https://ntrs.nasa.gov/search.jsp?R=19780018431 2020-03-22T03:11:34+00:00Z

NSG-3096

(NASA-CR-157240) COMPUTERIZED POWER SUPPLY ANALYSIS: STATE EQUATION GENERATION AND TERMINAL MODELS Final Report, 2 Feb. 1976 - 2 Apr. 1978 (University of South Florida) 123 p HC A06/MF A01 CSCL 09C G3/33

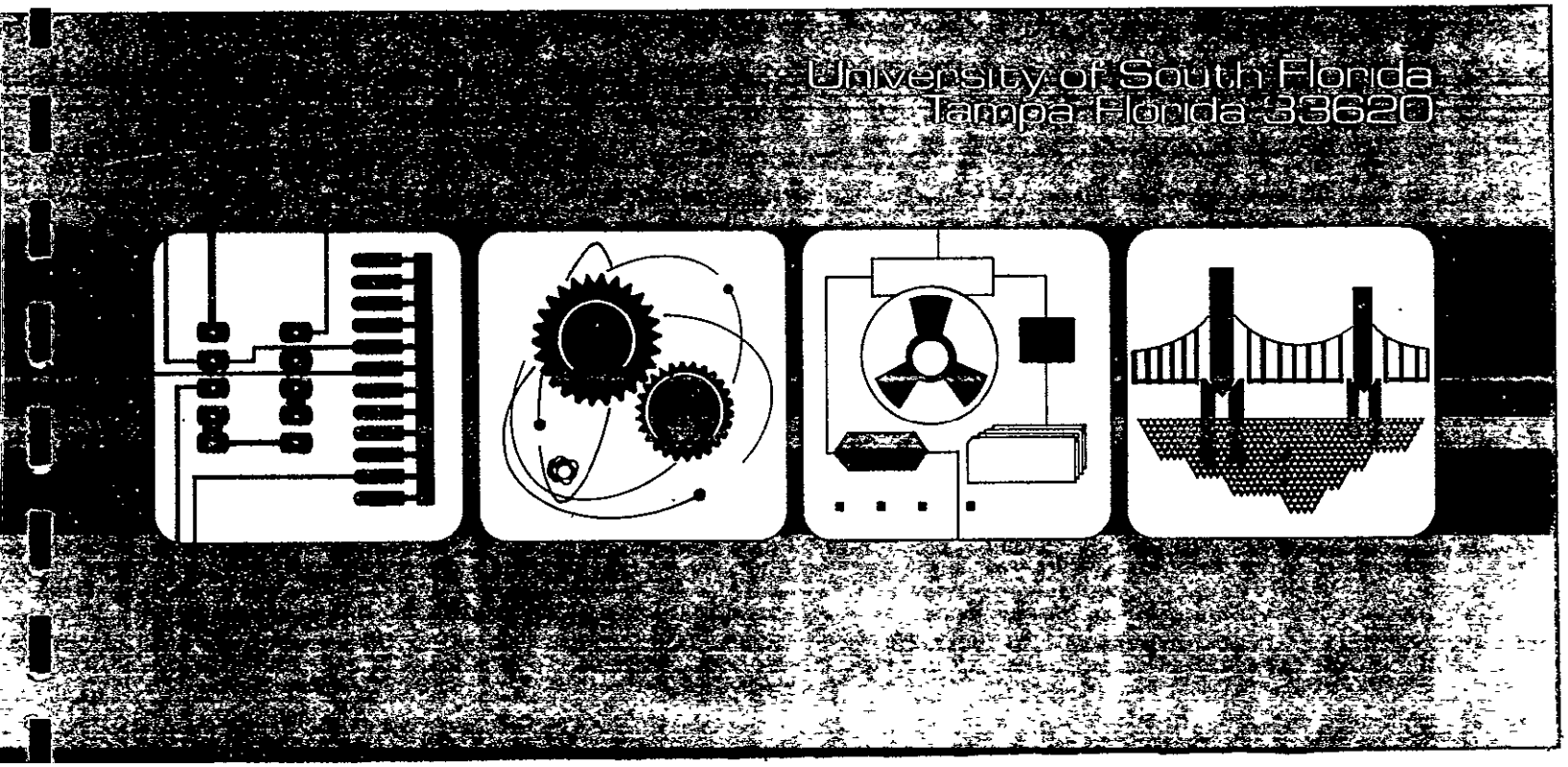
N78-26374

Unclas 23572

JUL 1978

College of Engineering

University of South Florida
Tampa, Florida 33620



Grant No. NSG-3096

COMPUTERIZED POWER SUPPLY ANALYSIS: STATE EQUATION GENERATION AND TERMINAL MODELS

Final Report

for the period from 2/2/76 to 4/2/78

University of South Florida
College of Engineering
Electrical and Electronic Systems Department

S. J. Garrett
Principal Investigator

ORIGINAL PAGE IS
OF POOR QUALITY

C O N T E N T S

	Page
1. Introduction	1
2. Analytical Background	3
3. USER's Manual for PROSE	12
4. Program Description	18
5. Examples	21
6. Summary and Conclusions	33
Appendix Listing of FORTRAN Programs	34

1. Introduction

To aid engineers that design power supply systems, special analytical tools are being introduced. These tools include integration routines that start with the description of a power supply in state-equation form and yield analytical results that include transient performance, steady-state equilibrium stability, and the response of the power supply to spurious inputs (audio susceptability).

The Department of Electrical Engineering at the University of South Florida has worked to develop two analysis tools that can be used with the state-equation analysis package.

The first tool uses a computer program that works with the SUPER*SCEPTRE circuit analysis program and prints the state equations for an electrical network. The state equations developed automatically by the computer program are used to develop an algorithm for reducing the number of state variables required to describe an electrical network. In this way a second tool is obtained in which the order of the network is reduced and a simpler terminal model is obtained. The price of the reduction in order is a reduction in the accuracy with which the model represents the system.

At the conclusion of the first year, the computer program that converts the circuit description of an electrical network to a set of state equations was completed.

In the second year the computer program was modified to accept a larger class of circuits including circuits with mutual inductance. The work on reducing the number of state variables required to describe an electrical network was completed and is described in this report.

The first task that was completed was the development of a computer program that would print the state-equations for an electrical network. The analytical background for the state equations of an electrical network is presented in Chapter 2. From the analysis in Chapter 2, the generality of the approach can be seen together with the reason for the presentation of data as matrices which are inverted in the state equations. The analytical basis for reducing the number of state variables using eigenvalue sensitivities concludes Chapter 2.

In Chapter 3 a description of the steps required to get a set of state equations is given. Chapter 3 is a user's manual for the program to PROduce State Equations (PROSE program). The user begins with an electrical network. By following the directions in Chapter 3, he can obtain the state equations for the network. Additional features of the program are also described in Chapter 3.

The description of the programs as written in FORTRAN is given in Chapter 4 and a complete listing of the program appears in the Appendix.

Several examples are given in Chapter 5 that illustrate the use of the PROSE program in obtaining the state equations for an electrical circuit.

A summary of the work accomplished under this grant appears together with recommendations for continued research in Chapter 6.

2. State Equations

The standard form of the state equations for an electrical or mechanical system is

$$\dot{x} = f(x,u,t)$$

$$y = g(x,u,t)$$

where x is an n x 1 state vector,

y is an m x 1 output vector,

u is a p x 1 input vector,

t is time,

f is the state transition function and

g is output function.

When the system is linear, the equations can be written as

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where A is n x n transition matrix

B is n x p

C is m x n

D is m x p

The matrices A, B, C, and D may be functions of time. In certain nonlinear problems the matrix representation is possible. The elements of the matrix are given as an algebraic expression in terms of circuit components such as resistors, inductors and capacitors. The value of a component may depend upon a voltage or current (the element may be nonlinear).

The state equations for an electrical circuit can be derived by the following steps:

The circuit diagram is the starting point. The model for each circuit device is obtain as a collection of two terminal elements. The final circuit diagram contains

a. capacitors
b. resistors
c. inductors
d. independent voltage sources
e. dependent voltage sources
f. independent current sources
g. dependent current sources

2. The graph of the circuit is used to obtain a tree with the tree branches selected in the following order:

   a. voltage sources
   b. capacitors
   c. resistors
   d. inductors

All voltage sources are selected before any capacitors are selected; capacitors are selected to add new nodes to the tree before any resistors are added; etc. The circuit elements which are not tree branches are called links.

3. The components are partitioned and listed in Table 1. The groups of components in the tree are listed across the top and the groups of components that are links are listed along the side.

TABLE 1

	C4	R5	L6	E7
C1	F14	0	0	F17
R2	F24	F25	0	F27
L3	F34	F35	F36	F37
J8	F84	F85	F86	F87

where

C1 is a diagonal matrix of capacitor links
R2 is a diagonal matrix of resistor links
L3 is a matrix of inductor links
J8 is a column matrix of current sources and

C4 is a diagonla matrix of capacitor tree branches
R5 is a diagonla matrix of resistor tree branches
L6 is a matrix of inductor tree branches
E7 is a column matrix of voltage sources and

The symbols F_{ij} are matrices that shown the connections between the i links and the j tree branches.

$$F_{ij} = [fpq] \quad \begin{matrix} 1 \leq p \leq \text{number of links in class p} \\ 1 \leq q \leq \text{number of tree branches in class j} \end{matrix}$$

where

$$fpq = \begin{cases} 1 & \text{if tree branch q is in the loop formed by placing link p back in the tree and if q is in the same direction in the loop as p.} \\ -1 & \text{if tree branch q is in the opposite direction in the loop defined by p.} \\ 0 & \text{if tree branch q is not in the loop defined by p.} \end{cases}$$

The state variables are selected as the column matrix $\begin{matrix} I3 \\ V4 \end{matrix}$.

Inductor Current State Variables

If mutual inductances are allowed, the equations describing the inductor currents are given as

$$L3\ \dot{I}3 + L36\ \dot{I}6 = V3 \tag{1}$$

$$L63\ \dot{I}3 + L6\ \dot{I}6 = V6 \tag{2}$$

where L3 is the inductance matrix for the inductor links

L36, L63 are the inductance matrices for mutual inductance between link inductors and tree-branch inductors and

L6 is the inductance matrix for tree-branch inductor.

Using the connection equations

$$I6 = F_{36}^{T}\ I3 + F_{86}^{T}\ J8 \tag{3}$$

which may be differentiated to give

$$\dot{I}6 = F_{36}^{T}\ \dot{I}3 + F_{86}^{T}\ \dot{J}8 \tag{4}$$

Equation (4) may be substituted into Equation (1) to get

$$[L3 + L36\ F_{36}^{T}]\ \dot{I}3 = V3 - L36\ F_{86}^{T}\ \dot{J}8 \tag{5}$$

The connection equations relating link voltages to tree branch voltages gives

$$V3 = -F_{34}\ V4 - F_{35}\ V5 - F_{36}\ V6 - F_{37}\ E7 \tag{6}$$

The inductor voltage matrix is removed by the following steps beginning with Equation (2) which is repeated next.

$$V6 = L63\ \dot{I}3 + L6\ \dot{I}6 \tag{2}$$

$$= L63\ \dot{I}3 + L6\ (F_{36}^{T}\ \dot{I}3 + F_{86}^{T}\ \dot{J}8) \tag{7}$$

The equation for $\dot{I}3$ becomes

$$[L3 + L36\ F_{36}^{T} + F36\ L63 + F36\ L6\ F_{36}^{T}]\ \dot{I}3 = -F35\ V4 - F35\ V5 - F37\ E7 \tag{8}$$

To reduce this equation so that only the state variables I3 and V4 together with independent sources E7, Ey, J8, and Jx appear, the resistor voltages and currents must be obtained as a function of the state variables.

Resistor Links and Tree Branches

The resistor voltages and currents can be expressed by the following matrix equations.

$$\begin{bmatrix} R2 & 0 \\ 0 & G5 \end{bmatrix} \begin{bmatrix} I2 \\ V5 \end{bmatrix} = \begin{bmatrix} V2 \\ I5 \end{bmatrix} = \begin{bmatrix} 0 & -F_{24} \\ F_{35}^T & 0 \end{bmatrix} \begin{bmatrix} I3 \\ V4 \end{bmatrix} + \begin{bmatrix} 0 & -F_{25} \\ F_{25}^T & 0 \end{bmatrix} \begin{bmatrix} I2 \\ V5 \end{bmatrix} + \begin{bmatrix} 0 & -F_{27} \\ F_{85}^T & 0 \end{bmatrix} \begin{bmatrix} J8 \\ E7 \end{bmatrix} \tag{9}$$

Equation (9) can be solved as

$$\begin{bmatrix} I2 \\ V5 \end{bmatrix} = \begin{bmatrix} R2 & F_{25} \\ -F_{25}^T & G5 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -F_{24} \\ F_{35}^T & 0 \end{bmatrix} \begin{bmatrix} I3 \\ V4 \end{bmatrix} + \begin{bmatrix} 0 & -F_{27} \\ F_{85}^T & 0 \end{bmatrix} \begin{bmatrix} J8 \\ E7 \end{bmatrix} \tag{10}$$

If the matrix $M_R^{-1} = \begin{bmatrix} R2 & F_{25} \\ -F_{25}^T & G5 \end{bmatrix}^{-1}$ is

partitioned as $M_R^{-1} = \begin{bmatrix} m1 & m2 \\ m3 & m4 \end{bmatrix}$,

then

$$M_R^{-1} \begin{bmatrix} R2 & F_{25} \\ -F_{25}^T & G5 \end{bmatrix} = \begin{bmatrix} m1R2 - m2F_{25}^T & m1F_{25} + m2G5 \\ m3R2 - m4F_{25}^T & m3F_{25} + m4G5 \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} \tag{11}$$

and

$$\begin{bmatrix} R2 & F_{25} \\ -F_{25}^T & G5 \end{bmatrix} M_R^{-1} = \begin{bmatrix} R2m1 + F_{25}m3 & R2m2 + F_{25}m4 \\ -F_{25}^T m1 + G5m3 & -F_{25}^T m2 + G5m4 \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} \tag{12}$$

where U is a unit matrix of the appropriate size. The solution for the matrix MR^{-1} is obtained from Equation (11) as

$$
\begin{aligned}
m1 &= \left[R2 + F25R5F_{25}^{T}\right]^{-1} \\
m2 &= -m1F25R5 \\
m3 &= R5F_{25}^{T}m1 \\
m4 &= \left[U - R5F_{25}^{T}m1F_{25}\right] R5
\end{aligned} \tag{13}
$$

Beginning with Equation (12) yields another expression for m_1, m_2, m_3, and m_4 as

$$
\begin{aligned}
m4 &= \left[F_{25}^{T}G2F_{25} + G5\right]^{-1} \\
m3 &= m4F_{25}^{T}G2 \\
m2 &= -G2F_{25}m4 \\
m1 &= G2\left[U - G2F_{25}m4F_{25}^{T}\right]
\end{aligned} \tag{14}
$$

Solving Equation (10) for V5 yields

$$
V5 = m4F_{35}I3 - m3F_{24}V4 - m3F_{27}E7 + m4F_{85}^{T}J8 \tag{15}
$$

Equation (15) is substituted into Equation (8) and solved to give the solution for state variable I3 as

$$
\begin{aligned}
\dot{I}3 = M_L^{-1}\Big[& -F_{35}m4F_{35}^{T}I_3 - \left[F_{34} - F_{35}m3F_{24}\right]V4 - \left[F_{37} - F_{35}m3F_{27}\right]E7 \\
& -F_{35}m4F_{85}^{T}J_8 - \left[L36F_{86}^{T} + F_{36}L6F_{86}^{T}\right]\dot{J}_8\Big]
\end{aligned} \tag{16}
$$

Capacitor Voltages

The voltage state equation can be obtained beginning with the capacitor equations

$$
\begin{aligned}
C1\dot{V}1 &= I1 \\
C4\dot{V}4 &= I4
\end{aligned} \tag{17}
$$

The solution begins with the second equation in (17)

$$
C4\dot{V}4 = I4 = F_{14}^{T}I1 + F_{24}^{T}I2 + F_{34}^{T}I3 + F_{84}^{T}J8 \tag{18}
$$

The variables I1 and I2 must be eliminated in the solution. Beginning with I1

$$
C1^{-1}I1 = \dot{V}1 = \left[-F_{14}\dot{V}4 - F_{17}\dot{E}7\right] \tag{19}
$$

$$C1^{-1}\dot{I1} = -F_{14}[C4^{-1}\{F_{14}^{T}I1 + F_{24}^{T}I2 + F_{34}^{T}I3 + J8\}] - F_{16}\dot{E7} \tag{20}$$

Denoting $C1^{-1}$ by S1 and $C4^{-1}$ by S4 and solving gives

$$\dot{I1} = M_s^{-1}\{-F_{14}\ S4[F_{24}^{T}I2 + F_{23}^{T}I3 + F_{84}^{T}J8] - F_{17}\dot{E7}\}$$

where $M_s = [S1 + F_{14}S4F_{14}^{T}]$ (21)

Solving Equation (10) for I2 given

$$I2 = -m1[F_{24}V4 + F_{20}E7] + m2\ [F_{35}^{T}\ I3 + F_{86}^{T}J8] \tag{22}$$

When Equations (21) and (22) are substituted into Equation (18) and the result simplified, the result is

$$\begin{aligned}\dot{V4} = S4\{&[[-[F_{14}^{T}\ M_s^{-1}\ F_{14}\ S4] + U_{44}]\ [F_{24}^{T}m2F_{35}^{T} + F_{34}^{T}]I3\\ &+ [F_{14}^{T}M_s^{-1}\ F_{14}S4F_{24}^{T} - F_{24}^{T}]\ m1F_{24}V4\\ &+ [-[F_{14}^{T}M_s^{-1}\ F_{14}S4 - U_{44}]\ [F_{84}^{T} + F_{24}^{T}\ m2F_{85}^{T}]\ J8\\ &+ [F_{14}^{T}M_s^{-1}\ F_{14}S4 - U_{44}]\ F_{24}^{T}m1F_{27}E7\\ &-F_{14}^{T}M_s^{-1}\ F_{17}\dot{E7}\}\end{aligned} \tag{23}$$

Combining Equation (16) and Equation (23) gives the state equations.

$$\begin{bmatrix}\dot{I3}\\ \dot{V4}\end{bmatrix} = \begin{bmatrix}M_L & 0\\ 0 & C4\end{bmatrix}^{-1}\begin{bmatrix}-F_{35}m4F_{35}^{T} & -F_{34} + F_{35}m3F_{24}\\ [U_{44}-F_{14}^{T}M_s^{-1}F_{14}S_4][F_{24}^{T}M2F_{35}^{T}+F_{34}^{T}] & -[U_{44}-F_{14}^{T}M_s^{-1}F_{14}S_4]F_{24}^{T}m1F_{24}\end{bmatrix}\begin{bmatrix}I3\\ V4\end{bmatrix}$$

$$+\begin{bmatrix}-F_{35}m4F_{85}^{T} & -F_{37} + F_{35}\ m3\ F_{27}\\ [U_{44}-F_{14}M_s^{-1}F_{14}S_4][F_{84}^{T}+F_{24}^{T}m2F_{85}^{T}] & -[U_{44}-F_{14}^{T}M_s^{-1}F_{14}S_4]F_{24}^{T}m1F_{27}\end{bmatrix}\begin{bmatrix}J8\\ E7\end{bmatrix}$$

$$+\begin{bmatrix}-L36F_{86}^{T} - F_{36}L6F_{86}^{T} & 0\\ 0 & -F_{14}^{7}\ M_s^{-1}\ F_{17}\end{bmatrix}\begin{bmatrix}\dot{J8}\\ \dot{E7}\end{bmatrix} \tag{24}$$

The state equations are produced as the output of the preprocessor computer program known as SCEPTRE I. These state equations are the input to the computer program SCEPTRE II which performs the integration required to obtain a transient solution.

Notice that Equation (24) appears in matrix form even though each element may depend on time or on another voltage or current that can be obtained at times when the state variables are known.

The state equations are calculated by processing the output of SCEPTRE I to obtain matrix equations in the form similar to Equation (24). While this equation is non-linear, it can be linearized around a quiescent value as specified by initial conditions for the state variables $\begin{bmatrix} I_3 \\ V_4 \end{bmatrix}$.

The linearized state equations are

$$\begin{bmatrix} \dot{I}_3 \\ \dot{V}_4 \end{bmatrix} = A \begin{bmatrix} I_3 \\ V_4 \end{bmatrix} + B \begin{bmatrix} J_8 \\ E_7 \end{bmatrix} + E \begin{bmatrix} \dot{J}_8 \\ \dot{E}_7 \end{bmatrix} \tag{26}$$

The eigenvalues of this equation are eigenvalues of the constant matrix, A.

In the representation of any physical system a compromise is made in choosing the order of the mathmatical model. The order may be order of the differential equation that describes the system or may be the number of poles in the transfer function for a linear system. In any physical system that is modeled with lumped parameters, a decision must be made on how many energy storage elements (capacitors, inductors) are included in the model. More energy storage elements increase accuracy and computational difficulty. The number of independent energy storage elements is equal to the number of state variables required.

As the number of state variables is increased the accuracy can be increased. However, in choosing the equivalent circuit for a physical system such as a power supply, some energy storage elements may be selected that do not affect the accuracy of the model significantly. A simpler description that did not reduce accuracy would give a better model.

In a state variable description of a physical system, when an independent energy storage element is discarded, the number of state variables is reduced by one. Energy storage elements that are incidental to the accurate description of a physical system usually contribute to only the shortest time constants or the largest eigenvalues of the A matrix.

The eigenvalues of the A matrix are obtained using a subroutine. The sensitivity is calculated for each parameter by increasing the parameter by 1% and calculating the eigenvalue. For example the sensitivity to a resistor is given by

$$S_R^P = \frac{R}{P} \frac{\Delta p}{\Delta R} = (100) \frac{\Delta p}{p} \tag{27}$$

where p is the eigenvalue,

Δp is the change in the eigenvalue

ΔR is the change in resistor and

R is the resistor value.

The sensitivities can be used to reduce the number of state variables. First the eigenvalue with the largest magnitude is selected. The sensitivities of this eigenvalue to each parameter is scanned. The circuit parameter (resistance, inductance, or capacitance) to which the largest eigenvalue has a sensitivity with a magnitude within 3 per cent of unity is selected. If the selected circuit parameter does not affect other eigenvalues, it is removed from the network. It is set equal to zero if the sensitivity is negative; if the sensitivity is positive, the circuit parameter is set to infinity.

The desire is to force the eigenvalue to negative infinity. From the equation

$$\frac{\Delta p}{p} = \sum_R S_{R_i}^P \frac{\Delta R}{R} + \sum_C S_c^p \frac{\Delta C}{C} + \sum_L S_L^p \frac{\Delta L}{L} \tag{28}$$

it can be inferred that the per cent change in an eigenvalue can be maximized by selecting a circuit element, ξ, with the largest sensitivity and maximizing

$$S_\xi^p \frac{\Delta \xi}{\xi} \tag{29}$$

for that term.

In every case the eigenvalue will be moved away from the origin toward infinity. If it is moved to infinity, a state variable is eliminated by the removal of a circuit element. The correlation of eigenvalue going to infinity and state variable disappearing can be observed from the characteristic equation:

$$|SI - A| = S^n + a_{n-1}S^{n-1} + \ldots + a_1 S + a_0 \tag{30}$$

In factored form the caharacteristic equation becomes

$$K \left(\frac{S}{p_1} + 1\right)\left(\frac{S}{p_2} + 1\right) \ldots \left(\frac{S}{p_n} + 1\right) \tag{31}$$

Obviously one term disappears if $p_i \to \infty$ for the largest eigenvalue p_i.

If not moved to infinity, the eigenvalue is made larger in magnitude and the process can be repeated so that other circuit elements are deleted and the eigenvalue is moved to infinity.

By considering the characteristic equation again it can be seen that

$$|SI - A| = S^n + a_{n-1}S^{n-1} + \ldots + a_0 \tag{32}$$

where

$$a_0 = p_1 \cdots p_n.$$

That is, a_0 is the product of the n eigenvalues. If the derivative of a_0 is taken with respect to a circuit element ξ where ξ is an R, L, or C, the result is

$$\frac{\partial a_0}{\partial \xi} = \sum_{i=1}^{n} (p_1 p_2 \cdots p_{i-1} p_{i+1} \cdots p_n) \frac{\partial p_i}{\partial \xi} \tag{33}$$

A little algebra yields

$$S_\xi^{a_0} = \frac{\xi}{a_0} \frac{\partial a_0}{\partial \xi} = \sum_{i=1}^{n} \frac{\xi}{p_i} \frac{\partial p_i}{\partial n} = \sum_{i=1}^{n} S_\xi^{p_i}. \tag{34}$$

For linearized state equations the constant a_0 has been observed to have the form

$$a_0 = \frac{(C1 + C2) \quad R1R2}{C1C2 \; L1L2(R1 + R2)} \; (+ \text{ other terms}) \tag{35}$$

clearly

$$|S_\xi^{a_0}| \leq 1 \tag{36}$$

for this case.

In words, the sum of the pole sensitivities to a parameter is less than or equal to 1. For this reason in most cases of interest if the sensitivity of an eigenvalue is nearly equal to one, the sensitivity of other eigenvalues will be very small. Thus, the circuit element is deleted will only cause one eigenvalue to have an appreciable change in magnitude.

The sensitivity calculations are made in the PROSE program for the eigenvalue with the maximum absolute value. If the magnitude of the sensitivity is within three per cent of unity for a circuit parameter, the circuit parameter can be removed from the circuit. In the examples in Chapter 5 the computer output shows how the circuit elements can be removed.

3. User's Manual for PROSE

3.1 Introduction

The user's manual for the computer program to PROduce State Equations is given in this section. The PROSE program can be used to derive the state equations for any electrical network that can be analyzed with the SCEPTRE* circuit analysis program and compute eigenvalue sensitivities with the following restrictions:

1. Each circuit component can be modeled by two-terminal elements which can be represented as either a voltage source, a current source, a resistor, a capacitor, or an inductor.
2. No FORTRAN Subroutines are allowed.
3. No linear, secondary, or primary dependent sources are permitted. (In place of the SCEPTRE dependent sources, voltage or current sources can be defined by expressions which can give a dependent source).

The PROSE program can be used by following these steps:

Step 1. A circuit diagram is drawn for the circuit for which the state equations are desired. Each component in the original circuit has been represented as a collection of two terminal elements in the circuit diagram.

Step 2. The circuit diagram is translated into a listing for tje SUPER*SCEPTRE circuit analysis program.*

Step 3. Input sources are requested under OUTPUTS subheading.

Step 4. After the RUN CONTROLS heading, the instruction EXECUTE SETUP PHASE ONLY is inserted.

Step 5. The control cards are added and the complete program is entered.

The PROSE program responds by outputting the state equations and a state transition matrix for a linear set of equations. In addition the PROSE program computes a set of "sensitivities" of the eigenvalues of the transition matrix that result from a change in each of the circuit elements.

---

* The format for the computer input can be found in one of the following:

1. Extended SCEPTRE Volume I. User's Manual, AD-A009-594.
2. James C. Bowers and Stephen R. Sedore, SCEPTRE: A Computer Program For Circuit Analysis and Systems Analysis, Prentice-Hall, Inc., Englewood, Cliffs, New Jersey 1971
3. User's Manual for SUPER*SCEPTRE, AD-A011-348

3.2 Example

To fix ideas consider as an example obtaining the state equations for the circuit in Figure 1.

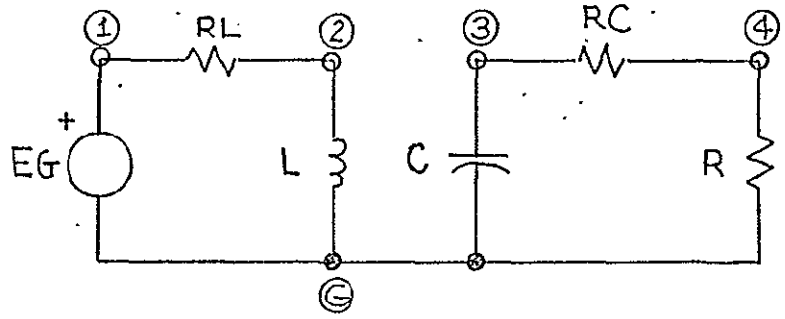


Fig. 1. Boost Power Supply Average Model TD Interval Model

The input description for PROSE use is:

	Line Number
//JOB card for 240 k byte region	1
//STEP1 EXEC PROSE	2
//INPUT DD *	3
CIRCUIT DESCRIPTION	4
BOOST POWER SUPPLY	5
AVERAGE MODEL	6
TD INTERVAL	7
ELEMENTS	8
EG,G-1=1	9
RL,1-2=1	10
L,2-G=2	11
C,3-G=3	12
RC,4-3=4	13
R,4-G=5	14
OUTPUTS	15
EG	16
RUN CONTROLS	17
EXECUTE SETUP PHASE ONLY	18
STOP TIME=5	19
END	20

Fig. 2. PROSE Input Listing for Circuit of Figure 1

The PROSE system output is condensed below.

```
DL=(-IL*RL+EG)/(L)
DC=-VC/((R+RC))/(C)
STATE VARIABLE NAMES
IL
VC

MATRIX A
  -5.00000D-01                    -5.00000D-01
   0.0                            -3.70370D-02

INDEPENDENT SOURCE NAMES
EG

MATRIX B
  -5.00000D-01
  -3.70370D-02

EIGENVALUES OF MATRIX A
     REAL PART                    IMAG. PART
     -5.0000000D-01               0.0
     -3.7037037D-02               0.0
```

Fig. 3. Summarized PROSE output for input listing of figure 2.

Figure 1 indicates the circuit whose state variable equations are computed. Each element is given a name and each node is labeled with an encircled character.

Figure 2 shows the IBM OS job control language (lines 1 through 3) to execute the system. Lines 4 through 14 in Figure 2 describe the circuit in standard SCEPTRE format.[1] The OUTPUTS request (lines 15 & 16) alerts PROSE to regard EG as an independent source in its determination of the B matrix.

The RUN CONTROLS (lines 17 through 19) include the special line:

[1] See EXTENDED SCEPTRE, Volume I. User's Manual, AD-A009-594

EXECUTE SETUP PHASE ONLY. Both this card and the STOP TIME = number dummy value are required. The END card finishes the PROSE input description.

The output of Figure 3 indicates the state variable equations for DL ($\frac{dIL}{dt}$) and DC ($\frac{dVC}{dt}$) as well as the state variable names (IL and VC), the linear transition matrix A, the independent sources (EG), the input matrix B, and the eigenvalues for the transition matrix.

The matrices are determined from the state variable equations and the element values in the form: $\dot{X} = AX + BU$, where X is the vector of state variables and U is the vector of independent sources.

Additional outputs include the sensitivities which will be further described later in this document.

3.3.0 PROSE Language

The PROSE system input language is the standard SCEPTRE transient solution language. For those unfamiliar with the SCEPTRE language, the reference of note 1 must be consulted.

3.3.1 Descriptions

PROSE will allow all standard SCEPTRE descriptions; i.e. CIRCUIT, MODEL, and RERUN. However, the rerun description is ignored and model description is used only if models under it are called in the circuit description.

3.3.2 Elements

All SCEPTRE elements (E,R,L,C,J) are allowed by PROSE. Element values may be constants, expressions or tables. PROSE will allow defined parameters in expressions but <u>will not</u> allow FORTRAN function subroutines, equations, secondary dependent sources, linear dependent sources, and primary dependent current sources. Voltage and current source derivatives are also allowed.

Models may be called as elements as in standard SCEPTRE, but model elements must also conform to the above rules.

Since PROSE is transient analysis oriented, no AC sources nor elements with bounds are permitted.

3.3.3 Defined Parameters

Standard SCEPTRE transient oriented real defined parameters may be utilized.

3.3.4 Outputs

At least one output request is necessary for proper execution. The OUTPUT requests of voltage or current sources implies that these sources are independent sources and are used in the PROSE computation of the B matrix.

3.3.5 Initial Conditions

Initial conditions may be supplied and sometimes are required since PROSE does not compute a DC operation point prior to matrix computations. The initial conditions are used to set the system operating point when a linear set of state equations are computed. If the initial conditions are not supplied, all initial state variable values are assumed to be zero.

3.3.6 Functions

Only Tables are allowed under functions. PROSE does not permit equations.

3.3.7 Run Controls

PROSE requires the special run control: EXECUTE SETUP PHASE ONLY for execution of the state variable equation generator.

A dummy STOP TIME is also required.

For debugging purposes, PROSE allows the run control: WRITE SIMUL8 DATA

All other run controls are illegal.

3.3.8 End

An END card is required at the end of the input to PROSE

No other heading or subheading cards may be included.

3.4 Features

3.4.1 PROSE system output

The output from PROSE begins with a FORTRAN routine. Within the routine is a listing of the state variable equations. These equations are of the form:

$$\text{DL or DC} = f(\text{elements}, I_L, V_c)$$

where $DL = \frac{dI_L}{dt}$ and $DC = \frac{dV_c}{dt}$

where I_L and V_c are the circuit state variables. Also, if any matrices appear in the state equations, the matrix elements will be listed and their inverse elements (elements of the inverted matrix) will be specified by a trailing "I" in the state equations.

Following the subroutine output, the linear circuit sensitivity analysis results are given.

The state variable names are listed, followed by the state transition matrix A. These are followed by the independent source names (if any) and the input matrix B (if any independent sources exist).

The eigenvalues of matrix A are then displayed followed by the sensitivity analysis. This analysis is computed by increasing each element value (except for independent sources) by 1% separately and recomputing the transition matrix eigenvalues.

PROSE lists the sensitivities of each element to the maximum eigenvalue and also gives an indication of elements which may be removed (either short or open circuited) to eliminate a state variable from the system.

3.4.2 Mechanical Systems

The standard SUPER*SCEPTRE mechanical description systems[2] may be used by PROSE giving state variable equations for these systems. Combinations of circuit and mechanical descriptions may also be analyzed.

3.5 Limitations

3.5.1 No equations may be specified

3.5.2 No FORTRAN functions

3.5.3 No linear or secondary or primary dependent sources

3.6 Error Messages

Error	
1	RUN INITIAL CONDITIONS ONLY specified
2	VECTOR EQUATIONS specified
3	RUN IC VIA IMPLICIT specified
4	RUN AC specified
5	EY type elements exist (Linear dependent source)
6	JX type elements exist (Linear dependent source)
7	J8 type elements exist (Primary dependent source)
8	J0 type elements exist (Secondary dependent source)
9	LOWER LIMIT TABLE specified
10	RUN MONTE CARLO specified
11	RUN OPTIMIZATION specified
12	RUN SENSITIVITY specified
13	RUN WORST CASE specified
14	EXECUTE SETUP PHASE ONLY not specified
15	SCEPTRE1 DTASAV dataset empty[3]
16	SCEPTRE1 PGMSAV dataset empty[3]
17	SCEPTRE1 PGMSAV dataset incorrect[3]
18	No state variables exist

---

[2] See User's Manual for SUPER*SCEPTRE, AD-A011-348

22	Pushdown stack overflow[4,5]
23	Invalid symbol encountered[5]
24	Invalid use of XTABLE format
25	Symbol divided incorrectly in FGMSAV[5]
26	More than 300 elements, tables, expressions, and defined parameters
27	Invalid symbol pushed into stack[5]
28	Invalid expression symbol pushed into stack[5]
29	Pushdown stack overflow[4,5] (same as 22)
30	A TABLE appeared incorrectly in an expression
31	Invalid use of subscript zero[5]
32	A state variable cannot be located[5]
33	Stack KSHUV overflow - possible computational delay
34	Passive element is variable - element set to zero
35	Matrix is too large for display

[3] Probably SCEPTRE1 syntax error

[4] Possibly a computational delay exists

[5] Please notify author; those are system errors.

All errors without note 5 are user errors.

Note: References refer to AD catalog numbers at:

National Technical Information Services
U. S. Department of Commerce
Springfield, VA 22161

4.0 PROSE Principles of Operation

4.1 Introduction

The PROSE system operates in conjunction with the SUPER*SCEPTRE package[6]. The system utilizes a standard unmodified SUPER*SCEPTRE system only as an equation generator, i.e. the execution phase (simulation) of SUPER*SCEPTRE is not required. The PROSE system is a "post-processor" to the equation generator. Thus, the total PROSE system operates in six sequential job steps.

4.2 Steps of the PROSE system

The first two steps of the operation of PROSE are the standard SUPER and SCEPT1 steps of a SUPER*SCEPTRE job. The SUPER step performs any processing required on a standard SUPER*SCEPTRE input stream[6] (i.e. mechanical, digital, transfer function, etc) to produce actual SCEPTRE

[6] See previous reference AD-A011-348

coding. The SCEPT1 step is also standard SCEPTRE input processing and equation generation[7]. The state variable equations generated are placed in a FORTRAN subroutine (SIMUL8) and written on dataset PGMSAV. Also, data required for execution of the equations is placed on dataset DTASAV by SCEPT1.

With this setup stage performed, the PROSE system begins processing on the PGMSAV and DTASAV files. This third step reads the DTASAV file and verifies its legality according to the criteria setup in the PROSE manual.

After verification, the system searches the PGMSAV file for the state variable equations. The equations are pushed out in an equation stack (NPUSH) with the variable name (left hand side of the equation) and a pointer to its equation in NPUSH placed in array KLOCA. All expressions and state equations are thus saved in memory. During this process, the element names are also saved in array NSYM.

The system next begins writing its FORTRAN routine. The routine is one which can compute the A and B matrices of the linearized, state variable form:

$$\dot{X} = AX + BU$$

where X is the vector of state variables and U is the vector of input sources.

Prior to the state equation generation, the system outputs the elements of any R,G,L, or S matrices which exist in the circuit.

The state equations are now generated by the following process. The symbol table (NSYM) is searched sequentially for state variables, and the equation for the derivative of that variable is initiated. The process involves location of the equation in the stack (NPUSH), popping the equation until a variable name is located, pushing the current equation pointer (onto KSHUV), and then recursively generating the equation for the current variable. At each location of a variable in an equation, either a set of parentheses are forced around the new equation (if the new equation has any sum terms in it), or not (if no sum terms exist); or, possibly, the variable is constant (element value) or a state variable itself, in which case the variable name is entered in the equation. This process may form awkwardly long state equations but decreases as much as paracticable the number of parentheses

---

[7] SCEPTRE SUPPORT II, VOLUME II, Revised Mathematical Fromulation, AD-882-385

and forces only constant valued variables to appear in the equations. Each state variable equation is thus generated by this same procedure.

The routine has now become a self-contained representation of the state equations of the input circuit.

The next two steps of PROSE perform the compilation of the SIMUL8 subroutine and linkage edit the SIMUL8 object with the object (load module in actuality) of the simulation portion of the PROSE system.

The simulation (GO step) reads the DTASAV file and begins the generation of the A and B matrices by alternately setting one and only one state variable to 1, all others are set to zero. Of course, all constant element values are set to their nominal values as specified in the input description. This A and B matrix generation is performed by subroutine EIGAN which also calls the IBM Scientific Subroutines HSBG and ATEIG to compute the eigenvalues of the real A matrix. Finally, the simulation performs a sensitivity analysis by increasing each constant non-input element value, respectively, by 1% and recomputing new A and B matrices and eigenvalues of the new A matrix.

5. Examples

Three examples are given which illustrate the use of the PROSE program in obtaining state equations for electrical networks. For each example the SCEPTRE listing is given together with the circuit diagram. The state equations are printed as the output. Notice that

DLX represents $\frac{dI_{LX}}{dt}$

and

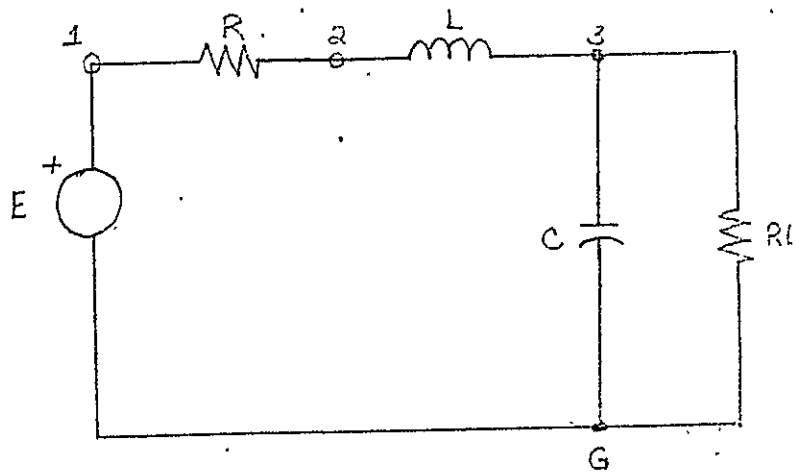
DCX represents $\frac{dV_{CX}}{dt}$.

5.1 RLC Circuit

A simple RLC circuit is used for the first example. The circuit diagram is shown together with the SCEPTRE input in Figure 5.1a. The state equations produced as the output of the PROSE program is also listed in Figure 5.1b.

In Figure 5.2 the A and B matrices are given. Next the eigenvalues are calculated together with the sensitivities of the maximum eigenvalue of one per cent change in each parameter. If the sensitivity of the circuit element is within 3% of unity, the circuit element can be removed from the network. How the circuit element can be removed (by setting it equal to zero or by setting it equal to infinity) is indicated in several of the examples that follow.

```
SU[redacted]-SCE[redacted] SY[redacted] IN[redacted]-- G[redacted] VE[redacted]N -[redacted]360[redacted]OS R[redacted]ASE 21.8
********************************************************************************

THIS IS VERSION N OF SUPER*SCEPTRE UPDATED  2/12/75
DEVELOPED BY THE UNIVERSITY OF SOUTH FLORIDA.
PLEASE REPORT ANY SOFTWARE ERRORS TO DR. JAMES C. BOWERS 813-974-2581

********************************************************************************

CIRCUIT DESCRIPTION
EXAMPLE 1
ELEMENTS
E,G-1=1
R,1-2=1
L,2-3=1
C,3-G=1
RL,3-G=1
OUTPUTS
E
RUN CONTROLS
STOP TIME=3
EXECUTE SETUP PHASE ONLY
END

END OF PRE-PROCESSOR
```



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 5.1a. RLC Circuit Example.

```
SIMULATION HAS BEEN DELETED
C
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><
C
C
C     STATE VARIABLE EQUATIONS
C
      DL=(-VC-IL*R+E)/(L)
      DC=(-VC/(RL)+IL)/(C)
C
C
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><
C
      END
```

Figure 5.1b. State Equations for Figure 5.1a.

```
NASA-LEWIS RESEARCH CENTER
---- ----- -------- ------

     PRCSE PROGRAM
     ----- -------
UNIVERSITY OF SOUTH FLORIDA
---------- -- ----- -------

EXAMPLE 1

----- -------- -----
STATE VARIABLE NAMES
----- -------- -----
IL
VC

------ -
MATRIX A
------ -
  -1.CCCCCD+CC  -1.CCCCCD+CC
   1.CCCCCD+CC  -1.CCCCCD+CC

---------- -------
INDEPENDENT SCURCES
---------- -------
E

------ -
MATRIX B
------ -
   1.CCCCCE+CC
   C.C

---------- -- ------ -
EIGENVALLES CF MATRIX A
---------- -- ------ -
            REAL PART          IMAG. PART
            ---- ----          ----- ----
          -1.CCCCCCCE+CC      1.CCCCCCCE+CC
          -1.CCCCCCCD+CC     -1.CCCCCCCE+CC

---------- --------
SENSITIVITY ANALYSIS
---------- --------
MAXIMUM EIGENVALLE #  2 MAGNITUDE =  1.4142136D+00

ELEMENT  SENSITIVITY
-------  -----------
 C          -C.496
 R           0.250
 RL         -C.248
 L          -C.496

NO SENSITIVITIES WITHIN 3% OF  1 .
C      MAY BE REMOVED (I.E. SET TO ZERO     IF SENSITIVITY < ZERO
                           SET TO INFINITY IF SENSITIVITY > ZERO.)
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 5.2. Matrices, Eigenvalues and Sensitivities for RLC Circuit.

5.2 Mutual Inductance

A simple transformer system is shown in Figure 5.3 together with the SUPER*SCEPTRE input. The state equations are given in Figure 5.4. Notice the generation of the inductor matrix equations, XML, and the references to the inverse elements of this matrix, XMLI. In Figure 5.5 the A matrix, B matrix, eigenvalues and sensitivities are listed.

```
SUPER*SCEPTRE SYSTEM INPUT -- 02/75 VERSION -- S/360 -- OS RELEASE 21.8
*****************************************************************************************************************

THIS IS VERSION A OF SUPER*SCEPTRE UPDATED  2/12/75
DEVELOPED BY THE UNIVERSITY OF SOUTH FLORIDA.
PLEASE REPORT ANY SOFTWARE ERRORS TO DR. JAMES C. BOWERS 813-974-2581

*****************************************************************************************************************

CIRCUIT DESCRIPTION
MUTUAL INDUCTANCE TEST FOR PROSE
ELEMENTS
E,1-2=1
R1,2-3=1
L1,3-1=1
L2,4-5=1
M,L1-L2=0.95
R2,4-5=1
OUTPUTS
E,VL1,VL2,IL1,IL2,IR2
RUN CONTROLS
STOP TIME=5
EXECUTE SETUP PHASE ONLY
END

END OF PRE-PROCESSOR
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 5.3. Mutual Inductance Example.

```
SIMULATION HAS BEEN DELETED
C
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
C
C
C      MATRIX ECUATIONS
C
       DIMENSION XML(   2,   2)
       XML(1,1)=L1
       XML(2,1)=(+R1)
       XML(1,2)=(+R1)
       XML(2,2)=L2
C
C
C
C      STATE VARIABLE EQUATIONS
C
       DL1=(+XMLI(1,1)*(-IL1*R1+E)+XMLI(1,2)*(-IL2)*R2)
       DL2=(+XMLI(2,1)*(-IL1*R1+E)+XMLI(2,2)*(-IL2)*R2)
C
C
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
C
       END
```

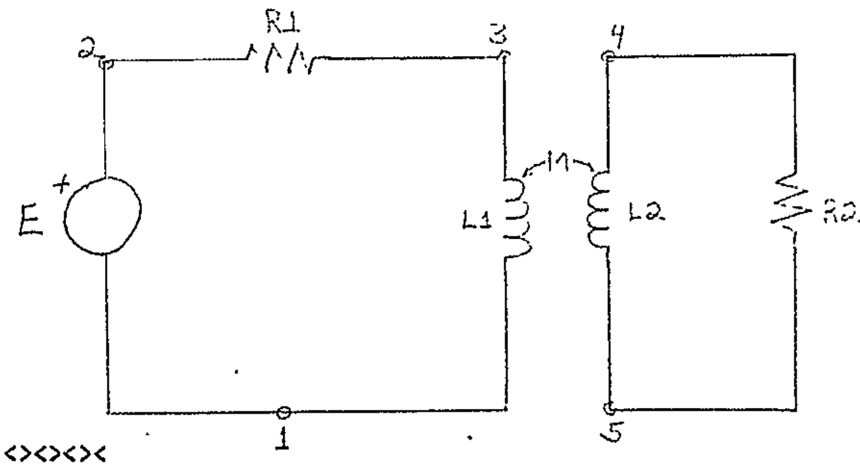


Figure 5.4. State Equations for Mutual Inductance Example.

```
NASA-LEWIS RESEARCH CENTER
---- ----- -------- ------

     PROSE PROGRAM
     ----- -------
UNIVERSITY OF SOUTH FLORIDA
---------- -- ----- -------

MUTUAL INDUCTANCE TEST FOR PROSE

----- -------- -----
STATE VARIABLE NAMES
----- -------- -----
IL1
IL2

------ -
MATRIX A
------ -
  -1.C2564D+C1   9.74359D+00
   9.743590+CC  -1.C2564D+C1

---------- -------
INDEPENDENT SOURCES
---------- -------
E

------ -
MATRIX B
------ -
   1.C2564D+01
  -9.743590+C0

---------- -- ------ -
EIGENVALLES OF MATRIX A
---------- -- ------ -
             REAL PART          IMAG. PART
             ---- ----          ----- ----
          -2.CCCGOCOD+01       0.0
          -5.1282C51D-C1       0.0

---------- --------
SENSITIVITY ANALYSIS
---------- --------
MAXIMUM EIGENVALLE #  1 MAGNITUDE =  2.CCCC000D+01

ELEMENT  SENSITIVITY
-------  -----------
 R1          C.5CC
 R2          0.5CC
 L1         -9.069
 L2         -9.C69

NO SENSITIVITIES WITHIN 3% OF  1 .
L1      MAY BE REMOVED (I.E. SET TO ZERO     IF SENSITIVITY < ZERO
                            SET TO INFINITY IF SENSITIVITY > ZERO.)
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 5.5. Matrices, Eigenvalues and Sensitivities for Mutal Inductance Example.

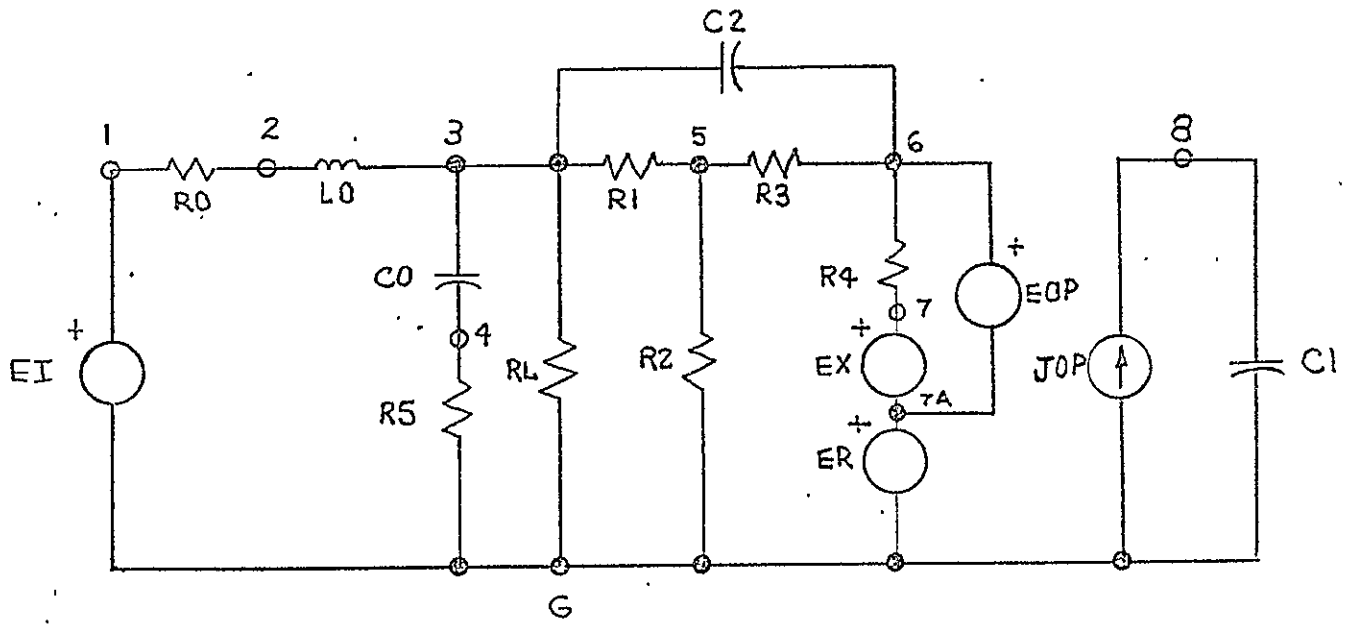
5.3 Integral Pulse Frequency Modulation Regulator

The circuit model for an integral pulse frequency modulation regulator* is given in Figure 5.5. The state equations follow in Figure 5.7. The A matrix is shown in Figure 5.8 with the eigenvalues for the A matrix. Also shown is the variation in eigenvalues that accompanies a one per cent change in each parameter.

---

* R. P. Iwens, Y. Yu, and J. Triner, "Time Domain Modeling and Stability Analysis of an Integral Pulse Frequency Modulated DC to DC Power Converter", PESC '75 Record

```
SUPER*SCEPTRE SYSTEM INPUT -- 02/75 VERSION -- S/360 -- OS RELEASE 21.8

**********************************************************************************************************************

THIS IS VERSION N OF SUPER*SCEPTRE UPDATED  2/12/75
DEVELOPED BY THE UNIVERSITY OF SOUTH FLORIDA.
PLEASE REPORT ANY SOFTWARE ERRORS TO DR. JAMES C. BOWERS 813-974-2581

**********************************************************************************************************************

CIRCUIT DESCRIPTION
INTEGRAL PFM DC TO DC CONVERTER
IVENS AND YU - PESC 75 RECORD P80-90
NASA LEWIS RESEARCH CENTER
ELEMENTS
EI,G-1=30
RO,1-2=0.015
LO,2-3=250E-6
CO,3-4=300E-6
R5,4-G=0.077
RL,3-G=1
R1,3-5=26.7E3
R2,5-G=13.3E3
C2,3-6=0.022E-6
R3,5-6=10E3
R4,6-7=100E3
EX,7A-7=X1((EI-ILO*RO-VRL)*PN)
ER,G-7A=X2((R2/(R1+R2))*PKR)
EOP,7A-8=X4(0.0)
EOP=0.0
JOP,G-8=X3(IEOP)
C1,8-G=220E-6
DEFINED PARAMETERS
PN=0.65
PKR=15
OUTPUTS
EI,VRL(EC),VC1(EC),ILO(I),VCO(VC)
RUN CONTROLS
STOP TIME=100E-6
EXECUTE SETUP PHASE ONLY
END

END OF PRE-PROCESSOR
```



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 5.6. Integral Pulse Frequency Modulation Regulator Model.

```
SIMULATION HAS BEEN DELETED
C
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><
C
C
C      STATE VARIABLE EQUATIONS
C
       DILO=(-VC2-ILO/(1./(RO))-(((R2/(R1+R2))*PKR))-(0.00)+EI)/(LO)
       DC1=((+(+VC2-VC0+(((R2/(R1+R2))*PKR))+(0.00))/(R5)+(+VC2+(((
     $R2/(R1+R2))*PKR))+(0.00))/(RL)+(+(+1./(R2)*(-(((R2/(R1+R2))*PKR))-
     $(0.00))-1./(R1)*(-VC2))/((1./(R3)+1./(R2)+1./(R1)))+(((R2/(R1+
     $R2))*PKR))+(0.00))/(R2)+(-((((EI-ILO*RO-(+VC2+(((R2/(R1+R2))*PKR))+
     $(0.00)))*PN))+(0.00))/(R4)-ILO))/(C1)
       DC2=(-(+VC2-VC0+(((R2/(R1+R2))*PKR))+(0.00))/(R5)-(+VC2+(((R2/(
     $R1+R2))*PKR))+(0.00))/(RL)-(+VC2-(+1./(R2)*(-(((R2/(R1+R2))*PKR))-
     $(0.00))-1./(R1)*(-VC2))/((1./(R3)+1./(R2)+1./(R1))))/(R1)+ILO)/(
     $C2)
       DC0=(+VC2-VC0+(((R2/(R1+R2))*PKR))+(0.00))/(R5)/(C0)
C
C
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><
C
       END
```

Figure 5.7. State Equations for Figure 5.6.

```
NASA-LEWIS RESEARCH CENTER
---- ----- -------- ------

     PROSE PROGRAM
     ----- -------
UNIVERSITY OF SOUTH FLORIDA
---------- -- ----- -------

INTEGRAL PFM DC TO DC CONVERTER
IWENS AND YU - PESC 75 RECORD P80-90
NASA LEWIS RESEARCH CENTER

----- -------- -----
STATE VARIABLE NAMES
----- -------- -----
ILO
VC1
VC2
VCO

------ -
MATRIX A
------ -
  -6.000000+04    0.0          -4.000000+03    0.0
  -4.54545D+06    0.0           5.948680+09   -5.903190+09
   4.54545D+07    0.0          -5.948860+08    5.903190+08
   0.0            0.0           4.329000+04   -4.329000+04

---------- -------
INDEPENDENT SOURCES
---------- -------
EI

------ -
MATRIX B
------ -
   4.000000+03
  -2.95455D+03
   0.0
   0.0

---------- -- ------ -
EIGENVALUES OF MATRIX A
---------- -- ------ -
            REAL PART          IMAG. PART
            ---- ----          ---- ----
           0.0                 0.0
          -5.94908200+08       0.0
          -3.4822979D+02       3.6234046D+03
          -3.4822979D+02      -3.6234046D+03

---------- --------
SENSITIVITY ANALYSIS
---------- --------
MAXIMUM EIGENVALUE #  2 MAGNITUDE =  5.94908200+08

ELEMENT  SENSITIVITY
-------  -----------
 EX          0.0
 ER          0.0
 C1          0.0
 C2         -0.999     *** ELEMENT MAY BE REMOVED BY SETTING TO ZERO.
 CO         -0.000
 RO         -0.000
 R3         -0.000
 R5         -0.983     *** ELEMENT MAY BE REMOVED BY SETTING TO ZERO.
 RL         -0.008
 R2         -0.000
 R1         -0.000
 R4          0.0
 LO          0.000
 JOP         0.0
```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 5.8. Matrices, Eigenvalues and Sensitivities for Figure 5.6.

6. Summary and Conclusions

The work during the two years of the NASA Grant NSG-3096 has resulted in a computer program which will convert a SCEPTRE listing for an electrical or mechanical system into a set of state equations for the network. Additional features of the program to PROduce State Equations include subroutines to:

1. linearize the state equations about an operating point as specified by the initial conditions
2. obtain the eigenvalues for the linearized state equations
3. obtain the sensitivities of the eigenvalue with the maximum magnitude
4. use the sensitivity calculations to indicate which circuit element can be removed from the network without adversely affecting the systems dynamic operation

At the end of the first year the PROSE program had limitations on the type of networks it would accept. In the second year several of these limitations were removed at a great cost in programming manpower. The most significant improvement is the ability PROSE now has to accept networks with mutual inductance.

Additional research and development could profitably be spent in applying the PROSE program to the design of automatic control systems. Using SUPER*SCEPTRE an electromechanical description of the plant to be controlled would be reduced to the state equations for the systems. The state equations could be used in an algorithm to automatically compute the control matrix.

Appendix

This appendix gives the Fortran listing for the computer program that PROduces State Equations.

```
C     PRCSE1                                                                  PR1 0001
C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0002
C                                                                             PR1 0003
C                                                                             PR1 0004
C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0005
C     COPYRIGHT 1977 BY JOHN E. O'REILLY, JR.                                 PR1 0006
C     10/18/77                                                                PR1 0007
C     COPYRIGHT 1978 BY JOHN E. O'REILLY, JR.                                 PR1 0008
C     05/13/78                                                                PR1 0009
C                                                                             PR1 0010
C                                                                             PR1 0011
C     TRICENTENNIAL VERSION OF P-R-O-S-E                                      PR1 0012
C                                                                             PR1 0013
C                                                                             PR1 0014
C          *  *  *  *  *  *//////////////////////////////////////////        PR1 0015
C           *  *  *  *  *  *//////////////////////////////////////////        PR1 0016
C          *  *  *  *  *  *                                                   PR1 0017
C           *  *  *  *  *  *//////////////////////////////////////////        PR1 0018
C          *  *  *  *  *  *//////////////////////////////////////////         PR1 0019
C           *  *  *  *  *                                                     PR1 0020
C          *  *  *  *  *  *//////////////////////////////////////////         PR1 0021
C           *  *  *  *  *  *//////////////////////////////////////////        PR1 0022
C          *  *  *  *  *  *                                                   PR1 0023
C          ////////////////////////////////////////////////////////////       PR1 0024
C          ////////////////////////////////////////////////////////////       PR1 0025
C                                                                             PR1 0026
C          ////////////////////////////////////////////////////////////       PR1 0027
C          ////////////////////////////////////////////////////////////       PR1 0028
C                                                                             PR1 0029
C          ////////////////////////////////////////////////////////////       PR1 0030
C          ////////////////////////////////////////////////////////////       PR1 0031
C                                                                             PR1 0032
C          ////////////////////////////////////////////////////////////       PR1 0033
C          ////////////////////////////////////////////////////////////       PR1 0034
C                                                                             PR1 0035
C                                                                             PR1 0036
0001       BLOCK DATA                                                         PR1 0037
C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0038
C                                                                             PR1 0039
C                                                                             PR1 0040
C     P-R-O-S-E - GENERATE STATE EQUATIONS FROM SCEPTRE                       PR1 0041
C                 BUILT PGMSAV AND DTASAV FILES. THE RGUTINES                 PR1 0042
C                 COMPUTE THE STATE VARIABLE EQUATIONS AND A AND B            PR1 0043
C                 MATRICES.                                                   PR1 0044
C                                                                             PR1 0045
C                                                                             PR1 0046
C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0047
0002       IMPLICIT REAL*8(A-H,O-Z)                                           PR1 0048
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                C
                C
                C     VARIABLE DEFINITIONS - NASA
                C     NSYM   - ELEMENTS SYMBOL STORAGE ARRAY
                C     KSYM   - NUMBER OF SYMBOLS IN NSYM
                C     NRAY   - INPUT AND OUTPUT 80 BYTE ARRAY
                C     KRAY   - CURRENT LOCATION POINTER IN NRAY
                C     NLEN   - ARRAY OF LENGTHS OF EACH TYPE OF ELEMENT
                C     NPT    - POINTERS TO ELEMENT TYPES IN NSYM
                C     NT     - TYPE OF ELEMENT VALUE(=0 IS NOT EJ TYPE)
                C
0003                  COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                     1)
                C
                C
                C     VARIABLE DEFINITIONS - PUSHED
                C     KLOCA  - ARRAY OF LEFT HAND SIDE ENCODED NAMES AND
                C              LOCATION OF EQUATION IN NPUSH AND
                C              SUM TERMS FLAGS
                C     KLOC   - LENGTH OF KLOCA
                C     NPUSH  - EQUATION ARRAY
                C     KPUSH  - LENGTH OF EQUATION ARRAY NPUSH
                C     KPMAX  - MAXIMUM LENGTH OF NPUSH(1000)
                C
0004                  COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
                C
                C
                C     VARIABLE DEFINITIONS - SHGR
                C     NSHUV  - PUSH DOWN STACK(FIFO TYPE)
                C     KSHUV  - CURRENT POINTER INTO NPUSH STACK
                C     KST    - START OF CURRENT EQUATION IN NPUSH
                C     KEND   - END OF CURRENT EQUATION IN NPUSH
                C     KSUB   - CURRENT VARIABLE'S SUBSCRIPT VALUE
                C     KEXTD  - CURRENT VARIABLE'S EXTENDED VALUE
                C     KPAR   - PARENTHESIS FLAG
                C
0005                  COMMON /SHGR/ NSHUV(500),KSHUV,KST,KEND,KSUB,KEXTD,KPAR
                C
                C
                C     VARIABLE DEFINITIONS - CNTRL
                C     LIST   - DEBUG FLAG
                C     L1     - RESERVED WORD
                C     CTROL  - CONTROLS FROM DTASAV
                C     KXM    - XML, XMRG, AND XMS ARRAY SIZES
                C
0006                  COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
                C
                C
```

```
                C     VARIABLE DEFINITIONS - FILES                                          PR1 0097
                C     NPRINT - PRINTER FILE NUMBER                                          PR1 0098
                C     NPGM   - SCEPTRE PROGRAM FILE NUMBER                                  PR1 0099
                C     NDTA   - SCEPTRE DATA FILE NUMBER                                     PR1 0100
                C                                                                           PR1 0101
CCC7            C     COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU                                  PR1 0102
                C                                                                           PR1 0103
                C                                                                           PR1 0104
                C     VARIABLE DEFINITIONS - CHARS                                          PR1 0105
                C     NSP    - SPACE                                                        PR1 0106
                C     NLP    - LEFT PARNTHESIS                                              PR1 0107
                C     NRP    - RIGHT PARENTHESIS                                            PR1 0108
                C     NCOM   - COMMA                                                        PR1 0109
                C     NEQ    - EQUAL SIGN                                                   PR1 0110
                C     ND     - ALPHA LETTER "D"                                             PR1 0111
                C     NPLUS  - PLUS SIGN                                                    PR1 0112
                C     NMINUS - MINUS SIGN                                                   PR1 0113
                C     NSL    - SLASH                                                        PR1 0114
                C     NZER   - ZERO NUMERIC                                                 PR1 0115
                C     NXM(3)- ALPHA 'L','R','S'                                             PR1 0116
                C     NUM(10)- NUMERICS                                                     PR1 0117
                C                                                                           PR1 0118
0008            C     COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3PR1 0119
                     1),NUM(10)                                                           PR1 0120
                C                                                                           PR1 0121
                C                                                                           PR1 0122
                C     VARIABLE DEFINITIONS - HOLDIT                                         PR1 0123
                C     HEAD   - HEADING CARD BUFFER                                          PR1 0124
                C     KHEAD  - NUMBER HEADING CARDS                                         PR1 0125
                C     NCUM1  - RESERVED WORD 1                                              PR1 0126
                C     ELTS   - ELEMENT VALUES                                               PR1 0127
                C     KELTS  - NUMBER OF ELEMENTS                                           PR1 0128
                C     KEL    - NUMBER ELEMENTS-NUMBER EJ ELEMENTS                           PR1 0129
                C     SDR    - SOURCE DERIVATIVE VALUES                                     PR1 0130
                C     NSDR   - SOURCE DERIVATIVE NAMES                                      PR1 0131
                C     KSDR   - NUMBER SOURCE DERIVATIVES                                    PR1 0132
                C     NCUM3  - RESERVED WORD 3                                              PR1 0133
                C     XIC    - INITAL STATE VARIABLE VALUES                                 PR1 0134
                C     KSV    - NUMBER STATE VARIABLES                                       PR1 0135
                C     NCUM4  - RESERVED WORD 4                                              PR1 0136
                C     PAR    - DEFINED PARAMETER VALUES                                     PR1 0137
                C     NPAR   - DEFINED PARAMETER NAMES                                      PR1 0138
                C     KDP    - NUMBER DEFINED PARAMETERS                                    PR1 0139
                C     NCUM5  - RESERVED WORD 5                                              PR1 0140
                C     XTAB   - TABULAR FUNCTION VALUES                                      PR1 0141
                C     NTAB   - TABLE NAMES                                                  PR1 0142
                C     KTAB   - NUMBER ENTRIES IN XTAB                                       PR1 0143
                C     KTAB2  - NUMBER OF TABLES                                             PR1 0144
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                C     KEJ    - NUMBER OF INPUT(INDEPENDENT) SOURCES                          PR1 0145
                C                                                                            PR1 0146
                C                                                                            PR1 0147
0009                  COMMON /HOLDIT/ HEAD(9,11),KHEAD,NDUM1,ELTS(300),KELTS,KEL,SDR(50)PR1 0148
                     1,NSDR(6,50),KSDR,NDUM3,XIC(50),KSV,NDUM4,PAR(100),NPAR(6,100),KDP,PR1 0149
                     2NDUM5,XTAB(500),NTAB(7,50),KTAB,KTAB2,KEJ                          PR1 0150
                C                                                                            PR1 0151
                C                                                                            PR1 0152
                C     SET CONSTANTS                                                          PR1 0153
                C     PUSHED                                                                 PR1 0154
0010                  DATA KLOC/0/                                                           PR1 0155
0011                  DATA KPUSH/1/,KPMAX/1000/                                              PR1 0156
                C     FILES                                                                  PR1 0157
0012                  DATA NPRINT/6/,NPGM/1/,NDTA/2/,NEQU/3/                                 PR1 0158
                C     CHARS                                                                  PR1 0159
0013                  DATA NSP/1H /,NLP/1H(/,NRP/1H)/,NCOM/1H,/                              PR1 0160
0014                  DATA NEQ/1H=/,ND/1HD/,NPLUS/1H+/,NMINUS/1H-/                           PR1 0161
0015                  DATA NSL/1H//,NZER/1H0/                                                PR1 0162
0016                  DATA NAM/1HL,1HR,1HS/                                                  PR1 0163
0017                  DATA NUM/1HC,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                      PR1 0164
                C     HOLDIT                                                                 PR1 0165
0018                  DATA KTAB/0/,KTAB2/0/,KSDR/0/,KELTS/0/,KDP/0/                          PR1 0166
                C                                                                            PR1 0167
0019                  DATA KXM/3*0/                                                          PR1 0168
0020                  END                                                                    PR1 0169
```

```
                C      <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                C      BEGIN MAIN ROUTINE
                C
                C      <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                C
0001                   IMPLICIT REAL*8(A-H,O-Z)
0002                   COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                      1)
0003                   COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0004                   COMMON /FILES/ NPRINT,NPGM,NCTA,NEQU
                C
                C      GET SYMBOLS
0005                   CALL SETUP
                C
                C
                C      GENERATE STATE EQUATIONS HEADER
0006                   CALL MICUP
                C      BUILD EQUATIONS
0007                   CALL BILD
                C
                C
                C      FIN
0008                   CALL FINIE
0009                   STOP 0
0010                   END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001           SUBROUTINE BACK (NAME,NCHAR)
       C
       C       <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
       C
       C       CHANGE NAME TO NCHAR IN MECHANICAL FORMAT
       C       CHANGES
       C       GP=TP      DXXX=ZXXX AND CHANGE XXX
       C       I=F EXCEPT ICW=TJ
       C       V=V EXCEPT VCW=WJ
       C       ELEMENTS
       C       CU=M    EU=U    JU=R    LU=K    RU=D
       C       CW=J    EW=G    JW=C
       C
       C       <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
       C
0002           DIMENSION NEL(5), NELF(8), NST(2), NSTF(4), NCHAR(6), NAME(6)
0003           EQUIVALENCE (NG,NELF(7)), (ND,NELF(5)), (NT,NSTF(3))
0004           EQUIVALENCE (NU,NELF(2)), (NC,NELF(8)), (NW,NSTF(4))
0005           DATA NEL/1HC,1HE,1HJ,1HL,1HR/
0006           DATA NELF/1HM,1HU,1HR,1HK,1HD,1HJ,1HG,1HC/
0007           DATA NST/1HI,1HV/
0008           DATA NSTF/1HF,1HV,1HT,1HW/
0009           DATA NZ/1HZ/,NP/1HP/,NBL/1H /
0010           KST=1
0011           KN=0
0012           DO 1 I=1,6
0013           IF(NAME(I).NE.NBL) KN=KN+1
0014     1     NCHAR(I)=NAME(I)
0015           IF(KN.LE.1) RETURN
0016           IF(NCHAR(1).NE.NG.OR.NCHAR(2).NE.NP) GO TO 3
0017           NCHAR(1)=NT
0018     2     RETURN
0019     3     DO 4 KINS=1,4
0020           KLOC=KN-KINS+1
0021           IF(KLOC.LE.1) RETURN
0022           IF(NCHAR(KLOC).EQ.NU.OR.NCHAR(KLOC).EQ.NW) GO TO 5
0023     4     CONTINUE
0024           RETURN
0025     5     IF(NCHAR(1).NE.ND) GO TO 6
0026           NCHAR(1)=NZ
0027           KST=2
0028     6     DO 7 K1=1,2
0029           IF(NCHAR(KST).NE.NST(K1)) GO TO 7
0030           NCHAR(KST)=NSTF(K1)
0031           KST=KST+1
0032           IF(NCHAR(KST).EQ.NC.AND.NCHAR(KLOC).EQ.NW) NCHAR(KST-1)=NSTF(K1+2)
0033           GO TO 8
0034     7     CONTINUE
```

```
0035              8     KEL=5                                                    PR1 0243
0036                    KELF=3                                                   PR1 0244
0037                    IF(NCHAR(KLOC).EQ.NU) GO TO 9                            PR1 0245
0038                    KEL=3                                                    PR1 0246
0039                    KELF=5                                                   PR1 0247
0040              9     DO 10 K1=1,KEL                                           PR1 0248
0041                    IF(NCHAR(KST).NE.NEL(K1)) GO TO 10                       PR1 0249
0042                    NCHAR(KST)=NELF(K1+KELF)                                 PR1 0250
0043                    GO TO 11                                                 PR1 0251
0044              10    CONTINUE                                                 PR1 0252
0045                    RETURN                                                   PR1 0253
0046              11    IF(KLOC.EQ.KN) GO TO 13                                  PR1 0254
0047                    KELF=KN-KLOC                                             PR1 0255
0048                    DO 12 K1=1,KELF                                          PR1 0256
0049              12    NCHAR(KLOC+K1-1)=NCHAR(KLOC+K1)                          PR1 0257
0050              13    NCHAR(KN)=NBL                                            PR1 0258
0051                    GO TO 2                                                  PR1 0259
0052                    END                                                      PR1 0260
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001          SUBROUTINE BCDOUT (KVAL)
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
        C     PUT THE BCD CHARS FOR (KVAL) INTO OUTPUT STRING
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
0002          COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
             1),KNUMS(10)
        C
0003          MVAL=KVAL
0004          NLEN=INT(ALOG10(AMAX0(MVAL,1)))+1
0005          MDIV=10**(NLEN-1)
0006          KFL=0
0007          DO 1 I=1,NLEN
0008          MPTR=MVAL/MDIV+1
0009          IF(MPTR.EQ.1.AND.KFL.EQ.0) GO TO 1
0010          CALL PUTCHR (KNUMS(MPTR))
0011          KFL=1
0012          MVAL=MVAL-(MPTR-1)*MDIV
0013    1     MDIV=MDIV/10
0014          RETURN
0015          END
```

```
0001                SUBROUTINE BILD
              C
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
              C     BUILD THE STATE VARIABLE EQUATIONS
              C
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
0002                IMPLICIT REAL*8(A-H,O-Z)
0003                COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                   1)
0004                COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0005                COMMON /CHARS/ NSP,NLP,NRP,NCCM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
                   1),NUM(10)
0006                COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0007                COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0008                DIMENSION MRAY(6), LXM(6)
0009                DATA MRAY(1)/1HC/
0010                DATA LXM/1HX,1HM,1H ,1H(,1H ,1H /
0011                KFL=0
              C     CLEAR OUT ARRAY ONCE ONLY
0012                DO 1 I=1,72
0013           1    NRAY(I)=NSP
0014                KRAY=7
              C     OUTPUT ANY MATRICES
0015                DO 4 I=1,3
0016                IF(KXM(I).EQ.0) GO TO 4
0017                IF(KFL.EQ.0) WRITE (NEQU,2)
0018                KFL=1
0019           2    FORMAT (22HC      MATRIX EQUATIONS,58X,/,1HC,79X)
0020                WRITE (NEQU,3) NXM(I),KXM(I),KXM(I)
0021           3    FORMAT (6X,12HDIMENSION XM,A1,1H(,I3,1H,,I3,1H),52X)
0022           4    CONTINUE
0023                DO 6 I=1,3
0024                IF(KXM(I).EQ.0) GO TO 6
0025                JN=KXM(I)
0026                JNN=JN*JN
0027                LXM(3)=NXM(I)
0028                DO 5 J=1,JNN
0029                NB=1+(J-1)/JN
0030                NA=J-(NB-1)*JN
0031                IF(NA.GT.NB) KVAL=100000*I+(NA-1)*JN+NB
0032                IF(NA.LE.NB) KVAL=100000*I+(NB-1)*JN+NA
0033                CALL PUTNAM (LXM)
0034                CALL BCDOUT (NA)
0035                CALL PUTCHR (NCCM)
0036                CALL BCDOUT (NB)
0037                CALL PUTCHR (NRP)
0038                CALL PUTCHR (NEQ)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0039                CALL POP (KVAL)                                                  PR1 0333
0040          5     CALL PUTLIN                                                      PR1 0334
0041          6     CONTINUE                                                         PR1 0335
0042                IF(KFL.EQ.1) WRITE (NEQU,7)                                      PR1 0336
0043          7     FORMAT (1HC,79X,/,1HC,79X,/,1HC,79X)                             PR1 0337
0044                WRITE (NEQU,8)                                                   PR1 0338
0045          8     FORMAT (3CHC      STATE VARIABLE EQUATIONS,50X,/,1HC,79X)        PR1 0339
              C     LOOP ON STATE VARIABLES                                          PR1 0340
0046                DO 11 KSVPT=3,4                                                  PR1 0341
0047                KSV=NLEN(KSVPT)                                                  PR1 0342
0048                IF(KSV.EQ.0) GO TO 11                                            PR1 0343
0049                LSV=NPT(KSVPT)-1                                                 PR1 0344
0050                DO 10 JSV=1,KSV                                                  PR1 0345
              C     POINT TO STATE VARIABLE NAME                                     PR1 0346
0051                LJSV=LSV+JSV                                                     PR1 0347
              C     PUT THE NAME                                                     PR1 0348
0052                DO 9 I=1,5                                                       PR1 0349
0053          9     MRAY(I+1)=NSYM(I,LJSV)                                           PR1 0350
0054                CALL PUTNAM (MRAY)                                               PR1 0351
              C     AND AN EQUALS SIGN                                               PR1 0352
0055                NRAY(KRAY)=NEQ                                                   PR1 0353
0056                KRAY=KRAY+1                                                      PR1 0354
              C     DECODE THE NAME                                                  PR1 0355
0057                CALL DECNAM (NSYM(1,LJSV),0,JLOC,KVAL)                           PR1 0356
              C     MAKE HER LOOK LIKE STATE VARIABLE DERIV.                         PR1 0357
0058                KVALSV=KVAL+10000                                                PR1 0358
              C     POP THE EQUATION FOR THIS DERIV.                                 PR1 0359
0059                CALL POP (KVALSV)                                                PR1 0360
              C     FORCE THE LINE OUT                                               PR1 0361
0060         10     CALL PUTLIN                                                      PR1 0362
0061         11     CONTINUE                                                         PR1 0363
0062                RETURN                                                           PR1 0364
0063                END                                                              PR1 0365
```

```
0001                  SUBROUTINE DECNAM (NAME,KSUB,KLOC,KVAL)                                  PR1 0366
                C                                                                              PR1 0367
                C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR1 0368
                C                                                                              PR1 0369
                C     DECODE (NAME) INTO SUBSCRIPT LOCATION AND VALUE                          PR1 0370
                C                                                                              PR1 0371
                C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR1 0372
                C                                                                              PR1 0373
0002                  IMPLICIT REAL*8(A-H,O-Z)                                                 PR1 0374
0003                  COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300PR1 0375
                     1)                                                                        PR1 0376
0004                  COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,NO,NPLUS,NMINUS,NSL,NZER,NXM(3PR1 0377
                     1),NUM(10)                                                                PR1 0378
0005                  COMMON /CNTRL/ LIST,LCTROL(160),KXM(3)                                   PR1 0379
0006                  COMMON /FILES/ NPRINT,NPGM,NDTA,NEOU                                     PR1 0380
0007                  DIMENSION NAME(6), NELTS(7), NTYPE(6)                                    PR1 0381
0008                  DIMENSION NXTAB(5)                                                       PR1 0382
0009                  DATA NELTS/1HC,1HR,1HL,1HE,1HJ,1HP/                                      PR1 0383
0010                  DATA NTYPE/1HD,1HV,1HI,1HS,1HG,1HX/                                      PR1 0384
0011                  DATA NN/1HN/,NM/1HM/                                                     PR1 0385
0012                  DATA NXTAB/1HT,1HA,1HB,1HL,1HE/                                          PR1 0386
0013                  EQUIVALENCE (NI,NTYPE(3)), (NP,NELTS(6))                                 PR1 0387
0014                  EQUIVALENCE (NXTAB(1),NELTS(7)), (NCHART,NXTAB(1))                       PR1 0388
                C                                                                              PR1 0389
0015                  KLOC=0                                                                   PR1 0390
0016                  KELTS=0                                                                  PR1 0391
0017                  KTYPE=0                                                                  PR1 0392
0018                  KEXTD=0                                                                  PR1 0393
                C     LOCATE SYMBOL TYPE                                                       PR1 0394
0019                  DO 1 I=1,6                                                               PR1 0395
0020                  IF(NAME(1).EQ.NTYPE(I)) GO TO 7                                          PR1 0396
0021                1 CONTINUE                                                                 PR1 0397
0022                  KST=1                                                                    PR1 0398
                C     LOCATE SYMBOL ELEMENT                                                    PR1 0399
0023                2 DO 3 I=1,7                                                               PR1 0400
0024                  IF(NAME(KST).EQ.NELTS(I)) GO TO 8                                        PR1 0401
0025                3 CONTINUE                                                                 PR1 0402
                C     MUTUAL TYPE?                                                             PR1 0403
0026                  IF(NAME(KST).NE.NM) GO TO 4                                              PR1 0404
0027                  KELTS=9                                                                  PR1 0405
0028                  GO TO 9                                                                  PR1 0406
                C     VOLTAGE/CURRENT TYPE                                                     PR1 0407
0029                4 IF(KTYPE.EQ.2.OR.KTYPE.EQ.3) GO TO 12                                    PR1 0408
                C     FORM SYMBOL VALUE                                                        PR1 0409
0030                5 KVAL=KLOC+1000*KELTS+10000*KTYPE+1000000*KEXTD                           PR1 0410
0031                  IF(LIST.EQ.1) WRITE (NPRINT,6) NAME,KLOC,KELTS,KTYPE,KEXTD               PR1 0411
0032                6 FORMAT (8H DECNAM=,6A1,6H KLOC=,I3,7H KELTS=,I3,7H KTYPE=,I3,7H KEPR1 0412
                     1XTD=,I3)                                                                 PR1 0413
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0033                RETURN                                                  PR1 0414
              C     GOTTA TYPE NAME                                         PR1 0415
              C     XPRESSION IS SPECIAL                                    PR1 0416
0034           7    IF(I.EQ.6) GO TO 17                                     PR1 0417
              C     NOT AN "X"-TYPE                                         PR1 0418
0035                KST=2                                                   PR1 0419
0036                KTYPE=I                                                 PR1 0420
0037                GO TO 2                                                 PR1 0421
              C     GOTTA ELEMENT                                           PR1 0422
0038           8    KELTS=I                                                 PR1 0423
              C     PARAMETER IS SPECIAL                                    PR1 0424
0039                IF(KELTS.EQ.7) GO TO 5                                  PR1 0425
              C     LOCATE NAME IN SYMBOL TABLE                             PR1 0426
0040           9    DO 11 I=1,KSYM                                          PR1 0427
0041                DO 10 J=1,5                                             PR1 0428
0042                KSTJM=KST+J-1                                           PR1 0429
0043                IF(NAME(KSTJM).NE.NSYM(J,I)) GO TO 11                   PR1 0430
0044          10    CONTINUE                                                PR1 0431
              C     FOUND NAME IN STABLE                                    PR1 0432
0045                KLOC=I                                                  PR1 0433
0046                GO TO 5                                                 PR1 0434
0047          11    CONTINUE                                                PR1 0435
              C     NOT IN STABLE OR V/I TYPE                               PR1 0436
0048          12    KSTAR=2                                                 PR1 0437
              C     V#/I# TYPE                                              PR1 0438
0049                IF(LPUMP(NAME,KSTAR,1,KSTYL).EQ.1) GO TO 5              PR1 0439
              C     V#D/I#D TYPE - STATE VARIABLE DERIV.                    PR1 0440
0050                IF(NAME(KSTAR).EQ.ND) GO TO 15                          PR1 0441
              C     V#P/I#P TYPE                                            PR1 0442
0051                IF(NAME(KSTAR).EQ.NP) GO TO 16                          PR1 0443
              C     FORM LOCATION OF SYMBOL                                 PR1 0444
0052          13    KLOC=NPT(KSTYL)+KSUB-1                                  PR1 0445
0053                IF(KSUB.EQ.1000000) KLOC=KLOC+1                         PR1 0446
0054                KPL=KLOC                                                PR1 0447
0055                IF(KPL.GE.1000000) KPL=KPL-1000000                      PR1 0448
              C     X-ELEMENT TYPE                                          PR1 0449
0056                DO 14 KELTS=1,5                                         PR1 0450
0057                IF(NSYM(1,KPL).EQ.NELTS(KELTS)) GO TO 5                 PR1 0451
0058          14    CONTINUE                                                PR1 0452
0059                KELTS=0                                                 PR1 0453
0060                GO TO 5                                                 PR1 0454
              C     STATE VARIABLES                                         PR1 0455
0061          15    KTYPE=1                                                 PR1 0456
0062                GO TO 13                                                PR1 0457
              C     V2P, V3P, XI1P, OR XI5P                                 PR1 0458
0063          16    KEXTD=KSTYL+3                                           PR1 0459
0064                KTYPE=0                                                 PR1 0460
0065                KLOC=KSUB                                               PR1 0461
```

```
0066                    GO TO 5
                  C     "X"-TYPE
0067               17   KSTAR=2
                  C     "XM" TYPE
0068                    IF(NAME(KSTAR).EQ.NM) GO TO 18
                  C     "XI" TYPE
0069                    IF(NAME(KSTAR).NE.NI) GO TO 21
0070                    KSTAR=3
                  C     "XI#" TYPE
0071                    IF(LPUMP(NAME,KSTAR,1,KSTYL).EQ.1) GO TO 5
0072                    GO TO 16
0073               18   KLOC=KSUB
                  C     "XM"-ELEMENT TYPE
0074                    DO 19 KEXTD=1,3
0075                    IF(NAME(3).EQ.NXM(KEXTD)) GO TO 20
0076               19   CONTINUE
0077                    KEXTD=0
0078                    KLOC=0
0079                    GO TO 5
                  C     XM$I - TYPE
0080               20   IF(NAME(4).EQ.NI.OR.NAME(5).EQ.NI) KTYPE=6
0081                    GO TO 5
                  C     X TYPE, NON SPECIAL - "XTABLE"?
0082               21   IF(LOCATE(NAME,NXTAB,2,5).EQ.0) GO TO 22
                  C     "XT" TYPE?
0083                    IF(NAME(2).EQ.NCHART) GO TO 23
                  C     "XN##" TYPE
0084                    KSTAR=3
0085                    KPUMP=LPUMP(NAME,KSTAR,2,KSTYL)
0086                    KSTYL=KSTYL/10
0087                    IF(KPUMP.EQ.0.AND.KSTYL.NE.0) GO TO 13
0088                    KEXTD=7
0089                    KST=1
0090                    GO TO 9
                  C     "XTABLE"
0091               22   KELTS=8
0092                    KTYPE=0
0093                    GO TO 5
                  C     ERRORS
                  C
                  C     TABLE APPEARED IN XPRESSION ILLEGALLY
0094               23   CALL ERROR (30)
0095                    END
```

```
PR1 0462
PR1 0463
PR1 0464
PR1 0465
PR1 0466
PR1 0467
PR1 0468
PR1 0469
PR1 0470
PR1 0471
PR1 0472
PR1 0473
PR1 0474
PR1 0475
PR1 0476
PR1 0477
PR1 0478
PR1 0479
PR1 0480
PR1 0481
PR1 0482
PR1 0483
PR1 0484
PR1 0485
PR1 0486
PR1 0487
PR1 0488
PR1 0489
PR1 0490
PR1 0491
PR1 0492
PR1 0493
PR1 0494
PR1 0495
PR1 0496
PR1 0497
PR1 0498
PR1 0499
PR1 0500
PR1 0501
PR1 0502
PR1 0503
PR1 0504
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001               SUBROUTINE ERROR (NUM)                                         PR1 0505
             C                                                                    PR1 0506
             C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0507
             C                                                                    PR1 0508
             C     PUT ERROR NUMBER                                               PR1 0509
             C                                                                    PR1 0510
             C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0511
             C                                                                    PR1 0512
0002               COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU                           PR1 0513
             C                                                                    PR1 0514
0003               DIMENSION NBAD(34)                                             PR1 0515
0004               DATA NBADL/18/                                                 PR1 0516
0005               DATA NBAD/1,4,5,6,7,8,17,18,19,20,23,24,27,28,30,31,32,34/     PR1 0517
0006               WRITE (NPRINT,1) NUM                                           PR1 0518
0007           1   FORMAT (//,14H ERROR NUMBER ,I2,20H SEE ERROR LIST.....,//)    PR1 0519
0008               DO 2 I=1,NBADL                                                 PR1 0520
0009               IF(NUM.EQ.NBAD(I)) GO TO 4                                     PR1 0521
0010           2   CONTINUE                                                       PR1 0522
0011               WRITE (NPRINT,3)                                               PR1 0523
0012           3   FORMAT (38H STATE VARIABLE PRODUCTION SUPPRESSED.)             PR1 0524
0013               STOP 0                                                         PR1 0525
0014           4   WRITE (NPRINT,5)                                               PR1 0526
0015           5   FORMAT (48H SEVERE ERROR - SENSITIVITY ANALYSIS SUPPRESSED.)   PR1 0527
0016               STOP 8                                                         PR1 0528
0017               END                                                            PR1 0529
```

```
0001                SUBROUTINE FINIE                                                   PR1 0530
              C                                                                        PR1 0531
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0532
              C                                                                        PR1 0533
              C     FINISH UP THE STATE EQUATIONS PRODUCTION.                          PR1 0534
              C                                                                        PR1 0535
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 0536
              C                                                                        PR1 0537
0002                COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU                               PR1 0538
              C                                                                        PR1 0539
0003                WRITE (NEQU,1)                                                     PR1 0540
0004            1   FORMAT (1HC,79X,/,1HC,79X,/,1HC,35(2H<>),1H<,8X,/,1HC,79X,/,9H    PR1 0541
               1      END,71X)                                                         PR1 0542
0005                RETURN                                                             PR1 0543
0006                END                                                                PR1 0544
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001               SUBROUTINE GETEQS
              C
              C    <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
              C    GET EQUATIONS FROM SIMTR INTO (NPUSH) POINTED TO BY
              C    (KLOCA) AND KLOC
              C
              C    <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
0002               IMPLICIT REAL*8(A-H,O-Z)
0003               COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0004               COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                  1)
0005               COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0006               COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0007               COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
                  1),NUM(10)
0008               DIMENSION NAME(6), NAUGM(6), NIONBX(6)
0009               DATA NO/1HO/,NM/1HM/
0010               DATA NC/1HC/,NP/1HP/,NX/1HX/,NI/1HI/
0011               DATA NAUGM/1H/,1HA,1HU,1HG,1HM,1H//
0012               DATA NIONBX/1HI,1HO,1HN,1H ,1HX,1HM/
              C
0013               KSUBSV=0
              C    GIT NEXT CARD
0014          1    NEOF=LREAD(NPGM,0)
0015          2    IF(NRAY(1).EQ.NC) GO TO 1
              C    CHECK FOR "="
0016               DO 3 KEQ=8,50
0017               IF(NRAY(KEQ).EQ.NEQ) GO TO 5
0018          3    CONTINUE
0019               IF(LOCATE(NRAY,NAUGM,13,6).EQ.0) GO TO 14
0020               IF(LOCATE(NRAY,NIONBX,13,6).EQ.0) GO TO 18
              C    TEST "END"
0021          4    IF(NEOF.EQ.0) GO TO 1
0022               KLOC1=KLOC+1
0023               KLOCA(1,KLOC1)=0
0024               KLOCA(2,KLOC1)=KPUSH
0025               RETURN
              C    FOUND "="
0026          5    IF(LIST.EQ.1) WRITE (NPRINT,6) NRAY
0027          6    FORMAT (1X,80A1)
              C    GO ON PARAMETER OR XPRESSION
0028               IF(NRAY(7).EQ.NP.OR.(NRAY(7).EQ.NX.AND.NRAY(8).NE.NI.AND.NRAY(8).N
                  1E.NM)) GO TO 7
              C    IGNORE "DO" LOOPS
0029               IF(NRAY(7).EQ.NC.AND.NRAY(8).EQ.NO) GO TO 11
0030               KRAY=7
```

```
0031                KSUB=0                                                    PR1 0593
0032                KEQ1=KEQ+1                                                PR1 0594
                    C   IGNORE ZERO VALUES                                    PR1 0595
0033                IF(NRAY(KEQ1).EQ.NZER) GO TO 1                            PR1 0596
                    C   GET SYMBOL                                            PR1 0597
0034                IF(LPLLL(NRAY,KRAY,NAME,0,KSUB).EQ.-1) GO TO 12           PR1 0598
                    C   PUSH EQUATION                                         PR1 0599
0035                KRAY=KEQ+1                                                PR1 0600
0036                IF(KSUBSV.NE.0.AND.KSUB.EQ.1000000) KSUB=KSUBSV           PR1 0601
0037                KPSHSV=KPUSH                                              PR1 0602
0038                CALL PUSH (NAME,KSUB)                                     PR1 0603
0039                KSUBSV=0                                                  PR1 0604
                    C   CHECK FOR NOTHING PUSHED                              PR1 0605
0040                IF(KPSHSV.EQ.KPUSH) KLOC=KLOC-1                           PR1 0606
                    C   GET NEXT STUFF                                        PR1 0607
0041                GO TO 2                                                   PR1 0608
                    C   PARM. OR XPRESSION - SAVE NAME IN SYMBOL TABLE         PR1 0609
0042                7   KSTAR=7                                               PR1 0610
0043                    KSYM=KSYM+1                                           PR1 0611
0044                    IF(KSYM.GT.300) GO TO 13                              PR1 0612
0045                    DO 8 I=1,6                                            PR1 0613
0046                8   NSYM(I,KSYM)=NSP                                      PR1 0614
0047                    DO 9 I=1,6                                            PR1 0615
0048                    IF(NRAY(KSTAR).EQ.NLP.OR.NRAY(KSTAR).EC.NEQ) GO TO 10 PR1 0616
0049                    NSYM(I,KSYM)=NRAY(KSTAR)                              PR1 0617
0050                    KSTAR=KSTAR+1                                         PR1 0618
0051                9   CONTINUE                                              PR1 0619
0052                10  KRAY=KEQ+1                                           PR1 0620
                    C   PUSH EQUATION                                         PR1 0621
0053                    KPSHSV=KPUSH                                          PR1 0622
0054                    CALL PUSH (NSYM(1,KSYM),0)                            PR1 0623
0055                    IF(KPSHSV.EQ.KPUSH) KLOC=KLOC-1                       PR1 0624
0056                    GO TO 2                                               PR1 0625
                    C   DETERMINE IF ONE PASS DO LOOP                         PR1 0626
0057                11  KSUBSV=NUMB(NRAY,18,20,II)                           PR1 0627
0058                    KSBSV1=NUMB(NRAY,24,26,II)                            PR1 0628
0059                    IF(KSUBSV.NE.KSBSV1.OR.KSUBSV.EQ.1) KSUBSV=0          PR1 0629
0060                    GO TO 1                                               PR1 0630
                    C   ERRORS                                                PR1 0631
                    C                                                         PR1 0632
                    C   BAD SYMBOL                                           PR1 0633
0061                12  CALL ERROR (25)                                       PR1 0634
                    C   SYMBOL TABLE OVERFLOW                                 PR1 0635
0062                13  CALL ERROR (26)                                       PR1 0636
                    C   PARSE OUT XML, XMR, AND XMS LENGTHS                   PR1 0637
0063                14  IF(NRAY(31).NE.NCOM) GO TO 4                          PR1 0638
0064                    DO 15 I=1,3                                           PR1 0639
0065                    IF(NRAY(22).EQ.NXM(I)) GO TO 16                       PR1 0640
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0066            15  CONTINUE                                              PR1 0641
0067                GO TO 4                                               PR1 0642
0068            16  NEN=30                                                PR1 0643
0069                NST=28                                                PR1 0644
0070            17  KXM(I)=NUMB(NRAY,NST,NEN,II)                          PR1 0645
0071                IF(KXM(I).GT.7) CALL ERROR (35)                       PR1 0646
0072                GO TO 4                                               PR1 0647
0073            18  IF(NRAY(28).NE.NCOM) GO TO 4                          PR1 0648
0074                DO 19 I=1,3                                           PR1 0649
0075                IF(NRAY(19).EQ.NXM(I)) GO TO 20                       PR1 0650
0076            19  CONTINUE                                              PR1 0651
0077                GO TO 4                                               PR1 0652
0078            20  NEN=27                                                PR1 0653
0079                NST=25                                                PR1 0654
0080                GO TO 17                                              PR1 0655
0081                END                                                   PR1 0656
```

```
0001                SUBROUTINE MIDUP
              C
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
              C     WRITE DEBUG, MATRICES, AND HEADER
              C
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
0002                IMPLICIT REAL*8(A-H,O-Z)
0003                COMMON /HOLDIT/ HEAD(9,11),KHEAD,NDUM1,ELTS(300),KELTS,KEL,SDR(50)
                   1,NSDR(6,50),KSDR,NDUM3,XIC(50),KSV,NDUM4,PAR(100),NPAR(6,100),KDP,
                   2NDUM5,XTAB(500),NTAB(7,50),KTAB,KTAB2,KEJ
0004                COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
                   1),NUM(10)
0005                COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0006                COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0007                COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                   1)
0008                COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0009                EQUIVALENCE (TIME,CTROL(1))
0010                DATA NE/1HE/,NJ/1HJ/
              C
              C     PLT DEBUG
0011                IF(LIST.NE.1) GO TO 19
              C     DEBUG FACILITY
0012                WRITE (NPRINT,1) NPT,NLEN,KSYM
0013              1 FORMAT (18H POINTERS TO NAMES,8I7,/,18H LENGTHS OF NAMES ,8I7,/,7H
                   1 LIMIT=,I7)
0014                DO 2 I=1,KSYM
0015              2 WRITE (NPRINT,3) I,(NSYM(J,I),J=1,6)
0016              3 FORMAT (1H ,I4,2H  ,6A1)
              C
              C
0017                WRITE (NPRINT,4) (I,NPUSH(I),NPUSH(I),I=1,KPUSH)
0018              4 FORMAT (200(/,1X,4(2I9,8X,A1,3X)))
0019                WRITE (NPRINT,5) (I,KLOCA(1,I),KLOCA(2,I),KLOCA(3,I),I=1,KLOC)
0020              5 FORMAT (200(/,1X,3(4I9,3X)))
0021                WRITE (NPRINT,6) KXM
0022              6 FORMAT (6H #XML=,I3,7H #XMRG=,I3,6H #XMS=,I3)
0023                WRITE (NPRINT,7) TIME,KELTS,KSV,KDP,KSDR,KTAB
0024              7 FORMAT (12H <><><>TIME=,1PD14.7,/,7H KELTS=,I3,5H KSV=,I3,5H KDP=,
                   1I3,6H KSDR=,I3,6H KTAB=,I3,//)
0025                DO 10 I=1,KELTS
0026                IF(NSYM(1,I).NE.NE.AND.NSYM(1,I).NE.NJ) GO TO 8
0027                IF(NT(I).EQ.0) GO TO 10
0028              8 WRITE (NPRINT,9) I,ELTS(I)
0029              9 FORMAT (12H <><><>ELNO=,I3,7H ELVAL=,1PD14.7)
0030             10 CONTINUE
```

ORIGINAL PAGE IS OF POOR QUALITY

```
                C     PUT DEFINED PARAMETER VALUES                                    PR1 0705
0031                  IF(KDP.EQ.0) GO TO 13                                           PR1 0706
0032                  DO 11 I=1,KDP                                                   PR1 0707
0033             11   WRITE (NPRINT,12) I,PAR(I)                                      PR1 0708
0034             12   FORMAT (13H <><><>DPNUM=,I3,7H DPVAL=,1PD14.7)                  PR1 0709
                C     PUT SOURCE DERIVATIVE VALUES                                    PR1 0710
0035             13   IF(KSDR.EQ.0) GO TO 16                                          PR1 0711
0036                  DO 14 I=1,KSDR                                                  PR1 0712
0037             14   WRITE (NPRINT,15) I,SDR(I)                                      PR1 0713
0038             15   FORMAT (14H <><><>SDRNUM=,I3,8H SDRVAL=,1PD14.7)                PR1 0714
                C     PUT TABLE VALUES                                                PR1 0715
0039             16   IF(KTAB.EQ.0) GO TO 19                                          PR1 0716
0040                  DO 17 I=1,KTAB                                                  PR1 0717
0041             17   WRITE (NPRINT,18) I,XTAB(I)                                     PR1 0718
0042             18   FORMAT (14H <><><>TABNUM=,I3,8H TABVAL=,1PD14.7)                PR1 0719
                C     PUT STATE VARIABLE EQUATION SEPARATOR                           PR1 0720
0043             19   WRITE (NEQU,20)                                                 PR1 0721
0044             20   FORMAT (1HC,79X,/,1HC,35(2H<>),1H<,8X,/,1HC,79X,/,1HC,79X)      PR1 0722
0045                  RETURN                                                          PR1 0723
0046                  END                                                             PR1 0724
```

```
0001                SUBROUTINE POP (KPOPO)
              C
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
              C     POP THE EQUATION POINTED TO BY VALUE (KPOPO) FROM (NPUSH)
              C
              C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
0002                IMPLICIT REAL*8(A-H,O-Z)
0003                COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                   1)
0004                COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0005                COMMON /SFGR/ NSHUV(500),KSHUV,KST,KEND,KSUB,KEXTD,KPAR
0006                COMMON /FILES/ NPRINT,NPGM,NCTA,NEQU
0007                COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,NC,NPLUS,NMINUS,NSL,NZER,NXM(3
                   1),NUM(10)
0008                COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0009                DIMENSION NDVI(3)
0010                DATA NDVI/1HD,1HV,1HI/
              C
0011                KSHUV=0
0012                KSUBNI=0
              C     LOCATE THIS SYMBOL VALUE IN STACK
0013                IF(LOCTST(ILOC,KPOPO).NE.0) GO TO 20
              C     CHECK IF PARENS. AROUND SUM TERMS REQUIRED
0014          1     KPAR=KLOCA(3,ILOC)
0015                IF(KPAR.EQ.1.OR.NRAY(KRAY-1).EQ.NSL) CALL PUTCHR (NLP)
              C     FORM POP LOCATERS
0016          2     KST=KLOCA(2,ILOC)
0017                ILOCP=ILOC+1
0018                KEND=KLOCA(2,ILOCP)-1
              C     POINT TO VALUE - NOT NAME/CHAR
0019          3     IF(NPUSH(KST).GT.0.AND.NPUSH(KST).LT.100000000) GO TO 7
              C     NAME/CHAR FOUND
0020                IF(KPAR.NE.0.OR.NPUSH(KST).NE.NPLUS) CALL PUTCHR (NPUSH(KST))
0021                KST=KST+1
              C     END OF POPPING?
0022          4     IF(KST.LE.KEND) GO TO 3
              C     YES - PUT RIGHT PAREN IF LEFT ONE PUT
0023                IF(KPAR.EQ.1) CALL PUTCHR (NRP)
              C     CHECK FOR MORE EQUATIONS THIS ITEM
0024          5     DO 6 ILCSV=2,KLOC
0025                IF(KEND+1.EQ.KLOCA(2,ILCSV).AND.KLOCA(1,ILCSV-1).EQ.KLOCA(1,ILCSV)
                   1) GO TO 19
0026          6     CONTINUE
              C     ANYTHING LEFT ON STACK?
0027                IF(LGRAB(KPOP).EQ.0) GO TO 17
              C     NO - WE DONE?
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0028                IF(KST.LE.KEND) GO TO 3                                    PR1 0773
                C   YES                                                        PR1 0774
0029                RETURN                                                     PR1 0775
                C   SYMBOL VALUE GIVEN                                         PR1 0776
0030             7  KPOP=NPUSH(KST)                                            PR1 0777
                C   PUT PARENS. IF DIVIDE                                      PR1 0778
0031                IF(NRAY(KRAY-1).NE.NSL) GO TO 8                            PR1 0779
0032                KPAR=1                                                     PR1 0780
0033                CALL PUTCHR (NLP)                                          PR1 0781
                C   CHECK FOR INVERTED ARRAY                                   PR1 0782
0034             8  KMX=MOD(KPOP,1000000)/10000                                PR1 0783
0035                IF(KMX.EQ.16.OR.KMX.EQ.26.OR.KMX.EQ.36) GO TO 16           PR1 0784
                C   FORM CORRECT VALUES FROM SUBSCRIPT                         PR1 0785
0036             9  IF(KSUB.NE.0.AND.KPOP.GT.1000000) KPOP=KPOP+KSUB-1000001   PR1 0786
0037                IF(KPOP/100000.NE.0.AND.KPOP/100000.NE.7) KPOP=KPOP+1      PR1 0787
0038                KST=KST+1                                                  PR1 0788
                C   PUSH THIS VALUE AND POINT TO NEXT EQUATION                 PR1 0789
0039                IF(LSHUV(KPOP,ILOC,KPOPO).EQ.0) GO TO 12                   PR1 0790
                C   GET NEXT EQUATION                                          PR1 0791
0040            10  KPOS=MOD(KPOPO,1000)                                       PR1 0792
                C   LOCATION NON-EXISTENT - ZERO VALUE THEN                    PR1 0793
0041                IF(KPOS.EQ.0) GO TO 15                                     PR1 0794
0042                KTYP=MOD(KPOPO,1000000)/10000                              PR1 0795
                C   EXTENDED SYMBOL AT BAD LOCATION                            PR1 0796
0043                IF(MOD(KPOPO,1000000)/100000.NE.0) GO TO 15                PR1 0797
                C   NOT ENOUGH ROOM TO PUT SYMBOL - PUT THE LINE               PR1 0798
0044                IF(KRAY+6.GT.72) CALL PUTLIN                               PR1 0799
                C   PUT "V/I" IF NECESSARY                                     PR1 0800
0045                IF(KTYP.LE.0) GO TO 11                                     PR1 0801
0046                CALL PUTCHR (NDVI(KTYP))                                   PR1 0802
                C   PUT NAME                                                   PR1 0803
0047            11  CALL PUTNAM (NSYM(1,KPOS))                                 PR1 0804
0048                GO TO 4                                                    PR1 0805
                C   NAME FOUND - PUT ELEMENT                                   PR1 0806
0049            12  IF(KPOP.LT.4000) GO TO 10                                  PR1 0807
0050                IF(KPOP/1000.EQ.9) GO TO 10                                PR1 0808
0051                IF(ILOC.EQ.0) GO TO 4                                      PR1 0809
                C   FORCE EQUATION IF NOT EXTENDED SYMBOL                      PR1 0810
0052                IF(KEXTD.EQ.0) GO TO 1                                     PR1 0811
0053                IF(KSHUV.EQ.0) GO TO 21                                    PR1 0812
                C   LOCATE THIS EQUATION SUBSCRIPT IN STACK                    PR1 0813
0054                DO 13 IS=1,KSHUV,5                                         PR1 0814
0055                IT=KSHUV-IS-3                                              PR1 0815
0056                IF(NSHUV(IT).NE.0) GO TO 14                                PR1 0816
0057            13  CONTINUE                                                   PR1 0817
                C   SUBSCRIPT NOT IN STACK                                     PR1 0818
0058                GO TO 21                                                   PR1 0819
                C   GET SUBSCRIPT                                              PR1 0820
```

```
0059          14   KPOP=NPUSH(KST-1)+NSHUV(IT)                                    PRI 0821
0060               GO TO 1                                                        PRI 0822
               C   PUT ZERO VALUE                                                 PRI 0823
0061           15  CALL PUTCHR (NZER)                                             PRI 0824
0062               GO TO 4                                                        PRI 0825
               C   START ARRAY MULTIPLY LOOP                                     PRI 0826
               C   INVALID SUBSCRIPT ON INVERTED GUY                             PRI 0827
0063           16  IF(KSUBN1.GT.1) GO TO 9                                        PRI 0828
0064               IF(KPOP.LT.1000000) GO TO 21                                   PRI 0829
               C   SAVE OLD SUBSCR AND GEN NEW ONE                              PRI 0830
0065               KSUBN=KSUB                                                     PRI 0831
0066               KSUBN1=1                                                       PRI 0832
               C   PUT INVERTED ARRAY NAME                                       PRI 0833
0067               KMXA=KMX/10                                                    PRI 0834
0068               KSUB=KSUBN1                                                    PRI 0835
0069               CALL PUTMX (KMXA,KSUBN,KSUB)                                   PRI 0836
0070               GO TO 9                                                        PRI 0837
               C   CONTINUE MATRIX MULTIPLY                                      PRI 0838
0071           17  KMXB=KPOP/100000                                              PRI 0839
0072               IF(KMXB.NE.16.AND.KMXB.NE.26.AND.KMXB.NE.36) GO TO 4           PRI 0840
               C   INCREMENT SUBSCRIPT 2                                         PRI 0841
0073               IF(KSUBN1.EQ.KXM(KMXA)) GO TO 18                               PRI 0842
0074               KSUBN1=KSUBN1+1                                                PRI 0843
0075               KSUB=KSUBN1                                                    PRI 0844
0076               KST=NSHUV(KSHUV-3)                                             PRI 0845
0077               NSHUV(KSHUV-4)=KSUB                                            PRI 0846
0078               KPOPC=KPOP                                                     PRI 0847
               C   PUT NEW INV. ARRAY NAME                                       PRI 0848
0079               CALL PUTMX (KMXA,KSUBN,KSUB)                                   PRI 0849
007A               GO TO 4                                                        PRI 085C
               C   END OF MATRIX MPY                                             PRI 0851
007B           18  KSUB=KSUBN                                                    PRI 0852
007C               NSHUV(KSHUV-3)=NSHUV(KSHUV-2)+1                                PRI 0853
007D               KSUBN1=0                                                       PRI 0854
007E               GO TO 5                                                        PRI 0855
007F           19  ILOC=ILOSV                                                    PRI 0856
0080               IF(KPAR.EQ.1) KRAY=KRAY-1                                      PRI 0857
0081               KPAR=KLOCA(3,ILOC)                                             PRI 0858
0082               GO TO 2                                                        PRI 0859
               C   CHECK FOR INVALID STATE VARIABLE                             PRI 0860
0083           20  IF(KPOPC/100000.LT.1.OR.KPOPC/100000.GT.3) GO TO 22            PRI 0861
               C   SIMPLY A ZERO VALUE ARRAY ELEMENT                            PRI 0862
0084               CALL PUTCHR (NZER)                                             PRI 0863
0085               RETURN                                                         PRI 0864
               C   ERRORS                                                        PRI 0865
               C                                                                  PRI 0866
               C   INVALID SUBSCRIPT                                            PRI 0867
0086           21  CALL ERROR (31)                                               PRI 0868
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                C     INVALID NAME TO POP                                  PR1 0869
 0C93            22   CALL ERROR (32)                                      PR1 0870
 0C94                 END                                                  PR1 0871
```

```
0001                  SUBROUTINE PUSH (NAME,KSUB)
                C
                C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                C
                C     PUSH THE EQUATION POINTED TO BY (NAME) INTO (NPUSH)
                C
                C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                C
0002                  COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                     1)
0003                  COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0004                  COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0005                  COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
                     1),NUM(10)
0006                  DIMENSION NAME(6), NAM1(6)
                C
0007                  KGO=0
                C     DECODE NAME - POINT TO SYMBOL TABLE
0008                  CALL DECNAM (NAME,KSUB,KLOC1,KVAL)
                C     BAD NAME - I.E. NOT IN SYMBOL TABLE
0009                  IF(KVAL.EQ.0) GO TO 4
                C     PUSH VALUE EQUATION
0010                  KLOC=KLOC+1
0011                  KLOCA(1,KLOC)=KVAL
0012                  KLOCA(2,KLOC)=KPUSH
0013                  KLOCA(3,KLOC)=0
0014             1    KGO=1
0015                  KSHSV=KPUSH
                C     GET EQUATION SYMBOL
0016                  IF(LPULL(NRAY,KRAY,NAM1,KDO,KSUB1).EQ.-1) RETURN
                C     FORM SUM OF PRODUCTS FLAG
0017                  IF(KDO.EQ.2.AND.(KGO.EQ.1.OR.NPUSH(KSHSV).EQ.NMINUS)) KLOCA(3,KLOC
                     1)=1
0018                  KGO=1
                C     DECODE EQUATION SYMBOL
0019                  CALL DECNAM (NAM1,KSUB1,KLOC1,KVAL)
                C     INVALID SYMBOL
0020                  IF(KVAL.EQ.0) GO TO 5
                C     CHECK TO PUSH FULL NAME
0021                  IF(KVAL.EQ.6000.OR.KVAL.EQ.7000.OR.KVAL.EQ.8000) GO TO 2
                C     "X=X+" FORMAT - IGNORE
0022                  IF(KVAL.EQ.KLOCA(1,KLOC)) GO TO 1
                C     PUSH EQUATION SYMBOL VALUE ONLY
0023                  NPUSH(KPUSH)=KVAL
0024                  KPUSH=KPUSH+1
0025                  IF(KPUSH.GT.KPMAX) GO TO 6
0026                  GO TO 1
                C     PUSH FULL NAME
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0027              2     KS=1                                                    PR1 0920
0028              3     NPUSH(KPUSH)=NAM1(KS)                                   PR1 0921
0029                    KPUSH=KPUSH+1                                           PR1 0922
0030                    KS=KS+1                                                 PR1 0923
0031                    IF(KPUSH.GT.KPMAX) GO TO 6                              PR1 0924
                  C     END OF NAME?                                            PR1 0925
0032                    IF(NAM1(KS).EQ.NSP) GO TO 1                             PR1 0926
0033                    IF(KS.GT.6) GO TO 1                                     PR1 0927
0034                    GO TO 3                                                 PR1 0928
                  C     ERRORS                                                  PR1 0929
                  C                                                             PR1 093C
                  C     PUSHED NAME NOT IN SYMBOL TABLE                         PR1 0931
0035              4     CALL ERROR (27)                                         PR1 0932
                  C     EQUATION NAME NOT IN SYMBOL TABLE                       PR1 0933
0036              5     CALL ERROR (28)                                         PR1 0934
                  C     STACK OVERFLOW                                          PR1 0935
0037              6     CALL ERROR (29)                                         PR1 0936
0038                    END                                                     PR1 0937
```

```
0001            SUBROUTINE PUTCHR (NCHAR)
           C
           C    <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
           C
           C    PLACE THE CHARACTER (NCHAR) INTO THE OUTPUT ARRAY (NRAY)
           C    IF (NRAY) LONGER THAN 72, PUT (NRAY) ON FT02F001 AS FORTRAN
           C
           C    <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
           C
0002            COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
               1)
           C
0003            IF(KRAY.GT.72) CALL PUTLIN
0004            NRAY(KRAY)=NCHAR
0005            KRAY=KRAY+1
0006            RETURN
0007            END
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0001                  SUBROUTINE PUTIT (KGO,KSTOP,NCHAR,KLEN,KNT,KPT)                    PR1 0955
                C                                                                        PR1 0956
                C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR1 0957
                C                                                                        PR1 0958
                C     PUT THE NAMES FROM ELEMENT LIST STARTING AT TYPE (KGO)             PR1 0959
                C     TO TYPE (KSTOP) IN OUTPUT, PRECEDE BY (NCHAR) IF                   PR1 0960
                C     KLEN NO ZERO AND SET NUMBER ELTS IN (KNT)                          PR1 0961
                C     AND POINTER OF FIRST NAME IN (KPT)                                 PR1 0962
                C                                                                        PR1 0963
                C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR1 0964
                C                                                                        PR1 0965
0002                  COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3PR1 0966
                     1),NUM(10)                                                          PR1 0967
0003                  COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300PR1 0968
                     1)                                                                  PR1 0969
0004                  COMMON /HOLDIT/ HEAD(9,11),KHEAD,NDUM1,ELTS(300),KELTS,KEL,SOR(50)PR1 0970
                     1,NSOR(6,50),KSOR,NDUM3,XIC(50),KSV,NDUM4,PAR(100),NPAR(6,100),KDP,PR1 0971
                     2NDUM5,XTAB(500),NTAB(7,50),KTAB,KTAB2,KEJ                          PR1 0972
0005                  DIMENSION MRAY(6)                                                  PR1 0973
                C                                                                        PR1 0974
                C     ZERO POINTER AND COUNT                                             PR1 0975
0006                  KPT=0                                                              PR1 0976
0007                  KNT=0                                                              PR1 0977
                C     LOOP THRU TYPES                                                    PR1 0978
0008                  DO 4 IPT=KGO,KSTOP                                                 PR1 0979
                C     FIND LENGTH OF THIS TYPE                                           PR1 0980
0009                  MPT=NLEN(IPT)                                                      PR1 0981
                C     ZERO LENGTH IS NO PUTSKIS                                          PR1 0982
0010                  IF(MPT.EQ.0) GO TO 4                                               PR1 0983
                C     FIND THIS GUYS POINTER LESS ONE                                    PR1 0984
0011                  LPT=NPT(IPT)-1                                                     PR1 0985
                C     LOOP ON THAT COUNT                                                 PR1 0986
0012                  DO 3 JPT=1,MPT                                                     PR1 0987
                C     PUT THE EXTRA CHAR IF ONE                                          PR1 0988
0013                  NNPT=0                                                             PR1 0989
0014                  IF(KLEN.NE.1) GO TO 1                                              PR1 0990
0015                  MRAY(1)=NCHAR                                                      PR1 0991
0016                  NNPT=1                                                             PR1 0992
                C     FORM THIS GUYS ABSOLUTE POINTER                                    PR1 0993
0017                1 LJPT=LPT+JPT                                                       PR1 0994
                C     BUILD FIRST POINTER                                                PR1 0995
0018                  IF(KPT.EQ.0) KPT=LJPT                                              PR1 0996
                C     PUT THIS NAME                                                      PR1 0997
0019                  MS=6-NNPT                                                          PR1 0998
0020                  DO 2 M=1,MS                                                        PR1 0999
0021                2 MRAY(M+NNPT)=NSYM(M,LJPT)                                          PR1 1000
0022                  CALL PUTNAM (MRAY)                                                 PR1 1001
                C     UPDATE COUNT                                                       PR1 1002
```

```
0023                    KNT=KNT+1                                       PR1 1003
                  C     PUT A COMMA BETWIXT EM                          PR1 1004
0024                    CALL PUTCHR (NCCM)                              PR1 1005
0025                3   CONTINUE                                        PR1 1006
0026                4   CONTINUE                                        PR1 1007
0027                    RETURN                                          PR1 1008
0028                    END                                             PR1 1009
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001             SUBROUTINE PUTLIN
           C
           C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
           C
           C     WRITE THE CHARACTER STRING (NRAY) IN FORTRAN FORMAT
           C     AFTER BLANKING END OF LINE. CONTINUATIONS ADDED
           C
           C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
           C
0002             COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
                1)
0003             COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0004             COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
                1),NUM(10)
0005             DIMENSION NCOMM(6)
0006             DATA NCOMM/1HC,1HO,1HM,1HM,1HO,1HN/
0007             DATA NAST/1H$/
           C
           C     NOTHING ON LINE NOW
0008             IF(KRAY.EQ.7) RETURN
0009             NRAY(6)=NSP
           C     IF "COMMON " OR "=" APPEARS, THEN NO CONTINUATION
0010             IF(LOCATE(NRAY,NCOMM,7,6).EQ.0) GO TO 2
0011             KND=MIN0(20,KRAY)
0012             DO 1 I=8,KND
0013             IF(NRAY(I).EQ.NEQ) GO TO 2
0014         1   CONTINUE
0015             NRAY(6)=NAST
           C     BLANK TO END OF LINE AND PUT LINE
0016         2   DO 3 I=KRAY,73
0017         3   NRAY(I)=NSP
0018             WRITE (NEQU,4) (NRAY(I),I=1,72)
0019         4   FORMAT (72A1,8X)
0020             KRAY=7
0021             RETURN
0022             END
```

```
0001          SUBROUTINE PUTMX (KMX,KSUBN,KSUBN1)
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
        C     PUT 'XM%I(##1,##2)' WHERE % IS 'L','R', OR 'S' AND
        C     ##1 IS KSUBN, AND ##2 IS KSUBN1
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
0002          COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
             1),NUM(10)
0003          DIMENSION NAMX(6)
0004          DATA NAMX/1HX,1HM,1H ,1HI,1H(,1H /
        C
0005          NAMX(3)=NXM(KMX)
0006          IF(KSUBN1.GT.1) CALL PUTCHR (NPLUS)
0007          CALL PUTNAM (NAMX)
0008          CALL BCDOUT (KSUBN)
0009          CALL PUTCHR (NCOM)
0010          CALL BCDOUT (KSUBN1)
0011          CALL PUTCHR (NRP)
0012          RETURN
0013          END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                  SUBROUTINE PUTNAM (NAME)                                                  PR1 1069
                    C                                                                           PR1 1070
                    C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1071
                    C                                                                           PR1 1072
                    C     PUT THE (UP TO) SIX CHARACTER LIST (NAME) IN THE OUTPUT               PR1 1073
                    C     STRING (NRAY). WRITE (NRAY) IF OVER 72 CHARS. LONG                    PR1 1074
                    C                                                                           PR1 1075
                    C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1076
                    C                                                                           PR1 1077
0002                  COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300PR1 1078
                     1)                                                                         PR1 1079
0003                  COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3PR1 1080
                     1),NUM(10)                                                                 PR1 1081
0004                  DIMENSION NAME(6), NEW(6)                                                 PR1 1082
                    C                                                                           PR1 1083
                    C     RE-FORM MECHANICAL NAME                                               PR1 1084
0005                  CALL BACK (NAME,NEW)                                                      PR1 1085
0006                  LEN=1                                                                     PR1 1086
0007                1 IF(KRAY.GT.72) CALL PUTLIN                                                PR1 1087
0008                  NRAY(KRAY)=NEW(LEN)                                                       PR1 1088
0009                  KRAY=KRAY+1                                                               PR1 1089
                    C     END OF LINE?                                                          PR1 1090
0010                  LEN=LEN+1                                                                 PR1 1091
0011                  IF(LEN.GT.6) RETURN                                                       PR1 1092
0012                  IF(NEW(LEN).EQ.NSP) RETURN                                                PR1 1093
0013                  GO TO 1                                                                   PR1 1094
0014                  END                                                                       PR1 1095
```

```
                      SUBROUTINE SETUP                                                  PR1 1096
                                                                                        PR1 1097
C0001                 C                                                                 PR1 1098
                      C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1099
                      C                                                                 PR1 1100
                      C     GET ELEMENT NAMES AND POINTERS INTO (NSYM),(NPT),(NLEN), AND PR1 1101
                      C     FROM SCEPTRE ROUTINE "SIMUL8"                               PR1 1102
                      C                                                                 PR1 1103
                      C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1104
                      C                                                                 PR1 1105
0002                        IMPLICIT REAL*8(A-H,O-Z)                                    PR1 1106
0003                        COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU                        PR1 1107
0004                        COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300PR1 1108
                           1)
0005                        COMMON /CNTRL/ LIST,L1,TIME,S1(21),XNOPRQ,S2,XNHEAD,S3(6),XRUNNO,SPR1 1109
                           14(11),XWRTSM,S5,XWPDBG,S6(16),XEXSTP,S7(27),XNEY,XNE7DT,XNBC,XNBR,PR1 1110
                           2XNBL,XNLC,XNLR,XNLL,XNJX,XNJ8DT,XNJO,XNJ9,S8(2),XNM33,XNM36,XNM66,PR1 1111
                           3XNDP,XNDPC,XNELTS,S9(50),KXM(3)                             PR1 1112
0006                        COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3PR1 1113
                           1),NUM(10)                                                   PR1 1114
0007                        COMMON /HOLDIT/ HEAD(9,11),KHEAD,NDUM1,ELTS(300),KELTS,KEL,SDR(50)PR1 1115
                           1,NSDR(6,50),KSDR,NDUM3,XIC(50),KSV,NDUM4,PAR(100),NPAR(6,100),KDP,PR1 1116
                           2NDUM5,XTAB(500),NTAB(7,50),KTAB,KTAB2,KEJ                   PR1 1117
                                                                                        PR1 1118
                      C                                                                 PR1 1119
0008                        DIMENSION NBAD(13), NELS(8), NLOC(8)                        PR1 1120
0009                        DIMENSION DLM(300), NMUT(6,50)                              PR1 1121
0010                        DIMENSION NSIMTR(5), NSRCDR(6), NTABLE(6), NMAT8(6)         PR1 1122
0011                        DIMENSION NXMUTL(5), NNAMES(5), NPRMTR(6), NDERIV(6)        PR1 1123
0012                        DIMENSION NIONBX(5), NOUTPT(6), NNDPC(3), NDUM(6)           PR1 1124
0013                        DIMENSION CTROL(160)                                        PR1 1125
0014                        EQUIVALENCE (CTROL(1),TIME)                                 PR1 1126
0015                        DATA NBAD/34,45,55,71,91,99,101,102,111,121,131,141,151/    PR1 1127
0016                        DATA NELS/96,97,98,93,94,95,92,100/                         PR1 1128
0017                        DATA NLOC/7,4,5,6,1,2,3,8/                                  PR1 1129
0018                        DATA NSIMTR/1HS,1HI,1HM,1HT,1HR/                            PR1 1130
0019                        DATA NNAMES/1HN,1HA,1HM,1HE,1HS/                            PR1 1131
0020                        DATA NPRMTR/1HP,1HR,1HM,1HT,1HR,1HS/                        PR1 1132
0021                        DATA NDERIV/1HC,1HE,1HR,1HI,1HV,1HS/                        PR1 1133
0022                        DATA NIONBX/1HI,1HC,1HN,1H ,1HX/                            PR1 1134
0023                        DATA NOUTPT/1HO,1HU,1HT,1HP,1HT,1HS/                        PR1 1135
0024                        DATA NAST/1H*/                                              PR1 1136
0025                        DATA NNDPC/1H,,1HD,1HP/                                     PR1 1137
0026                        DATA NSRCDR/1HS,1HR,1HC,1HD,1HR,1HV/                        PR1 1138
0027                        DATA NTABLE/1HT,1HA,1HB,1HL,1HE,1HS/                        PR1 1139
0028                        DATA NXMUTL/1HX,1HM,1HU,1HT,1HL/                            PR1 1140
                      C                                                                 PR1 1141
                      C     GET DATA                                                    PR1 1142
                      C     READ TILL SUBROUTINE SIMTR FOUND                            PR1 1143
0029                        KMUT=0
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0030          1   READ (NPCM,2,END=39) NRAY
0031          2   FORMAT (80A1)
0032              IF(LOCATE(NRAY,NSIMTR,9,5).EQ.1) GO TO 1
              C   CHECK WHICH SCEPTRE IS USED
0033              READ (NPGM,3) NRAY
0034          3   FORMAT (////////,80A1)
0035              NAC=70
0036              IF(NRAY(6).EQ.NAST) NAC=160
              C   READ CONTROLS
0037              READ (NDTA,4,END=38) (CTROL(I),I=1,NAC)
0038          4   FORMAT (1P5D14.7)
0039              IF(NAC.EQ.160) GO TO 6
0040              DO 5 I=71,160
0041          5   CTROL(I)=0.0D0
              C   LOCATE CERTAIN STRINGS
0042          6   IF(LREAD(NPGM,0).EQ.1) GO TO 40
0043          7   IF(LOCATE(NRAY,NSRCDR,15,6).EQ.0) GO TO 20
0044              IF(LOCATE(NRAY,NTABLE,15,6).EQ.0) GO TO 22
0045              IF(LOCATE(NRAY,NXMUTL,15,5).EQ.0) GO TO 24
0046              IF(NAC.EQ.160) GO TO 8
0047              IF(LOCATE(NRAY,NNAMES,15,5).EQ.0) GO TO 26
0048              IF(LOCATE(NRAY,NPRMTR,15,6).EQ.0) GO TO 28
0049              IF(LOCATE(NRAY,NDERIV,15,6).EQ.0) GO TO 30
0050              IF(LOCATE(NRAY,NIONBX,13,5).EQ.0) GO TO 32
0051          8   IF(LOCATE(NRAY,NOUTPT,15,6).EQ.0) GO TO 9
0052              GO TO 6
              C   CHECK FOR INVALID FLAGS IN CONTROLS.
0053          9   DO 10 IER=1,13
0054              KLOC=NBAD(IER)
0055              IF(CTROL(KLOC).NE.0.0D0) GO TO 36
0056         10   CONTINUE
              C   FORM DEBUG FLAG
0057              LIST=XWRDBG+XWRTSM
0058              IF(LIST.GT.0) LIST=1
              C   "EXECUTE SETUP PHASE ONLY MUST BE FLAGGED
0059              IF(XEXSTP.NE.1D0) GO TO 37
              C   FORMA ELEMENT TYPE POINTERS AND LENGTHS
0060              NPT(7)=1
0061              DO 11 I=1,8
0062              MPT=NLOC(I)
0063              KLOC=NELS(MPT)
0064              NLEN(MPT)=CTROL(KLOC)
0065              IF(MPT.EQ.7) GO TO 11
0066              MPTM=NLOC(I-1)
0067              NPT(MPT)=NPT(MPTM)+NLEN(MPTM)
0068         11   CONTINUE
0069              DO 12 I=1,8
0070              IF(NLEN(I).EQ.0) NPT(I)=0
```

PR1 1144
PR1 1145
PR1 1146
PR1 1147
PR1 1148
PR1 1149
PR1 1150
PR1 1151
PR1 1152
PR1 1153
PR1 1154
PR1 1155
PR1 1156
PR1 1157
PR1 1158
PR1 1159
PR1 1160
PR1 1161
PR1 1162
PR1 1163
PR1 1164
PR1 1165
PR1 1166
PR1 1167
PR1 1168
PR1 1169
PR1 1170
PR1 1171
PR1 1172
PR1 1173
PR1 1174
PR1 1175
PR1 1176
PR1 1177
PR1 1178
PR1 1179
PR1 1180
PR1 1181
PR1 1182
PR1 1183
PR1 1184
PR1 1185
PR1 1186
PR1 1187
PR1 1188
PR1 1189
PR1 1190
PR1 1191

```
0071          12   CONTINUE                                                    PR1 1192
              C    GIT HEADING CARDS                                           PR1 1193
0072               KHEAD=XNHEAD                                                PR1 1194
0073               IF(KHEAD.NE.0) READ (NDTA,13) HEAD                          PR1 1195
0074          13   FORMAT (9A8)                                                PR1 1196
              C    GIT ELEMENT VALUES AND NAMES                                PR1 1197
0075               KELTS=XNELTS                                                PR1 1198
0076               READ (NDTA,14) (ELTS(I),I=1,KELTS)                          PR1 1199
0077          14   FORMAT (5D14.7)                                             PR1 1200
0078               IF(NAC.EQ.160) READ (NDTA,15) ((NSYM(I,J),I=1,6),J=1,KELTS) PR1 1201
0079          15   FORMAT (72A1)                                               PR1 1202
0080               KSYM=KELTS                                                  PR1 1203
              C    SKIP OVER ELEMENT IC VALUES                                 PR1 1204
0081               READ (NDTA,14) (DUM(I),I=1,KELTS)                           PR1 1205
0082               READ (NDTA,14) (DUM(I),I=1,KELTS)                           PR1 1206
0083               IF(KMUT.NE.0) READ (NDTA,14) (DUM(I),I=1,KMUT)              PR1 1207
              C    GIT SOURCE DERV VALUES                                      PR1 1208
0084               IF(KSDR.NE.0) READ (NDTA,14) (SDR(I),I=I,KSDR)              PR1 1209
              C    SKIP OVER VNICT AND INICT VALUES                            PR1 1210
0085               NSAVE=NLEN(1)+NLEN(3)+NLEN(4)+NLEN(6)                       PR1 1211
0086               IF(NSAVE.NE.0) READ (NDTA,14) (DUM(I),I=1,NSAVE)            PR1 1212
              C    SKIP OVER DEF PARM DERIVS VALUES                            PR1 1213
0087               NDPD=XNDPD                                                  PR1 1214
0088               IF(NDPD.NE.0) READ (NDTA,14) (DUM(I),I=1,NDPD)              PR1 1215
              C    GIT I3DOT AND V4DOT VALUES                                  PR1 1216
0089               KSV=NLEN(3)+NLEN(4)                                         PR1 1217
0090               IF(KSV.EQ.0) GO TO 41                                       PR1 1218
0091               K2=NLEN(3)                                                  PR1 1219
0092               IF(K2.NE.0) READ (NDTA,14) (XIC(I),I=1,K2)                  PR1 1220
0093               K1=K2+1                                                     PR1 1221
0094               K2=NLEN(4)+K2                                               PR1 1222
0095               IF(K2.NE.NLEN(3)) READ (NDTA,14) (XIC(I),I=K1,K2)           PR1 1223
              C    GIT DP NAMES AND VALUES                                     PR1 1224
0096               KDP=XNDP                                                    PR1 1225
0097               IF(KDP.NE.0) READ (NDTA,14) (PAR(I),I=1,KDP)                PR1 1226
0098               IF(KDP.NE.0.AND.NAC.EQ.160) READ (NDTA,15) ((NPAR(I,J),I=1,6),J=1,PR1 1227
                  1KDP)                                                        PR1 1228
              C    GIT TABLES                                                  PR1 1229
0099               IF(KTAB.NE.0) READ (NDTA,14) (XTAB(I),I=1,KTAB)             PR1 1230
              C    COPY MUTUAL NAMES TO NSYM                                   PR1 1231
0100               DO 17 I=1,KMUT                                              PR1 1232
0101               KSYM=KSYM+1                                                 PR1 1233
0102               DO 16 J=1,6                                                 PR1 1234
0103          16   NSYM(J,KSYM)=NMUT(J,I)                                      PR1 1235
0104          17   CONTINUE                                                    PR1 1236
              C    PUSH EQUATIONS-->STACK                                      PR1 1237
0105               CALL GETEQS                                                 PR1 1238
              C    FORM ELEMENT TYPE ARRAY                                     PR1 1239
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0106                DO 19 I=1,KELTS                                          PR1 1240
0107                IF(ELTS(I).NE.C.CDO) GO TO 18                            PR1 1241
0108                CALL CECNAM (NSYM(1,I),0,IXLOC,IVAL)                     PR1 1242
0109                NT(I)=1-LCCTST(IXLOC,IVAL)                               PR1 1243
0110                GO TO 19                                                 PR1 1244
0111             18 NT(I)=1                                                  PR1 1245
0112             19 CONTINUE                                                 PR1 1246
0113                RETURN                                                   PR1 1247
                 C  SOURCE DERIVATIVES EXIST - GIT THEM                      PR1 1248
0114             20 KRAY=22                                                  PR1 1249
0115                KSDR=0                                                   PR1 1250
0116             21 IF(LPULL(NRAY,KRAY,NSDR(1,KSDR+1),0,KSB).EQ.-1) GO TO 7  PR1 1251
0117                KSDR=KSDR+1                                              PR1 1252
0118                GO TO 21                                                 PR1 1253
                 C  TABLES EXIST-GIT THEM TOO                                PR1 1254
0119             22 KRAY=22                                                  PR1 1255
0120                KTAB=0                                                   PR1 1256
0121                KTAB2=0                                                  PR1 1257
0122             23 IF(LPULL(NRAY,KRAY,NTAB(1,KTAB+1),0,NTAB(7,KTAB+1)).EQ.-1) GO TO 7PR1 1258
0123                KTAB=KTAB+NTAB(7,KTAB2)                                  PR1 1259
0124                KTAB2=KTAB2+1                                            PR1 1260
0125                GO TO 23                                                 PR1 1261
                 C  GET MUTUALS                                              PR1 1262
0126             24 KRAY=21                                                  PR1 1263
0127             25 IF(LPULL(NRAY,KRAY,NSYM(1,KMUT+1),0,KSB).EQ.-1) GO TO 7  PR1 1264
0128                KMUT=KMUT+1                                              PR1 1265
0129                GO TO 25                                                 PR1 1266
                 C  GET ELEMENT NAMES AND NUMBERS                            PR1 1267
0130             26 KRAY=21                                                  PR1 1268
0131             27 IF(LPULL(NRAY,KRAY,NSYM(1,KELTS+1),0,KSB).EQ.-1) GO TO 7 PR1 1269
0132                KELTS=KELTS+1                                            PR1 1270
0133                XNELTS=KELTS                                             PR1 1271
0134                GO TO 27                                                 PR1 1272
                 C  GOT NUMBER AND NAMES OF DPS                              PR1 1273
0135             28 KRAY=22                                                  PR1 1274
0136             29 IF(LPULL(NRAY,KRAY,NPAR(1,KDP+1),0,KSB).EQ.-1) GO TO 7   PR1 1275
0137                KDP=KDP+1                                                PR1 1276
0138                NXDP=KDP                                                 PR1 1277
0139                GO TO 29                                                 PR1 1278
                 C  GET NUMBER OF DEFINED PARAMETER DERIVS                   PR1 1279
0140             30 DO 31 I=22,70                                            PR1 1280
0141                IF(LOCATE(NRAY,NNDPD,I,3).EQ.0) XNDPD=XNDPD+1            PR1 1281
0142             31 CONTINUE                                                 PR1 1282
0143                GO TO 6                                                  PR1 1283
                 C  GET NUMBER OF ELEMENT TYPES                              PR1 1284
0144             32 IF(NPAY(18).EQ.NRAY(9)) GO TO 6                          PR1 1285
0145                KRAY=17                                                  PR1 1286
0146             33 IF(LPULL(NRAY,KRAY,NDUM,0,KSB).EQ.-1) GO TO 7            PR1 1287
```

```
0147                DO 34 I=2,10                                                  PR1 1288
0148                IF(NDUM(3).EQ.NUM(I)) GO TO 35                                PR1 1289
0149           34   CONTINUE                                                      PR1 1290
0150                I=2                                                           PR1 1291
0151           35   CTROL(NELS(I-1))=KSB                                          PR1 1292
0152                GO TO 33                                                      PR1 1293
               C    ERRORS                                                        PR1 1294
               C                                                                  PR1 1295
               C    INVALID FLAG ERRORS                                           PR1 1296
0153           36   CALL ERROR (IEP)                                              PR1 1297
               C    EXECUTE SETUP PHASE ONLY NOT SPECIFIED                        PR1 1298
0154           37   CALL ERROR (14)                                               PR1 1299
               C    NO CONTROLS(NO DTASAV)                                        PR1 1300
0155           38   CALL ERROR (17)                                               PR1 1301
               C    NO PROGRAM SIMTR(NO PGMSAV)                                   PR1 1302
0156           39   CALL ERROR (18)                                               PR1 1303
               C    NO TRANSIENT EQUATIONS IN SIMTR                               PR1 1304
0157           40   CALL ERROR (19)                                               PR1 1305
               C    NO STATE VARS                                                 PR1 1306
0158           41   CALL ERROR (20)                                               PR1 1307
0159                END                                                           PR1 1308
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                FUNCTION LGRAB (KPOP)                                          PR1 1309
                C                                                                  PR1 1310
                C   <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR1 1311
                C                                                                  PR1 1312
                C   PULL THE PAST LOCATORS FROM (KSHUV)                           PR1 1313
                C                                                                  PR1 1314
                C   <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR1 1315
                C                                                                  PR1 1316
0002                IMPLICIT REAL*8(A-H,O-Z)                                       PR1 1317
0003                COMMON /SHGR/ NSHUV(500),KSHUV,KST,KEND,KSUB,KEXTD,KPAR        PR1 1318
0004                COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)                       PR1 1319
0005                COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU                           PR1 1320
                C                                                                  PR1 1321
0006                IF(LIST.EQ.1) WRITE (NPRINT,1)                                 PR1 1322
0007              1 FORMAT (9H LGRABBED)                                           PR1 1323
0008                LGRAB=1                                                        PR1 1324
0009                IF(KSHUV.EQ.0) RETURN                                          PR1 1325
0010                LGRAB=0                                                        PR1 1326
0011                KPAR=NSHUV(KSHUV)                                              PR1 1327
0012                KSHUV=KSHUV-1                                                  PR1 1328
0013                KPOP=NSHUV(KSHUV)                                              PR1 1329
0014                KSHUV=KSHUV-1                                                  PR1 1330
0015                KEND=NSHUV(KSHUV)                                              PR1 1331
0016                KSHUV=KSHUV-1                                                  PR1 1332
0017                KST=NSHUV(KSHUV)                                               PR1 1333
0018                KSHUV=KSHUV-1                                                  PR1 1334
0019                KSUB=NSHUV(KSHUV)                                              PR1 1335
0020                KSHUV=KSHUV-1                                                  PR1 1336
0021                RETURN                                                         PR1 1337
0022                END                                                            PR1 1338
```

```
0001            FUNCTION LOCATE (NRAY,NSTR,KSTR,KLEN)                         PR1 1339
          C                                                                   PR1 1340
          C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1341
          C                                                                   PR1 1342
          C     LOCATE STRING (NSTR) IN ARRAY (NRAY) WITH                     PR1 1343
          C     LENGTH (KLEN) STARTING AT (KSTR).                             PR1 1344
          C     LOCATE=0 IF STRING FOUND                                      PR1 1345
          C     LOCATE=1 IF STRING NOT FOUND                                  PR1 1346
          C                                                                   PR1 1347
          C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1348
          C                                                                   PR1 1349
0002            DIMENSION NRAY(80), NSTR(80)                                  PR1 1350
          C                                                                   PR1 1351
0003            LOCATE=1                                                      PR1 1352
0004            DO 1 I=1,KLEN                                                 PR1 1353
0005            KSTRIM=KSTR+I-1                                               PR1 1354
0006            IF(NRAY(KSTRIM).NE.NSTR(I)) RETURN                            PR1 1355
0007        1   CONTINUE                                                      PR1 1356
0008            LOCATE=0                                                      PR1 1357
0009            RETURN                                                        PR1 1358
0010            END                                                           PR1 1359
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001            FUNCTION LOCTST (ILOC,IPOP)
          C
          C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
          C
          C     DETERMINE IF THE LOCATOR (IPOP) IS A VALID SYMBOL
          C     AND FIND ITS POINTER (ILOC) IN (KLOCA) AND SET LOCTST=0
          C     IF INVALID SET LOCTST=1.
          C
          C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
          C
0002            IMPLICIT REAL*8(A-H,O-Z)
0003            COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
               1)
0004            COMMON /PLSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0005            COMMON /SHGR/ NSHUV(500),KSHUV,KST,KEND,KSUB,KEXTD,KPAR
0006            COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0007            COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0008            DIMENSION MPT(8)
0009            DATA MPT/7,4,5,6,1,2,3,8/
          C
          C     FORM POINTERS
0010            KEXTD=0
0011            LOCTST=0
0012            ILOC=0
0013            MPOP=MOD(IPOP,1000000)
0014            MLOC=MOD(IPOP,1000)
          C     NO LOCATION GIVEN
0015            IF(MLOC.EQ.0) GO TO 5
0016            MMID=MPOP-MLOC
          C     CHECK FOR INVERTED ARRAYS
0017            MMIDA=MMID/10000
0018            IF(MMIDA.EQ.16.OR.MMIDA.EQ.26.OR.MMIDA.EQ.36) GO TO 8
0019            DO 4 ILOC=1,KLOC
          C     SEARCH STACK FOR EQUATION POINTERS
0020            JPOP=MOD(KLOCA(1,ILOC),1000000)
0021            JLOC=MOD(JPOP,1000)
          C     ALL BUT SUBSCRIPT MATCH
0022            IF(MPOP.EQ.JPOP) GO TO 6
          C     EXTENDED AND TYPE MATCH ONLY
0023            IF(MMID.NE.(JPOP-JLOC)) GO TO 4
          C     NOT EXTENDED
0024            IF(JPOP.LT.100000) GO TO 1
          C     NO POSSIBLE MATCH - TRY AGAIN
0025            IF(KLOCA(1,ILOC).LT.1000000) GO TO 4
          C     FOUND A MATCH
0026            KEXTD=JPOP/100000
0027            GO TO 7
          C     DETERMINE SUBSCRIPT POSSIBLE MATCH
```

```
0028            1     IF(IPOP.LT.1000000.AND.KLOCA(1,ILOC).LT.1CC0000) GO TO 4   -      PR1 1408
0029                  IF(MLOC.LT.JLOC) GO TO 4                                          PR1 1409
                C     FIND SUBSCRIPT POSSIBLE RANGE                                     PR1 1410
0030                  DO 2 J=1,8                                                        PR1 1411
0031                  JMPT=MPT(J)                                                       PR1 1412
0032                  IF(JLOC.LT.NPT(JMPT)) GO TO 3                                     PR1 1413
0033            2     CONTINUE                                                          PR1 1414
0034                  J=9                                                               PR1 1415
0035            3     J=J-1                                                             PR1 1416
                C     CHECK SUBSCRIPT WITHIN RANGE                                      PR1 1417
0036                  IF(J.LE.0) GO TO 4                                                PR1 1418
0037                  JMPT=MPT(J)                                                       PR1 1419
0038                  IF(NLEN(JMPT).EQ.0) GO TO 3                                       PR1 1420
0039                  IF((MLOC-JLOC).LE.NLEN(JMPT)-1) GO TO 6                           PR1 1421
0040            4     CONTINUE                                                          PR1 1422
                C     NO POSSIBLE MATCH                                                 PR1 1423
0041            5     LOCTST=1                                                          PR1 1424
0042                  GO TO 7                                                           PR1 1425
                C     FORM SUBSCRIPT FROM GIVEN RANGE                                   PR1 1426
0043            6     IF(IPOP.GT.1CCCCCO.OR.KLOCA(1,ILOC).GT.1CC0000.OR.KEXTD.NE.0) KSUBPR1 1427
                     1=MLOC-JLOC+1                                                      PR1 1428
0044            7     IF(IPOP.LT.4000.AND.LOCTST.EQ.0) GO TO 10                         PR1 1429
0045            8     IF(LIST.EQ.1) WRITE (NPRINT,9) LOCTST                             PR1 1430
0046            9     FORMAT (13H LOCTST VALUE,I2)                                      PR1 1431
0047                  RETURN                                                            PR1 1432
                C     INVALID VARIABLE ELEMENT                                          PR1 1433
0048            10    CALL ERROR (34)                                                   PR1 1434
0049                  END                                                               PR1 1435
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001              FUNCTION LPULL (NRAY,KSTR,NSYM,KDO,KSUB)
            C
            C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
            C
            C     PULL NEXT SYMBOL FROM (NRAY) BEGINNING AT (KSTR)
            C     UP TO COLUMN 72 INTO (NSYM) LEFT JUSTIFIED.  READ
            C     NEXT CARD IF END OF LINE FOUND AND CONTINUE
            C     NEXT CARD MUST BE CONTINUATION CARD, ELSE LPULL.NE.0
            C     UPDATE (KSTR) TO NEXT LOCATION.
            C     LPULL=0  GOT SYMBOL
            C     LPULL=1  GOT SYMBOL AND GOT NEXT CARD
            C     LPULL=-1 DID NOT GET SYMBOL BUT GOT NEXT CARD
            C
            C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
            C
0002              IMPLICIT REAL*8(A-H,O-Z)
0003              COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0004              COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0005              COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0006              COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
                 1),NUM(10)
0007              DIMENSION NRAY(80), NSYM(6), NALPHA(36), NXTAB(5)
0008              DATA NALPHA/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1H
                 1N,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H0,1H1,1H2,1H3,
                 21H4,1H5,1H6,1H7,1H8,1H9/
0009              DATA NXTAB/1HT,1HA,1HB,1HL,1HE/
0010              DATA NPER/1H./
0011              EQUIVALENCE (NN,NALPHA(14)), (NX,NALPHA(24))
            C
0012              LPULL=0
            C     FIND BEGINNING (ALPHA) CHAR OF SYMBOL
0013         1    IF(NRAY(KSTR).EQ.NSP) GO TO 3
0014              DO 2 I=1,26
0015              IF(NRAY(KSTR).EQ.NALPHA(I)) GO TO 8
0016         2    CONTINUE
            C     NOT A SYMBOL
0017         3    IF(KDO.EQ.0) GO TO 4
            C     PUSH NON-SYMBOL CHARACTER
0018              IF(NRAY(KSTR).EQ.NSP) GO TO 4
0019              NPUSH(KPUSH)=NRAY(KSTR)
0020              KPUSH=KPUSH+1
            C     SET KDO TO 2 IF SUM OF PRODUCTS TERM
0021              IF(NRAY(KSTR).EQ.NPLUS.OR.NRAY(KSTR).EQ.NMINUS) KDO=2
0022              IF(KPUSH.GT.KPMAX) GO TO 27
0023         4    KSTR=KSTR+1
            C     TRY TO FIND THE SYMBOL AGAIN
0024              IF(KSTR.LE.72) GO TO 1
            C     END OF RECORD-GET ANOTHER
```

```
0025                    NEOF=LREAD(NPGM,0)                                  PR1 1484
0026                 5  FORMAT (1X,80A1)                                   PR1 1485
                    C   CHECK CONTINUATION CHARACTER                        PR1 1486
                        IF(NRAY(6).EQ.NSP) GO TO 6                          PR1 1487
0027                                                                        PR1 1488
0028                    IF(LIST.EQ.1) WRITE (NPRINT,5) NRAY                 PR1 1489
                    C   FIND SYMBOL IN NEW RECORD                           PR1 1490
0029                    KSTR=7                                              PR1 1491
0030                    GO TO 1                                             PR1 1492
                    C   SYMBOL NOT IN THIS RECORD-NO CONTINUATION FOUND     PR1 1493
0031                 6  LPULL=-1                                           PR1 1494
0032                 7  KSTR=7                                             PR1 1495
0033                    RETURN                                              PR1 1496
                    C   IF SYMBOL IS "CCC" IGNORE IT                        PR1 1497
0034                 8  IF(I.EQ.4.AND.NRAY(KSTR-1).EQ.NPER) GO TO 3         PR1 1498
                    C   CLEAR SYMBOL                                        PR1 1499
0035                    DO 9 I=1,6                                          PR1 1500
0036                 9  NSYM(I)=NSP                                        PR1 1501
                    C   GET SYMBOL                                          PR1 1502
0037                    DO 14 I=1,6                                         PR1 1503
                    C   IGNORE SPACES IN SYMBOL                            PR1 1504
0038                10  IF(NRAY(KSTR).NE.NSP) GO TO 11                     PR1 1505
0039                    KSTR=KSTR+1                                         PR1 1506
0040                    IF(KSTR.LE.72) GO TO 10                             PR1 1507
                    C   END OF RECORD - FAKE END OF SYMBOL                  PR1 1508
0041                    KSTR=KSTR-1                                         PR1 1509
0042                    GO TO 15                                            PR1 1510
                    C   CHECK ALPHANUMERIC CHAR. HERE                       PR1 1511
0043                11  DO 12 J=1,36                                       PR1 1512
0044                    IF(NRAY(KSTR).EQ.NALPHA(J)) GO TO 13                PR1 1513
0045                12  CONTINUE                                           PR1 1514
                    C   END OF SYMBOL                                       PR1 1515
0046                    GO TO 14                                            PR1 1516
                    C   GOT AN ALPHANUM CHAR. - SAVE IT                    PR1 1517
0047                13  NSYM(I)=NRAY(KSTR)                                 PR1 1518
0048                    KSTR=KSTR+1                                         PR1 1519
                    C   CHECK FOR END OF RECORD                             PR1 1520
0049                    IF(KSTR.LE.72) GO TO 14                             PR1 1521
                    C   END OF RECORD - GET NEW RECORD                      PR1 1522
0050                    NEOF=LREAD(NPGM,0)                                  PR1 1523
                    C   CHECK FOR CONTINUATION FLAG                        PR1 1524
0051                    IF(NRAY(6).EQ.NSP) GO TO 25                         PR1 1525
0052                    IF(LIST.EQ.1) WRITE (NPRINT,5) NRAY                 PR1 1526
                    C   OK CONTINUE RECORD-FIND SYMBOL                      PR1 1527
0053                    KSTR=7                                              PR1 1528
0054                14  CONTINUE                                           PR1 1529
                    C   CHECK FOR SUBSCRIPT                                 PR1 1530
0055                15  IF(NRAY(KSTR).NE.NLP) RETURN                       PR1 1531
0056                    KSTR=KSTR+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                C     FAKE AN "N" SUBSCRIPT                                    PR1 1532
0057                  KSUB=1000000                                             PR1 1533
0058                  IF(NRAY(KSTR).EQ.NN) GO TO 19                            PR1 1534
                C     IF NOT "N", GET SUBSCRIPT NUMBER                         PR1 1535
0059                  IF(LPUMP2(NRAY,KSTR,KV1).EQ.1) GO TO 22                  PR1 1536
0060                  IF(NRAY(KSTR).EQ.NCOM) GO TO 16                          PR1 1537
0061                  KSUB=KV1                                                 PR1 1538
0062                  GO TO 19                                                 PR1 1539
0063                16 DO 17 II=1,3                                            PR1 1540
0064                   IF(NSYM(3).EQ.NXM(II)) GO TO 18                         PR1 1541
0065                17 CONTINUE                                                PR1 1542
0066                   GO TO 22                                                PR1 1543
0067                18 IF(KXM(II).EQ.0) GO TO 22                               PR1 1544
0068                   KSTR=KSTR+1                                             PR1 1545
0069                   IF(LPUMP2(NRAY,KSTR,KV2).EQ.1) GO TO 22                 PR1 1546
0070                   KSUB=KXM(II)*(KV2-1)+KV1                                PR1 1547
0071                19 KSTR=KSTR+1                                             PR1 1548
0072                   IF(KSUB.NE.1000000) GO TO 21                            PR1 1549
0073                20 IF(NRAY(KSTR-1).EQ.NRP) GO TO 21                        PR1 1550
0074                   KSTR=KSTR+1                                             PR1 1551
0075                   GO TO 20                                                PR1 1552
0076                21 RETURN                                                  PR1 1553
                C     INVALID SUBSCRIPT NUMBER-CHECK EXPRESSION                PR1 1554
0077                22 IF(NSYM(1).NE.NX) GO TO 28                              PR1 1555
0078                   KSUB=0                                                  PR1 1556
                C     CHECK FOR "XTABLE"                                       PR1 1557
0079                   IF(LOCATE(NSYM,NXTAB,2,5).EQ.0) GO TO 26                PR1 1558
                C     LOCATE END OF ARG STRING IN XPRESSION                    PR1 1559
0080                   DO 23 I=KSTR,50                                         PR1 1560
0081                   IF(NRAY(I).EQ.NRP) GO TO 24                             PR1 1561
0082                23 CONTINUE                                                PR1 1562
                C     NO END OF ARG. FOUND.                                    PR1 1563
0083                   GO TO 29                                                PR1 1564
0084                24 KSTR=I+1                                                PR1 1565
0085                   RETURN                                                  PR1 1566
0086                25 LPULL=1                                                 PR1 1567
0087                   GO TO 7                                                 PR1 1568
                C     "XTABLE" FOUND                                           PR1 1569
0088                26 KSTR=KSTR-1                                             PR1 1570
0089                   RETURN                                                  PR1 1571
                C     ERRORS                                                   PR1 1572
                C                                                              PR1 1573
                C     STACK OVERFLOW                                           PR1 1574
0090                27 CALL ERROR (22)                                         PR1 1575
                C     BAD SUBSCRIPT                                            PR1 1576
0091                28 CALL ERROR (23)                                         PR1 1577
                C     BAD EXPRESSION ARGUMENT STRING                           PR1 1578
0092                29 CALL ERROR (24)                                         PR1 1579
```

```
      CC93                    END
```

PR1 1580

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                 FUNCTION LPUMP (NRAY,KRAY,KLEN,KVAL)
              C
              C      <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
              C      PUMP OUT THE INTEGER VALUE (KVAL) FROM BCD ARRAY (NRAY)
              C      STARTING AT (KRAY) FOR LENGTH (KLEN). UPDATE (KRAY) AND
              C      IGNORE BLANKS.  LPUMP=0 IF OKAY CONVERSION
              C                      LPUMP=1 IF ERROR IN CONVERSION
              C
              C      <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
              C
0002                 COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,NO,NPLUS,NMINUS,NSL,NZER,NXM(3
                    1),NNUM(10)
0003                 DIMENSION NRAY(80)
              C
0004                 KVAL=0
0005                 LPUMP=0
0006                 DO 3 I=1,KLEN
0007                 IF(NRAY(KRAY).EQ.NSP) GO TO 3
0008                 DO 1 J=1,10
0009                 IF(NRAY(KRAY).EQ.NNUM(J)) GO TO 2
0010            1    CONTINUE
0011                 LPUMP=1
0012                 RETURN
0013            2    KVAL=KVAL*10+(J-1)
0014            3    KRAY=KRAY+1
0015                 RETURN
0016                 END
```

```
0001                FUNCTION LPUMP2 (NRAY,KRAY,KVAL)                                  PR1 1609
                C                                                                     PR1 1610
                C   <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1611
                C                                                                     PR1 1612
                C   PUMP THE NEXT SUBSCRIPT SURROUNDED NUMBER FROM NRAY              PR1 1613
                C   RETURN VALUE AS LPUMP                                            PR1 1614
                C                                                                     PR1 1615
                C   <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>PR1 1616
                C                                                                     PR1 1617
0002                COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3PR1 1618
                   1),NUM(10)                                                         PR1 1619
0003                DIMENSION NRAY(80)                                                PR1 1620
                C                                                                     PR1 1621
0004                LPUMP2=1                                                          PR1 1622
0005                LRAY=KRAY                                                         PR1 1623
0006              1 LRAY=LRAY+1                                                       PR1 1624
0007                IF(LRAY.GT.72) RETURN                                             PR1 1625
0008                IF(NRAY(LRAY).NE.NRP.AND.NRAY(LRAY).NE.NCOM) GO TO 1              PR1 1626
0009                KXRAY=LRAY-KRAY                                                   PR1 1627
0010                LPUMP2=LPUMP(NRAY,KRAY,KXRAY,KVAL)                                PR1 1628
0011                RETURN                                                            PR1 1629
0012                END                                                               PR1 1630
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001          FUNCTION LREAD (NFIL,LIST)
            C
            C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
            C
            C     READ A RECORD FROM FILE (NFIL) INTO (NRAY)
            C     LIST IF (LIST) IS =1. RETURN CODES:
            C     LREAD=0 NO END OF FILE AND "END" NOT IN COLS 7-9
            C          =1 END OF FILE OR "END" IN COLS 7-9
            C
            C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
            C
0002          COMMON /NASA/ NSYM(6,300),KSYM,NRAY(80),KRAY,NLEN(8),NPT(8),NT(300
             1)
0003          COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0004          DIMENSION NEND(3)
0005          DATA NEND/1HE,1HN,1HD/
            C
            C     SET POINTER AND FLAG
0006          KRAY=7
0007          LREAD=0
            C     READ NEW LINE
0008          READ (NFIL,1,END=3,ERR=3) NRAY
0009        1 FORMAT (80A1)
0010          IF(LIST.EQ.1) WRITE (NPRINT,2) NRAY
0011        2 FORMAT (1X,80A1)
            C     "END" APPEARS?
0012          IF(LOCATE(NRAY,NEND,7,3).EQ.0) LREAD=1
0013          RETURN
0014        3 LREAD=1
0015          RETURN
0016          END
```

```
0001          FUNCTION LSHUV (KPOP,IPOS,KPOPO)
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
        C     PUSH THE CURRENT SYMBOL LOCATORS INTO (KSHUV) IF THE
        C     NEXT SYMBOL IS LOCATED IN THE LIST (KLOCA) OF SYMBOLS
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
0002          IMPLICIT REAL*8 (A-H,O-Z)
0003          COMMON /PUSHED/ KLOCA(3,400),KLOC,NPUSH(1000),KPUSH,KPMAX
0004          COMMON /SHGR/ NSHUV(500),KSHUV,KST,KEND,KSUB,KEXTD,KPAR
0005          COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0006          COMMON /FILES/ NPRINT,NPOM,NCTA,NEQU
        C
0007          LSHUV=1
0008          KMOD=MOD(KPOP,1000)
        C     GO IF EQUATION FOUND
0009          IF(LOCTST(IPOS,KPOP).EQ.0) GO TO 1
        C     NO EQUATION FOR THIS VALUE
        C     USE LAST VALUE
0010          IPOS=KMOD
0011          KPOPO=KPOP
0012          GO TO 2
        C     PUSH SUBSCRIPT, POINTERS, VALUE AND SUM-PRODUCTS FLAG
0013     1    LSHUV=0
0014          IF(KSHUV+5.GT.500) GO TO 5
0015          KSHUV=KSHUV+1
0016          NSHUV(KSHUV)=KSUB
0017          KSHUV=KSHUV+1
0018          NSHUV(KSHUV)=KST
0019          KSHUV=KSHUV+1
0020          NSHUV(KSHUV)=KEND
0021          KSHUV=KSHUV+1
0022          NSHUV(KSHUV)=KPOPO
0023          KSHUV=KSHUV+1
0024          NSHUV(KSHUV)=KPAR
0025          KPOPO=KPOP
0026     2    IF(LIST.EQ.0) RETURN
0027          KS5=KSHUV-4
0028          WRITE (NPRINT,3) IPOS,KPOP,KSUB,KEXTD,LSHUV,KSHUV
0029          IF(KS5.GT.0) WRITE (NPRINT,4) (NSHUV(I),I=KS5,KSHUV)
0030     3    FORMAT (7H LSHUV ,6I9)
0031     4    FORMAT (1X,5I10)
0032          RETURN
        C     ERRORS
        C
        C     SHUV STACK OVERFLOW
```

```
0033          5     CALL ERROR (33)                                    PR1 1710
0034                END                                                PR1 1711
```

```
0001          FUNCTION NUMB (NRAY,KST,KEN,I)
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
        C     DECODE NUMBER AT KST IN NRAY UNITL KEN
        C
        C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
        C
0002          COMMON /FILES/ NPRINT,NPGM,NDTA,NEQU
0003          COMMON /CNTRL/ LIST,L1,CTROL(160),KXM(3)
0004          COMMON /CHARS/ NSP,NLP,NRP,NCOM,NEQ,ND,NPLUS,NMINUS,NSL,NZER,NXM(3
             1),NUMS(10)
0005          DIMENSION NRAY(80)
0006          NUMB=0
0007          DO 4 I=KST,KEN
0008          IF(I.LE.72) GO TO 2
0009          READ (NPGM,1) NRAY
0010    1     FORMAT (80A1)
0011          I=7
0012          KEN=KEN-72+6
0013    2     CONTINUE
0014          IF(NPAY(I).EQ.NSP) GO TO 4
0015          DO 3 J=1,10
0016          IF(NRAY(I).NE.NUMS(J)) GO TO 3
0017          NUMB=NUMB*10+J-1
0018    3     CONTINUE
0019    4     CONTINUE
0020          RETURN
0021          END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                C     PROSE2                                                                PR2 0001
                C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0002
                C                                                                           PR2 0003
                C     PRCSE PART TWO                                                        PR2 0004
                C                                                                           PR2 0005
                C     COPYRIGHT JOHN E. O'REILLY, JR. 1977                                  PR2 0006
                C     10/17/77                                                              PR2 0007
                C     COPYRIGHT JOHN E. O'REILLY, JR. 1978                                  PR2 0008
                C     04/05/78                                                              PR2 0009
                C                                                                           PR2 0010
                C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0011
                C                                                                           PR2 0012
0001                  IMPLICIT REAL*8(A-H,O-Z)                                              PR2 0013
                C     STANDARD SCEPTRE COMMON AREAS                                         PR2 0014
0002                  COMMON /CNTRLS/ TIME,S1(21),XNCPRC,S2,XNHEAD,S3(6),XRUNNO,S4(11),XPR2 0015
                     1WRTSM,S5(46),XNEY,XNE7DT,XNBC,XNBR,XNBL,XNLC,XNLR,XNLL,XNJX,XNJ8DTPR2 0016
                     2,XNJO,XNJ9,S6(2),XNM33,XNM36,XNM66,XNDP,XNDPD,XNELTS,S7(50)        PR2 0017
0003                  COMMON /CURNTS/ C(1)                                                  PR2 0018
0004                  COMMON /DERIVS/ D(1)                                                  PR2 0019
0005                  COMMON /DPNAMS/ DPS(1)                                                PR2 0020
0006                  COMMON /ELNAMS/ LELS(6,300)                                           PR2 0021
0007                  COMMON /HDCRDS/ HEAD(9,11)                                            PR2 0022
0008                  COMMON /NAMES/ XN(1)                                                  PR2 0023
0009                  COMMON /OUTPTS/ OUTR(5,1)                                             PR2 0024
0010                  COMMON /OUTPUT/ NOUTP(6,50)                                           PR2 0025
0011                  COMMON /PRMTRS/ P(1)                                                  PR2 0026
0012                  COMMON /SAVICT/ CICT(1)                                               PR2 0027
0013                  COMMON /SRCDRV/ S(1)                                                  PR2 0028
0014                  COMMON /TABLES/ T(1)                                                  PR2 0029
0015                  COMMON /TAPES/ NPRT,NDTA,NPGM                                         PR2 0030
0016                  COMMON /VOLTGS/ V(1)                                                  PR2 0031
0017                  COMMON /XMUTL/ XM(1)                                                  PR2 0032
                C     PROSE2 COMMON AREA                                                    PR2 0033
0018                  COMMON /PROSE/ A(50,50),B(50,50),L3R,L3S,L4P,L4S,NSV,NEJ,NEL,NIND(PR2 0034
                     13C0),IPERF(50)                                                      PR2 0035
                C                                                                           PR2 0036
0019                  DIMENSION CTROLS(160), ER(50), EI(50), XER(50), XEI(50)               PR2 0037
0020                  DIMENSION NLIN(80), NSRCDR(6), NTABLE(6), NOUTPT(6), NUMS(10)         PR2 0038
0021                  DIMENSION NLT(6), NLT2(6)                                             PR2 0039
0022                  DATA NSRCDR/1HS,1HR,1HC,1HD,1HR,1HV/                                  PR2 0040
0023                  DATA NTABLE/1HT,1HA,1HB,1HL,1HE,1HS/                                  PR2 0041
0024                  DATA NOUTPT/1HO,1HU,1HT,1HP,1HT,1HS/                                  PR2 0042
0025                  DATA NCOM/1H,/,NBL/1H /                                               PR2 0043
0026                  DATA NUMS/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                    PR2 0044
0027                  DATA NI/1HI/,NV/1HV/                                                  PR2 0045
0028                  EQUIVALENCE (TIME,CTROLS(1))                                          PR2 0046
                C                                                                           PR2 0047
                C     BEGIN MAINLINE CODE                                                   PR2 0048
```

```
                C                                                                       PR2 0049
                C     SET I/C DEVICE CODES                                              PR2 0050
0029                  NPRT=6                                                            PR2 0051
0030                  NDTA=1                                                            PR2 0052
0031                  NPGM=2                                                            PR2 0053
0032                  CALL SETUP2 (NAC,LELS,NSRC,NTAB)                                  PR2 0054
                C                                                                       PR2 0055
0033                  IF(XWRTSM.EQ.1.000) WRITE (NPRT,1) CTROLS                         PR2 0056
0034                1 FORMAT (21H <><><>CONTROLS<><><>,/,32(1X,1P5D14.7,/))             PR2 0057
0035                2 FORMAT (1P5D14.7)                                                 PR2 0058
                C     READ HEADING CARDS                                                PR2 0059
0036                  NHEAD=XNHEAD                                                      PR2 0060
0037                  IF(NHEAD.NE.0) READ (NDTA,4) HEAD                                 PR2 0061
0038                  IF(XWRTSM.EQ.1.000.AND.NHEAD.NE.0) WRITE (NPRT,3) ((HEAD(I,J),I=1,PR2 0062
                     19),J=1,NHEAD)                                                     PR2 0063
0039                3 FORMAT (26H <><><>HEADING CARDS<><><>,/,11(1X,9A8,/))             PR2 0064
0040                4 FORMAT (9A8)                                                      PR2 0065
                C     READ ELEMENT VALUES                                               PR2 0066
0041                  NEL=XNELTS                                                        PR2 0067
0042                  READ (NDTA,2) (XN(I),I=1,NEL)                                     PR2 0068
0043                  IF(XWRTSM.EQ.1.000) WRITE (NPRT,5) (XN(I),I=1,NEL)                PR2 0069
0044                5 FORMAT (27H <><><>ELEMENT VALUES<><><>,/,60(1X,1P5D14.7,/))       PR2 0070
                C     READ ELEMENTS NAMES                                               PR2 0071
0045                  IF(NAC.EQ.160) READ (NDTA,7) ((LELS(I,J),I=1,6),J=1,NEL)          PR2 0072
0046                  IF(XWRTSM.EQ.1.000) WRITE (NPRT,6) ((LELS(I,J),I=1,6),J=1,NEL)    PR2 0073
0047                6 FORMAT (26H <><><>ELEMENT NAMES<><><>,/,30(1X,72A1,/))            PR2 0074
0048                7 FORMAT (72A1)                                                     PR2 0075
                C     READ ELEMENTS VOLTAGES AND CURRENTS (INITIAL CONDITIONS)          PR2 0076
0049                  READ (NDTA,2) (V(I),I=1,NEL)                                      PR2 0077
0050                  IF(XWRTSM.EQ.1.000) WRITE (NPRT,8) (V(I),I=1,NEL)                 PR2 0078
0051                8 FORMAT (25H <><><>I/C VOLTAGES<><><>,/,60(1X,1P5D14.7,/))         PR2 0079
0052                  READ (NDTA,2) (C(I),I=1,NEL)                                      PR2 0080
0053                  IF(XWRTSM.EQ.1.000) WRITE (NPRT,9) (C(I),I=1,NEL)                 PR2 0081
0054                9 FORMAT (25H <><><>I/C CURRENTS<><><>,/,60(1X,1P5D14.7,/))       PR2 0082
                C     READ MUTUAL VALUES                                                PR2 0083
0055                  NMUT=XNM33+XNM36+XNM66                                            PR2 0084
0056                  IF(NMUT.NE.0) READ (NDTA,2) (XM(I),I=1,NMUT)                      PR2 0085
0057                  IF(XWRTSM.EQ.1.000.AND.NMUT.NE.0) WRITE (NPRT,10) (XM(I),I=1,NMUT)PR2 0086
0058               10 FORMAT (20H <><><>MUTUALS<><><>,/,10(1X,1P5D14.7,/))             PR2 0087
                C     READ SOURCE DERIVATIVES                                           PR2 0088
0059                  IF(NSRC.NE.0) READ (NDTA,2) (S(I),I=1,NSRC)                       PR2 0089
0060                  IF(XWRTSM.EQ.1.000.AND.NSRC.NE.0) WRITE (NPRT,11) (S(I),I=1,NSRC) PR2 0090
0061               11 FORMAT (25H <><><>SRC. DERIVS.<><><>,/,10(1X,1P5E14.7,/))        PR2 0091
                C     READ STATE VARIABLE SAVES I/C'S                                   PR2 0092
0062                  NIC=XNLC+XNLL+XNBC+XNBL                                           PR2 0093
0063                  IF(NIC.NE.0) READ (NDTA,2) (CICT(I),I=1,NIC)                      PR2 0094
0064                  IF(XWRTSM.EQ.1.000.AND.NIC.NE.0) WRITE (NPRT,12) (CICT(I),I=1,NIC)PR2 0095
0065               12 FORMAT (22H <><><>STATE I/C<><><>,/,10(1X,1P5D14.7,/))          PR2 0096
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                C .   COMPUTE NUMBER OF STATE VARIABLES                                PR2 0097
0066                  NSV=XNBC+XNLL                                                    PR2 0098
                C     READ DEFINED PARAMETER DERIVATIVES                               PR2 0099
0067                  NDPD=XNDPD                                                       PR2 0100
0068                  IF(NDPD.NE.0) READ (NDTA,2) (D(I+NSV-1),I=1,NDPD)                PR2 0101
0069                  IF(XWRTSM.EQ.1.0D0.AND.NDPD.NE.0) WRITE (NPRT,13) (D(I+NSV-1),I=1,PR2 0102
                     1NDPD)                                                            PR2 0103
0070                   13   FORMAT (31H <><><>DEF. PARM. DERIVS.<><><>,/,10(1X,1P5D14.7,/)) PR2 0104
                C     READ STATE VARIABLE I/C'S                                        PR2 0105
0071                  NLL=XNLL                                                         PR2 0106
0072                  IF(NLL.NE.0) READ (NDTA,2) (CICT(I),I=1,NLL)                     PR2 0107
0073                  IF(XWRTSM.EQ.1.0D0.AND.NLL.NE.0) WRITE (NPRT,14) (CICT(I),I=1,NLL)PR2 0108
0074                   14   FORMAT (19H <><><>LL I/C<><><>,/,10(1X,1P5D14.7,/))       PR2 0109
0075                  N1=NLL-1                                                         PR2 0110
0076                  NBC=XNBC                                                         PR2 0111
0077                  IF(NBC.NE.0) READ (NDTA,2) (CICT(I+N1),I=1,NBC)                  PR2 0112
0078                  IF(XWRTSM.EQ.1.0D0.AND.NBC.NE.0) WRITE (NPRT,15) (CICT(I+N1),I=1,NPR2 0113
                     1BC)                                                              PR2 0114
0079                   15   FORMAT (19H <><><>BC I/C<><><>,/,10(1X,1P5D14.7,/))       PR2 0115
                C     READ DEFINED PARAMETERS                                          PR2 0116
0080                  NDP=XNDP                                                         PR2 0117
0081                  IF(NDP.NE.0) READ (NDTA,2) (P(I),I=1,NDP)                        PR2 0118
0082                  IF(XWRTSM.EQ.1.0D0.AND.NDP.NE.0) WRITE (NPRT,16) (P(I),I=1,NDP)  PR2 0119
0083                   16   FORMAT (24H <><><>DEF. PARMS.<><><>,/,20(1X,1P5D14.7,/))   PR2 0120
0084                  IF(NDP.NE.0.AND.NAC.EQ.16C) READ (NDTA,17) (DPS(I),I=1,NDP)      PR2 0121
0085                  IF(XWRTSM.EQ.1.0D0.AND.NDP.NE.0.AND.NAC.EQ.160) WRITE (NPRT,18) (DPR2 0122
                     1PS(I),I=1,NDP)                                                   PR2 0123
0086                   17   FORMAT (12A6)                                              PR2 0124
0087                   18   FORMAT (29H <><><>DEF. PARM. NAMES<><><>,/,9(1X,12A6,/))   PR2 0125
                C     READ TABLES                                                      PR2 0126
0088                  IF(NTAB.NE.0) READ (NDTA,2) (T(I),I=1,NTAB)                      PR2 0127
0089                  IF(XWRTSM.EQ.1.0D0.AND.NTAB.NE.0) WRITE (NPRT,19) (T(I),I=1,NTAB) PR2 0128
0090                   19   FORMAT (19H <><><>TABLES<><><>,/,50(1X,1P5D14.7,/))       PR2 0129
                C     READ OUTPUT REQUESTS                                             PR2 0130
0091                  NOPRQ=XNOPRQ                                                     PR2 0131
0092                  READ (NDTA,17) ((OUTR(I,J),I=1,5),J=1,NOPRQ)                     PR2 0132
0093                  IF(XWRTSM.EQ.1.0D0) WRITE (NPRT,20) ((OUTR(I,J),I=1,5),J=1,NOPRQ) PR2 0133
0094                   20   FORMAT (28H <><><>OUTPUT REQUESTS<><><>,/,10(1X,12A6,/))   PR2 0134
0095                  NOPRQ=NOPRQ+1                                                    PR2 0135
0096                  READ (NDTA,7) ((NOUTP(I,J),I=1,6),J=1,NOPRQ)                     PR2 0136
0097                  IF(XWRTSM.EQ.1.0D0) WRITE (NPRT,21) ((NOUTP(I,J),I=1,6),J=1,NOPRQ)PR2 0137
0098                   21   FORMAT (20H <><><>OUTPUTS<><><>,/,10(1X,72A1,/))          PR2 0138
                C     ZERO OUT DERIVATIVES                                             PR2 0139
0099                  DO 22 I=1,NSV                                                    PR2 0140
0100                   22   D(I)=0.0D0                                                 PR2 0141
                C     CLEAR INDEPENDENT VARIABLE TABLE                                 PR2 0142
0101                  DO 23 I=1,NEL                                                    PR2 0143
0102                   23   NIND(I)=0                                                  PR2 0144
```

```
C     FORM INDEPENEDENT VARIABLE TABLE
      NEJ=0
      DO 26 I=2,NOPRQ
      DO 25 J=1,NEL
      DO 24 K=1,6
      IF(NOUTP(K,I).NE.LELS(K,J)) GO TO 25
   24 CONTINUE
      NIND(J)=1
      NEJ=NEJ+1
      XN(J)=0.0D0
      GO TO 26
   25 CONTINUE
   26 CONTINUE
C     COMPUTE OFFSETS TO STATE VARIABLE NAMES
      L4S=XNE7DT+1
      L4P=XNBC+L4S-1
      L3S=XNE7DT+XNBC+XNBR+XNBL+XNLC+XNLR+1
      L3P=XNLL+L3S-1
C     CLEAR STATE VARIABLES
      IF(L3S.GT.L3P) GO TO 28
      DO 27 I=L3S,L3P
   27 C(I)=0.0
   28 IF(L4S.GT.L4P) GO TO 30
      DO 29 I=L4S,L4P
   29 V(I)=0.0D0
C     PERFORM EIGAN ANALYSIS
   30 CALL EIGAN (ER,EI)
C     PRINT HEADER
      WRITE (NPRT,31)
   31 FORMAT (27H1NASA-LEWIS RESEARCH CENTER,/,27H ---- ----- -------- -
     1-----,//,27H       PROSE PROGRAM          ,/,27H      ----- -------
     2      ,/,28H UNIVERSITY OF SOUTH FLORIDA,/,28H ---------- -- -----
     3 -------,/)
      IF(NHEAD.NE.0) WRITE (NPRT,32) ((HEAD(I,J),I=1,9),J=1,NHEAD)
   32 FORMAT (11(1X,9A8,/))
      WRITE (NPRT,33)
   33 FORMAT (21H ----- -------- -----,/,21H STATE VARIABLE NAMES,/,21H
     1----- -------- -----)
C     WRITE I3 STATE VARIABLE NAMES
      IF(L3S.GT.L3P) GO TO 37
      DO 35 I=L3S,L3P
      DO 34 II=1,5
   34 NUT2(II+1)=LELS(II,I)
      NUT2(1)=NI
      CALL BACK (NUT2,NUT)
   35 WRITE (NPRT,36) NUT
   36 FORMAT (1X,6A1)
C     WRITE C4 STATE VARIABLE NAMES
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0140              37    IF(L4S.GT.L4P) GO TO 41                                          PR2 0193
0141                    DO 39 I=L4S,L4P                                                  PR2 0194
0142                    DO 38 II=1,5                                                     PR2 0195
0143              38    NUT2(II+1)=LELS(II,I)                                            PR2 0196
0144                    NUT2(1)=NV                                                       PR2 0197
0145                    CALL BACK (NUT2,NUT)                                             PR2 0198
0146              39    WRITE (NPRT,40) NUT                                              PR2 0199
0147              40    FORMAT (1X,6A1)                                                  PR2 02CC
                  C     WRITE A MATRIX HEADER                                            PR2 0201
0149              41    WRITE (NPRT,42)                                                  PR2 0202
0149              42    FORMAT (/,9H ------ -,/,9H MATRIX A,/,9H ------ -)               PR2 0203
                  C     WRITE A MATRIX                                                   PR2 0204
0150                    DO 43 I=1,NSV                                                    PR2 0205
0151              43    WRITE (NPRT,44) (A(I,J),J=1,NSV)                                 PR2 0206
0152              44    FORMAT (1X,1P7D14.5)                                             PR2 0207
                  C     WRITE INDEPENDENT SOURCE HEADER                                  PR2 0208
0153                    IF(NEJ.EQ.0) GO TO 50                                            PR2 0209
0154                    WRITE (NPRT,45)                                                  PR2 021C
0155              45    FORMAT (/,2CH ---------- -------,/,20H INDEPENDENT SOURCES,/,20H PR2 0211
                       1---------- -------)                                            PR2 0212
0156                    DO 47 I=1,NEL                                                    PR2 0213
0157                    IF(NIND(I).NE.1) GO TO 47                                        PR2 0214
0158                    CALL BACK (LELS(1,I),NUT)                                        PR2 0215
0159                    WRITE (NPRT,46) NUT                                              PR2 0216
0160              46    FORMAT (1X,6A1)                                                  PR2 0217
0161              47    CONTINUE                                                         PR2 0218
                  C     WRITE B MATRIX HEADER                                            PR2 0219
0162                    WRITE (NPRT,48)                                                  PR2 022C
0163              48    FORMAT (/,9H ------ -,/,9H MATRIX B,/,9H ------ -)               PR2 0221
                  C     WRITE B MATRIX                                                   PR2 0222
0164                    DO 49 I=1,NSV                                                    PR2 0223
0165              49    WRITE (NPRT,44) (B(I,J),J=1,NEJ)                                 PR2 0224
                  C     WRITE EIGENVALUES                                                PR2 0225
0166              50    WRITE (NPRT,51)                                                  PR2 0226
0167              51    FORMAT (/,24H ---------- -- ------ -,/,24H EIGENVALUES OF MATRIX PR2 0227
                       1A,/,24H ---------- -- ------ -,/,14X,9HREAL PART,10X,10HIMAG. PARPR2 0228
                       2T,/,14X,9H---- ----,10X,10H----- ----).                        PR2 0229
                  C     FORM LARGEST EIGENVALUE AND PRINT THEM                           PR2 023C
0168                    XL=0.CDC                                                         PR2 0231
0169                    DO 53 I=1,NSV                                                    PR2 0232
0170                    WRITE (NPRT,52) ER(I),EI(I)                                      PR2 0233
0171              52    FORMAT (12X,1PE14.7,5X,1PE14.7)                                  PR2 0234
0172                    XV=DSQRT(ER(I)*ER(I)+EI(I)*EI(I))                                PR2 0235
0173                    IF(DABS(XL).GT.DABS(XV)) GO TO 53                                PR2 0236
0174                    XL=XV                                                            PR2 0237
0175                    IL=I                                                             PR2 0238
0176              53    CONTINUE                                                         PR2 0239
                  C     DC SENSITIVITY ANALYSIS                                          PR2 024C
```

```
0177                WRITE (NPRT,54) IL,XL                                          PR2 0241
0178          54    FORMAT (/,21H ----------- --------,/,21H SENSITIVITY ANALYSIS,/,21PR2 0242
                   1H ----------- --------,/,21H MAXIMUM EIGENVALUE #,I3,13H MAGNITUDEPR2 0243
                   2 = ,1PE14.7,//,21H ELEMENT  SENSITIVITY,/,21H -------  -----------PR2 0244
                   3)                                                              PR2 0245
0179                EGL=0.0D0                                                      PR2 0246
0180                IX=0                                                           PR2 0247
0181                DO 61 I=1,NEL                                                  PR2 0248
              C     IGNORE IF INDEP. SOURCE                                        PR2 0249
0182                IF(NIND(I).EQ.1) GO TO 61                                      PR2 0250
              C     IGNORE IF ZERO VALUED                                          PR2 0251
0183                IF(XN(I).EQ.0.0D0) GO TO 61                                    PR2 0252
              C     SAVE OLD VALUE                                                 PR2 0253
0184                SV=XN(I)                                                       PR2 0254
              C     BUMP VALUE BY 10 PER CENT                                      PR2 0255
0185                XN(I)=XN(I)*1.01D0                                             PR2 0256
0186                CALL EIGAN (XER,XEI)                                           PR2 0257
              C     COMPUTE SENSITIVITIES                                          PR2 0258
0187                EGU=DSQRT(XER(IL)*XER(IL)+XEI(IL)*XEI(IL))-XL                  PR2 0259
0188                EG=EGL*100.0D0/XL                                              PR2 0260
0189                IF(EG.GE.-1.03D0.AND.EG.LE.-0.97D0) GO TO 56                   PR2 0261
0190                IF(EG.GE.0.97D0.AND.EG.LE.1.03D0) GO TO 58                     PR2 0262
0191                CALL BACK (LELS(1,I),NUT)                                      PR2 0263
0192                WRITE (NPRT,55) NUT,EG                                         PR2 0264
0193          55    FORMAT (2X,6A1,5X,F6.3)                                        PR2 0265
0194                IF(DABS(EG).LE.DABS(EGL)) GO TO 60                             PR2 0266
0195                EGL=EG                                                         PR2 0267
0196                NL=I                                                           PR2 0268
0197                GO TO 60                                                       PR2 0269
              C     GOT SENS APPROX. -1                                            PR2 0270
0198          56    CALL BACK (LELS(1,I),NUT)                                      PR2 0271
0199                WRITE (NPRT,57) NUT,EG                                         PR2 0272
0200          57    FORMAT (2X,6A1,5X,F6.3,5X,46H*** ELEMENT MAY BE REMOVED BY SETTINGPR2 0273
                   1 TO ZERO.)                                                     PR2 0274
0201                NIND(I)=-2                                                     PR2 0275
0202                IX=1                                                           PR2 0276
0203                GO TO 60                                                       PR2 0277
              C     GOT SENS. APPROX 1                                             PR2 0278
0204          58    CALL BACK (LELS(1,I),NUT)                                      PR2 0279
0205                WRITE (NPRT,59) NUT,EG                                         PR2 0280
0206          59    FORMAT (2X,6A1,5X,F6.3,5X,50H*** ELEMENT MAY BE REMOVED BY SETTINGPR2 0281
                   1 TO INFINITY.)                                                 PR2 0282
0207                NIND(I)=2                                                      PR2 0283
0208                IX=1                                                           PR2 0284
              C     RESTORE VALUE                                                  PR2 0285
0209          60    XN(I)=SV                                                       PR2 0286
0210          61    CONTINUE                                                       PR2 0287
0211                IF(IX.EQ.1) STOP                                               PR2 0288
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0212              CALL BACK (LELS(1,NL),NUT)                                    PR2 0288
0213              WRITE (NPRT,62) NUT                                           PR2 0290
0214           62 FORMAT (/,35H NO SENSITIVITIES WITHIN 3% OF  I .,/,1X,6A1,40H MAY PR2 0291
                  18E REMOVED (I.E. SET TO ZERO     IF,19H SENSITIVITY < ZERO,/,29X,3PR2 0292
                  29HSET TO INFINITY IF SENSITIVITY > ZERO.))                   PR2 0293
0215               STOP                                                         PR2 0294
0216               END                                                          PR2 0295
```

```
0001          SUBROUTINE ATEIG (M,A,RR,RI,IANA,IA)                              PR2 0296
              C                                                                 PR2 0297
              C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0298
              C                                                                 PR2 0299
              C        COMPUTE EIGENVALUES OF UPPER HESSENBERG MATRIX A        PR2 0300
              C        M=ORDER OF MATRIX A                                     PR2 0301
              C        RR=REAL EIGENVALUE RESULTS ARRAY                        PR2 0302
              C        RI=IMAGINARY EIGENVALUE ARRAY                           PR2 0303
              C        IANA=COMPUTE METHID ARRAY(0 OR 2=DIRECT)                PR2 0304
              C        IA=FIRST DIMENSION OF A MATRIX                          PR2 0305
              C                                                                PR2 0306
              C        IBM SSP ROUTINE                                         PR2 0307
              C                                                                PR2 0308
              C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0309
              C                                                                PR2 0310
0002          IMPLICIT REAL*8(A-H,O-Z)                                         PR2 0311
0003          DIMENSION A(1), RR(1), RI(1), PRR(2), PRI(2), IANA(1)            PR2 0312
0004          INTEGER*4 P,PI,Q                                                 PR2 0313
              C                                                                PR2 0314
0005          E7=1.0E-8                                                        PR2 0315
0006          E6=1.0E-6                                                        PR2 0316
0007          E10=1.0E-10                                                      PR2 0317
0008          DELTA=0.5                                                        PR2 0318
0009          MAXIT=30                                                         PR2 0319
              C                                                                PR2 0320
              C        INITIALIZATION                                          PR2 0321
              C                                                                PR2 0322
0010          N=M                                                              PR2 0323
0011        1 N1=N-1                                                           PR2 0324
0012          IN=N1*IA                                                         PR2 0325
0013          NN=IN+N                                                          PR2 0326
0014          IF(N1) 2,69,2                                                    PR2 0327
0015        2 NP=N+1                                                           PR2 0328
              C                                                                PR2 0329
              C        ITERATION COUNTER                                       PR2 0330
              C                                                                PR2 0331
0016          IT=0                                                             PR2 0332
              C                                                                PR2 0333
              C        ROOTS OF THE 2ND ORDER MAIN SUBMATRIX AT THE PREVIOUS   PR2 0334
              C        ITERATION                                               PR2 0335
              C                                                                PR2 0336
0017          DO 3 I=1,2                                                       PR2 0337
0018          PRR(I)=0.0                                                       PR2 0338
0019        3 PRI(I)=0.0                                                       PR2 0339
              C                                                                PR2 0340
              C        LAST TWO SUBDIAGONAL ELEMENTS AT THE PREVIOUS ITERATION PR2 0341
              C                                                                PR2 0342
0020          PAN=0.0                                                          PR2 0343
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0021                PAN1=C.0
                C
                C       ORIGIN SHIFT
                C
0022                R=C.0
0023                S=0.0
                C
                C       ROOTS CF THE LOWER MAIN 2 BY 2 SUBMATRIX
                C
0024                N2=N1-1
0025                IN1=IN-IA
0026                NN1=IN1+N
0027                N1N=IN+N1
0028                N1N1=IN1+N1
0029          4     T=A(N1N1)-A(NN)
0030                U=T*T
0031                V=4.0*A(N1N)*A(NN1)
0032                IF(DABS(V)-U*E7) 11,11,5
0033          5     T=U+V
0034                IF(DABS(T)-DMAX1(U,DABS(V))*E6) 6,6,7
0035          6     T=C.0
0036          7     U=(A(N1N1)+A(NN))/2.0
0037                V=DSQRT(DABS(T))/2.0
0038                IF(T) 15,8,8
0039          8     IF(U) 10,9,9
0040          9     RR(N1)=U+V
0041                RR(N)=U-V
0042                GO TO 14
0043         10     RR(N1)=U-V
0044                RR(N)=U+V
0045                GC TO 14
0046         11     IF(T) 13,12,12
0047         12     RR(N1)=A(N1N1)
0048                RR(N)=A(NN)
0049                GO TO 14
0050         13     RR(N1)=A(NN)
0051                RR(N)=A(N1N1)
0052         14     RI(N)=C.0
0053                RI(N1)=C.C
0054                GO TO 16
0055         15     RR(N1)=U
0056                RR(N)=U
0057                RI(N1)=V
0058                RI(N)=-V
0059         16     IF(N2) 68,68,17
                C
                C       TESTS OF CONVERGENCE
                C
```

```
0060          17   N1N2=N1N1-IA                                                    PR2 0392
0061               RMCD=RR(N1)*RR(N1)+RI(N1)*RI(N1)                                PR2 0393
0062               EPS=E10*DSQRT(RMCD)                                             PR2 0394
0063               IF(DABS(A(N1N2))-EPS) 68,68,18                                  PR2 0395
0064          18   IF(DABS(A(NN1))-E10*DABS(A(NN))) 69,69,19                       PR2 0396
0065          19   IF(DABS(PAN1-A(N1N2))-DABS(A(N1N2))*E6) 67,67,20                PR2 0397
0066          20   IF(DABS(PAN-A(NN1))-DABS(A(NN1))*E6) 67,67,21                   PR2 0398
0067          21   IF(IT-MAXIT) 22,67,67                                           PR2 0399
                C                                                                  PR2 0400
                C      COMPUTE THE SHIFT                                           PR2 0401
                C                                                                  PR2 0402
0068          22   J=1                                                             PR2 0403
0069               DO 24 I=1,2                                                     PR2 0404
0070               K=NP-I                                                          PR2 0405
0071               IF(DABS(RR(K)-PRR(I))+DABS(RI(K)-PRI(I))-DELTA*(DABS(RR(K))+DABS(RPR2 0406
                  1I(K))) 23,24,24                                                 PR2 0407
0072          23   J=J+I                                                           PR2 0408
0073          24   CONTINUE                                                        PR2 0409
0074               GO TO (25,26,26,27), J                                          PR2 0410
0075          25   R=0.0                                                           PR2 0411
0076               S=0.0                                                           PR2 0412
0077               GO TO 28                                                        PR2 0413
0078          26   J=N+2-J                                                         PR2 0414
0079               R=RR(J)*RR(J)                                                   PR2 0415
0080               S=RR(J)+RR(J)                                                   PR2 0416
0081               GO TO 28                                                        PR2 0417
0082          27   R=RR(N)*RR(N1)-RI(N)*RI(N1)                                     PR2 0418
0083               S=RR(N)+RR(N1)                                                  PR2 0419
                C                                                                  PR2 0420
                C      SAVE THE LAST TWO SUBDIAGONAL TERMS AND THE ROOTS OF THE    PR2 0421
                C      SUBMATRIX BEFORE ITERATION                                  PR2 0422
                C                                                                  PR2 0423
0084          28   PAN=A(NN1)                                                      PR2 0424
0085               PAN1=A(N1N2)                                                    PR2 0425
0086               DO 29 I=1,2                                                     PR2 0426
0087               K=NP-I                                                          PR2 0427
0088               PRR(I)=RR(K)                                                    PR2 0428
0089          29   PRI(I)=RI(K)                                                    PR2 0429
                C                                                                  PR2 0430
                C      SEARCH FOR A PARTITION OF THE MATRIX, DEFINED BY P AND Q    PR2 0431
                C                                                                  PR2 0432
0090               P=N2                                                            PR2 0433
0091               IF(N-3) 35,35,30                                                PR2 0434
0092          30   IPI=N1N2                                                        PR2 0435
0093               DO 34 J=2,N2                                                    PR2 0436
0094               IPI=IPI-IA-1                                                    PR2 0437
0095               IF(DABS(A(IPI))-EPS) 35,35,31                                   PR2 0438
0096          31   IPIP=IPI+IA                                                     PR2 0439
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0097              IPIP2=IPIP+IA                                                          PR2 0440
0098              D=A(IPIP)*(A(IPIP)-S)+A(IPIP2)*A(IPIP+1)+R                             PR2 0441
0099              IF(D) 32,33,32                                                         PR2 0442
0100          32  IF(DABS(A(IPI)*A(IPIP+1))*(DABS(A(IPIP)+A(IPIP2+1)-S)+DABS(A(IPIP2PR2 0443
                 1+2)))-DABS(D)*EPS) 36,36,33                                          PR2 0444
0101          33  P=N1-J                                                             PR2 0445
0102          34  CONTINUE                                                           PR2 0446
0103          35  Q=P                                                                PR2 0447
0104              GO TO 39                                                           PR2 0448
0105          36  P1=P-1                                                             PR2 0449
0106              Q=P1                                                               PR2 0450
0107              IF(P1-1) 39,39,37                                                  PR2 0451
0108          37  DO 38 I=2,P1                                                       PR2 0452
0109              IPI=IPI-IA-1                                                       PR2 0453
0110              IF(DABS(A(IPI))-EPS) 39,39,38                                      PR2 0454
0111          38  Q=Q-1                                                             PR2 0455
                C                                                                    PR2 0456
                C        QR DOUBLE ITERATION                                         PR2 0457
                C                                                                    PR2 0458
0112          39  II=(P-1)*IA+P                                                      PR2 0459
0113              DO 66 I=P,N1                                                       PR2 0460
0114              III=II-IA                                                          PR2 0461
0115              IIP=II+IA                                                          PR2 0462
0116              IF(I-P) 41,40,41                                                   PR2 0463
0117          40  IPI=II+1                                                          PR2 0464
0118              IPIP=IIP+1                                                         PR2 0465
                C                                                                    PR2 0466
                C        INITIALIZATION OF THE TRANSFORMATION                        PR2 0467
                C                                                                    PR2 0468
0119              G1=A(II)*(A(II)-S)+A(IIP)*A(IPI)+R                                 PR2 0469
0120              G2=A(IPI)*(A(IPIP)+A(II)-S)                                        PR2 0470
0121              G3=A(IPI)*A(IPIP+1)                                                PR2 0471
0122              A(IPI+1)=0.0                                                       PR2 0472
0123              GO TO 44                                                           PR2 0473
0124          41  G1=A(III)                                                          PR2 0474
0125              G2=A(III+1)                                                        PR2 0475
0126              IF(I-N2) 42,42,43                                                  PR2 0476
0127          42  G3=A(III+2)                                                       PR2 0477
0128              GO TO 44                                                           PR2 0478
0129          43  G3=0.0                                                            PR2 0479
0130          44  CAP=DSQRT(G1*G1+G2*G2+G3*G3)                                       PR2 0480
0131              IF(CAP) 45,48,45                                                   PR2 0481
0132          45  IF(G1) 46,47,47                                                   PR2 0482
0133          46  CAP=-CAP                                                          PR2 0483
0134          47  T=G1+CAP                                                          PR2 0484
0135              PSI1=G2/T                                                          PR2 0485
0136              PSI2=G3/T                                                          PR2 0486
0137              ALPHA=2.0/(1.0+PSI1*PSI1+PSI2*PSI2)                                PR2 0487
```

```
0138                GO TO 49                                          PR2 0488
0139          48    ALPHA=2.0                                         PR2 0489
0140                PSI1=0.0                                          PR2 0490
0141                PSI2=0.0                                          PR2 0491
0142          49    IF(I-Q) 50,53,50                                  PR2 0492
0143          50    IF(I-P) 51,52,51                                  PR2 0493
0144          51    A(II1)=-CAP                                       PR2 0494
0145                GO TO 53                                          PR2 0495
0146          52    A(II1)=-A(II1)                                    PR2 0496
              C                                                       PR2 0497
              C        ROW OPERATION                                  PR2 0498
              C                                                       PR2 0499
0147          53    IJ=II                                             PR2 0500
0148                DO 57 J=I,N                                       PR2 0501
0149                T=PSI1*A(IJ+1)                                    PR2 0502
0150                IF(I-N1) 54,55,55                                 PR2 0503
0151          54    IP2J=IJ+2                                         PR2 0504
0152                T=T+PSI2*A(IP2J)                                  PR2 0505
0153          55    ETA=ALPHA*(T+A(IJ))                               PR2 0506
0154                A(IJ)=A(IJ)-ETA                                   PR2 0507
0155                A(IJ+1)=A(IJ+1)-PSI1*ETA                          PR2 0508
0156                IF(I-N1) 56,57,57                                 PR2 0509
0157          56    A(IP2J)=A(IP2J)-PSI2*ETA                          PR2 0510
0158          57    IJ=IJ+IA                                          PR2 0511
              C                                                       PR2 0512
              C        COLUMN OPERATION                               PR2 0513
              C                                                       PR2 0514
0159                IF(I-N1) 59,58,58                                 PR2 0515
0160          58    K=N                                               PR2 0516
0161                GO TO 60                                          PR2 0517
0162          59    K=I+2                                             PR2 0518
0163          60    IP=IIP-I                                          PR2 0519
0164                DO 64 J=Q,K                                       PR2 0520
0165                JIP=IP+J                                          PR2 0521
0166                JI=JIP-IA                                         PR2 0522
0167                T=PSI1*A(JIP)                                     PR2 0523
0168                IF(I-N1) 61,62,62                                 PR2 0524
0169          61    JIP2=JIP+IA                                       PR2 0525
0170                T=T+PSI2*A(JIP2)                                  PR2 0526
0171          62    ETA=ALPHA*(T+A(JI))                               PR2 0527
0172                A(JI)=A(JI)-ETA                                   PR2 0528
0173                A(JIP)=A(JIP)-ETA*PSI1                            PR2 0529
0174                IF(I-N1) 63,64,64                                 PR2 0530
0175          63    A(JIP2)=A(JIP2)-ETA*PSI2                          PR2 0531
0176          64    CONTINUE                                          PR2 0532
0177                IF(I-N2) 65,66,66                                 PR2 0533
0178          65    JI=II+3                                           PR2 0534
0179                JIP=JI+IA                                         PR2 0535
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0180                JIP2=JIP+IA                                          PR2 0536
0181                ETA=ALPHA*PSI2*A(JIP2)                               PR2 0537
0182                A(JI)=-ETA                                           PR2 0538
0183                A(JIP)=-ETA*PSI1                                     PR2 0539
0184                A(JIP2)=A(JIP2)-ETA*PSI2                             PR2 0540
0185                 66 II=IIP+1                                         PR2 0541
0186                IT=IT+1                                              PR2 0542
0187                GO TO 4                                              PR2 0543
                    C                                                    PR2 0544
                    C     END OF ITERATION                               PR2 0545
                    C                                                    PR2 0546
0188                 67 IF(DABS(A(NN1))-DABS(A(N1N2))) 69,68,68          PR2 0547
                    C                                                    PR2 0548
                    C     TWO EIGENVALUES HAVE BEEN FOUND                PR2 0549
                    C                                                    PR2 0550
0189                 68 IANA(N)=0                                        PR2 0551
0190                IANA(N1)=2                                           PR2 0552
0191                N=N2                                                 PR2 0553
0192                IF(N2) 71,71,1                                       PR2 0554
                    C                                                    PR2 0555
                    C     ONE EIGENVALUE HAS BEEN FOUND                  PR2 0556
                    C                                                    PR2 0557
0193                 69 RR(N)=A(NN)                                      PR2 0558
0194                RI(N)=0.0                                            PR2 0559
0195                IANA(N)=1                                            PR2 0560
0196                IF(N1) 71,71,70                                      PR2 0561
0197                 70 N=N1                                             PR2 0562
0198                GO TO 1                                              PR2 0563
0199                 71 RETURN                                           PR2 0564
0200                END                                                  PR2 0565
```

```
      SUBROUTINE BACK (NAME,NCHAR)
C
C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
C
C     CHANGE NAME TO NCHAR IN MECHANICAL FORMAT .
C     CHANGES
C     GP=TP       DXXX=ZXXX AND CHANGE XXX
C     I=F EXCEPT ICW=TJ
C     V=V EXCEPT VCW=WJ
C     ELEMENTS
C     CU=M    EU=U    JU=R    LU=K    RU=D
C     CW=J    EW=G    JW=C
C
C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
C
      DIMENSION NEL(5), NELF(8), NST(2), NSTF(4), NCHAR(6), NAME(6)
      EQUIVALENCE (NG,NELF(7)), (ND,NELF(5)), (NT,NSTF(3))
      EQUIVALENCE (NU,NELF(2)), (NC,NELF(8)), (NW,NSTF(4))
      DATA NEL/1HC,1HE,1HJ,1HL,1HR/
      DATA NELF/1HM,1HU,1HR,1HK,1HD,1HJ,1HG,1HC/
      DATA NST/1HI,1HV/
      DATA NSTF/1HF,1HV,1HT,1HW/
      DATA NZ/1HZ/,NP/1HP/,NBL/1H /
      KST=1
      KN=0
      DO 1 I=1,6
      IF(NAME(I).NE.NBL) KN=KN+1
    1 NCHAR(I)=NAME(I)
      IF(KN.LE.1) RETURN
      IF(NCHAR(1).NE.NG.OR.NCHAR(2).NE.NP) GO TO 3
      NCHAR(1)=NT
    2 RETURN
    3 DO 4 KINS=1,4
      KLOC=KN-KINS+1
      IF(KLOC.LE.1) RETURN
      IF(NCHAR(KLOC).EQ.NU.OR.NCHAR(KLOC).EQ.NW) GO TO 5
    4 CONTINUE
      RETURN
    5 IF(NCHAR(1).NE.NC) GO TO 6
      NCHAR(1)=NZ
      KST=2
    6 DO 7 K1=1,2
      IF(NCHAR(KST).NE.NST(K1)) GO TO 7
      NCHAR(KST)=NSTF(K1)
      KST=KST+1
      IF(NCHAR(KST).EQ.NC.AND.NCHAR(KLOC).EQ.NW) NCHAR(KST-1)=NSTF(K1+2)
      GO TO 8
    7 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0035          8    KEL=5                                                    PR2 0614
0036               KELF=0                                                   PR2 0615
0037               IF(NCHAR(KLOC).EQ.NU) GO TO 9                            PR2 0616
0038               KEL=3                                                    PR2 0617
0039               KELF=5                                                   PR2 0618
0040          9    DO 10 K1=1,KEL                                           PR2 0619
0041               IF(NCHAR(KST).NE.NEL(K1)) GO TO 10                       PR2 0620
0042               NCHAR(KST)=NELF(K1+KELF)                                 PR2 0621
0043               GO TO 11                                                 PR2 0622
0044          10   CONTINUE                                                 PR2 0623
0045               RETURN                                                   PR2 0624
0046          11   IF(KLOC.EQ.KN) GO TO 13                                  PR2 0625
0047               KELF=KN-KLOC                                             PR2 0626
0048               DO 12 K1=1,KELF                                          PR2 0627
0049.         12   NCHAR(KLOC+K1-1)=NCHAR(KLOC+K1)                          PR2 0628
0050          13   NCHAR(KN)=NBL                                            PR2 0629
0051               GO TO 2                                                  PR2 0630
0052               END                                                      PR2 0631
```

```
0001          SUBROUTINE EIGAN (ER,EI)                                          PR2 0632
        C                                                                       PR2 0633
        C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 0634
        C                                                                       PR2 0635
        C     SUPERVISE COMPUTATION OF EIGENVALUES                              PR2 0636
        C     FROM SCEPTRE ROUTINE SIMTR                                        PR2 0637
        C                                                                       PR2 0638
        C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 0639
        C                                                                       PR2 0640
0002          IMPLICIT REAL*8(A-H,O-Z)                                          PR2 0641
0003          COMMON /CNTRLS/ TIME,S1(21),XNCPRQ,S2,XNHEAD,S3(6),XRUNNO,S4(11),XPR2 0642
             1WRTSM,S5(46),XNEY,XNE7DT,XNBC,XNBR,XNBL,XNLC,XNLR,XNLL,XNJX,XNJ8DTPR2 0643
             2,XNJ0,XNJ9,S6(2),XNM33,XNM36,XNM66,XNDP,XNDPD,XNELTS,S7(50)       PR2 0644
0004          COMMON /CURNTS/ C(1)                                              PR2 0645
0005          COMMON /DERIVS/ D(1)                                              PR2 0646
0006          COMMON /NAMES/ XN(1)                                              PR2 0647
0007          COMMON /OUTPUT/ OUTP(1)                                           PR2 0648
0008          COMMON /TAPES/ NPRT,NDTA,NPGM                                     PR2 0649
0009          COMMON /VOLTGS/ V(1)                                              PR2 0650
0010          COMMON /PROSE/ A(50,50),B(50,50),L3P,L3S,L4P,L4S,NSV,NEJ,NEL,NIND(PR2 0651
             1300),IPERH(50)                                                    PR2 0652
        C                                                                       PR2 0653
0011          DIMENSION ER(50), EI(50), XA(50,50), SSV(50)                      PR2 0654
        C                                                                       PR2 0655
        C     BEGIN CODE                                                        PR2 0656
        C                                                                       PR2 0657
        C     SET UP INITIAL CONSTANT OFFSETS                                   PR2 0658
0012          CALL SIMTR (KDUM,0)                                               PR2 0659
0013          DO 1 I=1,NSV                                                      PR2 0660
0014        1 SSV(I)=D(I)                                                       PR2 0661
        C     LOOP THRU STATE VARIABLES                                         PR2 0662
        C     THRU TYPE 3'S FIRST                                               PR2 0663
0015          IF(L3S.GT.L3P) GO TO 4                                            PR2 0664
0016          DO 3 I=L3S,L3P                                                    PR2 0665
0017          C(I)=1.0D0                                                        PR2 0666
        C     COMPUTE DERIVATIVES                                               PR2 0667
0018          CALL SIMTR (KDUM,0)                                               PR2 0668
        C     FORM AN A MATRIX ROW                                              PR2 0669
0019          DO 2 J=1,NSV                                                      PR2 0670
0020        2 A(J,I-L3S+1)=D(J)-SSV(J)                                          PR2 0671
0021        3 C(I)=0.0D0                                                        PR2 0672
        C     THRU TYPE 4 STATE VARIABLES                                       PR2 0673
0022        4 IF(L4S.GT.L4P) GO TO 7                                            PR2 0674
0023          DO 6 I=L4S,L4P                                                    PR2 0675
0024          V(I)=1.0D0                                                        PR2 0676
0025          CALL SIMTR (KDUM,0)                                               PR2 0677
0026          DO 5 J=1,NSV                                                      PR2 0678
0027        5 A(J,I+L3P-L3S-L4S+2)=D(J)-SSV(J)                                  PR2 0679
```

ORIGINAL PAGE IS OF POOR QUALITY

```
0028          6    V(I)=C.CD0                                                    PR2 0680
              C    FORM B MATRIX IF NECESSARY                                    PR2 0681
0029          7    IF(NEJ.EQ.0) GO TO 10                                         PR2 0682
0030               II=1                                                          PR2 0683
0031               DO 9 I=1,NEL                                                  PR2 0684
0032               IF(NIND(I).NE.1) GO TO 9                                      PR2 0685
0033               XN(I)=1.0D0                                                   PR2 0686
0034               CALL SIMTR (KDUM,0)                                           PR2 0687
0035               DO 8 J=1,NSV                                                  PR2 0688
0036          8    B(J,II)=D(J)-SSV(J)                                           PR2 0689
0037               II=II+1                                                       PR2 0690
0038               XN(I)=C.CD0                                                   PR2 0691
0039          9    CONTINUE                                                      PR2 0692
              C    FORM EIGEN VALUES                                             PR2 0693
              C    USE ANOTHER ARRAY (SINCE A IS DESTROYED BY HSBG)              PR2 0694
0040         10    DO 11 I=1,NSV                                                 PR2 0695
0041               DO 11 J=1,NSV                                                 PR2 0696
0042         11    XA(I,J)=A(I,J)                                                PR2 0697
0043               CALL HSBG (NSV,XA,50)                                         PR2 0698
0044               CALL ATEIG (NSV,XA,ER,EI,IPERF,50)                            PR2 0699
0045               RETURN                                                        PR2 0700
0046               END                                                           PR2 0701
```

```
0C01                SUBROUTINE HSBG (N,A,IA)                                        PR2 0702
                    C                                                               PR2 0703
                    C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0704
                    C                                                               PR2 0705
                    C     REDUCE A MATRIX TO UPPER HESSENBERG FORM                  PR2 0706
                    C     N=ORDER OF A, IA=FIRST DIMENSION OF A                     PR2 0707
                    C     IBM SSP ROUTINE                                           PR2 0708
                    C                                                               PR2 0709
                    C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0710
                    C                                                               PR2 0711
0002                IMPLICIT REAL*8(A-H,O-Z)                                        PR2 0712
0003                DIMENSION A(1)                                                  PR2 0713
0004                L=N                                                             PR2 0714
0005                NIA=L*IA                                                        PR2 0715
0006                LIA=NIA-IA                                                      PR2 0716
                    C                                                               PR2 0717
                    C        L IS THE ROW INDEX OF THE ELIMINATION                  PR2 0718
                    C                                                               PR2 0719
0007                 1  IF(L-3) 19,2,2                                              PR2 0720
0008                 2  LIA=LIA-IA                                                  PR2 0721
0009                    L1=L-1                                                      PR2 0722
0010                    L2=L1-1                                                     PR2 0723
                    C                                                               PR2 0724
                    C       SEARCH FOR THE PIVOTAL ELEMENT IN THE LTH ROW           PR2 0725
                    C                                                               PR2 0726
0011                    ISUB=LIA+L                                                  PR2 0727
0012                    IPIV=ISUB-IA                                                PR2 0728
0013                    PIV=DABS(A(IPIV))                                           PR2 0729
0014                    IF(L-3) 6,6,3                                               PR2 0730
0015                 3  M=IPIV-IA                                                   PR2 0731
0016                    DO 5 I=L,M,IA                                               PR2 0732
0017                    T=DABS(A(I))                                                PR2 0733
0018                    IF(T-PIV) 5,5,4                                             PR2 0734
0019                 4  IPIV=I                                                      PR2 0735
0020                    PIV=T                                                       PR2 0736
0021                 5  CONTINUE                                                    PR2 0737
0022                 6  IF(PIV) 7,18,7                                              PR2 0738
0023                 7  IF(PIV-DABS(A(ISUB))) 11,11,8                               PR2 0739
                    C                                                               PR2 0740
                    C        INTERCHANGE THE COLUMNS                                PR2 0741
                    C                                                               PR2 0742
0024                 8  M=IPIV-L                                                    PR2 0743
0025                    DO 9 I=1,L                                                  PR2 0744
0026                    J=M+I                                                       PR2 0745
0027                    T=A(J)                                                      PR2 0746
0028                    K=LIA+I                                                     PR2 0747
0029                    A(J)=A(K)                                                   PR2 0748
0030                 9  A(K)=T                                                      PR2 0749
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                C
                C        INTERCHANGE THE ROWS
                C
0031                  M=L2-M/IA
0032                  DO 10 I=L1,NIA,IA.
0033                  T=A(I)
0034                  J=I-M
0035                  A(I)=A(J)
0036             10   A(J)=T
                C
                C        TERMS OF THE ELEMENTARY TRANSFORMATION
                C
0037             11   DO 12 I=L,LIA,IA
0038             12   A(I)=A(I)/A(ISUB)
                C
                C        RIGHT TRANSFORMATION
                C
0039                  J=-IA
0040                  DO 14 I=1,L2
0041                  J=J+IA
0042                  LJ=L+J
0043                  DO 13 K=1,L1
0044                  KJ=K+J
0045                  KL=K+LIA
0046             13   A(KJ)=A(KJ)-A(LJ)*A(KL)
0047             14   CONTINUE
                C
                C        LEFT TRANSFORMATION
                C
0048                  K=-IA
0049                  DO 16 I=1,N
0050                  K=K+IA
0051                  LK=K+L1
0052                  S=A(LK)
0053                  LJ=L-IA
0054                  DO 15 J=1,L2
0055                  JK=K+J
0056                  LJ=LJ+IA
0057             15   S=S+A(LJ)*A(JK)*1.0D0
0058             16   A(LK)=S
                C
                C        SET THE LOWER PART OF THE MATRIX TO ZERO
                C
0059                  DO 17 I=L,LIA,IA
0060             17   A(I)=0.0
0061             18   L=L1
0062                  GO TO 1
0063             19   RETURN
```

0064 END PR2 0798

ORIGINAL PAGE IS
OF POOR QUALITY

```
CCC1                SLBROLTINE INTGR8                                                PR2 0799
                C   <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 0800
                C                                                                    PR2 0801
                C   DUMMY INTGR8 RCUTINE                                             PR2 0802
                C                                                                    PR2 0803
                C   <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 0804
CCC2                RETURN                                                           PR2 0805
CCC3                END                                                              PR2 0806
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                  SUBROUTINE INVERS (NAMMAT,AMAT,NR,NC,IERROR)                          PR2 0807
0002                  IMPLICIT REAL*8(A-H,O-Z),INTEGER*4(I-N)                               PR2 0808
                                                                                            PR2 0809
                     C
                     C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0810
                     C                                                                      PR2 0811
                     C                INNAMMAT = NAME OF MATRIX TO BE OPERATED ON           PR2 0812
                     C                    AMAT = THE MATRIX TO BE SOLVED OR INVERTED        PR2 0813
                     C                      NR = THE NUMBER OF ROWS IN AMAT                 PR2 0814
                     C                      NC = THE NUMBER OF COLUMNS IN AMAT              PR2 0815
                     C                                 A( NC = NR + 1 FOR EQUATION SOLUTION PR2 0816
                     C                                 B( NC = 2 * NR FOR MATRIX INVERSION  PR2 0817
                     C                  IERROR = ONE IF THE MATRIX IS SINGULAR,OTHERWISE ZEROPR2 0818
                     C                                                                      PR2 0819
                     C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0820
                     C                                                                      PR2 0821
0003                  REAL*4 NAMMAT                                                         PR2 0822
0004                  COMMON /TAPES/ NOUTTP,INTAPE,LISTP,INOUTP,NEDITP,NSAVTP               PR2 0823
0005                  COMMON /CNTRLS/ TIME,XSTOPT,XMXERT,XIR,XTISSS,XMNISS,XMXISS,XMNAIEPR2 0824
                     1,XMXAIE,XIC,XRERNO,XMXPAS,XMNRIE,XMXRIE,XMNISC,XMXICP,XICRER,XICAEPR2 0825
                     2R,XMXCTP,XXDIMS,XYDIMS,XICPAS,XNOPRQ,XNTNDX,XNHEAD,XNOFEQ,XERT,XNRPR2 0826
                     3ERN,XCNTNL,XSTPAC,XPASNO,XRUNAC,XSTPSZ                                PR2 0827
0006                  DIMENSION AMAT(NR,NC)                                                 PR2 0828
0007                  EQUIVALENCE (NOUTTP,OUTAPE)                                           PR2 0829
0008                  INTEGER OUTAPE                                                        PR2 0830
0009                  IF(XICPAS.LE.XMXICP) GO TO 2                                          PR2 0831
0010                  WRITE (NOUTTP,1) XMXICP                                               PR2 0832
0011             1    FORMAT (34H1THE NEWTON-RAPHSON PASS LIMIT OF ,F5.0,48H HAS BEEN EXPR2 0833
                     1CEEDED WITHOUT ATTAINING CONVERGENCE)                                 PR2 0834
0012                  IERROR=10                                                             PR2 0835
0013                  GO TO 20                                                              PR2 0836
0014             2    IERROR=0                                                              PR2 0837
0015                  NRX2=2*NR                                                             PR2 0838
0016                  INK=NR                                                                PR2 0839
0017                  IF(NRX2.NE.NC) INK=NC                                                 PR2 0840
                     C     CHECK FOR 1X1 MATRIX AND SOLVE IF FOUND                         PR2 0841
0018                  IF(NR.NE.1) GO TO 3                                                   PR2 0842
0019                  AMAT(1,2)=AMAT(1,2)/AMAT(1,1)                                         PR2 0843
0020                  GO TO 20                                                              PR2 0844
0021             3    NR2=2*NR                                                              PR2 0845
                     C     IF MATRIX INVERSION DESIRED, GENERATE IDENTITY MATRIX           PR2 0846
0022                  IF(NC.LT.NR2) GO TO 6                                                 PR2 0847
0023                  JINC=NR+1                                                             PR2 0848
0024                  DO 5 INCR=1,NR                                                        PR2 0849
0025                  INCROW=INCR+NR                                                        PR2 0850
0026                  DO 5 INCOLM=JINC,NC                                                   PR2 0851
0027                  IF(INCOLM.EQ.INCROW) GO TO 4                                          PR2 0852
0028                  AMAT(INCR,INCOLM)=0.                                                  PR2 0853
0029                  GO TO 5                                                               PR2 0854
```

```
0030              4     AMAT(INCR,INCOLM)=1.                                          PR2 0855
0031              5     CONTINUE                                                      PR2 0856
0032              6     DO 19 I=1,NR                                                  PR2 0857
0033                    M=I+1                                                         PR2 0858
0034                    J=I                                                           PR2 0859
                  C      CHECK FOR LAST DIAGONAL ELEMENT                            PR2 0860
0035                    IF(I.EQ.NR) GO TO 9                                           PR2 0861
                  C      FIND THE LARGEST VALUE BELOW DIAGONAL ELEMENT IN THIS COLUMN PR2 0862
0036                    DO 8 N=M,NR                                                   PR2 0863
0037                    IF(DABS(AMAT(J,I))-DABS(AMAT(N,I))) 7,8,8                     PR2 0864
0038              7     J=N                                                           PR2 0865
0039              8     CONTINUE                                                      PR2 0866
                  C      SEE IF THE LARGEST ELEMENT IN ROW IS ZERO                  PR2 0867
0040              9     IF(AMAT(J,I)) 12,10,12                                        PR2 0868
                  C      INDICATES THAT MATRIX IS SINGULAR                          PR2 0869
0041             10     IERROR=11                                                     PR2 0870
0042                    WRITE (NOUTTP,11) NAMMAT                                      PR2 0871
0043             11     FORMAT (8H MATRIX ,A4,12H IS SINGULAR)                        PR2 0872
0044                    GO TO 20                                                      PR2 0873
                  C      SEE IF ROW INTERCHANGE IS NEEDED TO GET LARGEST ELEMENT ON DIAG. PR2 0874
0045             12     IF(J-I) 15,15,13                                              PR2 0875
                  C      INTERCHANGE ROWS OF DIAGONAL AND THOSE TO THE RIGHT OF IT   PR2 0876
0046             13     DO 14 N=I,NC                                                  PR2 0877
0047                    SAVE=AMAT(I,N)                                                PR2 0878
0048                    AMAT(I,N)=AMAT(J,N)                                           PR2 0879
0049             14     AMAT(J,N)=SAVE                                                PR2 0880
0050             15     DIAG=AMAT(I,I)                                                PR2 0881
                  C      DIVIDE ALL ELEMENTS TO RIGHT OF DIAG. IN DIAG. ROW BY DIAG. ELEM. PR2 0882
0051                    DO 16 J=M,NC                                                  PR2 0883
0052             16     AMAT(I,J)=AMAT(I,J)/DIAG                                      PR2 0884
0053                    AMAT(I,I)=0.                                                  PR2 0885
0054                    DO 18 K=1,NR                                                  PR2 0886
0055                    IF(AMAT(K,I).EQ.0.) GO TO 18                                  PR2 0887
0056                    IP=I+1                                                        PR2 0888
0057                    DO 17 J=IP,NC                                                 PR2 0889
0058             17     AMAT(K,J)=AMAT(K,J)-AMAT(K,I)*AMAT(I,J)                       PR2 0890
0059             18     CONTINUE                                                      PR2 0891
0060             19     CONTINUE                                                      PR2 0892
0061             20     CONTINUE                                                      PR2 0893
0062                    RETURN                                                        PR2 0894
0063                    END                                                           PR2 0895
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                SUBROUTINE OUTP                                                    PR2 0896
              C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 0897
              C                                                                        PR2 0898
              C     DUMMY OUTP ROUTINE                                                 PR2 0899
              C                                                                        PR2 0900
              C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 0901
0002                RETURN                                                             PR2 0902
0003                END                                                                PR2 0903
```

```
0001          SUBROUTINE SAVEPD                                                PR2 0904
        C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0905
        C                                                                      PR2 0906
        C     DUMMY SAVEPD ROUTINE                                             PR2 0907
        C                                                                      PR2 0908
        C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0909
0002          RETURN                                                           PR2 0910
0003          END                                                              PR2 0911
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                  SUBROUTINE SETUP2 (NAC,LELS,NSRC,NTAB)                                    PR2 0912
                      C                                                                         PR2 0913
                      C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0914
                      C                                                                         PR2 0915
                      C     SETUP ROUTINE                                                       PR2 0916
                      C                                                                         PR2 0917
                      C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 0918
                      C                                                                         PR2 0919
0002                        IMPLICIT REAL*8(A-H,O-Z)                                            PR2 0920
0003                        COMMON /CNTRLS/ TIME,S1(21),XNCPRQ,S2,XNHEAD,S3(6),XRUNNO,S4(11),XPR2 0921
                           1WRTSM,S5(46),XNEY,XNE7DT,XN8C,XN8R,XN8L,XNLC,XNLR,XNLL,XNJX,XNJ8DTPR2 0922
                           2,XNJO,XNJ9,S6(2),XNM33,XNM36,XNM66,XNDP,XNDPD,XNELTS,S7(50)       PR2 0923
0004                        COMMON /TAPES/ NPRT,NDTA,NPGM                                       PR2 0924
0005                        DIMENSION LELS(6,300)                                               PR2 0925
0006                        DIMENSION NSIMTR(5), NSRCDR(6), NTABLE(6), NNDPD(3)                 PR2 0926
0007                        DIMENSION NCUTPT(6), NXMUTL(5), NNAMES(5), NPRMTR(6)                PR2 0927
0008                        DIMENSION NIONBX(5), NDERIV(6), NRAY(80), CTRDLS(160)               PR2 0928
0009                        DIMENSION NUMS(10), LOCS(8)                                         PR2 0929
0010                        DATA NSIMTR/1HS,1HI,1HM,1HT,1HR/                                    PR2 0930
0011                        DATA NSRCDR/1HS,1HR,1HC,1HD,1HR,1HV/                                PR2 0931
0012                        DATA NTABLE/1HT,1HA,1HB,1HL,1HE,1HS/                                PR2 0932
0013                        DATA NNDPD/1H,,1HD,1HP/                                             PR2 0933
0014                        DATA NCUTPT/1HC,1HU,1HT,1HP,1HT,1HS/                                PR2 0934
0015                        DATA NXMUTL/1HX,1HM,1HU,1HT,1HL/                                    PR2 0935
0016                        DATA NNAMES/1HN,1HA,1HM,1HE,1HS/                                    PR2 0936
0017                        DATA NPRMTR/1HP,1HR,1HM,1HT,1HR,1HS/                                PR2 0937
0018                        DATA NIONBX/1HT,1HC,1HN,1H ,1HX/                                    PR2 0938
0019                        DATA NDERIV/1HD,1HE,1HR,1HI,1HV,1HS/                                PR2 0939
0020                        DATA LOCS/96,97,98,93,94,95,92,100/                                 PR2 0940
0021                        DATA NUMS/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                  PR2 0941
0022                        DATA NAST/1H*/                                                      PR2 0942
0023                        EQUIVALENCE (TIME,CTRDLS(1))                                        PR2 0943
0024                        EQUIVALENCE (NCOM,NNDPD(1))                                         PR2 0944
0025                        EQUIVALENCE (NBLK,NIONBX(4))                                        PR2 0945
0026                        NTAB=0                                                              PR2 0946
0027                        NSRC=0                                                              PR2 0947
0028                  1     READ (NPGM,2,END=33) NRAY                                           PR2 0948
0029                  2     FORMAT (80A1)                                                       PR2 0949
0030                        IF(LOCATE(NRAY,NSIMTR,9,5).EQ.1) GO TO 1                            PR2 0950
0031                        READ (NPGM,3) NRAY                                                  PR2 0951
0032                  3     FORMAT (////////,80A1)                                              PR2 0952
0033                        NAC=70                                                              PR2 0953
0034                        IF(NRAY(6).EQ.NAST) NAC=160                                         PR2 0954
0035                        READ (NDTA,4,END=35) (CTRDLS(I),I=1,NAC)                            PR2 0955
0036                  4     FORMAT (1P5D14.7)                                                   PR2 0956
0037                        IF(NAC.EQ.160) GO TO 6                                              PR2 0957
0038                        DO 5 I=71,160                                                       PR2 0958
0039                  5     CTRDLS(I)=0.0D0                                                     PR2 0959
```

```
      6   READ (NPGM,2) NRAY                                          PR2 0960
      7   IF(LOCATE(NRAY,NSRCDR,15,6).EQ.0) GO TO 9                   PR2 0961
          IF(LOCATE(NRAY,NTABLE,15,6).EQ.0) GO TO 11                  PR2 0962
          IF(NAC.EQ.160) GO TO 8                                      PR2 0963
          IF(LOCATE(NRAY,NXMUTL,15,5).EQ.0) GO TO 12                  PR2 0964
          IF(LOCATE(NRAY,NNAMES,15,5).EQ.0) GO TO 14                  PR2 0965
          IF(LOCATE(NRAY,NPRMTR,15,6).EQ.0) GO TO 21                  PR2 0966
          IF(LOCATE(NRAY,NDERIV,15,6).EQ.0) GO TO 23                  PR2 0967
          IF(LOCATE(NRAY,NIONBX,13,5).EQ.0) GO TO 25                  PR2 0968
      8   IF(LOCATE(NRAY,NOUTPT,15,6).EQ.0) GO TO 32                  PR2 0969
          GO TO 6                                                     PR2 0970
      9   K=22                                                        PR2 0971
     10   NHOL=NCNT(NRAY,K)                                           PR2 0972
          NSRC=NSRC+NHOL                                              PR2 0973
          IF(NRAY(72).EQ.NBLK) GO TO 6                                PR2 0974
          READ (NPGM,2) NRAY                                          PR2 0975
          IF(NRAY(6).NE.NAST) GO TO 7                                 PR2 0976
          K=7                                                         PR2 0977
          GO TO 10                                                    PR2 0978
     11   NTAB=NUMB(NRAY,25,27,K)                                     PR2 0979
          GO TO 6                                                     PR2 0980
     12   K=21                                                        PR2 0981
     13   NHOL=NCNT(NRAY,K)                                           PR2 0982
          XNM33=XNM33+NHOL                                            PR2 0983
          IF(NRAY(72).EQ.NBLK) GO TO 6                                PR2 0984
          READ (NPGM,2) NRAY                                          PR2 0985
          IF(NRAY(6).NE.NAST) GO TO 7                                 PR2 0986
          K=7                                                         PR2 0987
          GO TO 13                                                    PR2 0988
     14   NELTS=0                                                     PR2 0989
          I=21                                                        PR2 0990
          N=1                                                         PR2 0991
     15   DO 16 J=1,6                                                 PR2 0992
     16   LELS(J,NELTS+1)=NBLK                                        PR2 0993
     17   IF(NRAY(I).EQ.NCOM) GO TO 18                                PR2 0994
          LELS(N,NELTS+1)=NRAY(I)                                     PR2 0995
          IF(NRAY(I).EQ.NBLK) GO TO 19                                PR2 0996
          N=N+1                                                       PR2 0997
          I=I+1                                                       PR2 0998
          IF(I.LE.72) GO TO 17                                        PR2 0999
          READ (NPGM,2) NRAY                                          PR2 1000
          IF(NRAY(6).NE.NAST) GO TO 20                                PR2 1001
          I=7                                                         PR2 1002
          GO TO 17                                                    PR2 1003
     18   NELTS=NELTS+1                                               PR2 1004
          XNELTS=NELTS                                                PR2 1005
          N=1                                                         PR2 1006
          I=I+1                                                       PR2 1007
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0088                IF(I.LE.72) GO TO 15                                          PR2 1008
0089                READ (NPGM,2) NRAY                                            PR2 1009
0090                IF(NRAY(6).NE.NAST) GO TO 7                                   PR2 1010
0091                I=7                                                           PR2 1011
0092                GO TO 15                                                      PR2 1012
0093             19 NELTS=NELTS+1                                                 PR2 1013
0094                XNELTS=NELTS                                                  PR2 1014
0095                GO TO 6                                                       PR2 1015
0096             20 NELTS=NELTS+1                                                 PR2 1016
0097                XNELTS=NELTS                                                  PR2 1017
0098                GO TO 7                                                       PR2 1018
0099             21 K=22                                                          PR2 1019
0100             22 NHOL=NCNT(NRAY,K)                                             PR2 1020
0101                XNDP=XNDP+NHOL                                                PR2 1021
0102                IF(NRAY(72).EQ.NBLK) GO TO 6                                  PR2 1022
0103                READ (NPGM,2) NRAY                                            PR2 1023
0104                IF(NRAY(6).NE.NAST) GO TO 7                                   PR2 1024
0105                K=7                                                           PR2 1025
0106                GO TO 22                                                      PR2 1026
0107             23 DO 24 I=22,70                                                 PR2 1027
0108                IF(LOCATE(NRAY,XNDPD,I,3).EQ.0) XNDPD=XNDPD+1                 PR2 1028
0109             24 CONTINUE                                                      PR2 1029
0110                GO TO 6                                                       PR2 1030
0111             25 IF(NRAY(18).EQ.NRAY(9)) GO TO 6                               PR2 1031
0112                K=19                                                          PR2 1032
0113             26 IF(NRAY(K).EQ.NBLK) GO TO 6                                   PR2 1033
0114                DO 27 I=2,10                                                  PR2 1034
0115                IF(NRAY(K).EQ.NUMS(I)) GO TO 29                               PR2 1035
0116             27 CONTINUE                                                      PR2 1036
0117                WRITE (NPRT,28) NRAY,K                                        PR2 1037
0118             28 FORMAT (42H *** ERROR *** - - INVALID NUMBER IN SIMTR,/,80A1,6H CPR2 1038
                   1OL=,I3)                                                      PR2 1039
0119                STOP 8                                                        PR2 1040
0120             29 J=I-1                                                         PR2 1041
0121                K=K+2                                                         PR2 1042
0122             30 IF(K.GT.72) READ (NPGM,2) NRAY                                PR2 1043
0123                IF(K.GT.72) K=K-72+6                                          PR2 1044
0124                IF(NRAY(K).EQ.NBLK) GO TO 31                                  PR2 1045
0125                K=K+2                                                         PR2 1046
0126                GO TO 30                                                      PR2 1047
0127             31 KST=K                                                         PR2 1048
0128                KEN=K+2                                                       PR2 1049
0129                CTROLS(LOCS(J))=NUMB(NRAY,KST,KEN,K)                          PR2 1050
0130                K=KEN+5                                                       PR2 1051
0131                IF(K.LE.72) GO TO 26                                          PR2 1052
0132                READ (NPGM,2) NRAY                                            PR2 1053
0133                IF(NRAY(6).NE.NAST) GO TO 7                                   PR2 1054
0134                K=K-72+6                                                      PR2 1055
```

```
                GO TO 26                                                    PR2 1056
          32    RETURN                                                      PR2 1057
          33    WRITE (NPRT,34)                                             PR2 1058
          34    FORMAT (39H *** ERROR *** - - END OF FILE IN SIMTR)         PR2 1059
                STOP 8                                                      PR2 1060
          35    WRITE (NPRT,36)                                             PR2 1061
          36    FORMAT (40H *** ERROR *** - - END OF FILE ON DTASAV)        PR2 1062
                STOP 8                                                      PR2 1063
                END                                                         PR2 1064
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                SUBROUTINE TPHEAD                                                    PR2 1065
              C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 1066
              C                                                                          PR2 1067
              C     DUMMY TPHEAD ROUTINE                                                 PR2 1068
              C                                                                          PR2 1069
              C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 1070
0002                RETURN                                                               PR2 1071
0003                END                                                                  PR2 1072
```

```
0001          SUBROUTINE XLOAD                                                  PR2 1073
              C <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 1074
              C                                                                 PR2 1075
              C     DUMMY XLOAD ROUTINE                                         PR2 1076
              C                                                                 PR2 1077
              C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<PR2 1078
0002          RETURN                                                            PR2 1079
0003          END                                                               PR2 1080
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001                  FUNCTION LOCATE (NRAY,NSTR,KSTR,KLEN)                                  PR2 1081
                    C                                                                        PR2 1082
                    C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR2 1083
                    C                                                                        PR2 1084
                    C     LOCATE STRING (NSTR) IN ARRAY (NRAY) WITH                          PR2 1085
                    C     LENGTH (KLEN) STARTING AT (KSTR).                                  PR2 1086
                    C     LOCATE=0 IF STRING FOUND                                           PR2 1087
                    C     LOCATE=1 IF STRING NOT FOUND                                       PR2 1088
                    C                                                                        PR2 1089
                    C     <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR2 1090
                    C                                                                        PR2 1091
0002                      DIMENSION NRAY(80), NSTR(80)                                       PR2 1092
                    C                                                                        PR2 1093
0003                      LOCATE=1                                                           PR2 1094
0004                      DO 1 I=1,KLEN                                                      PR2 1095
0005                      KSTRIM=KSTR+I-1                                                    PR2 1096
0006                      IF(NRAY(KSTRIM).NE.NSTR(I)) RETURN                                 PR2 1097
0007                1     CONTINUE                                                           PR2 1098
0008                      LOCATE=0                                                           PR2 1099
0009                      RETURN                                                             PR2 1100
0010                      END                                                                PR2 1101
```

```
0001                  FUNCTION NCNT (NRAY,NST)                                  PR2 1102
                                                                                PR2 1103
                C                                                               PR2 1104
                C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 1104
                C                                                               PR2 1105
                C                                                               PR2 1106
                C     COUNT NUMBER OF COMMAS+1 IN ARRAY NRAY                    PR2 1107
                C     STARTING AT NST                                           PR2 1108
                C                                                               PR2 1109
                C     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>PR2 1109
                C                                                               PR2 1110
0002                  DIMENSION NRAY(80)                                        PR2 1111
0003                  DATA NBLK/1H /,NCOM/1H,/                                  PR2 1112
0004                  NCNT=1                                                    PR2 1113
0005                  DO 1 I=NST,72                                             PR2 1114
0006                  IF(NRAY(I).EQ.NCOM) NCNT=NCNT+1                           PR2 1115
0007                  IF(NRAY(I).EQ.NBLK) RETURN                                PR2 1116
0008                  CONTINUE                                                  PR2 1117
0009                  RETURN                                                    PR2 1118
0010                  END                                                       PR2 1119
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          FUNCTION NUMB (NRAY,KST,KEN,I)
C
C         <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
C         DECODE NUMBER AT KST IN NRAY UNITL KEN
C
C         <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
          COMMON /TAPES/ NPRT,NDTA,NPGM
          DIMENSION NRAY(80), NUMS(10)
          DATA NUMS/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
          DATA NBLK/1H /
          NUMB=0
          DO 4 I=KST,KEN
          IF(I.LE.72) GO TO 2
          READ (NPGM,1) NRAY
    1     FORMAT (80A1)
          I=7
          KEN=KEN-72+6
    2     CONTINUE
          IF(NRAY(I).EQ.NBLK) GO TO 4
          DO 3 J=1,10
          IF(NRAY(I).NE.NUMS(J)) GO TO 3
          NUMB=NUMB*10+J-1
    3     CONTINUE
    4     CONTINUE
          RETURN
          END
```

```
0001                FUNCTION XTABLE (TABLE,X)                                          PR2 1148
0002                IMPLICIT REAL*8(A-H,O-Z),INTEGER*4(I-N)                            PR2 1149
                C                                                                      PR2 1150
                C   <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR2 1151
                C                                                                      PR2 1152
                C   FUNCTION TO PERFORM STRAIGHT LINE EXTRAPOLATION OF TABULAR DATA   PR2 1153
                C                                                                      PR2 1154
                C   <><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><PR2 1155
                C                                                                      PR2 1156
0003                DIMENSION TABLE(2)                                                 PR2 1157
0004                LOOP=TABLE(1)-2.                                                   PR2 1158
                C   LOOK FOR INTERVAL OR EXACT VALUE OF INDEPENDENT VARIABLE           PR2 1159
0005                DO 4 INDX=3,LOOP,2                                                 PR2 1160
                C   TEST LOWER COORDINATE                                             PR2 1161
0006                IF(TABLE(INDX)-X) 1,2,5                                            PR2 1162
                C   TEST UPPER COORDINATE                                             PR2 1163
0007              1 IF(TABLE(INDX+2)-X) 4,3,5                                          PR2 1164
                C   EXACT MATCH TO LOW COORDINATE - EQUAT  AND RETURN                 PR2 1165
0008              2 Y=TABLE(INDX+1)                                                    PR2 1166
0009                GO TO 6                                                            PR2 1167
                C   EXACT MATCH TO UPPER COORDINATE - EQUAT AND RETURN                 PR2 1168
0010              3 Y=TABLE(INDX+3)                                                    PR2 1169
0011                GO TO 6                                                            PR2 1170
0012              4 CONTINUE                                                           PR2 1171
0013                INDX=LOOP-1                                                        PR2 1172
                C   PERFORM STRAIGHT LINE EXTRAPOLATION                               PR2 1173
                C   CALCULATE SLOPE                                                   PR2 1174
0014              5 SLOPE=(TABLE(INDX+3)-TABLE(INDX+1))/(TABLE(INDX+2)-TABLE(INDX))  PR2 1175
                C   CALCULATE Y INTERCEPT                                            PR2 1176
0015                YINTR=TABLE(INDX+1)-SLOPE*TABLE(INDX)                              PR2 1177
                C   CALCULATE DEPENDENT VARIABLE                                     PR2 1178
0016                Y=SLOPE*X+YINTR                                                    PR2 1179
                C   OUTPUT PERTINANT VARIABLES                                        PR2 1180
0017              6 XTABLE=Y                                                           PR2 1181
0018                RETURN                                                             PR2 1182
0019                END                                                                PR2 1183
```

ORIGINAL PAGE IS
OF POOR QUALITY