APPENDIX G: User Defined Interface Friction and Conductivity

An easy-to-use user contact interface is provided in LS-DYNA where you can define the frictional coefficients (static and dynamic) as well as contact heat transfer conductance as functions of contact pressure, relative sliding velocity, separation and temperature. To be able to use this feature, an object version of the LS-DYNA code is required, and you must write your own Fortran (or C) code to define the contact parameters of interest.

The object version of LS-DYN comes with various text files in which user subroutines can be included. The subroutines of interest are

```
subroutine usrfrc(fstt,fdyn,...)
```

for defining the frictional coefficients fstt (static) and fdyn (dynamic) and

```
subroutine usrhcon(h,...)
```

for defining the heat transfer contact conductance, *h*. Subroutines `usrfrc` and `usrhcon` are dyn21cnt.F and dyn21.F. Due to the significant change in characteristics between the regular node-to-surface contact and segment-to-segment Mortar contact, the subroutine

```
subroutine mortar_usrfrc(…)
```

in dyn21cnt.F is intended for usage with Mortar contact. This routine is supposed to return the coefficient of friction for the current contact state and is in this sense very similar to the other routine. For explanation of all arguments, we refer to the source code.

We emphasize at this point that the user friction interface for non-Mortar differs between LS-DYNA (SMP) and MPP-DYNA (MPP), for reasons that have to do with how the contacts are implemented in general. In LS-DYNA (SMP) you are required not only to define the frictional coefficients but also to assemble and store contact forces and history, whereas in MPP-DYNA (MPP) only the frictional coefficients have to be defined. For Mortar contact, the same routine is called for both SMP and MPP, and only one frictional coefficient needs to be defined.

For the friction interface (SMP and MPP) you may associate history variables with each contact node (or segment for Mortar contact). Unfortunately, the user friction interface is currently not supported by all available contacts in LS-DYNA and MPP-DYNA, but it should cover the most interesting ones among others, *CONTACT_(FORMING_)NODES_TO_SURFACE, *CONTACT_(FORMING_)SURFACE_TO_SURFACE, *CONTACT_(FORMING_)ONE_WAY_SURFACE_TO_SURFACE and all Mortar contacts. Upon request by customers, additional contact types can be supported.

One of the arguments to the user contact routines is the curve array `crv`, also available in the user material interface. Note that when using this array, the curve identity must be converted to an internal number or the subroutine `crvval` may be utilized. For more information, see Appendix A on user materials.

For definition of user contact parameters, you must define the keywords

*USER_INTERFACE_FRICTION

or

*USER_INTERFACE_CONDUCTIVITY

The card format for these two keywords is identical and can be found in other sections in this manual.

There is an alternate route to defining the conductivity parameters for a user defined thermal contact. On the *CONTACT_..._THERMAL_FRICTION optional card the parameter FORMULA may be set to a negative number. This will automatically create a user defined conductivity interface and invoke reading of –FORMULA contact parameters immediately following the card including the FORMULA parameter. Note that FORMULA is related to NOC and NOCI in the *USER_INTERFACE_CONDUCTIVITY keyword as

$$-\text{FORMULA} = \text{NOC} = \text{NOCI}.$$

Note that the pressure is automatically computed for each user conductivity interface, meaning the keyword *LOAD_SURFACE_STRESS is not necessary.

A sample friction subroutine for the non-Mortar case is provided below for SMP, for Mortar contact we refer to the source code for an example.

```
      subroutine usrfrc(nosl,time,ncycle,dt2,intr,areat,xt,yt,zt,
     . lsv,ix1,ix2,ix3,ix4,arear,xx1,xx2,xx3,stfn,stf,fni,
     . dx,dy,dz,fdt2,ninput,ua,side,iisv5,niisv5,n1,n2,n3,fric1,
     . fric2,fric3,fric4,bignum,fdat,iseg,fxis,fyis,fzis,ss,tt,
     . ilbsv,stfk,frc,numnp,npc,pld,lcfst,lcfdt,temp,temp_bot,
     . temp_top,isurface)
c
c*******************************************************************
c|  LIVERMORE SOFTWARE TECHNOLOGY CORPORATION  (LSTC)              |
c|  ------------------------------------------------------------   |
c|  COPYRIGHT © 1987-2007 JOHN O. HALLQUIST, LSTC                  |
c|  ALL RIGHTS RESERVED                                            |
c*******************************************************************
c
c     user subroutine for interface friction control
c
c     note:  LS-DYNA uses an internal numbering system to
c            accomodate arbitrary node numbering.  To access
c            information for user node n, address array location m,
c            m=lqf(n,1).  To obtain user node number, n,
c            corresponding to array address m, set n=lqfinv(m,1)
c
c     arguments:
c
c          nosl       =number of sliding interface
c          time       =current solution time
c          ncycle     =ncycle number
```

```
c          dt2        =time step size at n+1/2
c          intr       =tracked node array where the nodes are stored
c                      in ls-dyna internal numbering.  User numbers
c                      are given by function: lqfinv(insv(ii),1)
c                      for tracked node ii.
c          areat(ii)  =tracked node area (interface types 5&10 only) for
c                      tracked node ii
c          xt(ii)     =x-coordinate tracked node ii (projected)
c          yt(ii)     =y-coordinate tracked node ii (projected)
c          zt(ii)     =z-coordinate tracked node ii (projected)
c          lsv(ii)    =reference segment number for tracked node ii
c          ix1(ii), ix2(ii), ix3(ii), ix4(ii)
c                     =reference segment nodes in ls-dyna internal
c                      numbering for tracked node ii
c          arear(ii)  =reference segment area for tracked node ii.
c          xx1(ii,4)  =x-coordinates reference surface (projected) for
c                      tracked node ii
c          xx2(ii,4)  =y-coordinates reference surface (projected) for
c                      tracked node ii
c          xx3(ii,4)  =z-coordinates reference surface (projected) for
c                      tracked node ii
c          stfn       =tracked node penalty stiffness
c          stf        =reference segment penalty stiffness
c          fni        =normal force
c          dx,dy,dz   =relative x,y,z-displacement between tracked node and
c                      reference surface.  Multipling by fdt2 defines the
c                      relative velocity.
c          n1,n2,n3   =x,y, and z components of reference segment's normal
c                      vector
c
c***********************************************************************
c       frictional coefficients defined for the contact interface
c
c          fric1      =static friction coefficient
c          fric2      =dynamic friction coefficient
c          fric3      =decay constant
c          fric4      =viscous friction coefficient (setting fric4=0
c                      turns this option off)
c
c***********************************************************************
c
c          bignum     =0.0 for one way surface to surface and
c                      for surface to surface, and 1.e+10 for nodes
c                      to surface contact
c          ninput     =number of variables input into ua
c          ua(*)      =users' array, first ninput locations
c                      defined by user.  The length of this
c                      array is defined on control card 10.
c                      This array is unique to interface nosl.
c
c          side       ='surfb' for first pass.  the surfb
c                       surface is the surface designated in the
c                       input
c                     ='surfa' for second pass after surfa and
c                      surfb surfaces have been switched as the
c                      tracked and reference surface for the type 3
c                      symmetric interface treatment.
c
c          iisv5      =an array giving the pointers to the active nodes
c                      in the arrays.
c
c          niisv5     =number of active nodes
c
c          fdat       =contact history data array
c          iseg       =contact reference segment from previous step.
c          fxis       =tracked node force component in global x dir.
```

```
c                      to be updated to include friction
c          fyis       =tracked node force component in global y dir.
c                      to be updated to include friction
c          fzis       =tracked node force component in global z dir.
c                      to be updated to include friction
c          ss(ii)     =s contact point (-1 to 1) in parametric coordinates
c                      for tracked node ii.
c          tt(ii)     =t contact point (-1 to 1) in parametric coordinates
c                      for tracked node ii.
c          ilbsv(ii)  =pointer for node ii into global arrays.
c          stfk(ii)   =penalty stiffness for tracked node ii which was used
c                      to compute normal interface force.
c          frc(1,lsv(ii))
c                    =Coulomb friction scale factor for segment lsv(ii)
c          frc(2,lsv(ii))
c                    =viscous friction scale factor for segment lsv(ii)
c
c***********************************************************************
c       parameters for a coupled thermal-mechanical contact
c
c          numnp      = number of nodal points in the model
c          npc        = load curve pointer
c          pld        = load curve (x,y) data
c          lcfst(nosl)= load curve number for static coefficient of
c                       friction versus temperture for contact
c                       surface nosl
c          lcfdt(nosl)= load curve number for dynamic coefficient of
c                       friction versus temperture for contact
c                       surface nosl
c          temp(j)    = temperature for node point j
c          temp_bot(j)= temparature for thick thermal shell bottom
c                       surface
c          temp_top(j)= temparature for thick thermal shell top
c                       surface
c          numsh12    = number of thick thermal shells
c          itopaz(1)  = 999 ==> thermal-mechanical analysis
c          isurface   = thick thermal shell surface pointer
c
c*******************************************************************************
```