

CTH Model Implementation

**Shane Schumacher
Engineering Sciences Center
Sandia National Laboratories
Albuquerque, NM**



Executive Summary

- **CTH equation of state implementation**
- **CTH strength model implementation**



CTH EOS Model Implementation



Overview

- **Material model divided into two “parts”**
 - Deviatoric
 - Spherical – Handled by Equation of State
- **EOS models are “primary” or “multi-state”**
 - **Primary EOS**
 - Mie-Gruneisen
 - Sesame
 - JWL
 - **Multi-state**
 - HVRB
 - IGRB



Files Added/Modified

- **data/EOS_data** – Material constants
- **src/eos/eosuiip.F**
- **src/eos/eos.F** – Add 1 file
 - **src/mat/ABB/eosabb.F**
- **src/eos/eosmre.F** – Add 1 file
 - **src/mat/ABB/eosabv.F**
- **src/eos/eos1si.F** – Add 1 file
 - **src/mat/ABB/eosabi.F**
- **src/eos/eos1sk.F** – Add 1 file
 - **src/mat/ABB/eosabk.F**
- **src/esh/elex.F** – Add 1 file
 - **src/mat/ABB/eosabx.F**
- **src/mat/"material_name"** – Directory to contain material model
- **src/mat /"material_name"/*F** – Add material model files here
- **src/mat/objslst** – Modify object list to include material model files

data/EOS_data

- **Abbreviated Model Name**
- **Keywords**
 - 5 letters and/or numbers
 - Total Keywords
- **Material id**
 - Ends with **#**

MGR -----Mie-Gruneisen EOS-----
22 0 0 keywords:
'R0' 'T0' 'CS' 'S1' 'G0' 'CV' 'ESFT' 'RP' 'PS' 'PE' 'CE' 'NSUB' 'S2' 'TYP'
'RO' 'TO' 'S' 'GO' 'B' 'XB' 'NB' 'PWR'

'USER' 0 2.567981E-2 0 0 0 0 0 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 2.00!
'ANTIMONY' 6.700 2.567981E-2 1.983E5 1.652 0.60 2.38E10 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 2.0 1! GMX-6 parameters
.
.
.
'TANTALUM_TEP' 16.59 2.567981E-2 3.391E5 0.743 1.69 1.60E10 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 2.0-2!



src/eos/eosuiip.F

- **Numbering scheme**
 - 1- 109 primary models
 - 110 – 149 multi-state models
 - 150 – 169 coupled models
 - 170 + multiphase models
- **Middle of file**

```
.  
. .  
. .  
IF (MODNAM(I)(1:3).EQ.'MGR') THEN  
  NEQ(NMAT,I) = 1  
  MSTRG = 'MGR'  
  MDSCR = 'Mie-Gruneisen EOS'  
ELSEIF (MODNAM(I)(1:3).EQ.'SES') THEN  
  NEQ(NMAT,I) = 2  
  MSTRG = 'SES'  
  MDSCR = 'Sesame tabular EOS'  
ELSEIF (MODNAM(I)(1:3).EQ.'JWL') THEN  
  NEQ(NMAT,I) = 4  
  MSTRG = 'JWL'  
  MDSCR = 'JWL EOS'  
  
. .  
. .  
ELSEIF(MODNAM(I)(1:3).EQ.'ABB')THEN  
  NEQ(NMAT,I)=14  
  MSTRG='ABB'  
  MDSCR='ABB EOS'  
  
. .  
. .
```

Items in red are changed for your model



src/eos/eos.F

- Calls EOS model to determine pressure and energy as a function of temperature and density
- Middle of file

```
.  
.
IF (MEQ.EQ.1) THEN
  CALL EOSMGR(1,1,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),DCEQ(LOC),
    &      RHO,TEMP,XV(1,NP),NXV(NP),P,E,CS,SX)
  DPR = SX(1)
  DPT = SX(2)
  CV = SX(3)
ELSEIF (MEQ.EQ.2) THEN
  CALL SESEOS(1,1,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),DCEQ(LOC),
    &      RHO,TEMP,XV(1,NP),NXV(NP),P,E,CS,SX,ISX)
  DPR = SX(1)
  DPT = SX(2)
  CV = SX(3)
.
.
.
ELSEIF (MEQ.EQ.14) THEN
  CALL EOSABB(1,1,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),DCEQ(LOC),
    &      RHO,TEMP,XV(1,NP),NXV(NP),P,E,CS,SX)
  DPR = SX(1)
  DPT = SX(2)
  CV = SX(3)
.
.
.
```

Items in red are changed for your model



src/mat/ABB/eosabb.F

- Calculates the following
 - Pressure
 - Energy
 - Sound Speed
 - $dP/d\rho$ @ constant entropy
 - dP/dT @ constant density
 - dE/dT @ constant density
 - $dE/d\rho$ @ constant entropy
- Reference manual
 - RefManEOS.pdf

```
SUBROUTINE EOSABB(MC,NC,UI,GC,DC,  
C   Input  
C   & RHO,TEMP,ALPH,NX,  
C   Output  
C   & PRES,ENRG,CS,  
C   Scratch  
C   & S)  
C  
C*****EOS Package*****  
C  
C PURPOSE. EOSABB: pressure & energy as functions of density  
C           & temperature using "Your Model" EOS.  
C  
C MIG INPUT.  
C MC      = dimensioning constant  
C NC      = number of points to process  
C UI      = user input array  
C GC      = global constants (not used here)  
C DC      = derived material constants array  
C  
C INPUT.  
C RHO     = MASS_DENSITY  
C TEMP    = ABSOLUTE_TEMPERATURE  
C ALPH    = EXTRA~1 (porosity parameter)  
C NX      = NX=1 if using porosity option, else NX=0  
C  
C OUTPUT.  
C PRES    = THERMODYNAMIC_PRESSURE  
C ENRG    = SPECIFIC_INTERNAL_ENERGY  
C CS      = SOUND_SPEED  
C  
C  
C  
C
```

Items in red are changed for your model

src/mat/ABB/eosabb.F cont.

```
.  
.   
.   
C SCRATCH.  
C S(MC,6) = temporary storage  
C S(K,1) = dP/dR on return  
C S(K,2) = dP/dT on return  
C S(K,3) = dE/dT on return  
C S(K,4) = dE/dR on return  
C  
#include "impdoubl.h"  
#ifdef CTH_DYN  
#include "mxsarr.h"  
#include "eoserr.h"  
#endif  
C  
  CHARACTER*10 DEN  
  PARAMETER (ZERO=0.D0,ONE=1.0D0,TWO=2.0D0,THR=3.0D0,FOR=4.0D0,  
    & FIV=5.0D0,SIX=6.0D0,HALF=0.5D0)  
  DIMENSION UI(*),GC(*),DC(*)  
  DIMENSION RHO(MC), TEMP(MC), ALPH(MC),  
    & PRES(MC), ENRG(MC), CS(MC), S(MC,8)  
C  
  R0 = UI(1)  
  T0 = UI(2)  
  C0 = UI(3)  
.  
.  
.  
C  
  DO 2 N=1,NC  
.  
.  
.  
2  CONTINUE  
  RETURN  
  END
```

Move user input to something easy
to follow

Loop over cells

Determine necessary output

Items in blue are your preference



src/eos/eosmre.F

- Calls EOS model to determine pressure and temperature as a function of energy and density
- Middle of file

```
.  
. .  
IF (MEQ.EQ.1) THEN  
  CALL EOSMGV(MC,NC,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),DCEQ(LOC),  
    &      RHO,ENRG,XV(1,1,NP),NXV(NP),PRES,TEMP,CS,SCV)  
ELSEIF (MEQ.EQ.2) THEN  
  CALL SESESV(MC,NC,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),DCEQ(LOC),  
    &      RHO,ENRG,XV(1,1,NP),NXV(NP),0,PRES,TEMP,CS,  
    &      SCV,SCV(1,23))  
. .  
IF (MEQ.EQ.14) THEN  
  CALL EOSABV(MC,NC,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),DCEQ(LOC),  
    &      RHO,ENRG,XV(1,1,NP),NXV(NP),PRES,TEMP,CS,SCV)  
. .  
.
```

Items in red are changed for your model

src/mat/ABB/eosabv.F cont.

```
.  
.   
C  
C SCRATCH.  
C S(MC,6) = temporary storage  
C S(K,1) = dP/dR on return  
C S(K,2) = dP/dT on return  
C S(K,3) = dE/dT on return  
C S(K,4) = dE/dR on return  
C  
#include "impdoubl.h"  
#ifdef CTH_DYN  
#include "mxsarr.h"  
#include "eoserr.h"  
#endif  
C  
CHARACTER*10 DEN  
PARAMETER (ZERO=0.D0,ONE=1.0D0,TWO=2.0D0,THR=3.0D0,FOR=4.0D0,  
& FIV=5.0D0,SIX=6.0D0,HALF=0.5D0)  
DIMENSION UI(*),GC(*),DC(*)  
DIMENSION RHO(MC), ENRG(MC), ALPH(MC),  
& PRES(MC), TEMP(MC), CS(MC), S(MC,8)  
C  
R0 = UI(1)  
T0 = UI(2)  
C0 = UI(3)  
.  
.  
.  
DO 1 N=1,NC  
.  
.  
2 CONTINUE  
RETURN  
END
```

Move user input to something easy
to follow

Loop over cells

Determine necessary output

Items in red are changed for your model
Items in blue are your preference



src/eos/eos1si.F

- **src/eos/eos1si.F**
 - **Calls EOS model for data check**

```
.  
.   
IF (MEQ.EQ.1) THEN  
    CALL EOSMGI(UI,GC,DC,UC,MDC,NDC,VI)  
ELSEIF (MEQ.EQ.2) THEN  
    CALL SESESI(UI,GC,DC,UC,MDC,NDC,VI)  
.   
.   
ELSEIF (MEQ.EQ.14) THEN  
    CALL EOSABI(UI,GC,DC,UC,MDC,NDC,VI)  
ENDIF  
.   
.   
.
```

Items in red are changed for your model

src/mat/ABB/eosabi.F

- Data check routine
- Echo model data

```
SUBROUTINE EOSABI (UI,GC,DC,  
C   Extended MIG Input  
C   & UC,MDC,NDC,VI)  
C*****EOS Package*****  
C  
C PURPOSE.  EOSABI: data check routine for "Your Model" EOS.  
C  
C STANDARD MIG INPUT.  
C   UI    = user input array  
C   GC    = global constants (not used here)  
C   DC    = derived material constants array  
C  
C EXTENDED MIG INPUT.  
C   UC    = character input array  
C   MDC    = amount of storage available in array DC  
C   NDC    = amount of storage used in array DC (here set to 12)  
C   VI    = default initial values and melting parameters  
C   VI(1) = initial density  
C   VI(2) = initial temperature  
C   VI(3) = minimum solid density  
C   VI(4) = melting temperature  
C   VI(5) = melting density  
C  
C Input parameters are:  
C  
C   UI(1) = R0  - initial density (required)  
C  
C  
C  
C
```

User input definitions

Items in red are changed for your model
Items in blue are your preference

Derived constant definitions

```

C Derived parameters are:
C
C DC(1) = A(1) - coefficient in temperature fit
C
C
C*****
C
C#include "impdoubl.h"
C#define CTH_DYN
C#include "iofiles.h"
C#include "mpcctl.h"
C#endif
C
C CHARACTER*1 UC(*)
C DIMENSION UI(*),GC(*),DC(*),VI(*),A(8)
C PARAMETER (ZERO=0.0D0,HALF=0.5D0,ONE=1.0D0,TWO=2.0D0,FIV=5.0D0
C & TEN=10.0D0,PD8=0.8D0,PM4=1.0D-4,PE9=1.0D9,PM6=1.0D-6)
C
C
C
C
C IF(LPARNT) THEN
C WRITE(KPT6,*) 'Mie-Gruneisen:'
C WRITE(KPT6,*) 'UI(1) = R0 =',UI(1)
C WRITE(KPT6,*) 'UI(2) = T0 =',UI(2)
C
C
C
C
C ENDIF
C
C
C
C RETURN
C END

```

Declaration of variables

Compute necessary data

Echo model data

Error messaging

Items in blue are your preference



src/eos/eos1sk.F

- **src/eos/eos1sk.F**
 - **Calls EOS model to define extra variables**

```
.  
.   
.   
IF (MEQ.EQ.1) THEN  
    CALL EOSMGK(UI,GC,DC,NX,NAMEA,KEYA,RINIT,RDIM,IADVCT,ITYPE)  
ELSEIF (MEQ.EQ.2) THEN  
    CALL SESESK(UI,GC,DC,NX,NAMEA,KEYA,RINIT,RDIM,IADVCT,ITYPE)  
.   
.   
.   
ELSEIF (MEQ.EQ.14) THEN  
    CALL EOSABK(UI,GC,DC,NX,NAMEA,KEYA,RINIT,RDIM,IADVCT,ITYPE)  
ENDIF  
.   
.   
.
```

Items in red are changed for your model

- **Defines extra variables**
 - History variables
 - Such as porosity

```

SUBROUTINE EOSABK (UI,GC,DC,
C   Output
C   & NX,NAMEA,KEYA,RINIT,RDIM,IADVCT,ITYPE)
C*****EOS Package*****
C
C   PURPOSE.  EOSABK: define extra variables for “Your Model” EOS.
C
C   INPUT.
C   UI       = user input array
C   GC       = global constants (not used here)
C   DC       = derived material constants array
C
C   OUTPUT.
C   NX       = number of extra variables
C   NAMEA    = extra variable names (packed using TOKENS routine)
C   KEYA     = plot variable keywords (packed using TOKENS routine)
C   RINIT    = initial values of extra variables
C   RDIM     = dimensions of extra variables
C   IADVCT   = advection option (1-volume weighted, 2-mass weighted)
C   ITYPE    = variable type
C
C*  INCLUDE IMPDOUBL
#include "impdoubl.h"
C*
C   CHARACTER*30 NAME(1)
C   CHARACTER*15 KEY(1)
C   CHARACTER*1 NAMEA(*),KEYA(*)
C   DIMENSION UI(*),GC(*),DC(*)
C   DIMENSION RINIT(*),RDIM(7,*),IADVCT(*),ITYPE(*)
C   PARAMETER (ONE=1.0D0,TEN=10.0D0)
C

```

Items in red are changed for your model



src/mat/ABB/eosabk.F cont.

Add extra variable



```
.  
.   
.   
NX = 0  
  NTYP = NINT(TEN*UI(14))  
  IF (DC(6).GT.ONE) THEN  
    NX = 1  
    NAME(1) = 'Alpha Parameter'  
    IF (NTYP.EQ.20 .OR. NTYP.EQ.40) THEN  
      KEY(1) = 'ALPI'  
    ELSEIF (NTYP.EQ.21 .OR. NTYP.EQ.41) THEN  
      KEY(1) = 'ALPF'  
    ELSE  
      KEY(1) = 'ALP'  
    ENDIF  
    RINIT(1) = DC(6)  
  ENDIF  
*  
  CALL TOKENS(NX,NAME,NAMEA)  
  CALL TOKENS(NX,KEY,KEYA)  
  RETURN  
END
```

Items in red are changed for your model



src/eos/elex.F

- **src/eos/elex.F**
 - **Calls EOS model to update extra variables**

```
.
.
.
      IF (MEQ.EQ.1) THEN
C
C      MGRUN model
C
      CALL EOSMGX(MAXARR,NV,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),
&      DCEQ(LOC),DT,NXV(NP,NMAT),XV(1,1,NP),
&      R0,E0,P0,T0,VT,ET,SCV)
      ELSEIF (MEQ.EQ.2) THEN
C
C      SESAME model
C
      CALL SESEX(MAXARR,NV,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),
&      DCEQ(LOC),DT,NXV(NP,NMAT),XV(1,1,NP),
&      R0,E0,P0,T0,VT,ET,SCV)
.
.
.
      ELSEIF (MEQ.EQ.14) THEN
      CALL EOSABX(MAXARR,NV,UIEQ(1,NMAT,NP),GCEQ(1,NMAT,NP),
&      DCEQ(LOC),DT,NXV(NP,NMAT),XV(1,1,NP),
&      R0,E0,P0,T0,VT,ET,SCV)
.
.
.
```

Items in red are changed for your model

src/mat/ABB/eosabx.F

- Updates extra variables

```
SUBROUTINE EOSABX(MC,NC,UI,GC,DC,DT,NX,  
C  
C      Input/Output  
C      &      ALPH,  
C  
C      Input/Scratch  
C      &      RI,EI,PI,TI,VT,DE,S)  
C  
C*****EOS Package*****  
C  
C  PURPOSE.  EOSABX: Update your extra variables during one time step  
C  
C  MIG INPUT.  
C  MC      = dimensioning constant  
C  NC      = number of points to process  
C  UI      = user input array  
C  GC      = global constants (not used here)  
C  DC      = derived material constants array  
C  
C  INPUT.  
C  DT      = TIME_STEP (not used here)  
C  NX      = NX=1 if using porosity option, else NX=0  
C  
C  INPUT/OUTPUT.  
C  ALPH    = EXTRA~1 (porosity parameter)  
C  
C  INPUT/SCRATCH.  
C  RI      = MASS_DENSITY at start of time step  
C  EI      = SPECIFIC_INTERNAL_ENERGY at start of time step  
C  PI      = THERMODYNAMIC_PRESSURE at start of time step  
C  TI      = ABSOLUTE_TEMPERATURE at start of time step  
C  VT      = volume change during time step  
C  DE      = energy change during time step  
C  S(MC,9) = additional scratch storage  
C  
C  
C
```

Items in red are changed for your model

src/mat/ABB/eosabx.F cont.

```
.  
.   
#include "impdoubl.h"  
C  
  LOGICAL KSTP  
  PARAMETER (ZERO=0.0D0,ONE=1.0D0,TWO=2.0D0,PRND=1.0D-5,  
    &      ONEP=ONE+PRND,ONEM=ONE-PRND)  
  
  DIMENSION UI(*),GC(*),DC(*)  
  DIMENSION RI(*),EI(*),PI(*),TI(*),VT(*),DE(*),ALPH(*),S(MC,9)  
C  
.  
.  
.  
C  
  DO 2 K=1,NC  
.  
.  
.  
  ALPH(K) = .....  
2  CONTINUE  
  RETURN  
  END
```

Computations per cell

Update extra variable



src/mat/objslst

- src/mat/objslst

```
LIB_OBJJS = \  
    bcj/svpchk.o bcj/svpdam.o bcj/svpdrv.o bcj/svpex.o \  
    bcj/yldchk.o \  
    bfk/concck.o bfk/concxv.o bfk/condis.o bfk/conhro.o \  
    bfk/conmfs.o bfk/elivco.o bfk/elpco.o bfk/eoscoi.o \  
    bfk/eoscov.o bfk/eoscox.o bfk/gauss.o \  
    bfk/random.o bfk/splco.o bfk/splint.o \  
    epr/tepsts.o jms/eosjbi.o jms/eosjbl.o jms/eosjbv.o \  
    ges/alode.o ges/determ.o ges/fnroot.o ges/geschk.o \  
    ges/gesdrv.o ges/gesexv.o ges/gesphy.o ges/gesret.o \  
    ges/gesfld.o ABB/abbdv.o ABB/abbxv.o ABB/abbchk.o \  
    ABB/abbmod.o
```

Items in red are changed for your model



Summary

- **data/EOS_data** – Material constants
- **src/eos/eosuiip.F**
- **src/eos/eos.F** – Add 1 file
 - **src/mat/ABB/eosabb.F**
- **src/eos/eosmre.F** – Add 1 file
 - **src/mat/ABB/eosabv.F**
- **src/eos/eos1si.F** – Add 1 file
 - **src/mat/ABB/eosabi.F**
- **src/eos/eos1sk.F** – Add 1 file
 - **src/mat/ABB/eosabk.F**
- **src/esh/elex.F** – Add 1 file
 - **src/mat/ABB/eosabx.F**
- **src/mat/"material_name"** – Directory to contain material model
- **src/mat /"material_name"/*F** – Add material model files here
- **src/mat/objslst** – Modify object list to include material model files



CTH Strength Model Implementation



Overview

- **Material model divided into two “parts”**
 - Deviatoric
 - Spherical – Handled by Equation of State
- **“Traditional” vs “Modern”**
- **Material model input from CTH**
 - Strain rate
 - Time step
- **Material model input/output user**
 - Model constants
 - Extra variables
- **Material model output to CTH**
 - Deviatoric stress



Limitations

- **Spherical and deviatoric components separated**
 - Efforts in place for certain models to couple between EOS and deviatoric model
- **Staggered solution scheme**
 - Pressure and density at cell center
 - Velocity on the cell faces
- **Stress**
 - 5 stresses used, 6th calculated
- **Failure currently performed after remap**
 - Development task moving fracture to Lagrange



Files Added/Modified MIG Interface

- **data/VP_data** – Material constants
- **src/reb/uinep.F** – Add 4 files
 - include/cth/m###p0a.h
 - include/cth/m###p0b.h
 - include/cth/m###p0c.h
 - include/cth/m###p0d.h
- **src/reb/uinchk.F** – Add 1 file
 - include/cth/m###p1.h
- **src/sas/uiniv.F** – Add 1 file
 - include/cth/m###p2.h
- **src/sas/elsg.F** – Add 1 file
 - include/cth/m###p3.h
- **src/mat/"material_name"** – Directory to contain material model
- **src/mat /"material_name"/*F** – Add material model files here
- **src/mat/objslst** – Modify object list to include material model files

data/VP_data

- **MODLEP = Model number**
- **Abbreviated Model Name**
- **Keywords**
 - 5 letters and/or numbers
 - Total Keywords
- **Material id**
 - Ends with - #

ZE ~~~~~ ZERILLI--ARMSTRONG ~~~~~ MODLEP=4

8 keywords:

	'C1ZE'	'C2ZE'	'C3ZE'	'C4ZE'	'C5ZE'	'AZE'	'NZE'	
matl name								material id
'USER'	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0 !
'COPPER'	0.000000E+00	8.900000E+09	3.249400E+01	1.334575E+00	0.000000E+00	6.500000E+08	1.000000E+00	1 !
	0.330000E+00							
'TANTALUM2'	9.750000E+09	0.000000E+00	5.222025E+01	3.191240E-01	5.250000E+09	4.000000E+08	5.000000E-01	-10 !
	0.330000E+00							
--WARNING--WARNING--WARNING--WARNING--								
Do not modify the above data set!								
It was generated from a MIG data file.								



src/reb/uinep.F include/cth/m###p0a.h

- **src/reb/uinep.F**
 - **Add m###p0a.h**
 - **Near top of file**

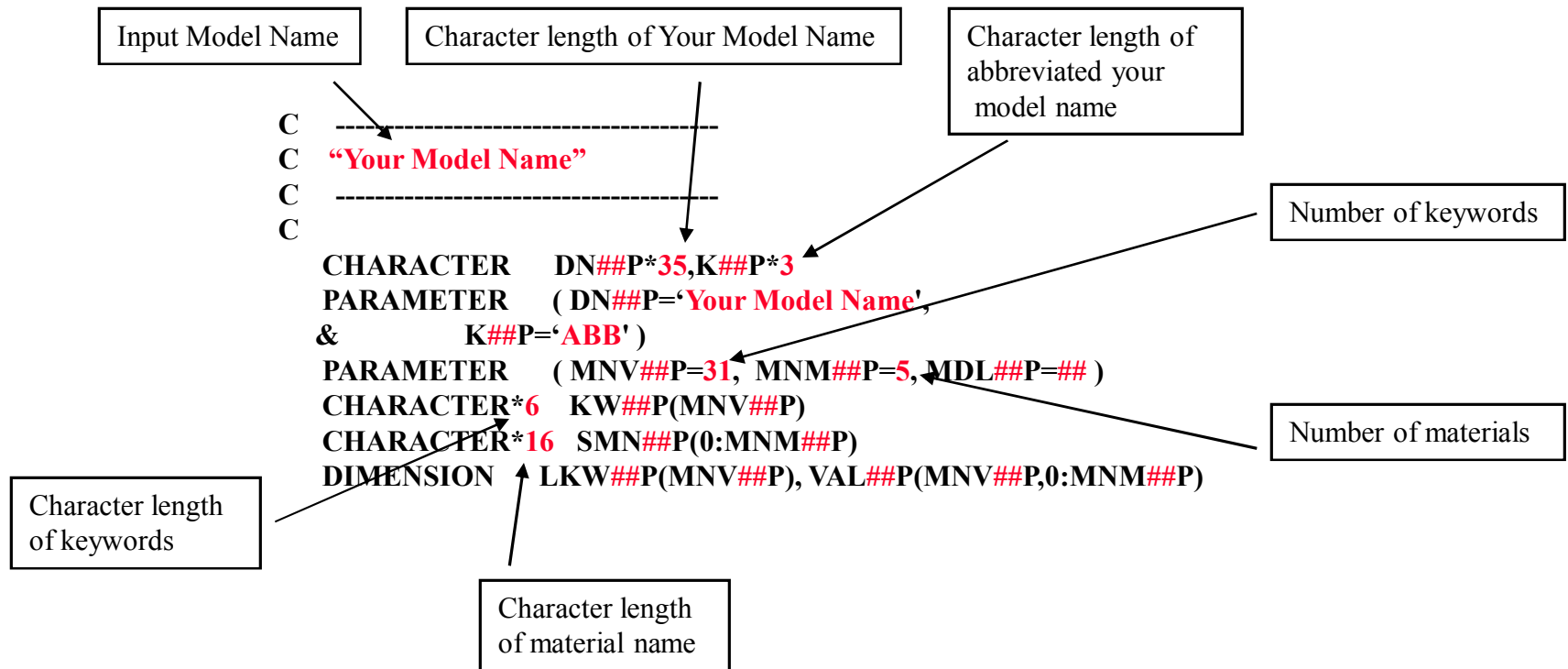
```
C
#include "m03p0a.h"
#include "m04p0a.h"
#include "m05p0a.h"
#include "m06p0a.h"
#include "m07p0a.h"
#include "m08p0a.h"
C
#include "m10p0a.h"
#include "m11p0a.h"
#include "m12p0a.h"
#include "m13p0a.h"
#include "m14p0a.h"
#include "m15p0a.h"
#include "m16p0a.h"
#include "m17p0a.h"
#include "m18p0a.h"
#include "m19p0a.h"
#include "m20p0a.h"
#include "m###p0a.h"
#include "m09p0a.h"
#include "m99p0a.h"
C
```

Items in red are changed for your model



```
src/reb/uinep.F
include/cth/m###p0a.h
```

- include/cth/m###p0a.h
 - ## = Model Number



Items in red are changed for your model



src/reb/uinep.F include/cth/m###p0b.h

- **src/reb/uinep.F**
 - **Add m###p0b.h**
 - **Near top-middle of file**

```
C
#include "m03p0b.h"
#include "m04p0b.h"
#include "m05p0b.h"
#include "m06p0b.h"
#include "m07p0b.h"
#include "m08p0b.h"
#include "m01f0b.h"
#include "m02f0b.h"
#include "m03f0b.h"
#include "m04f0b.h"
#include "m05f0b.h"
#include "m10p0b.h"
#include "m11p0b.h"
#include "m12p0b.h"
#include "m13p0b.h"
#include "m14p0b.h"
#include "m15p0b.h"
#include "m16p0b.h"
#include "m17p0b.h"
#include "m18p0b.h"
#include "m19p0b.h"
#include "m20p0b.h"
#include "m###p0b.h"
#include "m09p0b.h"
#include "m99p0b.h"
C
```

Items in red are changed for your model



```
src/reb/uinep.F  
include/cth/m###p0b.h
```

- include/cth/m###p0b.h
 - ## = Model Number

```
C -----  
C  "Your Model Name"  
C -----  
C  
  CALL RDATA(IOUVP,IAM,K##P,DN##P,MNM##P,MNV##P,0,1,1,0,  
&    NV##P,KW##P,LKW##P,NM##P,SMN##P,VAL##P,  
&    NDU1,KDU1,LDU1,VDU1,NDU2,KDU2,LDU2,VDU2 )
```

Items in red are changed for your model



src/reb/uinep.F include/cth/m###p0c.h

- **src/reb/uinep.F**
 - Add m###p0c.h
 - Near middle-bottom of file

```
C
#include "m03p0c.h"
#include "m04p0c.h"
#include "m05p0c.h"
#include "m06p0c.h"
#include "m07p0c.h"
#include "m08p0c.h"
#include "m01f0c.h"
#include "m02f0c.h"
#include "m03f0c.h"
#include "m04f0c.h"
#include "m05f0c.h"
#include "m10p0c.h"
#include "m11p0c.h"
#include "m12p0c.h"
#include "m13p0c.h"
#include "m14p0c.h"
#include "m15p0c.h"
#include "m16p0c.h"
#include "m17p0c.h"
#include "m18p0c.h"
#include "m19p0c.h"
#include "m20p0c.h"
#include "m###p0c.h"
#include "m09p0c.h"
#include "m99p0c.h"
C
```

Items in red are changed for your model



```
src/reb/uinep.F  
include/cth/m###p0c.h
```

- include/cth/m###p0c.h
 - ## = Model Number

```
C -----  
C  Look for “Your Model” keywords and parameters  
C -----  
C  
CALL  
LOOKFK(MDL##P,K##P,DN##P,MNM##P,MNV##P,NV##P,KW##P,LKW##P,  
      &    NM##P, SMN##P, VAL##P, KODFLG, MATUID,  
      &    IPOB,NVPPAR,VPUINP,UDEFVP,MODLEP,MATNAM,GOTONE)  
IF (GOTONE) GO TO 100
```

Items in red are changed for your model



src/reb/uinep.F include/cth/m###p0d.h

- **src/reb/uinep.F**
 - **Add m###p0d.h**
 - **Near bottom of file**

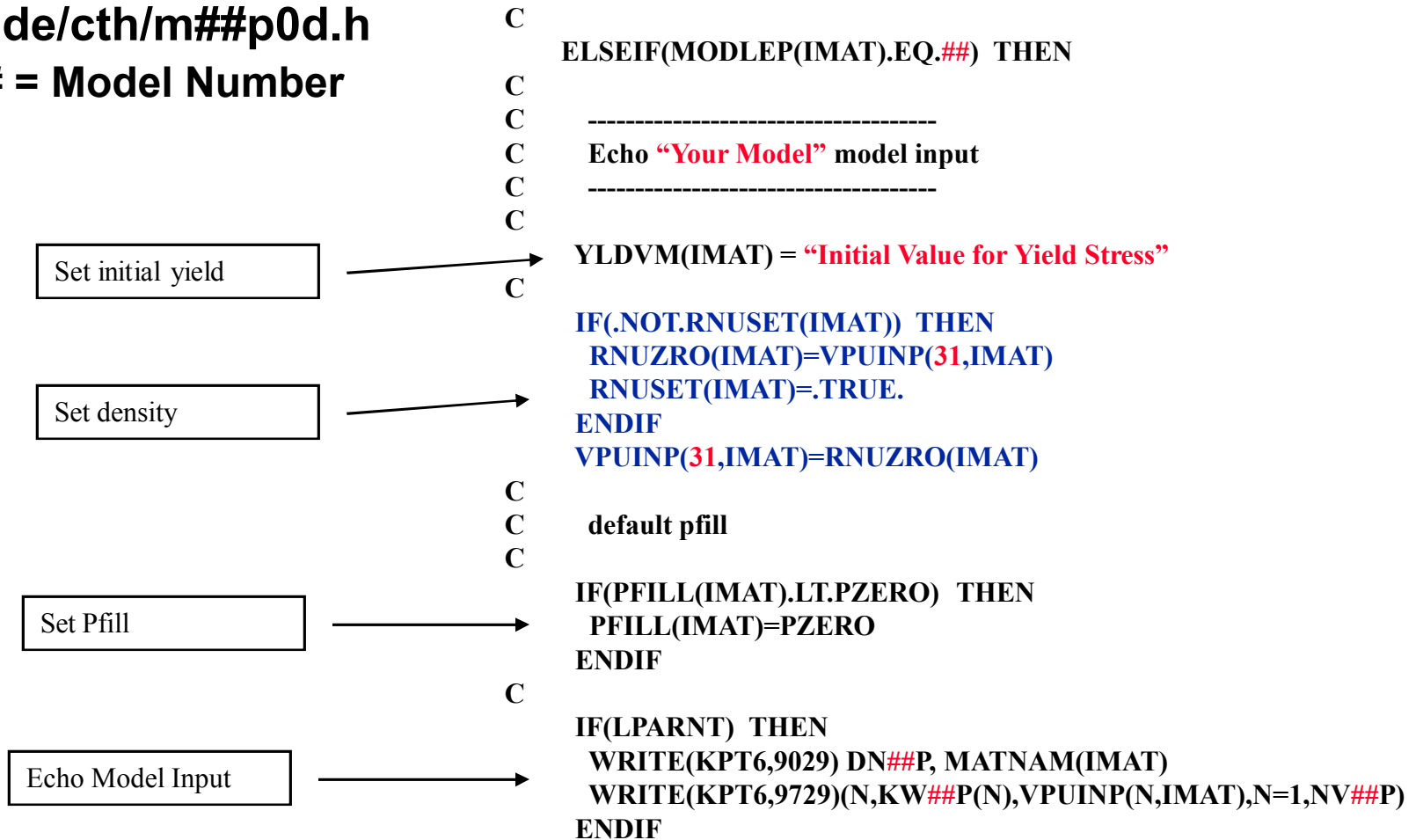
```
C
#include "m03p0d.h"
#include "m04p0d.h"
#include "m05p0d.h"
#include "m06p0d.h"
#include "m07p0d.h"
#include "m08p0d.h"
#include "m10p0d.h"
#include "m11p0d.h"
#include "m12p0d.h"
#include "m13p0d.h"
#include "m14p0d.h"
#include "m15p0d.h"
#include "m16p0d.h"
#include "m17p0d.h"
#include "m18p0d.h"
#include "m19p0d.h"
#include "m20p0d.h"
#include "m###p0d.h"
#include "m09p0d.h"
#include "m99p0d.h"
C
```

Items in red are changed for your model



src/reb/uinep.F include/cth/m###p0d.h

- include/cth/m###p0d.h
 - ## = Model Number



Items in red are changed for your model

Items in blue are your preference



src/reb/uinchk.F include/cth/m###p1.h

- **src/reb/uinchk.F**
 - **Add m###p1.h**
 - **Middle of file**

```
C
#include "m10p1.h"
#include "m11p1.h"
#include "x11p1.h"
#include "m12p1.h"
#include "m13p1.h"
#include "m14p1.h"
#include "m15p1.h"
#include "m16p1.h"
#include "m09p1.h"
#include "m17p1.h"
#include "m18p1.h"
#include "m19p1.h"
#include "m20p1.h"
#include "m###p1.h"
C
```

Items in red are changed for your model



src/mat/ABB/abbchk.F

- src/mat/ABB/abbchk.F
 - Initialize variables

```
SUBROUTINE ABBCHK ( UI, GC, DC )
*****
*****
C   REQUIRED MIG DATA CHECK ROUTINE
C   Checks validity of user inputs
C   Calculates and stores derived material constants.
C*****
C
C   #include "impdoubl.h"
C   #include "iofils.h"
C   #include "mpcthl.h"
C
C   DIMENSION UI(*), GC(*), DC(*)
C   CHARACTER*6 IAM
C   .
C   .
C   PARAMETER ( P90  = 90.0D0 )
C
C   get material constants from UI
C   ELAST  = UI(1)
C   SHEART  = UI(1)/(PTWO*(PONE+UI(2)))
C   POIST  = UI(2)
C   ELASC  = UI(5)
C   SHEARC  = UI(5)/(PTWO*(PONE+UI(6)))
C   POISC  = UI(6)
C   RHO    = UI(10)
C   CSB    = UI(11)
C   .
C   .
C   .
```

Items in red are changed for your model

src/mat//ABB/abbchk.F cont.

- src/mat/ABB/abbchk.F

```
.  
.   
.   
C   Elastic Constants:  
C   
C   Tension  
C   
      CONST = PONE-POIST**PTWO-POIST**PTWO  
      *      -POIST**PTWO-PTWO*POIST**PTHREE  
      CM11 = (PONE-POIST**PTWO)*ELAST/CONST  
      CM12 = (POIST+POIST*POIST)*ELAST/CONST  
      CM13 = CM12  
      CM22 = CM11  
      CM23 = CM12  
      CM33 = CM11  
      CM44 = PTWO * SHEART  
      CM55 = CM44  
      CM66 = CM44  
  
C   
C   Compression  
C   
      CONST = PONE-POISC**PTWO-POISC**PTWO  
      *      -POISC**PTWO-PTWO*POISC**PTHREE  
      CM11C = (PONE-POISC**PTWO)*ELASC/CONST  
      CM12C = (POISC+POISC*POISC)*ELASC/CONST  
      CM13C = CM12C  
      CM22C = CM11C  
      CM23C = CM12C  
      CM33C = CM11C  
      CM44C = PTWO * SHEARC  
      CM55C = CM44C  
      CM66C = CM44C  
  
C   
.   
.
```

Items in red are changed for your model



src/mat/ABB/abbchk.F cont.

- src/mat/ABB/abbchk.F

```
.  
.   
.   
C  
C Determine effective Poisson's ratio that will modify the  
C sound speed determined from the EOS package to come up with  
C the fastest dilatational wave speed for the model to be  
C used by other portions of CTH to determine an appropriate  
C time step:  
C  
CST = SQRT(ELAST/RHO)  
CSC = SQRT(ELASC/RHO)  
CSST = SQRT(SHEART/(PTWO*RHO))  
CSSC = SQRT(SHEARC/(PTWO*RHO))  
CSSST = SQRT(CST**2-PFOUR/PTHREE*CSST**2)  
CSSSC = SQRT(CSC**2-PFOUR/PTHREE*CSSC**2)  
CSMAX = MAX(CSSST,CSSSC)  
RNUZRO = (PTHREE*CSB**2 - CSMAX**2)/(PTHREE*CSB**2 +  
CSMAX**2)  
RNUZRO = MIN(MAX(PMP99,RNUZRO),PHALF)  
.   
.   
. 
```

Items in red are changed for your model



src/mat/ABB/abbchk.F cont.

- **src/mat/ABB/abbchk.F**
 - Update derived constants
 - Update global constants

```
.  
. .  
C  
DC(1) = CM11  
DC(2) = CM12  
DC(3) = CM13  
DC(4) = CM22  
DC(5) = CM23  
DC(6) = CM33  
DC(7) = CM44  
DC(8) = CM55  
DC(9) = CM66  
DC(10) = CM11C  
DC(11) = CM12C  
DC(12) = CM13C  
DC(13) = CM22C  
DC(14) = CM23C  
DC(15) = CM33C  
DC(16) = CM44C  
DC(17) = CM55C  
DC(18) = CM66C  
DC(19) = RNUZRO  
C  
RETURN  
END
```

Items in red are changed for your model



src/sas/uiniv.F include/cth/m##p2.h

- **src/sas/uiniv.F**
 - **Add m##p2.h**
 - **Middle of file**

```
C
#include "m03p2.h"
#include "m04p2.h"
#include "m05p2.h"
#include "m06p2.h"
#include "m07p2.h"
#include "m08p2.h"
#include "m10p2.h"
#include "m11p2.h"
#include "m12p2.h"
#include "m13p2.h"
#include "m14p2.h"
#include "m15p2.h"
#include "m16p2.h"
#include "m09p2.h"
#include "m17p2.h"
#include "m18p2.h"
#include "m19p2.h"
#include "m20p2.h"
#include "m##p2.h"
#include "m99p2.h"
```

Items in red are changed for your model



```
src/sas/uiniv.F
include/cth/m###p2.h
```

- include/cth/m###p2.h
 - ## = Model Number

	IF(MODLEP(IMAT).EQ.##) THEN
	C
	C -----
	C Request "Your Model" extra variables
	C -----
	C
	CALL MIGSEX(NXP(IMAT), NAMEA, KEYA, RNIT, RDIM, IADVCT, ITYPE)
	C
	C ui gc dc
Extra variable file	→ CALL ABBXV(VPUINP(1,IMAT), VPUINP(32,IMAT), VPUINP(34,IMAT),
	C
	C nx namea keya rnit rdim iadvct itype
	C & NXP(IMAT), NAMEA, KEYA, RNIT, RDIM, IADVCT, ITYPE)
	C
	IF(NXP(IMAT).GT.0) CALL MIGXT(IMAT, JXPMIG(IMAT),
	& NXP(IMAT), NAMEA, KEYA, RNIT,
	& RDIM, IADVCT, ITYPE)
	C
Add Plastic Strain Rate	→ CALL MIGADD(IMAT,'SCALAR_PLASTIC_STRAIN_RATE')
	C
Add Equivalent Plastic Strain	→ CALL MIGADD(IMAT,'EQUIVALENT_PLASTIC_STRAIN')
	C
	RSCNIT(1)=VPUINP(14,IMAT)
Add isothermal_elastic_shear_modulus	→ CALL MIGADD(-IMAT,'ISOTHERMAL_ELASTIC_SHEAR_MODULUS')
	ENDIF

Items in red are changed for your model

Items in blue are your preference

src/mat/ABB/abbxv.F cont.

```
.
.
.
C IADVCT = advection and material type option [DEFAULT = 0]
C =0 mass specific with mass advection
C =1 volume specific with mass advection
C =2 mass specific and stationary
C =3 volume specific and stationary
C
C ITYPE = variable type (see migtionary preface) [DEFAULT = 1]
C 1=scalar
C 2=special
C 3=vector
C 4=2nd-order skew-symmetric tensor
C 5=2nd-order symmetric deviatoric tensor
C 6=2nd-order symmetric tensor
C 7=4th-order tensor
C 8=4th-order minor-symmetric tensor
C 9=2nd-order tensor
C 10=4th-order major&minor-symmetric tensor
C 11=2nd-order symmetric tensor 6d
C
C #include "impdoubl.h"
C
C PARAMETER (MCN=30,MCK=5)
C PARAMETER (MXDIM=14)
C CHARACTER*(MCN) NAME(MXDIM)
C CHARACTER*(MCK) KEY(MXDIM)
C CHARACTER*1 NAMEA(*), KEYA(*)
C DIMENSION IADVCT(*),ITYPE(*)
C DIMENSION UI(*), GC(*), DC(*), RINIT(*), RDIM(7,*)
C
C NX=0
C
.
.
.
```

src/mat/ABB/abbxv.F cont,

Add extra variable

```
.
C      first extra variable
C
  NX=NX+1
  NAME(NX) = 'Yield Surf Tension'
  KEY(NX)  = 'FYLDT'
  ITYPE(NX) = 1
  RDIM(1,NX) = 0.0
  RDIM(2,NX) = 0.0
  RDIM(3,NX) = 0.0
  RINIT(NX) = UI(3)
.
.
.
C      eighth extra variable
C
  NX=NX+1
  NAME(NX) = 'Back Stress23'
  KEY(NX)  = 'FB23'
  ITYPE(NX) = 1
  RDIM(1,NX) = 0.0
  RDIM(2,NX) = 0.0
  RDIM(3,NX) = 0.0
  RINIT(NX) = 0.0
C
#####
#
  IF(NX.GT.MXDIM)CALL BOMBED
  $ ('INCREASE MXDIM IN ROUTINE ABBXV AND IN DATA FILE')
C~~~~~
C  convert NAME and KEY to character streams NAMEA and KEYA
  CALL TOKENS(NX,NAME,NAMEA)
  CALL TOKENS(NX,KEY ,KEYA )
  RETURN
  END
```

Items in red are changed for your model
Items in blue are your preference



src/sas/elsg.F include/cth/m###p3.h

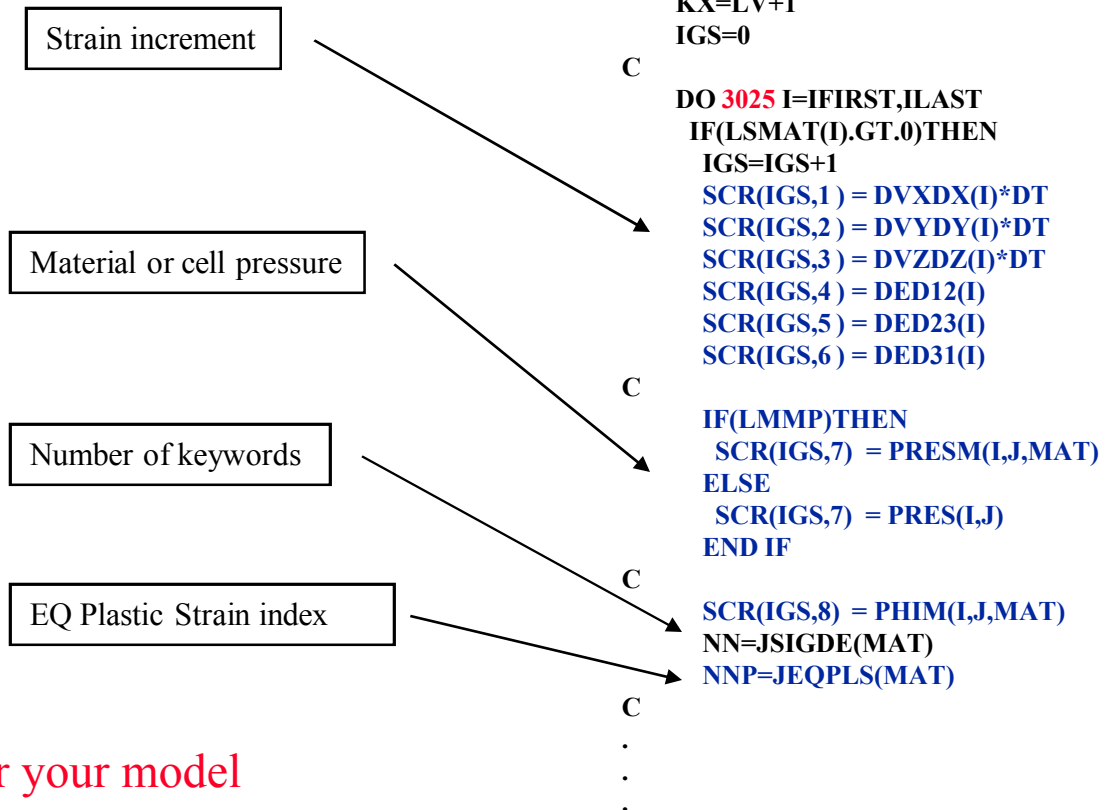
- **src/sas/elsg.F**
 - **Add m###p3.h**
 - **Middle of file**

```
C
#include "m03p3.h"
#include "m04p3.h"
#include "m05p3.h"
#include "m06p3.h"
#include "m07p3.h"
#include "m08p3.h"
#include "m10p3.h"
#include "m11p3.h"
#include "m12p3.h"
#include "m13p3.h"
#include "m14p3.h"
#include "m15p3.h"
#include "m16p3.h"
#include "m09p3.h"
#include "m17p3.h"
#include "m18p3.h"
#include "m19p3.h"
#include "m20p3.h"
#include "m###p3.h"
#include "m99p3.h"
```

Items in red are changed for your model

src/sas/elsg.F include/cth/m###p3.h

- include/cth/m###p3.h
 - ## = Model Number



Items in red are changed for your model
Items in blue are your preference

src/sas/elsg.F cont. include/cth/m##p3.h

Character length of
abbreviated your
model name



```
.
.
.
IF(NN.GT.0)THEN
  SCR(IGS,9) = EXVAR(I,J,NN+0)
  SCR(IGS,10) = EXVAR(I,J,NN+1)
  SCR(IGS,11) = EXVAR(I,J,NN+2)
  SCR(IGS,12) = EXVAR(I,J,NN+3)
  SCR(IGS,13) = EXVAR(I,J,NN+4)
```

```
ELSE
  SCR(IGS,9) = S11O(I)
  SCR(IGS,10) = S22O(I)
  SCR(IGS,11) = S12O(I)
  SCR(IGS,12) = S23O(I)
  SCR(IGS,13) = S31O(I)
```

```
END IF
```

EQ plastic strain variable



```
C
C  PLASTIC STRAIN
C
```

```
SCR(IGS,14) = EXVAR(I,J,NNP)
```

Extra variables



```
C
C  DO 3024 EXTRA=1,NXP(MAT)
C
C    SCR(IGS,LV+EXTRA) = EXVAR(I,J,NXPMIG+EXTRA)
C
C  3024  CONTINUE
C
C    YTEMP(I)=PONE
C  END IF
3025  CONTINUE
C
C    NGS = IGS
```

Items in red are changed for your model
Items in blue are your preference

src/sas/elsg.F cont. include/cth/m##p3.h

CTH to material model interface

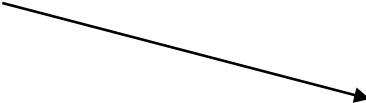
```
.  
. .  
C-----  
C\\CALL GATHER('VELOCITY~GRADIENT~SYM' ,SCR(1,1))  
C\\CALL GATHER('MECHANICAL_PRESSURE' ,SCR(1,7))  
C\\CALL GATHER('VOLUME_FRACTION_OF_MATERIAL' ,SCR(1,8))  
C\\CALL GATHER('STRESS~DEVIATOR' ,SCR(1,9))  
C\\CALL GATHER('EQUIVALENT PLASTIC STRAIN' ,SCR(1,14))  
C\\CALL GATHER('EXTRA~1' ,SCR(1,15))  
C\\CALL GATHER('EXTRA~2' ,SCR(1,16))  
C\\CALL GATHER('EXTRA~3' ,SCR(1,17))  
C\\CALL GATHER('EXTRA~4' ,SCR(1,18))  
C\\CALL GATHER('EXTRA~5' ,SCR(1,19))  
C\\CALL GATHER('EXTRA~6' ,SCR(1,20))  
C\\CALL GATHER('EXTRA~7' ,SCR(1,21))  
C\\CALL GATHER('EXTRA~8' ,SCR(1,22))  
C  
IF(NGS.GT.0) THEN  
  CALL ABBDRV(IMAX,NGS,VPUINP(1,MAT),VPUINP(12,MAT),  
    & VPUINP(19,MAT),IGM,ICYCLE,DT,SCR(1,1),SCR(1,7),  
    & SCR(1,8),SCR(1,9),SCR(1,14),SCR(1,15),SCR(1,16),  
    & SCR(1,17),SCR(1,18),SCR(1,19),SCR(1,20),SCR(1,21),  
    & SCR(1,22),SCR(1,23))  
ENDIF  
C-----  
. . .
```

Items in red are changed for your model
Items in blue are your preference

src/sas/elsg.F cont. include/cth/m##p3.h

```
.  
. .  
. .  
C-----  
      IGS=0  
C  
      DO 3031 I=IFIRST,ILAST  
        IF(LSMAT(I).GT.0)THEN  
          IGS=IGS+1  
          PH=PHIM(I,J,MAT)  
C  
C      update mixed cell quantities  
C  
      IF(IGM.GE.10.AND.IGM.LT.20)THEN  
        S11(I,J) = SCR(IGS,9)*PH + S11(I,J)  
        S22(I,J) = SCR(IGS,10)*PH + S22(I,J)  
      ELSEIF(IGM.GE.20.AND.IGM.LT.30)THEN  
        S11(I,J) = SCR(IGS,9)*PH + S11(I,J)  
        S22(I,J) = SCR(IGS,10)*PH + S22(I,J)  
        S12(I,J) = SCR(IGS,11)*PH + S12(I,J)  
      ELSE  
        S11(I,J) = SCR(IGS,9)*PH + S11(I,J)  
        S22(I,J) = SCR(IGS,10)*PH + S22(I,J)  
        S12(I,J) = SCR(IGS,11)*PH + S12(I,J)  
        S23(I,J) = SCR(IGS,12)*PH + S23(I,J)  
        S31(I,J) = SCR(IGS,13)*PH + S31(I,J)  
      END IF  
C  
. .  
. .
```

Update deviatoric stress



Items in red are changed for your model

src/sas/elsg.F cont. include/cth/m##p3.h

Update extra variables

```
.  
. .  
. .  
C      update material extra variables  
C  
      NN=JSIGDE(MAT)  
C  
      IF(NN.GT.0)THEN  
        EXVAR(I,J,NN+0 ) = SCR(IGS,9)  
        EXVAR(I,J,NN+1 ) = SCR(IGS,10)  
        EXVAR(I,J,NN+2 ) = SCR(IGS,11)  
        EXVAR(I,J,NN+3 ) = SCR(IGS,12)  
        EXVAR(I,J,NN+4 ) = SCR(IGS,13)  
      END IF  
C  
C      UPDATE PLASTIC STRAIN  
      NNP=JEQPLS(MAT)  
      EXVAR(I,J,NNP) = SCR(IGS,14)  
C  
      DO 1362 IXTA=1,NXP(MAT)  
        EXVAR(I,J,NXPMIG+IXTRA)=SCR(IGS,LV+IXTRA)  
1362    CONTINUE  
      ELSE  
      .  
      .  
      .
```

Items in red are changed for your model
Items in blue are your preference



src/sas/elsg.F cont. include/cth/m##p3.h

Clean extra variables

```
.  
. .  
C      zero out material extra variables  
C  
      NN=JSIGDE(MAT)  
C  
      IF(NN.GT.0)THEN  
        EXVAR(I,J,NN+0 ) = PZERO  
        EXVAR(I,J,NN+1 ) = PZERO  
        EXVAR(I,J,NN+2 ) = PZERO  
        EXVAR(I,J,NN+3 ) = PZERO  
        EXVAR(I,J,NN+4 ) = PZERO  
      END IF  
C  
      DO 1372 IXTRA=1,NXP(MAT)  
        EXVAR(I,J,NXPMIG+IXTRA)=PZERO  
1372    CONTINUE  
      END IF  
3031 CONTINUE  
C  
      GO TO 2000  
ENDIF
```

Items in red are changed for your model
Items in blue are your preference



src/mat/ABB/abbdrrv.F

- “Driver” routine for your model

```
SUBROUTINE ABDDRV (MSF,NGS,UI,GC,DC
C
C   input
C   ----
C   &,IGM,ICYCLE,DT,ROD,PRESUR,PHIM
C
C   input and output
C   -----
C   &,SIGDEV,EPLS,EXVAR1,EXVAR2,EXVAR3,EXVAR4,EXVAR5
C   &,EXVAR6,EXVAR7,EXVAR8
C
C   scratch
C   -----
C   &,SCR)
C*****
C   REQUIRED MIG DRIVER ROUTINE for “Your Model”
C   Loops over a gather-scatter array.
C
C   NGS: Number of gather-scatter "cells" to process
C   MSF: dimension (stride) for field arrays
C   UI: user input array
C   GC: global constants array
C   DC: derived material constants array
C   IGM: GEOM
C   ROD: VELOCITY~GRADIENT~SYM
C   ICYCLE: CYCLE
C   DT: TIME_STEP
C   PRESUR: MECHANICAL_PRESSURE
C   PHIM: VOLUME_FRACTION_OF_MATERIAL
C   SIGDEV: STRESS~DEVIATOR
C   EXVAR1: EXTRA~1
C   EXVAR2: EXTRA~2
C   EXVAR3: EXTRA~3
C   SCR: SCRATCH~1THRU36
C
C
```

Items in red are changed for your model

Items in blue are your preference

src/mat/ABB/abbdrrv.F cont.

```
#include "impdoubl.h"
C
  DIMENSION UI(*),GC(*),DC(*)
  DIMENSION ROD(MSF,6)
  &,PRESUR(MSF),PHIM(MSF),SIGDEV(MSF,5)
  &,EXVAR1(MSF),EXVAR2(MSF),EXVAR3(MSF),EXVAR4(MSF)
  &,EXVAR5(MSF),EXVAR6(MSF),EXVAR7(MSF),EPLS(MSF)
  &,EXVAR8(MSF)
  &,SCR(MSF,8)
C
  CHARACTER*6 IAM
  PARAMETER(IAM = 'ABBDrrv')
C Call routine to determine new values of Fiber deviatoric stress
C
  CALL ABBMOD(
C input
C -----
  * NGS,UI,DC,
  * ROD(1,1),ROD(1,2),ROD(1,3),ROD(1,4),ROD(1,5),
  * ROD(1,6),
  * IGM,ICYCLE,DT,PRESUR,PHIM,
C input and output
C -----
  *
EPLS,EXVAR1,EXVAR2,EXVAR3,EXVAR4,EXVAR5,EXVAR6,EXVAR7
,
  * EXVAR8,
  * SIGDEV(1,1),SIGDEV(1,2),SIGDEV(1,3),SIGDEV(1,4),SIGDEV(1,5),
C scratch
C -----
  * SCR(1,1),SCR(1,2),SCR(1,3),SCR(1,4),SCR(1,5),SCR(1,6),
  * SCR(1,7), SCR(1,8))
C
  RETURN
  END
```

Items in red are changed for your model

Items in blue are your preference

Dimension arrays

Dimension scratch arrays



src/mat/ABB/abbmod.F cont.

Add parameters

2d

3d

```
.  
. C      define parameters  
C      PARAMETER ( PZERO = 0.0D0 )  
.  
.  
C      PARAMETER ( PPNINE = 0.9D0 )  
C  
C      2dr & 2dc:  
C  
C      IF(IGM .LT. 30) THEN  
C      DO 110 I = 1,NGS  
.  
.  
.  
150 CONTINUE  
C  
C      3dr:  
C      ELSEIF(IGM.GE.30) THEN  
C      DO 310 I = 1,NGS  
C  
.  
.  
.  
350 CONTINUE  
C  
C      ENDIF  
C  
C      RETURN  
C      END
```

Items in red are changed for your model



src/mat/objslst

- src/mat/objslst

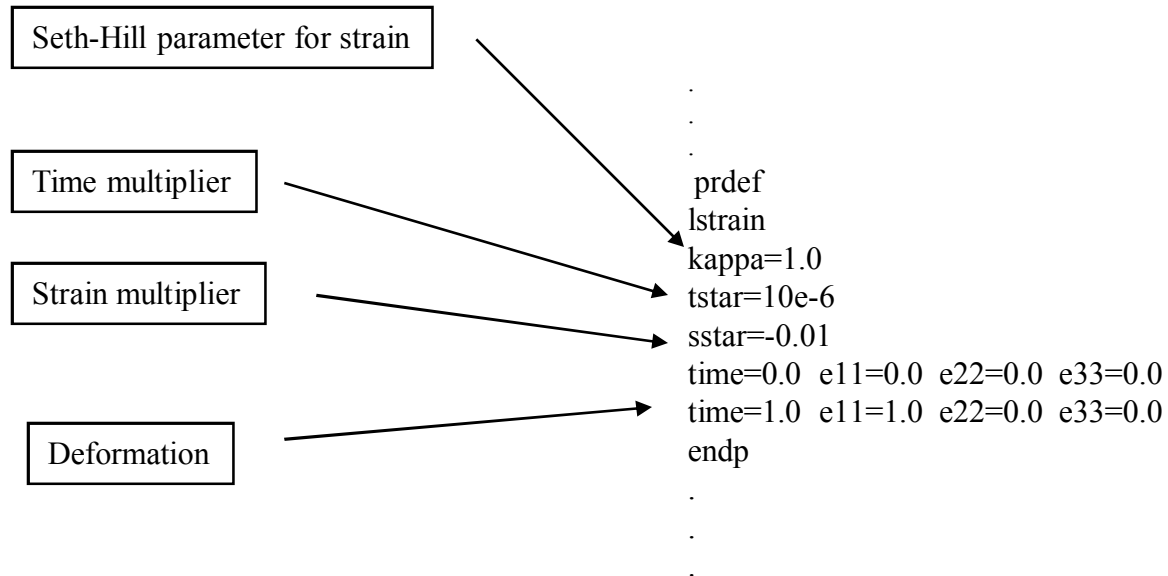
```
LIB_OBJJS = \  
    bcj/svpchk.o bcj/svpdam.o bcj/svpdrv.o bcj/svpex.o \  
    bcj/yldchk.o \  
    bfk/concck.o bfk/concxv.o bfk/condis.o bfk/conhro.o \  
    bfk/conmfs.o bfk/elivco.o bfk/elpco.o bfk/eoscoi.o \  
    bfk/eoscov.o bfk/eoscox.o bfk/gauss.o \  
    bfk/random.o bfk/splco.o bfk/splint.o \  
    epr/tepsts.o jms/eosjbi.o jms/eosjbl.o jms/eosjbv.o \  
    ges/alode.o ges/determ.o ges/fnroot.o ges/geschk.o \  
    ges/gesdrv.o ges/gesexv.o ges/gesphy.o ges/gesret.o \  
    ges/gesfld.o ABB/abbdv.o ABB/abbxv.o ABB/abbchk.o \  
    ABB/abbmod.o
```

Items in red are changed for your model

Mechanics Model Testing

- **PRDEF**

- User Manual for reference



Items in red are changed for your model



Summary

- **data/VP_data** – Material constants
- **src/reb/uinep.F** – Add 4 files
 - include/cth/m###p0a.h
 - include/cth/m###p0b.h
 - include/cth/m###p0c.h
 - include/cth/m###p0d.h
- **src/reb/uinchk.F** – Add 1 file
 - include/cth/m###p1.h
- **src/sas/uiniv.F** – Add 1 file
 - include/cth/m###p2.h
- **src/sas/elsg.F** – Add 1 file
 - include/cth/m###p3.h
- **src/mat/"material_name"** – Directory to contain material model
- **src/mat /"material_name"/*F** – Add material model files here
- **src/mat/objslst** – Modify object list to include material model files