ORNL/TM-6503

Contract No. W-7405 eng 26

Engineering Physics Division

CARP, A COMPUTER CODE AND ALBEDO DATA LIBRARY FOR USE BY BREESE, THE MORSE ALBEDO PACKAGE

M. B. Emmett*
W. A. Rhoades

October, 1978

*Computer Sciences Division

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

CONTENTS

	Page
ABSTRACT	1
Part I - Code Description	1
MAIN Program	4
ALOCAT (LOC, SUB)	4
Subroutine GIPR (LD, LIMD)	4
Subroutine SOURCE (A, F, ENER, DENG, VEL, INDX, THETI, THETO, NPSIL, DPSI, DETA, ABDO, W, EMU, ETA, NPSIX, KARY, MMIGM, FLX, IGFEW, FFLX, DOSE)	5
Subroutine BINR (DWT, ETA, DDETA, DETA, DDPSI, DPSI, EMU, NPSIL, NLETA, NDETA, NDANG, NANG, NEDIT, NOU)	7
Function TRUNCF (X, A, B)	8
Subroutine BLKIO (II, NT, A, M, IREC, KARY, NRECMX, NWBLKM)	8
Subroutine CLEARX (X, L1, L2)	9
Subroutine ERRO (ICC, ISEV, N, MESG)	9
The Free-Field Input Routines	10
Subroutine FIDOS (LL2, J3, N5, N6, D, LDTK, LOCO)	10
Subroutine IOERR (IRC, IERN)	11
Subroutine SEQIO (IO, NT, A, M, IREC)	11
Subroutine SKPEDI (K, N, M, T, NT)	11
Subroutine TIMEX	12
Subroutine WOT (X, NCOLD, LTBLD, LG, TOP1, TOP2, TOP3)	12
Subroutine WOT8 (A, L1, B, L2, C, L3, D, L4, E, L5, F, L6, G, L7, H, L8, NOU)	12
Subroutine WOT10 (A, L1, B, L2, C, L3, D, L4, E, L5, F, L6, G, L7, H, L8, P, L9, Q, L10, NOU)	12
Input Data	13
Part II - Albedo Data Library	16
REFERENCES	20
APPENDIX A. ARRANGEMENT OF DATA ON BREESE II ALBEDO TAPE	A-1
APPENDIX B. FIDO INPUT	B-1

BLANK PAGE

CARP, A COMPUTER CODE AND ALBEDO DATA LIBRARY FOR USE BY BREESE, THE MORSE ALBEDO PACKAGE

M. B. Emmett
W. A. Rhoades

ABSTRACT

The CARP computer code was written to allow processing of DOT angular flux tapes to produce albedo data for use in the MORSE computer code. An albedo data library has been produced containing several materials.

Part I - Code Description

The main purpose of the CARP* code is to process either a DOT[1] angular flux tape or a CARP storage tape and produce an albedo output tape which is in the format required for the BREESE[2] tape-writing program. BREESE prepares an albedo tape for input to MORSE.[3] During the processing of this data, the group structure may be collapsed, dose conversion factors may be applied to the fluxes, and in the case of group reduction, group fluxes may be used for weighting.

CARP uses the CCCC[4] package for I/O. This package is applicable only to IBM computers and must be modified by non-IBM users. There are internal comments indicating the functions of routines TIMER, REED, DOPC and DRED to guide users in converting to other machines. These routines call other routines which may or may not be needed by the user's adaptation.

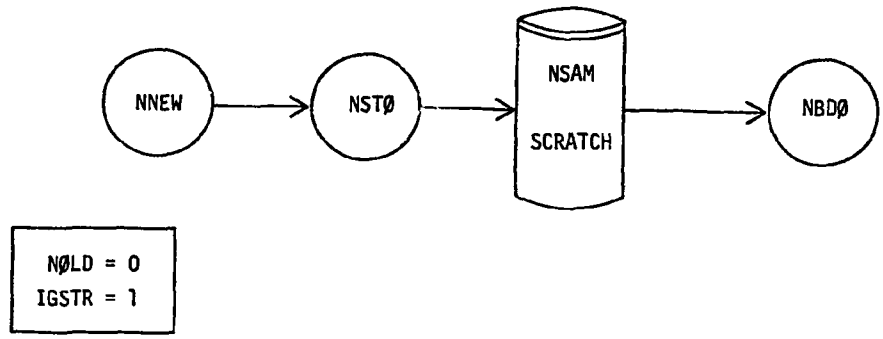
---

*CARP is an acronym for Compilation of Albedo Reflection Parameters.

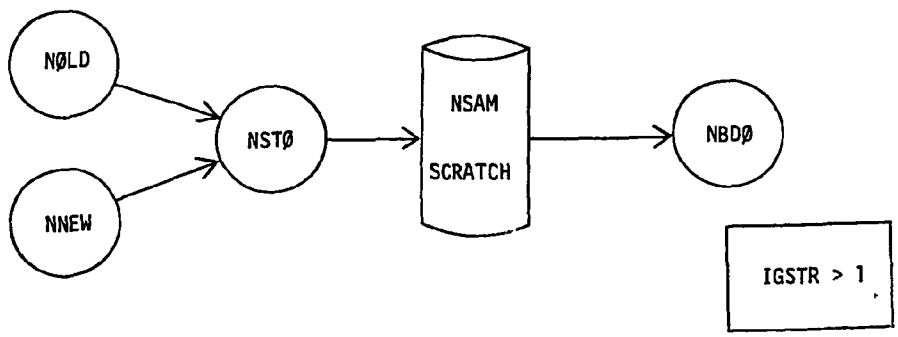
The assembler-language subroutine ALOCAT adds convenience in that the amount of memory available for data storage is automatically adjusted to fill the space allocated in the JCL REGION parameter. A FORTRAN substitute is also available, but it requires a fixed dimension.

There are five modes of operation of the CARP code as follows:

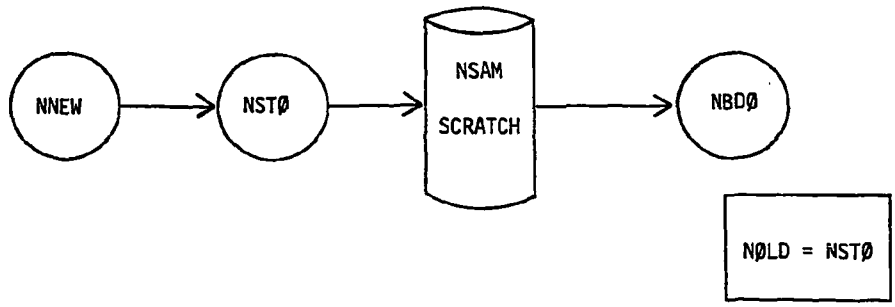
1. Process a single DOT tape (NNEW) to create a CARP storage tape (NSTO) and write the BREESE output tape (NBDO).



2. Combine data from two DOT tapes (NOLD, NNEW) to a new CARP storage tape (NSTO) and write BREESE output tape (NBDO).

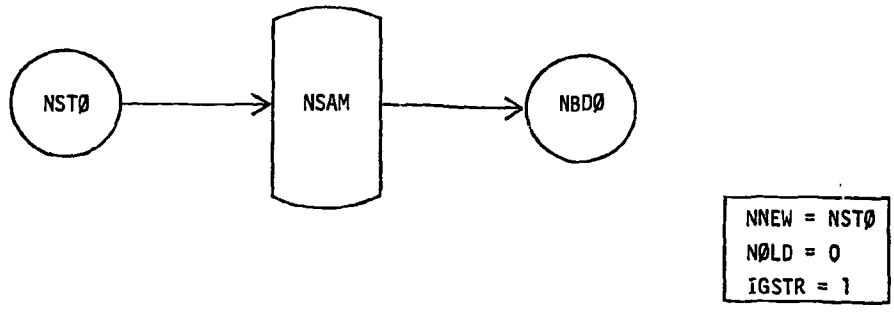


3. Add DOT data from NNEW to an old CARP storage tape (NSTO)* and write a BREESE output tape (NBDO).



*NSTO is an old tape that is added to in this run which makes this option different from 1 above.

4. Process an old CARP storage tape (NSTO) and write a BREESE output tape (NBDO).



5. Produce no BREESE output tape (NBDO=0) but print an edit of the data on the CARP storage tape. NSAM is not used. This option is available regardless of which of the previous options (1-3) was chosen.

MAIN Program

The main program for CARP does some initialization, reads in some of the input cards and the first record from the DOT tape and prints this input data.

Commons: IOREC, BULKBU

Routines called: ERRSET, TIMSET, CLEARX, DOPC, ALOCAT, FIDOS, SEQIO, TIMEX, SQRT, FLOAT, MOD

ALOCAT (LOC, SUB)

Subroutine ALOCAT in FORTRAN language requires a dimension at compile time, but the assembly language version provides run-time dimensioning on IBM equipment. The user must specify LOC which is the size of his data array, D, plus the buffer space needed for scratch units. (Refer to the FORTRAN listing of ALOCAT.) ALOCAT also calls SUB, the working subroutine, which in CARP is named GIPR.

Subroutine GIPR (LD, LIMD)

Subroutine GIPR is the working subroutine of CARP. It reads the input cards not read in the MAIN program, sets up and initializes storage locations in the large array, LD, for the smaller arrays such as

energy and velocity (See Fig. 1 for explanation) and calls the SOURCE routine which calculates the albedo data. LIMD is the size of array LD.

Commons: BULKBU, IOREC

Called from: ALOCAT

Routines Called: CLEARX, ERROD, FIDOS, ERRO, ROKK, SOURCE

Subroutine SOURCE (A, F, ENER, DENG, VEL, INDX, THETI, THETO, NPSIL, DPSI, DETA, ABDO, W, EMU, ETA, NPSIX, KARY, MMIGM, FLX, IGFEW, FFLX, DOSE)

Subroutine SOURCE is the routine which generates the albedos from the DOT fluxes. It, therefore, does the group reduction, applies dose factors and group fluxes where applicable, writes the albedo tape, and prints the total albedos on the standard output tape.

The parameters in SOURCE are defined as follows:

A - array of size sufficient to hold one block of NGO*MMO*NORD or two blocks of NGO*MMO.

F - array of fluxes from DOT tape - for diagnostic purposes to avoid reading the tape, F is specially defined.

ENER - the energy bin limits in eV in descending order (IGM values)

DENG - the delta ENER - the width of the energy bins in eV

VEL - the velocities corresponding to the energies

INDX - indices which are used for diagnostic purposes

THETI - the incoming polar angles in degrees in ascending order (NTHETI values)

THETO - cosine of outgoing polar angles in descending order

Starting Location	Mnemonic Variable Name	Length
1	LDARY	1
2	LIMD	1
3	KARY	1
4	Scratch space used by various I/Ø subroutines	143+ NØRD*IGMNEW/2†
LIGFEW	IGFEW (4$ array)	IGM
LFLX	FLX (5* array)	IGM
LDØSE	DØSE (6* array)	IGM
LENER	ENER (7* array	IGM+2
LDENG	DENG	IGM
LFFLX	FFLX	NGØ
LVEL	VEL	IGM
LINDX	INDX	MM
LTHETI	THETI	MM
LTHETØ	THETØ	NØRD
LNPSIL	NPSIL	NØRD
LDPSI	DPSI	MM
LDETA	DETA	NØRD+2
LW	W	MM
LEMU	EMU	MM
LETA	ETA	MM
LNPSIX	NPSIX	NØRD+1
LABDØ	ABDØ	NGØ*NØRD
LF	F	IGM*MM*NØRD
LA	A	(NGØ*MM*NØRD)/2
LIMD	LDEND	1

Fig. 1. Layout of Large Storage Array D

†NØRD is the number of polar angle levels.

NPSIL - the number of outgoing azimuthal angle bins as function of both polar angles

DPSI - cosines of the output azimuthal angle bins in decreasing order

DETA - eta boundaries

ABDO - values of total albedo for each few group

W - the fraction of the solid angle associated with the discrete direction ($\Sigma_i W_i = 1$)

EMU - cosine of angle between discrete direction and a unit vector in outward radial direction - from DOT quadrature

ETA - cosine of angle between discrete direction and a unit vector in the z direction - from DOT quadrature

NPSIX - level origin (NTHETI values)

KARY - an array of I/O control information

MMIGM - MM*NGO or number of angles (MM) times number of groups in collapsed set (NGO)

FLX - the group fluxes for group reduction (5* input array)

IGFEW - the integer array specifying the coarse group number for each fine group (4* array from input)

FFLX - flux for each few group

DOSE - the dose factors (see 6* input array)

Called from: GIPR

Commons required: BULKBU, IOREC

Routines called: SEQIO, CLEARX, BINR, WOT10, BLKIO, WOT, TIMEX, ATAN, COS, SQRT, TRUNCF, ABS

Subroutine BINR (DWT, ETA, DDETA, DETA, DDPSI, DPSI, EMU, NPSIL, NLETA, NDETA, NDANG, NANG, NEDIT, NOU)

Subroutine BINR finds the ϕ,μ bins with respect to the x-axis if the quadrature set is symmetric.

Called from: SOURCE

Commons Required: None

Routines Called: WOT8

Variables Required:
- DWT, EMU, ETA - the DOT quadrature consisting of weights, μ's and η's
- NANG - MM the number of directions
- NEDIT - signal =0 no edit
  ≠0 print edit of data
- NOU - logical unit for standard output tape

Variables Modified:
- DDETA - the eta increment
- DETA - the eta boundary
- DDPSI - psi increment
- DPSI - the psi boundary in radians
- NDETA - the number of downward η bins
- NLETA - the number of upward η bins
- NDANG - index of last downward bin
- NPSIL - the number of outgoing azimuthal angle bins per polar angle
- NDPSI - the number of psi boundaries

Function TRUNCF (X, A, B)

TRUNCF examines x to see if it is within e (10^{-5}) of A (if so, set TRUNCF=A) or B (if so, set TRUNCF=B).

Called by: SOURCE

Subroutine BLKIO (II, NT, A, M, IREC, KARY, NRECMX, NWBLKM)

Subroutine BLKIO is for block I/O control using REED and RITE.

Called by: SOURCE

Commons Required: IOREC

Routines Called: DOPC, DRED, DRIT, ERRO

Variables Required:
- IT - -2/-1/0/1/2 = OPEN/CLOSE/REWIND/READ/WRITE
  +0/10/20 = START & FINISH/START/FINISH TRANSMISSION
  +0/100/200 = LARGE BLOCK MODE/SMALL WITH OVERFLOW/SMALL

NT - data set reference number
A - first word transmitted
M - number of full words transmitted
IREC - index of logical record to be processed (=0 means next record in data set)
KARY - array of I/O control information
NRECMX - maximum number of records in data set (may be < 0 if records processed sequentially or if records all have same length)
NWBLKM - maximum record length for data set (=0 indicates fixed record length)

Variables Modified:
NEDUM - 0 = data set never opened
1 = data set has been rewound
+N = finished reading record N-1
-N = finished writing record N-1
IOMODD - position in KARY of control data (negative indicates an unchecked operation)
JERR - error signal

Subroutine CLEARX (X, L1, L2)

Subroutine CLEARX has three entry points: CLEARX, MULTR and SETR. CLEARX zeroes array X from the L1 to L2 locations. The MULTR (X, L1, XX) entry multiplies the first L1 members of the X array by XX. The SETR (X, L1, XX) sets the first L1 members of X equal to XX.

Subroutine ERRO (ICC, ISEV, N, MESG)

Subroutine ERRO has two entry points ERRO and ERROD (D, LIMD). This routine performs a diagnostic edit and may get a dump. A data statement defines logical 6 as standard output tape. ERROD sets NDLOC to minimum of LIMD or 1000.

Commons Required: BULKBU, IOREC

Called from: GIPR, BLKIO

Routines Called: ERRTRA, SKPEDI, SKPEDE, SKPEDZ, MAXO, MINO

Variables Required:
ICC
ISEV - signal on what to edit - > 6=edit commons
N - number of characters in message
MESG - message to be printed

The Free-Field Input Routines*

The free-form input routines were written by W. W. Engle and modified for CARP by W. A. Rhoades.

Subroutine FIDOS (LL2, J3, N5, N6, D, LDTK, LOCO)

FIDOS initializes L60 in COMMON COMDTF and calls FIDAS.

FIDAS (D, LDTK, LDAI, LL2, J3, N5, N6) calls FFREAD to read the input cards and then interprets them and stores them accordingly in D array.

Commons Required: COMDTF

Called by: FIDOS

Variables Required:
N5 and N6 - standard I/O units
LDTK - array containing pointers to correct location in D for storage of data
LL2 - used in calculating index in LDTK

Variables Modified:
J3 - error counter
D - data storage

FFREAD (IN, K, V, NF, NS, NG, IPRIRG) reads one input card, stores the data in IN, K, and V arrays, and counts the fields, NF.

---

*See Appendix B.

Subroutine IOERR (IRC, IERN)

Subroutine IOERR increments an error counter JERR in common IOREC. The parameters are not used in this version.

Subroutine SEQIO (IO, NT, A, M, IREC)

Subroutine SEQIO is for sequential I/O control using subroutines REED and RITE of the CCCC I/O package. Data statement sets standard output unit to logical 6.

Commons Required: IOREC

Called by: SOURCE

Routines Called: CLEARX, REED, RITE, ERRO

Variables Required:
- IO - a signal defined as follows: 0/1/2 = REWIND/INPUT/OUTPUT
- NT - data set reference number
- IREC - number of records to transfer; if = 0, transfer next record
- M - number of full words transmitted
- A - first word transmitted

Variables Modified:
- M, A - if IO = 1
- NEDUM - signal as follows: 0 = data set never opened; 1 = data set has been rewound; +N = finished reading record N-1; -N = finished writing record N-1
- JERR - error signal set to values < 0 if error occurs. If (JERR<0), subroutine ERRO is called to get diagnostic edit.

Subroutine SKPEDI (K, N, M, T, NT)

Subroutine SKPEDI edits N words of K array by M's and prints NT characters of title T. If M < 0, it skips to insure number of words printed is $\leq$ -M. Data statement sets standard output tape equal 6.

SKPEDI has two other entry points SKPEDE and SKPEDZ. The only difference in them is the type of output format used, I, E, or Z.

Subroutine TIMEX

The first entry to TIMEX or any entry to its other entry point TIMSET prints date and time. Subsequent entries to TIMEX print a cumulative and incremental time elapsed. A data statement sets standard output tape equal to 6.

Subroutine WOT (X, NCOLD, LTBLD, LG, TOP1, TOP2, TOP3)

Subroutine WOT has four entry points WOT, WOD, WIT, and WID. These entry points are used for output of one-, two-, or three-dimensional data arrays, X, on standard output unit 6.

Subroutine WOT8 (A, L1, B, L2, C, L3, D, L4, E, L5, F, L6, G, L7, H, L8, NOU)

Subroutine WOT8 prints tables of output (on unit NOU) of up to nine columns wide where columns 2-9 are arrays A through H of the specified length (L1 through L8); i.e., L1 values of A, L2 values of B, etc.

Subroutine WOT10 (A, L1, B, L2, C, L3, D, L4, E, L5, F, L6, G, L7, H, L8, P, L9, Q, L10, NOU)

Subroutine WOT10 is same as WOT8 except it allows up to 10 arrays and 11 columns.

Input Data

The input data for CARP is described below. Except for the title card, all other card input is in free-form and is read with FIDOS and FFREAD.

Card 1 - TITLE (I), I=1,18 18A4

This is the title which will appear on both the printer output and on the tape of albedo data.

Card 2 - 1$$ followed by an array of integers as follows:

IGSTR - First in-group on new tape

IGMNEW - Last in-group on new tape

IGFOLD - First in-group on old tape

IGNEUT - Last primary group

IGISTR - First in-group for albedo output (default is 1)

IGISTP - Last in-group for albedo output (default is IGMNEW)

IGOSTR - First out-group for albedo output (default is 1)

IGOSTP - Last out-group for albedo output (default is IGM where IGM is number of energy groups on the DOT tape)

NOLD - Unit number for old fluxes (default is 10)

NNEW - Unit number for new fluxes (default is 9)

NSTO - Unit number for stored fluxes (default is 8)

NBDO - Unit number for albedo output (default is 2)

NSAM - Unit number for sorting output (default is 91) (usually a scratch disk)

NEDIT - signal as follows:

= 0 no editing (default)

= 1 minimum editing

= 2 maximum editing

Card 3 - Contains a T in column 3 to indicate end of 1$$ array, or the T can appear on Card 2 after the last value.

The remainder of the input cards consists of the 4$, 5*, 6*, and 7* arrays. Only the 7* array is required; the others depend on the user's choice of options. The arrays may be entered in any order, but the last one entered must be followed by a terminator T as described for Card 3 above.

1. 4$ array is an integer array specifying the coarse-group number for each fine group. IGM values
   The default of coarse group equals fine-group is applied if the user omits this array.

2. 5* array is a floating-point array specifying the group flux for the group reduction. (This is not a spectrum or an average flux.) The array is used for weighting. IGM values
   The default is $\Delta E/v^2$

3. 6* array is a floating-point array specifying the dose factors to be used. IGM values

   The default is 1.0. If the user fills this array with -1.0, this causes a dose factor equal to the mean energy.

4. 7* array is a floating-point array specifying the energy group boundaries in MeV. IGM + 2 values

The rest of the input comes from the DOT tape which is unformatted (binary). The data used is as follows:

RECORD 1 - (IDOT(I),I=1,57) - the first 18 words are a title, IDOT(22)=MM, IDOT(26)=NPHI[†] and IDOT(45)=IGM.

RECORD 5 - (W(I),I=1,3*MM, where MM is the number of angles or directions in DOT).

The elements of the W array correspond to:

W(1 to MM)	Wgt
(MM + 1 to 2*MM)	μ
(2*MM + 1 to 3*MM)	η

RECORD - (F(I),I=1, NPHI*MM) $\sum_{I=IGFOLD}^{IGMNEW}$ (IGM - I + 1) records

where for I≥IGSTR, tape NNEW is read; and for I≤IGSTR, tape NOLD is read only if NSTO is not equal to NOLD. IF IGMNEW<IGM, the tapes may contain unused records at the end.

F is DOT angular fluxes.

[†]NPHI is number of outgoing angles.

Part II - Albedo Data Library

The DOT code has already been used to generate albedo tapes for four materials as follows:

1. 12-inch water
2. 12-inch ordinary concrete
3. 9-inch carbon steel (SA508)
4. 1/2-inch steel over 12-inch concrete

The compositions of these materials are listed in Table 1.

Table 1. Atomic Number Density (Atoms/Å^3) of Albedo Materials

Element	Density		
	Concrete	Water	Carbon Steel
O	4.386-2*	3.344-2	
Ca	2.915-3		
Al	2.388-3		
C			1.040-3
Si	1.580-2		
H	7.768-3	6.688-2	
K	6.900-4		
Mg	1.487-4		
Fe	3.128-4		8.380-2
Na	1.048-3		
Mn			7.760-4

*Read as 4.386×10^{-2}.

The 51 neutron-25 gamma group structure indicated in Table 2 was used. The amount of data required for this group structure is too large for

most computers, making it necessary for users to reduce the number of groups by running CARP.

The differential angular albedos are prepared as a function of five incident polar directions and 30 reflected directions, which consist of five polar angle levels and their corresponding azimuthal angles. These angles are derived from a half-symmetrical S-10 quadrature set. The cosines of the polar angles, θ_j, are computed from the weights corresponding to the μ's using the formula $\cos\theta_j = \mu_j$, where $\Delta\mu_j = 2 \sum_i W_{ij}$. This results in $\cos\theta_j$ having values 1.0, .93333, .78388, .56479, .29552 and 0.0. For the azimuthal angle, ϕ_i,

$$\Delta\phi_i = \frac{2\pi W_{ij}}{\Delta\mu_j}$$

since $\Delta\mu_j \cdot \Delta\phi_i \cong 2\pi W_{ij}$. Table 3 contains the values of $\cos\phi_i$ corresponding to each $\cos\theta_j$.

The albedo data is available from RSIC, the Radiation Shielding Information Center at Oak Ridge National Laboratory. Interested parties may send a tape to RSIC requesting the CARP Albedo Data Library.

Table 2. Group Structure for Albedo Data

Neutron		Gamma	
Group	Energy (eV)	Group	Energy (eV)
1	1.4918E 07	1	1.3000E 07
2	1.2214E 07	2	1.0197E 07
3	1.0000E 07	3	7.9983E 06
4	8.1873E 06	4	6.2737E 06
5	6.7032E 06	5	4.9210E 06
6	5.4881E 06	6	3.8599E 06
7	4.4933E 06	7	3.0277E 06
8	3.6788E 06	8	2.3748E 06
9	3.0119E 06	9	1.8628E 06
10	2.4660E 06	10	1.4611E 06
11	2.0190E 06	11	1.1461E 06
12	1.6530E 06	12	8.9896E 05
13	1.3534E 06	13	7.0513E 05
14	1.1080E 06	14	5.5309E 05
15	9.0718E 05	15	4.3383E 05
16	7.4274E 05	16	3.4029E 05
17	6.0810E 05	17	2.6692E 05
18	4.9787E 05	18	2.0937E 05
19	4.0762E 05	19	1.6422E 05
20	3.3373E 05	20	1.2881E 05
21	2.7324E 05	21	1.0104E 05
22	2.2371E 05	22	7.9252E 04
23	1.8316E 05	23	6.2164E 04
24	1.4996E 05	24	4.8760E 04
25	1.2277E 05	25	3.8247E 04
26	8.6517E 04		
27	5.2475E 04		
28	4.0868E 04		
29	3.1828E 04		
30	2.4788E 04		
31	1.9305E 04		
32	1.5034E 04		
33	7.1018E 03		
34	4.3074E 03		
35	3.3546E 03		
36	2.6126E 03		
37	2.0347E 03		
38	1.5846E 03		
39	1.2341E 03		
40	9.6112E 02		
41	4.5400E 02		
42	2.1445E 02		
43	1.0130E 02		
44	4.7851E 01		
45	2.2603E 01		
46	1.0677E 01		
47	5.0435E 00		
48	2.3824E 00		
49	1.1254E 00		
50	4.1400E -01		
51	1.0000E -01		

Table 3. Angular Structure of Albedo Data

	cosθ (polar)				
	1.0	.93333	.78388	.56479	.29552	0.
cosϕ (azimuthal)	1.00	1.00	1.00	1.00	1.00
	0.0	.51690	.71466	.84172	.93787
	-1.00	0.0	.52349	.71802	.80857
		-.51690	0.0	.15933	.50836
		-1.00	-.52349	0.0	.37786
			-.71466	-.15933	0.0
			-1.00	-.71802	-.37786
				-.84172	-.50836
				-1.00	-.80857
					-.93787
					-1.00

REFERENCES

1. W. A. Rhoades, D. B. Simpson, R. L. Childs and W. W. Engle, Jr., *The DOT-IV Two-Dimensional, Discrete Ordinates Transport Code with Space-Dependent Mesh and Quadrature*, ORNL-5283 (1978).

2. V. R. Cain, *BREESE: Auxiliary Routines for Implementing the Albedo Option in the MORSE Monte Carlo Code*, G.P.D.-Nuclear-T.D.-4.

3. M. B. Emmett, *The MORSE Monte Carlo Radiation Transport Code System*, ORNL-4972 (1975).

4. B. M. Carmichael, *Standard Interface Files and Procedures for Reactor Physics Codes, Version III*, LA-5486-MS (1974).

APPENDIX A

BLANK PAGE

APPENDIX A

ARRANGEMENT OF DATA ON BREESE II ALBEDO TAPE*

RECORD 1 (TITLE(I), I=1, 36)

TITLE is the title of this interface (see input card #1) plus the title of the DOT problem (from the DOT tape)

RECORD 2 - NTHETI, INRGYS, NMGPA

NTHETI = number of incoming angles (θ's)

INRGYS = number of incoming energy bins

NMGPA = number of last primary group

RECORD 3 - (ENER(I), I=1, IN) IF INRGYS>NMGPA,IN=INRGYS+2; otherwise, IN=INRGYS+1

ENER = energy bin limits in eV in descending order

RECORD 4 - (VEL(I), I=1, INRGYS)

VEL = velocity of each energy bin in same order as ENER

RECORD 5 - (THETIN(IN), IN=1, NTHETI)

THETIN is incoming angles (θ's) in degrees, in ascending order

RECORD 6 - (NTHETO(I), I=1, NTHETI)

NTHETO(I) = number of outgoing polar angle bins for incoming angle I

RECORD 7 through 7+NTHETI-1 - (THETA(K,IO),K=1,NTHETO(I)+1)

THETA - cosines of outgoing polar angle boundaries in descending order for each incoming angle IO

---

*Variable names used here are same as in BREESE II where the tape is read.

RECORD(7+NTHETI) through (7+NTHETI)+(NTHETI-1) - (NPHI (K,IO),K=1,
NTHETO(I))

NPHI = the number of outgoing azimuthal angle bins (ϕ's) per outgoing polar angle (same order as THETA above)

RECORD 7+2*NTHETI - ((PHI(L,K,IO),L=1,NPHI(K,IO)+1),K=1,NTHETO(IO)),
IO=1

RECORD 7+2*NTHETI+1 - ((PHI(L,K,IO),L=1,NPHI(K,IO)+1),K=1,
NTHETO(IO)),IO=2

to

RECORD 7+3*NTHETI-1 - ((PHI(L,K,IO),L=1,NPHI(K,IO)+1),K=1,
NTHETO(IO)),IO=NTHETI

PHI = cosines of outgoing azimuthal angle bins in descending order

RECORD (7+NTHETI*3) - (A(IGO,L,K), IGO=1,INRGYS),L=1,NPHI(K,IO)),
K=1,NTHETO(IO))

INRGYS*NTHETI records

A = current out per steradian per eV per current incident

APPENDIX B

BLANK PAGE

APPENDIX B

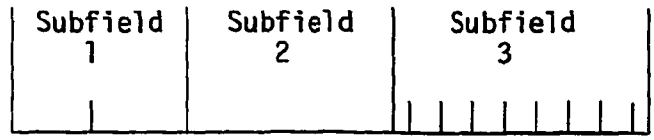
FIDO INPUT

The FIDO input method is especially devised to allow entering or modifying large data arrays with minimum effort. Special advantage is taken of patterns of repetition or symmetry wherever possible. The FIDO system was patterned after the input method used with the FLOCO coding system at Los Alamos and was first applied by Atomics International to the DTF-II code. Since that time, numerous features requested by users have been added, a free-field option has been developed, and the application of FIDO has spread to innumerable codes.

The data are entered in units called "arrays." An array comprises a group of contiguous storage locations which are to be filled with data at one time. These arrays usually correspond on a one-to-one basis with FORTRAN arrays used in the program. A group of one or more arrays read with a single call to the FIDO package forms a "block," and a special delimiter is required to signify the end of each block. Arrays within a block may be read in any order with respect to each other, but an array belonging to one block must not be shifted to another. The same array can be entered repeatedly within the same block. For example, an array could be filled with "0" using a special option, and then a few scattered locations could be changed by reading in a new set of data for that array. If no entries to the arrays in a block are required but the condition requiring the block is met, the delimiter alone satisfies the input requirement.

Three major types of input are available: fixed-field input, free-field input, and user-field input.

Fixed Field Input - Each card is divided into six 12-column data fields, each of which is divided into three subfields. The following sketch illustrates a typical data field. The three subfields always comprise two, one, and nine columns, respectively.



To begin the first array of a block, an array originator field is placed in any field on a card:

Subfield 1: An integer array identifier < 100 specifying the data array to read.

Subfield 2: An array-type indicator -

"$" if the array is integer data

"*" of the array is real data

Subfield 3: Blank

Data are then placed in successive fields until the required number of entries has been accounted for.

In entering data, it is convenient to think of an "index" or "pointer" which is under control of the user, and which specifies the position in the array into which the next data entry is to go. The pointer is always positioned at array location #1 by entering the array originator field. The pointer subsequently moves according to the data operator chosen. Blank fields are a special case in that they do not cause any data modification and do not move the pointer.

A data field has the following form:

Subfield 1: The data numerator, an integer < 100. We refer to this entry as N_1 in the following discussion.

Subfield 2: One of the special data operators listed below.

Subfield 3: A nine-character data entry, to be read in F9.0 format. It will be converted to an integer if the array is a "$" array or if a special array operator such as "Q" is being used. Note that an exponent is permissible but not required. If no decimal is supplied, it is assumed to be immediately to the left of the exponent, if any, otherwise to the right of the last column. This entry is referred to as N_3 in the following discussion.

A list of data operators and their effect on the array being input follows:

"Blank" indicates a single entry of data. The data entry in the third subfield is entered in the location indicated by the pointer, and the pointer is advanced by 1. However, an entirely blank field is ignored.

"+" or "-" indicates exponentiation. The data entry in the third field is entered and multiplied by $10^{\pm N_1}$, where N_1 is the data numerator in the first subfield, given the sign indicated by the data operator itself. The pointer is advanced by 1. In cases where an exponent is needed, this option allows the entering of more significant figures than the blank option.

"&" has the same effect as "+" on IBM systems.

"R" indicates that the data entry is to be repeated N_1 times. The pointer is advanced by N_1.

"I" indicates linear interpolation. The data numerator, N_1, indicates the number of interpolated points to be supplied. The data entry in the third subfield is entered, followed by N_1 interpolated entries equally spaced between that value and the data entry found in the third subfield of the next non-blank field. The pointer is advanced by $N_1 + 1$. The field following an "I" field is then processed normally, according to its own data operator. The "I" entry is especially valuable for specifying a spatial mesh. In "$" arrays, interpolated values will be rounded to the nearest integer.

"L" indicates logarithmic interpolation. The effect is the same as that of "I" except that the resulting data are evenly separated in log-space. This is especially convenient for specifying an energy mesh.

"Q" is used to repeat sequences of numbers. The length of the sequence is given by the third subfield, N_3. The sequence of N_3 entries is to be repeated N_1 times. The pointer is advanced by N_1*N_3. If either N_1 or N_3 is 0, then a sequence of $N_1 + N_3$ is repeated one time only, and the pointer is advanced by $N_1 + N_3$. This feature is especially valuable for geometry specification.

"G" has the same effect as Q, except that the sign of the sequence is changed each time it is entered.

The "N" option has the same effect as "Q", except that the order of the sequence is reversed each time it is entered. This is valuable for the type of symmetry possessed by quadrature coefficients.

"M" has the same effect as "N" except that the sign of each entry in the sequence is reversed each time the sequence is entered. For example, the entries:

1 2 3 2M2

would be equivalent to:

1 2 3 -3 -2 2 3

This option is also useful in entering quadrature coefficients.

"Z" causes $N_1 + N_3$ locations to be set to 0. The pointer is advanced by $N_1 + N_3$.

"C" causes the position of the last array item entered to be printed. This is the position of the pointer, less 1. The pointer is not moved.

"O" causes the print trigger to be turned on. The trigger is originally off. When the trigger is on, each card image is listed as it is read.

"P" causes the print trigger to be turned off.

"S" indicates that the pointer is to skip N_1 positions leaving those array positions unchanged. If the third subfield is non-blank, that data entry is entered following the skip, and the pointer is advanced by $N_1 + 1$.

"A" moves the pointer to the position N_3, specified in the third subfield.

"F" fills the remainder of the array with the datum entered in the third subfield.

"E" skips over the remainder of the array. The array length criterion is always satisfied by an "E", no matter how many entries have been specified. No more entries to an array may be given following an "E", except that data entry may be restarted with an "A".

The reading of data to an array is terminated when a new array origin field is supplied, or when the block is terminated. If an incorrect number of positions has been filled, an error edit is given, and a flag is set which will later abort execution of the problem. FIDO then continues with the next array if an array origin was read. Otherwise, it returns control to the calling program.

A block termination consists of a field having "T" in the second subfield. All entries following "T" on a card are ignored, and control is returned from FIDO to the calling program.

Comment cards can be entered within a block by placing a slash (/) in column 1. Then columns 2-80 will be listed, with column 2 being used for printer carriage control. Such cards have no effect on the data array or pointer.

Free-field Input - With free-field input, data are written without fixed restrictions as to field and subfield size and positioning on the

card. The options used with fixed-field input are available, although some are slightly restricted in form. In general, fewer data cards are required for a problem, the interpreting print is easier to read, a card listing is more intelligible, the cards are easier to keypunch, and certain common keypunch errors are tolerated without affecting the problem. Data arrays using fixed- and free-field input can be intermingled at will within a given block.

The concept of three subfields per field is still applicable to free-field input; but if no entry for a field is required, no space for it need be left. Only columns 1-72 may be used, as with fixed-field input.

The array originator field can begin in any position. The array identifiers and type indicators are used as in fixed-field input. The type indicator is entered twice, to designate free-field input (i.e., "$$" or "**"). The blank third subfield required in fixed-field input is not required. For example: 31** indicates that array 31, a real-data array, will follow in free-field format.

Data fields may follow the array origin field immediately. The data field entries are identical to the fixed-field entries with the following restrictions:

(1) Any number of blanks may separate fields, but at least one blank must follow a third subfield entry if one is used.

(2) If both first and second subfield entries are used, no blanks may separate them, i.e., 24S, but not 24 S.

(3) Numbers written with exponents must not have imbedded blanks, i.e., 1.0E+4, 1.0E4, 1.0+4, or even 1+4, but not 1.0 E4.

(4) In third-subfield data entries, only nine digits, including the decimal but not including the exponent field, can be used, i.e., 123456.89E07, but not 123456.789E07.

(5) The Z entry must be of the form: 738Z, not Z738 or 738 Z.

(6) The + or - data operators are not needed and are not available.

(7) The Q, N, and M entries are restricted: 3Q4, 1N4, or M4, but not 4Q, 4N, or 4M. G is similarly restricted.

(8) A field must not span two cards.

(9) All items on a card entered after a slash in any column except the first are ignored.

User-Field Input - If the user follows the array identifier in the array originator field with the character "U" or "V", the input format is to be specified by the user. If "U" is specified, the FORTRAN FORMAT to be used must be supplied in columns 1-72 of the next card. The format must be enclosed by the usual parentheses. Then the data for the entire array must follow on successive cards. The rules of ordinary FORTRAN input as to exponents, blanks, etc., apply. If the array data do not fill the last card, the remainder must be left blank.

"V" has the same effect as "U" except that the format read in the last preceding "U" array is used.

INTERNAL DISTRIBUTION

1. L. S. Abbott
2. D. E. Bartine
3. A. A. Brooks/CSD Library
4. H. P. Carter/CSD Library
5. S. N. Cramer
6-15. M. B. Emmett
16. W. W. Engle
17. G. F. Flanagan
18. T. A. Gabriel
19. H. Goldstein (consultant)
20. R. F. Hibbs
21. T. J. Hoffman
22. D. T. Ingersoll
23. J. L. Lucius
24. R. E. Maerker
25. F. C. Maienschein
26-35. W. A. Rhoades
36. R. T. Santoro
37. C. O. Slater
38. J. S. Tang
39. P. R. Vanstrum
40. C. R. Weisbin
41. J. T. West
42. G. E. Whitesides
43. A. Zucker
44. P. Greebler (consultant)
45. W. B. Loewenstein (consultant)
46. R. E. Uhrig (consultant)
47. Prof. R. Wilson (consultant)
48-49. Central Research Library
50. Y-12 Document Reference Section
51-53. Laboratory Records
54. Laboratory Records - RC
55. ORNL Patent Office
56. Radiation Shielding Information Center

EXTERNAL DISTRIBUTION

57. V. R. Cain, SAI, 800 Oak Ridge Turnpike, Oak Ridge, Tennessee, 37830

58. R. K. Disney, Westinghouse Electric Corporation, Advance Reactor Division, P.O. Box 158, Waltz Mill Site, Madison, Pennsylvania, 15663

59. W. C. Hopkins, Bechtel Power Corporation, P.O. Box 607, 15740 Shady Dell Road, Gaithersburg, Maryland 20760

60. J. N. Rogers, Division 8324, Sandia Laboratories, Livermore, California 94550

61. Walt Zobel, TVA, W-9 C149C, 400 Commerce Avenue, Knoxville, Tennessee 37902

62. Chief, Mathematics and Geoscience Branch, Department of Energy Washington, DC 20545

63-89. Technical Information Center, Department of Energy, Oak Ridge, Tennessee 37830

90. Research and Technical Support Division, Department of Energy, Oak Ridge Operations, Oak Ridge, Tennessee 37830