ABAQUS/Explicit: Use in Spontaneouse Rupture Problems

Elizabeth Templeton

March 5, 2008

1 Code Overview

I use the commercial finite element package, ABAQUS/Explicit 6.7-1, a product of Dessault Systemes (www.simulia.com). For the spontaneous rupture problems, I use the 8-node linear brick, reduced integration elements with hourglass control called C3D8R.

2 Time Step

An Explicit time stepping procedure is used (ABAQUS usage: *DYNAMIC, EXPLICIT) with a time step of approximately $0.4\Delta x/C_p$.

3 Element and Mesh Size

The 8-noded elements are regular hexahedrons with side length of 100 m. The finite element mesh extends 10 km along the fault perpendicular direction

4 Depth-Dependent Stresses

For TPV8 and TPV9, the initial stresses are specified using the ABAQUS/Explicit geostatic initial stresses option (ABAQUS usage: *INITIAL, TYPE=GEOSTATIC). This option allows the vertical and horizontal principal stresses to be specified as a linear function of depth.

5 Absorbing Boundary Conditions

Absorbing boundary conditions are implemented with the ABAQUS infinite elements, CIN3D8. The infinite elements act as dampers and minimize the reflection of dilatational and shear wave energy back in to the finite element mesh. The infinite element formulation does not provide perfect transmission of energy out of the mesh except in the case of plane body waves impinging orthogonally on the boundary.

6 Split-Node Procedure

The ABAQUS/Explicit contact pair (ABAQUS USAGE: *CONTACT PAIR) formulation is used to model surface interaction along the fault. Within ABAQUS/Explicit, a split node contact procedure is available for modeling frictional slip between two surfaces. The user subroutine *vfric.f* can be used to implement friction laws with dependence on slip, slip-rate, as well as any user-defined state variables and to model unidirectional slip along the fault. Two contact surfaces define the fault, each with its own set of nodes, so that a duplicate set of nodes is defined along the fault, as shown in Figure 1, with nodal mismatch between the surface given by $\alpha\Delta x$, where $0 \leq \alpha \leq 1/2$. We use infinitessimal strain kinematics and consider changes in α due to slip to be negligible. The nodal momentum equations in the tangential (x) direction are given by:

$$m_j^{(1)}\ddot{x}_j^{(1)} = F_j^{(1)} - T_j \quad (1)$$

$$m_j^{(2)}\ddot{x}_j^{(2)} = F_j^{(2)} + (1-\alpha)T_j + \alpha T_{j+1} \quad (2)$$

where $F_j^{(1)}$ and $F_j^{(2)}$ are the nodal forces per unit thickness into the diagram due to the stresses $\{\sigma\}$ in adjoining elements on surfaces (1) and (2).

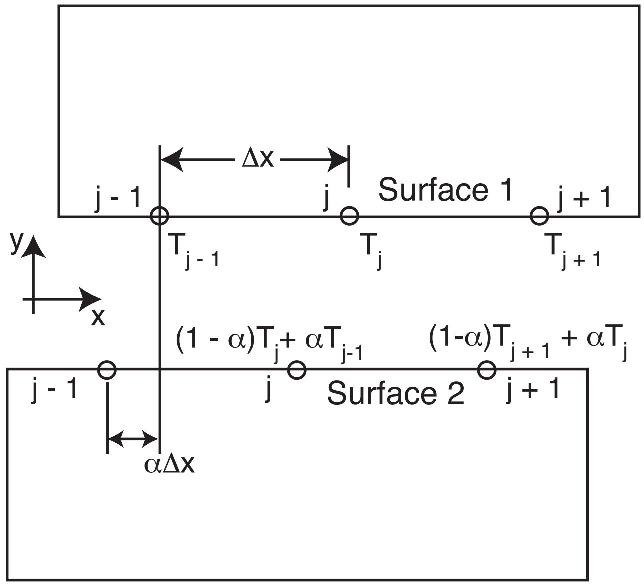

Figure 1: Schematic diagram showing node locations along the two contact surfaces defining the fault. The locations are taken as coincident ($\alpha = 0$) in the present analysis, based on linearized kinematics in the solid mechanics formulation.

T_j is the frictional force, due to the surface interactions, per unit thickness between the surfaces at each node, so that $T_j/\Delta x$ is the shear stress, τ_{xy}, along the contact surface. The nodal acceleration is given by $\ddot{x}_j$ and m_j is the mass per unit thickness at each node along the contact surface. We use meshes with uniform node spacing so that the mass at each node m along the contact surface is constant. The slip velocity between the contact surfaces, linearly interpolated, has time rate:

$$\dot{V}_j = \ddot{x}_j^{(1)} - (1-\alpha)\ddot{x}_j^{(2)} - \alpha\ddot{x}_{j-1}^{(2)} \tag{3}$$

which can be found using the nodal momentum equations:

$$\begin{aligned} m\dot{V}_j \;\; = \;\; & \left(F_j^{(1)} - (1-\alpha)F_j^{(2)} - \alpha F_{j-1}^{(2)}\right) \\ & -2\left[T_j + \alpha(1-\alpha)\left(\frac{T_{j+1}+T_{j-1}}{2} - T_j\right)\right] \end{aligned} \tag{4}$$

Since $\alpha(1-\alpha) < 1/4$, and the quantity within the final parentheses should approach zero for good mesh refinement, $\dot{V}_j$ can be approximated by:

$$m\dot{V}_j \approx F_j - 2T_j \tag{5}$$

where $F_j = F_j^{(1)} - (1-\alpha)F_j^{(2)} - \alpha F_{j-1}^{(2)}$. For uniform time steps, integrating Equation 5 gives the slip velocity during the next increment:

$$mV_j^{(t+\Delta t/2)} = mV_j^{(t-\Delta t/2)} + (F_j - 2T_j)\,\Delta t \tag{6}$$

The force per unit thickness to prevent slip during the next time step is:

$$T_j^{stop} = \frac{1}{2}\left(F_j + \frac{mV_j^{(t-\Delta t/2)}}{\Delta t}\right) \tag{7}$$

We use the split node procedure to implement the linear slip-weakening friction law. At each time step, the frictional strength of the fault per unit thickness, T_j^{fric}, is calculated at each node using the linear slip-weakening law:

$$T_j^{fric} = \begin{cases} (f_s - (f_s - f_d)\Delta u_j/D_c)\,(T_j^{norm}), & \Delta u_j < D_c \\ f_d(T_j^{norm}), & \Delta u_j > D_c \end{cases} \tag{8}$$

Where the T_j^{norm} are the nodal fault-normal forces per unit thickness, positive in compression, and f_s, f_d and D_c are the static and dynamic coefficients of friction and the critical slip-weakening distance. This spilt note contact procedure is implemented within the ABAQUS/Explicit user subroutine, VFRIC. At each node, the minimum of T_j^{fric} and T_j^{stop} is applied as the frictional force per unit thickness between the contact surfaces, T_j. Within the ABAQUS user subroutine, T_j^{stop} is provided under the name $fStickForce$ and the user must specify the frictional force between the contact surfaces, $fTangential$, or T_j. Within VFRIC we calculate the frictional strength of the fault based on the accumulated slip and to specify the the frictional force between the contact surfaces, T_j.

7 Sample *vfric.f*: Slip-weakening friction along a 2D planar fault with nucleation by a shear stress perturbation

```
C
C User subroutine VFRIC
      subroutine vfric (
C Write only -
     *     fTangential,
```

```
C Read/Write -
     *     statev,
C Read only -
     *     kStep, kInc, nContact, nFacNod, nSlvNod, nMstNod,
     *     nFricDir, nDir, nStateVar, nProps, nTemp, nPred, numDefTfv,
     *     jSlvUid, jMstUid, jConSlvid, jConMstid, timStep, timGlb,
     *     dTimPrev, surfInt, surfSlv, surfMast, lContType,
     *     dSlipFric, fStickForce, fTangPrev, fNormal, frictionWork,
     *     shape, coordSlv, coordMst, dircosSl, dircosN, props,
     *     areaSlv, tempSlv, preDefSlv, tempMst, preDefMst )
C
      include 'vaba_param.inc'
C
      dimension props(nProps), statev(nstateVar,nSlvNod),
     1     dSlipFric(nDir,nContact),
     2     fTangential(nFricDir,nContact),
     3     fTangPrev(nDir,nContact),
     4     fStickForce(nContact), areaSlv(nSlvNod),
     5     fNormal(nContact), shape(nFacNod,nContact),
     6     coordSlv(nDir,nSlvNod), coordMst(nDir,nMstNod),
     7     dircosSl(nDir,nContact), dircosN(nDir,nContact),
     8     jSlvUid(nSlvNod), jMstUid(nMstNod),
     9     jConSlvid(nContact), jConMstid(nFacNod,nContact),
     1     tempSlv(nContact), tempMst(numDefTfv),
     2     preDefSlv(nContact, nPred),
     3     preDefMst(numDefTfv, nPred)
C
      character*8 surfInt, surfSlv, surfMast
      parameter ( zero = 0.d0, one = 1.d0 )


C     user defined state variable, statev(nstatevar, nslvnod)
C     two state variables will be used in this analysis
C     statev(1, nslvnod) will define the slip
C     statev(2, nslvnod will define the velocity


C     Variables to be used in vfric:
```

```
C     slip is defined as slip = slip + dSlip, length ncontact
C     dSlipNslv is the same as dSlipFric, but with lenght nslvnod
C     jCon is a list of node numbers on slave surface in contact

      dimension slip(nContact)
      dimension dSlipNslv(nSlvNod)
      dimension jCon(nContact)


      double precision slip
      double precision dSlipNslv
      double precision jCon


C     Initialize variables
      dSlipNslv(:)=0
      slip(:)=0
      jCon(:)=0


C     Set values of friction coefficients and dc given by input file
C     These properties must be defined within the input file.

      mus = props(1)
      mud = props(2)
      dc = props(3)
      Lc = props(4)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C     Define an initial slip along to nucleate ruputre
C     Here, along a length L>L_c, a constant slip, Du>dc is applied
C            so that at time zero, the shear stress applied along the
C            nucleation patch with be tau_r

      if (kinc .eq. 0) then
          nucleationL = INT(1.25*Lc)
          Left = 501-nucleationL
```

```
          Right = 501+nucleationL

          do 15  k = Left, Right
             dSlipNslv(k) = dc*2
 15       end do
       end if

       if (kinc .eq. 0) then
          do 25 kSlv = 1, nSlvNod
            statev(1,kSlv) = statev(1,kSlv) + dSlipNslv(kSlv)
 25      end do
       end if

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


C      Update Slip at each time step
       do 35  kcon = 1, nContact
           jCon(kcon) = jSlvUid(JConSlvid(kcon))
           Slip(kcon) = statev(1,jConSlvid(kcon))+  dSlipFric(1,kcon)
           statev(surfnum,jConSlvid(kcon)) = Slip(kcon)
 35    end do


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC
CC                  Main Part of VFRIC
CC
CC    define the tangential force that should be applied to the nodes
CC    in contact based on slip weakenning law
CC
CC    if the nodes have not slipped yet, the tangential force is set
CC    to the peak shear force or the force required to prevent
CC    any acceleration at that node
CC
CC    if the nodes have slipped, but the slip is less than Dc, the
CC    critical slip displacement, the slip-weakening law is applied
CC
```

```
CC    if the slip is greater that Dc, then
CC
CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


C     define fTangential for slip from previous increment at nodes that
C     are in contact during this increment

      do 45 kcon = 1, nContact
             fn = fNormal(kcon)
             fs = fStickForce(kcon)
             fp = mus*fn
             fr = mud*fn
             mu= mud+(mus-mud)*(1-(Slip(kcon))/dc)
              if (Slip(kcon) .GT. dc) then
                 ft= min(fr,fs)
              else
                 ft= min(mu*fn,fs)
              end if

             fTangential(1,kcon)= -ft
 45     end do


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

*
      return
      end
```