CS_DE.R

Lampe

Wed Aug 20 21:54:52 2014

```r
# ==== DIFFERENTIAL EQUATION ====
strain_Rates.01 <- function(Time, State, Parm){
  with(as.list(c(State, Parm)),{
    # function for calculating axial and lateral strain rates
    # Input must be in vector or matrix form, no data frames
    # Eqns. referenced from:  SAND97-2601
    # CPar: EAT0, ETA1, ETA2, NF, AA1, PP, NSP, R1, R3, R4, QSR
    # FPar: KAP0, KAP1, KAP2, NK, DDT
    # TestData:

    # ---- Flow Potential Parameters (5) *KAP2 HELD CONST. ----
    #   KAP0  <- as.numeric(FPar[1])
    #   KAP1    <- as.numeric(FPar[2])
    #   KAP2    <- as.numeric(FPar[3])  # Constant = 1
    #   DDT     <- as.numeric(FPar[4])
    #   NK      <- as.numeric(FPar[5])
    # ---- Creep Consolidation Parameters (11) *ETA2 HELD CONST
    #   ETA0   <- as.numeric(CPar[1])
    #   ETA1    <- as.numeric(CPar[2])
    #   ETA2    <- 1 #as.numeric(CPar[3])   # Constant = 1
    #   NF      <- as.numeric(CPar[4])
    #   AA1     <- as.numeric(CPar[5])
    #   PP      <- as.numeric(CPar[6])
    #   NSP     <- as.numeric(CPar[7])
    #   R1      <- as.numeric(CPar[8])
    #   R3      <- as.numeric(CPar[9])
    #   R4      <- as.numeric(CPar[10])
    #   QSR     <- as.numeric(CPar[11])

    # ============= parameters hard coded into function directly ========
    KAP0 <- 10.119
    KAP1 <- 1.005
    DDT  <- 0.896
    NK   <- 1.331
    KAP2 <- 1

    ETA0  <- 0.102854        # -
    ETA1  <- 3.9387          # -
    ETA2    <- 1               # constant -
    NF      <- 3.5122          # -
    AA1     <- 0.3147          # -
    PP      <- 1.6332          # -
    NSP     <- 0.557621        # -
    R1      <- 1.041 * 10 ^ -6 # [K/(MPa-sec)]
    R3      <- 15.1281         # -
    R4      <- 0.1677765       # -
    QSR     <- 1077.46         # [K]
```

```r
# ---- Munson-Dawson Creep Parameters (17) ----
A1      <- 8.386e22
A2      <- 9.672e12
Q1R     <- 12581
Q2R     <- 5033
N1      <- 5.5
N2      <- 5.0
B1      <- 6.0856e6
B2      <- 3.034e-2
Q       <- 5335
S0      <- 20.57
M       <- 3
K0      <- 6.275e5
C       <- 9.198e-3
ALPHA <- -17.37
BETA    <- -7.738
DELTA <- 0.58
MU      <- 12400

# ==== parameters loaded into function =========
# ---- Values input into function (18)----
#   ICASE   <- as.numeric(parameters[,1])   # TEST TYPE (1:Hyd Cons;2:Shear Cons;3:compaction)
#   ITEST <- as.character(parameters[,2])# TEST ID
#   TIME    <- as.numeric(parameters[,3])   # TIME [SEC]
#   DT      <- as.numeric(parameters[,4])     # DELTA TIME [SEC]
#   TF      <- as.numeric(parameters[,5])   # TOTAL TEST TIME [SEC]
#   TEMP    <- as.numeric(parameters[,6])     # TEMP [K]
#   AS      <- as.numeric(parameters[,7])     # AXIAL STRESS [MPA]
#   LS      <- as.numeric(parameters[,8])     # LATERAL STRESS [MPA]
#   EVT     <- as.numeric(parameters[,9])     # TOTAL TRUE VOLUMETRIC STRAIN
#   EVC     <- as.numeric(parameters[,10])  # CREEP TRUE VOLUMETRIC STRAIN
#   EAT     <- as.numeric(parameters[,11])  # TOTAL TRUE AXIAL STRAIN
#   EAC     <- as.numeric(parameters[,12])  # CREEP TRUE AXIAL STRAIN
#   RHO     <- as.numeric(parameters[,13])  # CURRENT DENSITY [KG/M3]
#   D       <- as.numeric(parameters[,14])  # FRACTIONAL DENSITY
#   RHO0    <- as.numeric(parameters[,15])  # DENSITY AT THE START OF CONSOLIDATION (<RHOI)
#   RHOI    <- as.numeric(parameters[,16])  # DENSITY AT THE START OF CREEP
#   DD      <- as.numeric(parameters[,17])  # AVERAGE GRAIN SIZE [MM]
#   W       <- as.numeric(parameters[,18])  # WATER CONENT BY PERCENT WEIGHT

# ---- fitting assumptions ----
RHOIS <- 2160.0   # ASSUMED IN SITU SALT DENSITY
DSP     <- 0.64     # FRACTIONAL DENSITY OF RANDOM DENSE SPHERICAL PARTICLES

# ---- interpolated input variables ----
TIME <- time.interp(Time)
TEMP <- temp.interp(Time)
AS   <- as.interp(Time)
LS   <- ls.interp(Time)
D    <- d.interp(Time)

# ---- calculate variables ----
MS  <- (2.0 * LS + AS) / 3  # MEAN STRESS
```

```
  DS  <- LS - AS                    # STRESS DIFFERENCE
  #   ELC <- (EVC - EAC) / 2        # CREEP TRUE LATERAL STRAIN
  D0  <- 1382.4 / RHOIS             # EMPLACED FRACTIONAL DENSITY (0.64 FRAC DENSITY)
  DI  <- RHOI / RHOIS              # INITIAL FRACTIONAL DENSITY

  # ==== this portion has been moved to lambda <- function() =====
  #WT1 <- DT / NTIME    # WEIGHTING FUNCTION FOR CREEP CONSOLIDATION PARAMETERS
  #WT     <- 1                      # WEIGHTING FUNCTION FOR FLOW PARAMETERS
  # ==============================================================

 #   Z1    <- yini[1]  # Predicted axial strain (initial values)
 #   Z2    <- yini[2] # Predicted lateral strain (initial values)
 #   Z3    <- yini[3] # internal variable "xi" for the transient function (FU)
  # integral of Eqn 2-27, (initial values)

  # ==== define the differential equation ====
#     DZ <- ifelse(cbind(TIME > 0, TIME > 0, TIME > 0),{
  VOL     <- Z1 + 2*Z2                 # VOLUMETRIC STRAIN
  VOLT    <- VOL + log(DSP/DI)     # USED FOR INITIAL ESTIMATE OF VOLUMETRIC STRAIN
  #DEN        <- DI/exp(VOL)           # CURRENT FRACTIONAL DENSITY
  DEN   <- D                    # CURRENT FRACTIONAL DENSITY

#     ifelse(D >= 1,{
#       MD <- 0  # if fractional density is 1, disclocation creep = 0
#       SP <- 0},# if fractional density is 1, pressure solutioning = 0
#     {VAR <- ifelse(DEN <= DDT, DDT, DEN) # DEFINE DENSITY CEILING ISH

  VAR <- ifelse(DEN <= DDT, DDT, DEN) # DEFINE DENSITY CEILING ISH

   # ---- Equivalent Stress ----
   OMEGAA     <- ((1 - DEN) * NF / (1 - (1 - DEN)^(1/NF))^NF)^(2/(NF + 1))
   OMEGAK     <- ((1 - VAR) * NK / (1 - (1 - VAR)^(1/NK))^NK)^(2/(NK + 1))
   ETA        <- ETA0 * OMEGAA^ETA1
   KAP        <- KAP0 * OMEGAK^KAP1
   TERMA  <- ((2 - DEN)/DEN)^((2 * NF)/(NF + 1))
   TERMK  <- ((2 - DEN)/DEN)^((2 * NK)/(NK + 1))

   # ---- Eqn. 2-3 (SAND97-2601) ----
   # Equivalent stress measure for Disl. Creep and Press Sol'ing
   SEQF   <- sqrt(ETA * MS^2 + ETA2 * TERMA * DS^2)
   # Equivalent stress measure for Flow Potential
   SEQ        <- sqrt(KAP * MS^2 + KAP2 * TERMK * DS^2)

   # ---- Eqn. 2-17 (SAND97-2601) ----
   ALPHA2 <- KAP * MS / 3
   BETA2  <- KAP2 * TERMK * DS

   # ---- Eqn. 2-20, WithOUT dislocation creep and pressure solutioning ----
   F2A <-     (ALPHA2 - BETA2)/SEQ        # fit to axial strains
   F2L <- (ALPHA2 + 0.5 * BETA2)/SEQ  # fit to lateral strains
   F2V <-  3 * ALPHA2 / SEQ            # fit to volumetric strains

   # ==== START: equivalent inelastic strain rate form for dislocation creep ====
```

```
    # ---- Steady State Strain Rate Calc ----
    ES1 <- A1 * (SEQF / MU)^N1 * exp(-Q1R/TEMP)    # Dislocation climb - Eqn. 2-30
    ES2 <- A2 * (SEQF / MU)^N2 * exp(-Q2R/TEMP)    # Undefined Mechanism - Eqn. 2-31

    # Slip - Eqn. 2-32 (SAND98-2601)
    H   <- SEQF - S0                               # HEAVISIDE FUNCTION
    ARG <- Q * (SEQF - S0) / MU
    ES3 <- ifelse(H > 0, 0.5 * (B1 * exp(-Q1R / TEMP) +
                              (B2 * exp(-Q2R / TEMP)) *
                              (exp(ARG) - exp(-ARG))),0)

    ESS = ES1 + ES2 + ES3 # Steady-state strain rate, Eqn. 2-29 (SAND97-2601)

    # ---- EVALUATE TRANSIENT FUNCTION, 3 branches: work hardening, equilibrium, recovery
    EFT  <- K0 * exp(C * TEMP) * (SEQF / MU) ^ M  # Transient Strain Limit, Eqn. 2-28
    BIGD <- ALPHA + BETA * log10(SEQF / MU)       # Work-Hardening parameter, Eqn 2-28

   # ---- add an event function ----

    FU <- ifelse(Z3 == EFT, 1, ifelse(Z3 < EFT, exp(BIGD * (1 - Z3 / EFT) ^ 2),
                                  exp(-DELTA * (1 - Z3 / EFT) ^ 2)))

    MD <- FU * ESS  # equivalent inelastic strain rate form for dislocation creep, Eqn 2-23

    # ==== START: Equivalent Inelastic Strain Rate Form for Pressure Solutioning ====
    # ---- Calculate initial volumetric strain - Based on spherical packing ----
    CR <- abs(exp(VOLT) - 1)

    # ---- Determine functional form - either large or small strains, Eqn 2-34 ----
    GAMMA <- ifelse(CR <= 0.15, 1, abs((D0 - exp(VOLT)) / ((1 - D0) * exp(VOLT))) ^ NSP)
    # Small Strains (Vol Strain > - 15%)
    # Large Strains (Vol Strain < - 15%)

    # ---- component of eqn 2-35 ---
    X3 <- exp((R3 - 1) * VOLT) / (abs(1 - exp(VOLT))) ^ R4

    # ---- determine value of moisture function (w) ----
    M2 <- ifelse (W == 0, 0, W ^ AA1)

    # ---- Equivalent Inelastic Strain Rate Form for Pressure Solutioning, Eqn 2-35
    G2 <- 1 / DD ^ PP # calculate grain size function
    T2 <- exp(-QSR / TEMP) / TEMP
    SP <- R1 * M2 * G2 * T2 * X3 * GAMMA * SEQF #})

   DZ1 <- (MD + SP) * F2A # derivative: axial strain rate
   DZ2 <- (MD + SP) * F2L # derivative: lateral strain rate
   DZ3 <- (FU - 1) * ESS  # derivative of internal variable "xi"

 #   cbind(DZ1, DZ2, DZ3)},{cbind(0,0,0)})

   return(list(c(DZ1, DZ2, DZ3)))
 })
}
```