LA-8502-M
Manual
C.2

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

EOSMOD: Subroutine Package for Calculating Equations of State and Opacities

Los Alamos
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

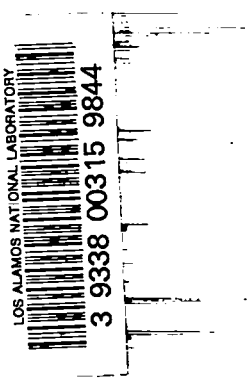
Edited by Helen M. Sinoradzki

LA-8502-M
Manual

UC-34
Issued: October 1981

EOSMOD: A Subroutine Package for Calculating Equations of State and Opacities

James M. Hyman
Morris M. Klein



Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CONTENTS

ABSTRACT 1

I. INTRODUCTION 1

II. EQUATION-OF-STATE AND OPACITY ROUTINES 2

A. Subroutine EOSDRE (input R and E, output P and T) 3
B. Subroutine EOSIPT (input P and T, output R and E) 5
C. Subroutine EOSIRT (input R and T, output P and E) 7
D. Subroutine EOSDRT (input R and T, output P and E) 9
E. Subroutine EOSORT (input R and T, output O) 11

III. SCALING THE TABLES 13

A. Density Scaling 13
B. Adding New Units 13

IV. USER NOTES 14

A. Information File 14
B. Graphic Output 14
C. Increasing LCM Allocation 14
D. Reducing the EOS Data Range 14
E. Error Flags and Messages 14
F. Creating an EOS/Opacity Table for Mixtures 14
G. Listing the Available Materials 15

V. EXAMPLES 15

A. Simplest Example 15
B. Advanced EOS Example 16

VI. MATERIALS AVAILABLE 18

VII. LOCATION OF COMPUTER FILES AT LOS ALAMOS 19

VIII. RELATIONSHIPS BETWEEN UNITS 20

ACKNOWLEDGMENTS 21

REFERENCES 22

APPENDIX A: SUBROUTINE LISTING 23

User Callable Subroutines 23

EOSDRE (input R and E, output P and T) 24
EOSDRT (input R and T, output P and E) 28
EOSIPT (input P and T, output R and E) 32
EOSIRT (input R and T, output P and E) 35
EOSORT (input R and T, output O) 38

Internal Subroutines 42

EOSBEG (initializes all the common block variables) 43
EOSCON (defines the table conversion factors) 46
EOSDSL (allows the user to scale the density) 49
EOSEFD (finds the EOS table) 51
EOSFAS (assigns the input files) 54
EOSGET (loads the Sesame EOS data files) 56
EOSKUT (checks the validity of KPARM) 59
EOSOFD (finds the opacity table) 61

T-4 Subroutines used by EOSMOD 63

DPACKX (packs real numbers) 64
GETEOSX (loads the total EOS tables) 65
GETINVX (gets inverted EOS tables) 67
GETRPOX (loads Rosseland/Planck opacity tables) 69
INBUFRX (sequential read) 71
INV301X (inverts a 301 table) 72
ISRCHKX (index search) 74
MATCHKX (checks if material table is loaded) 75
RATFN1X (one-dimensional interpolator) 76
TABRANX (fetches a given table for a given material from a Sesame II library) 78
T4DATIX (interpolates the inverted tables) 80
T4DATX (search/interpolate for atomic data as functions of region, density, and temperature) 83
T4INTPX (interpolates for a function z(x,y) and its derivatives) 86
T4PTREX (computes R and E from inverted table) 89
T4RTPEX (computes P and E from inverted table) 91

APPENDIX B: CROSS-REFERENCE DIRECTORY OF EOSLIB 93

APPENDIX C: TEST PROGRAM 100
OUTPUT 110

EOSMOD: A SUBROUTINE PACKAGE FOR CALCULATING EQUATIONS OF STATE AND OPACITIES

by

James M. Hyman and Morris M. Klein

ABSTRACT

The EOSMOD package includes a set of FORTRAN subroutines written to make the SESAME equation-of-state and opacity tables readily available . We have tried to make these routines as accessible as possible for casual users with routine problems and at the same time, allow sufficient flexibility for sophisticated users with complicated situations. This has caused only a slight loss in efficiency (~5%) compared to using the SESAME routines directly.

I. INTRODUCTION

The SESAME library is a collection of data files containing equations of state (EOS), Rosseland mean opacities, and other material properties over a wide range of temperatures, pressures, and densities. Group T-4 at Los Alamos National Laboratory developed a FORTRAN subroutine library[1-3] to access and analyze these data files. The EOSMOD subroutine package was developed to complement the T-4 routines and simplify the interface between these codes and the user's program. The EOSMOD routines do the bookkeeping, unit's conversion, and large core memory (LCM) initialization; reduce the number of parameters the casual user needs to be aware of; and are written in a structured modular design to allow a sophisticated user to modify and optimize the package easily for a particular problem.

To use the package, just access the SESAME data files as described in Sec. VIII and call the driver subroutine. The package then locates the requested data file, converts it to the units specified by the user, and stores it in LCM. On all subsequent calls, the package remembers the contents and location of the file and interpolates the data at the user's requested values.

If the EOS or opacity of a mixture is needed, then the directory of mixtures currently available (MIXDIR) should be checked. If the specific mixture is not in the library, then a SESAME data file may be generated using the procedure described in Refs. 4, 5, and 6.

II. EQUATION-OF-STATE AND OPACITY ROUTINES

The first time a routine is called for each new material (LMAT), the table is converted to the kind of units (KUNIT) requested by the user and copies into LCM. The location or material table number (IMATE) is returned to indicate the location of the EOS table or to indicate whether an error was encountered in the initialization; for example, if the material was not found in the library.

The EOS tables can be loaded in two different formats. The standard SESAME format is for calculating the pressure P and internal energy E of a material as a function of the density R and temperature T. Subroutine EOSDRT loads and reads the tables in this format.

The inverted SESAME format is for calculating P and T as a function of R and E with subroutine EOSDRE. These inverted tables also can be accessed in other ways using the iterative subroutines EOSIRT and EOSIPT. Subroutine EOSIPT calculates E and R as function of P and T, and subroutine EOSIRT calculates P and E as a function of R and T. These iterative subroutines use an iterative scheme that is slow compared to the direct methods used in subroutines EOSDRT and EOSDRE. Their main use is for initial state computations and occasional diagnostics.

To speed up the execution time when using subroutine EOSDRE or EOSDRT, the user can request that the package return only T, P, or E using the computational mode flag KBR. Computer time can also be saved by using the KFN flag to indicate a bilinear interpolation of the data rather than the more accurate (but slower) rational interpolation. These interpolation procedures are described in more detail in Refs. 1 and 7.

A. Subroutine EOSDRE (input R and E, output P and T)

The calling sequence when ρ and E are independent variables (input) and $P(\rho,E)$ and $T(\rho,E)$ are dependent variables (output) is

```
CALL EOSDRE (LMAT,R,E,P,T,KEOS,IMATE).
```

The arguments are defined as follows.

INPUT VARIABLES

LMAT — Material name in an A10 field; for example, LMAT = "HELIUM." The materials available are listed in Sec. VII and the file MIXDIR. (The material SESAME number can also be used to specify the material by setting LMAT to the SESAME number; for example, LMAT = "5760" for helium.)

R — Density (ρ).

E — Internal energy.

KEOS — Multiple parameter flag to describe how to write and retrieve the data file. KEOS has four decimal digits: KBR, KUNIT, KREPE, and KFN (KEOS = 1000*KBR + 100*KUNIT + 10*KREPE + KFN).

KBR — Computational flag to indicate which quantities and their partial derivatives are to be calculated and returned by the package.

- 0 — Compute P and T and their partial derivatives $\partial P/\partial\rho$, $\partial P/\partial E$, $\partial T/\partial\rho$, and $\partial T/\partial E$.
- 1 — Compute P and its partial derivatives $\partial P/\partial\rho$ and $\partial P/\partial E$.
- 2 — Compute T and its partial derivatives $\partial T/\partial\rho$ and $\partial T/\partial E$.

KUNIT — Kind of units for writing the data file. The units are explained fully in Sec. IX.

- 0 — SESAME EOS units.
- 1 — CGS units.
- 2 — Standard International Units (SIU).
- 3 — Hydrox EOS units.
- 4 — Hydrox opacity units.
- 5 — SESAME opacity units.
- 6 — LASNEX units.

KREPE — Computational flag to indicate whether E is to be represented as energy per unit mass or energy per unit volume.

- 0 — Energy in units of energy per unit mass (for example, ergs/g). This is the usual E.

1 Energy in units of energy per unit volume (for example, ergs/cm^3). This is the energy density ρE commonly computed in hydrodynamic computer codes.

KFN Indicates the form of the function used to interpolate the data tables.

0 Accurate rational function interpolation.
1 Fast bilinear function interpolation.

IMATE Indicates whether to load the data file if it does not exist for LMAT or go directly to a previously loaded file.

0 Check if the data file for LMAT has been loaded. If not, search for the file, convert it to the proper units, and copy it into LCM using the inverted SESAME format.
N>0 Equal to the LMAT table number. This number was returned by the package on a previous call to subroutine EOSDRE, EOSIPT , or EOSIRT and is distinct for each LMAT. This option is faster than IMATE = 0 because it skips searching the directory of previously loaded data files.

OUTPUT VARIABLES

P Array of dimension 3 containing the pressure and its partial derivatives.

P(1) Pressure.
P(2) Density derivative of the pressure (∂P/∂ρ).
P(3) Energy derivative of the pressure (∂P/∂E).

T Array of dimension 3 containing the temperature and its partial derivatives. This array must be dimensional even if the partial derivatives are not computed.

T(1) Temperature.
T(2) Density derivative of the temperature (∂T/∂ρ).
T(3) Energy derivative of the temperature (∂T/∂E).

IMATE Indicates the success or failure of locating and loading the data file for LMAT.

0 LMAT not found.
N>0 LMAT table number (success).
N<0 Insufficient LCM storage. The LCM memory allocation must be increased by at least |N| storage locations by the procedure described in Sec. V.

B. Subroutine EOSIPT (input P and T, output R and E)

The calling sequence when P and T are independent variables (input) and $\rho(P,T)$ and $E(P,T)$ are dependent variables (output) is

```
CALL EOSIPT (LMAT,P,T,R,E,KEOS,IMATE).
```

This routine reads the EOS data in the inverted SESAME format and uses an iterative method to interpolate the data. The arguments are defined as follows.

INPUT VARIABLES

LMAT — Material name in an A10 field; for example, LMAT = "HELIUM." The materials available are listed in Sec. VII and the file MIXDIR. (The material SESAME number can also be used to specify the material by setting LMAT to the SESAME number; for example, LMAT = "5760" for helium.)

P — Pressure.

T — Temperature.

KEOS — Multiple parameter flag to describe how to write and retrieve the data file. KEOS has three decimal digits: KUNIT, KREPE, and KFN (KEOS = 100*KUNIT + 10*KREPE + KFN).

KUNIT — Kind of units for writing the data file. The units are explained fully in Sec. IX.

0	SESAME EOS units.
1	CGS units.
2	Standard International Units (SIU).
3	Hydrox EOS units.
4	Hydrox opacity units.
5	SESAME opacity units.
6	LASNEX units.

KREPE — Computational flag to indicate whether E is to be represented as energy per unit mass or energy per unit volume.

0	Energy in units of energy per unit mass (for example, ergs/g). This is the usual E.
1	Energy in units of energy per unit volume (for example, ergs/cm^3). This is the energy density ρE commonly computed in hydrodynamic computer codes.

KFN — Indicates the form of the function used to interpolate the data tables.

0	Accurate rational function interpolation.
1	Fast bilinear function interpolation.

IMATE Indicates whether to load the data file if it does not exist for LMAT or go directly to a previously loaded file.

0	Check if the data file for LMAT has been loaded. If not, search for the file, convert it to the proper units, and copy it into LCM using the inverted SESAME format.
N>0	Equal to the LMAT table number. This number was returned by the package on a previous call to subroutine EOSDRE, EOSIPT, or EOSIRT and is distinct for each LMAT. This option is faster than IMATE = 0 because it skips searching the directory of previously loaded data files.

OUTPUT VARIABLES

R Density.

E Internal energy.

IMATE Indicates the success or failure of locating and loading the data file for LMAT.

0	LMAT not found or the iteration failed to converge.
N>0	LMAT table number (success).
N<0	Insufficient LCM storage. The LCM memory allocation must be increased by at least |N| storage locations by the procedure described in Sec. V.

C. Subroutine EOSIRT (input R and T, output P and E)

The calling sequence when ρ and T are independent variables (input) and P(ρ,T) and E(ρ,T) are dependent variables (output) is

CALL EOSIRT (LMAT,R,T,P,E,KEOS,IMATE).

This routine reads the EOS data in the inverted SESAME format and uses an iterative method to interpolate the data. The arguments are defined as follows.

INPUT VARIABLES

LMAT — Material name in an A10 field; for example, LMAT = "HELIUM." The materials available are listed in Sec. VII and the file MIXDIR. (The material SESAME number can also be used to specify the material by setting LMAT to the SESAME number; for example, LMAT = "5760" for helium.)

R — Density.

T — Temperature.

KEOS — Multiple parameter flag to describe how to write and retain the data file. KEOS has three decimal digits: KUNIT, KREPE, and KFN (KEOS = 100*KUNIT + 10*KREPE + KFN).

KUNIT — Kind of units for writing the data file. The units are explained fully in Sec. IX.

0	SESAME EOS units.
1	CGS units.
2	Standard International Units (SIU).
3	Hydrox EOS units.
4	Hydrox opacity units.
5	SESAME opacity units.
6	LASNEX units.

KREPE — Computational flag to indicate whether E is to be represented as energy per unit mass or energy per unit volume.

0	Energy in units of energy per unit mass (for example, ergs/g). This is the usual E.
1	Energy in units of energy per unit volume (for example, ergs/cm^3). This is the energy density ρE commonly computed in hydrodynamic computer codes.

KFN — Indicates the form of the function used to interpolate the data tables.

0	Accurate rational function interpolation.
1	Fast bilinear function interpolation.

IMATE Indicates whether to load the data file if it does not exist for LMAT or go directly to a previously loaded file.

0	Check if the data file for LMAT has been loaded. If not, search for the file, convert it to the proper units, and copy it into LCM using the inverted SESAME format.
N>0	Equal to the LMAT table number. This number was returned by the package on a previous call to subroutine EOSDRE, EOSIPT, EOSDRT, or EOSIRT and is distinct for each LMAT. This option is faster than IMATE = 0 because it skips searching the directory of previously loaded data files.

OUTPUT VARIABLES

P Pressure.

E Internal energy.

IMATE Indicates the success or failure of locating and loading the data file for LMAT.

0	Material not found or iteration failed to converge.
N>0	LMAT table number (success).
N<0	Insufficient LCM storage. The LCM memory allocation must be increased by at least |N| storage locations by the procedure described in Sec. V.

D. Subroutine EOSDRT (input R and T, output P and E)

The calling sequence when ρ and T are independent variables (input) and P(ρ,T) and E(ρ,T) are dependent variables (output) is

```
CALL EOSDRT (LMAT,R,T,P,E,KEOS,IMATE).
```

The arguments are defined as follows.

INPUT VARIABLES

LMAT — Material name in an A10 field; for example, LMAT = "HELIUM." The materials available are listed in Sec. VII and in the file MIXDIR. (The material SESAME number can also be used to specify the material by setting LMAT to the SESAME number; for example, LMAT = "5760" for helium.)

R — Density (ρ).

T — Temperature.

KEOS — Multiple parameter flag to describe how to write and retrieve the data file. KEOS has four decimal digits: KBR, KUNIT, KREPE, and KFN (KEOS = 1000*KBR + 100*KUNIT + 10*KREPE + KFN).

KBR — Computational flag to indicate which quantities and their partial derivatives are to be calculated and returned by the package.

- 0 — Compute P and E and their partial derivatives $\partial P/\partial \rho$, $\partial P/\partial T$, $\partial E/\partial \rho$, and $\partial E/\partial T$.
- 1 — Compute P and its partial derivatives $\partial P/\partial \rho$ and $\partial P/\partial T$.
- 2 — Compute T and its partial derivatives $\partial E/\partial \rho$ and $\partial E/\partial T$.

KUNIT — Kind of units for writing the data file. The units are explained fully in Sec. IX.

- 0 — SESAME EOS units.
  CGS units.
- 2 — Standard International Units (SIU).
- 3 — Hydrox EOS units.
- 4 — Hydrox opacity units.
- 5 — SESAME opacity units.
- 6 — LASNEX units.

KREPE — Computational flag to indicate whether E is to be represented as energy per unit mass or energy per unit volume.

- 0 — Energy in units of energy per unit mass (for example, ergs/g). This is the usual E.

1 Energy in units of energy per unit volume (for example, ergs/cm^3). This is the energy density ρE commonly computed in hydrodynamic computer codes.

KFN Indicates the form of the function used to interpolate the data tables.

0 Accurate rational function interpolation.
1 Fast bilinear function interpolation.

IMATE Indicates whether to load the data file if it does not exist for LMAT or go directly to a previously loaded file.

0 Check if the data file for LMAT has been loaded. If not, search for the file, convert it to the proper units, and copy it into LCM using the standard SESAME format.
N>0 Equal to the LMAT table number. This number was returned by the package on a previous call to subroutine EOSDRE, EOSIPT, EOSDRT, or EOSIRT and is distinct for each LMAT. This option is faster than IMATE = 0 because it skips searching the directory of previously loaded data files.

OUTPUT VARIABLES

P Array of dimension 3 containing the pressure and its partial derivatives.

P(1) Pressure.
P(2) Density derivative of the pressure ($\partial P/\partial \rho$).
P(3) Temperature derivative of the pressure ($\partial P/\partial T$).

E Array of dimension 3 containing the internal energy and its partial derivatives. This array must be dimensional even if the partial derivatives are not computed.

E(1) Internal energy.
E(2) Density derivative of the internal energy ($\partial E/\partial \rho$).
E(3) Energy derivative of the internal energy ($\partial E/\partial T$).

IMATE Indicates the success or failure of locating and loading the data file for LMAT.

0 LMAT not found.
N>0 LMAT table number (success).
N<0 Insufficient LCM storage. The LCM memory allocation must be increased by at least |N| storage locations by the procedure described in Sec. V.

E. Subroutine EOSORT (input R and T, output O)

The Rosseland mean opacity[8] of the material (LMAT) can be calculated by calling subroutine OPCRT. The density R and temperature T are the independent input variables, and the opacity O is the dependent output variable. When the internal energy, but not the temperature, is known, the user must first call subroutine EOSDRE to get the temperature. The (KUNIT) that the table is to be written in and the type of interpolation function (KFN) to be used must be provided by the user. The material opacity table number (IMATO) is returned by the package to indicate the location of the opacity table in LCM or if an error was encountered by the routine.

The calling sequence when ρ and T are independent variables (input) and O is the dependent variable (output) is

```
CALL EOSORT (LMAT,R,T,O,KOPC,IMATO)
```

The arguments are defined as follows.

INPUT VARIABLES

LMAT — Material name in an A10 field; for example, LMAT = "HELIUM." The materials available are listed in Sec. VII and in the file MIXDIR. (The material SESAME number can also be used to specify the material by setting LMAT to the SESAME number; LMAT = "15760".

R — Density (ρ).

T — Temperature.

KOPC — Multiple parameter flag to describe how to write and retrieve the data file. KEOS has three decimal digits: KUNIT, KREPO, and KFN (KEOS = 100*KUNIT + 10*KREPO + KFN).

KUNIT — Computational flag to indicate the kind of units for writing the data file. The units are explained fully in Sec. IX.

0	SESAME EOS units.
1	CGS units.
2	Standard International Units (SIU).
3	Hydrox EOS units.
4	Hydrox opacity units.
5	SESAME opacity units.
6	LASNEX units.

KREPO — Computational flag to indicate which representation to use for the opacity.

0	Opacity represented as κ in dimensional units of length^2/mass.
1	Opacity represented as a mean-free path, $\Lambda = 1/(\kappa\rho)$, in dimensional units of length.

KFN — Indicates the form of the function used to interpolate the data tables.

0	Accurate rational function interpolation.
1	Fast bilinear function interpolation.

IMATO — Indicates whether to load the data file if it does not exist for LMAT or go directly to a previously loaded file.

0	Check if the data file for LMAT has been loaded. If not, search for the file, convert it to the proper units, and copy it into LCM.
N>0	Equal to the LMAT table number. This number was returned by the package on a previous call to subroutine OPCRT and is distinct for each LMAT. This option is faster than IMATE = 0 because it skips searching the directory of previously loaded data files.

OUTPUT VARIABLES

O — Opacity.

IMATEO — Indicates the success or failure of locating and loading the data file for LMAT.

0	LMAT not found.
N>0	LMAT table number (success).
N<0	Insufficient LCM storage. The LCM memory allocation must be increased by at least |N| storage locations by the procedure described in Sec. V.

III. SCALING THE TABLES

A. Density Scaling

When the atomic mass of the desired material is different from the mass of the SESAME material and the material properties are expected to be similar, then a simple density scale factor can be incorporated into the EOS and opacity tables. This is usually sufficient for isotope mixtures of a specific material.

The density scale factor (DSFAC) for the material LMAT can be incorporated into the tables through the user supplied subroutine EOSSCL(LMAT, DSFAC). This routine will be called by EOSMOD, and the routine should return DSFAC equal to the ratio of the atomic mass of the SESAME material and the desired material; that is,

$$\text{DSFAC} = \frac{\text{atomic mass of SESAME material LMAT}}{\text{atomic mass of desired material}} .$$

In example B, Sec. VI, the EOS of a 60% deuterium/40% tritium mixture is approximated from the EOS of deuterium by defining

$$\text{DSFAC} = \frac{2}{0.60 \times 2 + 0.40 \times 3} = 0.833 .$$

B. Adding New Units

When the EOS and opacity tables are copied from the SESAME data file into LCM, they are converted to the user's specified units by multiplying the data by a conversion factor. The EOS conversion factors for temperature, density, pressure, and internal energy are TFACE, RFACE, PFACE, and EFACE, respectively. The opacity conversion factors for density, temperature, and opacity are RFACO, TFACO, and OFACO, respectively.

These factors are defined in the EOSMOD subroutine EOSCON according to the user's specifications (KUNIT) and communicated to the various EOSMOD routines through the common blocks

```
COMMON /EOSCCE/ TFACE,RFACE,PFACE,EFACE
COMMON /EOSCCO/ TFACO,RFACO,OFACO
```

If the desired units are not automatically available in the package, the user can reset the conversion factors at execution time (not in a data statement) in the user-provided routine EOSSCL (See Sec. V.A). This is done in example 5B.

The scale factor is the constant that the data in the original SESAME EOS table units (KUNIT = 0) or SESAME opacity table units (KUNIT = 5) must be multiplied by to convert the data to the desired units. The conversion factors for KUNIT = 0-6 are easily found in subroutine EOSCON listed in the Appendix A.

IV. USER NOTES

A. Information File

The file EOSINFO contains the latest user notes and information on the EOSMOD package. Users are encouraged to add notes to this file that will be helpful to others using the package.

B. Graphic Output

Several plotting routines exist for displaying EOS and opacity data stored in the SESAME format. The routines are maintained by Los Alamos Group T-4 and are described in the data file S2DHELP. See Ref. 9.

C. Increasing LCM Allocation

On the CDC 7600's at the Laboratory, the EOSMOD default LCM allocation is 12 000 words. This is enough storage to load approximately four EOS tables. The allocation can be increased to LCMX, say 20 000 words, by declaring

```
LEVEL 2, TBLS
COMMON /S2DIR/ LCMX
COMMON /SESDAT/ TBLS (20000)
LCMX=20000
```

in the main program. LCMX must be set to the dimension of TBLS at execution time not in a data statement. Example 2B in Sec. VI does this.

D. Reducing the EOS Data Range

Subroutine WINDOW in the Hydses package can be used to reduce the size of a standard temperature-based SESAME data table when the full density and temperature range are not needed. Because the use of this routine requires knowledge of where and how the tables are stored in LCM, we refer the interested user to the HYDSES report[2,10] for further information.

E. Error Flags and Messages

All error messages are written into a file called "OUTPUT." The name of this file is defined at compile time in the Hollerith variable LOUT in the common block

```
COMMON/EOSCZ/LOUT.
```

LOUT can be changed to another file name or unit number of the user at execution time before the first call to EOSMOD.

F. Creating an EOS/Opacity Table for Mixtures

The EOS/opacity tables suitable for gas mixtures can be created with the aid of the BCON controller MIXB.[5] These EOS mixture tables currently are prepared under the assumption that the ideal mixing of individual com-

ponent parts occurs. The pressure of the mixture is taken, as in Dalton's law, to be the weighted mole fraction of the partial pressures of the component parts. Opacity mixture tables are created by weighing the opacities of each component in frequency space according to the component's fraction of the total mass and then by integrating the resultant frequency spectrum to obtain the Rossland mean opacity for the mixture. These tables are generated in SESAME format with a simple input deck.

G. Listing the Available Materials

The EOS and opacity materials, and SESAME numbers are listed in the common blocks

```
COMMON/EOSC5/NMAT,LABMAT(40),IDMAT(40)
COMMON/EOSC7/NMATO,LABMO(40),IDMATO(40).
```

There are NMAT (NMATO) EOS (opacity) materials in the common blocks. The labels in LABMAT (LABMO) are in an A10 format and correspond to the SESAME material ID in IDMAT (IDMATO).

To list the EOS tables, execute the code

```
      DO 10  I = 1,NMAT
10    PRINT 20, LABMAT(I),IDMAT(I)
20    FORMAT (1X,A10,I10).
```

A complete description of each material can be found using the T-4 SESAME utility LSTX. (See Sec. VIII and Ref. 11.)

V. EXAMPLES

A. Simplest Example

The following program will compute the pressure of helium in microbars at a temperature of 300 K and a density of 0.01 g/cm^3.

```
      PROGRAM TST(OUTPUT)
      DIMENSION P(3),E(3)
      LMAT = "HELIUM"
      R = 0.01
      T = 300.0
      KEOS = 110
      IMATE = 0
      CALL EOSDRT(LMAT,R,T,P,E,KEOS,IMATE)
      PRINT 10, P(1)
10    FORMAT("PRESSURE = ", 1PE12.4," MICROBARS")
      CALL EXIT
      END
```

The execute line is

```
FTN (I=TEST, GLIB=EOSLIB, GO)
```

The output from this program is:

```
PRESSURE = 6.3498E+07 MICROBARS
```

B. Advanced EOS Example

In this example we scale the density for a 60% deuterium/40% tritium mixture, add a new set of MKS units, and increase LCM so we can load four EOS tables.

```
      PROGRAM TST(OUTPUT)
C
C     DECLARE THE COMMON BLOCKS USED BY EOSMOD(HYDSES) TO STORE THE TABLES
      LEVEL 2,TBLS
      COMMON /S2DIR/ LCMX
      COMMON /SESDAT/ TBLS(20000)
C
C     DECLARE THE COMMON BLOCK WITH THE FILE LABELS
      COMMON /EOSC2/ LF41,LF42,LF43,LF44,LF45
C
      DIMENSION P(3),E(3)
C
C     INCREASE THE LCM STORAGE AVAILABLE TO EOSMOD
C     THE MAXIMUM LCM STORAGE IN /SESDAT/ IS LCMX WORDS
      LCMX = 20000

C     SET THE DENSITY IN KILOGRAMS
C     AND THE TEMPERATURE IN DEGREES KELVIN

      R = 1.E-5
      T = 300.0
      KEOS = 110
C
C     THE TABLE FOR DEUTERIUM IS CONVERTED BY EOSMOD TO A MIXTURE BY
C     SCALING THE DENSITY IN SUBROUTINE EOSSLL.
      LMAT = "DEUTERIUM"
      IMATE = 0
      CALL EOSDRT(LMAT,R,T,P,E,KEOS,IMATE)
      PRINT 10,P(1)
   10 FORMAT ("40% D + 60% T, PRESSURE =", 1PE12.4, "BARS")
C
      LMAT = "HELIUM"
      IMATE = 0
      CALL EOSDRT(LMAT,R,T,P,E,KEOS,IMATE)
      PRINT 20,LMAT,P(1)
C
```

```
C     DECLARE THE NAME OF THE PRIVATE EOS DATA TABLES AND DIRECTORY
      LF44 = "MIXLIB"
      LF45 = "MIXDIR"
C
C     FIND EOS PRESSURE OF A 90% DEUTERIUM 10% NEON MIXTURE.
C     THE EOS IS CONTAINED IN THE PRIVATE TABLES MIXLIB AND MIXDIR
      LMAT = "D90NE10"
      IMATE = 0
      CALL EOSDRT(LMAT,R,T,P,E,KEOS,IMATE)
      PRINT 20,LMAT,P(1)
C
   20 FORMAT(1X,A10," PRESSURE = ",1PE12.4," "BARS")
C
      CALL EXIT
      END

      SUBROUTINE EOSSCL (LMAT,DSFAC)
      COMMON /EOSCCE/ TFACE,RFACE,PFACE,EFACE,KPE
C
C     SET UP A NEW SET OF UNITS TO BE USED BY THE CODE
C     PFACE AND EFACE ARE THE CONVERSION FACTORS FOR THE MKS SYSTEM
C     YOU ONLY NEED TO DEFINE THE CONVERSION FACTORS THAT DIFFER
C     FROM THE KEOS DECLARED UNITS (CGS SYSTEM)
C     CONVERT THE PRESSURE TO BARS
      PFACE = 1.E+6
      EFACE = 1.E+6
C
C     WE NOW RESCALE THE DENSITY FOR DEUTERIUM TO WHAT IT
C     SHOULD BE FOR A 60% DEUTERIUM AND 40% TRIDIUM MIXTURE

      IF(LMAT.EQ."DEUTERIUM") DSFAC=0.833
C
      RETURN
      END
```

The output from this program is

```
40% D + 60% T        PRESSURE = 6.1923E+04 BARS
HELIUM               PRESSURE = 5.6817E+04 BARS
D90NE10              PRESSURE = 7.3926E+05 BARS
```

VI. MATERIALS AVAILABLE

The following materials are currently available using EOSMOD.[11]

Material Name (LMAT)	EOS File Name	Opacity File Name	Temperature Maximum (K)	Density Minimum (g/cm^3)	Density Minimum (g/cm^3)	SESAME Material Number
ALLUVIUM (Nevada)	SESAME		4×10^8	0.018	4.7×10^4	7111
ALUMINUM	SESAME	SESAME	4×10^8	0.021	2.7×10^3	3710
AL2O3 (= Al_2O_3)	SESAME		4×10^8	0.031	4.0×10^8	7410
ARGON		SESAME				5170
BERYLLIUM	SESAME	SESAME	4×10^8	0.014	4.0×10^4	2020
BORON		SESAME				2330
BORON CARB(ide)	SESAME		6×10^4	0.0002	3.2	7081
BRASS	SESAME		4×10^8	0.066	1.7×10^5	4100
CALCIUM		SESAME				2030
CARBON		SESAME				2180
CH (= polystyrene)	SESAME		4×10^8	0.0082	2.1×10^4	7590
CH2 (= polyethylene)	SESAME		4×10^8	0.082	1.8×10^4	7170
CHLORINE		SESAME				5020
CHROMIUM		SESAME				3070
COPPER	SESAME		4×10^8	0.070	1.8×10^5	3330
DEUTERIUM	SESAME	SESAME	4×10^8	0.0	3.5×10^3	5263
GOLD	SESAME		4×10^8	0.15	3.9×10^5	2700
GRANITE	SESAME		4×10^8	0.021	5.3×10^4	7390
HELIUM	SESAME	SESAME	1×10^8	0.0018	4.7×10^3	5760
HE (= High Explosive)	SESAME		4×10^8	0.014	3.7×10^4	8180
IRON	SESAME	SESAME	4×10^8	0.061	1.6×10^5	2140
IRON2	SESAME		1.2×10^4	0.0	12.5	2145
LEAD	SESAME		4×10^8	0.088	2.3×10^5	3200
LITHIUM		SESAME				2290
6LID (= ^{6}LiD)	SESAME		4×10^8	0.0062	1.6×10^4	7240
6LIH (= ^{6}LiH)	SESAME		4×10^8	0.0053	1.4×10^4	7370
MAGNESIUM		SESAME				3080
MOLYBDENUM	SESAME		4×10^8	0.08	0.2×10^5	2980
NEON	SESAME		4×10^8	0.011	2.9×10^4	5410
NICKEL	SESAME		4×10^8	0.069	1.8×10^5	3100
NITROGEN		SESAME				5000
OXYGEN		SESAME				5010
PBX-9502		SESAME				8200
PHOSPHORUS		SESAME				3910
PLATINUM	SESAME		4×10^8	0.17	2.1×10^4	3730
POLYE (= polyethylene)	SESAME		4×10^8	0.0072	1.8×10^4	7170

Material Name (LMAT)	EOS File Name	Opacity File Name	Temperature Maximum (K)	Density Minimum (g/cm^3)	Density Minimum (g/cm^3)	SESAME Material Number
POLYS (= polystyrene)	SESAME		4×10^8	0.0082	2.1×10^4	7590
POTASSIUM		SESAME				2460
SIO2 (= SiO_2)	SESAME	SESAME	4×10^8	0.017	4.4×10^4	7380
SILICON		SESAME				3810
SODIUM	SESAME	SESAME	1×10^4	0.0	1.3	2448
SS (Steel)	SESAME	SESAME	4×10^8	0.062	1.6×10^5	4270
STAINLESS (Steel)	SESAME		4×10^8	0.062	1.6×10^5	4270
STEAM	SESAME		1300	0.0	0.9	7151
SULPHUR		SESAME				4010
TITANIUM		SESAME				2960
TITANIUM N (Nitride)		SESAME				6000
UO2 (= UO_2)	SESAME		3×10^4	0.0	14.3	7432
URANIUM	SESAME		4×10^8	0.15	4.0×10^5	1540
URETHANE	SESAME		4×10^9	0.0099	2.5×10^4	7560
VERMICULIT	SESAME		4×10^8	0.021	5.4×10^4	7520
WATER	SESAME	SESAME	1.8×10^8	2.0×10^{-6}	4.0×10^2	7150

On the CRAY-1 use only the first eight characters in the above material names.

VII. LOCATION OF COMPUTER FILES AT LOS ALAMOS

The files needed to execute any of the EOSMOD subroutines are available on the CDC 7600 computers at Los Alamos. For most users, it will be sufficient to attach the EOSMOD library,

```
MASS GET/EODMOD/EOSLIB
```

and load the binary source with their program, that is,

```
FTN (I = program, GLIB = EOSLIB, ...)
```

For more advanced users, we list the location of most of the SESAME files which may be useful in complicated situations

File Name	Description	CFS File Location
EOSFTN	FORTRAN source of EOSMOD	/EOSMOD/EOSFTN
EOSINFO	User information	/EOSMOD/EOSINFO
EOSLIB	Compiled FTN Library file of EOSMOD	/EOSMOD/EOSLIB

File Name	Description	CFS File Location
MIXDATA	T-7 mixture EOS and opacity data file	/EOSMOD/MIXLIB
MIXDIR	directory of MIXLIB	/EOSMOD/MIXDIR
EOSTST	Test program	/EOSMOD/EOSTST
MIXB	T-4 procedure to generate mixture	
HYDSES	Subroutine package for using SESAME in hydrodynamic codes	
SAC		/088077/SES/SAC
SAX	change files in a library	/088077/SES/SAX
SESAME	unclassified EOS data file	public
SESAMEA	classified EOS data file	secret
SESAME	unclassified opacity data file	public
S2MV2	create, modify, and print EOS data	/SESAME/SEMV2
S2DV3	plots SESAME data	/SESAME/S2DV3
S2DHELP	help package for SES2D	/SESAME/S2DHELP
DSPLX	computes Hugoniots, isentropes and isobars[12]	/SESAME/DSPLX
S3D	3-D graphics for EOS data	/SESAME/S3D
LSTX	list of current SESAME materials	/SESAME/LSTX

VIII. RELATIONSHIPS BETWEEN UNITS

The KUNIT parameter indicates the kind of units the table is to be written in for R, T, P, E, and O. This parameter can have the following integer values and corresponding meanings

KUNIT

0 SESAME EOS Units

R - grams/cm^3
E - megajoules/kilogram (= 10^{10} ergs/gram)
P - gigapascals (= 10^{10} dyne/cm^2)
T - degrees Kelvin
O - cm^2/gram[1]

1 CGS Units

R - grams/cm^3
E - ergs/gram
P - microbars (= 1 dyne/cm^2)
T - degrees Kelvin
O - cm^2/gram

2 Standard International Units (SIU)

R - kilograms/$meter^3$
E - joules/kilogram (= 10 ergs/gram)
P - pascals (= 10 dyne/cm^2)

T - degrees Kelvin
O - $meter^2/kilogram^2$

3 Hydrox EOS Units

R - $grams/cm^3$
E - megabar $cm^3/gram$ (= 10^{12} ergs/gram)
P - megabar (= 10^{12} dyn/cm^2)
T - degrees Kelvin
O - $gram/cm^2$

4 Hydrox Opacity Units

R - $grams/cm^3$
E - megabar $cm^2/gram$ (= 10^{12} ergs/gram)
P - megabars (= 10^{12} $dyne/cm^2$)
T - electron volts
O - $gram/cm^2$

5 SESAME Opacity Units

R - $grams/cm^3$
E - megajoules/kilogram (= 10^{10} ergs/gram)
P - gigapascals (= 10^{10} $dyne/cm^2$)
T - electron volts
O - $cm^2/gram^1$

6 LASNEX Units

R - $grams/cm^3$
E - jerks (= 1 erg/gram)
P - $jerks/cm^3$ (= 1 $dyne/cm^2$)
T - keV
O - $cm^2/gram$

The EOS and opacity tables are scaled according to the numerical value of KUNIT when the tables are copied into LCM. The scaling factors used to convert the tables are defined in subroutine EOSCON listed in the Appendixes. This subroutine can be changed easily by the user to write the tables in units other than those provided automatically by the package.

ACKNOWLEDGMENTS

We wish to thank J. Abdallah, Jr., M. Argo, S. Colgate, W. Huebner, J. D. Johnson, G. Kerley, and S. Lyon for their constructive criticism, advice, and help in developing this package.

REFERENCES

1. N. G. Cooper, "An Invitation to Participate in the LASL Equation-of-State Library," Los Alamos Scientific Laboratory report LASL-79-62 (1979).

2. J. Abdallah, Jr., G. I. Kerley, B. I. Bennett, J. D. Johnson, R. E. Albers, and W. F. Huebner, "HYDSES: A Subroutine Package for Using SESAME in Hydrodynamic Codes," Los Alamos Scientific Laboratory report LA-8209 (June 1980).

3. B. I. Bennett, J. D. Johnson, G. I. Kerley, and G. T. Rood, "Recent Developments in the SESAME Equation-of-State Library," Los Alamos Scientific Laboratory report LA-7130 (February 1978).

4. M. M. Klein, "User's Guide for Generating SESAME EOS and Opacity Tables for Mixtures," Los Alamos National Laboratory report (to be published).

5. Joseph Abdallah, Jr., "MIXB: A BCON Controller for Maintaining SESAME Library Files of Mixture Data," Los Alamos Scientific Laboratory report LA-8219-M (April 1980).

6. J. Abdallah, Jr. and W. F. Huebner, "QIKMIX: A Quick-Turnaround Computer Program for Computing Opacities of Mixtures," Los Alamos Scientific Laboratory report LA-7724-M (May 1979).

7. G. I. Kerley, "Rational Function Method of Interpolation," Los Alamos Scientific Laboratory report LA-6903-MS (August 1977).

8. W. F. Huebner, A. L. Merts, N. H. Magee, Jr., and M. F. Argo, "Astrophysical Opacity Library," Los Alamos Scientific Laboratory report LA-6760-M (August 1977).

9. J. D. Johnson, "SES2D Plotting Package," online computer writeup on Los Alamos Scientific Laboratory file CGS /SESAME/S2DHELP (1980).

10. G. I. Kerley, "Instructions for Use of the SESAME Library," Los Alamos National Laboratory memorandum T-4 (U) to Distribution, October 1977.

11. J. Abdullah, G. I. Kerley, and W. F. Huebner, "Contents of the SESAME Library," Los Alamos National Laboratory memorandum T-4 to Distribution, September 1980.

12. G. I. Kerley, "DSPLX - An LTSS Code for Displaying EOS Data from the SESAME Library," Los Alamos National Laboratory memorandum T-4 to Distribution, November 1977.

APPENDIX A

SUBROUTINE LISTING

User Callable Routines

EOSDRE (input R and E, output P and T) 24
EOSDRT (input R and T, output P and E) 28
EOSIPT (input P and T, output R and E) 32
EOSIRT (input R and T, output P and E) 35
EOSORT (input R and T, output O) 38

```
      SUBROUTINE EOSORE (LMAT,R,E,P,T,KEOS,IMATE)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     *  GIVEN THE OENSITY (R) AND ENERGY (E) OF A MATERIAL (LMAT)
C     *  THIS SUBROUTINE RETURNS THE PRESSURE (P) AND INTERNAL
C     *  TEMPERATURE (T) USING THE LASL T-4 SESAME EOS ROUTINES
C     *
C     * INPUT VARIABLES-
C     *
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USEO TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     *  R = OENSITY (RHO)
C     *
C     *  E = INTERNAL ENERGY
C     *
C     *  KEOS = MULTIPLE PARAMETER FLAG TO OESCRIBE HOW TO WRITE
C     *         AND RETRIEVE THE OATA FILE. KEOS HAS FOUR OECIMAL DIGITS,
C     *
C     *  KEOS = 1000*KBR + 100*KUNIT + 10*KREPE + KFN  WHERE
C     *
C     *   KBR = COMPUTATIONAL MOOE FLAG TO INDICATE WHICH
C     *         QUANTITIES AND THEIR PARTIAL OERIVATIVES ARE TO
C     *         BE CALCULATED AND RETURNEO BY THE PACKAGE.
C     *           = 0   COMPUTE PRESSURE AND TEMPERATURE
C     *           = 1   COMPUTE PRESSURE ONLY
C     *           = 2   COMPUTE TEMPERATURE ONLY
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-OEG.K,O-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-OEG.K,O-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-OEG.K,P-PA,E-J/KG,O-M**2/KG
C     *     3 (HYOROXE) R-G/CC,T-OEG.K,P=MBR,E-MBR*CC/GM,O-CM**2/G
C     *     4 (HYOROXO) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,O-CM**2/G
C     *     5 (SESAMEO) R-G/CC,T-EV,O-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,O-CM**2/GM,P-JRKS/CC,EO-JRKS/CC
C     *
C     *      LEGENO-
C     *             R = OENSITY
C     *             T = TEMPERATURE
C     *             O = OPACITY
C     *             P = PRESSURE
C     *             E = INTERNAL ENERGY
C     *
C     *             CC = CUBIC CENTIMETER
C     *             CM = CENTIMETER
C     *             OEG. K = OEGREES KELVIN
C     *             EV = ELECTRON VOLT
C     *             G = GRAM
C     *             GPA = GIGA PASCALS
C     *             J = JOULES
C     *             JRKS = JERKS
C     *             KEV = KILO ELECTRON VOLTS
C     *             KG = KILOGRAM
C     *             M = METER
C     *             MBR = MEGABAR
C     *             MUBR = MICROBAR
C     *             PA = PASCAL
```

```
C     *
C     *  KREPE = COMPUTATION FLAG TO INDICATE WHETHER E IS
C     *          IS TO BE REPRESENTED AS ENERGY PER UNIT MASS OR ENERGY
C     *          PER UNIT VOLUME
C     *       0 ENERGY IN UNITS OF ENERGY PER UNIT MASS. FOR EXAMPLE-
C     *          UNITS OF ERGS/GRAM WHEN KUNIT = 1.  THIS IS THE USUAL E.
C     *       1 ENERGY IN UNITS OF ENERGY PER UNIT VOLUME. FOR EXAPMLE-
C     *          UNITS OF ERGS/CM**3 WHEN KUNIT = 1. THIS IS THE ENERGY
C     *          DENSITY RHO*E COMMONLY COMPUTED IN HYORODYNAMIC
C     *          COMPUTER CODES.
C     *
C     *  KFN = KIND OF FUNTION INTERPOLATION IN THE TABLES
C     *    = 0 RATIONAL APPROXIMATIONS (ACCURATE)
C     *    = 1 BILINEAR APPROXIMATIONS (FAST)
C     *
C     * IMATE = INDICATES WHETHER TO LOAD THE DATA FILE IF
C     *         IT DOES NOT EXIST FOR LMAT OR GO DIRECTLY TO A
C     *         PREVIOUSLY LOADED FILE.
C     *   =0 CHECK IF THE DATA FILE FOR LMAT HAS BEEN LOADED.
C     *      IF NOT, SEARCH FOR THE FILE, COMVERT IT TO THE PROPER UNITS
C     *      AND COPY IT INTO LCM USING THE INVERTED SESAME FORMAT.
C     *   >0 EQUAL TO THE LMAT TABLE NUMBER.  THIS NUMBER WAS RETURNED
C     *      BY THE PACKAGE ON A PREVIOUS CALL TO SUBROUTINE EOSORE,
C     *      EOSIPT OR EOSIRT.  THIS OPTION IS FASTER THAN IMATE = 0
C     *      BECAUSE IT SKIPS SEARCH IN THE DIRECTOR OF THE
C     *
C     * OUTPUT VARIABLES-
C     *
C     * P = ARRAY OF DIMENSION 3 CONTAINING THE PRESSURE AND
C     *      ITS PARTIAL DERIVATIVES.  THIS ARRAY MUST BE DIMENSIONED
C     *      EVEN IF THE PARTIAL DERIVATIVES ARE NOT COMPUTED.
C     *
C     *  P(1) = PRESSURE
C     *  P(2) = DENSITY DERIVATIVE OF THE PRESSURE (DP/DR)
C     *  P(3) = TEMPERATURE DERIVATIVE OF THE PRESSURE (DP/DE)
C     *
C     * T = ARRAY OF DIMENSION 3 CONTAINING THE TEMPERATURE AND
C     *     ITS PARTIAL DERIVATIVES.  THIS ARRAY MUST BE DIMENSIONED
C     *     EVEN IF THE PARTIAL DERIVATIVES ARE NOT COMPUTED
C     *
C     *  T(1) = INTERNAL TEMPERATURE
C     *  T(2) = DENSITY DERIVATIVE OF THE TEMPERATURE (DT/DR)
C     *  T(3) = ENERGY DERIVATIVE OF THE TEMPERATURE (DT/DE)
C     *
C     * IMATE = INDICATES THE SUCCESS OR FAILURE OF
C     *         LOCATING AND LOADING THE DATA FILE FOR LMAT.
C     *
C     *       = N>0 MATERIAL TABLE NUMBER (SUCCESS)
C     *         0    MATERIAL (LMAT) NOT IN LIBRARY
C     *         -N  (N>1) INSUFFICIENT STORAGE
C     *             THE LCM STORAGE MUST BE INCREADED BY AT LEAST
C     *             N STORAGE LOCATIONS.  SEE THE EOSMOD MANUAL
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * SAMPLE DRIVER PROGRAM-
C     *
C     *     PROGRAM TST(OUTPUT)
C     *     DIMENSION P(3),E(3)
C     *     LMAT = "HELIUM"
C     *     R = 0.001
C     *     E = 1.0
```

```
C      *      KEOS = 110
C      *      IMATE = 0
C      *      CALL EOSORE(LMAT,R,E,P,T,KEOS,IMATE)
C      *      PRINT 10,P(1)
C      *   10 FORMAT(" PRESSURE = ",E10.2," MICROBARS")
C      *      CALL EXIT
C      *      END
C      *
C      * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C      *
C      * REFERENCE- J. M. HYMAN, M. M. KLEIN
C      *            EOSMOO- A SUBROUTINE PACKAGE FOR CALCULATING
C      *            EQUATIONS-OF-STATE AND OPACITIES
C      *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C      *
C      * OATE- MARCH 6, 1980
C      *
C      ****************************************************************
C
C      COMMON BLOCKS FOR THE SESAME EOS ROUTINES
       LEVEL 2, TBLS
       COMMON /S20IRX/ LCMX, NRS, LCFW(10,3)
       COMMON /SESOATX/ TBLS(11000)
       COMMON /INTOROX/ KFN
       COMMON /SESINX/ IRC, IOT, RHO, ENERGY, KBR, IFL
       COMMON /SESOUTX/ PRES(3), TEMP(3)
C
C      COMMON BLOCKS FOR THE EOSMOO ROUTINES
       COMMON /EOSCZ/ LOUT
       COMMON /EOSC3/ INIT, IRDIM, IR(60,3), KUP(60,3)
       COMMON /EOSC4/ NTABLE, NTABLO, IFLP, LCNT
C
       OIMENSION T(3), P(3)
       OATA KEOSS /-99/,KBRS/0/,KFNS/0/,LMATS/1H /
C
C      ****************************************************************
C
C      IDT=1 LOCATOR OF OATA TYPE FOR IR,GETINVX(,, IOT,...)
C      FOR THE INVERTEO SESAME FORMAT
       IOT=1
C
C      CHECK IF THE UNITS HAVE CHANGEO SINCE THE LAST CALL
       IF(KEOSS.NE.KEOS.OR.LMAT.NE.LMATS) GO TO 5
       LMATS=LMAT
       KBR=KBRS
       KFN=KFNS
       GO TO 10
     5 CONTINUE
C
C      CHECK THE VALIOITY OF THE INPUT PARAMETERS
       CALL EOSKUT(KEOS,KBR,KUNIT,KREPE,KFN,KEOSS,KBRS,KFNS,IMATE,IOT
      1 ,IERR)
       IF(IERR.LT.0) GO TO 80
    10 CONTINUE
C
C      FINO THE MATERIAL
       IF (IMATE.GT.0) GO TO 60
       CALL EOSGET(LMAT,KUNIT,KREPE,IMATE,IOT,IERR)
       IF(IMATE.LE.0.OR.IERR.LT.0) GO TO 80
    60 CONTINUE
C
C      TRANSFER INPUT CALL PARAMETERS TO COMMON BLOCK
```

```
      ENERGY=E
      RHO=R
      IRC=IR(IMATE,1)
C
C     CALCULATE THE EQUATION OF STATE
      CALL T4DATIX
C       * T4DATIX DOES NOT RETURN AN ERROR FLAG FOR DATA OUT OF RANGE
C
C     RESTORE OUTPUT VARIABLES FOR RETURN
   75 CONTINUE
      P(1)=PRES(1)
      P(2)=PRES(2)
      P(3)=PRES(3)
      T(1)=TEMP(1)
      T(2)=TEMP(2)
      T(3)=TEMP(3)
C
   80 CONTINUE
      IFLP=IERR
      RETURN
      END
```

```
      SUBROUTINE EOSORT (LMAT,R,T,P,E,KEOS,IMATE)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     *  GIVEN THE DENSITY (R) AND TEMPERATURE (T) OF A MATERIAL (LMAT)
C     *  THIS SUBROUTINE RETURNS THE PRESSURE (P) AND INTERNAL
C     *  ENERGY (E) USING THE LASL T-4 SESAME EOS ROUTINES
C     *
C     *
C     * INPUT VARIABLES-
C     *
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     *  R = DENSITY (RHO)
C     *
C     *  T = TEMPERATURE
C     *
C     *  KEOS = MULTIPLE PARAMETER FLAG TO DESCRIBE HOW TO WRITE
C     *         AND RETRIEVE THE DATA FILE. KEOS HAS FOUR DECIMAL DIGITS,
C     *
C     *  KEOS = 1000*KBR + 100*KUNIT + 10*KREPE + KFN  WHERE
C     *
C     *   KBR = COMPUTATIONAL MODE FLAG TO INDICATE WHICH
C     *         QUANTITIES AND THEIR PARTIAL DERIVATIVES ARE TO
C     *         BE CALCULATED AND RETURNED BY THE PACKAGE.
C     *           = 0   COMPUTE PRESSURE AND TEMPERATURE
C     *           = 1   COMPUTE PRESSURE ONLY
C     *           = 2   COMPUTE TEMPERATURE ONLY
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,D-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,D-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,D-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,D-CM**2/G
C     *     4 (HYDROXD) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,D-CM**2/G
C     *     5 (SESAMED) R-G/CC,T-EV.D-CM**2/G.P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,D-CM**2/GM,P-JRKS/CC,ED-JRKS/CC
C     *
C     *      LEGEND-
C     *             R = DENSITY
C     *             T = TEMPERATURE
C     *             D = OPACITY
C     *             P = PRESSURE
C     *             E = INTERNAL ENERGY
C     *
C     *             CC = CUBIC CENTIMETER
C     *             CM = CENTIMETER
C     *             DEG. K = DEGREES KELVIN
C     *             EV = ELECTRON VOLT
C     *             G = GRAM
C     *             GPA = GIGA PASCALS
C     *             J = JOULES
C     *             JRKS = JERKS
C     *             KEV = KILO ELECTRON VOLTS
C     *             KG = KILOGRAM
C     *             M = METER
C     *             MBR = MEGABAR
C     *             MUBR = MICROBAR
```

```
C     *            PA = PASCAL
C     *
C     *  KREPE = COMPUTATION FLAG TO INDICATE WHETHER E IS
C     *          IS TO BE REPRESENTED AS ENERGY PER UNIT MASS OR ENERGY
C     *          PER UNIT VOLUME
C     *       0 ENERGY IN UNITS OF ENERGY PER UNIT MASS. FOR EXAMPLE-
C     *          UNITS OF ERGS/GRAM WHEN KUNIT = 1.  THIS IS THE USUAL E.
C     *       1 ENERGY IN UNITS OF ENERGY PER UNIT VOLUME. FOR EXAPMLE-
C     *          UNITS OF ERGS/CM**3 WHEN KUNIT = 1. THIS IS THE ENERGY
C     *          DENSITY RHO*E COMMONLY COMPUTED IN HYDRODYNAMIC
C     *          COMPUTER CODES.
C     *
C     *  KFN = KIND OF FUNTION INTERPOLATION IN THE TABLES
C     *    = 0 RATIONAL APPROXIMATIONS (ACCURATE)
C     *    = 1 BILINEAR APPROXIMATIONS (FAST)
C     *
C     *  IMATE = INDICATES WHETHER TO LOAD THE DATA FILE IF
C     *          IT DOES NOT EXIST FOR LMAT OR GO DIRECTLY TO A
C     *          PREVIOUSLY LOADED FILE.
C     *    =0 CHECK IF THE DATA FILE FOR LMAT HAS BEEN LOADED.
C     *       IF NOT, SEARCH FOR THE FILE, COMVERT IT TO THE PROPER UNITS
C     *       AND COPY IT INTO LCM USING THE STANDARD SESAME FORMAT.
C     *    >0 EQUAL TO THE LMAT TABLE NUMBER.  THIS NUMBER WAS RETURNED
C     *       BY THE PACKAGE ON A PREVIOUS CALL TO SUBROUTINE EOSORE,
C     *       EOSIPT OR EOSIRT.  THIS OPTION IS FASTER THAN IMATE = 0
C     *       BECAUSE IT SKIPS SEARCH IN THE DIRECTOR OF THE
C     *
C     *  OUTPUT VARIABLES-
C     *
C     *  P = ARRAY OF DIMENSION 3 CONTAINING THE PRESSURE AND
C     *       ITS PARTIAL DERIVATIVES.  THIS ARRAY MUST BE DIMENSIONED
C     *       EVEN IF THE PARTIAL DERIVATIVES ARE NOT COMPUTED.
C     *
C     *   P(1) = PRESSURE
C     *   P(2) = DENSITY DERIVATIVE OF THE PRESSURE (DP/DR)
C     *   P(3) = TEMPERATURE DERIVATIVE OF THE PRESSURE (DP/DT)
C     *
C     *  E = ARRAY OF DIMENSION 3 CONTAINING THE ENERGY AND
C     *      ITS PARTIAL DERIVATIVES.  THIS ARRAY MUST BE DIMENSIONED
C     *      EVEN IF THE PARTIAL DERIVATIVES ARE NOT COMPUTED
C     *
C     *   E(1) = INTERNAL ENERGY
C     *   E(2) = DENSITY DERIVATIVE OF THE ENERGY (DE/DR)
C     *   E(3) = TEMPERATURE DERIVATIVE OF THE ENERGY (DE/DT)
C     *
C     *  IMATE = INDICATES THE SUCCESS OR FAILURE OF
C     *          LOCATING AND LOADING THE DATA FILE FOR LMAT.
C     *
C     *         = N>0 MATERIAL TABLE NUMBER (SUCCESS)
C     *           0    MATERIAL (LMAT) NOT IN LIBRARY
C     *           -N  (N>1) INSUFFICIENT STORAGE
C     *               THE LCM STORAGE MUST BE INCREADED BY AT LEAST
C     *               N STORAGE LOCATIONS.  SEE THE EOSMOD MANUAL
C     *
C     *  REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     *  SAMPLE DRIVER PROGRAM-
C     *
C     *      PROGRAM TST(OUTPUT)
C     *      DIMENSION P(3),E(3)
C     *      LMAT = "HELIUM"
C     *      R = 0.001
```

```
C     *      T = 300.0
C     *      KEOS = 110
C     *      IMATE = 0
C     *      CALL EOSORT(LMAT,R,T,P,E,KEOS,IMATE)
C     *      PRINT 10,P(1)
C     *   10 FORMAT(" PRESSURE = ",E10.2," MICROBARS")
C     *      CALL EXIT
C     *      END
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOO PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOO- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * OATE- MARCH 6, 1980
C     *
C     ****************************************************************
C
C     COMMON BLOCKS FOR THE SESAME EOS ROUTINES
      LEVEL 2, TBLS
      COMMON /S20IRX/ LCMX, NRS, LCFW(10,3)
      COMMON /SESOATX/ TBLS(11000)
      COMMON /INTOROX/ KFN
      COMMON /SESINX/ IRC, IOT, RHO, TEMP, KBR, IFL
      COMMON /SESOUTX/ PRES(3), ENERGY(3)
C
C     COMMON BLOCKS FOR THE EOSMOO ROUTINES
      COMMON /EOSCZ/ LOUT
      COMMON /EOSCCE/ TFACE, RFACE, PFACE, EFACE, KREPE
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC3/ INIT, IROIM, IR(60,3), KUP(60,3)
      COMMON /EOSC4/ NTABLE, NTABLO, IFLP, LCNT
C
      OIMENSION E(3), P(3)
      OATA KEOSS /-99/,KBRS/0/,KFNS/0/,LMATS/1H /
C
C     ****************************************************************
C
C     IOT=3 LOCATOR OF OATA TYPE FOR IR,GETEOSX(,, IOT,...)
C     FOR THE STANOARO SESAME FORMAT
      IOT=3
C
C     CHECK IF THE UNITS HAVE CHANGED SINCE THE LAST CALL
      IF(KEOSS.NE.KEOS.OR.LMAT.NE.LMATS) GO TO 5
      LMATS=LMAT
      KBR=KBRS
      KFN=KFNS
      GO TO 10
    5 CONTINUE
C
C     CHECK THE VALIOITY OF THE INPUT PARAMETERS
      CALL EOSKUT(KEOS,KBR,KUNIT,KREPE,KFN,KEOSS,KBRS,KFNS,IMATE,IOT
     1 ,IERR)
      IF(IERR.LT.0) GO TO 80
   10 CONTINUE
C
C     FIND THE MATERIAL
      IF (IMATE.GT.0) GO TO 60
```

```
      CALL EOSGET(LMAT,KUNIT,KREPE,IMATE,IOT,IERR)
      IF(IMATE.LE.0.OR.IERR.LT.0) GO TO 80
   60 CONTINUE
C
C     TRANSFER INPUT CALL PARAMETERS TO COMMON BLOCK
      TEMP=T
      RHO=R
      IRC=IR(IMATE,3)
C
C     CALCULATE THE EQUATION OF STATE
     *CALL T40ATX
C      * T40ATX DOES NOT RETURN AN EROR FLAG FOR DATA OUT OF RANGE
C
   75 CONTINUE
C     RESTORE OUTPUT VARIABLES FOR RETURN TO CALL
      P(1)=PRES(1)
      P(2)=PRES(2)
      P(3)=PRES(3)
      E(1)=ENERGY(1)
      E(2)=ENERGY(2)
      E(3)=ENERGY(3)
C
   80 CONTINUE
      IFLP=IERR
      RETURN
      END
```

```
      SUBROUTINE EOSIPT (LMAT,P,T,R,E,KEOS,IMATE)
C
C     ************************************************************
C     *
C     * PURPOSE-
C     *  GIVEN THE PRESSURE (P) AND TEMPERATURE (T) OF A MATERIAL (LMAT)
C     *  THIS SUBROUTINE RETURNS THE PRESSURE (P) AND INTERNAL
C     *  ENERGY (E) USING THE LASL T-4 SESAME EQUATION OF STATE ROUTINES
C     *
C     * AN ITERATIVE METHOD IS USED TO INTERPOLATE THE TABLES
C     *
C     * INPUT VARIABLES-
C     *
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     *  P = PRESSURE
C     *
C     *  T = TEMPERATURE
C     *
C     *  KEOS = MULTIPLE PARAMETER FLAG TO DESCRIBE HOW TO WRITE
C     *         AND RETRIEVE THE DATA FILE. KEOS HAS FOUR DECIMAL DIGITS,
C     *
C     *  KEOS = 100*KUNIT + 10*KREPE  WHERE
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,D-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,D-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,D-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,D-CM**2/G
C     *     4 (HYDROXD) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,D-CM**2/G
C     *     5 (SESAMED) R-G/CC,T-EV,D-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,D-CM**2/GM,P-JRKS/CC,ED-JRKS/CC
C     *
C     *      LEGEND-
C     *            R = DENSITY
C     *            T = TEMPERATURE
C     *            D = OPACITY
C     *            P = PRESSURE
C     *            E = INTERNAL ENERGY
C     *
C     *            CC = CUBIC CENTIMETER
C     *            CM = CENTIMETER
C     *            DEG. K = DEGREES KELVIN
C     *            EV = ELECTRON VOLT
C     *            G = GRAM
C     *            GPA = GIGA PASCALS
C     *            J = JOULES
C     *            JRKS = JERKS
C     *            KEV = KILO ELECTRON VOLTS
C     *            KG = KILOGRAM
C     *            M = METER
C     *            MBR = MEGABAR
C     *            MUBR = MICROBAR
C     *            PA = PASCAL
C     *
C     *  KREPE = COMPUTATION FLAG TO INDICATE WHETHER E IS
C     *          IS TO BE REPRESENTED AS ENERGY PER UNIT MASS OR ENERGY
C     *          PER UNIT VOLUME
C     *       0 ENERGY IN UNITS OF ENERGY PER UNIT MASS. FOR EXAMPLE-
```

```
C     *         UNITS OF ERGS/GRAM WHEN KUNIT = 1.  THIS IS THE USUAL E.
C     *      1 ENERGY IN UNITS OF ENERGY PER UNIT VOLUME. FOR EXAPMLE-
C     *         UNITS OF ERGS/CM**3 WHEN KUNIT = 1. THIS IS THE ENERGY
C     *         DENSITY RHO*E COMMONLY COMPUTED IN HYDRODYNAMIC
C     *         COMPUTER CODES.
C     *
C     * IMATE = INDICATES WHETHER TO LOAD THE DATA FILE IF
C     *        IT DOES NOT EXIST FOR LMAT OR GO DIRECTLY TO A
C     *        PREVIOUSLY LOADED FILE.
C     *   =0 CHECK IF THE DATA FILE FOR LMAT HAS BEEN LOADED.
C     *      IF NOT, SEARCH FOR THE FILE, COMVERT IT TO THE PROPER UNITS
C     *      AND COPY IT INTO LCM USING THE INVERTED SESAME FORMAT.
C     *   >0 EQUAL TO THE LMAT TABLE NUMBER.  THIS NUMBER WAS RETURNED
C     *      BY THE PACKAGE ON A PREVIOUS CALL TO SUBROUTINE EOSORE,
C     *      EOSIPT OR EOSIRT.  THIS OPTION IS FASTER THAN IMATE = 0
C     *      BECAUSE IT SKIPS SEARCH IN THE DIRECTOR OF THE
C     *
C     * OUTPUT VARIABLES-
C     *
C     * R = DENSITY
C     *
C     * E = INTERNAL ENERGY
C     *
C     * IMATE = INDICATES THE SUCCESS OR FAILURE OF
C     *         LOCATING AND LOADING THE DATA FILE FOR LMAT.
C     *
C     *        = N>0 MATERIAL TABLE NUMBER (SUCCESS)
C     *          0    MATERIAL (LMAT) NOT IN LIBRARY
C     *          -N  (N>1) INSUFFICIENT STORAGE
C     *              THE LCM STORAGE MUST BE INCREADED BY AT LEAST
C     *              N STORAGE LOCATIONS.  SEE THE EOSMOD MANUAL
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * SAMPLE DRIVER PROGRAM-
C     *
C     *     PROGRAM TST(OUTPUT)
C     *     DIMENSION P(3),E(3)
C     *     LMAT = "HELIUM"
C     *     R = 0.001
C     *     T = 300.0
C     *     KEOS = 110
C     *     IMATE = 0
C     *     CALL EOSIPT(LMAT,P,T,R,E,KEOS,IMATE)
C     *     PRINT 10,P(1)
C     *  10 FORMAT(" DENSITY = ",E10.2," MICROBARS")
C     *     CALL EXIT
C     *     END
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ************************************************************
C
```

```
C     COMMON BLOCKS FOR THE SESAME EOS ROUTINES
      LEVEL 2, TBLS
      COMMON /S2OIRX/ LCMX, NRS, LCFW(10,3)
      COMMON /SESOATX/ TBLS(11000)
      COMMON /SESINX/ DUM(4), KBR, DUM1
      COMMON /INTORDX/ KFN
C
C     COMMON BLOCKS FOR THE EOSMOD ROUTINES
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC3/ INIT, IROIM, IR(60,3), KUP(60,3)
      COMMON /EOSC4/ NTABLE, NTABLO, IFLP, LCNT
C
      DIMENSION R(3), E(3)
      DATA KEOSS /-99/,KBRS/0/,KFNS/0/,LMATS/1H /
C
C     ***************************************************************
C
C     IOT=1 LOCATOR OF DATA TYPE FOR IR,GETINVX(,, IOT,...)
C     FOR THE INVERTED SESAME FORMAT
      IDT=1
C
C     CHECK IF THE UNITS HAVE CHANGED SINCE THE LAST CALL
      IF(KEOSS.NE.KEOS.OR.LMAT.NE.LMATS) GO TO 5
      LMATS=LMAT
      KBR=KBRS
      KFN=KFNS
      GO TO 10
    5 CONTINUE
C
C     CHECK THE VALIDITY OF THE INPUT PARAMETERS
      CALL EOSKUT(KEOS,KBR,KUNIT,KREPE,KFN,KEOSS,KBRS,KFNS,IMATE,IOT
     1 ,IERR)
      IF(IERR.LT.0) GO TO 75
   10 CONTINUE
C
C     FIND THE MATERIAL
      IF (IMATE.GT.0) GO TO 60
      CALL EOSGET(LMAT,KUNIT,KREPE,IMATE,IOT,IERR)
      IF(IMATE.LE.0.OR.IERR.LT.0) GO TO 75
   60 CONTINUE
C
C     CALCULATE THE EQUATION OF STATE
      CALL T4PTREX (IR(IMATE,1),1,TBLS,P,T,R,E,IERR)
C
C     PRINT AN ERROR MESSAGE IF T4PTREX FAILES TO CONVERGE
      IF (IERR.EQ.0) WRITE(LOUT,80) LMAT,P,T
      IF (IERR.EQ.0) IMATE=0
C
   75 CONTINUE
      IFLP=IERR
      RETURN
C
   80 FORMAT(" FAILED TO CONVERGE WHEN ITERATING ON THE INVERTED",
     1 /," EOS TABLES IN SUBROUTINE T4PTREX CALLED BY EOSIPT",/,
     * 2 " THE REQUESTED VALUES FOR RHO, PRESSURE, TEMPERATURE AND ENERGY"
     2 ,/," MAY BE OUT OF RANGE OR NEAR THE EDGE OF THE TABLE",/,
     3 " CHECK THE EOSMOD WRITEUP FOR THE DATA RANGE",/,
     1   " LMAT =",A10,
     4 " PRESSURE =",1PE12.4," TEMPERATURE =",1PE12.4)
      END
```

```
      SUBROUTINE EOSIRT (LMAT,R,T,P,E,KEOS,IMATE)
C
C     ****************************************************************
C     *
C     * PURPOSE-
C     *  GIVEN THE DENSITY (R) AND TEMPERATURE (T) OF A MATERIAL )
C     *  (LMAT), THIS SUBROUTINE RETURNS THE PRESSURE (P) AND INTERNAL
C     *  ENERGY (T) USING THE LASL E-4 SESAME EQUATION OF STATE ROUTINES
C     *
C     * AN ITERATIVE METHOD IS USED TO INTERPOLATE THE TABLES
C     * SUBROUTINE EOSORT USES A FASTER DIRECT METHOD
C     *
C     * INPUT VARIABLES-
C     *
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     *  R = DENSITY (RHO)
C     *
C     *  T = TEMPERATURE
C     *
C     *  KEOS = MULTIPLE PARAMETER FLAG TO DESCRIBE HOW TO WRITE
C     *         AND RETRIEVE THE DATA FILE. KEOS HAS FOUR DECIMAL DIGITS,
C     *
C     *  KEOS = 100*KUNIT + 10*KREPE  WHERE
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,D-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,D-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,D-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,D-CM**2/G
C     *     4 (HYDROXD) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G.D-CM**2/G
C     *     5 (SESAMED) R-G/CC,T-EV,D-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,D-CM**2/GM,P-JRKS/CC,ED-JRKS/CC
C     *
C     *      LEGEND-
C     *             R = DENSITY
C     *             T = TEMPERATURE
C     *             D = OPACITY
C     *             P = PRESSURE
C     *             E = INTERNAL ENERGY
C     *
C     *             CC = CUBIC CENTIMETER
C     *             CM = CENTIMETER
C     *             DEG. K = DEGREES KELVIN
C     *             EV = ELECTRON VOLT
C     *             G = GRAM
C     *             GPA = GIGA PASCALS
C     *             J = JOULES
C     *             JRKS = JERKS
C     *             KEV = KILO ELECTRON VOLTS
C     *             KG = KILOGRAM
C     *             M = METER
C     *             MBR = MEGABAR
C     *             MUBR = MICROBAR
C     *             PA = PASCAL
C     *
C     *  KREPE = COMPUTATION FLAG TO INDICATE WHETHER T IS
C     *          IS TO BE REPRESENTED AS ENERGY PER UNIT MASS OR ENERGY
C     *          PER UNIT VOLUME
```

```
C     *      0 ENERGY IN UNITS OF ENERGY PER UNIT MASS. FOR EXAMPLE-
C     *        UNITS OF ERGS/GRAM WHEN KUNIT = 1.  THIS IS THE USUAL T.
C     *      1 ENERGY IN UNITS OF ENERGY PER UNIT VOLUME. FOR EXAPMLE-
C     *        UNITS OF ERGS/CM**3 WHEN KUNIT = 1. THIS IS THE ENERGY
C     *        DENSITY RHO*T COMMONLY COMPUTED IN HYDRODYNAMIC
C     *        COMPUTER CODES.
C     *
C     * IMATE = INDICATES WHETHER TO LOAD THE DATA FILE IF
C     *         IT DOES NOT EXIST FOR LMAT OR GO DIRECTLY TO A
C     *         PREVIOUSLY LOADED FILE.
C     *   =0 CHECK IF THE DATA FILE FOR LMAT HAS BEEN LOADED.
C     *      IF NOT, SEARCH FOR THE FILE, COMVERT IT TO THE PROPER UNITS
C     *      AND COPY IT INTO LCM USING THE INVERTED SESAME FORMAT.
C     *   >0 EQUAL TO THE LMAT TABLE NUMBER.  THIS NUMBER WAS RETURNED
C     *      BY THE PACKAGE ON A PREVIOUS CALL TO SUBROUTINE EOSORE,
C     *      EOSIPT OR EOSIRT.  THIS OPTION IS FASTER THAN IMATE = 0
C     *      BECAUSE IT SKIPS SEARCH IN THE DIRECTOR OF THE
C     *
C     * OUTPUT VARIABLES-
C     *
C     * P = PRESSURE
C     *
C     * E = INTERNAL ENERGY
C     *
C     * IMATE = INDICATES THE SUCCESS OR FAILURE OF
C     *         LOCATING AND LOADING THE DATA FILE FOR LMAT.
C     *
C     *       = N>0 MATERIAL TABLE NUMBER (SUCCESS)
C     *         0    MATERIAL (LMAT) NOT IN LIBRARY
C     *         -N  (N>1) INSUFFICIENT STORAGE
C     *             THE LCM STORAGE MUST BE INCREADED BY AT LEAST
C     *             N STORAGE LOCATIONS.  SEE THE EOSMOD MANUAL
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * SAMPLE DRIVER PROGRAM-
C     *
C     *     PROGRAM TST(OUTPUT)
C     *     DIMENSION P(3),T(3)
C     *     LMAT = "HELIUM"
C     *     R = 0.001
C     *     T = 300.0
C     *     KEOS = 110
C     *     IMATE = 0
C     *     CALL EOSIRT(LMAT,R,T,P,E,KEOS,IMATE)
C     *     PRINT 10,P(1)
C     *  10 FORMAT(" PRESSURE = ",E10.2," MICROBARS")
C     *     CALL EXIT
C     *     END
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ***************************************************************
```

```
C
C     COMMON BLOCKS FOR THE SESAME EOS ROUTINES
      LEVEL 2, TBLS
      COMMON /S2DIRX/ LCMX, NRS, LCFW(10,3)
      COMMON /SESOATX/ TBLS(11000)
      COMMON /SESINX/ DUM(4), KBR, DUM1
      COMMON /INTORDX/ KFN
C
C     COMMON BLOCKS FOR THE EOSMOO ROUTINES
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC3/ INIT, IRDIM, IR(60,3), KUP(60,3)
      COMMON /EOSC4/ NTABLE, NTABLO, IFLP, LCNT
C
      DIMENSION P(3), E(3)
      DATA KEOSS /-99/,KBRS/0/,KFNS/0/,LMATS/1H /
C
C     ****************************************************************
C
C     IOT=1 LOCATOR OF DATA TYPE FOR IR,GETINVX(., IDT,...)
C     FOR THE INVERTED SESAME FORMAT
      IOT=1
C
C     CHECK IF THE UNITS HAVE CHANGED SINCE THE LAST CALL
      IF(KEOSS.NE.KEOS.OR.LMAT.NE.LMATS) GO TO 5
      LMATS=LMAT
      KBR=KBRS
      KFN=KFNS
      GO TO 10
    5 CONTINUE
C
C     CHECK THE VALIDITY OF THE INPUT PARAMETERS
      CALL EOSKUT(KEOS,KBR,KUNIT,KREPE,KFN,KEOSS,KBRS,KFNS,IMATE,IOT
     1 ,IERR)
      IF(IERR.LT.0) GO TO 75
   10 CONTINUE
C
C     FIND THE MATERIAL
      IF (IMATE.GT.0) GO TO 60
      CALL EOSGET(LMAT,KUNIT,KREPE,IMATE,IDT,IERR)
      IF(IMATE.LE.0.OR.IERR.LT.0) GO TO 75
   60 CONTINUE
C
C     CALCULATE THE EQUATION OF STATE
      CALL T4RTPEX (IR(IMATE,1),1,TBLS,R,T,P,E,IERR)
C
C     PRINT AN ERROR MESSAGE IF T4RTPEX FAILED TO CONVERGE
      IF (IERR.EQ.0) WRITE(LOUT,80) LMAT,R,T
      IF (IERR.EQ.0) IMATE=0
C
   75 CONTINUE
      IFLP=IERR
      RETURN
C
   80 FORMAT(" FAILED TO CONVERGE WHEN ITERATING ON THE INVERTED",
     1 /," EOS TABLES IN SUBROUTINE T4PTREX CALLED BY EOSIRT",/,
     2 " THE REQUESTED VALUES FOR RHO, PRESSURE, TEMPERATURE AND ENERGY"
     2 ,/," MAY BE OUT OF RANGE OR NEAR THE EDGE OF THE TABLE",/,
     3 " CHECK THE EOSMOD WRITEUP FOR THE DATA RANGE",/,
     1  " LMAT =",A10,
     4 " DENSITY =",1PE12.4," TEMPERATURE =",1PE12.4)
      END
```

```
      SUBROUTINE EOSDRT (LMAT,R,T,O,KOPC,IMATO)
C
C*************************************************************
C     *
C     * PURPOSE-
C     *  GIVEN THE DENSITY (R) AND TEMPERATURE(T) OF A MATERIAL (LMAT)
C     *  THIS ROUTINE RETURNS THE OPACITY (O) USING THE LASL
C     *  HYDSES (T-4) ROUTINES
C     *                          .
C     * INPUT VARIABLES-
C     *
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     *  R = DENSITY (RHO)
C     *
C     *  T = TEMPERATURE
C     *
C     *  KOPC = MULTIPLE PARAMETER FLAG TO DESCRIBE HOW TO WRITE
C     *         AND RETRIEVE THE DATA FILE. KOPC HAS FOUR DECIMAL DIGITS,
C     *
C     *  KOPC = 100*KUNIT + 10*KREPE + KFN  WHERE
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,O-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,O-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,O-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,O-CM**2/G
C     *     4 (HYDROXO) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,O-CM**2/G
C     *     5 (SESAMEO) R-G/CC,T-EV,O-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,O-CM**2/GM,P-JRKS/CC,EO-JRKS/CC
C     *
C     *      LEGEND-
C     *             R = DENSITY
C     *             T = TEMPERATURE
C     *             O = OPACITY
C     *             P = PRESSURE
C     *             E = INTERNAL ENERGY
C     *
C     *             CC = CUBIC CENTIMETER
C     *             CM = CENTIMETER
C     *             DEG. K = DEGREES KELVIN
C     *             EV = ELECTRON VOLT
C     *             G = GRAM
C     *             GPA = GIGA PASCALS
C     *             J = JOULES
C     *             JRKS = JERKS
C     *             KEV = KILO ELECTRON VOLTS
C     *             KG = KILOGRAM
C     *             M = METER
C     *             MBR = MEGABAR
C     *             MUBR = MICROBAR
C     *             PA = PASCAL
C     *
C     *  KREPO = COMPUTATIONAL FLAG TO INDICATE WHICH REPRESENTATION
C     *          TO USE FOR THE OPACITY VARIABLE.
C     *        0 OPACITY REPRESENTED AS KAPPA IN DIMENSIONAL
C     *          UNITS OF LENGTH**2/MASS
C     *        1 OPACITY REPRESENTED AS A MEAN-FREE PATH,
C     *          LAMBDA = 1/(KAPPA*RHO), IN DIMENSIONAL
```

```
C     *           UNITS OF LENGTH.
C     *
C     *  KFN = KIND OF FUNTION INTERPOLATION IN THE TABLES
C     *    = 0 RATIONAL APPROXIMATIONS (ACCURATE)
C     *    = 1 BILINEAR APPROXIMATIONS (FAST)
C     *
C     * IMATO = INDICATES WHETHER TO LOAD THE DATA FILE IF
C     *         IT DOES NOT EXIST FOR LMAT OR GO DIRECTLY TO A
C     *         PREVIOUSLY LOADED FILE.
C     *   =0 CHECK IF THE DATA FILE FOR LMAT HAS BEEN LOADED.
C     *      IF NOT, SEARCH FOR THE FILE, CONVERT IT TO THE PROPER UNITS
C     *      AND COPY IT INTO LCM USING THE INVERTED SESAME FORMAT.
C     *   >0 EQUAL TO THE LMAT TABLE NUMBER.  THIS NUMBER WAS RETURNED
C     *      BY THE PACKAGE ON A PREVIOUS CALL TO SUBROUTINE EOSORT,
C     *      THIS OPTION IS FASTER THAN IMATO = 0
C     *      BECAUSE IT SKIPS SEARCH IN THE DIRECTOR OF THE
C     *
C     * OUTPUT VARIABLES-
C     *
C     *  D = OPACITY
C     *
C     * IMATO = INDICATES THE SUCCESS OR FAILURE OF
C     *         LOCATING AND LOADING THE DATA FILE FOR LMAT.
C     *
C     *        = N>0 MATERIAL TABLE NUMBER (SUCCESS)
C     *          0    MATERIAL (LMAT) NOT IN LIBRARY
C     *          -N  (N>1) INSUFFICIENT STORAGE
C     *              THE LCM STORAGE MUST BE INCREADED BY AT LEAST
C     *              N STORAGE LOCATIONS.  SEE THE EOSMOD MANUAL
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * SAMPLE DRIVER PROGRAM-
C     *
C     *      PROGRAM TST(OUTPUT)
C     *      DIMENSION P(3),E(3)
C     *      LMAT = "HELIUM"
C     *      R = 0.001
C     *      T = 300.0
C     *      KOPC = 500
C     *      IMATO = 0
C     *      CALL EOSORT(LMAT,R,T,D,KOPC,IMATO)
C     *      PRINT 10,P(1)
C     *   10 FORMAT(" OPACITY = ",E10.2)
C     *      CALL EXIT
C     *      END
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     *  QLOG10 = QUICK VERSION OF ALOG10
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7 LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1979
C     *
C     ****************************************************************
```

```
C
C     COMMON BLOCKS FOR THE SESAME EOS ROUTINES
C     DIMS TBLS,LCMX,NLBUF,LCFW(,),ZZ()
      LEVEL 2, TBLS
      COMMON /S2DIRX/ LCMX, NRS, LCFW(10,3)
      COMMON /SESOATX/ TBLS(11000)
      COMMON /INTORDX/ KFN
      COMMON /SESINX/ IRC, IOT, RHO, TEMP, KBR, IFL
      COMMON /SESOUTX/ OPACITY(3), PLANKO(3)
C
C     COMMON BLOCKS FOR THE EOSMOD ROUTINES
      COMMON /EOSCZ/ LOUT
      COMMON /EOSCCO/ TFACO, RFACO, OFACO, KREPO
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC3/ INIT, IROIM, IR(60,3), KUP(60,3)
      COMMON /EOSC4/ NTABLE, NTABLO, IFLP, LCNT
C
      DATA KOPCS /-99/, KFNS/0/,LMATS/1H /
C
C     ***************************************************************
C
C      CHECK POSITIVITY OF INPUT PARAMETERS
      IF((T.GT.0.0).AND.(R.GT.0.0)) GO TO 4
      IMATO=-1
C
      WRITE(LOUT,20)R,T
   20 FORMAT(" THE DENSITY =",1E12.4," OR THE TEMPERATURE ="
     1 ,E12.4," IS NONPOSITIVE")
C
      GO TO 75
    4 CONTINUE
C
C     SET INITIAL VARIABLES FOR OPACITY TABLES
C     IOT=2 LOCATOR OF DATA TYPE FOR IR AND SUB. GETPRX(,,IOT,,...)
      IDT=2
C
C     CHECK IF THE UNITS HAVE CHANGED SINCE THE LAST CALL
      IF(KOPCS.NE.KOPC.OR.LMAT.NE.LMATS) GO TO 5
      KFN=KFNS
      LMATS=LMAT
      KOPCS=KOPC
      GO TO 10
    5 CONTINUE
C
C     UNSCRAMBLE MULTIPLE FLAG KOPC
      CALL EOSKUT(KOPC,KDUMMY,KUNIT,KREPO,KFN,KOPCS,KBRS,KFNS,IMATO,IDT
     1 ,IERR)
      IF(IERR.LT.0) GO TO 75
   10 CONTINUE
C
C     FIND THE MATERIAL
      IF (IMATO.GT.0) GO TO 60
      CALL EOSGET(LMAT,KUNIT,KREPO,IMATO,IOT,IERR)
      IF(IMATO.LE.0.OR.IERR.LT.0) GO TO 75
C
   60 CONTINUE
C
C     TRANSFER INPUT CALL PARAMETERS TO COMMON BLOCK
      TEMP=QLOG10(T)
      RHO=QLOG10(R)
      KBR=1
      IRC=IR(IMATO,2)
```

```
187 C
188 C      CALCULATE THE EQUATION OF STATE
189        CALL T4OATX
190 C       T4OATIX OOES NOT RETURN AN ERROR FLAG
191 C      FOR DATA OUT OF BOUNDS
192 C
193 C      RESTORE OUTPUT VARIABLES FOR RETURN TO CALL
194        IFLP=IERR
195        O=10.**OPACITY(1)
196     75 CONTINUE
197        RETURN
198        END
```

INTERNAL SUBROUTINES

EOSBEG (initializes all the common block variables) 43
EOSCON (defines the table conversion factors) 46
EOSDSL (allows the user to scale the density) 49
EOSEFD (finds the EOS table) 51
EOSFAS (assigns the input files) 54
EOSGET (loads the Sesame EOS data files) 56
EOSKUT (checks the validity of KPARM) 59
EOSOFD (finds the opacity table) 61

```
      SUBROUTINE EOSBEG
C
C     ****************************************************************
C     *
C     * PURPOSE-
C     *     TO INITIALIZE ALL COMMON BLOCKS IN ONE PLACE OF CODE
C     *
C     * INPUT VARIABLES-
C     * NONE
C     *
C     * OUTPUT VARIABLES-
C     * ALL OUTPUT IS AT COMPILE TIME IN THE COMMON BLOCKS.
C     * THIS ALLOWS THE USER AN EASY WAY TO CHANGE THE VARIABLES
C     * BY SETTING THEM TO ANY OTHER VALUE AT EXECUTION
C     * TIME IN THEIR MAIN PROGRAM
C     *
C     * ---------------------------------------------------------
C     *
C     * LOCALLY DEFINED SESAME VARIABLES-
C     *  TBLS = ARRAY FOR STORAGE OF THE EOS TABLES
C     *  LCMX = LENGTH OF THE TBLS ARRAY
C     *  NRS = UPPER BOUND ON THE NUMBER OF MAT REGIONS LCFW(NRS,)
C     *  LCFW = ARRAY USED AS A DIRECTORY BY THE SESAME ROUTINES
C     *  IR = MATERIAL REGION NUMBER
C     *  IRC = IR (DEFINED TO PERMIT SUBROUTINE CALL
C     *  IDS2 = SESAME MATERIAL NUMBER
C     *  TBLS = NAME OF AN ARRAY DESIGNATED FOR THE STORAGE OF TABLES
C     *  LCNT = CRRRENT WORD IN THE ARRAY TBLS
C     *  LU41 = UNIT NUMBER ASSIGNED TO THE SESAME INPUT FILE SES2CL
C     *  LU42 = UNIT NUMBER ASSIGNED TO THE SESAME INPUT FILE SESAME
C     *  LU43 = UNIT NUMBER ASSIGNED TO SESAME/8 FILES
C     *  LU44 = UNIT NUMBER ASSIGNED TO MIXLIB (MIXTURES)
C     *  LU45 = UNIT NUMBER ASSIGNED TO MIXLIB DIRECTORY.
C     *  KFN = 0 RATIONAL APPRDXIMATIONS (ACCURATE)
C     *  KFN = 1 BILINEAR APPROXIMATIONS (FAST)
C     *  ZB (OUTPUT FROM GETINVX) AT. CHARGE,AT. CHG**2,MASS
C     *  IOT = DATA TYPE INDICATOR
C     *  MIO (IO) MATERIAL IO =1 INVERSE TABLES =2 OPACITY
C     *
C     * EXTERNAL FILES TO HANDLE EOS, OPACITIES AND MIXTURES
C     * SES2CL - CLASSIFIED SESAME LIBRARY
C     * SESAME - CUCLASSIFIED SESAME LIBRARY
C     *  SESAME - OPACITY TABLE FROM T4
C     *  MIXLIB - PRIVATE (EOS,OPC) TABLES CREATED BY MIXB(OR MIXER)
C     *  MIXDIR - DIRECTORY OF MIXTURES ON MIXLIB ( NAME,NO (A10.I3))
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     *  SESAME ROUTINES- S2GET,S2EDS
C     *  SESAME ROUTINES MATCHKX,TABRANX,INBUFRX,OPACKX,ISRCHKX.
C     *     T4INTPX,GETINVX,RATFN1X,T4DATIX,INV3D1X,T4RTPEX
C     *  SESAME COMMON BLOCKS-S2DIRX,RTBLK2X,SESDATX,SESINX,SESOUTX,INTORDX
C     *  EOSMOD COMMON BLOCKS- EOSC1,-7
C     *  EOSMOD COMMON (ALSO INSERTED INTO GETINVX) EOSCCE, EOSCCD
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT..LA-8502-M.1980
```

```
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ****************************************************************
C
C     COMMON BLOCKS FOR THE SESAME EOS ROUTINES
C     DIMS TBLS,LCMX,NLBUF,LCFW(,),ZZ()
C     MUST BE WATCHED IF CHANGING DIMENSIONS
C     APPEAR IN RTBLK2X,SESDATX,S2DIRX,DATA LCMX...
      LEVEL 2, TBLS
      COMMON /S2DIRX/ LCMX, NRS, LCFW(10,3)
      COMMON /SESDATX/ TBLS(11000)
      COMMON /SESINX/ DUM(4), KBR, DUM1
      COMMON /INTDROX/ KFN
C
C     COMMON BLOCKS FOR THE EOSMOD ROUTINES
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC2/ LF41, LF42, LF43, LF44, LF45
      COMMON /EOSC3/ INIT, IRDIM, IR(60,3), KUP(60,3)
      COMMON /EOSC4/ NTABLE, NTABLD, IFLP, LCNT
      COMMON /EOSC5/ NMAT, LABMAT(60), IDMAT(60), IMATEL
      COMMON /EOSC6/ NMCL, LABMCL(60), IDMCL(60)
      COMMON /EOSC7/ NMATO, LABMO(60), IDMATO(60), IMATOL
C
C     EOSMOD COMMON BLOCKS USED BY THE MODIFIED SESAME ROUTINES
      COMMON /EOSCCE/ TFACE, RFACE, PFACE, EFACE, KREPE
      COMMON /EOSCCO/ TFACO, RFACO, OFACO, KREPO
C
C     DESIGNATE THE OUTPUT FILE FOR THE ERROR MESSAGES
      DATA LOUT/"OUTPUT"/
C
      DATA LCMX /11000/, NRS /10/, LCNT /1/, LCFW /30*0/,
     1 IR/180*0/,NTABLE/1/,NTABLD/0/,INIT/0/,IRDIM/60/,KUP/180*(-1)/
C
      DATA LU41 /41/, LU42 /42/, LU43 /42/, LU44 /44/, LU45 /45/
      DATA LF41/6HSES2CL/,LF42/6HSESAME/,LF43/6HSESAME/,LF44/6HMIXLIB/
     1 ,LF45/6HMIXDIR/
C
C     INITIALIZE THE CONTENTS OF THE EOS TABLE SESAME
      DATA NMAT /32/
      DATA LABMAT/
     1 "ALLUVIUM","ALUMINUM","AL2O3","BERYLLIUM","BORON C",
     1 "BRASS","COPPER","DEUTERIUM","GOLD","GRANITE",
     2 "HELIUM","HE","IRON","IRON2","LEAD","6LIO","6LIH",
     3 "MOLY","NEON","NICKEL","PLATINUM","POLYE","POLYS","SIO2",
     4 "SODIUM","SS","STEAM","UO2","URANIUM","URETHANE","VERMICULIT",
     6 "WATER"/
      DATA IDMAT/7111,3710,7410,2020,7081,
     1 4100,3330,5263,2700,7390,
     2 5760,8180,2140,2145,3200,7240,7370,
     3 2980,5410,3100,3730,7170,7590,
     4 7380,2448,4270,7151,7432,1540,7560,7520,
     5 7150/
C
C     INITIALIZE THE CONTENTS OF THE CLASSIFIED EOS TABLE SES2CL
      DATA NMCL/0/
      DATA LABMCL/40*(1H )/
      DATA IDMCL/40*0/
C
C     INITIALIZE THE CONTENTS OF THE OPACITY TABLE SESAME
      DATA NMATO /27/
```

```
      DATA LABMO/"ALUMINUM","ARGON","BERYLLIUM","BORON","CALCIUM"
     1 ,"CARBON","CHLORINE","CHROMIUM","DEUTERIUM","HELIUM"
     2 ,"IRON","LITHIUM","MAGNESIUM","NITROGEN"
     3 ,"OXYGEN","PBX-9502","PHOSPHORUS","POTASSIUM","SIO2"
     4 ,"SILICON","SODIUM","SS","STAINLESS","SULPHUR"
     5 ,"TITANIUM","TITANIUM N","WATER"/
C
      DATA IOMATO/13710,15170,12020,12330,12030
     1 ,12180,15020,13070,15263,15760
     2 ,12140,12290,13080,15000
     3 ,15010,18200,13910,12460,17380
     4 ,13810,12448,14270,14270,14010
     5 ,12960,16000,17150/
C
      IMATEL=NMAT
      IMATOL=NMATO
C
      INIT=1
      RETURN
      END
```

```
      SUBROUTINE EOSCON(KUNIT,KREP,LMAT)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     *    TO PROVIDE THE APPROPRIATE  EOS SCALE FACTORS FOR THE
C     *       SYSTEM OF UNITS CHOSEN BY KUNIT (SEE BELOW)
C     *
C     * INPUT VARIABLES-
C     *    KUNIT = KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,O-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,O-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,O-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,O-CM**2/G
C     *     4 (HYOROXO) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,O-CM**2/G
C     *     5 (SESAMEO) R-G/CC,T-EV,O-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,O-CM**2/GM,P-JRKS/CC,EO-JRKS/CC
C     *
C     *      LEGEND-
C     *             R = OENSITY
C     *             T = TEMPERATURE
C     *             O = OPACITY
C     *             P = PRESSURE
C     *             E = INTERNAL ENERGY
C     *
C     *             CC = CUBIC CENTIMETER
C     *             CM = CENTIMETER
C     *             DEG. K = OEGREES KELVIN
C     *             EV = ELECTRON VOLT
C     *             G = GRAM
C     *             GPA = GIGA PASCALS
C     *             J = JOULES
C     *             JRKS = JERKS
C     *             KEV = KILO ELECTRON VOLTS
C     *             KG = KILOGRAM
C     *             M = METER
C     *             MBR = MEGABAR
C     *             MUBR = MICROBAR
C     *             PA = PASCAL
C     *
C     *
C     * OUTPUT VARIABLES- IN THE COMMON BLOCKS EOSCCE AND EOSCCO
C     *  TFACE = TEMPERATURE EOS SCALING FACTOR
C     *  RFACE = OENSITY EOS SCALING FACTOR
C     *  PFACE = PRESSURE EOS SCALING FACTOR
C     *  EFACE = ENERGY EOS SCALING FACTOR
C     *
C     *  TFACO = TEMPERATURE OPACITY SCALING FACTOR
C     *  RFACO = OENSITY OPACITY SCALING FACTOR
C     *  OFACO = OPACITY SCALING FACTOR
C     *
C     * LOCAL VARIABLES-
C     * NONE
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     * EOSMOO COMMON BLOCKS- EOSCCE,EOSCCO
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
```

```
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * OATE- MARCH 6, 1980
C     *
C     ****************************************************************
C
      COMMON /EOSCCE/ TFACE, RFACE, PFACE, EFACE, KREPE
      COMMON /EOSCCO/ TFACO, RFACO, OFACO, KREPO
C
      KREPE=KREP
      KREPO=KREP
C
C     OEFINE THE OEFAULE SESAME VALUES
C     THESE ARE THE UNITS THAT THE SESAME OATA FILES ARE WRITTEN IN
      TFACE=1.
      RFACE=1.
      PFACE=1.
      EFACE=1.
C
      TFACO=4.0646423
      RFACO=0.
      OFACO=0.
C
C     RESET THE SCALE FACTORS THAT ARE OIFFERENT FROM THE OEFAULT
      KP1=KUNIT+1
      GO TO (45,40,30,20,15,10,25), KP1
C
C     LASNEX UNITS
   25 PFACE=1.E-6
      EFACE=1.E-6
      TFACE=8.617346719E-8
      TFACEO=-3.
      GO TO 50
C
C     SESAME OPACITY UNITS
   10 TFACE=8.61703E-5
      TFACO=0.0
      GO TO 50
C
C     HYOROX OPACITY UNITS
   15 PFACE=.01
      EFACE=.01
      TFACE=8.61703E-8
      TFACO=-3.
      GO TO 50
C
C     HYOROX EOS UNITS
   20 PFACE=.01
      EFACE=.01
      GO TO 50
C
C     STANOARO INTERNATIONAL UNITS (SIU)
   30 RFACE=1.E+3
      PFACE=1.E+9
      EFACE=1.E+6
      RFACO=3.0
      OFACO=-1.0
      GO TO 50
C
C     CGS UNITS
```

```
125     40 PFACE=1.E+10
126        EFACE=1.E+10
127        GO TO 50
128 C
129 C      SESAME EOS UNITS
130     45 OFACO=2.0
131     50 CONTINUE
132 C
133 C      RESCALE THE TABLES IN THE USER PRESCRIBED SCALE FACTORS
134        DSFAC=1.0
135        CALL EOSOSL(LMAT,DSFAC)
136        EFACE=EFACE*OSFAC
137        RFACE=RFACE/OSFAC
138        RFACD=RFACO/DSFAC
139 C
140        RETURN
141        END
```

```
      SUBROUTINE EOSOSL(LMAT,OSFAC)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     * TO ALLOW A USER TO RESCALE THE MASS DENSITY IN
C     * THE EOS TABLES.  THIS IS A USEFUL ROUTINE TO APPROXIMATE THE
C     * EOS AND OPACITIES OF DIFFERENT ISOTOPES AND ISOTOPIC MIXTURES
C     * OF THE MATERIALS IN THE SESAME LIBRARY
C     *
C     * AN ALTERNATE PURPOSE IS TO ALLOW A USER TO DEFINE NEW
C     * SCALE FACTORS FOR THE UNITS THE TABLE IS TO BE WRITTEN IN
C     *
C     * INPUT VARIABLE-
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760' FOR HELIUM
C     *
C     * OUTPUT VARIABLE-
C     * OSFAC = DENSITY SCALE FACTOR EQUAL TO THE RATIO OF THE
C     *          ATOMIC MASSES OF THE MATERIALS. THAT IS,
C     *          OSFAC=ATOMIC MASS DENSITY OF THE SESAME MATERIAL)/
C     *               (ATOMIC MASS DENSITY OF THE DESIRED MATERIAL)
C     *
C     * FOR EXAMPLE- THE EOS OF A 60-40 MIXTURE OF DEUTERIUM-TRIDIUM
C     * CAN BE APPROXIMATED BY DEFINING OSFAC=2/(0.6*2+0.4*3)=0.833
C     * AND CALLING EOSMOO WITH LMAT="DEUTERIUM"
C     *
C     * INPUT-OUTPUT VARIABLES IN THE COMMON BLOCKS EOSCC2 AND EOSCCO
C     *  TFACE = TEMPERATURE EOS SCALING FACTOR
C     *  RFACE = DENSITY EOS SCALING FACTOR
C     *  PFACE = PRESSURE EOS SCALING FACTOR
C     *  EFACE = ENERGY EOS SCALING FACTOR
C     *
C     *  TFACO = TEMPERATURE OPACITY SCALING FACTOR
C     *  RFACO = DENSITY OPACITY SCALING FACTOR
C     *  OFACO = OPACITY SCALING FACTOR
C     *
C     * FOR FURTHER INFORMATION ON THESE FACTORS SEE SUBROUTINE EOSCON
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     * EOSMOO COMMON BLOCKS- EOSCCE, EOSCCO
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOO PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOO- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ***************************************************************
C
      COMMON /EOSCCE/ TFACE, RFACE, PFACE, EFACE, KREPE
      COMMON /EOSCCO/ TFACO, RFACO, OFACO, KREPO
C
C     DUMMY SUBROUTINE FOR THE PACKAGE.
C     IF A USER SUPPLIES THIS ROUTINE THEN THE FACTORS CAN BE
```

```
C     BE RESET AS DESCRIBED IN THE MANUAL WHEN THE TABLES ARE
C     WRITTEN, OR A DESITY SCALE FACTOR CAN BE INCLUDED
C     TO CHANGE THE DENSITY TABLES BY A CONSTANT FACTOR.
C
C     FOR EXAMPLE, FOR A 60-40 MIXTURE OF DEUTERIUM-TRIDIUM
C     THE CODE COULD BE WRITTEN AS-
C     IF(LMAT.EQ."DEUTERIUM") DSFAC=0.833
C
      RETURN
      END
```

```
      SUBROUTINE EOSEFO (LMAT,IO,IMATE)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     *   TO LOCATE EOS MATERIAL DEFINED BY HOLLERITH NAME
C     *   IN APPROPRIATE FILE AND ASSIGN IT A SESAME (OR
C     *     PRIVATE IO) EOS NUMBERUMBER
C     *
C     * INPUT VARIABLES-
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     * OUTPUT VARIABLES-
C     *  IO = SESAME OR PRIVATE IO NUMBER AS STORED IN LIBRARY
C     *    SESAME ( EOS LIBRARY)
C     *  IMATE  = LOCATION OF MATERIAL IN ARRAY LABMAT+ MIXOIR
C     *      PROVIDED MATERIAL HAS BEEN LOCATED
C     *         = 0 IF MATERIAL HAS NOT BEEN LOCATED BY ROUTINE
C     *      UPPER BOUND ON IMATE IS 60 (DIM IR(,))
C     *
C     * LOCAL VARIABLES-
C     *  ICFASE = 1 IF THE PUBLIC EOS FILES HAVE BEEN ASSIGNED
C     *  ICFASCL = 1 IF THE CLASSIFIED EOS FILES HAVE BEEN ASSIGNED
C     *  ICFASP = 1 IF THE PRIVATE EOS FILES HAVE BEEN ASSIGNED
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     * EOSMOO COMMON BLOCKS- EOSC1,2,3,5,6
C     * FTN ROUTINES- ENCODE, EOF
C     * LASL T-4 HYOSES ROUTINE- EOSFAS
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOO PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOO- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ***************************************************************
C
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC2/ LF41, LF42, LF43, LF44, LF45
      COMMON /EOSC3/ INIT, IROIM, IR(60,3), KUP(60,3)
      COMMON /EOSC5/ NMAT, LABMAT(60), IOMAT(60), IMATEL
      COMMON /EOSC6/ NMCL, LABMCL(60), IOMCL(60)
C
      OATA ICFASE/0/,ICFASCL/0/,ICFASP/0/,IOCNT/1/
C
C     CHECK IF THE MATERIAL IS IN THE STANDARO SESAME LIST
      OO 10 IMATE=1,IMATEL
      IF (LMAT.EQ.LABMAT(IMATE)) GO TO 40
   10 CONTINUE
C
C     CHECK IF THE MATERIAL IS IN THE CLASSIFIED EOS SESAME LIST
      OO 12 IMATE=1,NMCL
```

```
      IF (LMAT.EQ.LABMCL(IMATE)) GO TO 45
   12 CONTINUE
C
C     *** ASSIGN EOS PRIVATE FILES TO PROGRAM IF AVAILABLE
      IF(LU45.EQ.4HNONE) GO TO 31
      IMATE=NMAT
      IF(ICFASP .GT. 0 ) GO TO 15
      ICFASP=1
      CALL EOSFAS(3)
   15 CONTINUE
C
C     THE CURRENT PRIVATE LIST DIRECTORY (MIXDIR) IS ON UNIT LU45
C     CHECK IF LF45 EXISTS IN LOCAL FILE SPACE
      CALL FEXIST(LF45,IFFLAG)
      IF(IFFLAG .EQ. 0 ) GO TO 30
C      IFFLAG = 0 FILE NOT IN LOCAL FILE SPACE
C      IFFLAG = 1 FILE LOCAL
C
      REWIND LU45
   20 READ (LU45,80) LABEL,ID
      IMATE=IMATE+1
      IF (LMAT.EQ.LABEL) GO TO 50
      IF (EOF(LU45)) 30,20
   30 CONTINUE
   31 CONTINUE
C
C     CHECK IF THE LMAT IS A SESAME NUMBER
      IMAT1=AND(SHIFT(LMAT,6),77B)
CRAY CODE IMAT1=AND(SHIFT(LMAT,8),377B)
C
      IF(IMAT1.LT.20B) GO TO 35
      IF(IMAT1.GT.31B) GO TO 35
C
      DECODE(10,32,LMAT) ID
   32 FORMAT(I4)
C
      DO 33 IMATE=1,IMATEL
      IF(ID.EQ.IDMAT(IMATE)) GO TO 40
   33 CONTINUE
C
      IMATEL=IMATEL+1
      IMATE=IMATEL
      LABMAT(IMATE)=LMAT
      IDMAT(IMATE)=ID
C
C     ASSUME THE MATERIAL IS IN THE STANDARD SESAME LIST
C     IF IT IS NOT, A NONFATAL ERROR WILL OCCUR AT A LATER STEP
      GO TO 40
C
C     THE MATERIAL WAS NOT FOUND. PRINT AN ERROR MESSAGE
   35 IMATE=0
      WRITE(LOUT,75) LMAT
      GO TO 999
C
   40 ID=IDMAT(IMATE)
C
C     *** ASSIGN EOS FILES TO PROGRAM
      IF(ICFASE .GT. 0 ) GO TO 44
      ICFASE=1
      CALL EOSFAS(1)
   44 CONTINUE
C
```

```
      GO TO 999
C
   45 IO=IOMCL(IMATE)
C
C     *** ASSIGN CLASSIFIED EOS FILES TO PROGRAM
      IF(ICFASCL .GT. 0 ) GO TO 46
      ICFASCL=1
      CALL EOSFAS(4)
   46 CONTINUE
C
   50 CONTINUE
      IF (IMATE.LE.IROIM) GO TO 60
      IMATE=0
      WRITE(LOUT,90) LMAT
   60 CONTINUE
C
  999 CONTINUE
      RETURN
C
   75 FORMAT (" MATERIAL LMAT = ",A10," NOT FOUND")
   80 FORMAT (A10,I3)
   90 FORMAT ("  IMATE EXCEEDS UPPER BOUND IN SUBROUTINE EOSEFD "
     1 ,/," FOR MATERIAL LMAT=",A10)
      END
```

```
      SUBROUTINE EOSFAS(KTABLE)
C
C     ************************************************************
C     *
C     * PURPOSE-
C     * TO ASSIGN FILE NAMES TO THE EOS AND OPACITY DATA FILES
C     *
C     * INPUT VARIABLES-
C     * KTABLE = 1 EOS TABLE
C     *          2 OPACITY TABLE
C     *          3 PRIVATE TABLES
C     *          4 SES2CL TABLES
C     *
C     * OUTPUT VARIABLES-
C     * NONE
C     *
C     * LOCAL VARIABLES-
C     * INITE, INITECL,INITO AND INITP ARE SET TO 1 AFTER THE EOS, OPACITY
C     * AND PRIVATE FILES HAVE BEEN INITIALIZED
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     * EOSMOD COMMON BLOCKS- EOSC1, EOSC2
C     * FTN SUBROUTINES- QASSIGN, ASSIGN
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ************************************************************
C
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC2/ LF41, LF42, LF43, LF44, LF45
      DATA INITE/0/, INITO/0/, INITP/0/, INITECL/0/
C
      GO TO (10,20,30,40), KTABLE
C
C     EOS TABLE ASSIGNMENT CODING
   10 CONTINUE
      IF(INITE.NE.0) GO TO 999
      IF((INITO.NE.0).AND.(LF42.EQ.LF43)) GO TO 999
      INITE=1
      CALL QASSIGN (LU42,LF42,0,0)
      GO TO 999
C
C     OPACITY TABLE ASSIGNMENT CODING
   20 CONTINUE
      IF(INITO.NE.0) GO TO 999
      IF((INITE.NE.0).AND.(LF42.EQ.LF43)) GO TO 999
      INITO=1
      CALL QASSIGN (LU43,LF43,0,0)
      GO TO 999
C
C     PRIVATE TABLES
   30 CONTINUE
```

```
      IF(INITP.NE.0) GO TO 999
      INITP=1
      CALL QASSIGN (LU44,LF44,0,0)
      CALL ASSIGN (LU45,LF45,4000B)
      GO TO 999
C
C     CLASSIFIED EOS TABLE ASSIGNMENT CODING
   40 CONTINUE
      IF(INITECL.NE.0) GO TO 999
      INITECL=1
      CALL QASSIGN (LU41,LF41,0,0)
      GO TO 999
C
  999 RETURN
      END
```

```
      SUBROUTINE EOSGET(LMAT,KUNIT,KREP,IMAT,IOT,IERR)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     * LOAD THE SESAME EOS DATA FILES
C     *
C     * INPUT VARIABLES-
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,D-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,D-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,D-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,D-CM**2/G
C     *     4 (HYDROXD) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,D-CM**2/G
C     *     5 (SESAMED) R-G/CC,T-EV,D-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,D-CM**2/GM,P-JRKS/CC,ED-JRKS/CC
C     *
C     * IMAT = INDICATES WHETHER TO LOAD THE DATA FILE IF
C     *         IT DOES NOT EXIST FOR LMAT OR GO DIRECTLY TO A
C     *         PREVIOUSLY LOADED FILE.
C     *   =0 CHECK IF THE DATA FILE FOR LMAT HAS BEEN LOADED.
C     *      IF NOT, SEARCH FOR THE FILE, CONVERT IT TO THE PROPER UNITS
C     *      AND COPY IT INTO LCM .
C     *   >0 EQUAL TO THE LMAT TABLE NUMBER.  THIS NUMBER WAS RETURNED
C     *      BY THE PACKAGE ON A PREVIOUS CALL TO SUBROUTINE EOSORE,
C     *      EOSIPT OR EOSIRT.  THIS OPTION IS FASTER THAN IMAT = 0
C     *      BECAUSE IT SKIPS SEARCH IN THE DIRECTOR OF THE
C     *
C     * OUTPUT VARIABLES-
C     * IERR = 0 SUCCESSFULL
C     *      .NE.0 UNSUCCESSFUL
C     *
C     * TBLS = LCM FILE SPACE WHERE THE EOS TABLES ARE WRITTEN
C     *
C     * LOCAL VARIABLES-
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     *  EOSMOD COMMON BLOCKS- EOSC1,EOSC3
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ***************************************************************
C
C     LOAD THE SESAME EOS TABLES IN THE INVERTED FORMAT
C
C     COMMON BLOCKS FOR THE SESAME EOS ROUTINES
      LEVEL 2, TBLS
      COMMON /S2OIRX/ LCMX, NRS, LCFW(10,3)
```

```
      COMMON /SESOATX/ TBLS(11000)
      COMMON /SESINX/ OUM(4), KBR, OUM1
      COMMON /INTOROX/ KFN
C
C     COMMON BLOCKS FOR THE EOSMOD ROUTINES
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC3/ INIT, IRDIM, IR(60,3), KUP(60,3)
      COMMON /EOSC4/ NTABLE, NTABLD, IFLP, LCNT
C
      DIMENSION ZB(3)
C
      IERR=0
C
C     CALL THE FILE ASSIGNMENT ROUTINE TO ASSIGN READ AND WRITE
C     UNIT NUMBERS TO THE INPUT DATA FILES
      IF(IOT.NE.2) CALL EOSEFD (LMAT,ID,IMAT)
      IF(IOT.EQ.2) CALL EOSOFD (LMAT,ID,IMAT)
C
      IF (IMAT.LE.O) IERR=-1
      IF (IMAT.LE.O) GO TO 75
C
C     CHECK IF THE TABLES HAVE BEEN INITIALIZED
      IF (IR(IMAT,IOT).GT.O) GO TO 70
C
C     CONVERT TABLES TO APPROPRIATE UNITS
      CALL EOSCON(KUNIT,KREP,LMAT)
C
      MIXTST=ID/1000
C
      GO TO (10,20,30),IDT
C
C     LOAD THE EOS TABLES IN THE INVERTED FORMAT (IOT=1)
   10 CONTINUE
      IF(MIXTST.NE.O)CALL GETINVX (NTABLE,ID,IOT,TBLS,LCNT,LU42,IERR,ZB)
      IF(MIXTST.EQ.O)CALL GETINVX (NTABLE,ID,IOT,TBLS,LCNT,LU44,IERR,ZB)
      GO TO 40
C
C     LOAD THE OPACITY TABLES (IOT=2)
   20 CONTINUE
C
      IF (MIXTST.EQ.O) CALL GETRPOX (NTABLO,ID,IOT,TBLS,LCNT,LU44,IERR)
      IF (MIXTST.NE.O) CALL GETRPOX (NTABLO,ID,IOT,TBLS,LCNT,LU43,IERR)
      GO TO 40
C
C     LOAD THE EOS TABLES IN THE STANDARD FORMAT (IDT=3)
   30 CONTINUE
      IF(MIXTST.NE.O)CALL GETEOSX (NTABLE,ID,IOT,TBLS,LCNT,LU42,IERR,ZB)
      IF(MIXTST.EQ.O)CALL GETEOSX (NTABLE,ID,IOT,TBLS,LCNT,LU44,IERR,ZB)
   40 CONTINUE
C
C     IERR RETURNS NEGATIVE IF THERE IS AN INITIALIZATION ERROR
      IF (IERR.GT.O) GO TO 60
      IF(IERR.LT.O) WRITE(LOUT,90) IERR
      IF(IERR.EQ.O) WRITE(LOUT,100)LMAT
      IF(IERR.EQ.O) IERR=-2
      GO TO 75
C
C     THE TABLES HAS BEEN LOADED, SET THE EOSMOD VARIABLES
   60 IERR=0
      IF(IOT.NE.2) GO TO 65
C
```

```
C      OPACITY TABLE UPOATE
       IR(IMAT,IOT)=NTABLO
       KUP(IMAT,IDT)=10*KUNIT+KREP
       NTABLO=NTABLO+1
       GO TO 70
C
C      EOS TABLE UPOATE
   65  CONTINUE
       IR(IMAT,IOT)=NTABLE
       KUP(IMAT,IOT)=10*KUNIT+KREP
       NTABLE=NTABLE+1
   70  CONTINUE
C
C      CHECK IF THE UNITS ARE VALID
       IF(10*KUNIT+KREP.NE.KUP(IMAT,IOT)) IERR=-6
C
       IF(IERR.EQ.-6) WRITE(LOUT,80)KUNIT,KREP,KUP(IMAT,3)
   80  FORMAT(" THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE"
     2,/," THE CURRENT VALUES OF KUNIT AND KREP ARE ",2I4
     3,/," THE PREVIOUS VALUES OF KUNIT AND KREP WERE",I5)
C
   75  CONTINUE
       IF(IERR.LT.0) IMAT=IERR
C
       RETURN
C
   90  FORMAT(" INSUFFICIANT STORAGE IN LCM"
     1 ,/," LACK",I10," WOROS")
  100  FORMAT(" UNABLE TO LOCATE MATERIAL ",A10,/,
     1 " IN SUBROUTINE EOSGET")
       END
```

```
      SUBRDUTINE EOSKUT(KPARM,KBR,KUNIT,KREP,KFN,KEOSS,KBRS,KFNS,IMAT,
     2 IOT,IERR)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     * TO CHECK THE VALIDITY OF THE INPUT VALUE FOR KPARM
C     * AND TO SEPARATE OUT THE INTERNAL PARTS
C     *
C     * INPUT VARIABLES-
C     *  KPARM = MULTIPLE PARAMETER FLAG TO DESCRIBE HOW TO WRITE
C     *       AND RETRIEVE THE DATA FILE. KPARM HAS FOUR DECIMAL DIGITS,
C     *
C     *  KPARM = 1000*KBR + 100*KUNIT + 10*KREP + KFN  WHERE
C     *
C     * OUTPUT VARIABLES-
C     *   KBR = COMPUTATIONAL MODE FLAG TO INDICATE WHICH
C     *         QUANTITIES AND THEIR PARTIAL DERIVATIVES ARE TO
C     *         BE CALCULATED AND RETURNED BY THE PACKAGE.
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,D-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,D-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,D-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,D-CM**2/G
C     *     4 (HYDRDXD) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,D-CM**2/G
C     *     5 (SESAMED) R-G/CC,T-EV,D-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,D-CM**2/GM,P-JRKS/CC,ED-JRKS/CC
C     *
C     * IF KREP REFERS TO EOS TABLE UNITS
C     *  KREP = COMPUTATION FLAG TO INDICATE WHETHER E IS
C     *         IS TO BE REPRESENTED AS ENERGY PER UNIT MASS DR ENERGY
C     *         PER UNIT VOLUME
C     *       0 ENERGY IN UNITS OF ENERGY PER UNIT MASS. FOR EXAMPLE-
C     *         UNITS OF ERGS/GRAM WHEN KUNIT = 1.  THIS IS THE USUAL E.
C     *       1 ENERGY IN UNITS OF ENERGY PER UNIT VOLUME. FOR EXAPMLE-
C     *         UNITS OF ERGS/CM**3 WHEN KUNIT = 1. THIS IS THE ENERGY
C     *         DENSITY RHO*E COMMONLY COMPUTED IN HYDRODYNAMIC
C     *         COMPUTER CODES.
C     *
C     * IF KREP (KREPO) REFERS TO OPACITY TABLE UNITS
C     *  KREPO = COMPUTATIONAL FLAG TO INDICATE WHICH REPRESENTATION
C     *          TO USE FOR THE OPACITY VARIABLE.
C     *         0 OPACITY REPRESENTED AS KAPPA IN DIMENSIONAL
C     *           UNITS OF LENGTH**2/MASS
C     *         1 OPACITY REPRESENTED AS A MEAN-FREE PATH.
C     *           LAMBDA = 1/(KAPPA*RHO), IN DIMENSIONAL
C     *           UNITS OF LENGTH.
C     *
C     *  KFN = KIND OF FUNTION INTERPOLATION IN THE TABLES
C     *    = 0 RATIONAL APPRDXIMATIONS (ACCURATE)
C     *    = 1 BILINEAR APPROXIMATIONS (FAST)
C     *
C     * LOCAL VARIABLES-
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
```

```
C     *            EOSMOD- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ****************************************************************
C
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC3/ INIT, IROIM, IR(60,3), KUP(60,3)
C
      IERR=0
C
      IF(INIT.EQ.0) CALL EOSBEG
C
C     UNSCRAMBLE MULTIPLE FLAG KPARM
C
      KBR=KPARM/1000
      ITEMP=KPARM-KBR*1000
C
      KUNIT=ITEMP/100
      ITEMP=ITEMP-KUNIT*100
C
      KREP=ITEMP/10
      KFN=ITEMP-10*KREP
C
      KEOSS=KPARM
      KFNS=KFN
      KBRS=KBR
C
C     CHECK IF KPARM IS A VALID INPUT PARAMATER
      IF(KPARM.LT.0) IERR=-2
      IF(KBR.GT.2) IERR=-2
      IF(KUNIT.GT.6) IERR=-2
      IF(KREP.GT.1) IERR=-2
      IF(KFN.GT.1) IERR=-2
C
C     PRINT AN ERROR MESSAGE IF KUNIT IS NOT VALID
      IF(IERR.LT.0) WRITE(LOUT,10)KPARM,KBR,KUNIT,KREP,KFN
   10 FORMAT(" ERROR DETECTED IN KPARM VALUE IN SUBROUTINE EOSKUT"
     1 ,/," KPARM=",I5," KBR=",I5," KUNIT=",I5," KREP=",I5," KFN=",I5)
C
C     CHECK IF THE UNITS HAVE CHANGED
      IF(IMAT.LE.0) GO TO 90
      IF(10*KUNIT+KREP.NE.KUP(IMAT,IOT)) IERR=-6
      IF (IERR.GE.0) GO TO 90
C
      WRITE(LOUT,30)KUNIT,KREP,KUP(IMAT,IOT)
   30 FORMAT(" THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE"
     2,/," THE CURRENT VALUES OF KUNIT AND KREP ARE ",2I2
     3,/," THE PREVIOUS VALUES OF KUNIT AND KREP WERE",I5)
C
   90 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE EOSOFD (LMAT,IO,IMATO)
C
C     ****************************************************************
C     *
C     * PURPOSE-
C     * TO OBTAIN MATERIAL NUMBERS FOR OPACITY TABLES
C     *
C     * INPUT VARIABLES-
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     * OUTPUT VARIABLES-
C     *  IO= SESAME OR MIXTUREID NUMBER AS STORED
C     *       IN SESAME OR MIXOIR(MIXLIB)
C     *  IMATO = LOCATION OF MATERIAL IN ARRAY LABMO + MIXOIR
C     *       PROVIDED MATERIAL HAS BEEN LOCATED
C     *        = 0 IF MATERIAL HAS NOT BEEN LOCATED
C     *
C     * LOCAL VARIABLES-
C     *  EOSMOO COMMON BLOCKS- EOSC1,EOSC3
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOO- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * DATE- MARCH 6, 1980
C     *
C     ****************************************************************
C
      COMMON /EOSCZ/ LOUT
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC2/ LF41, LF42, LF43, LF44, LF45
      COMMON /EOSC3/ INIT, IROIM, IR(60,3), KUP(60,3)
      COMMON /EOSC7/ NMATO, LABMO(60), IOMATO(60), IMATOL
C
      DATA ICFASO,ICFASP/2*0/
C      **** ASSIGN OPACITY FILES TO PROGRAM
      IF(ICFASO .GT. 0) GO  TO 5
      ICFASO=1
      CALL EOSFAS(2)
    5 CONTINUE
C
C     CHECK IF THE MATERIAL IS IN THE STANDARD SESAME TABLES
      DO 10 IMATO=1,IMATOL
      IF (LMAT.EQ.LABMO(IMATO)) GO TO 40
   10 CONTINUE
      IMATO=IMATOL
C
C     *** ASSIGN PRIVATE OPACITY FILES TO PROGRAM
      IF(ICFASP .GT. 0) GO  TO 15
      ICFASP=1
      CALL EOSFAS(3)
   15 CONTINUE
C
C     THE CURRENT PRIVATE LIST DIRECTORY (MIXOIR) IS ON UNIT LU45
```

```
C       CHECK IF LF45 EXISTS IN LOCAL FILE SPACE
        CALL FEXIST(LF45,IFFLAG)
        IF(IFFLAG .EQ. 0 ) GO TO 30
C       IFFLAG = 0 FILE NOT IN LOCAL FILE SPACE
C       IFFLAG = 1 FILE LOCAL
        REWIND LU45
   20   READ (LU45,80) LABEL,IO
        IMATO=IMATO+1
        IF (LMAT.EQ.LABEL) GO TO 50
        IF (EOF(LU45)) 30,20
   30   CONTINUE
C
C       CHECK IF THE LMAT IS A SESAME NUMBER
        IMAT1=AND(SHIFT(LMAT,6),77B)
CRAY CODE IMAT1=AND(SHIFT(LMAT,8),377B)
C
        IF(IMAT1.LT.20B) GO TO 35
        IF(IMAT1.GT.31B) GO TO 35
C
        DECODE(10,32,LMAT) IO
   32   FORMAT(I5)
C
        DO 33 IMATO=1,IMATOL
        IF(IO.EQ.IOMATO(IMATO)) GO TO 40
   33   CONTINUE
C
        IMATOL=IMATOL+1
        IMATO=IMATOL
        LABMO(IMATO)=LMAT
        IOMATO(IMATO)=IO
C
C       ASSUME THE MATERIAL IS IN THE STANDARD SESAME LIST
C       IF IT IS NOT, A NONFATAL ERROR WILL OCCUR AT A LATER STEP
        GO TO 40
C
C
C       THE MATERIAL WAS NOT FOUND. PRINT AN ERROR MESSAGE
   35   CONTINUE
        IMATO=0
        WRITE(LOUT,70) LMAT
        GO TO 999
C
   40   IO=IOMATO(IMATO)
   50   CONTINUE
C
        IF (IMATO.LE.IROIM) GO TO 60
        IMATO=0
        WRITE(LOUT,90) LMAT
   60   CONTINUE
C
  999   CONTINUE
        RETURN
C
C****** IMPROVE THESE DIAGONISTICS*****************
   70   FORMAT (" MATERIAL LMAT = ",A10," NOT FOUND")
   80   FORMAT (A10,I3)
   90   FORMAT ("  IMATO EXCEEDS UPPER BOUND IN SUBROUTINE EOSOFO "
     1  ,/," FOR MATERIAL LMAT=",A10)
        END
```

T-4 SUBROUTINES USED BY EOSMOD

DPACKX (packs real numbers) 64
GETEOSX (loads the total EOS tables) 65
GETINVX (gets inverted EOS tables) 67
GETRPOX (loads Rosseland/Planck opacity tables) 69
INBUFRX (sequential read) 71
INV301X (inverts a 301 table) 72
ISRCHKX (index search) 74
MATCHKX (checks if material table is loaded) 75
RATFN1X (one-dimensional interpolator) 76
TABRANX (fetches a given table for a given material from a Sesame II library) 78
T4DATIX (interpolates the inverted tables) 80
T4DATX (search/interpolate for atomic data as functions of region, density, and temperature) 83
T4INTPX (interpolates for a function z(x,y) and its derivatives) 86
T4PTREX (computes R and E from inverted table) 89
T4RTPEX (computes P and E from inverted table) 91

```
      FUNCTION OPACKX(A,B)
C-----------------------------------------------------------------------
C
C  FUNCTION  OPACKX
C
C  PURPOSE   TO OOUBLE PACK ARGUMENTS A AND B INTO A SINGLE WORO
C
C  REMARKS   SYSTEM OEPENOENT SHIFT FUNCTION
C
C  PROGRAMMER  J.ABOALLAH,JR.
C
C  OATE        1 MAY 1979
C
C-----------------------------------------------------------------------
      EQUIVALENCE (I1,X1),(I2,X2)
      OATA MASK/7777777770000000000B/
      X1=A
      X2=B
      I1=I1.AND.MASK
      I2=I2.AND.MASK
      I2=SHIFT(I2,30)
      I1=I1.OR.I2
      OPACKX=X1
      RETURN
      END
```

```
      SUBROUTINE GETEOSX(IR,MID,IOT,TBLS,LCNT,LU,IFL,ZB)
C*************************************************************
C
C  SUBROUTINE  GETEOSX(IR,MID,IOT,TBLS.LCNT,LU,IFL,ZB)
C
C  PURPOSE     TO LOAD THE TOTAL EOS TABLES
C
C  ARGUMENTS   IR        (INPUT)   REGION NO.
C              MID       (INPUT)   MATERIAL ID.
C              IOT       (INPUT)   DATA TYPE INDICATOR
C              TBLS      (OUTPUT)  ARRAY FOR TABLE STORAGE
C              LCNT      (I/O)     POSITION IN ARRAY FOR STORING TABLES
C              LU        (INPUT)   SESAME LIBRARY UNIT NUMBER
C              IFL       (OUTPUT)  ERROR FLAG
C                                  = 2 FOR MATERIAL ALREADY LOADED
C                                  = 1 FOR SUCCESSFUL LOADING
C                                  = 0 FOR DATA NOT FOUND
C                                  = - NO. OF EXTRA WORDS NEEDED FOR
C                                      STORAGE
C              ZB        (OUTPUT) ATOMIC CHARGE,CHARGE**2,AND MASS
C                                  ZB(1) = Z
C                                  ZB(2) = Z**2
C                                  ZB(3) = A
C
C
C  REMARKS     THIS IS THE LASNEX VERSION OF GETEOS
C
C              PRESSURES AND ENERGIES ARE DOUBLE PACKED
C
C              ENERGY DENSITIES ARE PER UNIT VOLUME (NOT MASS)
C
C  EXTERNALS   MATCHKX,TABRANX,DPACKX
C
C  PROGRAMMER  R.C. ALBERS, T-4
C
C  DATE        25 APRIL 79
C
C*************************************************************
      LEVEL 2,TBLS
      DIMENSION TBLS(1),ZB(3)
C     REPLACE FOLLOWING LINE BY USER COMMON BLOCKS
      COMMON/S201RX/LCMX,NRS,LCFW(10,3)
      COMMON /EOSCCE/ TFACE, RFACE, PFACE, EFACE, KREPE
C
C     CHECK TO SEE IF TABLE HAS BEEN LOADED
      CALL MATCHKX(MID,NRS,LCFW(1,IOT),TBLS(1),IFLG)
      IF(IFLG.EQ.0) GO TO 10
      LCFW(IR,IOT) = IFLG
      IFL=2
      RETURN
   10 NLEFT = LCMX - LCNT - 1
C
C     FETCH THE 201 TABLE
      CALL TABRANX(MID,201.,LU,TBLS(LCNT+2),NLEFT,IFL)
      IF(IFL.LE.0) RETURN
      ZB(1) = TBLS(LCNT+2)
      ZB(2) = ZB(1)*ZB(1)
      ZB(3) = TBLS(LCNT+3)
      RHOO=TBLS(LCNT+4)
C
C     FETCH THE 301 TABLE
      CALL TABRANX(MID,301.,LU.TBLS(LCNT+2),NLEFT,IFL)
```

```
      IF(IFL.LE.0) RETURN
C
C     CONVERT TO LASNEX UNITS AND DOUBLE PACK
      NR = TBLS(LCNT+2)
      NT = TBLS(LCNT+3)
      DO 20 I=1,NT
      LOCT = I + (LCNT + NR + 3)
   20 TBLS(LOCT)=TFACE*TBLS(LOCT)
      NWDS = NR*NT
      DO 30 J=1,NR
      RHO = TBLS(J + LCNT + 3)*RFACE
      TBLS(J+LCNT+3)=RHO
      DO 40 I=1,NT
      LOCP = (I-1)*NR + J + (NT + NR + LCNT + 3)
      LOCE = LOCP + NWDS
      PTEM=TBLS(LOCP)*PFACE
      ETEM=TBLS(LOCE)*EFACE
      IF(KREPE.EQ.1)ETEM=ETEM*RHO
      TBLS(LOCP) = DPACKX(PTEM,ETEM)
   40 CONTINUE
   30 CONTINUE
C
C     RESET INPUT PARAMETERS AND END
      TBLS(LCNT)=FLOAT(MIO)
      TBLS(LCNT+1)=RHOO
      LCFW(IR,IOT)=LCNT
      LCNT = LCNT + 2 + IFL - NWDS
      IFL = 1
      RETURN
      END
```

```
      SUBROUTINE GETINVX(IR,MIO,IOT,TBLS,LCNT,LU,IFL,ZB)
C-----------------------------------------------------------------------
C
C  SUBROUTINE  GETINVX(IR,MID.IDT,TBLS,LCNT,LU.IFL,ZB)
C
C  PURPOSE     TO LOAD INVERTED (ENERGY BASED) SESAME II
C              EOS TABLES
C
C  ARGUMENTS   IR         (INPUT)   REGION NO.
C              MID        (INPUT)   SESAME MATERIAL ID
C              IOT        (INPUT)   DATA TYPE INDICATOR
C              TBLS       (INPUT)   TABLE STORAGE ARRAY
C              LCNT       (IN/OUT)  POSITION IN ARRAY FOR STORING TABLES
C              LU         (INPUT)   SESAME LIBRARY UNIT NO.
C              IFL        (OUTPUT)  ERROR FLAG
C                                    2=MATERIAL ALREADY LOADED
C                                    1=SUCCESSFUL LOADING
C                                    0=DATA NOT FOUND
C                                    LT.0 FOR - THE NO. OF EXTRA WORDS
C                                    NEEDED FOR LOADING
C              ZB         (OUTPUT)  ATOMIC CHARGE,CHARGE**2,AND MASS
C                                    ZB(1)=Z
C                                    ZB(2)=Z**2
C                                    ZB(3)=A
C
C  REMARKS     UNITS - ENERGY     MBAR*CC/GM
C                      TEMP       DEGREES KELVIN
C                      DENSITY    GRAMS/CC
C                      PRESSURE   MBAR
C
C              THIS ROUTINE WAS ORIGINALLY NAMED GETINV BEFORE THE
C              MODIFICATIONS WERE MADE SO IT WOULD INTERFACE WITH EOSMOD
C
C  EXTERNALS   MATCHKX,TABRANX,INV301X
C
C  PROGRAMMER  J.ABDALLAH,JR.
C
C  DATE        13 JUNE 1979
C
C-----------------------------------------------------------------------
C
      LEVEL 2,TBLS
      COMMON/S2DIRX/LCMX,NRS,LCFW(10,3)
      DIMENSION ZB(3),TBLS(1)
C
C     OBTAIN THE UNIT CONVERSION FACTORS FROM THE EOSMOD ROUTINES
      COMMON/EOSCCE/TFACE,RFACE,PFACE,EFACE, KREPE
C
      CALL MATCHKX(MIO,NRS,LCFW(1,IOT),TBLS(1),IFL)
      IF(IFL.EQ.0) GO TO 10
      LCFW(IR,IOT)=IFL
      IFL=2
      RETURN
   10 NL=LCMX-LCNT-1
C . . FETCH EOS TABLES
      CALL TABRANX(MIO,201.,LU,TBLS(LCNT+2),NL,IFL)
      IF(IFL.LE.0) RETURN
      ZB(1)=TBLS(LCNT+2)
      ZB(2)=ZB(1)*ZB(1)
      ZB(3)=TBLS(LCNT+3)
      TBLS(LCNT+1)=TBLS(LCNT+4)
      CALL TABRANX(MIO,301.,LU,TBLS(LCNT+2),NL,IFL)
```

```
      IF(IFL.LE.0) RETURN
      TBLS(LCNT)=FLOAT(MID)
C     CALL PERTCB(IR,TBLS(LCNT),ZB(1),ZB(3))
      NR=TBLS(LCNT+2)
      NT=TBLS(LCNT+3)
      NRT=NR*NT
      LOCP=LCNT+3+NR+NT
C
C . . CONVERT TO DESIRED UNITS
      DO 30 I=1,NT
C
      TBLS(3+I+LCNT+NR)=TFACE*TBLS(3+I+LCNT+NR)
C
      DO 30 J=1,NR
C
      IF(I.GT.1) GO TO 20
      TBLS(3+J+LCNT)=TBLS(3+J+LCNT)*RFACE
      RHO=TBLS(3+J+LCNT)
 20   LOCP=LOCP+1
C
      TBLS(LOCP)=PFACE*TBLS(LOCP)
      TBLS(LOCP+NRT)=EFACE*TBLS(LOCP+NRT)
      IF (KREPE.EQ.1) TBLS(LOCP+NRT)=TBLS(LOCP+NRT)*RHO
 30   CONTINUE
C
C . . WINDOW TABLES HERE AND RESET VALUES OF NR NT AND
C     NRT IF WINDOWING IS NEEDED
C . . INVERT TABLES
C . . CHECK TO SEE IF THERE IS ENOUGH ROOM TO INVERT THE TABLES
C     NINV IS THE LAST LOCATION NEEDED FOR TABLE INVERSION
      NINV=LCNT+3+2*NRT+2*NR+4*NT
      IF(NINV.LE.LCMX) GO TO 40
      IFL=LCMX-NINV
      RETURN
 40   RO=TBLS(LCNT+1)
      LOC=LCNT+2
      CALL INV301X(TBLS,LOC,RO,LDS)
C . . DOUBLE PACK DEPENDENT VARIABLES
      LOCP=LCNT+3+NR+NT+NR
      DO 50 I=1,NRT
      LOCP=LOCP+1
      PTEM=TBLS(LOCP)
      TTEM=TBLS(LOCP+NRT)
      TBLS(LOCP)=DPACKX(PTEM,TTEM)
 50   CONTINUE
C . . WRAP UP
      LCFW(IR,IOT)=LCNT
      LCNT=LCNT+2+LDS-NRT
      IFL=1
      RETURN
      END
```

```
      SUBROUTINE GETRPOX(IR,MID,IDT,TBLS,LCNT,LU.IFL)
C-----------------------------------------------------------------------
C
C  SUBROUTINE  GETRPOX(IR,MIO,IDT,TBLS,LCNT.LU,IFL)
C
C  PURPOSE     TO LOAD THE ROSSELAND/PLANCK OPACITY TABLE
C
C  ARGUMENTS   IR        (INPUT)   REGION NO.
C              MIO       (INPUT)   SESAME MATERIAL ID
C              IDT       (INPUT)   DATA TYPE INDICATOR
C              TBLS      (OUTPUT)  ARRAY FOR TABLE STORAGE
C              LCNT      (I/O)     POSITION IN ARRAY FOR
C                                  STORING TABLES
C              LU        (INPUT)   SESAME LIBRARY UNIT NO.
C              IFL       (OUTPUT)  ERROR FLAG
C                                  =1 FOR SUCCESSFUL LOADING
C                                  =0 FOR DATA NOT FOUND
C                                  =-NO. OF EXTRA WORDS NEEDED
C                                    TO STORE DATA
C
C  REMARKS     THE ROSSELAND/PLANCK TABLE IS DOUBLE PACKED ON DISK.
C              THIS VERSION OF GETRPOX IS SPECIALLY DESIGNED FOR
C              USE IN HYDROX. IF NECESSARY, MODIFICATIONS
C              MAY BE MADE HERE FOR ADAPTATION TO OTHER CODES.
C
C              THIS ROUTINE WAS ORIGINALLY NAMED GETRPO BEFORE THE
C              MODIFICATIONS WERE MADE SO IT WOULD INTERFACE WITH EOSMOD
C
C  EXTERNALS   MATCHKX,TABRANX,DPACKX
C              COMMON/EOSCCO/ FROM THE EOSMOD PACKAGE
C
C  PROGRAMMER      J.ABDALLAH,JR.
C
C  MODIFIED BY M. KLEIN, GROUP T-7, 11 DECEMBER 1979
C
C  DATE            24 APRIL 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION TBLS(1)
      COMMON/S2DIRX/LCMX,NRS,LCFW(10,3)
C
C     THE COMMON BLOCK EOSCCO PROVIDES THE UNIT CONVERSION FACTORS
C     FROM THE EOSMOD PACKAGE
      COMMON/EOSCCO/ TFACO,RFACD,OFACO,KREPD
C
C     UNITS..TEMP..DEG.K,RHO IN G/CC,OPACITY IN CM**2/G
C . . CHECK TO SEE IF TABLE HAS BEEN LOADED ALREADY
      CALL MATCHKX(MIO,NRS,LCFW(1,IOT),TBLS(1),IFL)
      IF(IFL.EQ.0) GO TO 10
      LCFW(IR,IOT)=IFL
      IFL=2
      RETURN
   10 NLEFT=LCMX-LCNT-1
C . . FETCH THE 502 TABLE
      CALL TABRANX(MIO,502.,LU,TBLS(LCNT+2),NLEFT,IFL)
      IF(IFL.LE.0) RETURN
C . . CONVERT TO DESIRED UNITS
C . . LINES THROUGH STATEMENT 60 MAY BE DELETED IF NO CONVERSION IS
C . . REQUIRED
      NR=TBLS(LCNT+2)
      NT=TBLS(LCNT+3)
```

```
      IPT=LCNT+3+NR+NT
      DO 60 K=1,NT
      TBLS(LCNT+3+NR+K)=TBLS(LCNT+3+NR+K) + TFACO
      DO 60 J=1,NR
      IF(K.GT.1) GO TO 50
      TBLS(LCNT+3+J)=TBLS(LCNT+3+J)+RFACO
50    IPT=IPT+1
      ROP=TBLS(IPT)
      POP=SHIFT(ROP,30)
      ROP=ROP+TBLS(LCNT+3+J)*KREPO+DFACO
      POP=POP+TBLS(LCNT+3+J)*KREPO+DFACO
      ROP=DPACKX(ROP,POP)
      TBLS(IPT)=ROP
60    CONTINUE
      TBLS(LCNT)=FLOAT(MIO)
      TBLS(LCNT+1)=FLOAT(IOT)
      LCFW(IR,IDT)=LCNT
      LCNT=LCNT+IFL+2
      IFL=1
      RETURN
      END
```

```
      SUBROUTINE INBUFRX(LU,Z,NW,IAD,IFLG)
C-----------------------------------------------------------------------
C
C  SUBROUTINE   INBUFRX(LU,Z.NW,IAD,IFLG)
C
C  PURPOSE      RANDOM I/O READ
C
C  ARGUMENTS    LU   (INPUT)  UNIT NO.
C               Z    (OUTPUT)  STORAGE AREA WHERE DAT IS RETURNED
C               NW   (INPUT)  NO. OF WORDS TO BE READ
C               IAD  (INPUT)  STARTING DISK ADDRESS OF DATA
C               IFLG (OUTPUT) 0=NORMAL
C                              1=EOF ENCOUNTERED
C                             -1=ERROR
C
C  REMARKS      NONE
C
C  EXTERNALS    RDISK
C
C  PROGRAMMER   J.ABDALLAH,JR.
C
C  DATE         1 MAY 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,Z
      CALL RDISK(LU,Z.NW,IAD)
      IF(UNIT(LU)) 10,20,30
 10   IFLG=1
      RETURN
 20   IFLG=0
      RETURN
 30   IFLG=-1
      RETURN
      END
```

```
      SUBROUTINE INV301X(OSTR,LOC,RO,LDS)
C-----------------------------------------------------------------
C
C  SUBROUTINE:  INV301X(OSTR,LOC,RO,LDS)
C
C  PURPOSE:     INVERT DATA STRING OF TYPE 301 TO TYPE 302.
C
C  ARGUMENTS:   OSTR (INPUT) - TABLE STORAGE ARRAY
C               LOC  (INPUT) - STARTING LOCATION OF DATA STRING
C                              IN OSTR
C               RO   (INPUT) - APPROXIMATE DENSITY OF SOLID
C               LDS (OUTPUT) - LENGTH OF NEW DATA STRING
C
C  REMARKS:     OSTR CAN BE DECLARED LCM ON THE CDC 7600.
C               THIS ROUTINE OVERWRITES LOCATIONS FOLLOWING THE
C               DATA STRING.  IT EXPANDS THE STRING BY NR WORDS,
C               WHERE NR IS THE NUMBER OF DENSITIES.  IT ALSO
C               USES 3*NT WORDS AS TEMPORARY STORAGE, WHERE NT
C               IS THE NUMBER OF TEMPERATURES.
C
C  EXTERNALS:   ISRCHKX, RATFN1X
C
C  PROGRAMMER:  G. I. KERLEY, T-4.
C
C  DATE:        4 OCTOBER 1977
C
C-----------------------------------------------------------------
      LEVEL 2,OSTR
      DIMENSION OSTR(1)
      COMMON/INTOROX/IFN
      COMMON/RTBLK1X/LOCX,NR,LOCY,KY,JX,NT,INT,ET,Z(2)
      INT=1
      IFNS=IFN
      IFN=0
      NR = OSTR(LOC)
      NT = OSTR(LOC+1)
      LOCT = 2+NR+LOC
      LCEC = LOCT+NT
      LOCP = LCEC+NR
      LOCE = LOCP+NR*NT
      LOCN = LOCE+NR*NT
      IMAX = 2*NR*NT
      DO 1 I=1,IMAX
    1 OSTR(LOCN-I) = DSTR(LOCN-I-NR)
      DO 2 I=1,NR
      JJ = LOCE+I-1
      Q = 1.E-12*ABS(DSTR(JJ))
      OSTR(LCEC+I-1) = OSTR(JJ)
      OSTR(JJ) = 0.
      DO 2 J=2,NT
      JJ = JJ+NR
      OSTR(JJ) = OSTR(JJ)-OSTR(LCEC+I-1)
      IF(OSTR(JJ)-OSTR(JJ-NR).LT.Q) DSTR(JJ)=DSTR(JJ-NR)+Q
    2 CONTINUE
      I = ISRCHKX(RO,DSTR(LOC+3),NR-2,1,0)+1
      DO 3 J=1,NT
      OSTR(LOCN+J-1) = OSTR(LOCT+J-1)
    3 DSTR(LOCT+J-1) = OSTR(LOCE+I-1+NR*(J-1))
      DO 5 I=1,NR
      LOCX = LOCE+I-1
      DO 4 J=1,NT
      ET = OSTR(LOCT+J-1)
```

```
      JX = ISRCHKX(ET,OSTR(LOCX+NR),NT-2,NR,0)+1
      LOCY = LOCP+I-1
      KY = NR
      CALL RATFN1X
      OSTR(LOCN+NT+J-1) = Z(1)
      LOCY = LOCN
      KY = 1
      CALL RATFN1X
4     DSTR(LOCN+NT+NT+J-1) = Z(1)
      DO 5 J=1,NT
      OSTR(LOCP+I-1+NR*(J-1)) = DSTR(LOCN+NT+J-1)
5     OSTR(LOCX+NR*(J-1)) = OSTR(LOCN+NT+NT+J-1)
      LDS = LOCN-LOC
      IFN=IFNS
      RETURN
      END
```

```
      FUNCTION ISRCHKX(X,TBLS,N,K,NSFT)
C-----------------------------------------------------------------
C
C  FUNCTION:    ISRCHKX(X,TBLS,N,K,NSFT)
C
C  PURPOSE:     FIND INDEX OF X IN AN ARRAY TBLS.  TABLE VALUES
C               NEED NOT BE CONTIGUOUS AND CAN BE IN EITHER
C               ASCENDING OR DESCENDING ORDER.
C
C  ARGUMENTS:   X     (INPUT) - VALUE TO BE LOCATED
C               TBLS  (INPUT) - TABLE TO BE SEARCHED
C               N     (INPUT) - NUMBER OF VALUES TO BE SEARCHED
C               K     (INPUT) - SPACING BETWEEN VALUES IN TABLE
C               THE VALUE OF THE FUNCTION = INDEX I, WHERE
C                 TBLS(1+K*(I-1)).LE.X.LT.TBLS(1+K*I), OR
C                 TBLS(1+K*(I-1)).GE.X.GT.TBLS(1+K*I), OR
C                 I=0 OR I=N IF X IS OUTSIDE RANGE OF TABLE.
C               NSFT (INPUT) - NO. OF BITS THE TABLE VALUES ARE
C                              TO BE SHIFTED
C
C  REMARKS:     TBLS CAN BE DECLARED LCM ON THE CDC 7600.
C
C  EXTERNALS:   SHIFT.
C
C  PROGRAMMER:  G. I. KERLEY, T-4, J.ABDALLAH,JR.
C
C  DATE:        19 NOVEMBER 1976,REVISED 6 JULY 1979
C
C-----------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION TBLS(1)
      ISRCHKX = 0
      J = N+1
      KI = 1-K
      S1=TBLS(1)
      S1=SHIFT(S1,NSFT)
      S=TBLS(KI+K*N)
      S=SHIFT(S,NSFT)
      S=S-S1
1     IF(J-ISRCHKX.EQ.1) RETURN
      JP = .5*(J+ISRCHKX)
      S1=TBLS(KI+K*JP)
      S1=SHIFT(S1,NSFT)
      IF(S*(X-S1).LT.0.0) GO TO 2
      ISRCHKX = JP
      GO TO 1
2     J = JP
      GO TO 1
      END
```

```
      SUBROUTINE MATCHKX(MID,NRS,LOC,TBLS,IFLG)
C-----------------------------------------------------------------------
C
C  SUBROUTINE  MATCHKX(MID,NRS.LOC,TBLS,IFLG)
C
C  PURPOSE     TO CHECK IF A MATERIAL HAS BEEN
C              PREVIOUSLY LOADED
C
C  ARGUMENTS   MID        (INPUT)   SESAME MATERIAL ID
C              NRS        (INPUT)   NUMBER OF REGIONS
C              LOC        (INPUT)   ARRAY OF FIRST WORD LOCATIONS
C                                   IN TABLE STORAGE ARRAY FOR
C                                   FOR EACH REGION
C              TBLS       (INPUT)   TABLE STORAGE ARRAY
C              IFLG       (OUTPUT)  =0 MATERIAL NOT PREVIOUSLY LOADED
C                                   GT.0 LOCATION OF TABLE IF LOADED
C                                   ALREADY
C
C  REMARKS     NONE
C
C  EXTERNALS   NONE
C
C  PROGRAMMER  J.ABDALLAH,JR.
C
C  DATE        26 APRIL 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION LOC(1),TBLS(1)
      IFLG=0
      DO 100 J=1,NRS
      LC=LOC(J)
      IF(LC.LE.0) GO TO 100
      ITEST=TBLS(LC)
      IF(MID.EQ.ITEST) GO TO 200
  100 CONTINUE
      RETURN
  200 IFLG=LC
      RETURN
      END
```

```
      SUBROUTINE RATFN1X
C-----------------------------------------------------------------
C
C  SUBROUTINE:  RATFN1X
C
C  PURPOSE:     INTERPOLATE FOR A FUNCTION Y(X) AND ITS
C               DERIVATIVE FROM TABLES LOCATED IN ARRAY TBLS.
C
C               THE ROUTINE ALSO REQUIRES COMMON BLOCKS,
C               COMMON/RTBLK1X/LOCX,KX,LOCY,KY,I,N,IP,X,Y(2)
C                 LOCX = LOCATION OF X VECTOR
C                 KX   = SPACING OF X VECTOR
C                 LOCY = LOCATION OF Y VECTOR
C                 KY   = SPACING OF Y VECTOR
C                 I    = INDEX OF X AND Y VECTORS
C                 N    = LENGTH OF X AND Y VECTORS
C               X    (INPUT) - INDEPENDENT VARIABLE
C               Y   (OUTPUT) - VECTOR OF LENGTH 2, WHERE
C                 Y(1) = VALUE OF FUNCTION
C                 Y(2) = DERIVATIVE OF FUNCTION
C               IP   (INPUT) - BRANCH PARAMETER
C                 IP.EQ.0, USE INPUT COEFFICIENTS IN YY
C                 IP.NE.0, CALCULATE YY VECTOR FIRST
C               COMMON/INTOROX/IFN
C               IFN  (INPUT) - INTERPOLATION TYPE
C                 IFN.NE.1, RATIONAL FUNCTION
C                 IFN.EQ.1, LINEAR
C               COMMON/SESOATX/TBLS
C                TBLS (INPUT) - TABLE STORAGE ARRAY
C
C
C  REMARKS:     UNLESS LINEAR FORM IS SPECIFIED, ROUTINE
C               USES RATIONAL FUNCTION METHOD WITH QUADRATIC
C               ESTIMATE OF DERIVATIVES AT THE MESH POINTS.
C               TBLS CAN BE DECLARED LCM ON THE CDC 7600.
C
C  EXTERNALS:   NONE, BUT A SEARCH ROUTINE MUST BE CALLED
C               FIRST, TO COMPUTE INDEX I.
C
C  PROGRAMMER:  G. I. KERLEY, T-4.
C
C  DATE:        18 JULY 1979
C
C-----------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION YY(6)
      COMMON/SESOATX/TBLS(10000)
      COMMON/INTOROX/IFN
      COMMON/RTBLK1X/LOCX,KX,LOCY,KY,I,N,IP,X,Y(2)
      IF(IFN.EQ.1) GO TO 6
      IF(IP.EQ.0) GO TO 3
C  CALCULATE COEFFICIENTS FOR RATIONAL FUNCTION INTERPOLATION
      IX = LOCX+KX*(I-1)
      IY = LOCY+KY*(I-1)
      YY(3) = TBLS(IX)
      YY(4) = TBLS(IX+KX)-YY(3)
      YY(1) = TBLS(IY)
      YY(2) = (TBLS(IY+KY)-YY(1))/YY(4)
      IF(I.EQ.N-1) GO TO 1
      SP = (TBLS(IY+KY+KY)-TBLS(IY+KY))/(TBLS(IX+KX+KX)-TBLS(IX+KX))
      YY(6) = (SP-YY(2))/(TBLS(IX+KX+KX)-YY(3))
      IF(I.GT.1) GO TO 1
```

```
      IF(YY(2)*(YY(2)-YY(4)*YY(6)).LE.0.) YY(6)=YY(2)/YY(4)
      YY(5) = YY(6)
      GO TO 2
 1    DM = YY(3)-TBLS(IX-KX)
      SM = (YY(1)-TBLS(IY-KY))/DM
      YY(5) = (YY(2)-SM)/(YY(4)+DM)
      IF(I.EQ.N-1) YY(6)=YY(5)
      IF(I.GT.2) GO TO 2
      IF(SM*(SM-DM*YY(5)).LE.0.) YY(5)=(YY(2)-SM-SM)/YY(4)
 2    IF(YY(6).NE.0.) YY(5)=YY(5)/YY(6)
C  EVALUATE RATIONAL FUNCTION FROM PRECALCULATED COEFFICIENTS
 3    Q = X-YY(3)
      R = YY(4)-Q
      IF(R.NE.0.) GO TO 4
      W = 1.
      GO TO 5
 4    W = 1.-1./(1.+ABS(YY(5)*Q/R))
 5    F = YY(6)*(W+YY(5)*(1.-W))
      Y(1) = YY(1)+Q*(YY(2)-R*F)
      Y(2) = YY(2)+(Q-R)*F+YY(4)*W*(F-YY(6))
      RETURN
C  CALCULATE COEFFICIENTS FOR LINEAR INTERPOLATION
 6    IF(IP.EQ.0) GO TO 7
      IX = LOCX+KX*(I-1)
      IY = LOCY+KY*(I-1)
      YY(3) = TBLS(IX)
      YY(1) = TBLS(IY)
      YY(2) = (TBLS(IY+KY)-YY(1))/(TBLS(IX+KX)-YY(3))
C  CALCULATE LINEAR ESTIMATE FROM PRECALCULATED COEFFICIENTS
 7    Y(1) = YY(1)+YY(2)*(X-YY(3))
      Y(2) = YY(2)
      RETURN
      END
```

```
      SUBROUTINE TABRANX(MID,TID,LIB,A,LEN,IFLAG)
C-----------------------------------------------------------------------
C
C  SUBROUTINE  TABRANX(MID,TID,LIB,A,LEN,IFLAG)
C
C  PURPOSE     TO FETCH A GIVEN TABLE FOR A GIVEN MATERIAL
C              FROM A SESAME II LIBRARY
C
C  ARGUMENTS   MID        (INPUT)   MATERIAL ID
C              TID        (INPUT)   TABLE NO.- IF 0.0 MATERIAL INDEX
C                                   IS RETURNED
C              LIB        (INPUT)   LIBRARY FILE UNIT NO.
C              A          (OUTPUT)  ARRAY FOR TABLE STORAGE
C              LEN        (INPUT)   NO. OF WORDS IN A AVAILABLE
C              IFLAG      (OUTPUT)  =0 IF TABLE COULD NOT BE LOCATED
C                                   GT. 0=NO. OF WORDS IN TABLE RETURNED
C                                   LT. 0 - NO. OF ADDITIONAL
C                                   WORDS NEEDED
C
C  REMARKS     A RANDOM I/O TECHNIQUE IS USED TO LOCATE AND LOAD
C              THE SPECIFIED TABLE FROM THE SESAME II LIBRARY.
C              THE MATERIAL INDEX AND ITS ADDRESS ARE TO SAVED
C              TO HASTEN THE FETCHING OF ANOTHER TABLE FOR THE SAME
C              MATERIAL AND LIBRARY FILE IN SUBSEQUENT CALLS TO
C              TABFCH.
C
C  EXTERNALS   INBUFRX
C
C  PROGRAMMER  J.ABDALLAH.,JR.
C
C  DATE        24 APRIL 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,A
      DIMENSION A(1),HINDEX(50)
      DATA HINDEX(1)/0.0/
      DATA LIBLST/0/
      IFLAG=0
C . . FIND NO. MATERIALS ON LIBRARY
      IF(LIB.NE.LIBLST) GO TO 50
      IDLAST=HINDEX(1)
      IF(IDLAST.NE.MID) GO TO 50
      IF(TID.NE.0.0) GO TO 230
      NW=HINDEX(5)
      NW=NW+NW+5
      IF(LEN.LT.NW) GO TO 999
      DO 30 J=1,NW
      A(J)=HINDEX(J)
30    CONTINUE
      IFLAG=NW
      RETURN
50    LIBLST=LIB
      NW=1
      IF(LEN.LT.NW) GO TO 999
      CALL INBUFRX(LIB,A,1,0,IER)
      N=A(1)
      NW=N+N+N
      IF(LEN.LT.NW) GO TO 999
      CALL INBUFRX(LIB,A,NW,3,IER)
C . . FIND ADDRESS OF MATERIAL FILE
      DO 100 J=1,N
      ITEST=A(J)
```

```
      IF(ITEST.NE.MIO) GO TO 100
      NW=A(J+N)
      IAO=A(J+N+N)
      GO TO 200
  100 CONTINUE
      RETURN
C . . GET MATERIAL INDEX
  200 IF(LEN.LT.NW) GO TO 999
      IADX=IAO
      CALL INBUFRX(LIB,A,NW,IAOX,IER)
      DO 210 J=1,NW
      HINDEX(J)=A(J)
  210 CONTINUE
      IF(TID.EQ.0.0) GO TO 500
  230 N=HINDEX(5)
      DO 300 J=1,N
      IF(TIO.NE.HINDEX(5+J)) GO TO 300
      NW=HINDEX(5+J+N)
      IAO=HINDEX(5+J+N+N)
      IAO=IADX+IAO
      GO TO 400
  300 CONTINUE
      RETURN
  400 IF(LEN.LT.NW) GO TO 999
C . . READ REQUESTED TABLE
      CALL INBUFRX(LIB,A,NW,IAO,IER)
  500 IFLAG=NW
      RETURN
  999 IFLAG=LEN-NW
      RETURN
      END
```

```
      SUBROUTINE T4DATIX
C-----------------------------------------------------------------
C
C  SUBROUTINE:  T4DATIX
C
C  PURPOSE:     SEARCH/INTERPOLATE FOR PRESSURE AND TEMPERATURE
C               AS FUNCTIONS OF REGION, DENSITY AND ENERGY,
C               USING PACKED SESAME 2 DATA STRING OF TYPE 302
C
C               COMMON/SESINX/IR,IOT,R,E,IBR,IFL
C               COMMON/SESOUTX/P(3),T(3)
C               IR    (INPUT) - MATERIAL REGION  NUMBER
C               IOT   (INPUT) - DATA TYPE INDICATOR
C               R     (INPUT) - DENSITY
C               E     (INPUT) - INTERNAL ENERGY
C               P,T (OUTPUT) - PRESSURE, TEMPERATURE VECTORS
C                 P(1),T(1) = PRESSURE AND TEMPERATURE
C                 P(2),T(2) = DENSITY DERIVATIVES
C                 P(3),T(3) = ENERGY DERIVATIVES
C               IBR   (INPUT) - 0=COMPUTE BOTH P AND T
C                               1=COMPUTE P ONLY
C                               2=COMPUTE T ONLY
C               COMMON/SESDATX/TBLS
C               TBLS (INPUT) - TABLE STORAGE ARRAY
C
C  REMARKS:     ADAPTED FROM T-4 SESAME 2 ROUTINES S2EOSI AND
C               LA302A.  PRESSURE AND TEMPERATURE ARE PACKED.
C               THE SEARCH INDICES AND INTERPOLATION CONSTANTS
C               ARE SAVED AND REUSED, IF POSSIBLE.
C
C        *****  SYSTEM DEPENDENT FEATURE.  THE CONSTANT NSFT
C        *****  IN STATEMENT 60 SHOULD BE SET TO 1/2 THE BIT
C        *****  LENGTH.  FOR A CDC 7600, NSFT = 30.
C
C  EXTERNALS:   RATFN1X (1-D INTERPOLATION ROUTINE)
C               T4INTPX (2-D INTERPOLATION ROUTINE)
C
C  PROGRAMMER:  G. I. KERLEY AND B. I. BENNETT, T-4.
C               J. ABDALLAH,JR.
C
C  DATE:        2 AUGUST 1978
C
C-----------------------------------------------------------------
      LEVEL 2,TBLS
      COMMON/S2DIRX/LCMX,NRS,LCFW(10,3)
      COMMON/RTBLK1X/LOCR,KX,LOCE,KY,IRX,N,ISAME,RX1,PX1(2)
      COMMON/RTBLK2X/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,
     $ RX2,ET,PX2(3),INT,IOS,ZZ(96)
      COMMON/SESINX/IR,IOT,R,E,IBR,IFL
      COMMON/SESOUTX/P(3),T(3)
      COMMON/SESDATX/TBLS(10000)
      DATA LOCLST,IP,IT/0,1,1/
C  LOC IS POINTER TO START OF DATA STRING FOR REGION IR
      LOC = LCFW(IR,IOT)+2
C . . THE FOLLOWING LINES OF CODE (THRU NZ=1) CAN BE
C     MOVED AFTER THE IF(LOC.EQ.LOCLST) GO TO 5
C     STATEMENT TO MAKE THE SUBROUTINE QUICKER FOR CODES WHICH
C     DO NOT ALSO USE TEMPERATURE BASED EOS TABLES.
      NX = TBLS(LOC)
      NY = TBLS(LOC+1)
      N = NX
      LOCR = LOC+2
```

```
      KX = 1
      LOCX = LOCR
      LOCY = LOCX+NX
      LOCE = LOCY+NY
      KY = 1
      LOCZ = LOCE+NX
      NZ = 1
C  TEST TO SEE IF THE MATERIAL IS THE SAME AS LAST CALL
      IF(LOC.EQ.LOCLST) GO TO 5
C  THE FOLLOWING OPERATIONS DO NOT NEED TO BE REPEATED
C  UNLESS A NEW REGION HAS BEEN ENTERED
      LOCLST=LOC
      IXLAST = 0
      IYLAST = 0
      LOCI = LOCX+NX/2-1
      LOCJ = LOCY+NY/2-1
      LOCNX=LOCX+NX-2
      LOCNY=LOCY+NY-2
C  SEARCH FOR DENSITY INDEX
 5    IF(R.LT.TBLS(LOCI)) GO TO 15
 10   IF(R.LT.TBLS(LOCI+1)) GO TO 20
      IF(LOCI.EQ.LOCNX) GO TO 20
      LOCI=LOCI+1
      GO TO 10
 15   IF(LOCI.EQ.LOCX) GO TO 20
      LOCI=LOCI-1
      IF(R.LT.TBLS(LOCI)) GO TO 15
 20   IX=LOCI-LOCX+1
C  INTERPOLATE FOR ENERGY ON COLD CURVE.  IF ISAME = 0, DENSITY
C  INDEX IS THE SAME AS IN THE LAST CALL TO THIS ROUTINE
      IRX = IX
      ISAME = IABS(IX-IXLAST)
      RX1=R
      CALL RATFN1X
      ET = AMAX1(0.,E-PX1(1))
      DECOR = PX1(2)
      RX2=R
C  SEARCH FOR ENERGY INDEX
      IF(ET.LT.TBLS(LOCJ)) GO TO 35
 30   IF(ET.LT.TBLS(LOCJ+1)) GO TO 40
      IF(LOCJ.EQ.LOCNY) GO TO 40
      LOCJ=LOCJ+1
      GO TO 30
 35   IF(LOCJ.EQ.LOCY) GO TO 40
      LOCJ=LOCJ-1
      IF(ET.LT.TBLS(LOCJ)) GO TO 35
 40   IY=LOCJ-LOCY+1
C  IF ISAME = 0,  DENSITY AND TEMPERATURE INDICES ARE
C  THE SAME AS IN THE LAST CALL TO THIS ROUTINE
      ISAME = ISAME+IABS(IY-IYLAST)
      IP = MINO(1,IP+ISAME)
      IT = MINO(1,IT+ISAME)
      IXLAST = IX
      IYLAST = IY
      IOS=(IOT-1)*32+1
      IF(IBR.EQ.2) GO TO 50
C  PRESSURE CALCULATION
      NSFT = 0
      INT=IP
      CALL T4INTPX
      P(1)=PX2(1)
      P(2)=PX2(2)-DECOR*PX2(3)
```

```
      P(3)=PX2(3)
      IP = 0
      IF(IBR.EQ.1) RETURN
C  TEMPERATURE CALCULATION
 50   NSFT = 30
      INT=IT
      IOS=IOS+16
      CALL T4INTPX
      T(1)=PX2(1)
      T(2)=PX2(2)-OECOR*PX2(3)
      T(3)=PX2(3)
      IT = 0
      RETURN
      END
```

```
      SUBROUTINE T4OATX
C-----------------------------------------------------------------
C
C  SUBROUTINE:  T4OATX
C
C  PURPOSE:     SEARCH/INTERPOLATE FOR ATOMIC DATA AS
C               FUNCTIONS OF REGION, DENSITY AND TEMPERATURE,
C               USING PACKED SESAME 2 DATA STRING
C
C               COMMON/SESINX/IR,IOT,AR,AT,IBR,IFL
C               IR   (INPUT) - MATERIAL REGION  NUMBER
C               IDT  (INPUT) - DATA TYPE INDICATOR
C               AR   (INPUT) - DENSITY
C               AT   (INPUT) - TEMPERATURE
C               IBR  (INPUT) - SPECIFIES VARIABLES REQUIRED
C                 IBR = 0, BOTH VARIABLES
C                 IBR = 1, FIRST HALF VARIABLE ONLY
C                 IBR = 2, SECOND HALF VARIABLE ONLY
C               IFL  (NOT USED)
C               COMMON/SESOUTX/P(3),E(3)
C               P,E (OUTPUT) - VARIABLES OF FIRST AND SECOND HALF
C                              OF PACKED DATA STRING
C                 P(1),E(1) = VALUE OF THE VARIABLES
C                 P(2),E(2) = DENSITY DERIVATIVES
C                 P(3),E(3) = TEMPERATURE DERIVATIVES
C
C  REMARKS:     ADAPTED FROM T-4 SESAME 2 ROUTINES S2EOS AND
C               LA301A.  TABLES OF 2 VARIABLES ARE DOUBLE PACKED.
C               THE SEARCH INDICES AND INTERPOLATION CONSTANTS
C               ARE SAVED AND REUSED, IF POSSIBLE.
C
C        *****  SYSTEM DEPENDENT FEATURE.  THE CONSTANT NSFT
C        *****  IN STATEMENT 60 SHOULD BE SET TO 1/2 THE BIT
C        *****  LENGTH.  FOR A CDC 7600, NSFT = 30.
C
C  EXTERNALS:   T4INTPX (RATIONAL FUNCTION AND BI LINEAR INTERPOLATION)
C               INTERPOLATION COEFFICIENTS FROM A PREVIOUS
C               CALL TO THE ROUTINE CAN BE REUSED.
C
C  PROGRAMMER:  G. I. KERLEY, T-4.,J.ABDALLAH,T-4
C
C  DATE:        11 JULY 1978,REVISED 27 APRIL 1979
C
C-----------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION LOCLST(3),IXLAST(3),IYLAST(3),
     1          IPLAST(3),IELAST(3)
      COMMON/S2DIRX/LCMX,NRS,LCFW(10,3)
      COMMON/RTBLK2X/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,
     1 R,T,Z(3),INT,IOS,ZZ(96)
      COMMON/SESDATX/TBLS(10000)
      COMMON/SESINX/IR,IOT,AR,AT,IBR,IFL
      COMMON/SESOUTX/P(3),E(3)
      DATA IPLAST/3*0/
      DATA IELAST/3*1/
      DATA LOCLST/3*0/
C  LOC IS POINTER TO START OF DATA STRING FOR REGION IR
      LOC = LCFW(IR,IOT)+2
      NZ=1
      NX=TBLS(LOC)
      NY=TBLS(LOC+1)
      LOCX=LOC+2
```

```
      LOCY=LOCX+NX
      LOCZ=LOCY+NY
      LOCNX=LOCX+NX-2
      LOCNY=LOCY+NY-2
      IF(LOC.EQ.LOCLST(IOT)) GO TO 2
      LOCLST(IDT)=LOC
      IX=NX/2
      IY=NY/2
      IXLAST(IOT)=0
      IYLAST(IOT)=0
      GO TO 3
 2    IX=IXLAST(IOT)
      IY=IYLAST(IOT)
 3    R=AR
      T=AT
C  SEARCH FOR DENSITY INDEX
 5    LOCI=LOCX+IX-1
      IF(R.LT.TBLS(LOCI)) GO TO 15
 10   IF(R.LT.TBLS(LOCI+1)) GO TO 20
      IF(LOCI.EQ.LOCNX) GO TO 20
      LOCI=LOCI+1
      GO TO 10
 15   IF(LOCI.EQ.LOCX) GO TO 20
      LOCI=LOCI-1
      IF(R.LT.TBLS(LOCI)) GO TO 15
 20   IX=LOCI-LOCX+1
C  SEARCH FOR TEMPERATURE INDEX
      LOCI=LOCY+IY-1
      IF(T.LT.TBLS(LOCI)) GO TO 35
 30   IF(T.LT.TBLS(LOCI+1)) GO TO 40
      IF(LOCI.EQ.LOCNY) GO TO 40
      LOCI=LOCI+1
      GO TO 30
 35   IF(LOCI.EQ.LOCY) GO TO 40
      LOCI=LOCI-1
      IF(T.LT.TBLS(LOCI)) GO TO 35
 40   IY=LOCI-LOCY+1
C  IF ISAME = 0,  DENSITY AND TEMPERATURE INDICES ARE
C  THE SAME AS IN THE LAST CALL TO THIS ROUTINE
      ISAME = IABS(IX-IXLAST(IOT))+IABS(IY-IYLAST(IOT))
      IXLAST(IOT) = IX
      IYLAST(IDT) = IY
      IOS=(IOT-1)*32+1
      IPLAST(IDT)=MINO(1,IPLAST(IOT)+ISAME)
      IELAST(IDT)=MINO(1,IELAST(IOT)+ISAME)
      IF(IBR.EQ.2) GO TO 50
      INT=IPLAST(IOT)
      NSFT=0
      CALL T4INTPX
      P(1)=Z(1)
      P(2)=Z(2)
      P(3)=Z(3)
      IPLAST(IDT)=0
      IF(IBR.EQ.1) RETURN
 50   INT=IELAST(IOT)
      IOS=IOS+16
      NSFT=30
      CALL T4INTPX
      E(1)=Z(1)
      E(2)=Z(2)
      E(3)=Z(3)
      IELAST(IDT)=0
```

```
125         RETURN
126         END
```

```
      SUBROUTINE T4INTPX
C-------------------------------------------------------------
C
C  SUBROUTINE:  T4INTPX
C
C  PURPOSE:     INTERPOLATE FOR A FUNCTION Z(X,Y) AND ITS
C               DERIVATIVES FROM TABLES LOCATED IN ARRAY TBLS.
C
C               THE ROUTINE REQUIRES COMMON BLOCKS,
C               COMMON/RTBLK2X/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,
C                              X,Y,Z(3),IP,IOS,ZZ
C                 LOCX = LOCATION OF X VECTOR
C                 IX   = INDEX OF X VECTOR
C                 NX   = LENGTH OF X VECTOR
C                 LOCY = LOCATION OF Y VECTOR
C                 IY   = INDEX OF Y VECTOR
C                 NY   = LENGTH OF Y VECTOR
C                 LOCZ = LOCATION OF Z(X,Y) ARRAY
C                 NZ   = SPACING OF Z ARRAY
C                 NSFT = BIT SHIFT PARAMETER
C               X,Y  (INPUT) - INDEPENDENT VARIABLES
C               Z    (OUTPUT) - VECTOR OF LENGTH 3, WHERE
C                 Z(1) = VALUE OF FUNCTION
C                 Z(2) = X DERIVATIVE OF FUNCTION
C                 Z(3) = Y DERIVATIVE OF FUNCTION
C               ZZ   (IN/OUT) - COEFFICIENT VECTOR OF LENGTH 16
C               IP    (INPUT) - BRANCH PARAMETER
C                 IP.EQ.0, USE INPUT COEFFICIENTS IN ZZ
C                 IP.NE.0, CALCULATE ZZ VECTOR FIRST
C               IDS  (INPUT) - DISPLACEMENT INTO ZZ FOR COEFFS.
C                              TO BE USED
C               COMMON/INTORDX/IFN.
C               IFN  (INPUT) - INTERPOLATION TYPE
C                 IFN.NE.1, RATIONAL FUNCTION
C                 IFN.EQ.1, BILINEAR
C               COMMON/SESOATX/TBLS
C                 TBLS IS THE TABLE STORAGE ARRAY
C
C
C  REMARKS:     UNLESS BILINEAR FORM IS SPECIFIED, ROUTINE
C               USES RATIONAL FUNCTION METHOD WITH QUADRATIC
C               ESTIMATE OF DERIVATIVES AT THE MESH POINTS.
C               TBLS CAN BE DECLARED LCM ON THE CDC 7600.
C
C        *****  SYSTEM DEPENDENT FEATURE.  THE Z-ARRAY CAN BE
C        *****  DOUBLE PACKED.  PARAMETER NSFT SPECIFIES THE
C        *****  NUMBER OF BITS TO BE SHIFTED WHEN UNPACKING THE
C        *****  RIGHT HALF OF THE WORD.  THIS ROUTINE USES
C        *****  THE LASL SHIFT FUNCTION
C
C  EXTERNALS:   NONE, BUT A SEARCH ROUTINE MUST BE CALLED
C               FIRST, TO COMPUTE INDICES IX AND IY.
C
C  PROGRAMMER:  G. I. KERLEY, T-4., J. ABDALLAH,T-4.
C
C  DATE:        01 AUG 1979
C
C-------------------------------------------------------------
      LEVEL 2,TBLS
      COMMON/RTBLK2X/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,X,Y,Z(3),
     $ IP,IDS,ZZ(96)
      COMMON/INTORDX/IFN
```

```
      COMMON/SESOATX/TBLS(10000)
C  CALCULATE COEFFICIENTS FOR RATIONAL FUNCTION INTERPOLATION
      IF(IFN.EQ.1) GO TO 13
      IF(IP.EQ.0) GO TO 8
      I = LOCX+IX-1
      IZ = LOCZ+NZ*(IX-1+NX*(IY-1))
      KZ = NZ
      IBR = IX
      NBR = NX-IX
      ZZ(IOS+4) = TBLS(I)
      DO 7 K=1,4
      KI=IOS+K-1
      IF(K.LT.4) GO TO 1
      IZ = IZ+NZ
      GO TO 4
1     IF(K.LT.3) GO TO 2
      ZZ(IOS+6) = 0
      I = LOCY+IY-1
      KZ = KZ*NX
      IZ = IZ-KZ
      IBR = IY
      NBR = NY-IY
      ZZ(IOS+5) = TBLS(I)
      GO TO 3
2     IF(K.LT.2) GO TO 3
      IZ = IZ+NX*NZ
      GO TO 4
3     D = TBLS(I+1)-TBLS(I)
4     ZZ(KI)=SHIFT(TBLS(IZ),NSFT)
      S=SHIFT(TBLS(IZ+KZ),NSFT)
      S = (S-ZZ(KI))/D
      IF(NBR.EQ.1) GO TO 5
      SP=SHIFT(TBLS(IZ+KZ+KZ),NSFT)
      SP = (SP-D*S-ZZ(KI))/(TBLS(I+2)-TBLS(I+1))
      G2 = (SP-S)/(TBLS(I+2)-TBLS(I))
      IF(IBR.GT.1) GO TO 5
      IF(S*(S-D*G2).LE.0.) G2=S/D
      G1 = G2
      GO TO 6
5     DM = TBLS(I)-TBLS(I-1)
      SM=SHIFT(TBLS(IZ-KZ),NSFT)
      SM = (ZZ(KI)-SM)/DM
      G1 = (S-SM)/(D+DM)
      IF(NBR.EQ.1) G2=G1
      IF(IBR.GT.2) GO TO 6
      IF(SM*(SM-DM*G1).LE.0.) G1=(S-SM-SM)/D
6     IF(G2.NE.0.) G1=G1/G2
      ZZ(KI+8) = G1
7     ZZ(KI+12) = G2
      ZZ(IOS+7)=0
      ZZ8=ZZ(IOS+7)
      ZZ7=ZZ(IOS+6)
      ZZ(IOS+2)=(ZZ(IOS+1)-ZZ(IOS))/ZZ8
      ZZ(IOS+1)=(ZZ(IOS+3)-ZZ(IOS))/ZZ7
      ZZ(IOS+3)=(S-ZZ(IOS+2))/ZZ7
      ZZ(IOS+12)=ZZ(IOS+12)/ZZ8
      ZZ(IOS+13)=ZZ(IOS+13)/ZZ8
      ZZ(IOS+14)=ZZ(IOS+14)/ZZ7
      ZZ(IOS+15)=ZZ(IOS+15)/ZZ7
C  EVALUATE RATIONAL FUNCTION FROM PRECALCULATED COEFFICIENTS
8     QX = X-ZZ(IOS+4)
      RX = ZZ(IDS+6)-QX
```

```
      QY = Y-ZZ(IOS+5)
      RY = ZZ(IOS+7)-QY
      IF(RX.NE.O.) GO TO 9
      W1 = 1.
      W2 = 1.
      GO TO 10
    9 W1 = 1.-1./(1.+ABS(ZZ(IOS+8)*QX/RX))
      W2 = 1.-1./(1.+ABS(ZZ(IOS+9)*QX/RX))
   10 F1 = ZZ(IOS+12)*(W1+ZZ(IOS+8)*(1.-W1))
      F2 = ZZ(IOS+13)*(W2+ZZ(IOS+9)*(1.-W2))
      Z(2) = ZZ(IOS+6)*(RY*(F1-ZZ(IOS+12))*W1+QY*(F2-ZZ(IOS+13))*W2)
      G1 = RY*F1+QY*F2
      IF(RY.NE.O) GO TO 11
      W1 = 1.
      W2 = 1.
      GO TO 12
   11 W1 = 1.-1./(1.+ABS(ZZ(IOS+10)*QY/RY))
      W2 = 1.-1./(1.+ABS(ZZ(IOS+11)*QY/RY))
   12 F3 = ZZ(IOS+14)*(W1+ZZ(IOS+10)*(1.-W1))
      F4 = ZZ(IOS+15)*(W2+ZZ(IOS+11)*(1.-W2))
      Z(3) = ZZ(IOS+7)*(RX*(F3-ZZ(IOS+14))*W1+QX*(F4-ZZ(IOS+15))*W2)
      G2 = RX*F3+QX*F4
      ZZ2=ZZ(IOS+1)
      ZZ3=ZZ(IOS+2)
      ZZ4=ZZ(IOS+3)
      Z(1) = ZZ(IOS)+(ZZ2+ZZ4*QY-RX*G1)*QX+(ZZ3-RY*G2)*QY
      Z(2) = Z(2)+ZZ2+QY*(ZZ4+RY*(F3-F4))+(QX-RX)*G1
      Z(3) = Z(3)+ZZ3+QX*(ZZ4+RX*(F1-F2))+(QY-RY)*G2
      RETURN
C  CALCULATE COEFFICIENTS FOR BILINEAR INTERPOLATION
   13 IF(IP.EQ.O) GO TO 14
      I=LOCX+IX
      IND=IOS+4
      ZZ(IND)=TBLS(I-1)
      DX=TBLS(I)-ZZ(IND)
      J=LOCY+IY
      IND=IOS+5
      ZZ(IND)=TBLS(J-1)
      DY=TBLS(J)-ZZ(IND)
      IZ=LOCZ+NZ*(IX-1+NX*(IY-1))
      ZZ(IOS)=SHIFT(TBLS(IZ),NSFT)
      IND=IOS+1
      ZZ(IND)=SHIFT(TBLS(IZ+NZ),NSFT)
      ZZ(IND)=(ZZ(IND)-ZZ(IOS))/DX
      IZ=IZ+NZ*NX
      IND=IOS+2
      ZZ(IND)=SHIFT(TBLS(IZ),NSFT)
      ZZ(IND)=(ZZ(IND)-ZZ(IOS))/DY
      IND=IOS+3
      ZZ(IND)=SHIFT(TBLS(IZ+NZ),NSFT)
      ZZ(IND)=(ZZ(IND)-ZZ(IOS)-ZZ(IOS+1)*DX-ZZ(IOS+2)*DY)/(DX*DY)
C  EVALUATE BILINEAR FUNCTION FROM PRECALCULATED COEFFICIENTS
   14 QX = X-ZZ(IOS+4)
      QY = Y-ZZ(IOS+5)
      Z(2) = ZZ(IOS+1)+ZZ(IOS+3)*QY
      Z(3) = ZZ(IOS+2)+ZZ(IOS+3)*QX
      Z(1) = ZZ(IOS)+Z(2)*QX+ZZ(IOS+2)*QY
      RETURN
      END
```

```
      SUBROUTINE T4PTREX(IR,IOT,TBLS,P,T,R,E,IFL)
C-----------------------------------------------------------------
C
C  SUBROUTINE:  T4PTREX(IR,IDT,TBLS,P,T,R,E,IFL)
C
C  PURPOSE:     FIND DENSITY AND INTERNAL ENERGY AS FUNCTIONS
C               OF PRESSURE AND TEMPERATURE FROM A 302
C               SESAME TABLE.  USES DOUBLE BINARY SEARCH TO
C               FIND INITIAL GUESS OF R AND E, THEN USES
C               DOUBLE NEWTONS METHOD.
C
C  ARGUMENTS:   IR   (INPUT) - REGION NO.
C               IOT  (INPUT) - DATA TYPE CORRESPONDING TO E BASED EOS
C               TBLS (INPUT) - TABLE STORAGE ARRAY
C               P    (INPUT) - PRESSURE
C               T    (INPUT) - TEMPERATURE
C               R   (OUTPUT) - DENSITY
C               E   (OUTPUT) - ENERGY
C               IFL (OUTPUT) - ERROR FLAG
C                 IFL=1, IF CALCULATION IS SUCCESSFUL
C                 IFL=0, IF CALCULATION FAILS
C
C  REMARKS:     TBLS CAN BE DECLARED LCM ON THE CDC 7600.
C
C  EXTERNALS:   T4DATIX, ISRCHKX.
C
C  PROGRAMMER:  G. I. KERLEY, T-4.
C
C  DATE:        3 MARCH 1978
C
C-----------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION TBLS(1)
      COMMON/S2DIRX/LCMX,NRS,LCFW(10,3)
      COMMON/SESINX/IRX,IDTX,RX,EX,IBR,IFLX
      COMMON/SESOUTX/ZP(3),ZT(3)
      IBR=0
      IFLX=1
      IRX=IR
      IOTX=IOT
      LOC=LCFW(IR,IDT)+1
      NR = TBLS(LOC+1)
      NT = TBLS(LOC+2)
      ILO = 1
      IHI = NR
1     I = .5*(ILO+IHI)
      LOCT = LOC+2+NR+NR+NT+I+NR
      J = ISRCHKX(T,TBLS(LOCT),NT-2,NR,30)+1
      LOCP = LOC+2+NR+NT+I+NR*(J)
      IF(IHI-I.EQ.1) GO TO 3
      IF(P.LT.TBLS(LOCP)) GO TO 2
      ILO = I
      GO TO 1
2     IHI = I
      GO TO 1
3     RX = TBLS(LOC+2+I)
      EX = TBLS(LOC+2+NR+J)+TBLS(LOC+2+NR+NT+I)
      K = 0
      IFL = 1
4     K = K+1
      IF(K.EQ.50) GO TO 6
      CALL T4DATIX
```

```
      R=RX
      E=EX
      PTEST = ABS(P-ZP(1))-1.E-05*(ABS(P)+1.E-05)
      IF(PTEST.GT.0.) GO TO 5
      TTEST = ABS(T-ZT(1))-1.E-05*(ABS(T)+1.E-02)
      IF(TTEST.LT.0) RETURN
5     DNOMR = ZT(3)*ZP(2)-ZP(3)*ZT(2)
      IF(DNOMR.EQ.0.) GO TO 6
      RX = RX+(ZT(3)*(P-ZP(1))-ZP(3)*(T-ZT(1)))/DNOMR
      DNOMR = ZT(2)*ZP(3)-ZP(2)*ZT(3)
      IF(DNOMR.EQ.0.) GO TO 6
      EX = EX+(ZT(2)*(P-ZP(1))-ZP(2)*(T-ZT(1)))/DNOMR
      GO TO 4
6     IFL = 0
      RETURN
      END
```

```
      SUBROUTINE T4RTPEX(IR,IDT,TBLS,R,T,P,E,IFL)
C-----------------------------------------------------------------------
C
C  SUBROUTINE T4RTPEX(IR,IDT,TBLS,R,T,P,E,IFL)
C
C  PURPOSE     TO FIND PRESSURE AND ENERGY AS FUNCTIONS
C              OF DENSITY AND TEMPERATURE FROM A
C              SESAME TYPE 302 TABLE USING NEWTONS METHOD.
C
C  ARGUMENTS   IR          (INPUT)   REGION NO.
C              IDT         (INPUT)   DATA TYPE FOR 302 TABLES
C              TBLS        (INPUT)   TABLE STORAGE ARRAY
C              T           (INPUT)   TEMPERATURE
C              P           (OUTPUT)  PRESSURE
C              E           (OUTPUT)  ENERGY
C              IFL         (OUTPUT)  OUTPUT FLAG
C                                    =1 FOR SUCCESS
C                                    =0 FOR FAILURE
C
C  REMARKS      NONE
C
C
C  PROGRAMMER  J.ABDALLAH,JR.
C
C  DATE         5 JULY 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,TBLS
      COMMON/S2DIRX/LCMX,NRS,LCFW(10,3)
      DIMENSION TBLS(1)
      COMMON/SESINX/IRXX,IDTX,RX,EX,IBR,IFLX
      COMMON/SESOUTX/ZP(3),ZT(3)
      IBR=0
      IFLX=1
      RX=R
      IRXX=IR
      IDTX=IDT
      LOC=LCFW(IR,IDT)
      NR=TBLS(LOC+2)
      NE=TBLS(LOC+3)
C . . GET INITIAL GUESS ON ENERGY
C . . FIND CLOSEST DENSITY INDEX
      LOCX=LOC+4
      IRX=1
      DELS=ABS(R-TBLS(LOCX))
      IF(NR.EQ.1) GO TO 20
      DO 10 J=2,NR
      LOCX=LOCX+1
      DEL=ABS(R-TBLS(LOCX))
      IF(DEL.GT.DELS) GO TO 10
      IRX=J
      DELS=DEL
   10 CONTINUE
C . . FIND THE ENERGY INDEX ASSOCIATED WITH THE CLOSEST TEMP
   20 LOCX=LOC+3+NR+NE+NR+IRX
      DELS=TBLS(LOCX)
      DELS=SHIFT(DELS,30)
      DELS=ABS(T-DELS)
      IEX=1
      IF(NE.EQ.1) GO TO 40
      DO 30 J=2,NE
      LOCX=LOCX+NR
```

```
63        DEL=TBLS(LOCX)
64        DEL=SHIFT(DEL,30)
65        DEL=ABS(T-DEL)
66        IF(DEL.GT.DELS) GO TO 30
67        IEX=J
68        DELS=DEL
69   30   CONTINUE
70 C . .  INITIAL GUESS ON ENERGY
71   40   EX=TBLS(LOC+3+NR+IEX)+TBLS(LOC+3+NR+NE+IRX)
72 C . .  ITERATE USING NEWTONS METHOD
73        K=0
74        IFL=1
75   50   K=K+1
76        IF(K.EQ.50) GO TO 90
77        CALL T40ATIX
78        E=EX
79        P=ZP(1)
80        TTEST=ABS(T-ZT(1))-1.0E-05*(ABS(T)+1.0E-02)
81        IF(TTEST.LT.0.) RETURN
82        D=-ZT(3)
83        IF(D.EQ.0.0) GO TO 90
84        EX=EX-(T-ZT(1))/D
85        GO TO 50
86   90   IFL=0
87        RETURN
88        END
```

APPENDIX B

CROSS-REFERENCE DIRECTORY OF EOSLIB

NO UNUSED SPACE

	ROUTINE	INDEX	RECORD LENGTH	DSKAD	EXTERNAL SYMBOLS		COMMON BLOCKS	
1.	DPACKX DPACKX	18	15	017113				
2.	EDSBEG EDSBEG	5	7	002354			S2DIRX SESINX EDSCZ EDSC2 EDSC4 EDSC6 EDSCCE	SESDATX INTORDX EDSC1 EDSC3 EDSC5 EDSC7 EDSCCD
3.	EDSCON EDSCON	1	81	000000	EDSDSL	GDTDER.	EDSCCE	EDSCCD
4.	EDSDRE EDSDRE	6	81	005404	T4DATIX EDSKUT	EDSGET	S2DIRX INTORDX SESOUTX EDSC3	SESDATX SESINX EDSC2 EDSC4
5.	EDSDRT EDSDRT	7	80	006173	T4DATX EDSKUT	EDSGET	S2DIRX INTORDX SESOUTX EDSCCE EDSC3	SESDATX SESINX EDSC2 EDSC1 EDSC4
6.	EDSDSL EDSDSL	2	6	000454			EDSCCE	EDSCCD
7.	EDSEFD EDSEFD	4	166	001320	OUTCI. EOF REWIND. EDSFAS	DECODI. INPCI. FEXIST	EDSCZ EDSC2 EDSC5	EDSC1 EDSC3 EDSC6
8.	EDSFAS EDSFAS	3	74	000644	ASSIGN GOTOER.	QASSIGN	EDSCZ EDSC2	EDSC1
9.	EDSGET EDSGET	11	231	011374	OUTCI. GETRPDX GOTOER. EDSOFD	GETEDSX GETINVX EDSCON EDSEFD	S2DIRX SESINX EDSCZ EDSC3	SESDATX INTORDX EDSC1 EDSC4
10.	EDSIPT EDSIPT	9	128	007722	OUTCI. EDSGET	T4PTREX EDSKUT	S2DIRX SESINX EDSCZ EDSC4	SESDATX INTORDX EDSC3
11.	EDSIRT EDSIRT	8	128	007020	OUTCI.	T4RTPEX	S2DIRX	SESDATX

					EDSGET	EDSKUT	SESINX EDSCZ EDSC4	INTORDX EDSC3
12.	EDSKUT EDSKUT	10	123	010624	OUTCI.	EDSBEG	EDSCZ	EDSC3
13.	EDSDFD EDSDFD	12	145	012524	OUTCI. EDF REWIND. EDSFAS	DECDDI INPCI. FEXIST	EDSCZ EDSC2 EDSC7	EDSC1 EDSC3
14.	EDSORT EDSORT	13	106	013464	XTDY$ QLOG1D EDSKUT	T4DATX EDSGET OUTCI.	S2DIRX INTORDX SESOUTX EDSCCD EDSC3	SESDATX SESINX EDSCZ EDSC1 EDSC4
15.	GETEDSX GETEDSX	14	171	014330	OPACKX MATCHKX	TABRANX	S2DIRX	EDSCCE
16.	GETINVX GETINVX	21	201	021107	OPACKX TABRANX	INV3D1X MATCHKX	S2DIRX	EDSCCE
17.	GETRPDX GETRPDX	26	139	025611	OPACKX MATCHKX	TABRANX	S2DIRX	EDSCCD
18.	INBUFRX INBUFRX	17	29	016627	UNIT	RDISK		
19.	INV3D1X INV3D1X	24	166	023711	RATFN1X	ISRCHKX	INTORDX	RTBLK1X
20.	ISRCHKX ISRCHKX	19	55	017303				
21.	MATCHKX MATCHKX	16	22	016437				
22.	RATFN1X RATFN1X	22	128	022237			SESDATX RTBLK1X	INTORDX
23.	T4DATIX T4DATIX	23	124	022770	T4INTPX	RATFN1X	S2DIRX RTBLK2X SESOUTX	RTBLK1X SESINX SESDATX
24.	T4DATX T4DATX	27	130	026551	T4INTPX		S2DIRX SESDATX SESOUTX	RTBLK2X SESINX
25.	T4INTPX T4INTPX	20	289	017702			RTBLK2X SESDATX	INTORDX

26.	T4PTREX T4PTREX	15	156	015403	T4DATIX	ISRCHKX	S2DIRX SESOUTX	SESINX	-
27.	T4RTPEX T4RTPEX	25	128	024613	T4DATIX		S2DIRX SESOUTX	SESINX	-
28.	TABRANX TABRANX	28	188	027566	INBUFRX				

1

NO.	ENTRY PT:	CALLED BY:				
1	EOSCON	EOSGET				
2	EOSDSL	EOSCON				
3	EOSFAS	EOSEFO	EOSOFO			
4	EOSEFO	EOSGET				
5	EOSBEG	EOSKUT				
6	EOSDRE					
7	EOSDRT					
8	EOSIRT					
9	EOSIPT					
10	EOSKUT	EOSORE	EOSORT	EOSIRT	EOSIPT	EOSORT
11	EOSGET	EOSORE	EOSORT	EOSIRT	EOSIPT	EOSORT
12	EOSOFO	EOSGET				
13	EOSORT					
14	GETEOSX	EOSGET				
15	T4PTREX	EOSIPT				
16	MATCHKX	GETEOSX	GETINVX	GETRPOX		
17	INBUFRX	TABRANX				
18	OPACKX	GETEOSX	GETINVX	GETRPOX		
19	ISRCHKX	T4PTREX	INV301X			
20	T4INTPX	T4OATIX	T4OATX			
21	GETINVX	EOSGET				
22	RATFN1X					

	T4DATIX	INV3O1X		
23 T4OATIX	EOSORE	T4PTREX	T4RTPEX	
24 INV3O1X	GETINVX			
25 T4RTPEX	EOSIRT			
26 GETRPOX	EOSGET			
27 T4OATX	EOSORT	EOSORT		
28 TABRANX	GETEOSX	GETINVX	GETRPOX	

1
COMMON BLOCK: USED BY:

EOSCCE	EOSCON	EOSDSL	EOSBEG	EOSORT	GETEOSX	GETINVX		
EOSCCO	EOSCON	EOSDSL	EOSBEG	EOSORT	GETRPOX			
EOSCZ	EOSFAS EOSGET	EOSEFD EOSOFD	EOSBEG EOSORT	EOSORE	EOSORT	EOSIRT	EOSIPT	EOSKUT
EOSC1	EOSFAS	EOSEFD	EOSBEG	EOSORT	EOSGET	EOSOFD	EOSORT	
EOSC2	EOSFAS	EOSEFD	EOSBEG	EOSOFD				
EOSC3	EOSEFD EOSOFD	EOSBEG EOSORT	EOSORE	EOSORT	EOSIRT	EOSIPT	EOSKUT	EOSGET
EOSC5	EOSEFD	EOSBEG						
EOSC6	EOSEFD	EOSBEG						
S201RX	EOSBEG T4PTREX	EOSORE GETINVX	EOSORT T4DATIX	EOSIRT T4RTPEX	EOSIPT GETRPOX	EOSGET T4DATX	EOSORT	GETEOSX
SESDATX	EOSBEG RATFN1X	EOSORE T4DATIX	EOSORT T4DATX	EOSIRT	EOSIPT	EOSGET	EOSORT	T4INTPX
SESINX	EOSBEG T4DATIX	EOSORE T4RTPEX	EOSORT T4DATX	EOSIRT	EOSIPT	EOSGET	EOSORT	T4PTREX
INTORDX	EOSBEG RATFN1X	EOSORE INV3O1X	EOSORT	EOSIRT	EOSIPT	EOSGET	EOSORT	T4INTPX
EOSC4	EOSBEG	EOSORE	EOSORT	EOSIRT	EOSIPT	EOSGET	EOSORT	
EOSC7	EOSBEG	EOSOFD						
SESOUTX	EOSORE	EOSORT	EOSORT	T4PTREX	T4DATIX	T4RTPEX	T4DATX	
RTBLK2X	T4INTPX	T4DATIX	T4DATX					
RTBLK1X	RATFN1X	T4DATIX	INV3O1X					

APPENDIX C

TEST PROGRAM

```
      PROGRAM SES (INPUT,OUTPUT,TAPE6=OUTPUT,TAPE59=TTY,TAPE3)
C
C     ************************************************************
C     *
C     * PURPOSE-
C     *    TO PROVIDE THE APPROPRIATE EOS SCALE FACTORS FOR THE
C     *       SYSTEM OF UNITS CHOSEN BY KUNIT (SEE BELOW)
C     *
C     * INPUT VARIABLES-
C     *
C     *  LMAT = MATERIAL IN AN A10 FIELD, FOR EXAMPLE- LMAT = "HELIUM"
C     *         THE MATERIAL SESAME NUMBER CAN ALSO BE USED TO SPECIFY
C     *         THE MATERIAL BY SETTING LMAT TO THE SESAME
C     *         NUMBER, FOR EXAMPLE- LMAT = "5760" FOR HELIUM
C     *
C     *  R = DENSITY (RHO)
C     *
C     *  E = INTERNAL EN
C     *
C     *  KEOS = MULTIPLE PARAMETER FLAG TO DESCRIBE HOW TO WRITE
C     *         AND RETRIEVE THE DATA FILE. KEOS HAS FOUR DECIMAL DIGITS.
C     *
C     *  KEOS = 1000*KBR + 100*KUNIT + 10*KREPE + KFN  WHERE
C     *
C     *   KBR = COMPUTATIONAL MODE FLAG TO INDICATE WHICH
C     *         QUANTITIES AND THEIR PARTIAL DERIVATIVES ARE TO
C     *         BE CALCULATED AND RETURNED BY THE PACKAGE.
C     *           = 0   COMPUTE PRESSURE AND TEMPERATURE
C     *           = 1   COMPUTE PRESSURE ONLY
C     *           = 2   COMPUTE TEMPERATURE ONLY
C     *
C     *  KUNIT= KIND OF UNITS
C     *     0 (SESAMEE) R-G/CC,T-DEG.K,D-CM**2/G,P-GPA,E-MJ/KG
C     *     1 (CGS) R-G/CC,T-DEG.K,D-CM**2/GM,P-MUBR,E-ERGS/GM
C     *     2 (SIU) R-KG/M**3,T-DEG.K,P-PA,E-J/KG,D-M**2/KG
C     *     3 (HYDROXE) R-G/CC,T-DEG.K,P=MBR,E-MBR*CC/GM,D-CM**2/G
C     *     4 (HYDROXD) R-G/CC,T-KEV,P-MBR,E-MBR*CC/G,D-CM**2/G
C     *     5 (SESAMED) R-G/CC,T-EV,D-CM**2/G,P-GPA,E-MJ/KG
C     *     6 (LASNEX) R-G/CC,T-KEV,D-CM**2/GM,P-JRKS/CC,ED-JRKS/CC
C     *
C     *      LEGEND-
C     *             R = DENSITY
C     *             T = TEMPERATURE
C     *             D = OPACITY
C     *             P = PRESSURE
C     *             E = INTERNAL EN
C     *
C     *             CC = CUBIC CENTIMETER
C     *             CM = CENTIMETER
C     *             DEG. K = DEGREES KELVIN
C     *             EV = ELECTRON VOLT
C     *             G = GRAM
C     *             GPA = GIGA PASCALS
C     *             J = JOULES
C     *             JRKS = JERKS
C     *             KEV = KILO ELECTRON VOLTS
C     *             KG = KILOGRAM
C     *             M = METER
C     *             MBR = MEGABAR
C     *             MUBR = MICROBAR
C     *             PA = PASCAL
C     *
```

```
C     *  KREPE = COMPUTATION FLAG TO INDICATE WHETHER E IS
C     *          IS TO BE REPRESENTED AS EN PER UNIT MASS OR EN
C     *          PER UNIT VOLUME
C     *       0 EN IN UNITS OF EN PER UNIT MASS. FOR EXAMPLE-
C     *          UNITS OF ERGS/GRAM WHEN KUNIT = 1.  THIS IS THE USUAL E.
C     *       1 EN IN UNITS OF EN PER UNIT VOLUME. FOR EXAPMLE-
C     *          UNITS OF ERGS/CM**3 WHEN KUNIT = 1. THIS IS THE EN
C     *          DENSITY RHO*E COMMONLY COMPUTED IN HYDRODYNAMIC
C     *          COMPUTER CODES.
C     *
C     *  KFN = KIND OF FUNTION INTERPOLATION IN THE TABLES
C     *    = 0 RATIONAL APPROXIMATIONS (ACCURATE)
C     *    = 1 BILINEAR APPROXIMATIONS (FAST)
C     *
C     * IMATE = INDICATES WHETHER TO LOAD THE DATA FILE IF
C     *         IT DOES NOT EXIST FOR LMAT OR GO DIRECTLY TO A
C     *         PREVIOUSLY LOADED FILE.
C     *   =0 CHECK IF THE DATA FILE FOR LMAT HAS BEEN LOADED.
C     *      IF NOT, SEARCH FOR THE FILE, COMVERT IT TO THE PROPER UNITS
C     *      AND COPY IT INTO LCM USING THE STANDARD SESAME FORMAT.
C     *   >0 EQUAL TO THE LMAT TABLE NUMBER.  THIS NUMBER WAS RETURNED
C     *      BY THE PACKAGE ON A PREVIOUS CALL TO SUBROUTINE EOSDRE.
C     *      EOSIPT OR EOSIRT.  THIS OPTION IS FASTER THAN IMATE = 0
C     *      BECAUSE IT SKIPS SEARCH IN THE DIRECTOR OF THE
C     *
C     * OUTPUT VARIABLES-
C     *
C     * P = ARRAY OF DIMENSION 3 CONTAINING THE PRESSURE AND
C     *      ITS PARTIAL DERIVATIVES.  THIS ARRAY MUST BE DIMENSIONED
C     *      EVEN IF THE PARTIAL DERIVATIVES ARE NOT COMPUTED.
C     *
C     *  P(1) = PRESSURE
C     *  P(2) = DENSITY DERIVATIVE OF THE PRESSURE (DP/DR)
C     *  P(3) = TEMPERATURE DERIVATIVE OF THE PRESSURE (DP/DE)
C     *
C     * T = ARRAY OF DIMENSION 3 CONTAINING THE TEMPERATURE AND
C     *     ITS PARTIAL DERIVATIVES.  THIS ARRAY MUST BE DIMENSIONED
C     *     EVEN IF THE PARTIAL DERIVATIVES ARE NOT COMPUTED
C     *
C     *  T(1) = INTERNAL TEMPERATURE
C     *  T(2) = DENSITY DERIVATIVE OF THE TEMPERATURE (DT/DR)
C     *  T(3) = EN DERIVATIVE OF THE TEMPERATURE (DT/DE)
C     *
C     * IMATE = INDICATES THE SUCCESS OR FAILURE OF
C     *         LOCATING AND LOADING THE DATA FILE FOR LMAT.
C     *
C     *       = N>0 MATERIAL TABLE NUMBER (SUCCESS)
C     *         0    MATERIAL (LMAT) NOT IN LIBRARY
C     *         -N  (N>1) INSUFFICIENT STORAGE
C     *             THE LCM STORAGE MUST BE INCREADED BY AT LEAST
C     *             N STORAGE LOCATIONS.  SEE THE EOSMOD MANUAL
C     *
C     * REMARKS- THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * SAMPLE DRIVER PROGRAM-
C     *
C     *     PROGRAM TST(OUTPUT)
C     *     DIMENSION P(3),E(3)
C     *     LMAT = "HELIUM"
C     *     R = 0.001
C     *     E = 1.0
C     *     KEOS = 110
```

```
C     *      IMATE = 0
C     *      CALL EOSORE(LMAT,R.E.P,T.KEOS,IMATE)
C     *      PRINT 10,P(1)
C     *   10 FORMAT(" PRESSURE = ",E10.2," MICROBARS")
C     *      CALL EXIT
C     *      END
C     *
C     * OUTPUT VARIABLES- IN THE COMMON BLOCKS EOSCCE AND EOSCCO
C     *   TFACE = TEMPERATURE EOS SCALING FACTOR
C     *   RFACE = DENSITY EOS SCALING FACTOR
C     *   PFACE = PRESSURE EOS SCALING FACTOR
C     *   EFACE = EN EOS SCALING FACTOR
C     *
C     *   TFACO = TEMPERATURE OPACITY SCALING FACTOR
C     *   RFACO = DENSITY OPACITY SCALING FACTOR
C     *   OFACO = OPACITY SCALING FACTOR
C     *
C     *   TBLS = ARRAY FOR STORAGE OF THE EOS TABLES
C     *   LCMX = LENGTH OF THE TBLS ARRAY
C     *   NRS = UPPER BOUND ON THE NUMBER OF MAT REGIONS LCFW(NRS,)
C     *   LCFW = ARRAY USED AS A DIRECTORY BY THE SESAME ROUTINES
C     *   IR = MATERIAL REGION NUMBER
C     *   IRC = IR (DEFINED TO PERMIT SUBROUTINE CALL
C     *   IOS2 = SESAME MATERIAL NUMBER
C     *   TBLS = NAME OF AN ARRAY DESIGNATED FOR THE STORAGE OF TABLES
C     *   LCNT = CRRRENT WORD IN THE ARRAY TBLS
C     *   LU41 = UNIT NUMBER ASSIGNED TO THE SESAME INPUT FILE SES2CL
C     *   LU42 = UNIT NUMBER ASSIGNED TO THE SESAME INPUT FILE SES2L
C     *   LU43 = UNIT NUMBER ASSIGNED TO SES2OP/8 FILES
C     *   LU44 = UNIT NUMBER ASSIGNED TO MIXLIB (MIXTURES)
C     *   LU45 = UNIT NUMBER ASSIGNED TO MIXLIB DIRECTORY.
C     *   KFN = 0 RATIONAL APPROXIMATIONS (ACCURATE)
C     *   KFN = 1 BILINEAR APPROXIMATIONS (FAST)
C     *   ZB (OUTPUT FROM GETINVX) AT. CHARGE,AT. CHG**2,MASS
C     *   IOT = DATA TYPE INDICATOR
C     *   MIO (IO) MATERIAL IO =1 INVERSE TABLES =2 OPACITY
C     *
C     * EXTERNAL FILES TO HANDLE EOS, OPACITIES AND MIXTURES
C     * SES2CL - CLASSIFIED SESAME LIBRARY
C     * SES2L - CUCLASSIFIED SESAME LIBRARY
C     *   SES2OP - OPACITY TABLE FROM T4
C     *   MIXLIB - PRIVATE (EOS,OPC) TABLES CREATED BY MIXB(OR MIXER)
C     *   MIXDIR - DIRECTORY OF MIXTURES ON MIXLIB ( NAME,NO (A10,I3))
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     *   SESAME ROUTINES- S2GET,S2EOS
C     *   SESAME ROUTINES MATCHKX,TABFCHX,INBUFRX,OPACKX,ISRCHKX,
C     *      T4INTPX,GETINVX,RATFN1X,T4DATIX,INV3D1X,T4RTPEX
C     *   SESAME COMMON BLOCKS-S2DIRX,RTBLK2X,SESDATX,SESINX,SESOUTX,INTO
C     *   EOSMOD COMMON BLOCKS- EOSC1,-7
C     *   EOSMOD COMMON (ALSO INSERTED INTO GETINVX) EOSCCE, EOSCCO
C     *
C     * LOCAL VARIABLES-
C     * NONE
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     * EOSMOD COMMON BLOCKS- EOSCCE,EOSCCO
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
```

```
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOO- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT.,LA-8502-M,1980
C     *
C     * OATE- MARCH 6, 1980
C     *
C     ************************************************************
C
      LEVEL 2, TBLS
      COMMON /S20IR/ LCMX, NRS, LCFW(10.3)
      COMMON /SESOAT/ TBLS(11000)
      COMMON /SESIN/ IRC, IOT, OUM(2), KBR, OUM1
      COMMON /INTORO/ KFN
C
C     COMMON BLOCKS FOR THE EOSMOO ROUTINES
      COMMON /EOSC1/ LU41, LU42, LU43, LU44. LU45
      COMMON /EOSC2/ LF41, LF42. LF43, LF44, LF45
      COMMON /EOSC3/ INIT, IROIM, IR(60.3), KUT(60,3)
      COMMON /EOSC4/ NTABLE, NTABLO, IFLP. LCNT
      COMMON /EOSC5/ NMAT, LABMAT(60), IOMAT(60)
      COMMON /EOSC6/ NMCL, LABMCL(60). IOMCL(60)
      COMMON /EOSC7/ NMATO, LABMO(60), IOMATO(60)
C
C     EOSMOO COMMON BLOCKS USEO BY THE MOOIFIEO SESAME ROUTINES
      COMMON /EOSCCE/ TFACE, RFACE, PFACE, EFACE, KREPE
      COMMON /EOSCCO/ TFACO, RFACO. OFACO, KREPO
C
      NAMELIST /INP/ KUNIT,KREP,NLMAT,NBR,NFN,IIPT,IIRT,IORT,IORE.LCMX
     1 ,NRS,OUM,KBR,OUM1,KFN.INIT,IROIM,NTABLE.NTABLO.IFLP.LCNT,NMAT
     2 ,NMCL,NMATO,TFACE,RFACE,EFACE,KREPE,TFACO,RFACO.OFACO,KREPO.RHO
     3 ,TEMP,PR,EN,IRC,IOT,IORT,NGO
      OATA RHO /0.16/, TEMP /5.E4/, PR /19.2/, EN /310./
C
      OATA KUNIT /0/, KREP /0/, NLMAT /2/, NBR /1/, NFN /1/
      OATA IIPT /1/, IORT /1/, IIRT /1/, IORE /1/, IORT /1/, NGO /0/
C
      WRITE (3,210)
      OO 10 I=1,NMAT
      WRITE (3,220) LABMAT(I),IOMAT(I)
   10 CONTINUE
C
      WRITE (3,230)
      OO 20 I=1,NMATO
      WRITE (3,220) LABMO(I),IOMATO(I)
   20 CONTINUE
C
   30 CONTINUE
C**   READ (59,INP)
C
      WRITE (6,160)
C
      OO 60 LMATP=1.NLMAT
      LMAT=6HHELIUM
      LMATO=LMAT
      IF (LMATP.EQ.2) LMAT="5760"
      IF (LMATP.EQ.2) LMATO="15760"
C
      OO 50 KBRP=1,NBR
      KBR=KBRP-1
C
      OO 40 KFNP=1,NFN
```

```
      KFN=KFNP-1
C
      KEOS=1000*KBR+100*KUNIT+10*KREP+KFN
C
      CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,IIPT,IORT,IIRT,IORE,IORT,1
     1 )
C
   40 CONTINUE
   50 CONTINUE
   60 CONTINUE
C
      WRITE (59,170)
C**   READ (59,180) LANS
      IF (LANS.EQ.1HY) GO TO 30
C
      IF (NGO.GT.0) GO TO (70,80,90,100,110,130), NGO
C
C     CHECK THE ERROR MESSAGES
C
C     CHANGE UNITS
   70 CONTINUE
      KEOS=KEOS+100
      CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,IIPT,IORT,IIRT,IORE,IORT,1
     1 )
      KEOS=KEOS-100
C
C     CHANGE TABLE FORMAT FOR THE ENERGY AND OPACITY REPRESENTATION
   80 CONTINUE
      KEOS=KEOS+10
      CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,IIPT,IORT,IIRT,IORE,IORT,1
     1 )
      KEOS=KEOS-10
C
C     NON MATERIAL TEST
   90 CONTINUE
      LMAT="NOMAT"
      CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,IIPT,IORT,IIRT,IORE,IORT,1
     1 )
C
      LMAT="99999"
      CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,IIPT,IORT,IIRT,IORE,IORT,1
     1 )
C
C     NEGATIVE DENSITY TEST
  100 CONTINUE
      R=-RHO
      CALL ES (LMAT,LMATO,R,TEMP,PR,EN,KEOS,IIPT,IORT,IIRT,IORE,IORT,1)
      GO TO 150
C
C     CHECK IF ALL THE MATERIALS ARE AVAILABLE
C
C     EOS MATERIALS CHECK
  110 CONTINUE
      DO 120 I=1,NMAT
C
      LMAT=LABMAT(I)
      CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,0,1,0,0,0,0)
C
      ENCODE (4,190,LMAT) IOMAT(I)
C
      CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,0,1,0,0,0,0)
C   *
```

```
311    120 CONTINUE
312 C
313 C      OPACITY MATERIALS CHECK
314    130 CONTINUE
315        DO 140 I=1,NMATO
316 C
317        LMATO=LABMO(I)
318        CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,0,0.0,0,1.0)
319 C
320        ENCODE (5,200,LMATO) IDMATO(I)
321 C
322        CALL ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,0.1,0.0.0,0)
323    140 CONTINUE
324    150 CONTINUE
325 C
326        CALL EXITA (1)
327 C
328    160 FORMAT ("          KEOS IMATE  R         T          P          E")
329    170 FORMAT (" CONTINUE?")
330    180 FORMAT (A1)
331    190 FORMAT (I4)
332    200 FORMAT (I5)
333    210 FORMAT (" THE EOS MATERIALS ARE")
334    220 FORMAT (1X,A10,I10)
335    230 FORMAT (" THE OPACITY MATERIALS ARE")
336        END
```

```
      SUBROUTINE ES (LMAT,LMATO,RHO,TEMP,PR,EN,KEOS,IIPT,IORT,IIRT,IORE
     1 ,IORT,IWNL)
C
C     ***************************************************************
C     *
C     * PURPOSE-
C     *
C     * INPUT VARIABLE-
C     *
C     * INPUT-OUTPUT VARIABLES IN THE COMMON BLOCKS EOSCC2 AND EOSCCO
C     *
C     * EXTERNALS AND COMMON BLOCKS-
C     *
C     * REMARKS-  THIS SUBROUTINE IS PART OF THE EOSMOD PACKAGE
C     *
C     * PROGRAMMER- J. M. HYMAN AND M. KLEIN, GROUP T-7, LASL
C     *
C     * REFERENCE- J. M. HYMAN, M. M. KLEIN
C     *            EOSMOO- A SUBROUTINE PACKAGE FOR CALCULATING
C     *            EQUATIONS-OF-STATE AND OPACITIES
C     *            LOS ALAMOS SCIENTIFIC LABORATORY RPT..LA-8502-M,1980
C     *
C     * OATE- MARCH 22, 1981
C     *
C     ***************************************************************
C
      OIMENSION R(3), P(3), T(3), E(3)
C
      LEVEL 2, TBLS
      COMMON /S2OIR/ LCMX, NRS, LCFW(10,3)
      COMMON /SESOAT/ TBLS(11000)
      COMMON /SESIN/ IRC, IOT, OUM(2), KBR, OUM1
      COMMON /INTORO/ KFN
C
C     COMMON BLOCKS FOR THE EOSMOO ROUTINES
      COMMON /EOSC1/ LU41, LU42, LU43, LU44, LU45
      COMMON /EOSC2/ LF41, LF42, LF43, LF44, LF45
      COMMON /EOSC3/ INIT, IROIM, IR(60,3), KUT(60,3)
      COMMON /EOSC4/ NTABLE, NTABLO, IFLP, LCNT
      COMMON /EOSC5/ NMAT, LABMAT(60), IOMAT(60)
      COMMON /EOSC6/ NMCL, LABMCL(60), IOMCL(60)
      COMMON /EOSC7/ NMATO, LABMO(60), IOMATO(60)
C
C     EOSMOO COMMON BLOCKS USEO BY THE MOOIFIEO SESAME ROUTINES
      COMMON /EOSCCE/ TFACE, RFACE, PFACE, EFACE, KREPE
      COMMON /EOSCCO/ TFACO, RFACO, OFACO, KREPO
C
      NAMELIST /INP/ KUNIT,KREP,NLMAT,NBR,NFN,IIPT,IIRT,IORT,IORE,LCMX
     1 ,NRS,OUM,KBR,OUM1,KFN,INIT,IROIM,NTABLE,NTABLO,IFLP,LCNT,NMAT
     2 ,NMCL,NMATO,TFACE,RFACE,EFACE,KREPE,TFACO,RFACO,OFACO,KREPO,RHO
     3 ,TEMP,PR,EN,IRC,IOT,IORT
C
      R(1)=RHO
      T(1)=TEMP
      P(1)=PR
      E(1)=EN
C     *************** EOSIPT TEST ****************
C
      IF (IIPT.EQ.0) GO TO 10
      WRITE (6,70)
      WRITE (3,70)
      IMATE=0
```

```
      R(1)=R(2)=R(3)=0.0
      E(1)=E(2)=E(3)=0.0
      CALL EOSIPT (LMAT,P,T,R,E,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
C
      R(1)=R(2)=R(3)=0.0
      E(1)=E(2)=E(3)=0.0
      CALL EOSIPT (LMAT,P,T,R,E,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
   10 CONTINUE
C
C     *************** EOSDRT TEST ***************
C
      IF (IORT.EQ.0) GO TO 20
      WRITE (6,80)
      WRITE (3,80)
      IMATE=0
      P(1)=P(2)=P(3)=0.0
      E(1)=E(2)=E(3)=0.0
      CALL EOSDRT (LMAT,R,T,P,E,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
C
      P(1)=P(2)=P(3)=0.0
      E(1)=E(2)=E(3)=0.0
      CALL EOSDRT (LMAT,R,T,P,E,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
   20 CONTINUE
C
C     *************** EOSIRT TEST ***************
C
      IF (IIRT.EQ.0) GO TO 30
      WRITE (6,90)
      WRITE (3,90)
      IMATE=0
      P(1)=P(2)=P(3)=0.0
      E(1)=E(2)=E(3)=0.0
      CALL EOSIRT (LMAT,R,T,P,E,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
C
      P(1)=P(2)=P(3)=0.0
      E(1)=E(2)=E(3)=0.0
      CALL EOSIRT (LMAT,R,T,P,E,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
   30 CONTINUE
```

```
C
C     *************** EOSORE TEST ****************
C
      IF (IDRE_EQ.0) GO TO 40
      WRITE (6,110)
      WRITE (3,110)
      IMATE=0
      P(1)=P(2)=P(3)=0.0
      T(1)=T(2)=T(3)=0.0
      CALL EOSORE (LMAT,R,E,P,T,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL_NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
C
      P(1)=P(2)=P(3)=0.0
      T(1)=T(2)=T(3)=0.0
      CALL EOSORE (LMAT,R,E,P,T,KEOS,IMATE)
      WRITE (6,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      WRITE (3,60) LMAT,KEOS,IMATE,(R(I),T(I),P(I),E(I),I=1,3)
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATE.LE.0) IMATE=0
   40 CONTINUE
C
C     *************** EOSORT TEST ****************
C
      IF (IORT_EQ_0) GO TO 50
      KOPC=KEOS
      WRITE (6,100)
      WRITE (3,100)
      IMATO=0
      OPACITY=0.0
      CALL EOSORT (LMATO,R,T,OPACITY,KOPC,IMATO)
      WRITE (6,60) LMAT,KOPC,IMATO,R(1),T(1),OPACITY
      WRITE (3,60) LMAT,KOPC,IMATO,R(1),T(1),OPACITY
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATO.LE.0) IMATO=0
C
      OPACITY=0.0
      CALL EOSORT (LMATO,R,T,OPACITY,KOPC,IMATO)
      WRITE (6,60) LMATO,KOPC,IMATO,R(1),T(1),OPACITY
      WRITE (3,60) LMATO,KOPC,IMATO,R(1),T(1),OPACITY
      IF (IWNL.NE.0) WRITE (3,INP)
      IF (IMATO.LE.0) IMATO=0
   50 CONTINUE
C
      RETURN
C
   60 FORMAT (1X,A10,I4,I3,12(1PE9.2))
   70 FORMAT (" EOSIPT")
   80 FORMAT (" EOSORT")
   90 FORMAT (" EOSIRT")
  100 FORMAT (" EOSORT")
  110 FORMAT (" EOSDRE")
      END
```

OUTPUT

```
      KEOS INATE  R         T         P         E
EOSIPT
HELIUM      0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        0.        0.        0.        0.        0.        0.        0.
HELIUM      0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        0.        0.        0.        0.        0.        0.        0.
EOSDRT
HELIUM      0 11 1.59E-01 5.00E+04 1.91E+01 3.10E+02 0.        0.        1.15E+02-1.62E+02 0.        0.        4.73E-04 1.13E-02
HELIUM      0 11 1.59E-01 5.00E+04 1.91E+01 3.10E+02 0.        0.        1.15E+02-1.62E+02 0.        0.        4.73E-04 1.13E-02
EOSIRT
HELIUM      0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        0.        0.        0.        0.        0.        0.        0.
HELIUM      0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        0.        0.        0.        0.        0.        0.        0.
EOSDRE
HELIUM      0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
HELIUM      0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
EOSDRT
HELIUM      0 10 1.59E-01 5.00E+04 1.27E+07
HELIUM      0 10 1.59E-01 5.00E+04 1.27E+07
EOSIPT
5760        0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
5760        0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
EOSDRT
5760        0 11 1.59E-01 5.00E+04 1.91E+01 3.10E+02 0.        1.43E+04 1.15E+02-1.62E+02 0.        8.86E+01 4.73E-04 1.13E-02
5760        0 11 1.59E-01 5.00E+04 1.91E+01 3.10E+02 0.        1.43E+04 1.15E+02-1.62E+02 0.        8.86E+01 4.73E-04 1.13E-02
EOSIRT
5760        0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 0.        0.        0.        8.86E+01 0.        0.
5760        0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 0.        0.        0.        8.86E+01 0.        0.
EOSDRE
5760        0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
5760        0 11 1.59E-01 5.00E+04 1.92E+01 3.10E+02 0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
EOSDRT
5760        0 10 1.59E-01 5.00E+04 1.27E+07
15760       0 10 1.59E-01 5.00E+04 1.27E+07
EOSIPT
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE    1   0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE    0
5760      100 -6 0.       5.00E+04 1.92E+01 0.        0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE    1   0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE    0
5760      100 -8 0.       5.00E+04 1.92E+01 0.        0.        1.43E+04 1.23E+02 0.        0.        8.86E+01 3.82E-02 0.
EOSDRT
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE    1   0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE    0
5760      100 -6 0.       5.00E+04 0.       0.        0.        1.43E+04 0.        0.        0.        8.86E+01 0.        0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE    1   0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE    0
5760      100 -8 0.       5.00E+04 0.       0.        0.        1.43E+04 0.        0.        0.        8.86E+01 0.        0.
EOSIRT
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE    1   0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE    0
5760      100 -6 0.       5.00E+04 0.       0.        0.        1.43E+04 0.        0.        0.        8.86E+01 0.        0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE    1   0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE    0
5760      100 -8 0.       5.00E+04 0.       0.        0.        1.43E+04 0.        0.        0.        8.86E+01 0.        0.
EOSDRE
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE    1   0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE    0
```

```
5760        100 -8 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     1    0
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760        100 -8 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
EOSORT
THE DENSITY =  0.          OR THE TEMPERATURE =  0.          IS NONPOSITIVE
5760        100 -1 0.        0.        0.
THE DENSITY =  0.          OR THE TEMPERATURE =  0.          IS NONPOSITIVE
15760       100 -1 0.        0.        0.
EOSIPT
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        5.00E+04 1.92E+01 0.        0.        0.        0.        0.        0.        0.        0.        0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        5.00E+04 1.92E+01 0.        0.        0.        0.        0.        0.        0.        0.        0.
EOSORT
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        5.00E+04 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        5.00E+04 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
EOSIRT
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        5.00E+04 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        5.00E+04 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
EOSDRE
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
THE UNITS HAVE CHANGED SINCE THE LAST USE OF THE TABLE
THE CURRENT VALUES OF KUNIT AND KREP ARE     0    1
THE PREVIOUS VALUES OF KUNIT AND KREP WERE     0
5760         10 -8 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
EOSORT
THE DENSITY =  0.          OR THE TEMPERATURE =  0.          IS NONPOSITIVE
5760         10 -1 0.        0.        0.
THE DENSITY =  0.          OR THE TEMPERATURE =  0.          IS NONPOSITIVE
15760        10 -1 0.        0.        0.
EOSIPT
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT         0 -1 0.      5.00E+04 1.92E+01 0.        0.        0.        0.        0.        0.        0.        0.        0.
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT         0 -1 0.      5.00E+04 1.92E+01 0.        0.        0.        0.        0.        0.        0.        0.        0.
EOSORT
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT         0 -1 0.      5.00E+04 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT         0 -1 0.      5.00E+04 0.        0.        0.        0.        0.        0.        0.        0.        0.        0.
EOSIRT
```

```
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT        0 -1 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT        0 -1 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSDRE
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT        0 -1 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
MATERIAL LMAT = NOMAT      NOT FOUND
NOMAT        0 -1 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSDRT
THE DENSITY = 0.           OR THE TEMPERATURE = 0.          IS NONPOSITIVE
NOMAT        0 -1 0.       0.       0.
THE DENSITY = 0.           OR THE TEMPERATURE = 0.          IS NONPOSITIVE
15760        0 -1 0.       0.       0.
EOSIPT
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 1.92E+01 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 1.92E+01 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSDRT
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSIRT
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSORE
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSORT
THE DENSITY = 0.           OR THE TEMPERATURE = 0.          IS NONPOSITIVE
99999        0 -1 0.       0.       0.
THE DENSITY = 0.           OR THE TEMPERATURE = 0.          IS NONPOSITIVE
15760        0 -1 0.       0.       0.
EOSIPT
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 1.92E+01 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 1.92E+01 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSDRT
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSIRT
```

```
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       5.00E+04 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSORE
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
UNABLE TO LOCATE MATERIAL 99999
IN SUBROUTINE EOSGET
99999        0 -2 0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
EOSORT
THE DENSITY = 0.          OR THE TEMPERAIURE = 0.          IS NONPOSITIVE
99999        0 -1 0.       0.       0.
THE DENSITY = 0.          OR THE TEMPERATURE = 0.          IS NONPOSITIVE
18760        0 -1 0.       0.       0.
```

☆U.S. GOVERNMENT PRINTING OFFICE:1981-0-576-020/150

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche $3.50 (A01)

Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code
001-025	$ 5.00	A02	151-175	$11.00	A08	301-325	$17.00	A14	451-475	$23.00	A20
026-050	6.00	A03	176-200	12.00	A09	326-350	18.00	A15	476-500	24.00	A21
051-075	7.00	A04	201-225	13.00	A10	351-375	19.00	A16	501-525	25.00	A22
076-100	8.00	A05	226-250	14.00	A11	376-400	20.00	A17	526-550	26.00	A23
101-125	9.00	A06	251-275	15.00	A12	401-425	21.00	A18	551-575	27.00	A24
126-150	10.00	A07	276-300	16.00	A13	426-450	22.00	A19	576-600	28.00	A25
									601-up	†	A99

†Add $1.00 for each additional 25-page increment or portion thereof from 601 pages up.