zubaexy 3 minutes ago MgUMAT as obtained from Jeff on 22 June 2015

0 contributors

256 lines (218 sloc) 8.219 kB Raw Blame History

```
c=======================================================================
c=======================================================================
c  Calculate strength for basal slip
c-----------------------------------------------------------------------

      subroutine calc_str_bas(gambas, gamtw, epsl, temp, hbastype,
     & hbas1, hbas2, hbas3, hbas4, hbas5, hbas6, qbastw, qbassl, ysbas)

      implicit none

      !input
      double precision gambas, epsl, gamtw, temp
      integer hbastype
      double precision hbas1, hbas2, hbas3, hbas4, hbas5, hbas6
      double precision qbastw, qbassl
      dimension gamtw(6)

      !output
      double precision ysbas

      !util
      integer n
      double precision gamtwtot

      gamtwtot = 0.0d+0
      do n=1,6
        gamtwtot = gamtwtot + dabs(gamtw(n))
      end do

      if (hbastype.eq.1) then
c-----------------------------------------------------------------------
c     hbastype = 1: Power law hardening
c-----------------------------------------------------------------------
      !sigy = sig0 + B*(gambas + qbastw * gamtwtot + qbassl*epsl)^n
      !hbas1 = sig0
      !hbas2 = B
      !hbas3 = n
      !hbas4-6 = not used
        ysbas = hbas1 + hbas2*(dabs(gambas) + qbastw*gamtwtot +
     &    qbassl*dabs(epsl))**hbas3
      else if (hbastype.eq.2) then
c-----------------------------------------------------------------------
c     hbastype = 2: Chang and Kochmann
c-----------------------------------------------------------------------
      !sigy = tau0 + sig0(1-exp(-h*gam/sig0))
      !hbas1 = tau0
      !hbas2 = sig0
      !hbas3 = h
      !hbas3-6 = not used
```

```
        ysbas = hbas1+hbas2*(1.0d+0-dexp(-hbas3*dabs(gambas)/hbas2))
      else
        print*, 'BASAL STRENGTH TYPE ', hbastype, ' NOT IMPLEMENTED'
      end if

      return
      end


c=======================================================================
c=======================================================================
c  Calculate strength for tensile twinning
c-----------------------------------------------------------------------

      subroutine calc_str_tw(gambas, gamtw, epsl, temp, htwtype,
     & htw1, htw2, htw3, htw4, htw5, htw6, qtwbas, qtwsl, ystw, twcap,
     & captw)

      implicit none

      !input
      double precision gambas, epsl, gamtw, temp
      integer htwtype
      double precision htw1, htw2, htw3, htw4, htw5, htw6
      double precision qtwbas, qtwsl
      dimension gamtw(6)

      !output
      double precision ystw(6), twcap
      logical captw

      !util
      integer n
      double precision gamtwtot, PI

      PI = 3.14159265358979d+0
      gamtwtot = gamtw(1)+gamtw(2)+gamtw(3)+gamtw(4)+gamtw(5)+gamtw(6)

      if (htwtype.eq.1) then
c-----------------------------------------------------------------------
c     htwtype = 1: Bilinear - Graff et al., IJP 07
c-----------------------------------------------------------------------
c     if gamtwtot < gamthresh
c       tau^a = tau0 + h0*(gamtwtot + qtwbas*gambas + qtwsl*epsl)
c     else
c       tau^a = tau0 + h0*( (gamtwtot/gamthresh)^(m-1)
c              + qtwbas*gambas + qtwsl*epsl)
c     htw1 = tau0
c     htw2 = h0
c     htw3 = gamthresh
c     htw4 = m (power law penalty exponent)
c     htw5-6 = not used

      if (gamtwtot.lt.htw3) then
        do n=1,6
          ystw(n) = htw1 + htw2*(qtwbas*gambas + gamtwtot
     &       + qtwsl*epsl)
        end do
       else
        do n=1,6
          ystw(n) = htw1 + htw2*((gamtwtot/htw3)**(htw4-1.0d+0)
     &       + qtwbas*gambas + qtwsl*epsl)
        end do

        twcap = htw3
```

```
        if (gamtwtot.ge.twcap) then
          captw = .true.
        else
          captw = .false.
        end if
      end if

      else if (htwtype.eq.2) then
c-----------------------------------------------------------------------
c     htwtype = 2: tangent hardening with taylor approx for twin, no
c                  other latent hardening from other defm modes
c-----------------------------------------------------------------------
c     tau^a = tau0 + h0*tan(pi*gamtwtot/(2.0*frac*gamthresh))
c     htw1 = tau0
c     htw2 = h0
c     htw3 = gamthresh
c     htw4 = frac (0<=frac<=1) shifts cutoff max gamthresh
c     htw5-6 = not used
        twcap = htw3*htw4
        do n=1,6
          ystw(n) = htw1 + htw2*dtan(gamtwtot*PI/(2.0d+0*htw3))
        end do
        if (gamtwtot.ge.twcap) then
          captw = .true.
        else
          captw = .false.
        end if

      else if (htwtype.eq.3) then
c-----------------------------------------------------------------------
c     htwtype = 3: Chang and Kochmann
c-----------------------------------------------------------------------
c     htw1 = tau0
c     htw2 = h1
c     htw3 = h2
c     htw4 = gamthresh
c     htw5 = frac (0<=frac<=1) shifts cutoff max gamthresh
c     htw6 = not used
        twcap = htw4*htw5
        do n=1,6
          ystw(n)=htw1+(htw2*gamtw(n)+htw3*(gamtwtot-gamtw(n)))
     &     / htw4
        end do
        if (gamtwtot.ge.twcap) then
          captw = .true.
        else
          captw = .false.
        end if
      else
        print*, 'TWIN STRENGTH TYPE ', htwtype, ' NOT IMPLEMENTED'
      end if

      return
      end

c=======================================================================
c=======================================================================
c  Calculate strength for non-basal slip
c-----------------------------------------------------------------------

      subroutine calc_str_sl(bis, gambslip, gamtw, epsl, epdsl, temp,
     & tempmelt,hsltype, hsl1, hsl2, hsl3, hsl4, hsl5, hsl6, hsl7, hsl8,
     & qslbas, qsltw, yssl, dyysdepd)

      implicit none
```

```
180
181         !input
182         logical bis
183         double precision gambslip, epsl, epdsl, gamtw, temp, tempmelt
184         integer hsltype
185         double precision hsl1, hsl2, hsl3, hsl4, hsl5, hsl6, hsl7, hsl8
186         double precision qslbas, qsltw
187         dimension gamtw(6)
188
189         !output - yield strength, and deriv of ys wrt strain rate
190         double precision yssl, dyysdepd
191
192         !util
193         double precision thom
194
195   c     IF MELTED, SET YSSL = 0, OTHERWISE CALC THOM
196         if (temp.ge.tempmelt) then
197           yssl = 0.0d+0
198         else
199           thom = (temp-293d+0)/(tempmelt-293d+0)
200           if (hsltype.eq.1) then
201   c-----------------------------------------------------------------------
202   c       hsltype = 1: Johnson-Cook, no latent from other defm modes
203   c                    uses a min epdsl (str rate) of props(6)
204   c-----------------------------------------------------------------------
205   c       sigy = (A+B*epsl**n)(1+C*ln(epdsl))(1-((t-t0)/(tmelt-t0))**m)
206   c       hsl1 = A
207   c       hsl2 = B
208   c       hsl3 = n
209   c       hsl4 = C
210   c       hsl5 = m
211   c       hsl6-7 = not used
212   c       hsl8 = cutoff strain rate
213
214             if (epdsl.le.hsl8) then
215               epdsl = hsl8
216             end if
217             yssl=(hsl1+hsl2*dabs(epsl)**hsl3)*(1.0d+0+hsl4*dlog(epdsl))
218        &      *(1.0d+0-thom**hsl5)
219
220             if (bis) return
221
222             dyysdepd=hsl4*(1.0d+0-thom**hsl5)*(hsl1+hsl2*dabs(epsl)**hsl3)
223             dyysdepd = dyysdepd / epdsl
224
225           else if (hsltype.eq.2) then
226   c-----------------------------------------------------------------------
227   c       hsltype = 2: Chang and Kochmann
228   c-----------------------------------------------------------------------
229   c       sigy = (tau0+sig0*(1-exp(-h1*eps/sig0))+h2*eps)*(edot/edot0)**m
230   c       hsl1 = tau0
231   c       hsl2 = sig0
232   c       hsl3 = epd0
233   c       hsl4 = m
234   c       hsl5 = h1
235   c       hsl6 = h2
236   c       hsl7 = not used
237   c       hsl8 = cutoff strain rate
238
239             if (epdsl.le.hsl8) then
240               epdsl = hsl8
241             end if
242             yssl=(hsl1+hsl2*(1.0d+0-dexp(-hsl5*epsl/hsl2))+hsl6*epsl)*
243        &      (epdsl/hsl3)**hsl4
244
245             if (bis) return
```

```
246             dyysdepd = hsl4/hsl3*(epdsl/hsl3)**(hsl4-1.0d+0)*
247      &        (hsl1+hsl2*(1.0d+0-dexp(-hsl5*epsl/hsl2))+hsl6*epsl)
248           else
249             print*, 'NB SLIP STRENGTH TYPE ', hsltype, ' NOT IMPLEMENTED'
250           end if
251
252         end if
253
254         return
255         end
```