LA-8209

2

HYDSES: A Subroutine Package for Using Sesame in Hydrodynamic Codes

University of California

LOS ALAMOS NATIONAL LABORATORY

3 9338 00317 8927

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663   Los Alamos. New Mexico 87545

Edited by Lidia G. Morales

Photocomposition by Cleo H. Gutierrez and Diana Ortiz
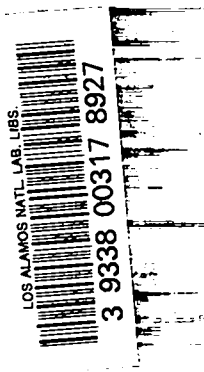
HYDSES: A Subroutine Package for Using Sesame in Hydrodynamic Codes

J. Abdallah, Jr.
G. I. Kerley
B. I. Bennett
J. D. Johnson
R. C. Albers
W. F. Huebner

HYDSES: A SUBROUTINE PACKAGE
FOR USING SESAME IN HYDRODYNAMIC CODES

by

J. Abdallah, Jr., G. I. Kerley, B. I. Bennett,
J. D. Johnson, R. C. Albers, and W. F. Huebner

ABSTRACT

HYDSES is a subroutine package developed for using Sesame equation-of-state and opacity data in hydrodynamic codes. This package is faster, more space efficient, and includes more capability than the previous package. The HYDSES package has been used successfully in hydrodynamic calculations. This report describes the package and its use in hydrodynamic codes.

I. INTRODUCTION

The Sesame library is a file of equation-of-state (EOS) and other material properties that can be accessed and used by application computer programs in need of such data. There also is a set of library-associated subroutines[1,2] that can be used in the application programs to facilitate the retrieval of Sesame data. HYDSES is a new subroutine package including many new features and improvements. This report discusses HYDSES and its use in application programs.

For most cases the use of Sesame data within a code is a two-phase process. Phase one is initialization or setup. This phase consists of retrieving the desired data table from the disk, converting it into the proper format, storing it into a block of core, and saving its location for subsequent use. After this is done, it may be desirable to perform a one-time initial-state computation based on various sets of independent variables. Initialization is generally a one-pass phase that requires a small amount of time compared to the total run time of the application. After initialization the application program is ready

for phase two, the computational phase. Here the application program cycles and enters the Sesame routines that perform interpolation[3] on the Sesame data tables an enormous number of times. The computational phase is usually the time-consuming phase.

A. HYDSES Features

The following features have been incorporated into the HYDSES package.

(1) HYDSES is fast. New computational-phase Sesame subroutines (T4DAT, T4DATI, T4INTP, RATFN1) have been developed to provide increased performance. The new features include the following.

a. Recomputation of interpolation coefficients is avoided if possible.

b. Computational-phase Sesame subroutines all avoid the use of calling sequences. Instead, parameters are now passed through common blocks. This method of programming was adopted to improve computational speed

because of a large percentage of time was spent in address setup for calling parameters.

c. Coding internal to the interpolation package (T4INTP, T4DAT, T4DATI) was optimized.

In addition to the time-saving features introduced into the computational-phase subroutines, random I/O is used in the HYDSES package to speed up the initialization-phase subroutines (see Appendix A). The use of random I/O eliminates the need for time-consuming sequential searches through the Sesame library file. An optional sequential I/O capability has been provided (see Appendix A) for users who encounter difficulties adapting this random I/O method to their computer system.

(2) HYDSES is space efficient. Dependent variables of Sesame tables are double-packed into the data block by initialization routines after being read into core. This reduces the space needed to store a table by a factor just less than two. In addition, HYDSES provides routines (WINDOW, WDWCHK, ISRCH) to window out and verify unwanted portions of a given temperature-based table. These routines can be used to cut down on storage space when only a certain density and temperature range is required by the application program.

(3) HYDSES can accommodate multiple data types. Because types of data other than EOS are becoming available, the package includes new initialization-phase subroutines to access radiative Rosseland mean opacities[4] (GETRPO, GETRTL), Planck mean opacities[4] (GETRPO), total (radiative and thermal conduction) opacities[4] (GETRTL), thermal conductivity of electrons[4] (GETTCE), ion charge (number of free electrons)[4] (GETNFE), and two-temperature equation-of-state tables.[5] These subroutines can be adapted easily into various application programs. They are also written in a modular manner so that if a new data table became available, the associated initialization subroutine would be easy to code.

(4) Several utility subroutines (T4EOSA, T4RTPE, T4PTRE) have been introduced to enhance capabilities when inverted (energy-based) EOS tables are used. Inverted tables are stored so that the dependent variables are P and T and the independent variables are ρ and E, whereas the non-inverted tables are temperature-based and corres-

pond to $P(\rho,T)$ and $E(\rho,T)$. Here ρ, T, P, and E are the thermodynamic quantities of density, temperature, pressure, and energy. The temperature-based tables are obtained by using GETEOS, and the inverted tables are obtained by using GETINV. Temperature-based tables exist in the Sesame library. They are converted to the inverted form by INV301.

Two of the new subroutines (T4RTPE and T4PTRE) are intended primarily for (but not restricted to) initial-state determination when inverted tables are used. T4RTPE computes $P(\rho,T)$ and $E(\rho,T)$, whereas T4PTRE computes $\rho(P,T)$ and $E(P,T)$. A third new subroutine (T4EOSA) may be used in the computational phase to scale the EOS and/or to treat foamed materials and special phase transitions.

B. System-Dependent Features

The package presented here is designed for use on the Los Alamos Scientific Laboratory (LASL) CDC 7600 LTSS computers using the FTN/LTSS compiler and large-core memory (LCM). The system-dependent features of HYDSES include double-packing and random I/O. All random I/O is isolated to subroutine INBUFR, which calls the system-dependent RDISK subroutine. All double-packing and unpacking is performed through the system-dependent in-line shift function that assumes 60-bit words. Use of the shift function is scattered throughout the package, whereas all double-packing is performed through the HYDSES function DPACK. The user should be aware of the Level 2 statements used for LCM declaration.

C. Summary

Table I summarizes all the subroutines appearing in the HYDSES package. Section II describes the initialization-phase subroutines, Sec. III describes the computational-phase subroutines, Sec. IV describes the general use of the package, Sec. V presents a test example, and Sec. VI describes how the HYDSES package may be obtained.

2

TABLE I

HYDSES SUBROUTINES

Subroutine	Type[a]	Phase[b]	Purpose
T4DAT	.U	I,C	Interpolate on T-based tables
,GETRPO	U	I	Get radiative Rosseland/Planck table
GETTCE	U	I	Get thermal conductivity of electrons
,GETNFE	U	I	Get mean ion charge table
MATCHK	A	I	Checks to see if a material has been loaded
TABFCH	A	I	Fetch table from library (random I/O)
INBUFR	A	I	Random I/O Read
DPACK	A	I	Double-packs two real numbers
GETEOS	U	I	Get EOS table
GETNUC	U	I	Get nuclear EOS table
SUBNUC	U	I	Subtract nuclear EOS from EOS
IONEOS	A	I	EOS from Cowan model
WINDOW	U	I	Window T-based tables
ISRCH	A	I	Index search
WDWCHK	U	I	Window check routine
GETRTL	U	I	Get Rosseland/Total table
ISRCHK	U	I	Index search
T4PTRE	U	I	Computes ρ,E from P,T from inverted table
T4INTP	A	I,C	Two-dimensional interpolator
GETINV	U	I	Get inverted EOS table
RATFN1	A	I,C	One-dimensional interpolator
T4DATI	U,A	I,C	Interpolate on inverted table
T4RTPE	U	I	Compute. $P(\rho,T),E(\rho,T)$ from inverted table
INV301	A	I	Inverts 301 table
T4EOSA	U	I,C	Computes a scaled EOS and includes a phase transition
TABSEQ	A	I	Fetch table from library (sequential I/O)
INBUF	A	I	Sequential read

[a]U means that the routine is intended to be called directly by the user program. An A means that the routine is an auxiliary routine called by HYDSES subroutines.

[b]C stands for computational phase, and I stands for initialization phase.

II. INITIALIZATION-PHASE SUBROUTINES

All HYDSES subroutines are used in the initialization phase. However, several of them are intended primarily for use in the computational phase. Detailed descriptions of these subroutines are presented in Sec. III, and their use in the initialization phase is summarized in Sec. II.D.

Section II.A describes the initialization-phase "get" subroutines that are important to all HYDSES users.

Section II.B describes several optional utility subroutines that are useful for "windowing" data tables and for obtaining electronic EOS tables.

Section II.C describes subroutines that provide initial-state computation from inverted tables.

Section II.D describes utility subroutines that are called by those discussed in II.A, II.B, and II.C. Most users will have no need for the material presented in II.D.

A. HYDSES "get" Subroutines

The primary purpose of the HYDSES "get" subroutines is to load data from Sesame library disk files into data arrays. Each "get" subroutine corresponds to different types of data. Only those "get" subroutines that correspond to data required by the user application program need be considered. The internal operation of a typical "get" subroutine is outlined here.

(1) The "get" subroutine checks to see if the requested data were loaded previously. This check is done to avoid loading identical data tables.

(2) The appropriate data are read in from the Sesame library disk file.

(3) The data are transformed to the appropriate data type and/or units. For example, the electron thermal conductivity is computed from the electron conduction opacity. The electron conduction opacity is stored in the Sesame library.

(4) Dependent variables are double-packed.

(5) The location of the loaded data is saved for later use.

Table II summarizes the various "get" subroutines and their associated calling sequences. Here IDT is a data-type number 1, 2, 3, ... etc., assigned by the application program to the various table types. Thus, for one application IDT may be set to 1 for EOS tables, to 2 for Rosseland/Planck opacity tables, and to 3 for inverted EOS tables. The IDT flag is used

later in the computational phase to locate the desired data upon interpolation. This feature allows the use of multiple Sesame table types in a given application program.

Table III summarizes the tables obtained by the HYDSES "get" routines. Simple modifications may be made to these routines to obtain the tables in other desired units. Note that subroutines GETEOS, GETNUC, and GETRPO are coded to give units for the dependent variables that are independent of mass. To obtain the mass-dependent units, the factor of density in the unit conversion of these routines should be eliminated. Note that subroutines GETEOS and GETINV return the normal density (ρ_o) in TBLS (LCNT+1). Table IV shows the detailed format of Sesame data tables in array TBLS upon return from the "get" subroutines. Table V presents the format of the Sesame data tables used by HYDSES as they appear in the Sesame library files. See Appendix A for the Sesame library file structure.

B. Initialization-Phase Utility Subroutines

1. WINDOW. This subroutine is used to reduce the size of a temperature-based data table when the full Sesame density and temperature range is not required by the applications program. If windowing is desired, the corresponding "get" subroutine can be modified easily to call WINDOW. The WINDOW calling sequence is discussed next.

CALL WINDOW (DSTR,RLIM,TZRO,TLIM, LOC,NTAB)

DSTR(INPUT/
OUTPUT)   - LCM data string starting at
               NR (see Table V).
RLIM(INPUT)   - Array of lower and upper
               bounds on density.
TZRO(INPUT)   - Temperature floor. A low
               temperature near TZRO will
               appear in the windowed
               tables if possible.
TLIM(INPUT)   - Array of lower and upper
               bounds on temperature.
LOC(OUTPUT)   - New length of data table. It
               should be used to adjust
               LCNT for the next table in
               the associated "get" routine.

TABLE II

HYDSES "GET" ROUTINE CALLING SEQUENCES

Subroutine	Calling Sequence	Purpose
GETEOS	IR,MID,IDT,TBLS,LCNT,LU,IFL,ZB	Get EOS tables
GETINV	IR,MID,IDT,TBLS,LCNT,LU,IFL,ZB	Get inverted EOS tables
GETNUC	IR,MID,IDT,TBLS,LCNT,LU,IFL	Get nuclear (ion core) EOS tables
GETRPO	IR,MID,IDT,TBLS,LCNT,LU,IFL	Get radiative Rosseland and Planck opacity tables
GETTCE	IR,MID,IDT,TBLS,LCNT,LU,IFL	Get electronic thermal conductivity tables
GETNFE	IR,MID,IDT,TBLS,LCNT,LU,IFL	Get ion charge (number of free electron) tables
GETRTL	IR,MID,IDT,ITOT,TBLS,LCNT,LU,IFL	Get the radiative Rosseland and/or total opacity tables

Calling Parameter Definitions

Parameter	Mode	Description
IR	INPUT	Material region number for which data is to be loaded
MID	INPUT	Sesame material identification number (id) for data to be loaded
IDT	INPUT	Data type to be assigned to loaded data
TBLS	OUTPUT	LCM array where tables are to be stored
LCNT	INPUT/ OUTPUT	Position in TBLS where data are to be stored Position for storing next table
LU	INPUT	Sesame library unit No.
IFL	OUTPUT	Flag. IFL < 0 means that $|IFL|$ extra words are needed in TBLS; IFL $= 0$ means the requested data could not be located; IFL $= 1$ means that the requested data were loaded successfully; IFL $= 2$ means that the data were previously loaded
ZB(1)	OUTPUT	Average atomic number (number weighted mean z)
ZB(2)	OUTPUT	Z'
ZB(3)	OUTPUT	Atomic mass (number weighted mean)
ITOT	INPUT	ITOT $= 0$ for radiative Rosseland table only; ITOT $= 1$ for both radiative Rosseland and total Rosseland opacity tables

TABLE III

HYDSES "GET" ROUTINE SUMMARY

Subroutine	Tables	Units
GETEOS	$P/E(\rho,T)$	P in 10^9 J/cm^3, E in 10^9 J/cm^3, ρ in g/cm^3, T in keV
GETINV	$P/T(\rho,E)$	P in Mbar, E in Mbar·cm^3/g, ρ in g/cm^3, T in K
GETNUC	$P_N/E_N(\rho,T)$	P_N in 10^9 J/cm^3, E_N in 10^9 J/cm^3, ρ in g/cm^3, T in keV
GETRPO	$\kappa_R/\kappa_P(\rho,T)$[a]	κ_R in cm^{-1}, κ_P in cm^{-1}, ρ in g/cm^3, T in keV
GETTCE	$\nu_e/0(\rho,T)$[a]	ν_e in 10^9 J/(keV·cm·10^{-8}s), ρ in g/cm^3, T in keV
GETNFE	$N_f/0(\rho,T)$[a]	N_f dimensionless, ρ in g/cm^3, T in keV
GETRTL	$\kappa_R/\kappa_T(\rho,T)$[a]	κ_R in cm^2/g, κ_T in cm^2/g, ρ in g/cm^3, T in keV

Legend:

P is total pressure,
E is total internal energy,
ρ is density,
T is temperature,
P_N is nuclear pressure,
E_N is nuclear internal energy,
κ_R is radiative Rosseland mean opacity,
κ_P is Planck opacity,
ν_e is thermal conductivity of electrons,
N_f is the number of free electrons/atom, and
κ_T is total Rosseland opacity.

---

[a]All quantities are actually stored as base 10 logarithms f/g.
x/y indicates double-packed dependent variables (y = 0 means
zero is stored in the second half-word).
(x,y) indicates independent variables.
1 jerk = 10^9 J.
1 shake = 10^{-8} s.

TABLE IV

FORMAT OF SESAME TABLES IN ARRAY TBLS[a]

Temperature-Based Tables

Index	Description
LCNT	Material id
LCNT+1	ρ_0 for EOS or inverted EOS; IDT otherwise
LCNT+2	Number of densities (NR)
LCNT+3	Number of temperatures (NT)
LCNT+4	NR densities
LCNT+4+NR	NT temperatures
LCNT+4+NR+NT	NR*NT double-packed dependent variables (see Table III)

Energy-Based EOS Tables

Index	Description
LCNT	Material id
LCNT+1	ρ_0
LCNT+2	Number of densities (NR)
LCNT+3	Number of energies (NE)
LCNT+4	NR densities
LCNT+4+NR	NE energy differences $[E(\rho_0,T_i) - E(\rho_0,T_1), i = 1,NE]$
LCNT+4+NR+NE	NR low-temperature energies $[E(\rho_i,T_1), i = 1, NR]$
LCNT+4+NR+NE+NR	NE*NR double-packed P/T values

[a]All values are stored as floating-point numbers.

TABLE V

FORMAT OF SESAME TABLES IN LIBRARY

Table 201 - Basic Data

Word #	Description
1	Average atomic number (number weighted mean z)
2	Mean atomic mass (a)
3	Normal (solid) density ρ_0
4	Solid bulk modulus (= 0 for material id's > 9999)
5	Exchange coefficient (= 0 for material id's > 9999)

Table 301 - EOS Tables

Word #	Description
1	Number of densities (NR)
2	Number of temperatures (NT)
3	NR densities (Mg/m^3)
NR+3	NT temperatures (K)
NR+NT+3	NR*NT pressures (GPa): ordered with density index increasing fastest
NR+NT+NR*NT+3	NR*NT energies (MJ/kg): ordered with density index increasing fastest

Table 502 - Radiative Rosseland/Planck Opacity Table

Word #	Description
1	Number of densities (NR)
2	Number of temperatures (NT)
3	NR ($\log_{10}$) densities (g/cm^3)
3+NR	NT ($\log_{10}$) temperatures (eV)
3+NR+NT	NR*NT double-packed ($\log_{10}$) Rosseland and Planck opacities (cm^2/g) ordered with density index increasing fastest

Table 503 - Electron Conduction Opacity Table

Word #	Description
1	Number of densities (NR)
2	Number of temperatures (NT)
3	NR ($\log_{10}$) densities (g/cm^3)
3+NR	NT ($\log_{10}$) temperatures (eV)
3+NR+NT	NR*NT ($\log_{10}$) electron conductive opacities (cm^2/g) ordered with density index increasing fastest

Table 504 - Ion Charge (Number Of Free Electrons) Table

Word #	Description
1	Number of densities (NR)
2	Number of temperature (NT)
3	NR ($\log_{10}$) densities (g/cm^3)
3+NR	NT ($\log_{10}$) temperatures (eV)
3+NR+NT	NR*NT ($\log_{10}$) mean ion charges (number of free electrons/atom) ordered with density index increasing fastest

Notes: Table 301 appears for material id's ≤ 9999 only.

Tables 502, 503, and 504 appear for material id's ≥9999 only.

In general, the opacity material id = 10000 + Sesame EOS material id.

NTAB(INPUT) - Number of data tables appearing in the data string. (A double-packed table counts as one).

WINDOW can be used to reduce storage of inverted tables if applied before inversion occurs in GETINV. Note that all densities and temperatures should be in the units appearing in the data string, and they should be logarithmic if the table is logarithmic.

2. WDWCHK. This subroutine is used to see if a temperature-based table falls within user-specified limits.

CALL WDWCHK(NRN,IDT,TBLS,RWDW, TWDW,RTBL,TTBL,IWDW)

NRN(INPUT)	- Material region number.
IDT(INPUT)	- Data type of temperature-based table.
TBLS(INPUT)	- LCM table-storage area.
RWDW(INPUT)	- Array of lower and upper density limits.
TWDW(INPUT)	- Array of lower and upper temperature limits.
RTBL(OUTPUT)	- Lower and upper density limits of table.
TTBL(OUTPUT)	- Lower and upper temperature limits of table.
IWDW(OUTPUT)	- = 0 if RWDW or TWDW fall within table limits and = 1 if RWDW or TWDW are outside table limits.

This routine may be used any time after the table is loaded.

3. SUBNUC. This subroutine is used when two-temperature equations of state are used.[5] If, for example, the user wants separated electronic and nuclear (ion core) EOS tables, GETEOS should be used to get the total EOS table, GETNUC should be used to get the nuclear EOS table, and then SUB-NUC may be used to obtain the electronic EOS. The electronic EOS is obtained by subtracting the nuclear contribution from the total EOS. SUBNUC replaces the total EOS table by the electronic EOS table. If SUBNUC finds no nuclear table available, the Cowan nuclear model[6] is used to perform the subtraction. Note that SUBNUC assumes for all tables involved that density is in grams per cubic centimeter, temperature is in kiloelectron volts, pressure is in 10^9 Joules per cubic centimeter, and energy is in 10^9 Joules per cubic centimeter (not per unit mass). GETEOS, GETNUC, and SUBNUC must all be modified to get tables in other units. SUBNUC has the following calling sequence.

CALL SUBNUC(IR,MID,IDTE,IDTN, TBLS,IFLG)

IR(INPUT)	- Material region number.
MID(INPUT)	- Sesame material identification number (id).
IDTE(INPUT)	- Data-type indicator for electronic EOS; a total EOS table must be loaded under this data type before use by SUBNUC.
IDTN(INPUT)	- Data-type indicator for nuclear EOS table.
TBLS(INPUT/ OUTPUT)	- Table-storage array (LCM).
IFLG(OUTPUT)	- = 0 if total EOS table is missing, = 1 if nuclear table is subtracted, and = 2 if Cowan model is subtracted.

SUBNUC assumes the availability of COMMON/PAUL/AMASS(20) where AMASS elements contain the atomic mass and the atomic number double-packed (in that order) for each region. Thus, the atomic mass for region five is stored in the first 30 bits of AMASS(5) and the atomic number is stored in the last 30 bits.

9

C. Initial-State Computation Subroutines

The following subroutines may be used for initial-state computation from inverted EOS tables. Although the main function of these routines is initial-state computation, various applications may find other uses for them. Because both routines use an iterative scheme, they enter the Sesame interpolation package numerous times causing execution time to be slow. Hence, these routines are not recommended for extensive use. Note that if scaling, ramps, and special-phase transitions are not required by the application, the calls to T4EOSA in T4RTPE and T4PTRE may be replaced by calls to T4DATI (see Sec. III).

1. T4RTPE. This subroutine computes $P(\rho,T)$ and $E(\rho,T)$ and has the following calling sequence.

CALL T4RTPE(IR,IDT,TBLS,R,T,P,E,IFL)

IR(INPUT)	- Material region number.
IDT(INPUT)	- Data-type indicator for inverted EOS table.
TBLS(INPUT)	- LCM array where Sesame tables are stored.
R(INPUT)	- Density.
T(INPUT)	- Temperature.
P(OUTPUT)	- Pressure.
E(OUTPUT)	- Energy.
IFL	- Output flag = 1 for success and = 0 for failure.

2. T4PTRE. This subroutine computes $\rho(P,T)$ and $E(P,T)$ and has the following calling sequence.

CALL T4PTRE (IR,IDT,TBLS,P,T,R,E,IFL)

IR(INPUT)	- Material region number.
IDT(INPUT)	- Data type for inverted EOS tables.
TBLS(INPUT)	- LCM array where Sesame tables are stored.
P(INPUT)	- Pressure.
T(INPUT)	- Temperature.
R(OUTPUT)	- Density.
E(OUTPUT)	- Energy.
IFL(OUTPUT)	- Flag = 1 for success and = 0 for failure.

D. Auxiliary Subroutines

The following are auxiliary subroutines used by the HYDSES initialization-phase subroutines. Each subroutine will be presented with a brief statement of its purpose and a description of its calling sequence. The HYDSES user need not be familiar with them.

1. MATCHK. This subroutine checks to see if a data table for a given material has been loaded previously.

CALL MATCHK(MID,NRS,LOC,TBLS,IFLG)

MID(INPUT)	- Material id.
NRS(INPUT)	- Number of regions.
LOC(INPUT)	- Array of first-word locations for table storage.
TBLS(INPUT)	- Table-storage array (LCM).
IFLG(OUTPUT)	- IFLG = 0 if the material has not been loaded. IFLG gives the location of the table if it has been loaded.

2. TABFCH. This subroutine is used to fetch tables from the Sesame library using random I/O.

CALL TABFCH(MID,TID,LIB,A,LEN,IFLAG)

MID(INPUT)	- Material id.
TID(INPUT)	- Floating-point table id.
LIB(INPUT)	- Library unit number.
A(OUTPUT)	- LCM array for table storage.
LEN(INPUT)	- Length of A.
IFLAG (OUTPUT)	- Flag = 0 for table not found, > 0 for number of words in table returned, and < 0 for minus the number of extra words needed to store the table.

If TID = 0.0, the material index is returned. See Appendix A for a more detailed discussion.

3. INBUFR. This subroutine is used to perform the random read operation.

CALL INBUFR(LU,Z,NW,IAD,IFLG)

LU(INPUT)	- Library unit number.
Z(OUTPUT)	- LCM array where data are to be stored.
NW(INPUT)	- Number of words to be read.
IAD(INPUT)	- Starting disk address of data to be read.
IFLG(OUTPUT)	- Flag = 0 if end of file is encountered, = 1 if data are read in normally, and = -1 if error is encountered.

Note: This version of INBUFR uses the LASL/LTSS/FTN random I/O routine RDISK. Thus, INBUFR needs to be modified for use in other systems.

4. DPACK. This function is used to double-pack floating-point numbers into one word.

DPACK(A,B)

A(INPUT)	- Floating-point number.
B(INPUT)	- Floating-point number.
DPACK(OUTPUT)	- Word packed with A in first 30 bits, and B in last 30 bits.

Note that DPACK assumes the use of 60-bit computer words and uses the system-dependent SHIFT function.

5. IONEOS. This subroutine is used to compute the ionic contribution to the equation of state. See Ref. 6 for a discussion of its use.

6. ISRCH. This function is used to find an index associated with a value in an array.

ISRCH(X,TBLS,N)

X(INPUT)	- Value to be located.
TBLS(INPUT)	- Array to be searched (LCM).
N(INPUT)	- Number of values to be searched.
ISRCH(OUTPUT)	- = I if TBLS(I) ≤ X < TBLS(I+1) and

= 0 or N if X is outside table.

Note that values in TBLS must be contiguous and in ascending order. A more flexible search routine (ISRCHK) is discussed next.

7. ISRCHK. This function is used to find an index associated with a value in an array.

ISRCHK(X,TBLS,N,K,NSFT)

X(INPUT)	- Value to be located.
TBLS(INPUT)	- LCM array to be searched.
N(INPUT)	- Number of values to be searched.
K(INPUT)	- Spacing between values.
NSFT(INPUT)	- Number-of-bits array values are to be shifted.

Note that array TBLS may be either in ascending or descending order; array values need not be contiguous, and double-packed words may be interrogated.

8. INV301. This subroutine is used in subroutine GETINV to convert a temperature-based EOS table into an energy-based EOS table.

CALL INV301(DSTR,LOC,RO,LDS)

DSTR(INPUT)	- LCM table-storage array.
LOC(INPUT)	- Location of Sesame-type 301 table in DSTR.
RO(INPUT)	- Approximate solid density.
LDS(OUTPUT)	- Length of inverted table.

Note that the inverted table replaces the 301 table.

9. T4DAT, T4INTP, RATFN1, T4DATI, and T4EOSA. Although these subroutines are involved in the initialization phase, the major use of these routines is in the computational phase. These computational-phase routines are discussed in the next section. Table VI summarizes their use in the initialization phase.

10. TABSEQ. This subroutine is used to fetch tables from the Sesame library using sequential I/O. The TABSEQ calling sequence is identical to that of TABFCH discussed in Sec. II.D.2. See Appendix A for more details.

TABLE VI

USE OF COMPUTATIONAL-PHASE SUBROUTINES IN THE INITIALIZATION PHASE

Subroutine	Use in Initialization Phase
T4DAT	Called by SUBNUC to interpolate nuclear EOS to total EOS grid
T4EOSA	Used by T4RTPE and T4PTRE for initial-state computation
T4DATI	Called by T4EOSA
T4INTP	Called by T4DAT and T4DATI
RATFN1	Used by INV301 to invert temperature-based tables; also called by T4DATI

11. **INBUF.** This subroutine is used to perform the binary sequential read operation.

CALL INBUF (LU,Z,N,IM)

LU(INPUT)	- Library unit number.
Z(OUTPUT)	- LCM array where data are stored.
N(INPUT)	- Number of words to be read.
IM(OUTPUT)	- Flag

= 0 if end of file is encountered and
= 1 if data are read normally.

INBUF uses the FORTRAN binary read and end-of-file test. The end-of-file test varies from system to system. Hence, INBUF may need to be modified for use in other systems.

III. COMPUTATIONAL-PHASE SUBROUTINES

The computational-phase subroutines, T4DAT, T4DATI, and T4EOSA are intended for direct use by the application program. Subroutines RATFN1 and T4INTP are interpolation routines. T4DAT calls the two-dimensional interpolator (T4INTP) to perform temperature-based interpolations, whereas T4DATI uses both the one-dimensional interpolator RATFN1, and T4INTP for inverted EOS computation. T4EOSA is optional and is included to add a higher level of capability when inverted EOS tables are used. Each of these routines has no calling sequence, therefore all parameters are passed in common blocks. The detailed descriptions of the subroutines (given below) include their purpose and common block I/O descriptions. Note that all the computational-phase routines assume that the LCM table-storage array TBLS appears in COMMON/SESDAT/TBLS($TDIM$), where $TDIM$ is the dimension of TBLS.

A. T4DAT

Subroutine T4DAT performs interpolation on temperature-based tables. Input is passed through common block /SESIN/ and output through /SESOUT/.

COMMON/SESIN/IR,IDT,X,Y,IBR,IFL

IR	- Material region number.
IDT	- Data-type indicator.
X	- Density.
Y	- Temperature.
IBR	- Flag
	0 = returns both dependent variables,
	1 = returns only first dependent variable, and
	2 = returns only second dependent variable.
IFL	- not used.

COMMON/SESOUT/Z1(3),Z2(3)

$Z1(1)$ = first dependent variable $Z1 = Z1(x,y)$,
$Z1(2)$ = $\partial Z1/\partial x$,
$Z1(3)$ = $\partial Z1/\partial y$,
$Z2(1)$ = second dependent variable $Z2 = Z2(x,y)$,
$Z2(2)$ = $\partial Z2/\partial x$, and
$Z2(3)$ = $\partial Z2/\partial y$.

The region and data-type numbers must be valid, that is, data tables must be loaded for the given IR and IDT. Table III lists the available dependent variables. Electronic pressure and energy (P_e, E_e) are available if SUBNUC is used. For certain tables (see Table III), $X = \log_{10}(\rho)$ and $Y = \log_{10}(T)$ should be input, and $\log_{10} \kappa_R$, $\log_{10} \kappa_P$, $\log_{10}\nu_e$, $\log_{10} N_f$, and $\log \kappa_T$ are returned.

B. T4DATI

Subroutine T4DATI performs interpolation on inverted equation-of-state tables. Input is passed through common block /SESIN/ and output through /SESOUT/.

COMMON/SESIN/IR,IDT,X,Y,IBR,IFL

IR	- Material region number.
IDT	- Data-type indicator for inverted tables.
X	- Density.
Y	- Internal energy.
IBR	- Flag
	0 = returns both P and T,
	1 = returns P only, and
	2 = returns T only.
IFL	- Not used.

COMMON/SESOUT/Z1(3), Z2(3)

$Z1(1)$ - $P(x,y)$,
$Z1(2)$ - $\partial P/\partial x$,
$Z1(3)$ - $\partial p/\partial y$,
$Z2(1)$ - $T(x,y)$,
$Z2(2)$ - $\partial T/\partial x$, and
$Z3(3)$ - $\partial T/\partial y$.

C. T4EOSA

Subroutine T4EOSA is optional and provides an added level of capability over T4DATI.

T4EOSA requires as input the /SESIN/ block as described for T4DATI (except here the IFL flag has meaning) and the following common block.

COMMON/EOSCOM/SR($NREG$),ES ($NREG$),RO($NREG$),A1($NREG$), A2($NREG$), A3($NREG$),EM($NREG$).

Here, $NREG$ is the dimension used for regional arrays. SR, ES, RO, A1, A2, A3, and EM are supplied on a region basis and are discussed below in conjunction with the features provided by T4EOSA. Output from T4EOSA is returned in /SESOUT/ as discussed for T4DATI. Because of the scaling features of T4EOSA, the values of x and y (in /SESIN/) upon exit may differ from those values when the routine was entered. T4EOSA is used to provide the following features for inverted EOS computation.

(1) Density and energy scaling.

$\quad$ SR = density scale factor, and

$\quad$ ES = energy shift in user units.

Using these parameters, the EOS is scaled according to the following relations.

$$P(\rho, E) = P_{TAB}(\rho_T, E_T),$$

$$\rho_T = (SR)\rho, \text{ and}$$

$$E_T = (E + ES)/SR,$$

where P, ρ, and E are the pressure, density, and energy variables used by the code, and $P_{TAB}(\rho_T, E_T)$ is the tabular EOS.

The parameter SR is useful for treating isotopic mixtures. If A_T is the atomic mass for the EOS table, an EOS for an atomic mass A is obtained by setting

$$SR = A_T/A \quad .$$

For example, set SR = 2 to scale the Sesame D_2 EOS, #5263, to H_2. Similarly, set SR = 0.80 to obtain an EOS for a 50:50 DT mixture.

The parameter ES can be used to change the energy zero of the table. It is intended for use primarily with the "ramp" option, discussed in (2).
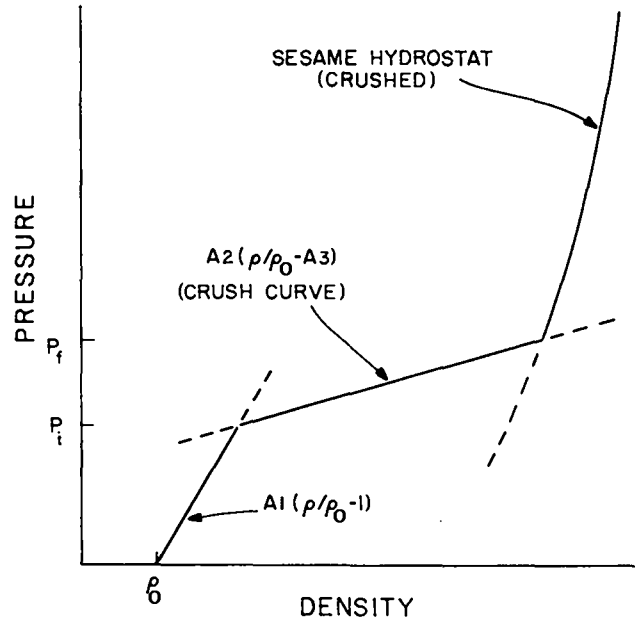

Fig. 1.

Sesame option for foams and phase transitions.

(2) Foams and phase transitions.

A1, A2, A3	= ramp parameters,
ES	= energy shift in user units,
ES	= energy shift in user units,
EM	= "melt" energy in user units, and
RO = ρ_0	= initial density.

For treatment of foams and certain types of phase transitions, it is possible to modify the Sesame EOS by adding a "ramp" that describes the behavior of the material at low stress. This ramp is shown in Fig. 1.

The material starts out in either a porous state or low-density phase. The EOS is given by

$$P = A1 \ (\rho/\rho_0 - 1) \ , \tag{1}$$

where ρ_0 is the initial density and A1 is the bulk modulus. A1 can be computed from

$$A1 = \rho_0 c_0^2 \ (GPa) \quad ,$$

where C_0 is the bulk sound speed (in km/s). If A1 = 0, no ramp calculation is performed.

At some pressure P_1, the material will begin to "crush," or transform to the high-density phase

described by the Sesame EOS. The EOS of the crush curve is

$$P = A2 \, (\rho/\rho_o - A3) \quad .$$

(If A2 = 0, there is no crush region and Eq. (1) is continued on until it crosses the Sesame hydrostat.) A2 is related to the transition pressure by

$$A2 = \frac{P_i A1}{P_i + A1(1 - A3)} \quad \text{(user units)}$$

The transition pressure for foams is usually rather small (<1 kbar). For a phase transition, P_i must be obtained from experiment. The parameter A3 can be adjusted to give the correct slope of the crush curve. In the absence of data, the default value (A3 = 0) should give acceptable results.

At some pressure P_t, the ramp crosses the Sesame hydrostat. At that point, the substance is said to be "crushed." Subsequently, the material may behave either reversibly (follow the ramp on expansion) or irreversibly (stay in the high-density phase on expansion). Foams normally are irreversible, but phase transitions may be either. The behavior is controlled by setting the IFL flag appropriately. If IFL = 0 before T4EOSA is entered, the code will choose between the ramp and the tables for EOS values. It will return IFL = 0 if the ramp was used, and IFL = 1 if the tables were used. If IFL = 1 before T4EOSA is entered, the code will always go to the tables for the EOS. Thus, if IFL = 0 upon each entry to T4EOSA, the transition is reversible. If IFL=0 initially and not reset by the user's code, then the transition is irreversible. The material will also behave irreversibly if it melts, that is, if the melt energy is exceeded. EM depends on the path and must be treated carefully. If EM is sufficiently high, "melting" will not occur in most cases of interest.

The energy shift is defined by

$$ES = -\Delta E \ (\text{initial} \rightarrow \text{final}) \quad ,$$

where ΔE is the energy required to transform the low-density phase to the high-density phase. Hence, there are two cases. If the initial phase is *stable*, ES is *negative*. If the initial phase is *metastable*, ES is *positive*.

(3) Recommended default values.

Recommended default values for parameters appearing in /EOSCOM/ are

SR = 1.0,
ES = 0,
RO = normal solid density (g/cm^3),
A1 = 0,
A2 = 0,
A3 = 0, and
EM = 1000.0 Mbar·cm^3/g.

D. RATFN1

Subroutine RATFN1 is the one-dimensional interpolation routine. RATFN1 communicates with other subroutines using the following common blocks.

COMMON/INTORD/IFN
COMMON/RTBLK1/LOCX,KX,LOCY,KY,I, N,IP,X,Y(2)

where

IFN(INPUT)	-	Integration type IFN = 0 for rational function interpolation and IFN = 1 for linear interpolation
LOCX(INPUT)	-	Location of x vector in array TBLS.
KX(INPUT)	-	Spacing of x vector.
LOCY(INPUT)	-	Location of y vector in TBLS.
KY(INPUT)	-	Spacing of y vector.
I(INPUT)	-	Index into x and y vector corresponding to requested x value.
N(INPUT)	-	Length of x and y vectors.
IP(INPUT)	-	Flag = 0 if coefficients from previous call are to be used and = 1 if coefficients are to be recalculated.
X(INPUT)	-	Value of x for which y is requested.
Y(OUTPUT)	-	Y(1) = y(x) and Y(2) = $\partial y/\partial x$.

E. T4INTP

Subroutine T4INTP is the two-dimensional interpolation subroutine. T4INTP communicates with subroutines calling it through the following common blocks.

COMMON/INTORD/IFN
COMMON/RTBLK2/LOCX,IX,NX,LOCY,
IY,NY,LOCZ,NZ,NSFT,X,Y,Z(3),IP,IDS,
ZZ($CDIM$)

IFN(INPUT)	- = 0 for rational function interpolation and = 1 for bilinear interpolation.
LOCX(INPUT)	- Location of the x vector in array TBLS.
IX(INPUT)	- Index into x vector corresponding to x value.
NX(INPUT)	- Length of x vector.
LOCY(INPUT)	- Location of the y vector in array TBLS.
IY(INPUT)	- Index into y vector corresponding to y value.
NY(INPUT)	- Length of y vector.
LOCZ(INPUT)	- Location of dependent variable array Z(x,y) in array TBLS.
NZ(INPUT)	- Spacing of values in Z array.
NSFT(INPUT)	- Number of bits that dependent variables are shifted when accessed.
X(INPUT)	- Independent variable, x-value.
Y(INPUT)	- Independent variable, y-value.
Z(OUTPUT)	- Array of length 3 where $Z(1) = Z(x,y)$, $Z(2) = \partial Z/\partial x$, and $Z(3) = \partial Z/\partial y$.
IP(INPUT)	- Flag = 0 use previous interpolation coefficients and = 1 recompute interpolation coefficients.
IDS(INPUT)	- Index into array ZZ where coefficients are to be stored or located.
ZZ(INPUT)	- Interpolation coefficient array of dimension $CDIM$.

IV. GENERAL USE OF HYDSES

This section describes how to install HYDSES into an applications program.

A. Preliminaries

The user should determine
(1) the maximum number of regions ($NREG$) needed by the application for dimensioning purposes.
(2) the maximum number of Sesame data types ($NDAT$) needed by the application for dimensioning purposes.
(3) the amount of LCM table storage required for dimensioning purposes. (The user should allow approximately 3000 words for an EOS table and 1550 words for an opacity table.)
(4) the interpolation type to be used (see Sec. III.D, III.E).

B. Specifications

The following must be specified in the user's code before HYDSES subroutines are called.

(1) The Sesame (in-core) directory common block

COMMON/S2DIR/LCMX,NRS,LCFW ($NREG$,$NDAT$)

Set LCMX to the number of words available for table storage; NRS to the number of regions in the problem ($\leq$ $NREG$), and LCFW($NREG$, $NDAT$) to all zeroes. After initialization, LCFW will contain the first-word location in the storage array of the table to be used for each region and data type.

(2) The Sesame LCM common block for table storage

COMMON/SESDAT/TBLS($TDIM$)

(3) The interpolation-type common block

COMMON/INTORD/IFN

Note that changing IFN during the computational phase may cause problems. Set IFN as discussed in Secs. III.D and III.E.
(4) Initialize the table-storage-pointer word (LCNT) to 1.

(5) Assign unit numbers to the Sesame data files to be used.

C. Replacements and Selection of Subroutines

The user should then replace (using a text editor) all occurrences of $NREG$, $NDAT$, and $TDIM$ in HYDSES by those chosen in Sec. IV.A. Also, $C-DIM$ should be replaced with the product of $NDAT$ and 32. The user should then determine which of the HYDSES subroutines are needed. The "get" routines should be checked to see if they yield tables in the appropriate units (see Tables III and IV).

D. Initialization Phase

Now, using the subroutines described in Sec. II, the user may code the initialization phase appropriate to the application.

E. Computational Phase

After initialization is complete, the routines described in Sec. III may be used for computational-phase coding.

17

V. TEST EXAMPLE

The following example (Fig. 2) illustrates the use of the HYDSES package. The code used to generate the example is shown in Appendix B. The purpose of the code is to compute P, E, κ_R, κ_p, ν_θ, and N_f as functions of ρ and T. The user is first prompted for the number of regions and a set of associated material numbers. After these have been entered, the program enters an initialization phase where the desired Sesame tables are loaded from the disk libraries for each region. In the example that follows, three regions are loaded with data; the first with deuterium, the second with aluminum, and the third with iron. After the tables are loaded, the program enters a "mini" computational phase. Here the user is prompted for region number, density, and temperature. Note that the units correspond to those in Table III. After these values are entered, the dependent variables are computed in the same manner they would be in an applications program. After the program prints out the dependent variables at the terminal, the user is prompted for further values of region, density, and temperature. This sequence continues until the user types in 0 0 0.

18

```
HYDTX / 1 1.6
 ENTER THE NUMBER OF REGIONS
? 3
 ENTER SESAME EOS MATERIAL NUMBERS
? 5263 3710 2140
 ENTER REGION,DENSITY,TEMPERATURE
? 1 8.5 .7
 PRESSURE   =    5.7322E-01
 ENERGY     =    8.6419E-01
 ROSS OP    =    2.8531E+00
 PLNK OP    =    2.6046E+01
 TH COND    =    1.3743E-05
 NO FR E    =    1.0000E+00
 ENTER REGION,DENSITY,TEMPERATURE
? 2 8.5 .7
 PRESSURE   =    2.7150E-01
 ENERGY     =    5.0821E-01
 ROSS OP    =    6.5492E+02
 PLNK OP    =    3.1040E+03
 TH COND    =    4.6022E-06
 NO FR E    =    1.2248E+01
 ENTER REGION,DENSITY,TEMPERATURE
? 3 8.5 .7
 PRESSURE   =    2.2553E-01
 ENERGY     =    4.9409E-01
 ROSS OP    =    3.9797E+02
 PLNK OP    =    3.3300E+03
 TH COND    =    3.4905E-06
 NO FR E    =    2.2048E+01
 ENTER REGION,DENSITY,TEMPERATURE
? 1 4.25 .35
 PRESSURE   =    1.4288E-01
 ENERGY     =    2.1662E-01
 ROSS OP    =    3.0243E+00
 PLNK OP    =    7.0083E+01
 TH COND    =    2.8165E-06
 NO FR E    =    1.0000E+00
 ENTER REGION,DENSITY,TEMPERATURE
? 2 4.25 .35
 PRESSURE   =    6.1422E-02
 ENERGY     =    1.2676E-01
 ROSS OP    =    8.4304E+02
 PLNK OP    =    5.3521E+03
 TH COND    =    1.1244E-06
 NO FR E    =    1.0854E+01
 ENTER REGION,DENSITY,TEMPERATURE
? 3 4.25 .35
 PRESSURE   =    4.6630E-02
 ENERGY     =    1.1895E-01
 ROSS OP    =    3.0076E+03
 PLNK OP    =    1.2515E+04
 TH COND    =    9.0489E-07
 NO FR E    =    1.8891E+01
 ENTER REGION,DENSITY,TEMPERATURE
? 0 0 0
 STOP FTN
HYDTX      LTSS TIME     3.691 SECONDS
CPU=      .223      SYS=      .024      I/O=     3.444

 ALL DONE
```

Fig. 2.
Test example.

19

VI. PROGRAM AVAILABILITY

HYDSES may be obtained from the Common File System (CFS) for LASL users by using the LTSS execute line

MASS GET /088077/HYD/HYDSES.

Non-LASL users may obtain HYDSES by contacting

Sesame Library, MS 925
Los Alamos Scientific Laboratory
P.O. Box 1663
Los Alamos, NM 87545

and providing a magnetic tape on which the HYDSES source code will be written.

REFERENCES

1. 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).

2. G. I. Kerley, "Instructions for Use of the Sesame Library," Los Alamos Scientific Laboratory, personal communication, October 1977.

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

4. 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).

5. R. C. Albers, "An Introduction to Two-Temperature Equations of State," Los Alamos Scientific Laboratory report LA-7934-MS (July 1979).

6. C. W. Cranfill and R. More, "IONEOS: A Fast, Analytic, Ion Equation-of-State Routine," Los Alamos Scientific Laboratory report LA-7313-MS (October 1978).

APPENDIX A

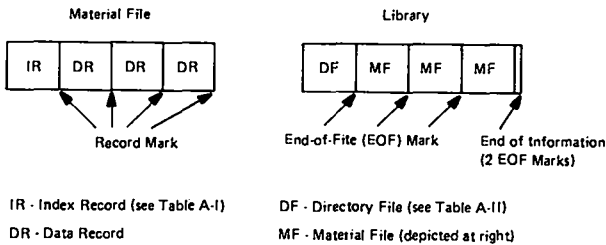## SESAME LIBRARY STRUCTURE AND RANDOM I/O

Fig. A-1.
Structure of the Sesame data library.

A Sesame library consists of a directory file followed by a variable number of material files (see Fig. A-1). Each material file consists of an index record followed by a variable number of data records. The index record format is presented in Table A-I. The data records include the tables described in Sec. II. The directory file consists of two records. These records are described in Table A-II.

The address table appearing in the second record of the directory file supplies addresses that can be used to locate the beginning of each material file. This address table is generated by a new and improved version of the Sesame update package. This package will be sent to users along with the HYDSES package upon request. The addresses in this table are computed assuming that each record is terminated by a one-word record mark and each file is terminated by a one-word file mark. The address computation is performed in subroutines S2BLDL before statement 10. This subroutine must be modified if a different addressing scheme is to be used. Note that the library generated by this package can be read with both sequential and random I/O on LASL/LTSS computers. On other systems, the use of a postprocessing subroutine may be necessary to write the library randomly if the random I/O option is desired.

Subroutine TABFCH (see Sec. II) is used to fetch specified data tables from the Sesame library using random I/O. TABFCH uses the random I/O read subroutine INBUFR. The first word of the directory file is read. This is used to compute the length of the second record of the directory file. This record is then read in starting at address 5. The address of the material index record and its length are obtained from the second record of the directory file for the requested material. The index record is read randomly using the obtained address and length. The address for the requested data table is computed from the starting address of the material file and from length information stored in the index record. The data table is obtained using the address. This subroutine must be modified to implement other addressing schemes.

Subroutine TABSEQ is included in the HYDSES package for those users who wish to avoid system-dependent random I/O. TABSEQ is a sequential I/O version of TABFCH. The TABSEQ calling sequence is identical to that of TABFCH. TABSEQ calls the sequential binary read subroutine INBUF (instead of INBUFR). Use of the address table is avoided by TABSEQ. To use TABSEQ in HYDSES, all the user must do is replace calls to TABFCH by calls to TABSEQ. Note that use of sequential I/O (TABSEQ) is much slower than random I/O (TABFCH). Installations that already support a Sesame library may use TABSEQ to avoid the regeneration of the library (using the new update package) to include the address table.

TABLE A-I

STRUCTURE OF THE INDEX RECORD FOR
A SESAME MATERIAL FILE

MATID, DATE1, DATE2, VERS, NREC, (TBLID(I), I = 1, NREC), (NWDS(I), I = 1, NREC)

MATID	- Material id number (real number).
DATE1	- Date of creation of material.
DATE2	- Date when material file was last updated.
VERS	- Version number of most recent update.
NREC	- Number of data records (real number).
TBLID	- Table of catalog numbers for the data records, in the same order as they are stored in the material file.
NWDS	- For each TBLID(I), NWDS(I) is the number of words in the data record (real number).

TABLE A-II

STRUCTURE OF THE DIRECTORY FILE
FOR THE SESAME LIBRARY

Record 1. Consists of the three words N, DATE, and VERS.

N	- Number of material files in the library (real number).
DATE	- Date of current version.
VERS	- Version number.

Record 2. Consists of the three arrays (MATID(I), I = 1, N), (NWDS(I), I = 1, N), and (IADR(I), I = 1, N).

MATID	- Table of material id numbers, in the same order as the material files in the library (real numbers).
NWDS	- For each material file MATID(I), NWDS(I) is the number of words in the index record (real numbers).
IADR	- IADR(I) is the address of each index record (real numbers).

APPENDIX B

CODE USED TO GENERATE THE TEST EXAMPLE

LASL Identification No. LP-2047

```
 1 $FTN (I=%ME,CNAME=%MEX,L=LIST,GLIB=HYDBIN)
 2       PROGRAM TEST(TTY,TAPE59=TTY)
 3 C . . HYDBIN IS THE BINARY RELOCATABLE CORRESPONDING THE HYDSES
 4 C     SOURCE CODE
 5       LEVEL 2,TBLS
 6       DIMENSION ZB(3),MAT(10)
 7 C . . DIRECTORY COMMON BLOCK
 8       COMMON/S2DIR/LCMX,NRS,LCFW(20,5)
 9 C . . TABLE STORAGE AREA
10       COMMON/SESDAT/TBLS(25000)
11 C . . INTERPOLATION TYPE COMMON
12       COMMON/INTORD/IFN
13 C . . COMMONS TO COMMUNICATE WITH COMPUTATIONAL PHASE SUBROUTINES
14       COMMON/SESIN/IR,IDT,X,Y,IBR,IFL
15       COMMON/SESOUT/Z1(3),Z2(3)
16 C . . INSURE THAT THE LCFW ARRAY IS ZEROED
17       DATA LCFW/100*0/
18       LTTY=59
19 C . . SET FOR RATIONAL FUNCTION INTERPOLATION
20       IFN=0
21 C . . SET DIMENSION OF TBLS
22       LCMX=25000
23 C . . SET TABLE POINTER WORD
24       LCNT=1
25 C . . ASSIGN EOS LIBRARY
26       LES=1
27       CALL QASSIGN(LES,5HSES2L,0,0)
28 C . . ASSIGN OPACITY LIBRARY
29       LOP=2
30       CALL QASSIGN(LOP,6HSES2OP,0,0)
31       WRITE(LTTY,9000)
32  9000 FORMAT(" ENTER THE NUMBER OF REGIONS")
33       READ(LTTY,*) NRS
34       WRITE(LTTY,9010)
35  9010 FORMAT(" ENTER SESAME EOS MATERIAL NUMBERS")
36       READ(LTTY,*) (MAT(J),J=1,NRS)
37 C . . START INITIALIZATION PHASE
38       DO 100 J=1,NRS
39       MID=MAT(J)
40 C . . GET EOS TABLE FOR THIS REGION AND ASSIGN DATA TYPE = 1
```

```
41 c      FOR EOS DATA
42        CALL GETEOS(J,MID,1,TBLS,LCNT,LES,IFL,ZB)
43 c . . GET THE ROSSELAND/PLANCK OPACITY TABLE AND ASSIGN IDT=2
44 c     INCREMENT MID BY 10000 TO GET OPACITY NO.
45        MID=MID+10000
46        CALL GETRPO(J,MID,2,TBLS,LCNT,LOP,IFL)
47 c . . GET THE THERMAL CONDUCTIVITY TABLE AND ASSIGN IDT=3
48        CALL GETTCE(J,MID,3,TBLS,LCNT,LOP,IFL)
49 c . . GET THE NUMBER OF FREE ELECTRONS TABLE AND ASSIGN IDT=4
50        CALL GETNFE(J,MID,4,TBLS,LCNT,LOP,IFL)
51   100  CONTINUE
52 c . . START OF COMPUTATIONAL PHASE
53   200  WRITE(LTTY,9020)
54  9020  FORMAT(" ENTER REGION,DENSITY,TEMPERATURE")
55        READ(LTTY,*) IR,X,Y
56        IF(IR.EQ.0) STOP
57 c . . COMPUTE EOS
58        IDT=1
59        IBR=0
60        CALL T4DAT
61        P=z1(1)
62        E=z2(1)
63 c . . COMPUTE THE ROSSELAND AND PLANCK OPACITY
64        IDT=2
65        X=ALOG10(X)
66        Y=ALOG10(Y)
67        CALL T4DAT
68        ROP=10.0**z1(1)
69        POP=10.0**z2(1)
70 c . . COMPUTE THE THERMAL CONDUCTIVITY
71        IDT=3
72        IBR=1
73        CALL T4DAT
74        VE=10.0**z1(1)
75 c . . COMPUTE THE NUMBER OF FREE ELECTRONS
76        IDT=4
77        CALL T4DAT
78        FNF=10.0**z1(1)
79 c . . WRITE OUTPUT TO TTY
80        WRITE(LTTY,9050) P,E,ROP,POP,VE,FNF
81        GO TO 200
82  9050  FORMAT(1x,10HPRESSURE  ,2H= ,1PE12.4,/,
83        $1x,10HENERGY    ,2H= ,E12.4,/,
84        $1x,10HROSS OP   ,2H= ,E12.4,/,
85        $1x,10HPLNK OP   ,2H= ,E12.4,/,
86        $1x,10HTH COND   ,2H= ,E12.4,/,
87        $1x,10HNO FR E   ,2H= ,E12.4)
88        END
```

001-025	4.00	126-150	7.25	251-275	10.75	376-400	13.00	501-525	15.25
026-050	4.50	151-175	8.00	276-300	11.00	401-425	13.25	526-550	15.50
051-075	5.25	176-200	9.00	301-325	11.75	426-450	14.00	551-575	16.25
076-100	6.00	201-225	9.25	326-350	12.00	451-475	14.50	576-600	16.50
101-125	6.50	226-250	9.50	351-375	12.50	476-500	15.00	601-up	

Microfiche $3.00

001-025	4.00	126-150	7.25	251-275	10.75	376-400	13.00	501-525	15.25
026-050	4.50	151-175	8.00	276-300	11.00	401-425	13.25	526-550	15.50
051-075	5.25	176-200	9.00	301-325	11.75	426-450	14.00	551-575	16.25
076-100	6.00	201-225	9.25	326-350	12.00	451-475	14.50	576-600	16.50
101-125	6.50	226-250	9.50	351-375	12.50	476-500	15 00	601-up	

Note: Add $2.50 for each additional 100-page increment from 601 pages up.