1 Cyclic Loading of a Specimen with a Slipping Crack

1.1 Problem Statement

This problem concerns an elastic block with an inclined internal closed crack (Figure 1.1) subject to a cycle of uniaxial loading.*

A constant axial displacement, u_a, is applied to one end of the block, and the other end is fixed. The resulting load causes inelastic slip on the crack. At some point, the sense of displacement on the end of the block is reversed until the original position is reestablished. Olsson (1982) showed that the stress-displacement relation for the loaded specimen is composed of three distinct components (Figure 1.2):

(1) a loading segment (OA), which involves elastic deformation and slip along the crack;

(2) an initial unloading segment (AB), where the crack does not slip; and

(3) a final unloading segment (BO), again with elastic deformation and slip.

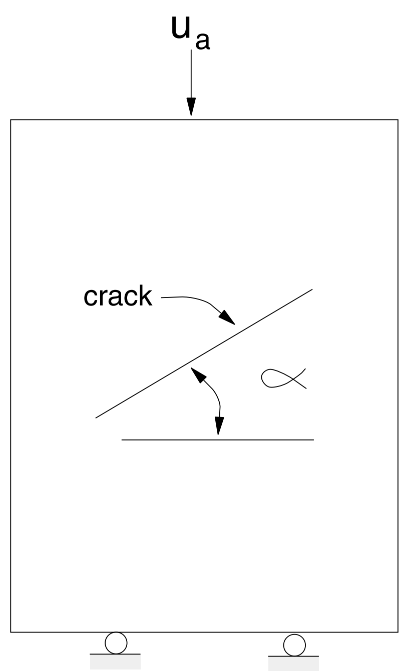

Figure 1.1 Specimen with embedded crack

* This section was prepared for the Center for Nuclear Waste Regulatory Analysis (CNWRA) under Contract No. NRC-02-88-005.

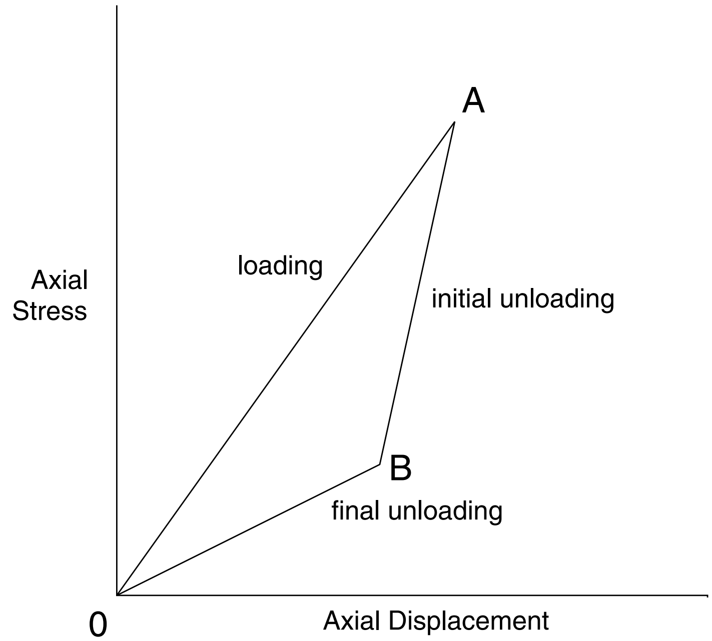

Figure 1.2 Stress-displacement relation for elastic specimen with embedded crack subjected to uniaxial load

The objective of this analysis is to evaluate the capability of *UDEC* using either the Coulomb slip model or the continuously yielding joint model to produce this behavior.

The following problem conditions are prescribed for this evaluation. A single inclined crack is located in an elastic medium. The medium has several mechanical properties:

elastic modulus (E')	80.0 MPa
Poisson's ratio (ν')	0.33
height (H)	2 m
width (W)	1 m

The crack has several properties:

joint normal stiffness (K_n)	20 GPa/m
joint shear stiffness (K_s)	20 GPa/m
joint friction angle (ϕ)	16°
joint inclination (α)	45°
slipping portion of crack (ℓ)	0.54 m

1.2 Conceptual Model

Several investigators have proposed simple conceptual models of a single, closed crack to explain phenomena associated with the deformational response of jointed rock (e.g., Walsh 1965, and Jaeger and Cook 1976). One such model is a single crack embedded in an elastic solid subjected to a cycle of uniaxial compression.

Brady et al. (1985) present relations for the three slopes (Figure 1.2) in terms of the elastic stiffness of the solid: the elastic and frictional properties of the crack, and the orientation of the crack. The conceptual model is illustrated in Figure 1.3.

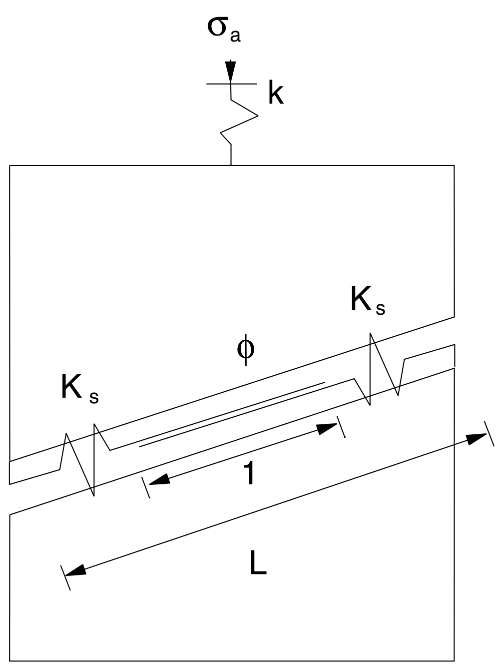

Figure 1.3 Conceptual model of an elastic specimen containing an embedded crack

In the conceptual model, k is the equivalent axial elastic stiffness of the specimen, including the throughgoing discontinuity. The equivalent elastic stiffness is given by

$$\frac{1}{k} = \frac{H}{W E'} + \frac{\cos^2 \alpha}{K_n\ L} + \frac{\sin^2 \alpha}{K_s\ L} \tag{1.1}$$

where $L = W / \cos \alpha$.

Note that the term $(H/W E')$ in Eq. (1.1) represents the uniaxial elastic stiffness of the solid in the conceptual model for plane-stress conditions. The analysis in *UDEC* is based on plane-strain conditions. However, the formal equivalence between the plane-stress and plane-strain conditions is represented by the relations between Young's modulus and Poisson's ratio for plane strain and plane stress:

$$E = \frac{1 + 2\,\nu'}{(1+\nu')^2}\,E' \tag{1.2}$$

$$\nu = \frac{\nu'}{1+\nu'} \tag{1.3}$$

where E and ν are the Young's modulus and the Poisson's ratio for plane strain; and
E' and ν' are the equivalent plane stress parameters.

The stiffnesses for the three slopes are shown by Brady et al. (1985):

$$\text{slope OA} = \frac{k}{1 + \frac{k\ \sin\alpha\ \sin(\alpha-\phi)}{K_s\ (L-1)\ \cos\phi}} \tag{1.4}$$

$$\text{slope AB} = k \tag{1.5}$$

$$\text{slope BO} = \frac{k}{1 + \frac{k\ \sin\alpha\,\sin(\alpha+\phi)}{K_s\ (L-1)\ \cos\phi}} \tag{1.6}$$

1.3 *UDEC* Model

The *UDEC* analysis assumes that the specimen is restrained perpendicular to the plane of analysis (i.e., plane-strain conditions). It is further assumed that the crack can be represented by a single throughgoing discontinuity, with only the central section of the discontinuity allowed to slip. The ends of the discontinuity are prevented from slipping by specifying these sections as construction joints. The procedure to do this is described below.

The material in which the crack is embedded is linearly elastic, homogeneous and isotropic. In order to simulate the elastic deformability in *UDEC*, the blocks are discretized into constant strain finite difference triangles, as shown in Figure 1.4.

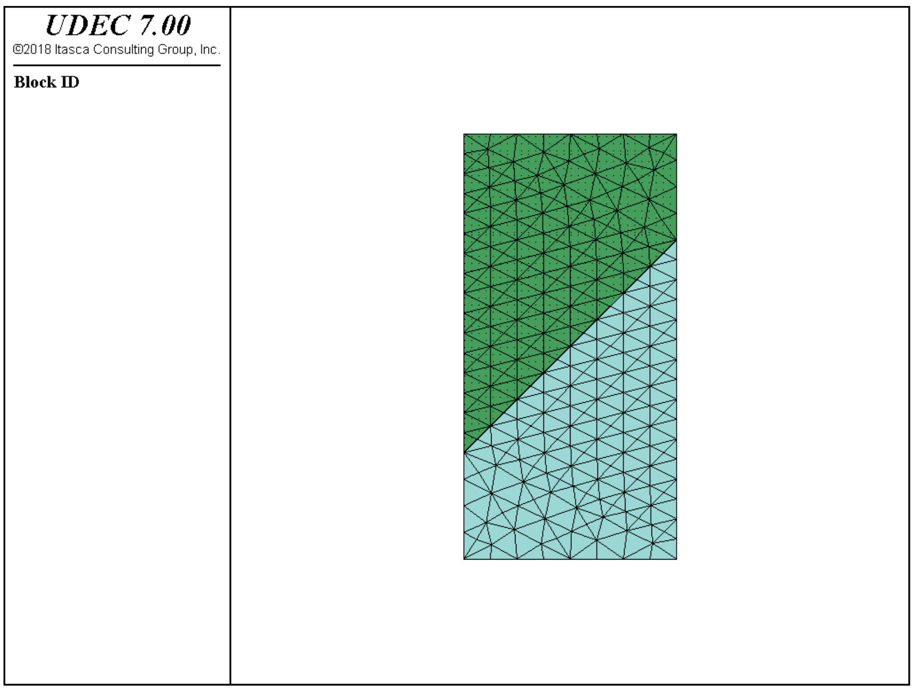

Figure 1.4 Zoning for slipping crack model

Two alternatives for the joint constitutive relation were studied:

Case A – standard linear deformation, Coulomb friction model

Case B – continuously yielding model

In both cases, the elastic, non-slipping sections of the crack are modeled as construction joints. The crack is first created as a throughgoing construction joint using the command

```
block cut crack (0,0.5) (1,1.5) join
```

Then, the center section is unjoined using the command

```
block contact join by-cont off range 0.3,0.7 0.74,1.28
```

The center section of the crack is assigned parameters that permit slip to occur. The specific *UDEC* parameters used for each joint relation are presented in Table 1.1.

Table 1.1 Joint parameters

Coulomb Friction (**jcons = 2**)	Continuously Yielding (**jcons = 3**)
st-no = 20 GPa/m	**st-no** = 20 GPa/m
st-sh = 20 GPa/m	**st-sh** = 20 GPa/m
fric = 16°	**fric** = 16°
	exp-no = 0
	exp-sh = 0
	fri-init = 16°
	rough = 1.0×10^{-10} m

The Coulomb model is a linear elastic-perfectly plastic constitutive relation, and corresponds to the concepts used in developing the expressions for the three stiffnesses (Eqs. (1.4) to (1.6)) in the conceptual model. The continuously yielding joint model is nonlinear, and therefore does not comply with the concepts used to develop the conceptual model. The model can be made to approximate the Coulomb slip model by setting the initial friction angle (**friction-init**) equal to the intrinsic friction angle (**friction**), and by setting the joint roughness parameter (**roughness**) to a small number.

It is also necessary to set the elastic stiffnesses of the construction joint sections to be consistent with those of the slipping section of the crack. This is accomplished with the command

```
block contact join-stiff-normal 2e10 join-stiff-shear 2e10
```

Note that, by default, elastic stiffnesses of construction joints are determined based upon the stiffness of the surrounding deformable block zones. (See Section 3.2.3 in the **User's Guide**.)

A *FISH* function (**slip_load**) is used to calculate the axial stress and axial displacement during the calculation and store the results in a table (Table 1). In *FISH* function **conc_slip**, axial stresses versus displacements are calculated using the stiffnesses given in Eqs. (1.4) through (1.6), and stored in Table 2, for comparison to Table 1. The *FISH* functions are contained in the file "SLIP.FIS" listed in Example 1.3.

1.4 Results and Discussion

There is no analytical solution to the problem of an elastic body with an internal inclined slipping crack, because stress conditions at each end of the crack are very complex. However, the simple conceptual model described here does capture the essential features of the problem (i.e., three distinctly different global stiffnesses) observed in cyclic loading.

The results for both of the joint constitutive models in *UDEC* compare reasonably well with those from the conceptual model, as shown in Figures 1.5 and 1.6. The *UDEC* results produce the three distinct stress-displacement components, as indicated in the figures.

The calculated slopes cannot be expected to agree closely with those determined from the conceptual model, because of the simplifying assumptions of the conceptual model. This is expected because the conceptual model assumes uniform distribution of normal stress on the crack and elastic extensions, and stress concentrations (particularly joint normal stress) become more significant as the length of the slipping crack increases.

Nevertheless, the ability of *UDEC* to produce the distinct characteristic upon loading and unloading of the slipping crack is shown. The model also provides a way to evaluate the influence of the properties of the medium and the crack on the load-displacement response.

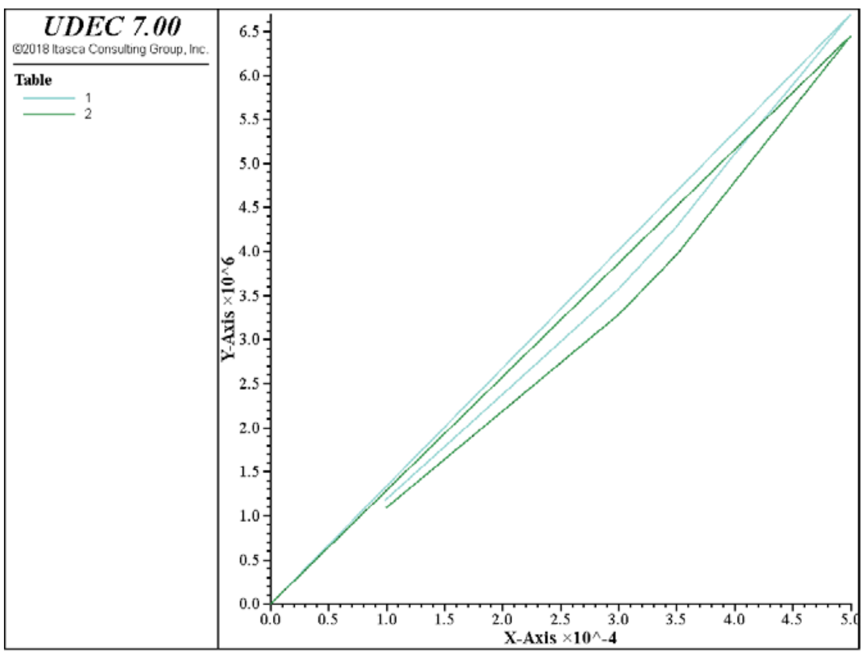

Figure 1.5 Axial stress versus axial displacement for load cycling of a specimen with a slipping crack modeled with the Coulomb friction law

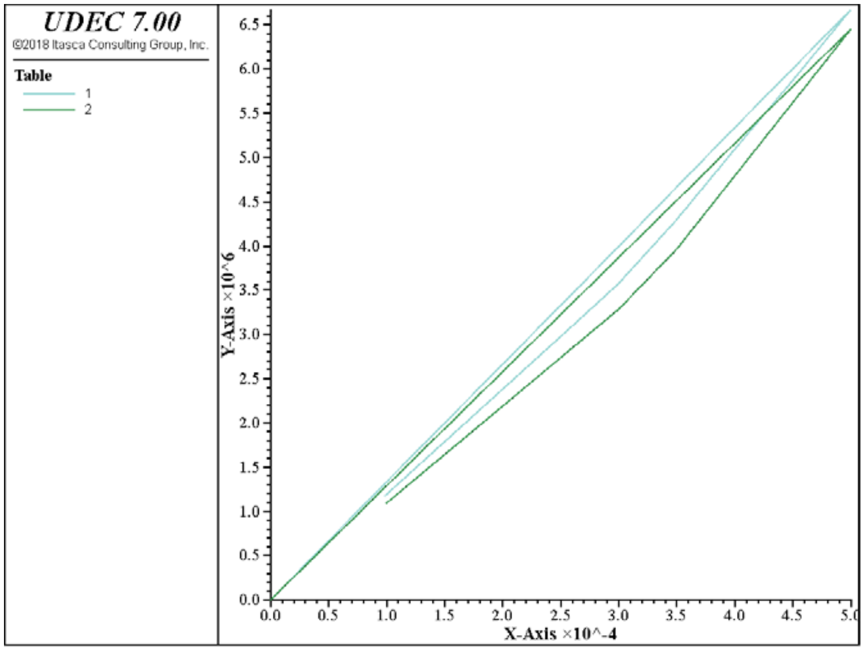

***Figure 1.6** Axial stress versus axial displacement for load cycling of a specimen with a slipping crack modeled with the continuously yielding joint model*

1.5 References

Brady, B. H. G., M. L. Cramer and R. D. Hart. “Preliminary Analysis of a Loading Test on a Large Basalt Block,” (Tech. Note), *Int. J. Rock Mech. Min. Sci. & Geomech. Abst.*, ***22***(5), 345-348 (1985).

Jaeger, J. C., and N. G. W. Cook. ***Fundamentals of Rock Mechanics***, 2nd Ed., pp. 329-333. London: Chapman and Hall (1976).

Olsson, W. A. “Experiments on a Slipping Crack,” *Geophys. Res. Letters*, ***9***(8), 797-800 (1982).

Walsh, J. B. “The Effect of Cracks on the Compressibility of Rock,” ***J. Geophys. Res.***, ***70***(2), 381-389 (1965).

1.6 Listing of Data Files

Example 1.1 SLIP.DAT

```
model new
;File:SLIP.dat
model Title 'Cyclic Loading of a Specimen with a Slipping Crack'
fish def const
  bl_k = 50.0e9
  bl_g = 30.0e9
  bl_h = 2.0
  bl_w = 1.0
  cr_l = 0.54
  bl_ang = 45.0
  j_kn = 20.0e9
  j_ks = 20.0e9
  j_fr = 16.0
  ntab = 1
end
@const
call 'slip.fis'
block tolerance corner-round-length 0.001
block tolerance minimum-edge-length 0.002
block create polygon 0,0 0,2 1,2 1,0
block cut crack (0,0.5) (1,1.5)
block zone gen edge 0.15
;
; elastic block properties
block zone group 'block'
bl zone cmodel assign elastic dens 2.85E3 bulk 5E10 shear 3E10 ...
  range group 'block'
;
bl contact cmodel assign area st-shear 2E10 st-normal 2E10 ...
  cohesion 1e20 friction 45
bl contact group 'joint' range position-x 0.3 0.7 position-y 0.74 1.28
bl contact cmodel assign area st-shear 2E10 st-normal 2E10 ...
  friction 16 range group 'joint'
; new contact default
bl contact cmodel default area st-shear 2E10 st-normal 2E10 friction 16
;
block gridpoint history disp-y 0.5,2.0
block zone history stress-yy 0.5,2.0
;
;  fix the bottom boundary
bl grid apply vel-y 0 range position-x -0.1 1.1 position-y -0.1 0.1
;
```

```
model save 'slip.sav'
;
;load step 1
bl grid apply vel-y -4.43E-2 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
bl grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step 2
bl grid apply vel-y -4.43E-2 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
bl grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step3
block grid apply vel-y -4.43E-2 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step 4
block grid apply vel-y -4.43E-2 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step 5
block grid apply vel-y -4.43E-2 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;
;unload step 1
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 2
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
@slip_load
;unload step 3
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
```

```
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 4
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 5
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 6
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 7
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 8
block grid apply vel-y 0.0222 range pos-x -0.1 1.1 pos-y 1.9 2.1
block cycle 400
block grid apply vel-y -0 range pos-x -0.1 1.1 pos-y 1.9 2.1
block solve force 10.0
@slip_load
model save 'slip1.sav'
```

Example 1.2 SLIP_CY.DAT

```
model new
;File:SLIP_CY.dat
model Title 'Cyclic Loading of a Specimen with a Slipping Crack - CY Model'
fish def const
  bl_k = 50.0e9
  bl_g = 30.0e9
  bl_h = 2.0
  bl_w = 1.0
  cr_l = 0.54
  bl_ang = 45.0
  j_kn = 20.0e9
  j_ks = 20.0e9
  j_fr = 16.0
  ntab = 1
end
@const
call 'slip.fis'
block tolerance corner-round-length 0.001
block tolerance minimum-edge-length 0.002
block create polygon 0,0 0,2 1,2 1,0
block cut crack (0,0.5) (1,1.5)
block zone gen edge 0.15
;
; elastic block properties
block zone group 'block'
block zone cmodel assign elastic density 2.85E3 bulk 50e9 ...
  shear 30e9 range group 'block'

block contact cmodel assign area stiffness-shear 2E10 ...
  stiffness-normal 2E10 cohesion 1e20 friction 45
; crack properties,  continuously yielding contact model
block contact group 'cyjoint' range pos-x 0.3,0.7 pos-y 0.74,1.28
block contact cmodel assign cy friction 16 friction-initial 16 ...
  roughness 1E-10 stiffness-shear 2E10 stiffness-normal 2E10 ...
  jen 0 jes 0 range group 'cyjoint'
; new contact default
block contact cmodel default area stiffness-shear=2E10 ...
  stiffness-normal=2E10 friction=16
;
block gridpoint history disp-y 0.5,2.0
block zone history stress-yy 0.5,2.0
;
;  fix the bottom boundary
```

```
bl grid app velocity-y 0 ...
  range pos-x -2.91139E-2,1.0319 pos-y -2.481E-2,1.97469E-2
;
model save 'slip_cy.sav'
;
;load step 1
bl grid app velocity-y -4.43E-2 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step 2
bl grid app velocity-y -4.43E-2 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step3
bl grid app velocity-y -4.43E-2 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step 4
bl grid app velocity-y -4.43E-2 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;load step 5
bl grid app velocity-y -4.43E-2 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;
;unload step 1
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 2
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
```

```
@slip_load
;unload step 3
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 4
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 5
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 6
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 7
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
;unload step 8
bl grid app velocity-y 0.0222 range pos-x -0.1 1.1  pos-y 1.9 2.1
block cycle  400
bl grid app velocity-y -0 range pos-x -0.1 1.1  pos-y 1.9 2.1
block solve force 10.0
@slip_load
model save 'slip2.sav'
```

Example 1.3 SLIP.FIS

```
fish define ana_slip
   bl_rad = bl_ang * math.degrad
   j_frad = j_fr * math.degrad
   bl_l   = bl_w / math.cos(bl_rad)
   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))
   bl_pp  = bl_p / (1.0 - bl_p)
   bl_ep  = bl_e * ((1.0 + bl_pp)^2) / (1.0 + 2.0 * bl_pp)
;
   k_1 = bl_h / (bl_w * bl_ep)
   k_2 = (math.cos(bl_rad))^2 / (j_kn * bl_l)
   k_3 = (math.sin(bl_rad))^2 / (j_ks * bl_l)
   k_ab = 1.0 / (k_1 + k_2 + k_3)
;
   t_1 = k_ab * math.sin(bl_rad) * math.sin(bl_rad - j_frad)
   t_2 = j_ks + (bl_l - cr_l) * math.cos(j_frad)
   k_oa = k_ab / (1.0 + (t_1 / t_2))
;
   t_4 = k_ab * math.sin(bl_rad) * math.sin(bl_rad + j_frad)
   k_bo = k_ab / (1.0 + (t_4 / t_2))
;
   loop ntab2 (1,14)
      ax_dis  = (float(ntab2) - 1.0) * 1.0e-4
      if ax_dis > 5.0e-4 then
         ax_dis = 5.0e-4 - ((float(ntab2) - 6.0) * 0.5e-4)
      endif
      if ntab2 <= 6 then
         ax_str  = ax_dis * k_oa
         ax_strt = ax_str
      endif
      if ntab2 > 6 then
         if ntab2 < 10 then
            ax_str = ax_strt - (((float(ntab2) - 6.0) * 0.5e-4) * k_ab)
         endif
      endif
      if ntab2 >= 10 then
         ax_str  = ax_dis * k_bo
      endif
      table.x(2,ntab2) = ax_dis
      table.y(2,ntab2) = ax_str
   endloop
end
@ana_slip
```

```
;
fish def slip_load
   ntab = ntab + 1
   tot_str = 0.0
   n_z = 0
   x_z = 0.0
   loop n (1,10)
      x_z = 0.1 * float(n)
      iz = bl.zone.near(x_z,2.0)
      tot_str = tot_str + bl.zo.str.yy(iz)
      n_z = n_z + 1
   endloop
   p_ytp = bl.gp.near(0.5,2.0)
   y_disp = bl.gp.disp.y(p_ytp)
   ver_str = - tot_str / n_z
   table.x(1,ntab) = -y_disp
   table.y(1,ntab) = ver_str
end
table 1 add (0,0)
ret
```