

## TECHNICAL NOTE #1

---

**Title:** Lagrange Implementation of Coulomb Friction Using User Subroutine **FRIC**

**Date:** May 29, 2001

---

### Introduction

This technical note illustrates how to implement Coulomb friction in user subroutine **FRIC** using the Lagrange method. The two-dimensional problem is presented first. The logic of the friction algorithm is discussed without introducing complex partial differential equations. Once the friction logic is understood, it is straightforward to extend the two-dimensional formulation to a three-dimensional formulation, i.e., sliding is permitted in two directions.

The input files and example user subroutines have been updated and checked for ABAQUS Version 6.2.

### Analysis of the Model

The standard Coulomb friction model assumes that no relative motion occurs if the frictional stress,  $\tau$ , is less than the critical stress,  $\tau_{\text{crit}}$ , which is proportional to the contact pressure,  $p$ , i.e.,

$$\tau_{\text{crit}} = \mu p \quad (\text{Eq. 1})$$

where  $\mu$  is the coefficient of friction.

User subroutine **FRIC** will be called if the body has come into contact with the surface even though the contact pressure may be negative. Therefore, a check for this condition is included in the subroutine. If this condition is true, the current estimated value of the slip is checked. If this slip value is zero, the sticking condition is set and no further variables need be set. Otherwise, the slipping condition is set together with the other variables as shown in Listing 1.

ABAQUS sets **LM** to 2 when it calls this routine if the contact point was open in the previous iteration. If the friction coefficient is zero, the friction state is set to frictionless sliding by setting **LM** to 2 and exiting this routine. Otherwise, the friction state is set to sticking.

ABAQUS sets **LM** to 1 when it calls this routine if the contact point was sticking in the previous iteration. In this case, the value of the frictional stress,  $\tau$ , is passed to user subroutine **FRIC** via the variable **TAULM**. This value is then compared with Equation 1. If this value does not exceed  $\tau_{\text{crit}}$ , then the friction state remains sticking. Otherwise, the friction state is changed to slipping and the other variables are set properly.

Slip reversal is also checked by comparing the sign of the previous slip, **DSLIP**, to the current estimated slip value, **DGAM**. If slip reversal occurs, the friction state is set to sticking.

For the two-dimensional problem, Equation 1 must account for the direction of the frictional stress. It can be written as

$$\tau = \frac{\Delta\gamma}{|\Delta\gamma|} \mu p, \quad (\text{Eq. 2})$$

where  $\Delta\gamma$  is the incremental slip. Using Equation 2 we can compute the partial derivatives and obtain the following:

$$\frac{\partial \tau}{\partial p} = \frac{\Delta\gamma}{|\Delta\gamma|} \mu, \quad (\text{Eq. 3})$$

$$\frac{\partial \tau}{\partial \gamma} = 0. \quad (\text{Eq. 4})$$

In user subroutine **FRIC**, the coefficient in front of  $\mu$  in Equations 2 and 3 is coded using the FORTRAN intrinsic function **SIGN()**. There are situations where  $\Delta\gamma$  in Equations 2 and 3 is zero. In such cases, we can obtain the direction of the frictional stress from the sign of the variable **TAULM**.

For the three-dimensional problem, Equation 2 is modified due to the existence of another slip direction. For isotropic Coulomb friction the equation is

$$\tau_i = \frac{\Delta\gamma_i}{\Delta\gamma_{eq}} \mu p, \text{ for } i = 1, 2 \quad (\text{Eq. 5})$$

where

$$\Delta\gamma_{eq} = \sqrt{\Delta\gamma_1^2 + \Delta\gamma_2^2}. \quad (\text{Eq. 6})$$

Using Equation 5, we can compute the partial derivatives and obtain the following:

$$\frac{\partial \tau_i}{\partial p} = \frac{\Delta\gamma_i}{\Delta\gamma_{eq}} \mu \quad (\text{Eq. 7})$$

$$\frac{\partial \tau_i}{\partial \gamma_j} = \frac{\mu p}{\Delta\gamma_{eq}} \left( \delta_{ij} - \frac{\Delta\gamma_i}{\Delta\gamma_{eq}} \frac{\Delta\gamma_j}{\Delta\gamma_{eq}} \right). \quad (\text{Eq. 8})$$

If the state changes from sticking to slipping, the above equations cannot be used because the  $\Delta\gamma_i$  are zero. In this case we assume that the direction of the frictional stress is proportional to the direction of the variable **TAULM**, i.e.,

$$\tau_i = \frac{\hat{\tau}_i}{\hat{\tau}_{eq}} \mu p, \text{ for } i = 1, 2 \quad (\text{Eq. 9})$$

where

$$\hat{\tau}_{eq} = \sqrt{\hat{\tau}_1^2 + \hat{\tau}_2^2} \quad (\text{Eq. 10})$$

and the  $\hat{\tau}_i$  are the components of the variable **TAULM**. We also assume that the incremental slip is given by

$$\Delta \gamma_i^{sl} = \Delta \gamma_i = \frac{\hat{\tau}_i}{k_s}. \quad (\text{Eq. 11})$$

Differentiate Equations 5 and substitute the expression for  $d\hat{\tau}_i$  from Equation 11 to obtain the following:

$$\frac{\partial \tau_i}{\partial p} = \frac{\hat{\tau}_i}{\hat{\tau}_{eq}} \mu \quad (\text{Eq. 12})$$

$$\frac{\partial \tau_i}{\partial \gamma_j} = k_s \frac{\mu p}{\hat{\tau}_{eq}} \left( \delta_{ij} - \frac{\hat{\tau}_i}{\hat{\tau}_{eq}} \frac{\hat{\tau}_j}{\hat{\tau}_{eq}} \right). \quad (\text{Eq. 13})$$

The above equations are implemented for three-dimensional Coulomb friction using the Lagrange multiplier method as shown in Listing 2. The variable  $k_s$  should be chosen so that it is several orders of magnitude larger than the largest value of  $\mu p$  throughout the analysis. We have adopted an arbitrary value of  $10^6$  for  $k_s$  in Listing 2.

# Listing 1. Lagrange Implementation of Two-Dimensional Coulomb Friction

```

*HEADING
TECHNOTE_FRICTION_1: LAGRANGE IMPLEMENTATION OF 2-D COULOMB FRICTION
*RESTART,WRITE
*NODE
1,0.,0.
2,0.,1.0
100,0.,0.
*ELEMENT,TYPE=B21,ELSET=BEAM
1,1,2
*BEAM SECTION,SECT=CIRC,ELSET=BEAM,MATERIAL=ELAS
0.1,
*MATERIAL,NAME=ELAS
*ELASTIC
30.E6,0.3
*surface,type=node, NAME=CNS
1,
*rigid body,analytical surface=RIGS,REF NODE=100
*surface,TYPE=SEGMENTS,NAME=RIGS
START,-1.0,0.
LINE,200.,0.
*CONTACT PAIR, INTERACTION=INT1
CNS,RIGS
*SURFACE INTERACTION, NAME=INT1
1.0,
*FRICTION,USER,PROPERTIES=1
0.5,
*****
*STEP,NLGEOM,UNSYMM=YES
ESTABLISH CONTACT
*STATIC
1.,1.
*EL PRINT
S,E
*EL FILE,F=100
S,E
*CONTACT PRINT
CSTRESS,CDISP
*CONTACT FILE,F=100
CSTRESS,CDISP
*BOUNDARY
100,1,2,0.0
100,6,6,0.0
2,1,1,0.0
2,2,2,-1.5E-4
2,6,6,0.0
*PRINT,CONTACT=YES
*END STEP
*****
*STEP,NLGEOM
X SLIDE
*STATIC
.1,1.
*BOUNDARY,OP=NEW

```

```

100,1,2,0.0
100,6,6,0.0
2,1,1,4.0
2,2,2,-1.5E-4
2,6,6,0.0
*PRINT,CONTACT=YES
*END STEP
*****
*STEP,NLGEOM
X SLIDE (2)
*STATIC
.1,1.
*BOUNDARY,OP=NEW
100,1,2,0.0
100,6,6,0.0
2,1,1,0.0
2,2,2,-1.5E-4
2,6,6,0.0
*PRINT,CONTACT=YES
*END STEP
C
C PUT SUBROUTINE FRIC INTO SEPARATE FORTRAN FILE
C
      SUBROUTINE FRIC(LM,TAU,DDTDDG,DDTDDP,DSLIP,SED,SFD,
1      DDTDDT,PNEWDT,STATEV,DGAM,TAULM,PRESS,DPRESS,DDPDDH,
2      SLIP,KSTEP,KINC,TIME,DTIME,NOEL,CINAME,SLNAME,
3      MSNAME,NPT,NODE,NPATCH,COORDS,RCOORD,DROT,TEMP,
4      PREDEF,NFDIR,MCRD,NPRED,NSTATV,CHRLNGTH,PROPS,NPROPS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CINAME,SLNAME,MSNAME
      DIMENSION TAU(NFDIR),DDTDDG(NFDIR,NFDIR),DDTDDP(NFDIR),
1      DSLIP(NFDIR),DDTDDT(NFDIR,2),STATEV(*),
2      DGAM(NFDIR),TAULM(NFDIR),SLIP(NFDIR),
3      COORDS(MCRD),RCOORD(MCRD),DROT(2,2),TEMP(2),
4      PREDEF(2,*),TIME(2),PROPS(NPROPS)
C
      PARAMETER(ZERO=0.0D0,PRECIS=1.0D-14)
C
C          LAGRANGE IMPLEMENTATION OF
C          2-D COULOMB FRICTION
C          XMU IS THE COEFFICIENT OF FRICTION
C
      XMU = PROPS(1)
C
C          CHECK IF PRESSURE IS NON-POSITIVE
C
      IF (PRESS .LE. ZERO) THEN
C
C          PROVIDE THE DTAU/DP, ETC
C
      IF (ABS(DGAM(1)) .EQ. ZERO) THEN
C
C          IF PRESSURE IS NON-POSITIVE

```

```

C          AND NO SLIP, SET CONDITION
C          TO STICKING
C
C          LM = 1
C      ELSE
C          DDTDDG(1,1) = ZERO
C          DDTDDP(1)   = SIGN(XMU,DGAM(1))
C          TAU(1)      = ZERO
C          DSLIP(1)     = DGAM(1)
C          LM = 0
C      ENDIF
C      RETURN
C  ENDIF
C
C          COMPUTE FOR CRITICAL SHEAR STRESS
C          FOR COULOMB FRICTION
C
C      TAUCRIT = XMU * PRESS
C      IF (LM .EQ. 2) THEN
C
C          GAP IS OPEN IN PREVIOUS
C          INCREMENT
C
C          IF (XMU .GT. PRECIS) THEN
C
C              NONZERO FRICTION COEFFICIENT,
C              SET CONDITION TO STICKING
C
C              LM = 1
C          END IF
C      ELSE IF (LM .EQ. 1) THEN
C
C          STICKING IN PREVIOUS ITERATION
C
C          COMPARE WITH CRITICAL SHEAR STRESS
C
C          IF (ABS(TAULM(1)) .LE. TAUCRIT) THEN
C
C              CONDITION IS STILL STICKING
C
C              LM = 1
C          ELSE
C
C              CONDITION CHANGED FROM STICK TO SLIP
C
C              LM = 0
C              TAU(1) = SIGN(TAUCRIT,TAULM(1))
C              DDTDDG(1,1) = ZERO
C              DDTDDP(1) = SIGN(XMU,TAULM(1))
C              DSLIP(1) = DGAM(1)
C          ENDIF
C      ELSE
C
C          SLIPPING IN PREVIOUS ITERATION
C
C          CHECK FOR SLIP REVERSAL

```

```

C
    IF (DSLIP(1) .EQ. ZERO) THEN
C
C           THIS CONDITION OCCURS IF IT WAS STICKING
C           PRIOR TO SLIPPING IN PREVIOUS
C           ITERATION
C
        CHECK = ABS(DGAM(1))
    ELSE
        CHECK = DSLIP(1)*DGAM(1)
    ENDIF
    IF (CHECK .GT. ZERO) THEN
C
C           NO REVERSAL IN SLIP
C
        LM = 0
        TAU(1)      = SIGN(TAUCRIT,DGAM(1))
        DDTDDG(1,1) = ZERO
        DDTDDP(1)   = SIGN(XMU,DGAM(1))
        DSLIP(1)    = DGAM(1)
    ELSE
C
C           REVERSAL IN SLIP
C
        LM = 1
    ENDIF
ENDIF
RETURN
END

```

## Listing 2. Lagrange Implementation of Three-Dimensional Coulomb Friction

```

*HEADING
TECHNOTE_FRICTION_2: LAGRANGE IMPLEMENTATION OF 3-D COULOMB FRICTION
*RESTART,WRITE
*NODE
1,0.,0.,0.
2,0.,0.,1.0
100,0.,0.,0.
*ELEMENT,TYPE=B31,ELSET=BEAM
1,1,2
*BEAM SECTION,SECT=CIRC,ELSET=BEAM,MATERIAL=ELAS
0.1,
0.,1.,0.
*MATERIAL,NAME=ELAS
*ELASTIC
30.E6,0.3
*surface,type=node, NAME=CNS
1,
*rigid body,analytical surface=RIGS,REF NODE=100
*surface,TYPE=CYLINDER,NAME=RIGS
-100.,-100.,0.,0.,-100.,0.
-100.,0.,0.
START,0.,0.
LINE,200.,0.
*CONTACT PAIR, INTERACTION=INT1
CNS,RIGS
*SURFACE INTERACTION, NAME=INT1
1.0,
*FRICTION,USER,PROPERTIES=1
0.5,
*****
*STEP,NLGEOM,UNSYMM=YES
ESTABLISH CONTACT
*STATIC
1.,1.
*EL PRINT
S,E
*EL FILE
S,E
*CONTACT PRINT
CSTRESS,CDISP
*CONTACT FILE
CSTRESS,CDISP
*BOUNDARY
100,1,6,0.0
2,1,1,0.0
2,2,2,0.0
2,4,6,0.0
2,3,3,-1.5E-4
*PRINT,CONTACT=YES
*END STEP
*****
*STEP,NLGEOM
SLIDE IN X DIRECTION

```



```

*STATIC
0.1,1.
*BOUNDARY,OP=NEW
100,1,6,0.0
2,1,1,3.0
2,2,2,0.0
2,4,6,0.0
2,3,3,-1.5E-4
*PRINT,CONTACT=YES
*END STEP
*****
*STEP,NLGEOM
SLIDE IN Y DIRECTION
*STATIC
.1,1.
*BOUNDARY,OP=NEW
100,1,6,0.0
2,1,1,3.0
2,2,2,3.0
2,4,6,0.0
2,3,3,-1.5E-4
*PRINT,CONTACT=YES
*END STEP
*****
*STEP,NLGEOM
SLIDE BACK TO THE ORIGIN
*STATIC
.1,1.
*BOUNDARY,OP=NEW
100,1,6,0.0
2,1,1,0.0
2,2,2,0.0
2,4,6,0.0
2,3,3,-1.5E-4
*PRINT,CONTACT=YES
*END STEP
C
C PUT SUBROUTINE FRIC INTO SEPARATE FORTRAN FILE
C
      SUBROUTINE FRIC(LM,TAU,DDTDDG,DDTDDP,DSLIP,SED,SFD,
1      DDTDDT,PNEWDT,STATEV,DGAM,TAULM,PRESS,DPRESS,DDPDDH,
2      SLIP,KSTEP,KINC,TIME,DTIME,NOEL,CINAME,SLNAME,
3      MSNAME,NPT,NODE,NPATCH,COORDS,RCOORD,DROT,TEMP,
4      PREDEF,NFDIR,MCRD,NPRED,NSTATV,CHRLNGTH,PROPS,NPROPS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CINAME,SLNAME,MSNAME
      DIMENSION TAU(NFDIR),DDTDDG(NFDIR,NFDIR),DDTDDP(NFDIR),
1      DSLIP(NFDIR),DDTDDT(NFDIR,2),STATEV(*),
2      DGAM(NFDIR),TAULM(NFDIR),SLIP(NFDIR),
3      COORDS(MCRD),RCOORD(MCRD),DROT(2,2),TEMP(2),
4      PREDEF(2,*),TIME(2),PROPS(NPROPS)
C
      DIMENSION AN(2)

```

```

C
PARAMETER(ZERO=0.0D0,ASMALL=1.0D-27,PRECIS=1.D-14,XKS=1.D6)
C
C
C          LAGRANGE IMPLEMENTATION OF 3-D
C          COULOMB FRICTION
C          XMU IS THE COEFFICIENT OF FRICTION
C
XMU = PROPS(1)
C
C          CHECK IF PRESSURE IS NON-POSITIVE
C
IF (PRESS .LE. ZERO) THEN
C
C          PROVIDE THE DTAU/DP, ETC
C
GAMEQV = SQRT(DGAM(1)**2 + DGAM(2)**2)
IF (GAMEQV .EQ. ZERO) THEN
    LM = 1
ELSE
    DDTDDG(1,1) = ZERO
    DDTDDG(1,2) = ZERO
    DDTDDG(2,1) = ZERO
    DDTDDG(2,2) = ZERO
    DDTDDP(1)   = XMU * DGAM(1)/GAMEQV
    DDTDDP(2)   = XMU * DGAM(2)/GAMEQV
    TAU(1)      = ZERO
    TAU(2)      = ZERO
    DSLIP(1)    = DGAM(1)
    DSLIP(2)    = DGAM(2)
    LM = 0
ENDIF
RETURN
ENDIF
C
C          COMPUTE FOR CRITICAL SHEAR STRESS
C
TAUCRIT = XMU * PRESS
IF (LM .EQ. 2) THEN
C
C          GAP IS OPEN IN PREVIOUS
C          INCREMENT
C
IF (XMU .GT. PRECIS) THEN
C
C          NONZERO FRICTION COEFFICIENT,
C          SET CONDITION TO STICKING
C
    LM = 1
    END IF
ELSE IF (LM .EQ. 1) THEN
C
C          STICKING IN PREVIOUS ITERATION
C
C          COMPARE WITH CRITICAL SHEAR STRESS
C

```

```

    TAULEQV = SQRT(TAULM(1)**2 + TAULM(2)**2)
    IF (TAULEQV .LE. TAUCRIT) THEN
C
C
C
        CONDITION IS STILL STICKING

        LM = 1
    ELSE
C
C
C
        CONDITION CHANGED FROM STICK TO SLIP

        LM = 0
        AN(1)      = TAULM(1)/TAULEQV
        AN(2)      = TAULM(2)/TAULEQV
        TAU(1)     = TAUCRIT * AN(1)
        TAU(2)     = TAUCRIT * AN(2)
        COEFF      = XKS*TAUCRIT/TAULEQV
        DDTDDG(1,1) = COEFF*AN(2)*AN(2)
        DDTDDG(1,2) = -COEFF*AN(1)*AN(2)
        DDTDDG(2,1) = DDTDDG(1,2)
        DDTDDG(2,2) = COEFF*AN(1)*AN(1)
        DDTDDP(1)   = XMU * AN(1)
        DDTDDP(2)   = XMU * AN(2)
        DSLIP(1)    = TAULM(1)/XKS
        DSLIP(2)    = TAULM(2)/XKS
    ENDIF
ELSE
C
C
C
        SLIPPING IN PREVIOUS ITERATION

        DSLIPEQV = SQRT(DSLIP(1)**2 + DSLIP(2)**2)
        GAMEQV   = SQRT(DGAM(1)**2 + DGAM(2)**2)
        IF (DSLIPEQV .EQ. ZERO .OR. GAMEQV .EQ. ZERO) THEN
C
C
C
C
C
            THIS OCCURS IF IT WAS STICKING
            PRIOR TO SLIPPING IN PREVIOUS
            ITERATION

            CHECK = GAMEQV
        ELSE
C
C
C
            CHECK FOR REVERSAL

            DOTP = DSLIP(1)*DGAM(1) + DSLIP(2)*DGAM(2)
            DOTPN = DOTP/(DSLIPEQV*GAMEQV)
            IF (DOTPN .EQ. ZERO) THEN
                CHECK = ASMALL
            ELSE
                CHECK = DOTPN
            ENDIF
        ENDIF
    IF (CHECK .GT. ZERO) THEN
        LM = 0
        CONST = TAUCRIT/GAMEQV
        AN(1) = DGAM(1)/GAMEQV
        AN(2) = DGAM(2)/GAMEQV
    
```

```

        TAU(1)      = TAUCRIT * AN(1)
        TAU(2)      = TAUCRIT * AN(2)
        DDTDDG(1,1) = AN(2)*AN(2)*CONST
        DDTDDG(1,2) = -AN(1)*AN(2)*CONST
        DDTDDG(2,1) = DDTDDG(1,2)
        DDTDDG(2,2) = AN(1)*AN(1)*CONST
        DDTDDP(1)   = XMU * AN(1)
        DDTDDP(2)   = XMU * AN(2)
        DSLIP(1)    = DGAM(1)
        DSLIP(2)    = DGAM(2)
    ELSE
C
C                      REVERSAL IN SLIP
C
        LM = 1
    ENDIF
ENDIF
RETURN
END

```