APPENDIX H: User-Defined Thermal Material Model

The addition of a thermal user material routine is fairly straightforward. The keyword *MAT_THERMAL_USER_DEFINED, described in Volume II, controls the thermal user material.

The thermal user material can be used alone or in conjunction with any given mechanical material model in a coupled thermal-mechanical solution. A heat source can be included and the specific heat updated so that it is possible to model, for instance, phase transformations including melt energy.

If, for the same part (shell or solid elements), both a thermal and mechanical user material model are defined, the two user material models have (optionally) read access to each other's history variables. If the integration points of the thermal and mechanical elements are not coincident, interpolation or extrapolation is used when reading history variables. Linear interpolation or extrapolation using history data from the two closest integration points is used in all cases except when reading history variables for thick thermal shells (THSHEL = 1 on *CONTROL_SHELL). For the thermal shell, the shape functions of the element are used for the interpolation or extrapolation.

The thermal user materials are thermal material types 11 through 15. File dyn21tumat.F contains these thermal user material subroutines as subroutines `thumat11`, …, and `thumat15`. Subroutine `thusrmat` calls these thermal user material subroutines. File dyn21tumat.F also includes the source code of subroutine `thusrmat`. The comments of subroutines `thusrmat`, `thumat12`, and `umat46` contain additional useful information. Subroutine `umat46` resides in source file dyn21umats.F.

Thermal history variables

Setting NVH on *MAT_THERMAL_USER_DEFINED greater than 0 enables using thermal history variables. Thermal history variables are output to the tprint file; see *DATABASE_TPRINT.

Interchange of history variables with mechanical user material

In a coupled thermo-mechanical solution, a corresponding thermal element exists for each mechanical shell, thick shell, or solid element. A pair consisting of a mechanical and a corresponding thermal element both have integration points and possibly history variables. The mechanical and thermal elements do not necessarily have the same number of integration points. By setting IHVE to 1 on *MAT_THERMAL_USER_DEFINED, a thermal user material model can read, but not write, the history variables from a mechanical user material model and vice versa.

If the locations of the points where the history variables are located differ between the mechanical and thermal elements, interpolation or extrapolation is used to calculate the history value. More information is available in the comments to the subroutines `thusrmat` and `thumat12`.

Limitations:

The thermal user material implementation has a few limitations. LS-DYNA will, in most cases, give an appropriate warning or error message when such a limit is violated. The limitations include:

1. Option IHVE = 1 is only supported for a limited range of mechanical elements:

   a) Solid elements: ELFORM = 1, 2, 10, 13.

   b) Shell elements: ELFORM = 2, 3, 4, 14, 15, 16. Note that user-defined integration rules are not supported.

2. Thermal history variables limitations:

   a) Thermal history variables are not output to d3plot.

3. The thermal solver includes not only the plastically dissipated energy as a heat source but also wrongly the elastic energy. The latter, however, is, in most cases, not of practical importance.

Example source code:

Example source code for thermal user material models is available in `thumat11` and `thumat12` as well as in `umat46`. Note that there is space for up to 64 material parameters in `r_matp` (the material parameter array), but only 32 can be read from the *MAT_THERMAL_USER_DEFINED card. The material parameters in `r_matp(i)` with $i = 41$ - 64, which are initially set to 0.0, may be used to store additional data.

Subroutines `curveval` and `curveval_v` evaluate load curves. Note that when using `curveval` and `curveval_v`, the load curves are first re-interpolated to 100 equidistant points. See the Load curves and tables subsection of Appendix A for more information on these subroutines.

Here, we provide example keyword input for a thermal user material. This input enables orthotropic conduation. The sample input has SI units.

```
*KEYWORD
   ⋮
```

```
*MAT_THERMAL_USER_DEFINED
$      MID        RO        MT       LMC       NVH      AOPT    IORTHO      IVHE
        21    7800.0      12.0       6.0       1.0       0.0       1.0       0.0
$       XP        YP        ZP        A1        A2        A3
       0.0       0.0       0.0       0.0       0.0       0.0
$       D1        D2        D3
       0.0       0.0       0.0
$  Materials parameter card.  C1 - C3 are the heat conductions in the 11, 22, and
$  33 directions of the material coordinate system.  HC is the heat capacity.
$  HSRC is a load curve giving heat source output as a function of time.  HCFAC
$  is the ID of a load curve giving a scale factor for the heat capacity as a
$  function of time.
$       C1        C2        C3        HC      HSRC     HCFAC
      25.0      25.0      20.0     470.0      11.0      12.0
   ⋮
*END
```

The source code for the user material is:

```
      subroutine thumat12(c1,c2,c3,cvl,dcvdtl,hsrcl,dhsrcdtl,
     1     hsv,hsvm,nmecon,r_matp,crv,
     2     nel,nep,iep,eltype,dt,atime,ihsrcl)
      character*(*) eltype
      dimension hsv(*),hsvm(*),r_matp(*),crv(101,2,*)
      include 'iounits.inc'
c
c     Thermal user-material number 12.
c
c     See comments at the beginning of subroutine thusrmat
c     for instructions.
c
c     Example: isotropic/orthotropic material with k1=P1 and
c     cvl=P2 for solid and shell elements including optional
c     change of heat capacity and a heat source, both functions
c     of time input as load curves.
c
c     Print out some info on start-up, use material parameter 64
c     as a flag.
      if(nint(r_matp(64)).eq.0) then
         r_matp(64)=1.
         write( *,1200)  (r_matp(8+i),i=1,6)
         write(iohsp,1200) (r_matp(8+i),i=1,6)
         write(59,1200) (r_matp(8+i),i=1,6)
      endif
c
c     Calculate response
      c1=r_matp(8+1)
      c2=r_matp(8+2)
      c3=r_matp(8+3)
      cvl=r_matp(8+4)
      dcvdtl=0.0
      eid=nint(r_matp(8+6))
      if(nint(eid).gt.0) then
         call curveval(eid,atime,cvlfac,tmp1)
         cvl=cvl*cvlfac
         dcvdtl=0.0
      endif
c
c     If flux or time step calculation then we are done.
      if(eltype.eq.'soliddt'.or.eltype.eq.'flux'.or.
     .     eltype.eq.'shelldt') return
      eid=nint(r_matp(8+5))
      if(nint(eid).gt.0) then
         ihsrcl=1
         call curveval(eid,atime,hsrcl,tmp1)
         dhsrcdtl=0.0
```

```
      endif
c
c     Update history variables
      hsv(1)=cvl
      hsv(2)=atime
      hsv(3)=hsv(3)+1.0
c
c     Done
      return
 1200 format(/'This is thermal user defined material #12. '/
     1        /' Material parameter c1-c3       : ',3E10.3
     2        /' Material parameter hc          : ',E10.3
     3        /' Heat source load curve         : ',F10.0
     4        /' hc scale factor load curve     : ',F10.0
     5        /' Thermal History variable 1     : cv'
     6        /' Thermal History variable 2-3   : Dummy'/)
      return
      end
```