/ / ? / / /

IN 39

325643

p 40

NASA Contractor Report 187470

Procedure for Implementation of Temperature-Dependent Mechanical Property Capability in the Engineering Analysis Language (EAL) System

David E. Glass
Analytical Services & Materials, Inc.
Hampton, VA

and

James C. Robinson
ODU Research Foundation
Norfolk, VA

Contract NAS1-18599
Cooperative Agreement NCC1-99
December 1990

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

SUMMARY

This paper presents a procedure to allow temperature dependent mechanical properties to be used in the Engineering Analysis Language (EAL) System for solid structural elements. The implementation of this procedure is accomplished by including a modular runstream in the main EAL runstream. The procedure is applicable for models with multiple materials and with anisotropic properties, and can easily be incorporated into an existing EAL runstream. The procedure is applicable for EAL elastic solid elements and is described in detail in the paper, followed by a description of the validation of the procedure. A listing of the EAL runstream used to validate the procedure is included in the Appendix.

INTRODUCTION

Finite element analysis is widely used to study various thermal and mechanical problems. One finite element program that is frequently used is the Engineering Analysis Language (EAL) System (ref. 1). EAL is a modular system of individual analysis processors which may be used in any appropriate sequence to perform a variety of analyses. In performing thermal analyses, EAL allows the use of temperature dependent thermal properties, i.e., thermal conductivity, specific heat, density, and emissivity. However, in the solution of thermal stress problems, EAL limits the use of temperature dependent mechanical properties to one- and two-dimensional isotropic materials. For orthotropic or anisotropic three-dimensional materials, EAL does not allow for the inclusion of temperature dependent mechanical properties, e.g., modulus of elasticity, shear modulus, coefficient of thermal expansion, and Poisson's ratio. In many thermal stress problems, mechanical properties may vary greatly in space and/or time, and thus it is necessary to include this temperature dependence to insure accurate results.

The purpose of this paper is to present a procedure to implement temperature dependent mechanical properties for solid structural elements in EAL. The implementation of this procedure is accomplished by including a few modifications to an existing EAL runstream. The procedure is applicable for models with multiple materials and with anisotropic properties, and is written for EAL S81, S61, and S41 elements, but can be extended to other elements. After a brief description of EAL, the implementation procedure is described, followed by a description of the procedure. A listing of the EAL runstream used to validate the procedure is included in the Appendix.

SYMBOLS

DNS8	name given to data set of temperature dependent densities
E	EAL processor used in structural analysis
E(x)	Young's modulus
E43	plate elements
EAL	Engineering Analysis Language
EI1	EAL processor
FLUX K81	EAL table generated by processor TAFP containing element temperatures
L	rod length
PROP BTAB	EAL name for table containing flexibility matrix
RECI	EAL function to calculate reciprocal
S41	4-node tetrahedron element
S61	6-node wedge element
S81	8-node isoparametric brick element
T(x)	temperature
T_0	rod temperature at x = 0
TAFP	EAL processor that calculates average element temperatures

TK81	name given to data set of average element temperatures
u(x)	displacement
x	coordinate in rod length direction
XNT1	interpolation function in EAL
*	EAL symbol preceding all executive control statements
$	signifies comment statement in EAL
!	signifies register action command (used for simple mathematics) in EAL

<u>Greek symbols</u>

σ	stress
β	coefficient in linear variation of elastic modulus with temperature
γ	coefficient in linear variation of thermal expansion with temperature
α	linear coefficient of thermal expansion

DESCRIPTION OF EAL

The Engineering Analysis Language (EAL) System is a high-level finite-element program which evolved from the SPAR computer program (ref. 2). The EAL System contains individual processors which communicate through a data base consisting of one or more libraries of data sets. A sequence of processor execution commands, which are written using an executive control language, is referred to as a runstream. Data sets typically contain data describing the finite-element model (e.g., material properties and nodal coordinates), as well as response information such as displacements, stresses, and temperatures. The EAL system uses flexible, FORTRAN-like statements which allow branching, looping, testing of data, and calling of runstreams (similar to calling a FORTRAN subroutine). The procedure presented here is a modular runstream which is called from within a main EAL runstream (similar to a FORTRAN main program with subroutines).

PROCEDURE FOR IMPLEMENTATION OF TEMPERATURE-DEPENDENT PROPERTIES

The runstream used to determine temperature dependent mechanical properties for solid structural elements and to calculate the structural response for two example problems is listed in the Appendix. The procedure for implementing the temperature dependent properties includes a runstream which creates tables of each property as a function of temperature for each material (called DATA TABL in the Appendix) and an additional runstream to calculate the temperature dependent properties for each element (called TMDP MAPR in the Appendix). In general, the runstream TMDP MAPR defines element temperatures (average of the element nodal temperatures), interpolates individual flexibility terms for element temperatures, and replaces individual element stiffness matrices stored in the data base. Implementing the procedure in EAL requires modifications to an existing EAL runstream to provide tables of properties as a function of temperature and a call to a newly created runstream (TMDP MAPR) which calculates the temperature dependent properties. Details of the procedure required to calculate the temperature dependent properties are discussed in this section.

To use temperature dependent mechanical properties, one must first provide a property table called PROP BTAB for each different material used (see runstream DATA TABL in the Appendix). These are the tables that are normally used to input constant mechanical properties, and no change is required in the structure of these tables. However, it is important that the density of each material be unique, since it is the density that is used in the present runstreams to differentiate between materials with different properties in this procedure. Hence, if two different materials happen to have the same density, it will be necessary to use “dummy densities” in the PROP BTAB tables. Since the density values from the PROP BTAB tables are not used in the structural analysis, there is no risk of using incorrect data. The actual density is obtained from a separate table of temperature dependent densities. After generating the PROP BTAB tables for each material, a unique table of temperature dependent properties is created for each of the non-zero terms. Thus, up to 31 temperature dependent tables may be required for each material since there are 31 entries in PROP BTAB. Although the modulus may be linear, EAL stores the flexibility coefficients which are related to the inverse of the moduli. Hence, it is imperative that enough data points be included so that a linear interpolation between two points produces a satisfactory approximation to the inverse of the modulus.

After generating the tables of temperature dependent properties (tables labeled CDEN, CA11, CA21,..., etc.), the runstream TMDP MAPR is called to insert the temperature dependent properties into the EAL data base between the execution of the processors E and EKS in the structural analysis. A listing of the runstream TMDP MAPR is included in the Appendix, and the steps followed are outlined in the flow chart in figure 1. In the description that follows, the term vector is used to denote a one-dimensional list of values.

The first step in the runstream TMDP MAPR is to obtain the element temperatures. This can be done in one of two methods. If a thermal analysis has been performed previously using EAL (and thus a STAT TEMP data set exists), the processor TAFP can be executed to calculate the element temperatures. Once the element temperatures are obtained with processor TAFP, a separate vector of element temperatures (called TK81 in the Appendix) must be created. This vector must be created if processor TAFP is used since the element temperatures that are created by processor TAFP are part of another table (i.e., FLUX K81 if K81 brick elements are used). (The method using TAFP is included in the runstream in the Appendix, however, it is commented out and not used since a thermal analysis was not performed prior to the structural analysis.) If a thermal analysis does not precede the structural analysis, the element temperatures can be input directly into a vector called TK81, or nodal temperatures can be input and averaged to obtain the element temperature vector TK81. The procedure used in the runstream listed in the Appendix assumes no prior thermal analysis has been performed.

The remaining portion of runstream TMDP MAPR should be transparent to the user, and is now discussed. If elements other than S81, S61 or S41 are used, or more than two materials are used, this portion of runstream TMDP MAPR would need to be modified. With the calculation of the element temperatures complete, the element temperatures are now used to interpolate the temperature dependent properties for each element. The EAL function XNT1 is used in TMDP MAPR to linearly interpolate between the element temperature (stored in TK81) and the tables with temperature dependent properties. This is done for every property for each material and results in a vector for each temperature dependent property for each material, assuming all the same type of elements are of the same material. Thus, if two materials are used with S81 elements, there will be a vector for each property (up to 31) assuming all S81 elements are material 1, and a second set of vectors assuming all S81 elements are material 2.

It is necessary to operate on the temperature dependent property vectors previously created to obtain one vector for each property for each element type. This vector will contain values for a given property as a function of temperature and material. To create this vector, the material dependent densities of each element type are first extracted from the EAL data base. This is done using the EI1 processor. After the densities are extracted, they need to be placed in a single vector (called DNS8 in the Appendix) containing the constant, but material dependent, densities. To illustrate this, consider a density vector extracted with the EI1 processor with 3 elements and 2 different materials represented by

$$\text{DNS8} \equiv \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_1 \end{bmatrix}$$

The density vector DNS8 is now used to obtain unit vectors for each material. The unit vector for material 2 will have a 1 for each element with material 2 and a 0 for each element of any other material. The unit vector for material 1 will have a 1 for each element with material 1 and a 0 for each element of any other material. To obtain the unit vector for material 2, subtract ρ_1 from each density in DNS8 as follows

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_1 \end{bmatrix} - \begin{bmatrix} \rho_1 \\ \rho_1 \\ \rho_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \rho_2 - \rho_1 \\ 0 \end{bmatrix}$$

Then, using the EAL function RECI, multiply each entry by its reciprocal, noting that in EAL, the reciprocal of 0 is 0.

$$\begin{bmatrix} 0 \\ \rho_2 - \rho_1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1/(\rho_2 - \rho_1) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

This is the unit vector for material 2. It has a 1 for each element of material 2, and a 0 for each element of material 1. To obtain the unit vector for material 1, subtract the unit vector for material 2 from the identity vector, as shown.

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

A similar procedure would follow if more that two materials are used.

These unit vectors are now used to obtain a vector of temperature dependent properties for each element. This is done by multiplying the unit vector for each material by the property vector that was generated by assuming all of that element type were of that material. To illustrate this, consider the coefficient of thermal expansion in the x direction as a function of temperature assuming all S81 elements are made of material 1. The coefficient of thermal expansion vector calculated previously for material 1 is of the form

$$\text{AL1C} \equiv \alpha_{x1} = \begin{bmatrix} \alpha_{x1}(T) & \alpha_{x1}(T) & \alpha_{x1}(T) \end{bmatrix}$$

Likewise, assuming all the elements are of material 2, the coefficient of thermal expansion vector is given by,

$$\text{AL1W} \equiv \alpha_{x2} = \begin{bmatrix} \alpha_{x2}(T) & \alpha_{x2}(T) & \alpha_{x2}(T) \end{bmatrix}$$

Now multiply the material 1 coefficient of thermal expansion vector by the material 1 unit vector to redefine AL1C as

$$\text{AL1C} \equiv \begin{bmatrix} \alpha_{x1}(T) & \alpha_{x1}(T) & \alpha_{x1}(T) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & \alpha_{x1}(T) & 0 \end{bmatrix}$$

and multiply the material 2 coefficient of thermal expansion vector by the material 2 unit vector to redefine AL1W as

$$\text{AL1W} \equiv \left[\alpha_{x2}(T) \quad \alpha_{x2}(T) \quad \alpha_{x2}(T) \right] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \left[\alpha_{x2}(T) \quad 0 \quad \alpha_{x2}(T) \right]$$

Now sum the individual coefficient of thermal expansion vectors as follows

$$\text{AL18} \equiv \left[0 \quad \alpha_{x1}(T) \quad 0 \right] + \left[\alpha_{x2}(T) \; 0 \; \alpha_{x2}(T) \right]$$

$$= \left[\alpha_{x2}(T) \quad \alpha_{x1}(T) \quad \alpha_{x2}(T) \right]$$

to obtain a vector of the temperature dependent coefficient of thermal expansion for all S81 elements. This same procedure is repeated for each of the non-zero properties in the flexibility matrix.

These property vectors are now put into a table (called NEW PROP in the Appendix) with 31 columns (the number of entries in the PROP BTAB table) and the number of rows equal to the number of elements of a given type. Entries that are zero need not be put in this table. This temperature dependent flexibility matrix is now embedded into the EAL data set by the EI1 processor. Now all of the S81 elements have the correct temperature dependent mechanical properties. If S61 or S41 elements are used in the runstream, the same procedure should be repeated for each element type. If elements other than S81, S61, or S41 are used, and temperature dependent properties are required, a similar procedure could be followed. However, the manner in which the properties are saved (and thus extracted and embedded) within EAL in the E state is different.

EXAMPLE PROBLEMS USED FOR PROCEDURE VALIDATION

To validate the procedure using temperature dependent mechanical properties, two example problems were considered. One example compares stresses in a thin plate modeled using E43 (two-dimensional plate) elements with those which assume a three-dimensional model consisting of S81 (three-dimensional solid) elements. A second example compares stresses in a rod calculated using S81 elements with a closed-form analytical solution. In both examples, the S81 elements used temperature dependent mechanical properties calculated using the present procedure.

Flat Plate Constrained Along All Four Edges and Subject to a Bi-linear Temperature Distribution

This validation compares stresses in a thin plate modeled with S81 solid elements with the same plate modeled with E43 plate elements. The plate is unconstrained in the through the thickness direction and is constrained in both the x and y in-plane directions. An illustration of the plate, including element numbers and nodal temperatures, is shown in figure 2. Two different materials are used in the analysis for generality. The reason for comparing an E43 element solution with an S81 element solution is that EAL allows temperature dependent isotropic properties for E43 elements. Thus, since the plate was modeled assuming isotropic S81 elements, one can compare the current temperature dependent mechanical properties solution using S81 elements with an established EAL solution using temperature dependent E43 elements. Although this does not allow the validation of all of the entries in the flexibility matrix (since EAL only allows isotropic temperature dependent E43 elements), it does allow the validation of the present procedure for isotropic properties. Since the procedure is the same for anisotropic properties, it is assumed that if the procedure works correctly for isotropic properties, the procedure also works correctly for anisotropic properties.

The temperature dependent properties used for the thin plate are given in Table 1, and the resulting stresses from the S81 and E43 solutions are compared in Table 2. The comparison of the stress results shown in Table 2 indicate that stresses calculated using S81 solid elements (temperature dependent properties obtained using the present procedure) compare quite favorably with those calculated using E43 plate elements (temperature dependence of the properties calculated internally in EAL). A comparison of stresses, where the temperature dependant properties (for S81 elements) are interpolated between entries 1000 °F apart as well as 100 °F apart, is also shown in Table 2. Young's modulus for the E43 elements is obtained directly in the EAL runstream, rather than from the inverse of the modulus as with S81 elements. Thus the E43 element solution should not be as sensitive to property interpolation as long as the properties vary nearly linearly with temperature. It is anticipated that the 100 °F interpolation would be more accurate for the S81 solution than the 1000 °F interpolation. However, since the stresses are compared only with another EAL solution, nothing can be said about the accuracy of the solution, only that the solutions from the two methods are consistent.

Rod Constrained at Both Ends and Subject to a Linear Temperature Distribution

The second method of procedure validation involves comparing stresses from a rod modeled using S81 elements with stresses from a corresponding closed-form analytical solution. The temperature distribution in the rod, which is constrained at both ends, is assumed to be a linear function of x as

$$T(x) = T_0 + \frac{x\,\Delta T}{L} \tag{1}$$

where x is the coordinate along the length of the rod, T_0 is the temperature at $x = 0$, ΔT is the temperature difference between the ends of the rod, and L is the length of the rod. In addition, the modulus of elasticity E and coefficient of thermal expansion α are assumed to be linear functions of temperature. Since the temperature is a linear function of x, both E and α are linear functions of x, given as

$$E(x) = E_0\,[1 + \beta\,T(x)] = E_0\left[1 + \beta\left\{T_0 + \frac{x\,\Delta T}{L}\right\}\right] \tag{2}$$

$$\alpha(x) = \alpha_0 \gamma T(x) = \alpha_0 \gamma \left[T_0 + \frac{x\,\Delta T}{L}\right] \tag{3}$$

where β and γ are constants. Using the symbolic mathematics program Mathematica (ref. 3), an analytical closed form solution is obtained for the displacement in the rod and is given as

$$u(x) = \frac{L\,\alpha_0\,\gamma\left[T_0 + \frac{x\,\Delta T}{L}\right]^3}{3\,\Delta T}$$

$$+ \frac{L\,\alpha_0\,(3\,T_0^2 + 3\,T_0\,\Delta T + \Delta T^2)\,\gamma \log\left[1 + \beta\left[T_0 + \frac{x\,\Delta T}{L}\right]\right]}{3\,\left[\log[1 + \beta\,T_0] - \log[1 + \beta\,(T_0 + \Delta T)]\right]}$$

$$+ \Big\{L\,\alpha_0\,\gamma\,\Big[\,T_0^3 \log\left(1 + \beta\,T_0\right) \quad + 3\,T_0^3\,\Delta T \log\left(1 + \beta\,T_0\right)$$

$$+ 3\,T_0\,\Delta T^2 \log\left(1 + \beta\,T_0\right) + \Delta T^3 \log\left(1 + \beta\,T_0\right)$$

$$- \;T_0^3 \log\left(1 + \beta\,(T_0 + \Delta T)\right)\,\Big]\Big\}$$

$$/\;\Big\{3\,\Delta T\;\Big[\log\left(1 + \beta\,T_0\right) - \log\left(1 + \beta\,(T_0 + \Delta T)\right)\,\Big]\Big\} \tag{4}$$

Though the above equation for the displacement is not valid for $\Delta T = 0$, it does approach the correct displacement in the limit as $\Delta T \rightarrow 0$. From the displacement, the stress in the rod is calculated from

$$\sigma(x) = E(x)\,\frac{du(x)}{dx} - E(x)\,\alpha(x)\,T(x) \tag{5}$$

to be

$$\sigma = \frac{E_0\,\alpha_0\,\beta\,\gamma\,\Delta T\,\left(3\,T_0^2 + 3\,T_0\,\Delta T + \Delta T^2\right)}{3\,\left[\log\left(1 + \beta\,T_0\right) - \log\left(1 + \beta\,(T_0 + \Delta T)\right)\,\right]} \tag{6}$$

Equation (6) shows that the stress is constant for the linear temperature, modulus, and coefficient of thermal expansion assumed in this case. Equation (6) is not valid at $\Delta T = 0$, i.e., for a constant temperature rod. However, in the limit as $\Delta T \rightarrow 0$, eq. (6) approaches the correct stress. The constant terms used in eqs. (4, 6) are

$$E_0 = 1.0 \times 10^6 \text{ psi} \tag{7a}$$

$$\alpha_0 = \frac{1.0 \times 10^{-6} \text{ in.}}{\text{in. } ^\circ\text{F}} \tag{7b}$$

$$T_0 = 200.0 \ ^\circ\text{F} \tag{7c}$$

$$\Delta T = 800.0 \ ^\circ\text{F} \tag{7d}$$

$$\beta = \frac{1}{1000 \ ^\circ\text{F}} \tag{7e}$$

$$\gamma = \frac{1}{1000 \ ^\circ\text{F}} \tag{7f}$$

$$L = 1 \text{ in.} \tag{7g}$$

The analytical displacement obtained from eq. (4) is compared with the EAL solution using 20 elements in Table 3. The mechanical properties in the EAL solution are interpolated from property data every 100 °F. Although the percent error between the analytical and EAL displacements is not zero everywhere, it is small, with the largest error being 4.15 percent. The analytical stress obtained from eq. (6) is compared with the EAL stress using 20 elements in Table 4. As can be seen in Table 4, when the rod is assumed to be a constant temperature of 200 °F or 1000 °F, eq. (6) and EAL give exactly the same stress. When the temperature distribution in the rod is linear, the EAL solution depends on the accuracy of interpolating the inverse of the modulus. If only the end point values are used for interpolation (i.e., 200 °F and 1000 °F), the EAL stress is -619.8 psi, compared with the exact value of -647.3 psi. However, when the table for the inverse of the modulus includes values at 100 °F intervals, the EAL stress is -646.7 psi. This increased accuracy is due to the fact that the inverse of the modulus is nearly linear between values in the table spaced at 100 °F intervals.

CONCLUDING REMARKS

A procedure to allow the Engineering Analysis Language (EAL) System to use temperature dependent mechanical properties has been presented and shown to be accurate. The procedure involves calling a runstream (subroutine) from the main EAL runstream to calculate the temperature dependent properties. Though the procedure described here has been written for the EAL S81, S61, and S41 elements, it can be modified for other elements. A modular approach has been taken in the development of the procedure, thus enabling an existing EAL runstream to be easily modified.

APPENDIX

This Appendix contains a listing of the EAL runstream used to calculate the stresses in the validation of the procedure assuming temperature dependent mechanical properties. The stresses can be calculated in either a rod or a plate with this runstream. The stresses in the plate can be calculated using either E43 plate elements or S81 isoparametric "brick" elements. The procedure used to calculate the temperature dependent mechanical properties is contained in a subroutine called TMDP MAPR. In the listing, two different materials are used, and are differentiated by "C" and "W". Comments are preceded by the symbol, $, and are generally in lower case.

```
DD 'NN
TITLE' TEST MODEL
'This is a test program. It will calculate the temperature dependent
' thermal stresses in a plate using either S81 elements or E43
' elements. Change the parameter GEOM to get either plate (E43) or
' solid (S81) elements. Since the E43 elements only allow temperature
' dependent isotropic properties, temperature dependent isotropic
' properties are also used for the S81 elements.
$
*ECHO 1,2,3,4
*ABORT 1
*XQT U1
*CM=150000
$
$*********************************************************************
$
$  This subroutine does the structural analysis
$
$*********************************************************************
$
* (29 STRU ANAL)                ZSTRU
$
*XQT TAB
$
  !GEOM = 0                     $ 0 for plates, 1 for solids
  !DIMN = 2                     $ 1 or 2 depending on dimension of problem desired
$ Determine temperature for TEMP NODE data set
  !TINC = 40.0                  $ temperature increment for each node (2-D problem)
  !TND1 = 40.0                  $ temperature of node 1 (2-D problem)
  !TX0 = 200.0                  $ temperature at one side of rod (1-D problem)
  !TXL = 1000.0                 $ temperature at other side of rod (1-D problem)
$
  !XLEN = 1.0                   $ length in x direction
  !YLEN = 1.0                   $ length in y direction
  !ZLEN = 0.1                   $ thickness
  !NX = 6                       $ number of nodes in x direction
  !NY = 7                       $ number of nodes in y direction (odd number)
  !NZ = 2                       $ number of nodes in z direction
```

```
  !NELX = NX - 1              $ number of elements in x direction
  !NELY = NY - 1              $ number of elements in y direction
  !NELZ = NZ - 1              $ number of elements in z direction
  !NNOD = NZ * NX * NY        $ total number of of nodes
 START "NNOD"  4 5 6
  !N1 = 1
  !N2 = NX * NY + 1
  !DX = XLEN/NELX
  !DY = YLEN/NELY
  !DZ = ZLEN/NELZ
  !XINC = 1
$ Generate the nodes
 JLOC
 FORMAT=1
  "N1"   0. 0. 0.   "XLEN" 0. 0.   "NX" "XINC" "NY"
   "NX"   0. "YLEN" 0.   "XLEN" "YLEN" 0.
  "N2"   0. 0. "DZ"   "XLEN" 0. "DZ"   "NX" "XINC" "NY"
   "NX"   0. "YLEN" "DZ"   "XLEN" "YLEN" "DZ"
$
*DCALL (29 STRU BCON)
$
MATC                          $ for plate elements
 1 1.0E+6 0.3 0.69 1.0E-6
 2 1.0E+6 0.3 0.69 1.0E-6
SA
 1 "ZLEN"
 NMAT=2
 2 "ZLEN"
$
*XQT AUS
*DCALL (29 DATA TABL)       $ insert properties as a function of temperature
$
*XQT DCU
 PRINT 1 MATC
 PRINT 1 NODA TEMP
$
*XQT ELD
  !IINC = 1
  !JINC = NX
  !KINC = NX * NY
  !NELY = NY - 1 / 2
  !N2 = NX * NELY + 1
*IF("GEOM" EQ 0): *JUMP 600
 S81
 NSECT = 1
 GROUP = 1
  1 "NELX" "NELY" "NELZ" "IINC" "JINC" "KINC" 0 1
 NSECT = 2
  "N2" "NELX" "NELY" "NELZ" "IINC" "JINC" "KINC" 0 1
*JUMP 610
*LABEL 600
$
 E43
 NSECT=1
  !J1 = 1
  !J2 = J1 + IINC
  !J3 = J2 + JINC
  !J4 = J1 + JINC
```

```
 "J1" "J2" "J3" "J4" 1 "NELX" "NELY"
NSECT=2
 !J1 = NX * NELY + 1
 !J2 = J1 + IINC
 !J3 = J2 + JINC
 !J4 = J1 + JINC
 "J1" "J2" "J3" "J4" 1 "NELX" "NELY"
$
*LABEL 610
$
*XQT U1
*TI (1 TD MATC 1 1)
   0.0   1.494E+7  0.4  2.00E-6
 1000.0   1.300E+7  0.3  3.26E-6
 2000.0   1.146E+7  0.2  5.28E-6
 3000.0   1.025E+7  0.1  6.31E-6
*TI (1 TD MATC 2 1)
   0.0   5.903E+7  1.054E-1  1.00E-6
 1000.0   3.712E+7  1.034E-1  2.26E-6
 2000.0   2.685E+7  1.046E-1  3.28E-6
 3000.0   2.103E+7  8.433E-2  4.31E-6
*TI (1 TD THICKNESS 1 1)
 "ZLEN"
 "ZLEN"
$
*XQT DCU
*IF("GEOM" EQ 1):   PRINT 1 DEF S81
$
*XQT E
 T=1.E-20  -.1E-2  0.01  0.01  40.  0.01  0.01  0.01
$
*IF("GEOM" EQ 1):  *DCALL (29 TMDP MAPR)  $ calculate temp. dep. material prop.
$
*XQT EKS
*IF("GEOM" EQ 1):  *JUMP 720
 RESET TLIB=1
 RESET N3T=1
*LABEL 720
$RESET SRFZ=0.0          $ to avoid imminent singularity error
$
*XQT SEQ                 $ joint elimination sequence
$
*XQT TAN
$
*XQT K
*ONLINE=2
*XQT DRSI
*ONLINE=1
$
*XQT EQNF                $ equivalent nodal forces
$
*XQT SSOL                $ displacements, reactions
$
*XQT VPRT
PRINT STAT DISP 1 1
PRINT STAT REAC 1 1
$
*XQT GSF
```

```
$
*XQT PSF
 DIV 1000. 1000. 1000. 1000.
$
*XQT ES
$
*XQT AUS
DEFI S=STAT DISP 1
GDSP=LTOG(S)
$
*XQT DCU
CHANGE 1 GDSP AUS 1 1 PATR DISP 1 1
PRINT 1 JLOC BTAB
TOC 1
$
*RETURN
*                              ZSTRU
$
$
$*************************************************************************
$
$  This subroutine calculates temperature dependent material properties
$
$*************************************************************************
$
* (29 TMDP MAPR)                    ZTMDP
$
$ Create a vector of element center temperatures and then interpolate
$   to find the corresponding material properties
*XQT AUS
  !NK81 = TOC NJ(1 S81 EFIL MASK MASK)
  !NK81 =
  !NELM = NX - 1
  !NELM = NY - 1 * NELM
  !NELM =
  !NY =
  !NX =
 TABLE(NI=1,NJ="NK81"): TK81 1
*IF("DIMN" EQ 2):  *JUMP 5132   $ vary temperatures in 1 direction
   !DELT = NY - 1.0
   !DELT = TXL - TX0 / DELT
   !NLY = NY - 1
   !INK = 1
*LABEL 7196
*IF("INK" GT "NLY"):  *JUMP 7197
   !I1 = INK - 1 * NELX + 1
   !I2 = INK * NELX
   !TELM = INK - 1.0 * DELT + TX0
   !TELN = INK * DELT + TX0
   !TELM = TELM + TELN / 2.0
  J="I1","I2": "TELM"
   !INK = INK + 1
*JUMP 7196
*LABEL 7197
*JUMP 5133
$
$
*LABEL 5132
```

```
$
 !ND = 1
*LABEL 900
*IF("ND" GT "NK81"):  *JUMP 910
 !I13 = DS,13,"ND",1(1,DEF,S81,18,8)
 !I14 = DS,14,"ND",1(1,DEF,S81,18,8)
 !I15 = DS,15,"ND",1(1,DEF,S81,18,8)
 !I16 = DS,16,"ND",1(1,DEF,S81,18,8)
 !TPND = I13 + I14 + I15 + I16 / 4 * TINC          $ average element temperature
*IF("TPND" GT 0.1): J="ND": "TPND"
*IF("TPND" LT 0.1): J="ND": "TND1"                 $ for constant temperature (i.e., TINC=0)
 !ND = ND + 1
*JUMP 900
*LABEL 910
*LABEL 5133
$
$
$  If one calculates nodal temperatures in EAL, use the following procedure to get element temperatures
$*XQT TAFP              $ to get element center temperatures
$ TEMP=STAT TEMP
$ COMPUTE ELEMENT FLUXES: K81
$
$*XQT DCU
$ PRINT 1 FLUX K81
$
$*XQT AUS
$   !NK81 = TOC NJ(1 S81 EFIL MASK MASK)
$ DEFI F1=FLUX K81 1 1
$ TABLE(NI=1,NJ="NK81"): TK81
$ TRAN(SOUR=F1,ILIM=1,JLIM="NK81",SSKI=2)
$
$ Now the element temperatures (TK81) have been embedded from one of the 2 methods, print them out
*XQT DCU
 PRINT 1 TK81
$
$Now use those element temperatures to interpolate and find properties of S81's
*XQT AUS
 DENC=XNT1(CDEN,TK81)        $ interpolate to find the density
 DENW=XNT1(WDEN,TK81)
 A11C=XNT1(CA11,TK81)        $ interpolate to find a11
 A11W=XNT1(WA11,TK81)
 A21C=XNT1(CA21,TK81)        $ interpolate to find a21
 A21W=XNT1(WA21,TK81)
 A22C=XNT1(CA22,TK81)        $ interpolate to find a22
 A22W=XNT1(WA22,TK81)
 A31C=XNT1(CA31,TK81)        $ interpolate to find a31
 A31W=XNT1(WA31,TK81)
 A32C=XNT1(CA32,TK81)        $ interpolate to find a32
 A32W=XNT1(WA32,TK81)
 A33C=XNT1(CA33,TK81)        $ interpolate to find a33
 A33W=XNT1(WA33,TK81)
 A44C=XNT1(CA44,TK81)        $ interpolate to find a44
 A44W=XNT1(WA44,TK81)
 A55C=XNT1(CA55,TK81)        $ interpolate to find a55
 A55W=XNT1(WA55,TK81)
 A66C=XNT1(CA66,TK81)        $ interpolate to find a66
 A66W=XNT1(WA66,TK81)
 AL1C=XNT1(CAL1,TK81)        $ interpolate to find alpha1
```

```
AL1W=XNT1(WAL1,TK81)
AL2C=XNT1(CAL2,TK81)          $ interpolate to find alpha2
AL2W=XNT1(WAL2,TK81)
AL3C=XNT1(CAL3,TK81)          $ interpolate to find alpha3
AL3W=XNT1(WAL3,TK81)
YXXC=XNT1(CYXX,TK81)          $ interpolate to find Yxx
YXXW=XNT1(WYXX,TK81)
YYYC=XNT1(CYYY,TK81)          $ interpolate to find Yyy
YYYW=XNT1(WYYY,TK81)
YZZC=XNT1(CYZZ,TK81)          $ interpolate to find Yzz
YZZW=XNT1(WYZZ,TK81)
YXYC=XNT1(CYXY,TK81)          $ interpolate to find Yxy
YXYW=XNT1(WYXY,TK81)
YXZC=XNT1(CYXZ,TK81)          $ interpolate to find Yxz
YXZW=XNT1(WYXZ,TK81)
YYZC=XNT1(CYYZ,TK81)          $ interpolate to find Yzy
YYZW=XNT1(WYYZ,TK81)
$
*XQT EI1                       $ Get the vector of material densities
EXTRACT
SOURCE = S81
CONT SPEC: SECTION 1          $ density only
CREATE DNS8 S81 1 1           $ vector containing densities of S81's
$
*XQT DCU
PRINT DNS8 S81 1 1            $ print the density vector
$
$ Set up unit vectors to be used in the calculation
*XQT AUS
 DEFI DN=DNS8 S81 1 1         $ material dependent densities
TABLE(NI=1,NJ="NK81"): CARB VECT: J=1,"NK81": "CRHO"
 DEFI CV=CARB VECT
V1 = SUM(DN,-1.0 CV)
DI = RECI(V1)
VT = PROD(V1,DI)              $ unit vector
TABLE(NI=1,NJ="NK81"): UNIT VECT: J=1,"NK81": 1.0
 DEFI UV=UNIT VECT
VC = SUM(VT,-1.0 UV)
VC = PROD(VC,-1.0 UV)         $ unit vector
$
$ Now multiply the unit vectors by the temperature dependent properties and sum to get temperature and
$  material dependent properties
*XQT AUS
DENC = PROD(DENC,VC)
DENW = PROD(DENW,VT)
DEN8 = SUM(DENC,DENW)         $ vector of temperature dependent densities
A11C = PROD(A11C,VC)
A11W = PROD(A11W,VT)
A118 = SUM(A11C,A11W)         $ vector of temperature dependent a11
A21C = PROD(A21C,VC)
A21W = PROD(A21W,VT)
A218 = SUM(A21C,A21W)         $ vector of temperature dependent a21
A22C = PROD(A22C,VC)
A22W = PROD(A22W,VT)
A228 = SUM(A22C,A22W)         $ vector of temperature dependent a22
A31C = PROD(A31C,VC)
A31W = PROD(A31W,VT)
A318 = SUM(A31C,A31W)         $ vector of temperature dependent a31
```

```
A32C = PROD(A32C,VC)
A32W = PROD(A32W,VT)
A328 = SUM(A32C,A32W)      $ vector of temperature dependent a32
A33C = PROD(A33C,VC)
A33W = PROD(A33W,VT)
A338 = SUM(A33C,A33W)      $ vector of temperature dependent a33
A44C = PROD(A44C,VC)
A44W = PROD(A44W,VT)
A448 = SUM(A44C,A44W)      $ vector of temperature dependent a44
A55C = PROD(A55C,VC)
A55W = PROD(A55W,VT)
A558 = SUM(A55C,A55W)      $ vector of temperature dependent a55
A66C = PROD(A66C,VC)
A66W = PROD(A66W,VT)
A668 = SUM(A66C,A66W)      $ vector of temperature dependent a66
AL1C = PROD(AL1C,VC)
AL1W = PROD(AL1W,VT)
AL18 = SUM(AL1C,AL1W)      $ vector of temperature dependent alpha1
AL2C = PROD(AL2C,VC)
AL2W = PROD(AL2W,VT)
AL28 = SUM(AL2C,AL2W)      $ vector of temperature dependent alpha2
AL3C = PROD(AL3C,VC)
AL3W = PROD(AL3W,VT)
AL38 = SUM(AL3C,AL3W)      $ vector of temperature dependent alpha3
YXXC = PROD(YXXC,VC)
YXXW = PROD(YXXW,VT)
YXX8 = SUM(YXXC,YXXW)      $ Yxx
YYYC = PROD(YYYC,VC)
YYYW = PROD(YYYW,VT)
YYY8 = SUM(YYYC,YYYW)      $ Yyy
YZZC = PROD(YZZC,VC)
YZZW = PROD(YZZW,VT)
YZZ8 = SUM(YZZC,YZZW)      $ Yzz
YXYC = PROD(YXYC,VC)
YXYW = PROD(YXYW,VT)
YXY8 = SUM(YXYC,YXYW)      $ Yxy
YXZC = PROD(YXZC,VC)
YXZW = PROD(YXZW,VT)
YXZ8 = SUM(YXZC,YXZW)      $ Yxz
YYZC = PROD(YYZC,VC)
YYZW = PROD(YYZW,VT)
YYZ8 = SUM(YYZC,YYZW)      $ Yyz
$
$Now insert each of these vectors into a dataset
TABLE(NI=31,NJ="NK81"): NEW PROP
 TRAN(SOUR=DEN8,ILIM=1,JLIM="NK81",SBAS=0,DBAS=0,DSKI=30)
 TRAN(SOUR=A118,ILIM=1,JLIM="NK81",SBAS=0,DBAS=1,DSKI=30)
 TRAN(SOUR=A218,ILIM=1,JLIM="NK81",SBAS=0,DBAS=2,DSKI=30)
 TRAN(SOUR=A228,ILIM=1,JLIM="NK81",SBAS=0,DBAS=3,DSKI=30)
 TRAN(SOUR=A318,ILIM=1,JLIM="NK81",SBAS=0,DBAS=4,DSKI=30)
 TRAN(SOUR=A328,ILIM=1,JLIM="NK81",SBAS=0,DBAS=5,DSKI=30)
 TRAN(SOUR=A338,ILIM=1,JLIM="NK81",SBAS=0,DBAS=6,DSKI=30)
 TRAN(SOUR=A448,ILIM=1,JLIM="NK81",SBAS=0,DBAS=10,DSKI=30)
 TRAN(SOUR=A558,ILIM=1,JLIM="NK81",SBAS=0,DBAS=15,DSKI=30)
 TRAN(SOUR=A668,ILIM=1,JLIM="NK81",SBAS=0,DBAS=21,DSKI=30)
 TRAN(SOUR=AL18,ILIM=1,JLIM="NK81",SBAS=0,DBAS=22,DSKI=30)
 TRAN(SOUR=AL28,ILIM=1,JLIM="NK81",SBAS=0,DBAS=23,DSKI=30)
 TRAN(SOUR=AL38,ILIM=1,JLIM="NK81",SBAS=0,DBAS=24,DSKI=30)
```

```
 TRAN(SOUR=YXX8,ILIM=1,JLIM="NK81",SBAS=0,DBAS=25,DSKI=30)
 TRAN(SOUR=YYY8,ILIM=1,JLIM="NK81",SBAS=0,DBAS=26,DSKI=30)
 TRAN(SOUR=YZZ8,ILIM=1,JLIM="NK81",SBAS=0,DBAS=27,DSKI=30)
 TRAN(SOUR=YXY8,ILIM=1,JLIM="NK81",SBAS=0,DBAS=28,DSKI=30)
 TRAN(SOUR=YXZ8,ILIM=1,JL M="NK81",SBAS=0,DBAS=29,DSKI=30)
 TRAN(SOUR=YYZ8,ILIM=1,JLIM="NK81",SBAS=0,DBAS=30,DSKI=30)
$
$Now embed the just created dataset of properties into EI1
*XQT EI1
 EMBED
 SOURCE = NEW PROP
 CONT SPEC: SECTION 1,31
 DEST = S81
$
*RETURN
*                          ZTMDP
$
$
$*************************************************************************
$
$  This subroutine sets the boundary conditions for the structural analysis
$
$*************************************************************************
$
* (29 STRU BCON)                ZBCON
$
CON=1
$
  !I1 = 1
 ZERO 2
*LABEL 200
*IF("I1" GT "NX"):  *JUMP 100
  !I2 = NX * NELY + I1
  !I3 = NX * NY + I1
  !I4 = NX * NY + I2
 "I1"
 "I2"
 "I3"
 "I4"
  !I1 = I1 + 1
*JUMP 200
*LABEL 100
$
ZERO 1 2 3 4 5 6: 1
*RETURN
*                          ZBCON
$
$ Define some of the constraints needed
$
$*************************************************************************
$
$  Create the tables of data
$
$*************************************************************************
$
* (29 DATA TABL)               ZDATA
$  Now input the structural data sets
$
```

```
$ Set up the NODA TEMP data set
 TABLE(NI=1,NJ="NNOD"): NODA TEMP 1
*IF("DIMN" EQ 2): *JUMP 5128
  !DELT = NY - 1.0
  !DELT = TXL - TX0 / DELT
  !INK = 1
*LABEL 7192
*IF("INK" GT "NY"): *JUMP 7193
  !I1 = INK - 1 * NX + 1
  !I2 = INK * NX
  !TELM = INK - 1.0 * DELT + TX0
 J="I1","I2": "TELM"   $ 1-D problem
  !INK = INK + 1
*JUMP 7192
*LABEL 7193
$
  !INK1 = 1
*LABEL 7194
*IF("INK1" GT "NY"): *JUMP 7195
  !I1 = INK - 1 * NX + 1
  !I2 = INK * NX
  !TELM = INK1 - 1.0 * DELT + TX0
 J="I1","I2": "TELM"
  !INK1 = INK1 + 1
  !INK = INK + 1
 *JUMP 7194
 *LABEL 7195
 *JUMP 5129
 $
 $
 *LABEL 5128
  !ND = 1
  !NN2 = NNOD/2
 *LABEL 210
 *IF("ND" GT "NN2"): *JUMP 200
   !TMPD = ND * TINC
 *IF("TMPD" EQ 0.0): !TMPD = TND1
  !ND =
  !TMPD =
  J="ND": "TMPD"   $ 2-D problem
   !ND = ND + 1
 *JUMP 210
 *LABEL 200
 *IF("ND" GT "NNOD"): *JUMP 220
  !TMPD = ND - NN2 * TINC
 *IF("TMPD" EQ 0.0): !TMPD = TND1
  J="ND":  "TMPD"
   !ND = ND + 1
 *JUMP 200
 *LABEL 220
 *LABEL 5129
 $
   !CRHO = 0.2
   !WRHO = 0.6
 $ These are tje property tables normally required by EAL.  These numbers will be replaced later.
  TABLE(NI=31,NJ=2): PROP BTAB 2 21
   J=1   $ Material 1
   "CRHO">
```

```
 2.0E-6>
 0.0000E-9  2.0000E-6>
 0.0000E-9  0.0000E-9  2.000E-6>
 0.0    0.0    0.0    1.0E-6>
 0.0    0.0    0.0    0.0    1.0E-6>
 0.0    0.0    0.0    0.0    0.0    1.0E-6>
 1.0E-6  1.0E-6  1.0E-6>
 1.0 1.0 1.0 1.0 1.0 1.0$
$
J=2   $ Material 2
 "WRHO">
 2.121E-8>
-6.364E-9  2.121E-8>
-6.364E-9  -6.364E-9  2.121E-8>
 0.0      0.0      0.0      5.605E-8>
 0.0      0.0      0.0      0.0      5.605E-8>
 0.0      0.0      0.0      0.0      0.0      5.605E-8>
 2.969E-6  2.969E-6  2.969E-6>
 1.0 1.0 1.0 1.0 1.0 1.0$
$
$Set up tables for temperature dependent properties of material 1
TABLE(NI=2,NJ=2):  CDEN: I=1,2       $ density
   J=1:     0.0    "CRHO"
   J=2: 3000.0     "CRHO"
TABLE(NI=2,NJ=20):  CA11: I=1,2      $ a11 = 1/E1
   J=1:     0.0     6.694E-8
   J=2:   100.0     6.782E-8
   J=3:   200.0     6.872E-8
   J=4:   300.0     6.965E-8
   J=5:   400.0     7.060E-8
   J=6:   500.0     7.158E-8
   J=7:   600.0     7.259E-8
   J=8:   700.0     7.363E-8
   J=9:   800.0     7.469E-8
   J=10:  900.0     7.579E-8
   J=11: 1000.0     7.694E-8
   J=12: 1100.0     7.785E-8
   J=13: 1200.0     7.879E-8
   J=14: 1300.0     7.976E-8
   J=15: 1400.0     8.075E-8
   J=16: 1500.0     8.177E-8
   J=17: 1600.0     8.281E-8
   J=18: 1700.0     8.388E-8
   J=19: 2000.00    8.724E-8
   J=20: 3000.00    9.754E-8
TABLE(NI=2,NJ=20):  CA21: I= 1,2     $ a21 = -v12/E1
   J=1:     0.0    -2.678E-8
   J=2:   100.0    -2.636E-8
   J=3:   200.0    -2.595E-8
   J=4:   300.0    -2.555E-8
   J=5:   400.0    -2.517E-8
   J=6:   500.0    -2.479E-8
   J=7:   600.0    -2.443E-8
   J=8:   700.0    -2.408E-8
   J=9:   800.0    -2.373E-8
   J=10:  900.0    -2.340E-8
   J=11: 1000.0    -2.308E-8
   J=12: 1100.0    -2.236E-8
```

```
   J=13: 1200.0    -2.168E-8
   J=14: 1300.0    -2.104E-8
   J=15: 1400.0    -2.044E-8
   J=16: 1500.0    -1.987E-8
   J=17: 1600.0    -1.934E-8
   J=18: 1700.0    -1.883E-8
   J=19: 2000.00   -1.745E-8
   J=20: 3000.00   -9.754E-9
TABLE(NI=2,NJ=20):  CA22: I=1,2        $ a22 = 1/E2
   J=1:     0.0     6.694E-8
   J=2:   100.0     6.782E-8
   J=3:   200.0     6.872E-8
   J=4:   300.0     6.965E-8
   J=5:   400.0     7.060E-8
   J=6:   500.0     7.158E-8
   J=7:   600.0     7.259E-8
   J=8:   700.0     7.363E-8
   J=9:   800.0     7.469E-8
   J=10:  900.0     7.579E-8
   J=11: 1000.0     7.694E-8
   J=12: 1100.0     7.785E-8
   J=13: 1200.0     7.879E-8
   J=14: 1300.0     7.976E-8
   J=15: 1400.0     8.075E-8
   J=16: 1500.0     8.177E-8
   J=17: 1600.0     8.281E-8
   J=18: 1700.0     8.388E-8
   J=19: 2000.00    8.724E-8
   J=20: 3000.00    9.754E-8
TABLE(NI=2,NJ=20):  CA31: I=1,2        $ a31 = -v13/E1
   J=1:     0.0    -2.678E-8
   J=2:   100.0    -2.636E-8
   J=3:   200.0    -2.595E-8
   J=4:   300.0    -2.555E-8
   J=5:   400.0    -2.517E-8
   J=6:   500.0    -2.479E-8
   J=7:   600.0    -2.443E-8
   J=8:   700.0    -2.408E-8
   J=9:   800.0    -2.373E-8
   J=10:  900.0    -2.340E-8
   J=11: 1000.0    -2.308E-8
   J=12: 1100.0    -2.236E-8
   J=13: 1200.0    -2.168E-8
   J=14: 1300.0    -2.104E-8
   J=15: 1400.0    -2.044E-8
   J=16: 1500.0    -1.987E-8
   J=17: 1600.0    -1.934E-8
   J=18: 1700.0    -1.883E-8
   J=19: 2000.00   -1.745E-8
   J=20: 3000.00   -9.754E-9
TABLE(NI=2,NJ=20):  CA32: I=1,2        $ a32 = -v23/E2
   J=1:     0.0    -2.678E-8
   J=2:   100.0    -2.636E-8
   J=3:   200.0    -2.595E-8
   J=4:   300.0    -2.555E-8
   J=5:   400.0    -2.517E-8
   J=6:   500.0    -2.479E-8
   J=7:   600.0    -2.443E-8
```

```
   J=8:  700.0   -2.408E-8
   J=9:  800.0   -2.373E-8
   J=10: 900.0   -2.340E-8
   J=11: 1000.0  -2.308E-8
   J=12: 1100.0  -2.236E-8
   J=13: 1200.0  -2.168E-8
   J=14: 1300.0  -2.104E-8
   J=15: 1400.0  -2.044E-8
   J=16: 1500.0  -1.987E-8
   J=17: 1600.0  -1.934E-8
   J=18: 1700.0  -1.883E-8
   J=19: 2000.00 -1.745E-8
   J=20: 3000.00 -9.754E-9
TABLE(NI=2,NJ=20):  CA33: I=1,2      $ a33 = 1/E3
   J=1:    0.0   6.694E-8
   J=2:  100.0   6.782E-8
   J=3:  200.0   6.872E-8
   J=4:  300.0   6.965E-8
   J=5:  400.0   7.060E-8
   J=6:  500.0   7.158E-8
   J=7:  600.0   7.259E-8
   J=8:  700.0   7.363E-8
   J=9:  800.0   7.469E-8
   J=10: 900.0   7.579E-8
   J=11: 1000.0  7.694E-8
   J=12: 1100.0  7.785E-8
   J=13: 1200.0  7.879E-8
   J=14: 1300.0  7.976E-8
   J=15: 1400.0  8.075E-8
   J=16: 1500.0  8.177E-8
   J=17: 1600.0  8.281E-8
   J=18: 1700.0  8.388E-8
   J=19: 2000.00 8.724E-8
   J=20: 3000.00 9.754E-8
TABLE(NI=2,NJ=20):  CA44: I=1,2      $ a44 = 1/G23
   J=1:    0.0   1.874E-7
   J=2:  100.0   1.886E-7
   J=3:  200.0   1.898E-7
   J=4:  300.0   1.910E-7
   J=5:  400.0   1.922E-7
   J=6:  500.0   1.935E-7
   J=7:  600.0   1.948E-7
   J=8:  700.0   1.960E-7
   J=9:  800.0   1.973E-7
   J=10: 900.0   1.987E-7
   J=11: 1000.0  2.000E-7
   J=12: 1100.0  2.009E-7
   J=13: 1200.0  2.018E-7
   J=14: 1300.0  2.027E-7
   J=15: 1400.0  2.036E-7
   J=16: 1500.0  2.046E-7
   J=17: 1600.0  2.055E-7
   J=18: 1700.0  2.065E-7
   J=19: 2000.00 2.094E-7
   J=20: 3000.00 2.146E-7
TABLE(NI=2,NJ=20);  CA55: I=1,2      $ a55 = 1/G31
   J=1:    0.0   1.874E-7
   J=2:  100.0   1.886E-7
```

```
   J=3:  200.0     1.898E-7
   J=4:  300.0     1.910E-7
   J=5:  400.0     1.922E-7
   J=6:  500.0     1.935E-7
   J=7:  600.0     1.948E-7
   J=8:  700.0     1.960E-7
   J=9:  800.0     1.973E-7
   J=10:  900.0     1.987E-7
   J=11: 1000.0     2.000E-7
   J=12: 1100.0     2.009E-7
   J=13: 1200.0     2.018E-7
   J=14: 1300.0     2.027E-7
   J=15: 1400.0     2.036E-7
   J=16: 1500.0     2.046E-7
   J=17: 1600.0     2.055E-7
   J=18: 1700.0     2.065E-7
   J=19:  2000.00    2.094E-7
   J=20:  3000.00    2.146E-7
TABLE(NI=2,NJ=20):  CA66: I=1,2        $ a66 = 1/G12
   J=1:     0.0    1.874E-7
   J=2:  100.0     1.886E-7
   J=3:  200.0     1.898E-7
   J=4:  300.0     1.910E-7
   J=5:  400.0     1.922E-7
   J=6:  500.0     1.935E-7
   J=7:  600.0     1.948E-7
   J=8:  700.0     1.960E-7
   J=9:  800.0     1.973E-7
   J=10:  900.0     1.987E-7
   J=11: 1000.0     2.000E-7
   J=12: 1100.0     2.009E-7
   J=13: 1200.0     2.018E-7
   J=14: 1300.0     2.027E-7
   J=15: 1400.0     2.036E-7
   J=16: 1500.0     2.046E-7
   J=17: 1600.0     2.055E-7
   J=18: 1700.0     2.065E-7
   J=19:  2000.00    2.094E-7
   J=20:  3000.00    2.146E-7
TABLE(NI=2,NJ=4):  CAL1: I=1,2       $ alpha1
   J=1:     0.0   2.00E-6
   J=2:  1000.0   3.26E-6
   J=3:  2000.0   5.28E-6
   J=4:  3000.0   6.31E-6
TABLE(NI=2,NJ=4):  CAL2: I=1,2       $ alpha2
   J=1:   000.0   2.00E-6
   J=2:  1000.0   3.26E-6
   J=3:  2000.0   5.28E-6
   J=4:  3000.0   6.31E-6
TABLE(NI=2,NJ=4):  CAL3: I=1,2       $ alpha3
   J=1:   000.0   2.00E-6
   J=2:  1000.0   3.26E-6
   J=3:  2000.0   5.28E-6
   J=4:  3000.0   6.31E-6
TABLE(NI=2,NJ=2):  CYXX: I=1,2       $ Yxx
   J=1:    0.0    1.0
   J=2: 30000.0    1.0
TABLE(NI=2,NJ=2):  CYYY: I=1,2       $ Yyy
```

```
   J=1:     0.0    1.0
   J=2: 30000.0    1.0
 TABLE(NI=2,NJ=2):  CYZZ: I=1,2       $ Yzz
   J=1:     0.0    1.0
   J=2: 30000.0    1.0
 TABLE(NI=2,NJ=2):  CYXY: I=1,2       $ Yxy
   J=1:     0.0    1.0
   J=2: 30000.0    1.0
 TABLE(NI=2,NJ=2):  CYXZ: I=1,2       $ Yxz
   J=1:     0.0    1.0
   J=2: 30000.0    1.0
 TABLE(NI=2,NJ=2):  CYYZ: I=1,2       $ Yyz
   J=1:     0.0    1.0
   J=2: 30000.0    1.0
$
$Set up tables for temperature dependent properties of material 2
 TABLE(NI=2,NJ=2):  WDEN: I=1,2       $ density
   J=1:    0.0    "WRHO"
   J=2: 3000.0    "WRHO"
 TABLE(NI=2,NJ=22):  WA11: I=1,2      $ a11 = 1/E1
  J= 1:      .0   1.694E-08
  J= 2:   100.0   1.759E-08
  J= 3:   200.0   1.830E-08
  J= 4:   300.0   1.906E-08
  J= 5:   400.0   1.989E-08
  J= 6:   500.0   2.080E-08
  J= 7:   600.0   2.179E-08
  J= 8:   700.0   2.289E-08
  J= 9:   800.0   2.410E-08
  J=10:   900.0   2.544E-08
  J=11:  1000.0   2.694E-08
  J=12:  1100.0   2.771E-08
  J=13:  1200.0   2.852E-08
  J=14:  1300.0   2.938E-08
  J=15:  1400.0   3.029E-08
  J=16:  1500.0   3.126E-08
  J=17:  1600.0   3.230E-08
  J=18:  1700.0   3.341E-08
  J=19:  1800.0   3.459E-08
  J=20:  1900.0   3.587E-08
  J=21:  2000.0   3.724E-08
  J=22: 3000.00   4.754E-8
 TABLE(NI=2,NJ=22):  WA21: I=1,2      $ a21 = -v12/E1
  J= 1:      .0  -1.786E-09
  J= 2:   100.0  -1.852E-09
  J= 3:   200.0  -1.924E-09
  J= 4:   300.0  -2.001E-09
  J= 5:   400.0  -2.085E-09
  J= 6:   500.0  -2.176E-09
  J= 7:   600.0  -2.276E-09
  J= 8:   700.0  -2.385E-09
  J= 9:   800.0  -2.505E-09
  J=10:   900.0  -2.638E-09
  J=11:  1000.0  -2.786E-09
  J=12:  1100.0  -2.867E-09
  J=13;  1200.0  -2.954E-09
  J=14:  1300.0  -3.046E-09
  J=15:  1400.0  -3.144E-09
```

J=16: 1500.0 -3.248E-09
J=17: 1600.0 -3.360E-09
J=18: 1700.0 -3.479E-09
J=19: 1800.0 -3.607E-09
J=20: 1900.0 -3.745E-09
J=21: 2000.0 -3.895E-09
J=22: 3000.00 -4.009E-9
TABLE(NI=2,NJ=22): WA22: I=1,2 $ a22 = 1/E_2
J= 1: .0 1.694E-08
J= 2: 100.0 1.759E-08
J= 3: 200.0 1.830E-08
J= 4: 300.0 1.906E-08
J= 5: 400.0 1.989E-08
J= 6: 500.0 2.080E-08
J= 7: 600.0 2.179E-08
J= 8: 700.0 2.289E-08
J= 9: 800.0 2.410E-08
J=10: 900.0 2.544E-08
J=11: 1000.0 2.694E-08
J=12: 1100.0 2.771E-08
J=13: 1200.0 2.852E-08
J=14: 1300.0 2.938E-08
J=15: 1400.0 3.029E-08
J=16: 1500.0 3.126E-08
J=17: 1600.0 3.230E-08
J=18: 1700.0 3.341E-08
J=19: 1800.0 3.459E-08
J=20: 1900.0 3.587E-08
J=21: 2000.0 3.724E-08
J=22: 3000.00 4.754E-8
TABLE(NI=2,NJ=22): WA31: I=1,2 $ a31 = -ν_{13}/E_1
J= 1: .0 -1.786E-09
J= 2: 100.0 -1.852E-09
J= 3: 200.0 -1.924E-09
J= 4: 300.0 -2.001E-09
J= 5: 400.0 -2.085E-09
J= 6: 500.0 -2.176E-09
J= 7: 600.0 -2.276E-09
J= 8: 700.0 -2.385E-09
J= 9: 800.0 -2.505E-09
J=10: 900.0 -2.638E-09
J=11: 1000.0 -2.786E-09
J=12: 1100.0 -2.867E-09
J=13: 1200.0 -2.954E-09
J=14: 1300.0 -3.046E-09
J=15: 1400.0 -3.144E-09
J=16: 1500.0 -3.248E-09
J=17: 1600.0 -3.360E-09
J=18: 1700.0 -3.479E-09
J=19: 1800.0 -3.607E-09
J=20: 1900.0 -3.745E-09
J=21: 2000.0 -3.895E-09
J=22: 3000.00 -4.009E-9
TABLE(NI=2,NJ=22): WA32: I=1,2 $ a32 = -ν_{23}/E_2
J= 1: .0 -1.786E-09
J= 2: 100.0 -1.852E-09
J= 3: 200.0 -1.924E-09
J= 4: 300.0 -2.001E-09

```
  J= 5:   400.0  -2.085E-09
  J= 6:   500.0  -2.176E-09
  J= 7:   600.0  -2.276E-09
  J= 8:   700.0  -2.385E-09
  J= 9:   800.0  -2.505E-09
  J=10:   900.0  -2.638E-09
  J=11:  1000.0  -2.786E-09
  J=12:  1100.0  -2.867E-09
  J=13:  1200.0  -2.954E-09
  J=14:  1300.0  -3.046E-09
  J=15:  1400.0  -3.144E-09
  J=16:  1500.0  -3.248E-09
  J=17:  1600.0  -3.360E-09
  J=18:  1700.0  -3.479E-09
  J=19:  1800.0  -3.607E-09
  J=20:  1900.0  -3.745E-09
  J=21:  2000.0  -3.895E-09
 J=22: 3000.00  -4.009E-9
TABLE(NI=2,NJ=22):  WA33; I=1,2       $ a33 = 1/E3
   J= 1:      .0  1.694E-08
  J= 2:   100.0  1.759E-08
  J= 3:   200.0  1.830E-08
  J= 4:   300.0  1.906E-08
  J= 5:   400.0  1.989E-08
  J= 6:   500.0  2.080E-08
  J= 7:   600.0  2.179E-08
  J= 8:   700.0  2.289E-08
  J= 9:   800.0  2.410E-08
  J=10:   900.0  2.544E-08
  J=11:  1000.0  2.694E-08
  J=12:  1100.0  2.771E-08
  J=13:  1200.0  2.852E-08
  J=14:  1300.0  2.938E-08
  J=15:  1400.0  3.029E-08
  J=16:  1500.0  3.126E-08
  J=17:  1600.0  3.230E-08
  J=18:  1700.0  3.341E-08
  J=19:  1800.0  3.459E-08
  J=20:  1900.0  3.587E-08
  J=21:  2000.0  3.724E-08
   J=22: 3000.00   4.754E-8
TABLE(NI=2,NJ=22):  WA44: I=1,2       $ a44 = 1/G23
   J= 1:      .0  3.7450E-08
  J= 2:   100.0  3.8889E-08
  J= 3:   200.0  4.0443E-08
  J= 4:   300.0  4.2127E-08
  J= 5:   400.0  4.3957E-08
  J= 6:   500.0  4.5953E-08
  J= 7:   600.0  4.8138E-08
  J= 8:   700.0  5.0543E-08
  J= 9:   800.0  5.3200E-08
  J=10:   900.0  5.6151E-08
  J=11:  1000.0  5.9450E-08
  J=12:  1100.0  6.1147E-08
  J=13:  1200.0  6.2943E-08
  J=14:  1300.0  6.4848E-08
  J=15:  1400.0  6.6872E-08
  J=16:  1500.0  6.9026E-08
```

```
J=17:  1600.0  7.1324E-08
J=18:  1700.0  7.3780E-08
J=19:  1800.0  7.6411E-08
J=20:  1900.0  7.9237E-08
J=21:  2000.0  8.2280E-08
 J=22: 3000.00   1.031E-7
TABLE(NI=2,NJ=22):  WA55: I=1,2        $ a55 = 1/G31
J= 1:      .0  3.7450E-08
J= 2:   100.0  3.8889E-08
J= 3:   200.0  4.0443E-08
J= 4:   300.0  4.2127E-08
J= 5:   400.0  4.3957E-08
J= 6:   500.0  4.5953E-08
J= 7:   600.0  4.8138E-08
J= 8:   700.0  5.0543E-08
J= 9:   800.0  5.3200E-08
J=10:   900.0  5.6151E-08
J=11:  1000.0  5.9450E-08
J=12:  1100.0  6.1147E-08
J=13:  1200.0  6.2943E-08
J=14:  1300.0  6.4848E-08
J=15:  1400.0  6.6872E-08
J=16:  1500.0  6.9026E-08
J=17:  1600.0  7.1324E-08
J=18:  1700.0  7.3780E-08
J=19:  1800.0  7.6411E-08
J=20:  1900.0  7.9237E-08
J=21:  2000.0  8.2280E-08
J=22: 3000.00   1.031E-7
TABLE(NI=2,NJ=22):  WA66: I=1,2        $ a66 = 1/G12
J= 1:      .0  3.7450E-08
J= 2:   100.0  3.8889E-08
J= 3:   200.0  4.0443E-08
J= 4:   300.0  4.2127E-08
J= 5:   400.0  4.3957E-08
J= 6:   500.0  4.5953E-08
J= 7:   600.0  4.8138E-08
J= 8:   700.0  5.0543E-08
J= 9:   800.0  5.3200E-08
J=10:   900.0  5.6151E-08
J=11:  1000.0  5.9450E-08
J=12:  1100.0  6.1147E-08
J=13:  1200.0  6.2943E-08
J=14:  1300.0  6.4848E-08
J=15:  1400.0  6.6872E-08
J=16:  1500.0  6.9026E-08
J=17:  1600.0  7.1324E-08
J=18:  1700.0  7.3780E-08
J=19:  1800.0  7.6411E-08
J=20:  1900.0  7.9237E-08
J=21:  2000.0  8.2280E-08
 J=22: 3000.00   1.031E-7
TABLE(NI=2,NJ=4):  WAL1: I=1,2        $ alpha1
  J=1:     0.0  1.00E-6
  J=2:  1000.0   2.26E-6
  J=3:  2000.0   3.28E-6
  J=4:  3000.0   4.31E-6
TABLE(NI=2,NJ=4):  WAL2: I=1,2        $ alpha2
```

```
    J=1:   000.0   1.00E-6
    J=2:  1000.0   2.26E-6
    J=3:  2000.0   3.28E-6
    J=4:  3000.0   4.31E-6
 TABLE(NI=2,NJ=4):  WAL3: I=1,2       $ alpha3
    J=1:   000.0   1.00E-6
    J=2:  1000.0   2.26E-6
    J=3:  2000.0   3.28E-6
    J=4:  3000.0   4.31E-6
 TABLE(NI=2,NJ=2):  WYXX: I=1,2       $ Yxx
    J=1:     0.0   1.0
    J=2: 30000.0   1.0
 TABLE(NI=2,NJ=2):  WYYY: I=1,2       $ Yyy
    J=1:     0.0   1.0
    J=2: 30000.0   1.0
 TABLE(NI=2,NJ=2):  WYZZ: I=1,2       $ Yzz
    J=1:     0.0   1.0
    J=2: 30000.0   1.0
 TABLE(NI=2,NJ=2):  WYXY: I=1,2       $ Yxy
    J=1:     0.0   1.0
    J=2: 30000.0   1.0
 TABLE(NI=2,NJ=2):  WYXZ: I=1,2       $ Yxz
    J=1:     0.0   1.0
    J=2: 30000.0   1.0
 TABLE(NI=2,NJ=2):  WYYZ: I=1,2       $ Yyz
    J=1:     0.0   1.0
    J=2: 30000.0   1.0
$
$
$ Free up space
  !LINT = FREE ( )
  !LTOT = FREE ( )
  !L0 = FREE ( )
  !L1 = FREE ( )
  !L2 = FREE ( )
  !L3 = FREE ( )
  !L4 = FREE ( )
  !FLX = FREE ( )
  !MTCC = FREE ( )
$
*RETURN
*                         ZDATA
$
*XQT AUS
$
*DCALL (29 STRU ANAL)
*XQT EXIT
```

REFERENCES

1. Whetstone, W. D.: EISI-EAL Engineering Analysis Language Reference Manual - EISI-EAL System Level 2091. Engineering Information Systems, Inc. July 1983.

2 Whetstone, W. D.: SPAR Structural Analysis System Reference Manual - System Level 13A. Volume I: Program Execution. NASA CR-158970-1, 1978.

3. Wolfram, S.: Mathematica for the Macintosh, Wolfram Research, Inc., Champaign, IL, 1989.

Table 1 - Temperature Dependent Properties Used in S81 and E43 Elements

Temperature	E, psi	ν	α, in./in.°F
		Material 1	
0.0	1.494×10^{7}	0.4000	2.00×10^{-6}
1000.0	1.300×10^{7}	0.3000	3.26×10^{-6}
2000.0	1.146×10^{7}	0.2000	5.28×10^{-6}
		Material 2	
0.0	5.903×10^{7}	0.1054	1.00×10^{-6}
1000.0	3.712×10^{7}	0.1034	2.26×10^{-6}
2000.0	2.685×10^{7}	0.1046	3.28×10^{-6}

Table 2 - Comparison of Plate Stresses in the y Direction for S81 and E43 Elements

Element	σ_{S81}, psi [a]	σ_{S81}, psi [b]	σ_{E43}, psi
1	-43.6	-43.9	-43.2
2	-40.2	-40.6	-40.2
3	-42.2	-42.6	-42.1
4	-45.1	-45.4	-45.0
5	-53.3	-53.7	-53.0
6	-41.4	-41.8	-41.3
7	-42.0	-42.4	-41.8
8	-42.8	-43.1	-42.6
9	-47.1	-47.5	-46.9
10	-51.1	-51.5	-51.0
11	-40.9	-41.3	-40.9
12	-42.1	-42.4	-41.7
13	-43.5	-43.9	-43.3
14	-47.2	-47.5	-46.8
15	-50.7	-51.1	-50.7
16	-40.9	-41.2	-40.8
17	-41.0	-41.4	-40.8
18	-45.9	-46.2	-45.7
19	-45.7	-46.0	-45.5
20	-50.9	-51.4	-50.8
21	-36.0	-36.3	-35.6
22	-43.9	-44.3	-43.9
23	-49.9	-50.3	-49.8
24	-48.5	-48.9	-48.5
25	-46.1	-46.5	-45.7
26	-29.1	-29.3	-28.6
27	-49.1	-49.6	-49.1
28	-53.1	-53.6	-53.2
29	-53.9	-54.4	-54.0
30	-39.2	-39.4	-38.6

a Material properties interpolated over 1000 °F interval
b Material properties interpolated over 100 °F interval

Table 3 - Comparison of Analytical and EAL Displacements in a 1-D Rod

Position, in.	EAL displacement, in.	Eq. (4) displacement, in.	% error
0.00	0.0	0.0	0.00
0.05	-0.231×10^{-4}	-0.241×10^{-4}	4.15
0.10	-0.448×10^{-4}	-0.464×10^{-4}	3.45
0.15	-0.649×10^{-4}	-0.668×10^{-4}	2.84
0.20	-0.831×10^{-4}	-0.851×10^{-4}	2.35
0.25	-0.994×10^{-4}	-0.101×10^{-3}	1.58
0.30	-0.114×10^{-3}	-0.115×10^{-3}	0.87
0.35	-0.125×10^{-3}	-0.127×10^{-3}	1.57
0.40	-0.135×10^{-3}	-0.136×10^{-3}	0.74
0.45	-0.142×10^{-3}	-0.142×10^{-3}	0.00
0.50	-0.146×10^{-3}	-0.146×10^{-3}	0.00
0.55	-0.147×10^{-3}	-0.147×10^{-3}	0.00
0.60	-0.145×10^{-3}	-0.145×10^{-3}	0.00
0.65	-0.140×10^{-3}	-0.139×10^{-3}	0.72
0.70	-0.131×10^{-3}	-0.130×10^{-3}	0.77
0.75	-0.119×10^{-3}	-0.118×10^{-3}	0.85
0.80	-0.104×10^{-3}	-0.102×10^{-3}	1.96
0.85	-0.839×10^{-4}	-0.826×10^{-4}	1.57
0.90	-0.602×10^{-4}	-0.591×10^{-4}	1.86
0.95	-0.323×10^{-4}	-0.317×10^{-4}	1.89
1.0	0.0	0.0	0.00

Table 4 - Comparison of Analytical and EAL Stresses in a 1-D Rod

T_0,°F	T_L,°F	σ_{EAL}, psi	$\sigma_{Eq. (6)}$, psi	% error
200.0	200.0	-48.0	-48.0	0.0
1000.0	1000.0	-2000.0	-2000.0	0.0
200.0	1000.0	-619.8[a]	-647.3	4.25
200.0	1000.0	-646.7[b]	-647.3	0.09

a Material properties interpolated between endpoint temperatures
b Material properties interpolated over 100 °F interval

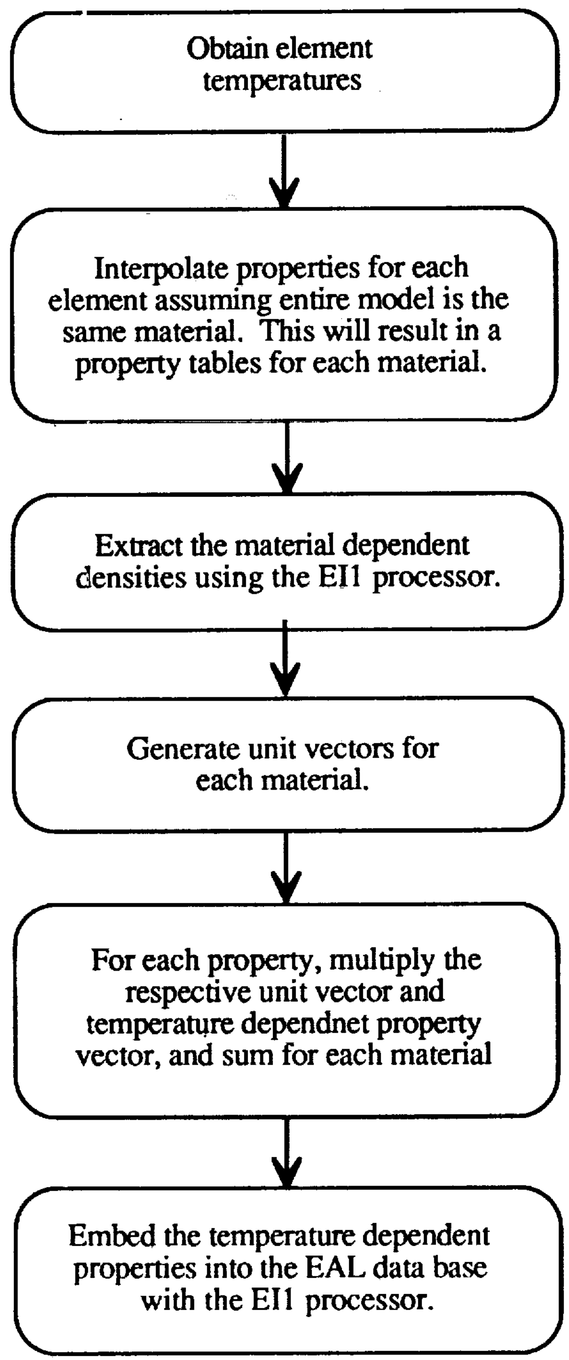

Figure 1. - Flow chart of additional runstream needed to calculate temperature dependent properties.

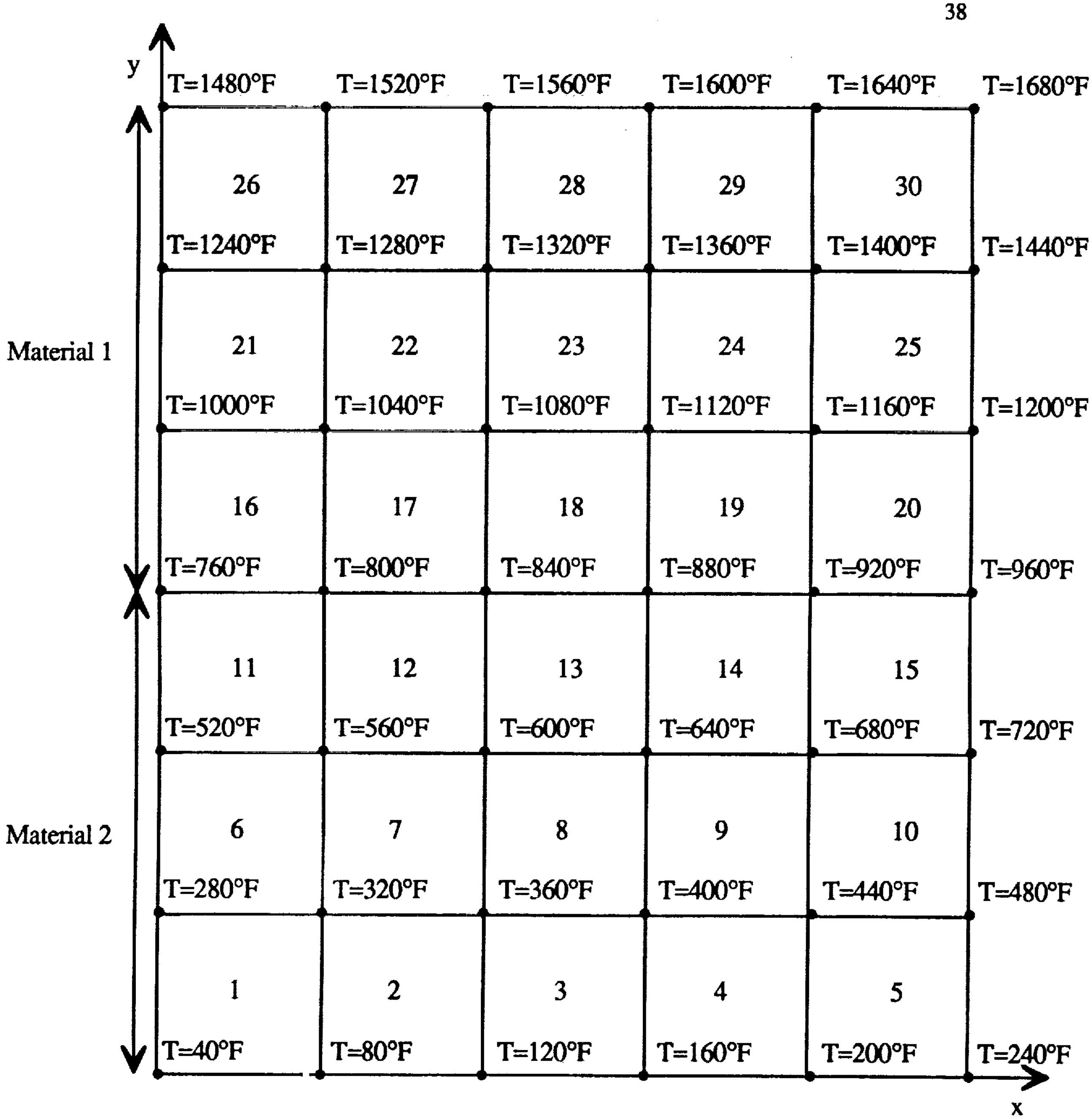

Figure 2. - EAL model used for stress analysis of a thin plate showing element numbers and nodal temperatures.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-187470		

4. Title and Subtitle	5. Report Date
Procedure for Implementation of Temperature-Dependent Mechanical Property Capability in the Engineering Analysis Language (EAL) System	December 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
David E. Glass and James C. Robinson	
	10. Work Unit No. 506-43-71-04
9. Performing Organization Name and Address Analytical Services and Materials, Inc. Hampton, VA 23666 Old Dominion University Research Foundation Norfolk, VA 23508	11. Contract or Grant No. NAS1-18599 NCC1-99
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	13. Type of Report and Period Covered Contractor Report
	14. Sponsoring Agency Code

15. Supplementary Notes

D. E. Glass: Analytical Services and Materials, Inc., Hampton, Virginia.
J. C. Robinson: Old Dominion University Research Foundation, Norfolk, Virginia.
C. J. Camarda: Langley Technical Monitor

16. Abstract

This paper presents a procedure to allow the use of temperature dependent mechanical properties in the Engineering Analysis Language (EAL) System for solid structural elements. This is accomplished by including a modular runstream in the main EAL runstream. The procedure is applicable for models with multiple materials and with anisotropic properties, and can easily be incorporated into an existing EAL runstream. The procedure (which is applicable for EAL elastic solid elements) is described in detail, followed by a description of the validation of the routine. A listing of the EAL runstream used to validate the procedure is included in the Appendix.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
temperature dependent mechanical properties finite element analysis	Unclassified - Unlimited Subject Category - 39

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	39	A03

NASA FORM 1626 OCT 86