos-x 1.9 2.1 pos-y -.1 2.1
            bl solve force 100
         endcommand
      endif
      ntab = ntab + 1
      tot_sstr = 0.0
      tot_nstr = 0.0
      n_sz = 0
      loop n (1,10)
         y_z = 0.2 * float(n)
         iz = bl.zone.near(1.0,y_z)
         tot_sstr = tot_sstr + bl.zone.stress.xx(iz)
         n_sz = n_sz + 1
      endloop
      p_s1 = bl.gp.near(0.0,1.0)
      p_s2 = bl.gp.near(2.0,1.0)
      p_n1 = bl.gp.near(1.0,0.0)
      p_n2 = bl.gp.near(1.0,2.0)
      s_strn = (bl.gp.disp.x(p_s1)-bl.gp.disp.x(p_s2))/2.0
      n_strn = (bl.gp.disp.y(p_n1)-bl.gp.disp.y(p_n2))/2.0
      s_strs = - tot_sstr / n_sz
      table.x(1,ntab) = s_strn
      table.y(1,ntab) = s_strs
      table.x(2,ntab) = n_strn
      table.y(2,ntab) = s_strs
   endloop
end
table ’1’ add 0 2e6
table ’2’ add 0 2e6
fish set @ntab = 1
@load_bl
;
call ’singh.fis’
call ’s_load.fis’
model save ’s_load.sav’
return
```

***Example 6.3 SINGH.FIS***

```
fish define trans_iso
   b1 = 1.0 - (j_sp_stg / j_sp_n)
   b2 = (j_ks_ver * j_sp_stg) / (j_kn_hor * j_sp_s)
   b_nn = 1.0 / (1.0 + b2 * b1)
   b3 = (j_kn_ver * j_sp_stg) / (j_ks_hor * j_sp_s)
   b_sn = 1.0 / (1.0 + b3 * b1)
   t1 = 1.0 + (b_nn * bl_e) / (j_kn_hor * j_sp_n)
   t2 = 1.0 + bl_e / (j_kn_ver * j_sp_s)
   gt = bl_g * j_sp_s * j_sp_n * j_ks_hor * j_ks_ver
   g1 = j_sp_s * j_sp_n * j_ks_hor * j_ks_ver
   g2 = bl_g * b_sn * j_sp_s * j_ks_ver
   g3 = bl_g * j_sp_n * j_ks_hor
   pt = j_kn_hor * j_sp_n
   p1 = j_kn_hor * j_sp_n
   p2 = b_nn * bl_e
; Modulus of elasticity in n-direction
   em_n = bl_e / t1
; Modulus of elasticity in s-direction
   em_s = bl_e / t2
; Shear modulus
   gm_ns = gt / (g1 + g2 + g3)
; Poisson's ratio
   pm_ns = bl_p * pt / (p1 + p2)
end
@trans_iso
fish list @em_n
fish list @em_s
fish list @gm_ns
fish list @pm_ns
;
```

***Example 6.4 N_LOAD.FIS***

```
fish define trans_iso_nload
   loop n_t (1,9)
      str_n  = float(n_t) * 2.0e6
      if str_n > 10e6 then
         str_n = 10e6 - ((float(n_t) - 5.0) * 2.0e6)
      endif
      str_s = 5.0e6
      str_t  = bl_p * str_s + (pm_ns * em_s / em_n) * str_n
      pm_em  = pm_ns / em_n
```

```
;
      strn_n = str_n / em_n - (str_s + str_t) * pm_em
      if n_t = 1 then
         strn_n1 = strn_n
      endif
      strn_n = strn_n - strn_n1
;
      strn_s = (str_s - bl_p * str_t) / em_s - pm_em * str_n
      if n_t = 1 then
         strn_s1 = strn_s
      endif
      strn_s = strn_s - strn_s1
;
      table.x(3,n_t) = strn_n
      table.y(3,n_t) = str_n
      table.x(4,n_t) = strn_s
      table.y(4,n_t) = str_n
   endloop
end
@trans_iso_nload
table 1 label 'Normal Strain  - UDEC    '
table 2 label 'Shear Strain   - UDEC    '
table 3 label 'Normal Strain  - Analytic'
table 4 label 'Shear Strain   - Analytic'
ret
```

### *Example 6.5 S_LOAD.FIS*

```
fish define trans_iso_sload
   loop n_t (1,9)
      str_s  = float(n_t) * 2.0e6
      if str_s > 10e6 then
         str_s = 10e6 - ((float(n_t) - 5.0) * 2.0e6)
      endif
      str_n = 5.0e6
      str_t  = bl_p * str_s + (pm_ns * em_s / em_n) * str_n
      pm_em  = pm_ns / em_n
;
      strn_n = str_n / em_n - (str_s + str_t) * pm_em
      if n_t = 1 then
         strn_n1 = strn_n
      endif
      strn_n = strn_n - strn_n1
;
      strn_s = (str_s - bl_p * str_t) / em_s - pm_em * str_n
```

```
      if n_t = 1 then
         strn_s1 = strn_s
      endif
      strn_s = strn_s - strn_s1
;
      table.x(3,n_t) = strn_s
      table.y(3,n_t) = str_s
      table.x(4,n_t) = strn_n
      table.y(4,n_t) = str_s
   endloop
end
@trans_iso_sload
table 1 label 'Normal Strain  - UDEC    '
table 2 label 'Shear Strain   - UDEC    '
table 3 label 'Normal Strain  - Analytic'
table 4 label 'Shear Strain   - Analytic'
ret
```