6.15 Example O: FEAMAC Analysis

Sample Input Files For **FEAMAC** Analysis

Problem Summary:

Load Type:	Thermomechanical; Load Control (**ABAQUS**)
Load History Data:	From **ABAQUS**: STEP 1: Cool down Temperature: 860.0 to 427.0^{o}C Strain: 0.0 - 0.0 STEP 2: Transverse loading Temperature held fixed at 427.0^{o}C Strain: 0.0 to 0.015
Micromechanics model:	Double Periodic
Fiber Packing Arrangement:	Square Fiber (idp=2) Triangular Pack at 33% fiber volume ratio Circular Fiber (idp=6) Rectangular Pack at 33% fiber volume ratio
Integration Algorithm:	Forward Euler (**ABAQUS**)
Constituent Material Model:	Fiber: Elastic (ncmd = 7) Matrix: TGVIPS (ncmd = 7)
Constituents:	Fiber: SCS-6 Matrix: Ti-6-4 (isotropic)
Debonding:	First implementation of debond model
Finite Element Model:	Consists of 2 (8-noded brick) elements each with a different material. In this way the example demonstrates the use of multiple material groups.

☞ **FEAMAC Note:** All data that is struck-out in the listing is not required by **FEAMAC** and is ignored if present. Warnings are generated informing the user that the data provided is being ignored. **FEAMAC** can be employed with any size of a finite element mesh. However, **FEAMAC** requires that an appropriate amount of state variable space be allocated for each user material found in the finite element model. The amount of state variable space to be allocated for a given user material is based on the size of the RVE requested in the corresponding **MAC/**

GMC input deck. The number of state variables must be less than 1000. The ***DEPVAR** option under each of the ***USER MATERIAL** sections should be used to define the size of this space. The number of CONSTANTS should be set to one on the ***USER MATERIAL** line and on the following line the value should be set to one. If the material model is linear elastic and no micro curve data files have been requested then no state variable space is required for that material.

To determine the amount of space to request use the following equation:

NG = Number of cells in the gamma direction

NB = Number of cells in the beta direction

NDEPVAR = Number of state variables to be to be provided to the *DEPVAR option.

$$NDEPVAR = 46 + (42 * NG * NB) \quad < 1000$$

Therefore, based on the predefined RVE's in **section 4.2.13**:

IDP	NDEPVAR	
1	214	
2	718	
3	718	
4	718	
6	2104	Too Large: Must use alternate method described below
9	2104	Too Large: Must use alternate method described below
11	214	
13	28,438	Too Large: Must use alternate method described below

EXAMPLE:

```
...
*MATERIAL,NAME=MACDECK1
*USER MATERIAL, CONSTANTS=1
1
*DEPVAR
214
...
```

Where MACDECK1 references an IDP of size 2x2.

ALTERNATE METHOD:

If a chosen IDP requires more than 1000 state variables then the following alternate procedure must be followed:

On the *USER MATERIAL line set the number of CONSTANTS equal to 1and on the following line set the value of this constant to be two. Also set the number of state variables on the *DEPVAR option to zero.

```
EXAMPLE:
...
*MATERIAL,NAME=MACDECK2
*USER MATERIAL, CONSTANTS=1
2
*DEPVAR
0
...
```

Where MACDECK2 references an IDP that would require more than 1000 state variables (e.g. IDP = 9).

☞ **FEAMAC Note:** This alternate procedure will cause execution times to increase greatly.

☞ **FEAMAC Note:** It is highly recommended that linear elastic materials with no requests for micro curve data be used in all areas of the model except in areas of suspected material nonlinearity, irrespective of memory approach.

MAC/GMC Input Listings: MACMAN1:

```
FEAMAC DEMO - Material group No. 1, identified in ABAQUS deck as MACMAN1
*PRINT
  NPL=1 %
*LOAD
  LCON=3 LOP=3 LSS=1 %
*MECH
  NPTW=3 TI=0.0, 58000.0, 58150.0 LO=0.0, 0.0, 0.015 %
*THERM
  NPTT=3 TI=0.0, 58000.0, 58150.0 TE=860.0, 427.0, 427.0 %
*MODEL
  MOD=1 %
*SOLVER
  NTF=1 &
  NPTS=3 TIM=0.0, 58000.0, 58150.0 STP=2000.0, 1.0 %
*FIBER
  NFIBS=1
  NF=1 MF=4 NDPT=1 MAT=U IFM=1 &
  EL=58.E3, 0.32, 3.5E-06 &
  VI=0.8E-9,1.E20,0.1E-5,0.,0.85E-4,0.05,1.,1.,1.,3.3,1.8,1.35,1.,0.01 %
*MATRIX
  NMATX=1
  NM=1 MM=7 NDPT=2 MAT=A D=1.0, 1.0, 1.0 %
*MRVE
  IDP=2 VF=0.33 %
*DEBOND
 NII=2
 DBCH=1 NBI=1 NGI=2 FACE=2 RN=0.1 BDN=35.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=1 NGI=3 FACE=2 RN=0.1 BDN=35.0 RS=0.0 BDS=0.0 TI=58000.0
 %
*CURVE
  NP=1 %
*MACRO
  NT=1
  NC=1 X=3 Y=9 NAM=macman1_s3_e3n %
*END
```

MAC/GMC Input Listings: MACMAN2:

```
FEAMAC DEMO - Material group No. 2, identified in ABAQUS deck as MACMAN2
*PRINT
  NPL=1 %
*LOAD
  LCON=3 LOP=3 LSS=1 %
*MECH
  NPTW=3 TI=0.0, 58000.0, 58150.0 LO=0.0, 0.0, 0.015 %
*THERM
  NPTT=3 TI=0.0, 58000.0, 58150.0 TE=860.0, 427.0, 427.0 %
*MODEL
  MOD=1 %
*SOLVER
  NTF=1 &
  NPTS=3 TIM=0.0, 58000.0, 58150.0 STP=2000.0, 1.0 %
*FIBER
  NFIBS=1
  NF=1 MF=9 NDPT=2 MAT=A D=1.0, 0.0, 0.0 %
*MATRIX
  NMATX=1
  NM=1 MM=7 NDPT=2 MAT=A D=1.0, 1.0, 1.0 %
*MRVE
  IDP=6 VF=0.33 R=0.75%
*DEBOND
 NII=10
 DBCH=1 NBI=4 NGI=1 FACE=2 RN=0.1 BDN=14.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=4 NGI=6 FACE=2 RN=0.1 BDN=14.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=2 NGI=3 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=2 NGI=4 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=3 NGI=2 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=3 NGI=5 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=6 NGI=3 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=6 NGI=4 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=5 NGI=2 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=5 NGI=5 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
#
*CURVE
  NP=1 %
*MACRO
  NT=1
  NC=1 X=3 Y=9 NAM=macman2_s3_e3n %
*END
```

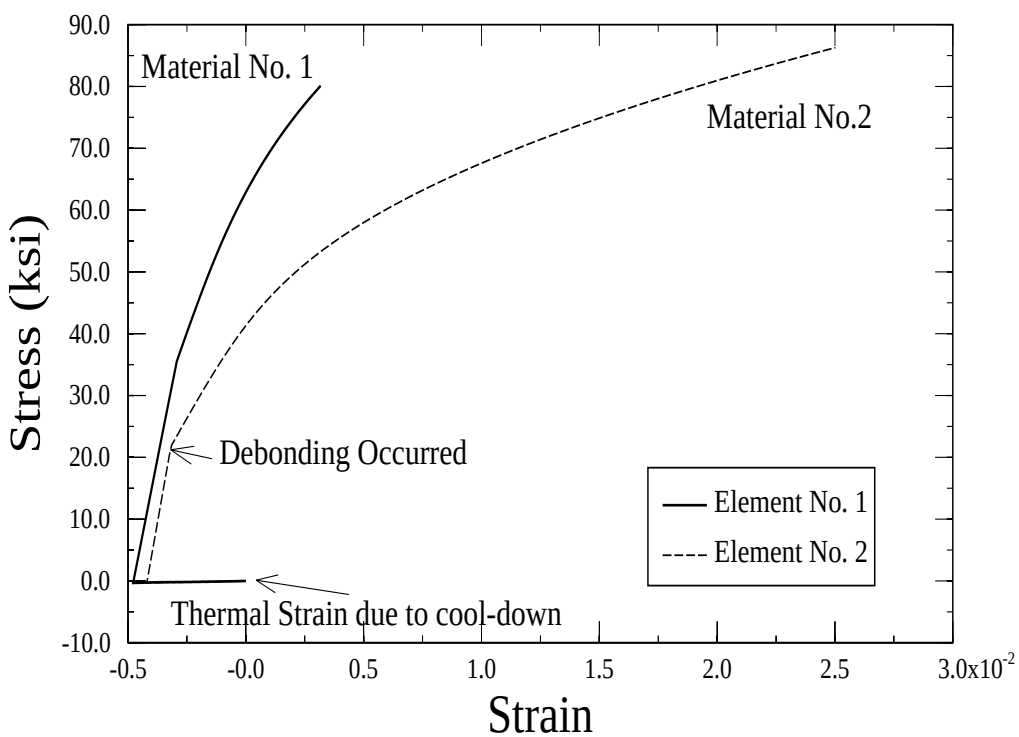
ABAQUS 5.8 Input Listing:

```
*HEADING
 VERIFICATION OF FEAMAC
*NODE,NSET=ALL
1, 0.0, 0.0, 0.0
2, 0.0, 1.0, 0.0
3, 0.0, 1.0, 1.0
4, 0.0, 0.0, 1.0
5, 1.0, 0.0, 0.0
6, 1.0, 1.0, 0.0
7, 1.0, 1.0, 1.0
8, 1.0, 0.0, 1.0
9, 2.0, 0.0, 0.0
10, 2.0, 1.0, 0.0
11, 2.0, 1.0, 1.0
12, 2.0, 0.0, 1.0
*ORIENTATION, NAME=TRANS
0.0, 0.0, 1.0, 0.0, 1.0, 0.0
*ELEMENT, TYPE=C3D8, ELSET=EL1
1, 1, 2, 3, 4, 5, 6, 7, 8
*ELEMENT, TYPE=C3D8, ELSET=EL2
2, 5, 6, 7, 8, 9, 10, 11, 12
*SOLID SECTION, MATERIAL=MACMAN1, ELSET=EL1, ORIENTATION=TRANS
1.0
*SOLID SECTION, MATERIAL=MACMAN2, ELSET=EL2, ORIENTATION=TRANS
1.0
*MATERIAL,NAME=MACMAN1
*USER MATERIAL, CONSTANTS=1
1
*DEPVAR
718
*USER DEFINED FIELD
*EXPANSION, TYPE=ANISO, USER
*MATERIAL,NAME=MACMAN2
*USER MATERIAL, CONSTANTS=1
2
*USER DEFINED FIELD
*EXPANSION, TYPE=ANISO, USER
*USER SUBROUTINE
*INCLUDE, INPUT=/home/smdant/macdev/feabench/man_demo/umacman.f
*INITIAL CONDITIONS, TYPE=TEMPERATURE
ALL, 860.0
*INITIAL CONDITIONS, TYPE=FIELD, VARIABLE=1
ALL, 1.0
**-------------------------------------------
*STEP, AMPLITUDE=RAMP, INC=2000000
```

```
 STEP ONE - COOL DOWN
*STATIC, DIRECT
2000.0, 58000.0
*TEMPERATURE
ALL, 427.0
*BOUNDARY
1, 1, 3
2, 1
2, 3
3, 1
4, 1, 2
5, 2, 3
6, 3
8, 2
9, 2, 3
10, 3,
12, 2
*NSET, NSET=ALL
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
*EL PRINT, FREQ=1000
S
*EL PRINT, FREQ=1000
E
*EL PRINT, FREQ=1000
THE, TEMP
*NODE PRINT, FREQ=1000, NSET=ALL
U
*NODE PRINT, FREQ=1000, NSET=ALL
CF
*NODE PRINT, FREQ=1000, NSET=ALL
RF
*END STEP
**-----------------------------------------
*STEP, AMPLITUDE=RAMP, INC=2000000
 STEP TWO - LOAD UP
*STATIC, DIRECT
1.0, 150.0
*BOUNDARY
1, 1, 3
2, 1
2, 3
3, 1
4, 1, 2
5, 2, 3
6, 3
8, 2
```

```
9, 2, 3
10, 3,
12, 2
9, 1, 1, 0.015
10, 1, 1, 0.015
11, 1, 1, 0.015
12, 1, 1, 0.015
*TEMPERATURE
ALL, 427.0
*END STEP
**-----------------------------------------
```

The following figure was obtained from the macro x-y plot data files produced by the present example.



ABAQUS UMAT Subroutine Listing:

```
C COMMON TO SOLIDS AND SHELLS
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678 2
C    __  _______ ____  __    __________
C   / / / / ___// __ \/ /   / ____/ __ \
C  / / / /\__ \/ / / / /   / /_  / / / /
C / /_/ /___/ / /_/ / /___/ __/ / /_/ /
C \____//____/_____/_____/_/   /_____/
C
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678 9
C


C ORDER OF ABAQUS FUNCTION CALLS
C
C                        - UMAT  (SOLID)
C         (if needed)  /
C USDFLD -=> UEXPAN -=<    or
C                      \
C                        - UGENS (SHELL) <-= TBD

C ------------------

      SUBROUTINE USDFLD(FIELD, STATEV, PNEWDT, DIRECT, T, CELENT,
     &     TIME, DTIME, CMNAME, ORNAME, NFIELD, NSTATV, NOEL, NPT,
     &     LAYER, KSPT, KSTEP, KINC, NDI, NSHR)


      INCLUDE 'ABA_PARAM.INC'

      PARAMETER (ZERO = 0.0D0, HALF = 0.5D0)
      PARAMETER (ONE = 1.0D0, TWO = 2.0D0, THREE = 3.0D0)
      PARAMETER (NFROM_UTHERM = 1, NFROM_UMECH = 2)

      CHARACTER*8 CMNAME, ORNAME
      DIMENSION FIELD(NFIELD), STATEV(NSTATV), DIRECT(3,3)
      DIMENSION T(3,3), TIME(2)

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH
      CHARACTER*80 TSTRING

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2

      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP

      LOGICAL SIGAL_STATV


      CHARACTER*8 FLGRAY(15)
      DIMENSION ARRAY(15), JARRAY(15)

      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
```

```
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION

      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR

      COMMON / KTHERMONLY / DTHERMSTRESS(6), DTHERMSTRAIN(6),
     &     DDTHERMSTRAIN_DT(6), THERMSTRESS(6), THERMSTRAIN(6)


      DATA NFIRSTELEM / 0 /
      DATA NFIRSTIP / 0 /
      DATA NFIRSTTIME / 1 /


      NFEA_FROM = 0


C*********************************
C INITIALIZE
C*********************************
      NNOEL = NOEL
      NNPT = NPT
      NNDI = NDI
      NNSHR = NSHR

      NCALL = 0

      IF (NFIRSTTIME .EQ. 1) THEN
         CALL FEAMAC_INIT(NOEL, NPT)
      ENDIF

      AB_PRINT = GAB_PRINT

C*********************************
C DEFINE MAC INPUT DECK NAME
C*********************************
      CALL FILL_STRING(TSTRING, ' ')
      CALL FILL_STRING(FNAME, ' ')

      CALL STRING_COPY(FNAME, CMNAME)
      CALL STRING_APPEND(FNAME, EXTSION)

      TSTRING = PATH
      CALL STRING_APPEND(TSTRING, FNAME)
      FNAME = TSTRING

C*********************************
C DEFINE STEP INFORMATION
C*********************************
      IF ( ( NOEL .EQ. NFIRSTELEM )
     &     .AND. ( NPT .EQ. 1 )
     &     .AND. ( (KINC - 1) .NE. INC) ) THEN
         NCYCLE = -1
      ENDIF

      IF ( ( NOEL .EQ. NFIRSTELEM )
     &     .AND. ( NPT .EQ. 1 ) ) THEN
         NCYCLE = NCYCLE + 1
```

```
      ENDIF

      AB_STEP = KSTEP
      GDT = DTIME
      STEP_TIME = TIME(1)
      TOT_TIME = TIME(2)
      INC = KINC - 1

C WE CAN MAKE CALLS TO GETVRM FROM HERE TO GET ADDTIONAL INFO IF NEEDED
      CALL GETVRM('THE', ARRAY, JARRAY, FLGRAY, JRDC)

      DO J = 1, 6
         THERMSTRAIN(J) = ARRAY(J)
         THERMSTRESS(J) = 0.0
         DTHERMSTRESS(J) = 0.0
         DTHERMSTRAIN(J) = 0.0
         DDTHERMSTRAIN_DT(J) = 0.0
      ENDDO


      IF ( AB_PRINT .GE. 1 ) THEN
         WRITE(NDU, *) ' '
         WRITE(NDU, *) '==========================================',
     &         '=============================================='
         WRITE(NDU, *) '-=> START OF USDFLD <=-'
         WRITE(NDU, *) '-=> VALUE OF STEP TIME =', TIME(1), STEP_TIME
         WRITE(NDU, *) '-=> TOTAL TIME         =', TIME(2), TOT_TIME
         WRITE(NDU, *) '-=> DTIME              =', DTIME, GDT
         WRITE(NDU, *) '-=> KSTEP              =', KSTEP, AB_STEP
         WRITE(NDU, *) '-=> KINC               =', KINC, INC+1
         WRITE(NDU, *) '-=> NOEL               =', NOEL, NNOEL
         WRITE(NDU, *) '-=> NPT                =', NPT, NNPT
         WRITE(NDU, *) '-=> NDI                =', NDI
         WRITE(NDU, *) '-=> NSHR               =', NSHR
         WRITE(NDU, *) '-=> NCYCLE             =', NCYCLE
         WRITE(NDU, *) '-=> NFIRSTELEM         =', NFIRSTELEM
         WRITE(NDU, *) '-=> NFIRSTIP           =', NFIRSTIP
         WRITE(NDU, *) '-=> NCALL              =', NCALL
         WRITE(NDU, *) '-=> FNAME =|', FNAME, '|'
      ENDIF


      RETURN
      END
```

```
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
  012
C
C
C
C
C
C
C
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
  012

      SUBROUTINE FEAMAC_INIT(NOEL, NPT)

      INCLUDE 'ABA_PARAM.INC'

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH
      CHARACTER*80 TSTRING

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2


      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP


      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION

      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR

      LOGICAL SIGAL_STATV

      CALL FILL_STRING(PATH, ' ')
      CALL FILL_STRING(FNAME, ' ')
      CALL FILL_STRING(EXTSION, ' ')
      CALL FILL_STRING(DFNAME, ' ')
      CALL FILL_STRING(TSTRING, ' ')
      CALL FILL_STRING(TNAME1, ' ')
      CALL FILL_STRING(TNAME2, ' ')

C ****************************************
C
C
C
C
C
C
```

```
C      __  _______ __________
C    / / / / ___// ____/ __  \
C   / / / /\__ \/ __/ / /_/ /
C  / /_/ /___/ / /___/ _, _/
C  \____//____/_____/_/ |_|
C      ___________ _________________
C     / ____/  __ \/  _/_   __/ ___/
C    / __/ / / / // /  / /   \__ \
C   / /___/ /_/ // /  / /   ___/ /
C  /_____/_____/___/ /_/   /____/
C
C ****************************************
C *** BEGIN USER EDITS ***
C *** USER NEEDS TO CHANGE THE NEXT LINES AS NEEDED...
C PATH TO WORKING DIRECTORY
      PATH = '/home/smdant/macdev/feabench/man_demo'

C NOTE: ABAQUS WILL CONVERT THE MATERIAL NAME TO UPPER CASE.
C MAC INPUT DECK NAME EXTENSION
      EXTSION = '.MAC'


C THE FOLLOWING IS ONLY USED TO RESOLVE VERSION DIFFERENCES BETWEEN
  SUN AND SGI
C NAME OF DIAGNOSTIC FILE
      DFNAME = 'MACDIAG'


C     MAC PLOT FILES:
C     - # OF POINTS TO PLOT, I.E # OF PAIRS TO FOLLOW
      NPEL = 2

C     - ELEMENT # AND INTEGRATION POINT # PAIRS TO PLOT
C           N_PEN - ELEMENT NUMBERS
C           N_PEN - INTEGRATION POINT NUMBERS

      J = 1
      N_PEN(J) = 1
      N_PIN(J) = 1

      J = J + 1
      N_PEN(J) = 2
      N_PIN(J) = 1


C     PATRAN PLOT FILES:
C     - # OF POINTS TO PLOT PATRAN DATA, I.E # OF PAIRS TO FOLLOW
      NPATPEL = 0

C     - ELEMENT # AND INTEGRATION POINT # PAIRS TO PLOT PATRAN DATA
C           N_PATPEN - ELEMENT NUMBERS
C           N_PATPEN - INTEGRATION POINT NUMBERS

C      J = 1
C      N_PATPEN(J) = 1
C      N_PATPIN(J) = 1

C      J = J + 1
C      N_PATPEN(J) = 2
C      N_PATPIN(J) = 1


C THE FOLLOWING IS DEPRECIATED OPTION AND MAY NOT BE SUPPORTED
C IN THE NEXT RELEASE OF FEAMAC.  IT'S USE IS NOT RECOMMENDED.
```

```
C
C SET THE AL_STORE VARIABLE TO 1 IF YOU WANT TO POPULATE THE STATEV
C  ARRAY PER ALLISON ENGINE COMPANY'S SPECS. (IDP 1 & 11 ONLY)
C
C  NOTES: 1) THIS OPTION REQUIRES THE USE OF THE ALTERNATE METHOD OF
C            USER MATERIAL DEFINTION AS DESCRIBED IN THE MAC MANUAL.
C
C         2) A STATE VARIABLE SPACE OF 900 IS REQUIRED.
C
C                   ---------------
C                  |       |       |
C                  |  M1   |  M2   |
C                  |       |       |
C                  |  (3)  |  (4)  |
C                  |       |       |
C                   ---------------
C          BETA    |       |       |
C           ^      |   F   |  M3   |
C           |      |       |       |
C           |      |  (1)  |  (2)  |
C           |      |       |       |
C           |      ---------------
C           |
C           ----------> GAMMA
C    INDICES
C    1 -  12  -=> MATRIX STRESS AND STRAIN IN SUBCELL M1
C   23 -  34  -=> FIBER STRESS AND STRAIN
C  132 - 137  -=> MATRIX MECHANICAL STRAIN COMPONENTS IN SUBCELL M1
C  301 - 312  -=> MATRIX STRESS AND STRAIN IN SUBCELL M2
C  432 - 437  -=> MATRIX MECHANICAL STRAIN COMPONENTS IN SUBCELL M2
C  601 - 612  -=> MATRIX STRESS AND STRAIN IN SUBCELL M3
C  732 - 737  -=> MATRIX MECHANICAL STRAIN COMPONENTS IN SUBCELL M3

      AL_STORE = 0


      AB_PRINT = 0


C ****************************************
C *** TEMP FILE LOCATIONS
C *** SYS ADMIN OR USER NEEDS TO CHANGE THE NEXT LINES AS NEEDED...
      TNAME1 = '/var/tmp/feamac1'
      TNAME2 = '/var/tmp/feamac2'


C ****************************************
C *** END OF USER EDITS ***
C      ________  _______
C     / ____/ | / / __ \
C    / __/ /  |/ / / / / /
C   / /___/ /|  / /_/ /
C  /_____/_/ |_/_____/
C     __  _______ __________
C    / / / / ___// ____/ __ \
C   / / / /\__ \/ __/ / /_/ /
C  / /_/ /___/ / /___/ _, _/
C  \____//____/_____/_/ |_|
C      __________ ______________
C     / ____/ __ \/  _/_  __/ ___/
C    / __/ / / / // /  / /  \__ \
C   / /___/ /_/ // /  / /  ___/ /
C  /_____/_____/___/ /_/  /____/
C
```

```
C ****************************************
      TSTRING = PATH
      CALL STRING_APPEND(TSTRING, '/ ')
      PATH = TSTRING

      TSTRING = PATH
      CALL STRING_APPEND(TSTRING, DFNAME)
      DFNAME = TSTRING

      DO 100 J = 1, NPEL
         IF ( (N_PIN(J) .LT. 1) .OR. (N_PIN(J) .GT. 8) ) THEN
            N_PIN(J) = 1
         ENDIF
 100  CONTINUE

      NDU = 7
C UN-COMMENT NEXT 2 LINES IF YOUR ON A SGI
C         NDU = 18
C         OPEN(UNIT = NDU, FILE = DFNAME, STATUS = 'UNKNOWN')

      WRITE(NDU, *) 'BEGIN'

      NFIRSTELEM = NOEL
      NFIRSTIP = NPT

      NCYCLE = -1

      GAB_PRINT = AB_PRINT

      RETURN
      END


C
   1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678 9
   012
C     __  ____  ______  ______
C    / / / /  |/  /   |/_  __/
C   / / / / /|_/ / /| | / /
C  / /_/ / /  / / ___ |/ /
C  \____/_/  /_/_/  |_/_/
C
C
   1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678 9
   012

      SUBROUTINE UMAT(STRESS, STATEV, DDSDDE, SSE, SPD, SCD, RPL,
     &     DDSDDT, DRPLDE, DRPLDT, STRAN, DSTRAN, TIME, DTIME, TEMP,
     &     DTEMP, PREDEF, DPRED, CMNAME, NDI, NSHR, NTENS, NSTATV,
     &     PROPS, NPROPS, COORDS, DROT, PNEWDT, CELENT, DFGRD0,
     &     DFGRD1, NOEL, NPT, LAYER, KSPT, KSTEP, KINC)


C
   1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678 9
   012
C
C     STRESS(NTENS) - PASSED IN AS THE STRESSES TENSOR AT THE BEGINNING
C          OF THE INCREMENT, SIGMA_I_J.  (THIS SHOULD BE UPDATED IN
C          THIS ROUTINE.)
C          -IN/OUT
```

```
C     STATEV(NSTATV) - SOLUTION DEPENDANT STATE VARIABLES.
C          -IN/OUT
C            Format of state variable space...
C
C                 ____________________________________________________________
C                |
C                | total strain      (1-6)
C                |                                “MACRO” quantities when
C                | cauchy stress     (7-12)       micromechanics model
C                |                                or continuum model
C                | inelastic strain  (13-18)      is used.
C                |
C                | state variable(s) (19-30)
C                | (space for 2 (6x1) vectors)
C                |
C                | thermal strain    (31-36)
C                |
C                | temperature       (37)
C                |
C                | future use        (38-40)
C                |
C                | future use        (41-43)
C                |
C                | future use        (44-46)
C                |
C                |-----------------------------------------------------
C                |
C                | total strain      (47-52)
C                |
C                | cauchy stress     (53-58)    “MICRO” quantities:
C                |                               this block of NSVARS = 42
C                | inelastic strain  (59-64)    is repeated for each subcell
C                |
C                | internal state variable(s)
C                |                   (65-76)
C                | (space for 2 [6x1] vectors)
C                |
C                | thermal strain    (77-82)
C                |
C                | rn & rd debond    (83-88)
C                | parameters
C                |
C                |.........................
C                |            :
C                |           etc.    (89-...)
C                |____________________________________________________________
C
C
C     DDSDDE(NTENS, NTENS) - JACOBIAN MATRIX Of THE CONSTITUTIVE
C          MODEL.
C          -OUT
C     SSE - SPECIFIC ELASTIC STRAIN ENERGY.
C          -OUT
C     SPD - PLASTIC DISSIPATION.
C          -OUT
C     SCD - CREEP DISSIPATION.
C          -OUT
C     RPL - VOLUMETRIC HEAT GENERATION PER UNIT TIME AT THE END
C          OF THE INC.
C          -OUT(COUPLED)
C     DDSDDT(NTENS) - VARIATION OF STRESS INCREMENTS WITH
C          RESPECT TO TEMPERATURE.
C          -OUT(COUPLED)
C     DRPLDE(NTENS) - VARIATION OF RPL WRT STRAIN INCREMENTS.
```

```
C          -OUT(COUPLED)
C     DRPLDT - VARIATION OF RPL WRT THE TEMPERATURE INCREMENT.
C          -OUT(COUPLED)
C     STRAN(NTENS) - TOTAL STRAINS AT BEGINNING OF THE INCREMENT.
C          -IN
C     DSTRAN(NTENS) - STRAIN INCREMENTS
C          -IN
C     TIME(1) - VALUE OF STEP TIME AT BEGINNING OF CURRENT INCREMENT
C          -IN
C     TIME(2) - TOTAL TIME AT BEGINNING OF CURRENT INCREMENT
C          -IN
C     DTIME - TIME INCREMENT
C          -IN
C     TEMP - TEMPERATURE AT START OF INCREMENT
C          -IN
C     DTEMP - INCREMENT OF TEMPERATURE
C          -IN
C     PREDEF - ARRAY OF INTERPOLATED VALUES OF PREDEFINED FIELD
C          VARIABLES.
C          -IN
C     DPRED - ARRAY OF INCREMENTS OF PREDEFINED FIELD VARIABLES.
C          -IN
C     CMNAME - NAME GIVEN ON *MATERIAL OPTION, LEFT JUSTIFIED.
C          -IN
C     NDI - NUMBER OF DIRECT STRESS COMPONENTS AT THIS POINT.
C          -IN
C     NSHR - NUMBER OF ENGINEERING SHEAR STRESS COMPONENTS AT
C          THIS POINT.
C          -IN
C     NTENS - SIZE OF THE STRESS OR STRAIN COMPONENT ARRAY
C          (NDI + NSHR)
C          -IN
C     NSTATV - NUMBER OF SOLUTION DEPENDANT STATE VARIABLES.
C          -IN
C     PROPS(NPROPS) - ARRAY OF MATERIAL CONSTANTS.
C          -IN
C     NPROPS - NUMBER OF MATERIAL CONSTANTS.
C          -IN
C     COORDS - ARRAY CONTAINING THE COORDINATES OF THIS POINT.
C          -IN
C     DROT(3,3) - ROTAION INCREMENT MATRIX.
C          -IN
C     PNEWDT - RATIO OF SUGGESTED NEW TIME INCREMENT TO THE
C          TIME INCREMENT BEING USED.
C          -UPDATE
C     CELENT - CHARACTERISTIC ELEMENT LENGTH.
C          -IN
C     DFGRD0(3,3) - ARRAY CONTAINING DEFORMATION GRADIENT
C          AT THE BEGINNING OF THE INCREMENT
C          -IN
C     DFGRD1(3,3) - ARRAY CONTAINING DEFORMATION GRADIENT
C          AT THE END OF THE INCREMENT
C          -IN
C     NOEL - ELEMENT NUMBER
C          -IN
C     NPT - INTEGRATION POINT NUMBER
C          -IN
C     LAYER - LAYER NUMBER (FOR COMPOSITE SHELLS
C          AND LAYERED SOLIDS).
C          -IN
C     KSPT - SECTION POINT NUMBER WITHIN THE CURRENT LAYER
C          -IN
C     KSTEP - STEP NUMBER
C          -IN
```

```
C     KINC - INCREMENT NUMBER
C          -IN
C
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678 9
  012

      INCLUDE 'ABA_PARAM.INC'
      PARAMETER (ZERO = 0.0D0, HALF = 0.5D0)
      PARAMETER (ONE = 1.0D0, TWO = 2.0D0, THREE = 3.0D0)
      PARAMETER (NFROM_UTHERM = 1, NFROM_UMECH = 2)

      CHARACTER*8 CMNAME

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2


      DIMENSION STRESS(NTENS)
      DIMENSION STATEV(NSTATV)
      DIMENSION DDSDDE(NTENS, NTENS)
      DIMENSION DDSDDT(NTENS)
      DIMENSION DRPLDE(NTENS)
      DIMENSION STRAN(NTENS)
      DIMENSION DSTRAN(NTENS)
      DIMENSION TIME(2)
      DIMENSION PREDEF(1)
      DIMENSION DPRED(1)
      DIMENSION PROPS(NPROPS)
      DIMENSION COORDS(3)
      DIMENSION DROT(3, 3)
      DIMENSION DFGRD0(3, 3)
      DIMENSION DFGRD1(3, 3)


      DIMENSION PLSTRAN(6)

      DIMENSION OSTRESS(6)
      DIMENSION OSTRAIN(6)
      DIMENSION ODSTRAIN(6)

      INTEGER N(2)
      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP


      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION
      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR
```

```
      COMMON / KTHERMONLY / DTHERMSTRESS(6), DTHERMSTRAIN(6),
     &     DDTHERMSTRAIN_DT(6), THERMSTRESS(6), THERMSTRAIN(6)


      LOGICAL SIGAL_STATV

C ==========================================

      NFEA_FROM = NFROM_UMECH

C*******************************
C KEEP TRACK OF THE NUMBER OF CALLS TO FEAMAC
C*******************************
C IF NCALL == 1->2 : WE HAVE ALREADY BEEN TO THERMAL
C IF NCALL == 0->1 : NO THERMAL WAS REQUESTED OR DTEMP IS ZERO
      NCALL = NCALL + 1

C*******************************
C FLAG TO DETERMINE HOW DATA WILL BE STORED
C  IF SIGAL_STATV == TRUE  - DATA STORED IN STATE VARIABLE
C                            SPACE PROVIDED BY ABAQUS
C  IF SIGAL_STATV == FALSE - DATA STORED IN FILES
C*******************************
      SIGAL_STATV = .TRUE.
      AB_PRINT = GAB_PRINT
      IF( NPROPS .GE. 1 ) THEN
         IF( PROPS(1) .EQ. 2 ) SIGAL_STATV = .FALSE.
         IF( NPROPS .GE. 2 ) AB_PRINT = INT(PROPS(2))
      ELSE
         SIGAL_STATV = .FALSE.
      ENDIF


      N(1) = NOEL
      N(2) = NOEL

      IF ( AB_PRINT .GE. 1 ) THEN
         WRITE(NDU, *) ' '
         WRITE(NDU, *) '==========================================',
     &        '==============================================='
         WRITE(NDU, *) '-=> START OF UMAT <=-'
         WRITE(NDU, *) '-=> STORE IN STATEV ?  =', SIGAL_STATV
         WRITE(NDU, *) '-=> VALUE OF STEP TIME =', TIME(1), STEP_TIME
         WRITE(NDU, *) '-=> TOTAL TIME         =', TIME(2), TOT_TIME
         WRITE(NDU, *) '-=> DTIME              =', DTIME, GDT
         WRITE(NDU, *) '-=> KSTEP              =', KSTEP, AB_STEP
         WRITE(NDU, *) '-=> KINC               =', KINC, INC+1
         WRITE(NDU, *) '-=> NOEL               =', NOEL, NNOEL
         WRITE(NDU, *) '-=> NPT                =', NPT, NNPT
         WRITE(NDU, *) '-=> NCYCLE             =', NCYCLE
         WRITE(NDU, *) '-=> NFIRSTELEM         =', NFIRSTELEM
         WRITE(NDU, *) '-=> NFIRSTIP           =', NFIRSTIP
         WRITE(NDU, *) '-=> NCALL              =', NCALL
         WRITE(NDU, *) '-=> FNAME =|', FNAME, '|'
         WRITE(NDU, *) '-=> TEMP =', TEMP
         WRITE(NDU, *) '-=> DTEMP =', DTEMP
         WRITE(NDU, *) 'STRAIN IN:'
         WRITE(NDU, *) '|', STRAN(1), STRAN(2), STRAN(3)
```

```
      WRITE(NDU, *) '|', STRAN(4), STRAN(5), STRAN(6)
      WRITE(NDU, *) 'DSTRAIN IN:'
      WRITE(NDU, *) '|', DSTRAN(1), DSTRAN(2), DSTRAN(3)
      WRITE(NDU, *) '|', DSTRAN(4), DSTRAN(5), DSTRAN(6)
      WRITE(NDU, *) 'STRESS IN:'
      WRITE(NDU, *) '|', STRESS(1), STRESS(2), STRESS(3)
      WRITE(NDU, *) '|', STRESS(4), STRESS(5), STRESS(6)

      WRITE(NDU, *) ' CMNAME =|', CMNAME, '|'
      WRITE(NDU, *) ' NDI = ', NDI
      WRITE(NDU, *) ' NSHR = ', NSHR
      WRITE(NDU, *) ' NTENS = ', NTENS
      WRITE(NDU, *) ' NSTATV = ', NSTATV
      WRITE(NDU, *) ' NPROPS = ', NPROPS
      WRITE(NDU, *) ' CELENT = ', CELENT
      WRITE(NDU, *) ' NOEL = ', NOEL
      WRITE(NDU, *) ' NPT = ', NPT
      WRITE(NDU, *) ' LAYER = ', LAYER
      WRITE(NDU, *) ' KSPT = ', KSPT
      WRITE(NDU, *) ' KSTEP = ', KSTEP
      WRITE(NDU, *) ' KINC = ', KINC

      WRITE(NDU, *) '-=> N=', N, '|'

      WRITE(NDU, *) '-=> INC    =', INC
      WRITE(NDU, *) '-=> NCYCLE =', NCYCLE

      WRITE(NDU, *) '-=> TEMP =', TEMP
      WRITE(NDU, *) '-=> DTEMP =', DTEMP

      WRITE(NDU, *) '-=> BEFORE FEAMAC_PRE <=-'
   ENDIF



   DO J = 1, 6
      OSTRESS(J) = STRESS(J)

      STRAN(J) = STRAN(J) + THERMSTRAIN(J)
      DSTRAN(J) = DSTRAN(J) + DTHERMSTRAIN(J)

      OSTRAIN(J) = STRAN(J)
      ODSTRAIN(J) = DSTRAN(J)

   ENDDO


   CALL FEAMAC_PRE(DDSDDE, PLSTRAN, STRAN, DSTRAN, STRESS, STATEV,
  &     NSTATV, NTENS, N, NPT, LAYER, NDI, NSHR,
  &     TEMP, DTEMP, THERMSTRAIN)


   IF( (NCALL .EQ. 2) .AND. (NCYCLE .NE. 0) ) THEN

      DO I = 1, 6
         STRESS(I) = OSTRESS(I)
         DO J = 1, 6
            STRESS(I) = STRESS(I)
  &              +  DDSDDE(I, J) * (ODSTRAIN(J) - DTHERMSTRAIN(J))
         ENDDO
      ENDDO


      DO J = 1, 6
```

```
            IF( DTEMP .NE. 0.0 ) THEN
               DDSDDT(J) = DTHERMSTRESS(J) / DTEMP
            ELSE
               DDSDDT(J) = 0.0
            ENDIF
            DRPLDE(J) = 0.0
         ENDDO
         RPL = 0.0
         DRPLDT = 0.0
      ENDIF


      IF ( AB_PRINT .GE. 1 ) THEN
         WRITE(NDU, *) '-=> AFTER FEAMAC_PRE <=-'

         WRITE(NDU, *) 'DTHERMAL STRESS:'
         WRITE(NDU, *) '|', DTHERMSTRESS(1), DTHERMSTRESS(2),
     &        DTHERMSTRESS(3)
         WRITE(NDU, *) '|', DTHERMSTRESS(4), DTHERMSTRESS(5),
     &        DTHERMSTRESS(6)

         WRITE(NDU, *) 'DTHERMAL STRAIN:'
         WRITE(NDU, *) '|', DTHERMSTRAIN(1), DTHERMSTRAIN(2),
     &        DTHERMSTRAIN(3)
         WRITE(NDU, *) '|', DTHERMSTRAIN(4), DTHERMSTRAIN(5),
     &        DTHERMSTRAIN(6)

         WRITE(NDU, *) 'DDSDDT:'
         WRITE(NDU, *) '|', DDSDDT(1), DDSDDT(2),
     &        DDSDDT(3)
         WRITE(NDU, *) '|', DDSDDT(4), DDSDDT(5),
     &        DDSDDT(6)

         WRITE(NDU, *) '-=> END OF UMAT <=-'
      ENDIF

C WHATEVER IS IN STRAN/DSTRAN WILL SHOW UP AS MECHANICAL STRAIN
C MECHANICAL STRAIN + THERMAL STRAIN WILL GIVE GLOBAL STRAIN
C WHATEVER IS IN STRESS WILL BE GIVEN AS GLOBAL STRESS
      DO I = 1, 6
         STRAN(I) = OSTRAIN(I) - THERMSTRAIN(I)
         DSTRAN(I) = ODSTRAIN(I) - DTHERMSTRAIN(I)
      ENDDO

      NFIRSTTIME = 0

      RETURN
      END

C
   1111111111222222222233333333334444444444555555555566666666667777777
   7778
C2345678901234567890123456789012345678901234567890123456789012345678901
   01234
C     __  _________  __ ____  ___    _   __
C    / / / / ____/ |/ // __ \/   |  / | / /
C   / / / / __/  |   // /_/ / /| | /  |/ /
C  / /_/ / /___ /   |/ ____/ ___ |/ /|  /
C  \____/_____//_/|_/_/   /_/  |_/_/ |_/
C
C
   1111111111222222222233333333334444444444555555555566666666667777777
   7778
```

```
C234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234

      SUBROUTINE UEXPAN(EXPAN, DEXPANDT, TEMP, TIME, DTIME, PREDEF,
     &     DPRED, STATEV, CMNAME, NSTATV)

      INCLUDE 'ABA_PARAM.INC'

      PARAMETER (ZERO = 0.0D0, HALF = 0.5D0)
      PARAMETER (ONE = 1.0D0, TWO = 2.0D0, THREE = 3.0D0)
      PARAMETER (NFROM_UTHERM = 1, NFROM_UMECH = 2)


      CHARACTER*8 CMNAME

      DIMENSION EXPAN(*), DEXPANDT(*), TEMP(2), TIME(2), PREDEF(*)
      DIMENSION DPRED(*), STATEV(NSTATV)

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2

      DIMENSION STRAN(6), DSTRAN(6)
      DIMENSION STRESS(6)
      DIMENSION PLSTRAN(6)
      DIMENSION DDEDDT(6, 6)

      INTEGER N(2)
      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP


      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION
      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR
      COMMON / KTHERMONLY / DTHERMSTRESS(6), DTHERMSTRAIN(6),
     &     DDTHERMSTRAIN_DT(6), THERMSTRESS(6), THERMSTRAIN(6)


      LOGICAL SIGAL_STATV

C ====================================================
      NFEA_FROM = NFROM_UTHERM

      AB_PRINT = GAB_PRINT

      DO J = 1, 6
         PLSTRAN(J) = 0.0
         STRAN(J) = 0.0
         DSTRAN(J) = 0.0
         STRESS(J) = 0.0
         PLSTRAN(J) = 0.0
```

```
            DTHERMSTRESS(J) = 0.0
            DTHERMSTRAIN(J) = 0.0
            DDTHERMSTRAIN_DT(J) = 0.0
            DO I = 1, 6
               DDEDDT(I, J) = 0.0
            ENDDO
            EXPAN(J) = 0.0
            DEXPANDT(J) = 0.0
         ENDDO

C NOTE THE "TIME" ARRAY FOR THIS ROUTINE IS PROVIDED AT THE  -E N D-
   OF THE INCREMENT
         TIME(1) = TIME(1) - DTIME
         TIME(2) = TIME(2) - DTIME

C NOTE THE "TEMP" ARRAY FOR THIS ROUTINE IS PROVIDED AT THE  -E N D-
   OF THE INCREMENT
         DTEMP = TEMP(2)
         TEMP(1) = TEMP(1) - DTEMP
         TEMPERATURE = TEMP(1)


         IF ( DTEMP .EQ. 0.0 ) THEN
C            WRITE(NDU, *) 'UEXPAN EARLY EXIT - DTEMP = ', DTEMP,
C     &          ' TEMP = ', TEMPERATURE
            RETURN
         ENDIF

         NCALL = NCALL + 1
         SIGAL_STATV = .FALSE.

         IF ( AB_PRINT .GE. 1 ) THEN
            WRITE(NDU, *) ' '
            WRITE(NDU, *) '=========================================',
      &          '=============================================='
            WRITE(NDU, *) '-=> START OF UEXPAN <=-'
            WRITE(NDU, *) '-=> VALUE OF STEP TIME =', TIME(1), STEP_TIME
            WRITE(NDU, *) '-=> TOTAL TIME         =', TIME(2), TOT_TIME
            WRITE(NDU, *) '-=> DTIME              =', DTIME, GDT
            WRITE(NDU, *) '-=> KSTEP              =', AB_STEP
            WRITE(NDU, *) '-=> KINC               =', INC+1
            WRITE(NDU, *) '-=> NOEL               =', NNOEL
            WRITE(NDU, *) '-=> NPT                =', NNPT
            WRITE(NDU, *) '-=> NNDI               =', NNDI
            WRITE(NDU, *) '-=> NMSHR              =', NNSHR
            WRITE(NDU, *) '-=> NCYCLE             =', NCYCLE
            WRITE(NDU, *) '-=> NFIRSTELEM         =', NFIRSTELEM
            WRITE(NDU, *) '-=> NFIRSTIP           =', NFIRSTIP
            WRITE(NDU, *) '-=> NCALL              =', NCALL
            WRITE(NDU, *) '-=> FNAME =|', FNAME, '|'
            WRITE(NDU, *) '-=> TEMPERATURE =', TEMPERATURE
            WRITE(NDU, *) '-=> DTEMP =', DTEMP

            WRITE(NDU, *) '-=> BEFORE FEAMAC_PRE <=-'
         ENDIF


         N(1) = NNOEL
         N(2) = NNOEL
         LAYER = 0
         NTENS = 6
```

```
C NOTE: HERE DSTRAN IS NOT A RATE BUT THE ALPHAS INSTEAD - I.E. THE
   VARIATION OF THERMAL
C           STRAINS WITH RESPECT TO TEMPERATURE (NOT TIME)

      CALL FEAMAC_PRE(DDEDDT, PLSTRAN, STRAN, DSTRAN, STRESS,
   STATEV,
     &     NSTATV, NTENS, N, NNPT, LAYER, NNDI, NNSHR,
     &     TEMPERATURE, DTEMP, THERMSTRAIN)

C NOTE: THESE ARE INCREMENTS IN THERMAL STRAIN/STRESS
C NEED TO FIGURE OUT A GOOD WAY TO GET THE SIZES OF THESE
      DO J = 1, 6
         EXPAN(J) = STRAN(J)
         DTHERMSTRAIN(J) = STRAN(J)
         DDTHERMSTRAIN_DT(J) = DSTRAN(J)
         DEXPANDT(J) = DSTRAN(J)
      ENDDO

      DO I = 1, 6
         DTHERMSTRESS(I) = 0.0
         DO J = 1, 6
            DTHERMSTRESS(I) = DTHERMSTRESS(I)
     &           +  DDEDDT(I, J) * DTHERMSTRAIN(J)
         ENDDO
         STRESS(I) = DTHERMSTRESS(I)
      ENDDO

      RETURN
      END


C
   1111111111222222222233333333334444444444555555555566666666667777
C234567890123456789012345678901234567890123456789012345678901234567890123456
```