



A11106 262707

NIST  
PUBLICATIONS

REFERENCE

FILE COPY  
DO NOT TAKE

**NBSIR 87-3534**

# **DCTDOS: Neutron and Gamma Penetration in Composite Duct Systems**

---

L. V. Spencer

U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards  
Center for Radiation Research  
Gaithersburg, MD 20899

February 1987

C  
00  
56  
7-3534  
1987

Prepared for:  
**Federal Emergency Management Agency**  
Washington, DC 20472



NBSIR 87-3534

**DCTDOS: NEUTRON AND GAMMA  
PENETRATION IN COMPOSITE DUCT  
SYSTEMS**

---

L. V. Spencer

U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards  
Center for Radiation Research  
Gaithersburg, MD 20899

February 1987

Prepared for:  
Federal Emergency Management Agency  
Washington, DC 20472



---

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*  
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*



DCTDOS: NEUTRON AND GAMMA PENETRATION IN COMPOSITE DUCT SYSTEMS

	page
List of Figures . . . . .	iii
I. Introduction . . . . .	1
II. Structure of DCTDOS . . . . .	5
III. Input Data and Options . . . . .	7
A. Run-Description Card . . . . .	7
B. Problem Description Cards . . . . .	8
C. Configuration Cards . . . . .	10
IV. Output Selections and Elementary Examples . . . . .	13
V. Comparison of Results with Some Classic Experiments . . . . .	17
A. ORNL Studies . . . . .	17
B. Armour Research Foundation Experiments Using Co <sup>60</sup> Gamma Rays. . . . .	23
VI. Procedures, Variables, and Basic Data . . . . .	28
VII. Concluding Remarks and Acknowledgment . . . . .	36
References and Bibliography . . . . .	37
Appendix A. Some Notes on Subprograms . . . . .	44
Appendix B. Sources of Data . . . . .	52
Appendix C. Remarks on Inputs and Outputs with Examples . . . . .	55
Appendix D. FORTRAN Listing of DCTDOS . . . . .	62



## List of Figures

	Page
Fig. 1. Diagram of DCTDOS . . . . .	6
Fig. 2. Example of elementary input and output of DCTDOS . . . . .	14
Fig. 3. Example of elementary input and output of DCTDOS . . . . .	15
Fig. 4. Detector pattern for ROOMX and DOSX, NW = 3, NL = 4 . . . . .	16
Fig. 5. Top view of source and duct geometry . . . . .	18
Fig. 6. Thermal-neutron flux in various duct configurations due to incident thermal neutrons . . . . .	19
Fig. 7. Thermal-neutron fluxes in various duct configurations due to incident epicalcium neutrons . . . . .	20
Fig. 8. Comparison of calculated and measured center-line single- collision fast-neutron dose rates . . . . .	21
Fig. 9. Cumulative epithermal neutron spectra . . . . .	24
Fig. 10. Gamma dose distribution measurements in the 6 by 6 foot concrete entranceway . . . . .	25
Fig. 11. Measured gamma dose rate in 6 by 6 foot concrete entranceway with two right angle bends . . . . .	27
Fig. C1. Example of output from IPRINT=3, . . . . .	56
Fig. C2. Input data for computations of the ORNL duct . . . . .	60
Fig. C3. Input data for computations of the ARF duct . . . . .	61





# DCTDOS: NEUTRON AND GAMMA PENETRATION IN COMPOSITE DUCT SYSTEMS

L. V. Spencer

National Bureau of Standards

Washington, D.C. 20234

## I. Introduction

This paper describes computer methods for estimating neutron and gamma ray fluence rate, dose, and even spectral features due to penetration through a series of duct segments. The procedure links together data for individual segments -- straight sections and bends -- in arbitrary combinations; and the resulting composite can include computations for a room at the end, if there is one. This particular method was developed for rapid estimates in the protection problems against nuclear weapons; but the concepts which are employed are more broadly applicable.

The study of radiation penetration in ducts and other cavities has developed a large literature, which can be subdivided into a) studies of neutron and gamma ray albedos,<sup>5,13,18-24,29,38,43-45,58,61,78</sup>, b) experiments,<sup>1,3-4,6,8-9,12,15-16,27-28,30,33-34,37,39-40,48,54,59,70-71</sup>, c) Monte Carlo duct analysis,<sup>17,25,33-37,43,46-50,52-59,75</sup>, and d) some form of analytical analysis.<sup>7,12-13,26,31-32,41,53,60,62-70</sup> Summary discussions of the various problems, data, and methods are to be found in many places, among which are Refs. 15,41-42,55-58, and 74.

Many reactor shielding studies are not referenced above.

The program described here concludes a series of studies which started in the middle and latter 1960's. In these studies various aspects of radiation penetration through ducts were examined in regards to the possibility of viewing complex duct configurations as composites of elementary segments.<sup>67-69</sup> Many ideas appearing in this paper were described previously in Ref. 69. While this approach fits mainly with category d) above, it is neither a quadrature nor a curve-fitting method, nor does it emphasize center-line dose.

This report presents a usable computer program (DCTDOS) based on concepts described in that earlier study and gives applications to some well known experimental duct studies. Several of our basic assumptions are as follows:

1) That the relevant flux,<sup>\*</sup> which depends on both space and direction variables, can be usefully described by introducing the order-of-scattering index  $n$  (or  $N=n+1$ ). In DCTDOS we include up to  $n = 20$ , which is barely large enough for complex systems with large reduction, and may eventually be enlarged.

2) The albedo angular distribution for neutrons is well approximated by the elementary distributions of Chandrasekhar<sup>11</sup> for isotropic scatterers.<sup>69</sup> Since the sensitivity of most problems to details of this angular distribution is rather low, we use it to obtain solutions to elementary spatial configurations in terms of the variable  $\underline{n}$ . That is, we use one-velocity, isotropic-scatterer albedos to obtain data for simple segments as functions of  $\underline{n}$ .

3) The neutron flux component which has been backscattered from concrete walls (say) a total of  $\underline{n}$  times has a spectral distribution (for a given radiation source) which is assumed to depend on energy only through  $\underline{n}$ . This takes advantage of a near-factoring of the single-reflection albedo into a directional part and an energy part.<sup>67-69</sup> But

---

\* Here and throughout this manuscript the term flux stands for fluence rate. (See ASTM Standard E170, 1986 Annual Book of ASTM Standards, Section 12.) Flux can also designate a time integral, i.e., a fluence if the source is likewise integrated over time.

in turn this leads us to assume a factoring of the whole problem into space-and-direction studies, and multi-reflection energy studies. Tables of multi-reflection energy albedos as functions of both initial and eventual spectral energy are required.<sup>69</sup>

4) In using the output of one segment as input into another, some information about the direction of particle currents is a requirement. We have reduced this requirement to its least restrictive form by using only a single parameter: For this we use the mean reciprocal cosine of the obliquity, and which we refer to in the convenient form  $(-1 + 2/\langle \sec\theta \rangle) = s$ . Note that  $\langle \sec\theta \rangle$  is also the ratio flux/current. We record and use  $\underline{s}$  as a matching parameter, distinguishing between unreflected and reflected components in the preceding of any two segments being matched, since these can be very different.

(5) Radiation reflected in cylinders of circular cross section, spheres, and between infinite plane surfaces has the property of reentrance at the emergence obliquity when travel in the wall can be ignored. Hence multi-reflection albedos in these configurations are identical if the initial incidence distribution and spectrum are identical. We take advantage of the fact that cylinders with circular and square cross sections have a rough equivalence, as do cubes and spheres. Thus we believe that in a fairly usable first approximation multi-reflection distributions between infinite plane surfaces has a general applicability to far more complex configurations.

Combination of segments proceeds by simple convolutions of successive functions of  $\underline{n}$ . Spectra and different types of dose are extracted by additional convolutions with multi-reflection energy data. Hence DCTDOS marches to its solutions by a series of convolutions over data interpolated for elementary segments. As may be readily guessed, such a process is fast and can be flexible.

Most of the data on duct sections which DCTDOS utilizes were developed in a series of unpublished studies by S. Woolf.<sup>76-77</sup> The

multi-reflection albedo energy data for neutrons is due to Hagan and Simmons,<sup>38</sup> and corresponding data for gamma rays has been computed by Spencer and Woolf.<sup>78</sup> Since all data are rather simple in concept and readily obtained on file, we anticipate supplementation and replacement if these methods are further developed.

Gamma ray albedos are more complex than neutron albedos, with strong dependence on energy above 0.5 MeV [and for single scattering]. If the single-scattering is given special attention, there is merit in treating gamma rays as if they were neutrons, but using multi-reflection albedo energy data for gamma rays specifically. This simple approach turns out to work rather well. This confirms that the  $n = 1$  reflection component for gamma rays dominates the difference from the corresponding neutron angular distribution, and also that the grazing incidence case, sometimes termed "skipping," is comparatively unimportant. Differences of albedo magnitude in the basic segment data can be roughly corrected by use of simple ratios raised to the  $n$ 'th power. Recomputation of the space-angular data for gamma rays specifically has not yet been done.

In Section V we present results for both neutrons and gamma rays in comparison with experimental data and results from well known and detailed computations: for neutron data in the Maerker and Muckenthaler duct<sup>46-49</sup> and for gamma ray data in one of the AMF ducts.<sup>73</sup> We discuss spectral and order-of-reflection comparisons as well. A high degree of precision is neither attained nor expected; but we feel that absolute values and trends are satisfactorily reproduced for both types of radiation.

We have organized this report to put practical information necessary to apply DCTDOS towards the front, with explanatory and supplementary material located towards the rear. The comparison of results for classic experiments constitutes a transition section between these two types of information.

## II. Structure of DCTDOS

In the diagram of Fig. 1, the capitalized words represent subroutines, the function of each being suggested by choice of name. Data input are of three different types. (INPUT) represents problem parameter data. The basic data on which the program depends are introduced by calling ENTDAT to set up elementary segment data and SRSDET for spectra and multi-reflection response weights, both making use of types of multi-reflection albedo energy data.

Parameter input begins with new Run data, and later is indicated by four parallel lines designated Run, Problem, Length, and Weights. A new Length changes only a single (configuration) parameter, while a new Run brings in all the data necessary for a new duct computation. A new Problem uses previous source and detector information, while "New Weights" changes source and/or detector while retaining the previous configuration.

Branch points have been identified by lower-case letters a through i, their significance being as follows:

- a) Choice between DCTISO (isotropic source flux) and DCTSLT (mono-obliquity source angular distribution).
- b,c,d) The option here has to do with completion vs. an additional segment of duct, according to the complexity of the configuration.
- e) The configuration here may call for room computations or for dose computations in the duct.
- f) Choice of multi-reflection albedo energy data and application of these data to the dose weights: neutrons (SND) or gamma rays (SGD).
- g) This option permits recomputation with different dose weights. The rerun through SRSDET gives wide latitude to this option.
- h) This option permits modification of one of the straight segment lengths, followed by recalculation with all other parameters fixed.
- i) The choices here are for a new configuration with unchanged source and detector data (16) or a fully new situation (New Run, 13). Termination is also an option at this point.



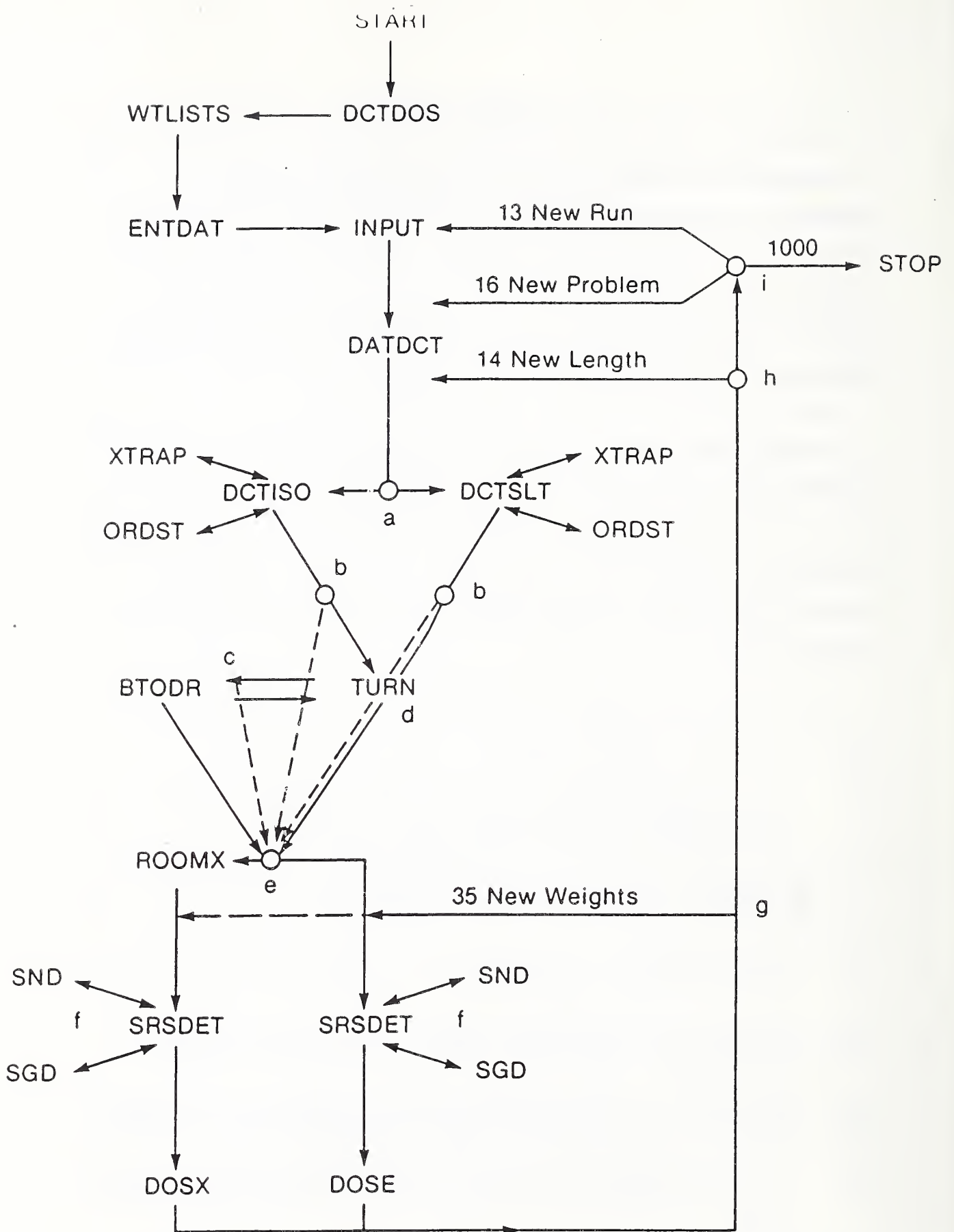


Fig. 1. Diagram of DCTDOS. The numbers identify orders in the control routine.

### III. Input Data and Options

We use the term "card" here with reference to a line of input data, which formerly was punched on an IBM card.

In preparing for a DCTDOS run, several different types of data card are required. The first specifies mainly source and detector data to be used, and we refer to this as a Run-Description card. Following this is a card with index numbers mainly for specifying outputs, new or old dose weights, and a series of segment lengths or detector positions, together with a title card. We refer to these as Problem-Description cards. Finally, there is a set of cards giving dimensions for a sequence of duct segments, which we refer to here as Configuration cards. Sample run decks are indicated in Figs. 2 and 3; here we mainly give descriptions, together with the options currently available.

A. Run-Description Card. The first run card specifies five integers, followed by a decimal number in exponential form, using the Fortran format (515,1PE10.3). The numbers, in their order, have the following interpretations:

- ISORS specifies the type of radiation spectrum selected,
- IDOSE specifies the type of detector response selected,
- ING specifies gamma rays or neutrons, with the inclusion or dismissal of wall-capture gamma rays,
- LPIPE selects from several types of (straight) duct data available for neutrons (only a single type of gamma ray data is used),
- IEL first specifies the type of output by quick identification with IPRINT. Later this parameter has a variety of other uses through its COMMON storage in most subprograms.
- DIMEN is a numerical factor which can change or rescale output units.

Table 1 lists options currently available according to values of the five index numbers referred to above. But special spectra or response functions can be readily inserted, either by adding to the lists above, or by insertion as a replacement of one of the cases listed in Table 1. We have found this useful in substituting  $^{137}\text{Cs}$  source data, say, for a  $^{60}\text{Co}$  gamma spectrum.

Table 1. Options currently available, as specified by the different input integers.

ISORS

- 1: Neutron INR spectrum (neutrons/group/incident dose), see Ref. 2, Table 10)
- 2: Thermal neutron spectrum (normalized to unity)
- 3: Epithermal neutron spectrum (neutrons/sec/watt/cm<sup>2</sup>) incident. See Ref. 48, Fig. 2, based on 34,200 n/sec/watt into (91.44 cm)<sup>2</sup>.
- 4: Gamma ray INR spectrum (gammas/group/incident dose). See Ref. 2, Table 10).
5.  $^{60}\text{Cobalt}$  gamma rays (normalized to unity). 50 Curies, 2 photons per interaction; but only half moves forward into a duct of 6'x6' cross sectional area. This gives 3.319E9 gammas/cm<sup>2</sup>/min.

IDOSE

- 1: Snyder-Neufeld dose (rads/(n/cm<sup>2</sup>))
- 2: Unit weights (for flux computations, e.g. neutrons/(n/cm<sup>2</sup>))
- 3: Henderson tissue dose data (rads/(n/cm<sup>2</sup>))
- 4: Thermal neutron response only (neutrons/(n/cm<sup>2</sup>))
- 5: Gamma-ray absorbed dose data (rads/(gamma/cm<sup>2</sup>))

ING

- 1: Wall-capture gammas only
- 0: Neutrons only, or gammas only
- 1: Neutrons + wall-capture gammas

LPIPE

- 1: Finite-duct (exit) neutron data
- 2: Semi-infinite-duct data, forward
- 3: Ditto, both forward and return
- 1-3: Any value for gamma rays

IEL

- 0: (IPRINT)-- No print-out, first case;
- 1-4: (IPRINT)-- Print-out (Table 2, also see discussion in Section IV)

DIMEN

This constant factor can change dimensions for the weights and/or renormalize the input current.

B. Problem Description Cards. Let us first dismiss the title card, which is the last of these cards. Its format is presently (55H ... ).



The first input card has again the format (5I5,1PE10.3). The integers specified have the options and interpretations listed in Table 2 below, with entries in the same order as on the card.

Following the main input card is a card with a single integer specifying the number (JDOS) of alternative sets of weights. And this is followed by JDOS alternative-weights cards, each again using the format (5I5,1PE10.3). These cards list values for IPRBLM, ISORS, IDOSE, ING, LP, and DIMEN in that order. It has turned out that of these, IPRBLM is a dummy without application, and can be equated to its value on the first input card.

Table 2. Options for the "Problem" Parameters

- IPRBLM: 1: The after this after this one is to use the same run-weights card.  
2: The computation after this one will have a new run-weights card.  
3: This is the last computation.  
-1,-2,-3: The present computation will use weights already computed and stored from the run-weights card which precedes.
- NEL: This gives the number of duct elements (plus one, for the source). This is also the number of case cards which follow.
- IPRINT: Acceptable values are 0, 1, 2, 3, 4. Zero indicates no printout; 1 means printout of input information together with computed dose and dose ratio; 2 also gives weights plus dose information at earlier junctions; 3 adds group spectra and mean secant directional information; and 4 adds intermediate printout from the subroutines for use in checking parameter sizes and other inquiry.
- JVARY: This gives the index (IC) for the straight section whose length variation may be desired. Thus, JVARY = 2 would refer to the first straight section of a duct.
- JDEL: This index gives the number of values which the straight section specified by JVARY is to have. JDEL = 1 means that only one value is to be computed, the most common case: The values of JVARY and DEL are then not relevant and DEL may be omitted or set to zero.
- DEL: This fixed point number gives the increment by which the length of the JVARY segment is to be increased for each of JDEL cases. It is important that DEL is length, i.e., length divided by the square root of the cross sectional area.

C. Configuration Cards. These cards in sequence describe the configuration. We discuss them in terms of the names of the lists into which they are put (in DATDCT). These lists are NEXT, WPTH, HGHT, LNTH, OTHR, SELECT: The Format used is (I5, 5F10.3)).

The first entry, NEXT, gives IEL values for a series of subprograms to be called in the sequence given (DCTDOS, order 35). These are listed in Table 3 below for easy reference:

Table 3. NEXT indices used to call the main subprograms.

<u>IEL</u>	<u>SUBROUTINE</u>	<u>Type of Section</u>
0		Descriptive parameters, only used with IC = 1
1	DCTISO	Straight segment, isotropic source
2	DCTSLT	Straight segment, slant source
3	TURN	Bend segment
4	BTODR	Straight segment occurring after a bend
5	BTODR	Straight segment after a bend
6	ROOMX	Room segment
7	DOSE	Applies weights and computes total, detector in or at the end of, a bend or straight section
8	DOSX	Applies weights and computes total, detector pattern in a room

The remaining entries on the case description cards include widths, heights, and lengths. We summarize these in Table 4 using the abbreviations, W, H, and L, respectively. We use dimensioned numbers, usually in feet, in practice; the program does its own scaling. Those parameters which are not W, H, or L require special explanations, which follow Table 4.

Table 4. Input data assignments for the case description cards. (Note Appendix B.) NEXT corresponds to IEL in Table 5.

	<u>NEXT</u>	<u>WPTH</u>	<u>HGHT</u>	<u>XLTH</u>	<u>OTHR</u>	<u>ELECT</u>
(IC=1)	IZD	SO	PEN	ZDOOR	RMCase	
	1	W	H	L	ARAT	S = 0
	2	W	H	L	ARAT	S = -1,0,1
	3	W	H	RCSS	F	S = -1,0,1
	4,5	W	H	L	ARAT	S = -1,0,1
	6	W,WTH	H	L	WD	S = -1,0,1
	7			RDLNTH		S = -1,0,1
	8	WDP	NW-1	NL-1	AEX	S = -1,0,1

The following interpretations apply to the various quantities in Tables 3 and 4.

- SELECT(IC) is normally zero or omitted. If it equals 1., the reflected component is equated to zero for computations by the subroutine selected by NEXT(IC). If it equals -1., the unreflected component is equated to zero. If 0., the normal case, both contribute. Note that "unreflected" and "reflected" here refer only to the preceding section and not to  $N = 1$  and  $N > 1$  in general.
- SO This index is zero for the cosine (current) source. In general this quantity is  $= -1 + 2./\langle \sec \theta_0 \rangle$  for slant obliquity  $\theta_0$ .
- PEN is an assumed effective scattering depth in side-walls prior to reemergence. It has normally been set equal to zero; its dimensions are those of W, H, and L.
- ZDOOR is a door or wall thickness assumed (for  $IZD > 0$ ) at the entrance to the shelter. Note: It is limited to dimensions of feet of concrete presently, and assumes no modification of source spectrum or directional distribution.
- RMCASE has values +1., 0., -1.: The value +1. is used if the flux entering the room (i.e., ROOMX) and unreflected by the preceding segment, is directed towards a nearby sidewall. The value 0. is assigned if this flux component has no strong bias towards or away from a sidewall. The value -1. is assigned if this flux component (VELO) is directed towards the array of detectors. Most commonly, RMCASE = 0.
- RCSS gives the depth of the recess, in cases of a bend with a trap. Units are the same as for other XLTH values.
- F is an azimuthal factor for the radiation entering the bend. Normally it has the value zero.  $(1 - F)$  is applied as a factor to weight the component which traverses the bend without wall reflections in the bend. Hence, for  $F=1$ , the source entering the bend cannot exit without reflections, while for  $F=-1$ , the current exiting without reflections (i.e., VELO) is doubled. (Note that the configuration can modify this component in ways not expressed by basic bend data.)
- RDLNTH Measures the distance from detector to the backscattering endwall of a straight duct segment. It is used in RETRN to estimate and include backscattered current. It should be omitted or set zero if this component is ignored, which is the most common case.
- WD is the width of the entrance door into a room section (ROOMX).

WDP is the width of the pattern of detectors, i.e. the distance from the center of the duct exit to the wall on the side where the detectors are positioned, in the shelter room (ROOMX).

NW,NL (both fixed point numbers quickly converted to integers) are lateral and longitudinal numbers of detectors in the array (see ROOMX).  $NW \cdot NL \leq 20$ , presently, by dimension statement.

AEX is a ROOMX parameter. It is the area of the entrance to the room (IEL=6). Note that IEL = 7,8 are alternatives; and DOSX, rather than DOSE would be used with ROOMX.

ARTO is a ratio: (initial duct cross section area)/(final duct cross section area). Here, initial and final refer to the first and last duct segments. For DOSX, the final aperture area is the same as AEX.

RAT is the ratio of duct cross-sectional area taking account of a mean scattering depth (PEN) to that with PEN = 0. It would normally be unity.

ARAT is the ratio of cross-sectional areas at a junction (the value for the next segment divided by that for the preceding) if there is a reduction.

#### IV. Output Selections and Elementary Examples

The input parameter IPRINT has values 0-4 as already noted. The value IPRINT = 0 suppresses printout and has no interest to us here. The value IPRINT = 4 prints additional intermediate data and parameters, and is useful mainly to inquire into the causes of hard-to-explain output; we will not attempt to discuss this option further. Desired intermediate output may change with one or another particular requirement or investigation.

IPRINT = 1 causes only input data, together with minimum output, to be printed. Input data have been described in the preceding section and are readily recognizable in the printout. Also these cards and their sequence can be confirmed from the list INPUT lines added to the top of Figs. 2 and 3. OUTPUT begins by recording the input data under suitable headings. This is followed by values for the dose and dose reduction factor as indicated by printed titles. The heading OUTPUT has been added to clarify the figure and does not accompany the output. The heading "NEXT CASE = .500," begins the printout of results for Fig. 2, and this heading appears in general with output results. The number .500 gives distance down the final straight segment specified by JVARY, and may be zero if the dose is computed at the entering point for a new segment.

If JDEL = 1, the usual case, this distance will be the scaled value corresponding to XLTH for the segment identified by JVARY. Thus, the duct of Maercker and Muckenthaler (see Fig. 5) has a mean lateral extension of  $\sqrt{(3' \times 3')} = 3'$ ; and for JVARY = 4 the value NEXT CASE = .500 identifies the position  $.5 \times 3' = 1.5'$  down the segment identified by IC = 4. Since IC = 2 is the first straight segment and IC = 3 is the first bend, IC = 4 is the second straight segment. The possibility for incrementing the XLTH value specified by JVARY by one or more distances DEL up to a total of JDEL-1 times exists. Then output of the form described is repeated, occurring JDEL times in succession, with appropriately changed headings, as shown in Fig. 2.



```

INPUT
  2  2  0  3  1 1.973+000
  1  5  1  4  4 1.333+000
  0
*** M+M, 2-LEG, THERMAL FLUX (USES BTODR), 8/29/86 ***
  0  0.414  0.000  0.000  0.000  0.000
  2  3.000  3.000  13.500  0.000  0.000
  3  3.000  3.000  0.000  0.000  0.000
  5  3.000  3.000  1.500  0.000  0.000
  7  0.000  0.000  0.000  0.000  0.000

```

```

OUTPUT
          $=====
ISORS IDOSE ING LPIPE IEL DIMEN
  2  2  0  3  1 1.973E+00
IPRBLM NEL IPRINT JVARY JDEL DEL
  1  5  1  4  4 1.333E+00

*** M-M, 2-LEG, THERMAL FLUX (USES BTODR), 8/29/86 ***

IEL  WIDTH-FT  HEIGHT-FT  LENGTH-FT  OTHER  SELECT
  0  .414      .000        .000        .000    .000
  2  3.000     3.000     13.500     .000    .000
  3  3.000     3.000     .000       .000    .000
  5  3.000     3.000     1.500     .000    .000
  7  .000      .000      .000      .000    .000

```

```

          NEXT CASE .500
NEXT(I)= 5 DOSE= 2.361E-02 REDUX FACTR= 8.459E-03

```

```

          NEXT CASE 1.833
NEXT(I)= 5 DOSE= 4.369E-03 REDUX FACTR= 1.565E-03

```

```

          NEXT CASE 3.166
NEXT(I)= 5 DOSE= 1.215E-03 REDUX FACTR= 4.355E-04

```

```

          NEXT CASE 4.499
NEXT(I)= 5 DOSE= 5.283E-04 REDUX FACTR= 1.893E-04

```

FIG. 2. Example of elementary input and output of DCTDOS. Data are for a duct with concrete walls and 3'x3' passageway cross section. Output is for the 2nd leg, at 1.5', 5.5', 9.5', and 13.5' beyond the bend, which in this case is cubic in shape. The (neutron) source is incident at 45° at all parts of the entrance plane and all azimuths.

```

INPUT
  2  2  0  3  1 1.973+000
  3  4  1  2  1 0.000
  0
**3X3X18 1-LEG DUCT INTO ROOM, THERMAL N FLUX, 8-15-83
  0  0.000  0.000  0.000  0.000  0.000
  1  3.000  3.000  18.000  0.000  0.000
  6  10.000  8.000  15.000  3.000  0.000
  8  5.000  1.000  2.000  9.000  0.000

```

```

OUTPUT
          $=====
ISORS IDOSE ING LPIPE IEL DIMEN
  2  2  0  3  1 1.973E+00
IPRBLM NEL IPRINT JVARY JDEL DEL
  3  5  1  2  1 .000E+00

```

```

**3X3X18 1-LEG DUCT INTO ROOM, THERMAL N FLUX, 8-15-83

IEL  WIDTH-FT  HEIGHT-FT  LENGTH-FT  OTHER  SELECT
  0  .000      .000      .000      .000   .000
  1  3.000     3.000     18.000   .000   .000
  6  10.000    8.000     15.000   3.000  .000
  8  5.000     1.000     2.500    9.000  .000
  7  .000      .000      .000     .000   .000

```

NEXT CASE 6.000

```

DOSE/ENTRANCE DOSE  W(FT)= 5.00  L(FT)= 15.00
1.0653E-02  9.0436E-03  5.3005E-03
1.4363E-03  1.5541E-03  1.3522E-03

```

```

DOSE
4.2036E-02  3.5686E-02  2.0916E-02
5.6678E-03  6.1324E-03  5.3359E-03

```

Fig. 3 Example of elementary input and output of DCTDOS. A single straight segment 3'x3'x18' ends in a room of dimensions 10'x8'x15'. The source of thermal neutrons is isotropic at all parts of the entrance plane. See Fig. 4 for the detector pattern: For 6 detectors, 1, 3, 5, are opposite the entrance and 2, 4, 6, are against the sidewall.

DOSX output for IPRINT = 1 gives arrays rather than single numbers. Thus in Fig. 3 output is in the form of two arrays -- for the dose ratio as well as for the dose itself. The heading for the former is DOSE/ENTRANCE DOSE, and for the latter, DOSE, as shown in Fig. 3. Both arrays have, as their upper left value, the entrance value to the room. The top line gives a sequence along the entering axis in equidistant positions crossing the room. Lower lines of detectors are displaced laterally in the room by equal amounts from entrance to sidewall and from front to back, as shown in Fig. 4 below.

1	4	7	10
2	5	8	11
3	6	9	12

Fig. 4. Detector pattern for ROOMX and DOSX, NW = 3, NL = 4. This figure differs from that for the Fig. 3 data in its proportions and in a non-centered entrance.

IPRINT = 2 introduces several additional types of information. Values are printed for the dose weights; also given are DOSE and DOSE/(ENTRANCE DOSE) data for the successive segment junctions of the system. Thus to obtain the dose weights used, one need run only the simplest of programs using IPRINT=2. But dose weights are also to be found in the file PREWTS. The types of data which are given with IPRINT=2 may be required, or they may provide a better understanding as to why the dose or dose reduction turn out to have the values printed.

IPRINT = 3 extends the output to include spectral values together with minimum intermediate parameters, mostly mean secants at junctions.

IPRINT = 4 prints additional intermediate values. Consult the program listing in Appendix D to identify and possibly supplement this seldom-used data selection.



## V. Comparison of Results with Some Classic Experiments

The comparisons which follow in this section are absolute, based on computations using known source strengths and absolute detector data. No relative comparisons have been included.

A. ORNL Studies. The most extensive studies of neutrons in concrete ducts are those carried out at ORNL by Maerker, Muckenthaler, and others.<sup>17-24,43-51</sup> Their 3' x 3' duct eventually reached 45' in length, with options of 1, 2, or 3 bends as shown in Fig. 5. In all their studies the source was a beam, directed through the opening at an angle of 45° relative to the duct axis. Measurements were made at points on the centerline.

The ORNL studies included computations of the albedo angular and energy functions for concrete, with a follow-on computer Monte Carlo study based on the resulting albedo data. These computations gave good agreement with experiments; and they also permitted examination of intermediate spectra and orders-of-reflection distributions. Hence there is a satisfying completeness about the information given in this set of reports. (Note particularly Ref. 49.)

Data were developed for a thermal neutron source and for a fast neutron source. In the latter case as well as in the former, thermal neutron flux was measured and computed. For the latter case measurements and computations for a Hurst dosimeter were also developed.

Fig. 6 shows the thermal neutron flux due to a thermal neutron source. Fig. 7 gives the thermal neutron flux due to an epithermal neutron source. Fig. 8 similarly gives the fast neutron dose due to the "epithermal" neutron source. In each case we have added points representing computations using DCTDOS, and the added points are readily identified. The following paragraphs discuss aspects of these (DCTDOS) results.

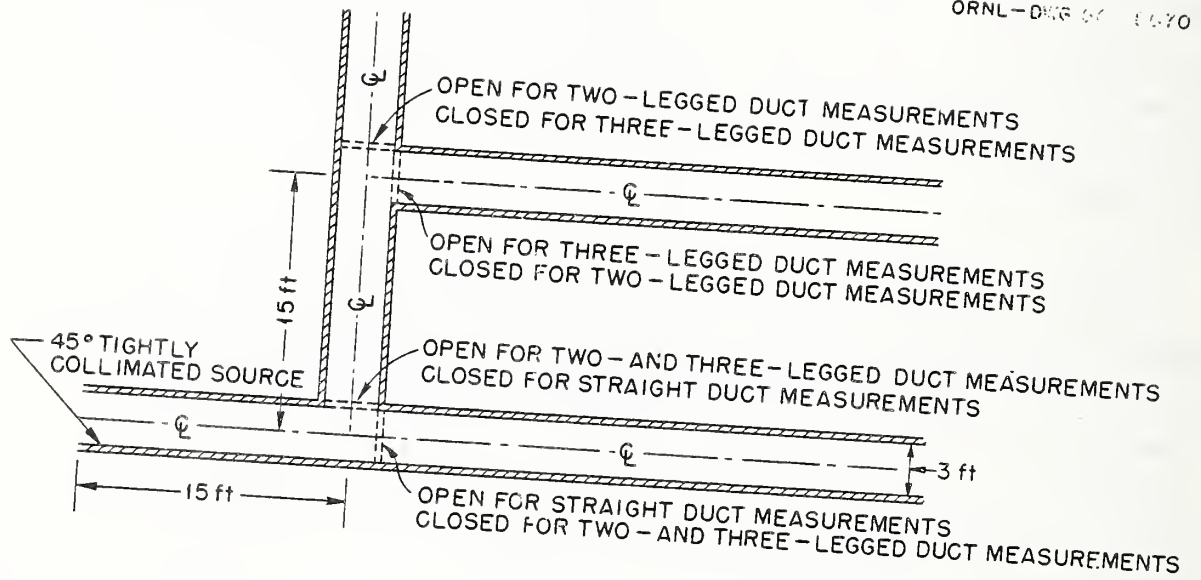


Fig. 5. Top view of source and duct geometry.

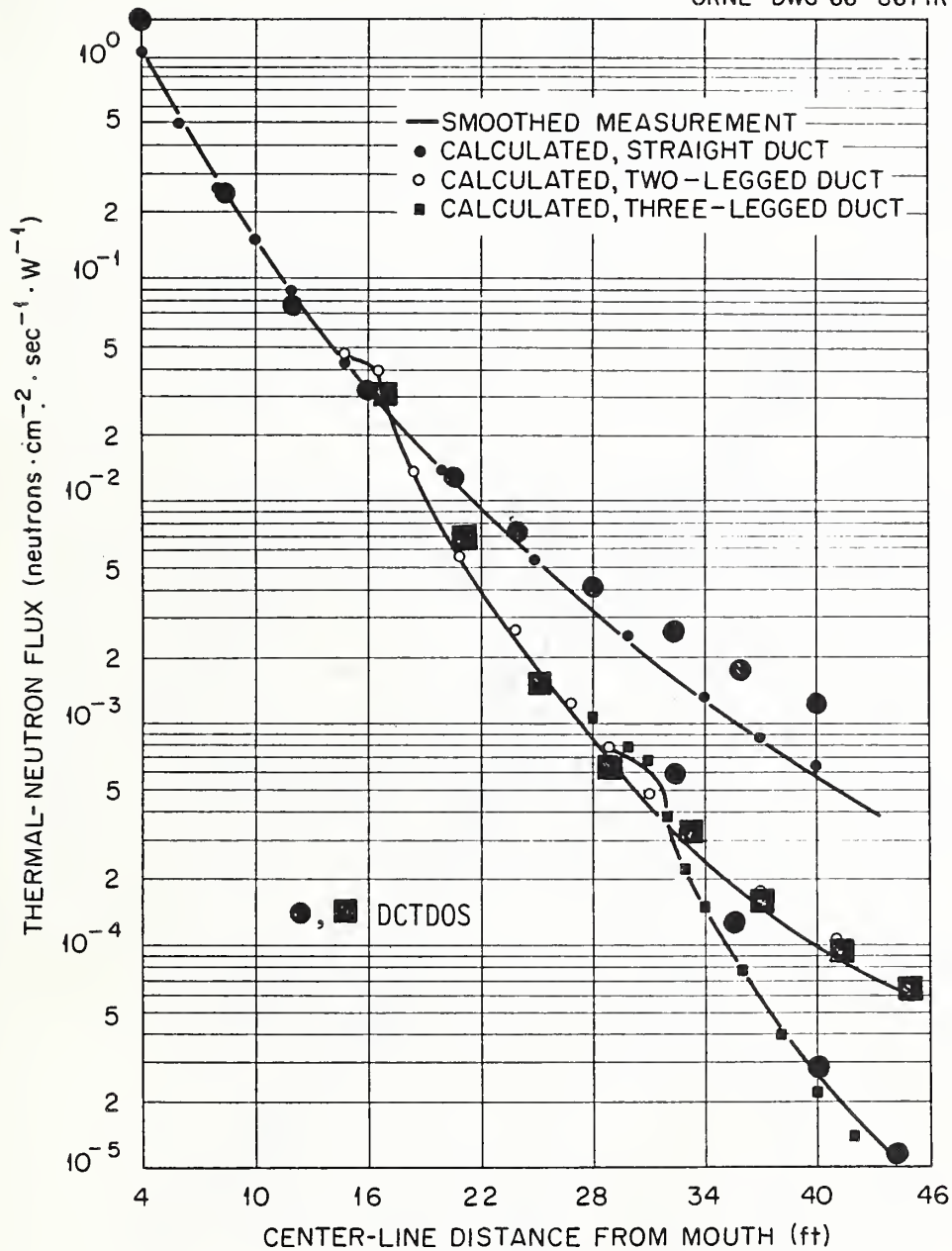


Fig. 6. Thermal-neutron flux in various duct configurations due to incident thermal neutrons

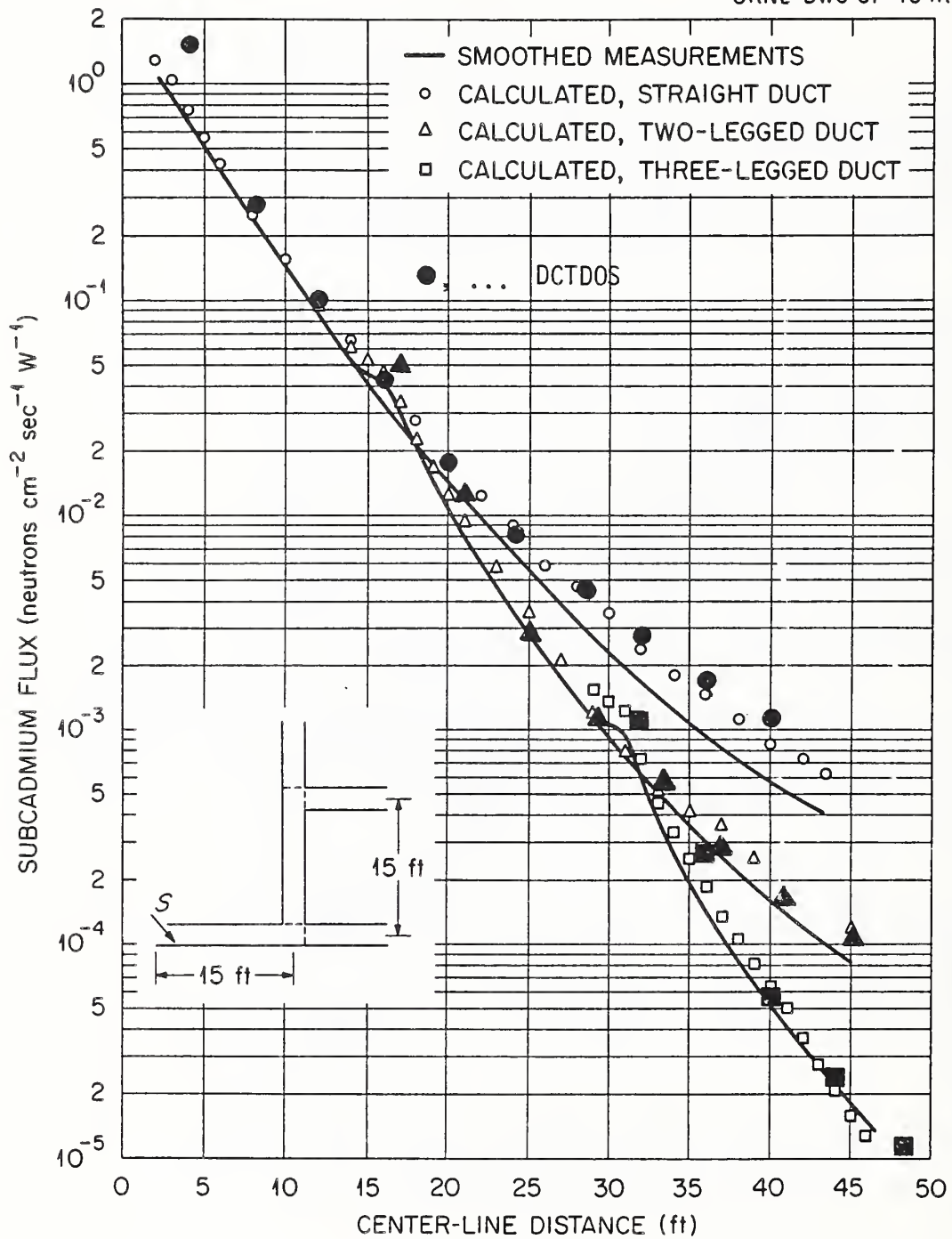


Fig. 7. Thermal-neutron fluxes in various duct configurations due to incident episcadmium neutrons.

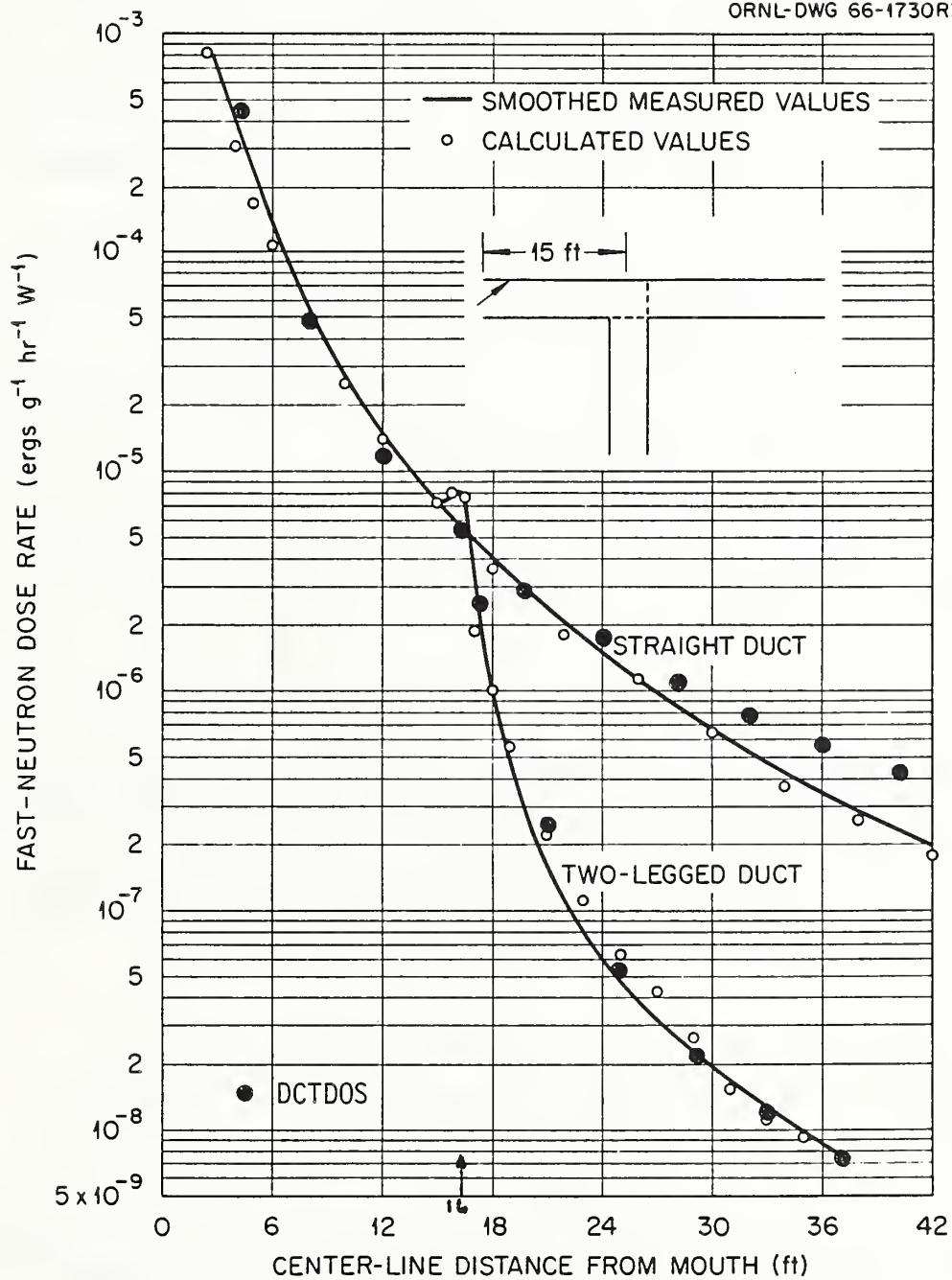


Fig. 8. Comparison of calculated and measured center-line single-collision fast-neutron dose rates.

A divergence near the source most likely reflects the difference between the experimental beam source and our source, which covered the aperture uniformly and at all azimuths, and which used interpolated data for the 45° obliquity angle rather than expressing that obliquity sharply.

An increasing divergence beyond 22' in the first straight segment is, we believe, a failure of our model. But note that a somewhat similar effect occurs in the ORNL computations. It may well indicate that near-grazing trajectories are poorly described in both models. In our computations the discrepancy occurs at distances requiring extrapolation beyond our data set; but a faulty extrapolation formula does not seem to account fully for the effect.

A rise as a bend is approached is probably due to back-reflection from surfaces marking the endwall of the bend. This can be estimated if required.

High values computed by DCTDOS just following a bend can result in part from differences between a centerline detector and the DCTDOS detector, which averaged over the lateral plane. This will be evident when lateral intensity variations are strong, as occurs when part only of the duct cross section is exposed to intensities direct from two segments preceding and possibly part of the source plane.

The general agreement with the experiments appears to be comparable with that of the Monte Carlo computations, if account is taken of the differences between source and detector configurations.

To pursue comparisons in greater detail, we computed both cumulative group spectra and orders-of-reflection components at several positions in the duct. The ORNL studies published computed examples of both types. Orders-of-reflection distributions were compared for equal centerline distances both down a duct continuation and around a bend. Results show that the bend shifts the peak to values for the orders-of-



reflection which are larger by 4-5, for both first and second bends. Our computations showed about the same shift. But our peaks were consistently located at smaller orders-of-scattering by about the same amount. We think that this results partly from our use of a source configuration which permits neutron injection initially to greater depths down the first leg. But in both ORNL computations and in ours, peak location is not strongly sensitive to distance down a given leg.

We chose cumulative rather than differential spectra for a comparison because the ORNL energy group structure is quite different from ours.<sup>49</sup> Fig. 9 gives the comparison between the two types of computed spectra at 14' beyond the center of the first bend. Below about .05 MeV the ORNL spectrum rises higher to a maximum excess factor of about 4/3 at an energy just above the thermal component. This is consistent with, although it may not fully result from, the larger number of scatterings apparently undergone by the ORNL flux at the same position. Note that the thermal components are about equal at this position, as shown in Fig. 7 at 29'.

#### B. Armour Research Foundation Experiments Using Co<sup>60</sup> Gamma Rays.

These experiments were not supplemented with computations; hence the types of comparison are much more limited than the above, although these data have also be analyzed since by other authors.

In Fig. 10, we show the experimental duct with three legs used by Terrell et al.<sup>70-73</sup> We have computed gamma attenuation in this concrete duct and compared the data from DCTDOS with the experimental data of Ref. 73. Our computations used a one-group approximation to the Co<sup>60</sup> spectrum. Similarly, we do not distinguish between three bends of the Z shape as distinct from the U shape, a difference which seems to be of minor importance here. Note that a point, 50 Curie source was located at S<sub>10</sub>, 3' inside the first leg at a point which we take to

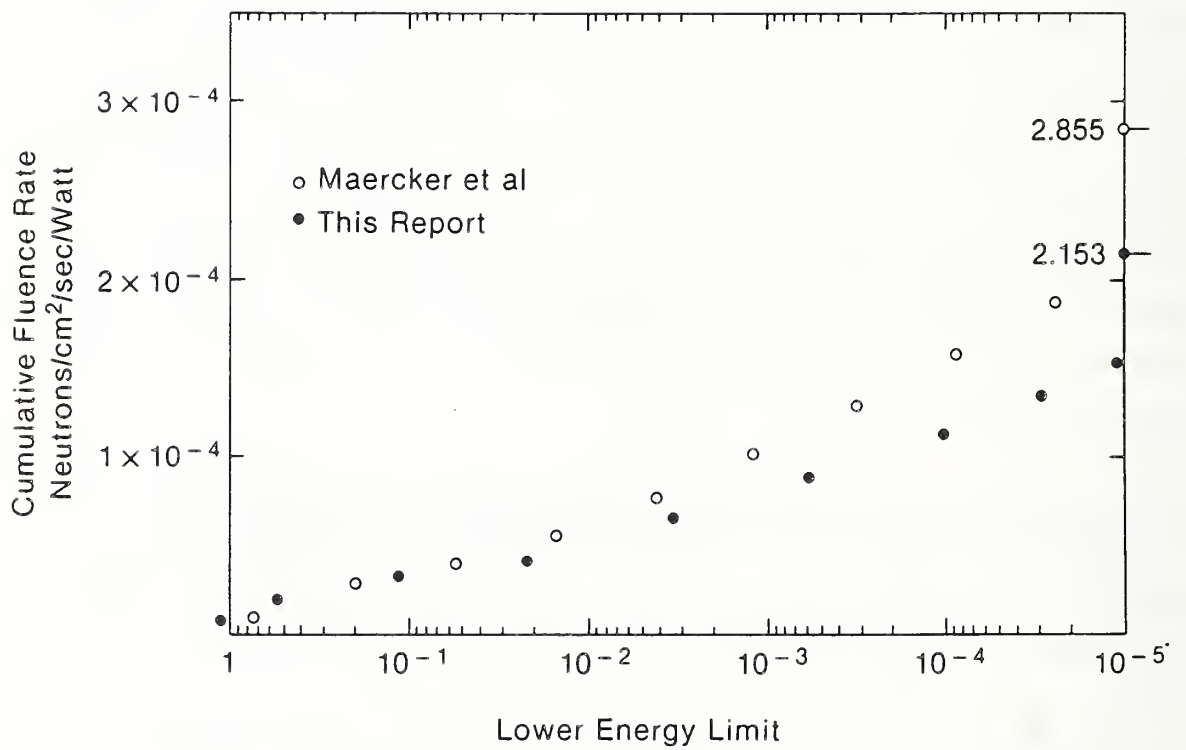


Fig. 9. Cumulative epithermal neutron fluence rates.



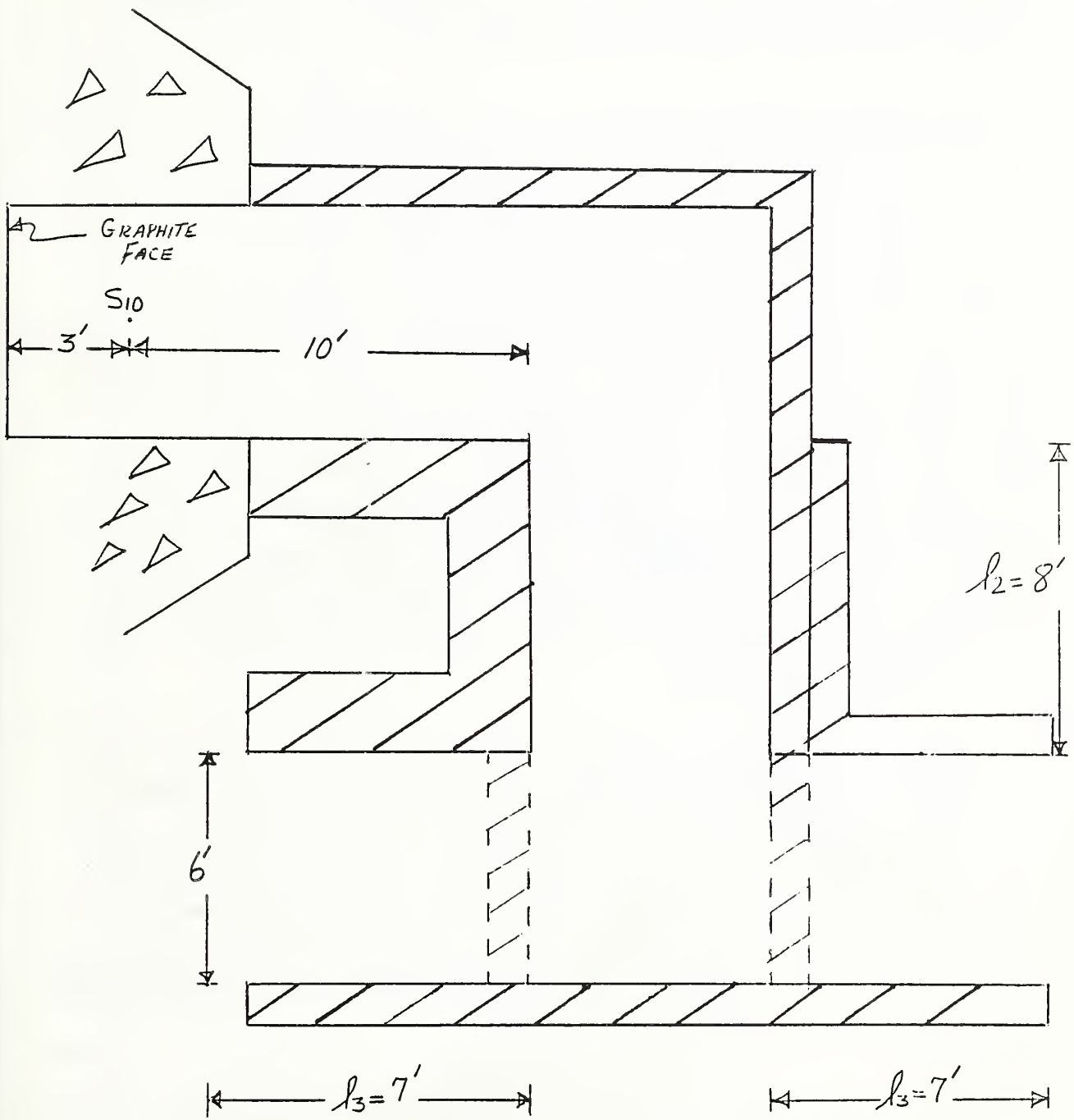


Fig. 10. Top view of the 6 by 6 foot concrete entranceway.

be the duct entrance for the DCTDOS computations. Recall that these are based on an isotropic input which is assumed uniform over the lateral entrance plane. Again the centerline measurements are compared with an average over the lateral plane used by DCTDOS. We have also ignored source gammas back-reflected into the duct.

The comparison is given in Fig. 11. Our results show less satisfactory agreement with the experiments than is the case for neutrons, as might be expected. The most evident difference appears within and just beyond the bend, where our computations do not reproduce the feature that as long as the (point-like) source can be viewed by the detector, the main falloff in the second leg is postponed. The consequence of this is that our main falloff occurs 1-2 feet later than in the experiment. We believe here also that at least part of the explanation for this effect results from a centerline approach vs an approach by lateral averaging. At the first bend, the large experimental reduction occurs when the point detector is first shielded from the point source. By contrast, a plane source is only gradually shielded from a plane detector, in this case total shielding occurring nearly 7' beyond the vertical lines which mark the turns to a new leg at the bend centers.

Regarding the factor of roughly 2 discrepancy in the third bend, we think that this is due to inadequacies of data expressing our model. In addition to differences of source and detector configuration, no attempt has at this point been made to incorporate and apply gamma ray albedo angular distributions to computations of basic segment data; neutron albedo angular distributions have been used instead, as noted. The substitution of more realistic data should improve accuracy somewhat, at the expense of duplicating much of ENTDAT.

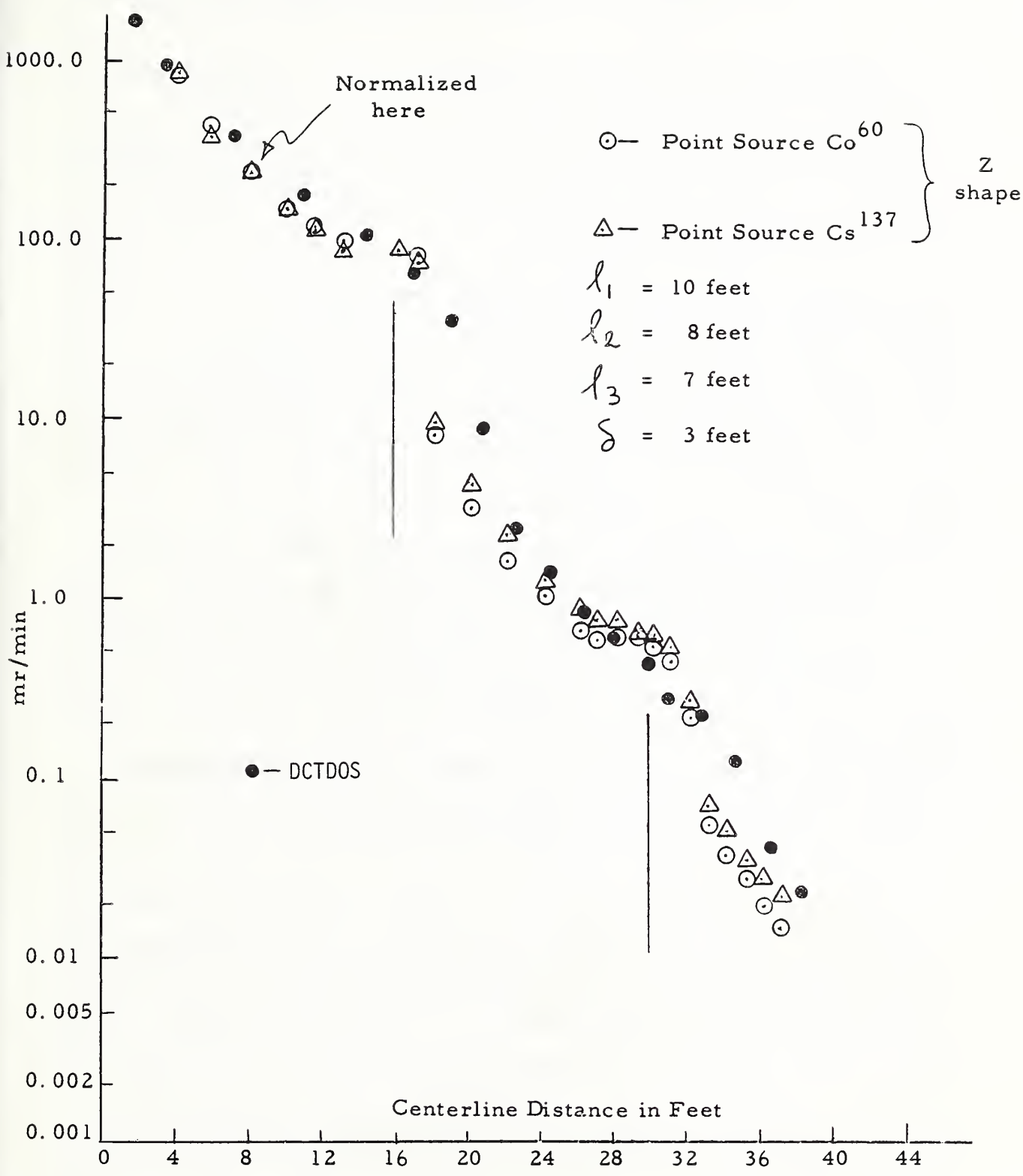


Fig. 11. Measured gamma dose rate in 6 by 6 foot concrete entranceway with two right angle bends (Z-Shape).

## VI. Procedures, Variables, and Basic Data

1) Position, Energy, and Direction (N, SO, SP, IC, DLNGTH, SENVAL). We identify position with the entrance or exit lateral planes in a straight duct segment, except in the special case of a detector pattern in a room. IC values sequence duct segments, and DLNGTH is straight segment length or detector position.

The energy variable is expressed through the order of backscattering, n or  $N=(n+1)$ . While there is expected to be a dependence of the energy on the sequence of types of backscattering, e.g., angles of incidence and emergence, in practice this dependence is assumed to be weak and we keep track only of the number. In addition, there is a dependence of the albedo on the neutron energy, which we neglect because it is not strong except in special cases which do not concern us here. Our basic data (for concrete) is based on an assumed albedo value of 0.696 throughout although more recent evaluations suggest that 0.75 is a better figure. Rather than recompute the basic data at this point, we neglect the differences of angular distribution as well as spectrum and merely include a correction factor  $(0.75/0.696)^n$  to take account of this, because these differences do not appear to have a strong influence on segment outputs. Since the dose weights have incorporated the correct albedo factor, only the correction factor  $(1/0.696)^n$  is applied in DOSE and DOSX.

The angular distribution is characterized, basically, by the ratio flux/current. This is an average of the reciprocal cosine over the radiation current. (It is also the reciprocal of the mean cosine averaged over the normalized flux.) We use this parameter partly because it gives a simple way of transforming currents into fluxes and vice versa. (Such a relationship can only be approximate because dose or other energy weighting factors give variant flux-to-current ratios.

We have found it convenient to use a derived quantity

$$s = \frac{2}{\langle \sec \theta \rangle} - 1 \quad . \quad (1)$$

Values of  $s$  for forward currents range from -1 to +1; and the largest value ( $\langle \sec \theta \rangle = 1$ ) corresponds to neutrons travelling parallel to the axis of penetration, which is nearly always the reference axis.

Dependence on only this single direction parameter is one of the basic limitations of our approach; but we have found that it works rather well. It is not difficult to identify configurations and directional distributions which would be poorly described so as to lead to unsatisfactory results. But such configurations are about as likely to suggest a requirement for new data types as to suggest a change in approach.

In DCTDOS, N or NREF always identifies  $(n + 1)$  rather than  $n$ ; and SO(IC), SP(IC) correspond to  $s$  values for unreflected and reflected components, respectively, emerging from the segment IC. (But note that IC = 2 is the first real segment of a duct system being evaluated.) SENVAL is  $s$  value for the current entering a bend.

2) Dependent Variables (VELO, VELP, RMTX). Our computations are based on currents, rather than fluxes, and VELO, VELP identify currents entering or leaving a duct segment. Specifically, VELP(IC,N) is the component of the current of neutrons reflected a total of  $(N-1)$  times and emerging from the duct segment identified by IC, having had at least one reflection in the IC'th segment. Correspondingly, VELO(IC,N) is the component of the emerging current which passed through the IC'th segment without any reflections. Its dependence on N signifies that it has no doubt been reflected in other segments, perhaps many times.

RMTX is intended to stand for "reflection matrix". The current entering any duct segment will be reflected 0, 1, 2, ...  $n$  times in that segment. These are in addition to, and must be combined with, previous reflections. The general procedure for doing this is the

evaluation of a simple folding computation. This is cast in the form of a matrix operation in which the incoming currents, VELO and VELP, are treated as vectors multiplied by an elementary matrix, RMTX, which expresses the effect of the new segment. This product then yields new "vectors" VELO and VELP constituting the output current. If one has a vector with components  $V_0, V_1, V_2, \dots$  together with a matrix which is zero to the right of the diagonal, has the value  $C_0$  on the diagonal,  $C_1$  in the positions just left of the diagonal,  $C_2$  in all the positions just left of the  $C_1$  elements, etc., then the product is

$$\begin{vmatrix} C_0 & 0 & 0 & 0 \\ C_1 & C_0 & 0 & 0 \\ C_2 & C_1 & C_0 & 0 \\ C_3 & C_2 & C_1 & C_0 \end{vmatrix} * \begin{vmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{vmatrix} = \begin{vmatrix} C_0 V_0 \\ C_1 V_0 + C_0 V_1 \\ C_2 V_0 + C_1 V_1 + C_0 V_2 \\ C_3 V_0 + C_2 V_1 + C_1 V_2 + C_0 V_3 \end{vmatrix} \quad (2)$$

Let the subscripts now stand for numbers of reflections, the  $V_n$  be values for the current entering an element, and  $C_n$  for the probability of emerging reflections in this element. Then the matrix to the right has a form suitable for the corresponding current emerging from the element. This assumes that the  $V_n$  are essentially interchangeable upon entrance. Sometimes we make this assumption; but more often we perform different computations for VELO and VELP, since the entering angular differences seem often to be important for these components.

RMTX always is given the general form of the matrix shown above, but if VELO and VELP are treated separately, RMTX should more accurately be computed separately for each, a refinement not in DCTDOS presently.

### 3) Data Forms for Structural Elements (BEND, DUCT0, DUCT1, DUCT2).

S. Woolf has performed many computations for duct segments of square cross section and different lengths or different positions in a very long, straight duct.<sup>76</sup> Other data from these computations explore more general shapes: rectangular rather than square cross section shapes.



Still other computations are for bends of square cross section but with lengths  $1 \times 1$  (cubical),  $1 \times 2$ , etc.; and recessed bends were examined. Study of Woolf's data leads to the empirical result that successive orders of reflection beyond  $n = 1$  nearly always follow rather well the simple rule

$$C_n = A \cdot a^n, \text{ for } n \geq 1, \quad (3)$$

where  $C_n$  is the  $n$ 'th reflected current at the duct exit, say. We assign to all of these components combined the angular parameter  $P$ . The  $n = 0$  case is usually quite different, and hence is assigned the different value  $S_0$ , as already stated.

Thus, to render Woolf's data in a form suitable for our computations, we require five parameters:\*

KX = 1: Co  
           2: A  
           3: a  
           4: S0  
           5: SP

---

\*These types of data are used in the basic programs DCTISO, DCTSLT, and TURN, where the sequence of orders-of-reflection computed is not that of Eq. (3) above, but a somewhat more complicated form which agrees better with the basic data. (Note the remarks about the subroutine ORDST, in Appendix A.)

Fixing the square cross section as basic, and the  $1 \times 1$  bend shape likewise as basic, we have, in the case of straight ducts, either the duct length or the distance to the detector as variable: All the above types of parameter are thus functions of this distance parameter. By contrast, in the case of the duct bend the dimensions are fixed but the angular input is not; and all of the above types of parameter are

functions of the s value of the incoming current. Thus, straight duct data are tabulated as a function of length or distance to the detector, both identified by DLNGTH. Bend data are rendered as a function of the s value (SEINVAL) for the entering current, which covers the entrance surface.

Straight duct data are based on an entering current proportional to  $\cos\theta$  relative to the duct axis in DUCT0, and on entering currents with mono-obliquity angles  $\cos\theta = 0.25$  and  $0.75$ , in DUCT1 and DUCT2. Other data are available, but we have preferred to interpolate or combine these, as required. In addition, data for finite ducts (LP = 1), for a fixed distance in a semi-infinitely long duct (LP = 3), and for the forward current only in the latter case (LP = 2) are included in ENTDAT. LPIPE and LP specify the input data type to be used.

4) Interpolation, Extrapolation of Segment Data. We rely mainly on a simple 2-point interpolation procedure applied to the logarithm of data points. Thus, if we wish data for duct of length L which falls between tabulated values  $L_1$  and  $L_2$ , and if the tabulated values for these two points are  $y_1$  and  $y_2$  respectively, the value y or the desired value at L is given by

$$y = y_1(y_2/y_1)^{(L-L_1)/(L_2-L_1)}. \quad (4)$$

This is exact for exponential distributions.

In using such a simple interpolation procedure, various data sets are modified to take account of the following problems:

- a) reduce range of variation by known divisor functions for direct use of DATA statements,
- b) ensure that neither  $y_1$  nor  $y_2$  is zero in a tabulation,
- c) counter a strongly non-exponential variation.

Such modifications are removed after an interpolation.

Also, an extrapolation procedure has been included to make possible rough estimates for extreme cases such as very long and narrow ducts.



This is based on the assumption that the distribution being extrapolated can be described by the general form

$$y = Ax^{-\mu} + Bx^{-\mu-\epsilon} \quad (5)$$

for very large values of  $x$ . (See the discussion of the subroutine XTRAP.)

5) Combination of Distributions. If one wishes to combine additive distributions characterized by different values of  $\underline{s}$  such as VELO and VELP, a value for  $\underline{s}$  for the combination must be obtained by taking proper account of the nature of  $\underline{s}$  as a mean value. Let us designate two currents with values  $\underline{s}_1$  and  $\underline{s}_2$  as  $C_1(\underline{s}_1, \cos\theta)$  and  $C_2(\underline{s}_2, \cos\theta)$ . Total currents are designated  $\underline{c}_1$ ,  $\underline{c}_2$ , respectively. Then

$$\begin{aligned} \langle \sec\theta \rangle &= \frac{\int_0^1 d(\cos\theta) \sec\theta [C_1(\underline{s}_1, \cos\theta) + C_2(\underline{s}_2, \cos\theta)]}{c_1 + c_2} \\ &= \frac{c_1 \langle \sec\theta_1 \rangle}{c_1 + c_2} + \frac{c_2 \langle \sec\theta_2 \rangle}{c_1 + c_2} \quad (6) \end{aligned}$$

Equation (1) can be used to express this equation in terms of  $s$  and  $s_i$ ,  $i=1,2$ .

6) Correction for Eccentric Duct Cross Sections. Data due to Woolf<sup>76</sup> show that for the reflected component (VELP) in a straight duct, a rough correction for eccentricity ( $W$  not equal to  $H$ ) can be made which is relatively independent of duct length over the range of our data. This correction is not highly accurate, but it only becomes large when the ratio of side widths becomes more extreme than 0.1. A simple fit to this correction factor which is independent of  $n$  has been included.

For the unreflected component (VELO), the correction is more complex. It turns out that duct length is more important for this case. The empirical formula which we have incorporated attempts to modify the simple form for the unreflected component to include the distance

parameter in about the simplest manner that can represent the data. The results are not as dependable as in the case of the reflected component; but the correction is smaller and again it does not become a major factor for duct eccentricities much less extreme than 0.1.

If  $\epsilon < 1$  is the ratio of the smaller to the larger of the two lateral dimensions, then our correction factors have the forms

$$K_p = \exp(-.109(\ln \epsilon)^2) \quad (7)$$

$$K_0 = \exp(-.06(\ln \epsilon)^2) L / \sqrt{(L^2 + \pi^{-1})} \quad (7')$$

The factor  $K_p$  multiplies the output list VELP, while  $K_0$  multiplies the output list VELO, both in DCTISO and in DCTSLT.

No correction of this type is presently applied to TURN.

7) Sudden Changes of Duct Size. If the duct cross sectional area should suddenly increase, the total current would remain unchanged and would rather quickly expand to fill the larger lateral space. By contrast, a sudden reduction in the cross sectional area of a duct, say by a factor DELR, would in first approximation decrease the total current by DELR but would not otherwise be expected to change spectrum or angular distribution.

8) Computations of Spectrum and Dose. At the conclusion of a sequence of duct elements the currents VELO and VELP are represented by angular parameters S0, SP, respectively and by values for all orders of reflection, and the problem then is to turn these data into spectra, dose reduction factors, and possibly fluences. Let us consider the problem of spectra first.

A basic requirement at this point is an evaluation of the neutron spectrum as a function of initial energy and order of reflection. To this end we have defined a special energy-loss problem in which neutrons in a succession of 23 energy groups from 15 MeV to thermal are injected into the space between plane concrete walls with directions

proportional initially to  $\cos\theta$ . If the initial energy group is identified with the index IE, and the final energy group by KE, the resulting function  $A(IE,KE,N)$  is to be applied for most of our purposes.\* The final spectrum of  $C(KE)$  of a current  $V(N)$  resulting from an input source with spectrum  $S(IE)$  would then be simply

$$C(KE) = \sum_{IE=KE}^{\infty} S(IE) \sum_{N=1}^{\infty} V(N) * A(IE,KE,N). \quad (8)$$

If the dose assigned to each spectral group is designated  $D(KE)$ , it may be obtained as usual with

$$\text{Dose} = \sum_{KE=1}^{\infty} D(KE) C(KE), \quad (9)$$

$$= \sum_{IE=1}^{\infty} S(IE) \sum_{N=1}^{\infty} V(N) \left[ \sum_{KE=1}^{\infty} (D(KE) A(IE,KE,N)) \right]. \quad (9')$$

In eq (9'), the brackets identify a two-index quantity which we designate WTSD. Rearrangement of (9') leads to an alternative form with an interchange between positions of D and S. The corresponding bracket term is then designated SWTS.

9) Flux and Flux Ratios. The entering current can readily be weighted by dose factors to obtain a quantity which we refer to as "current dose". Note that  $N = 1$  is the only component, and presumably has the the input spectrum. Similarly one can compute a "current dose" for other positions, as well as the ratios to that at the entrance.

The problem of the flux and flux reduction factors is more complex in that currents with different angular distributions must be combined.

---

\*Later considerations suggest that it may be desirable to supplement with a function for mono-obliquity incidence cases.

Our approach is the simple procedure of computing current dose for emerging SO and SP components which are then transformed by the simple rule

$$\begin{aligned} \text{Flux} &= \frac{\text{flux}}{\text{current}} * \text{current} = \langle \text{sec} \theta \rangle * \text{current} \\ &\approx \frac{2}{1 + s} * \text{current} . \end{aligned} \quad (10)$$

Appropriate combinations and ratios then readily follow which have an accuracy which we think is consistent with many applications.

## VII. Concluding Remarks and Acknowledgment

Additional information is given in the Appendices.

A listing of the computer program is presented in Appendix D. Appendix A describes the various subroutines and their purpose. Appendix B describes various types of data used and gives some information about data sources. Appendix C gives additional examples of input and output of the program.

Albedo data has not been printed here, partly because it nearly doubles the number of pages and partly because it is most practical in magnetic disk form for direct use with computers.

Despite its apparent promise and potential usefulness, we do not regard the DCTDOS program to be completed in the sense of some of the best-known Monte Carlo programs currently available. But we hope that the ideas incorporated will stimulate additional work of this type leading to improvements and possibly applications to similar problems in other technical fields such as reactor engineering.

In regard to ideas and data developed herein, the author appreciates the many contributions from, and interest expressed by Dr. Stanley Woolf, of ARCON Corp. in Wakefield, Massachusetts. The use of "we" and "our" throughout this manuscript expresses the author's feeling of partnership during the investigation. Dr. G. L. Simmons of SAIC in San Diego assisted similarly with important aspects of the work.

## REFERENCES AND BIBLIOGRAPHY

1. Albright, G.H., Beck, E.J., LeDoux, J.C., and Mitchell, R.A., Evaluation of Buried Conduits as Personnel Shelters, Report No. WT-1421, Operation Plumbbob, Project 3.2, Defense Atomic Support Agency (1960).
2. Auxier, J.A., Burson, Z.G., French, R.L., Haywood, F.F., Mooney, L.G., and Straker, E.A., Nuclear Weapons Free-Field Environment Recommended for Initial Radiation Shielding Calculations, Report No. ORNL-TM-3396, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1972).
3. Beck, C. (Editor), Test of French Underground Personnel Shelters, Report No. WT-1453, Operation Plumbbob, Project 30.6, Defense Atomic Support Agency (1962).
4. Beck, C. (Editor), Nuclear Weapons Effects Tests of Blast-Type Shelters, Report No. CEX-68.3, Civil Effects Test Operations, U.S. Atomic Energy Commission (1969).
5. Berger, M.J., and Morris, E.E., Dose Albedo and Transmission Coefficients for Cobalt-60 and Cesium-137 Gamma Rays Incident on Concrete Slabs, Report No. 9071, National Bureau of Standards (1966).
6. Bramblett, R.M., Ewing, R.I., and Bonner, T.W., Nuclear Instruments and Methods 9, 12 (1960).
7. Byoun, T.Y., Babel, P.J., and Danjani, A.T., Computer Program SCAP-BR for Gamma-Ray Streaming Through Multilegged Ducts, ANS Transactions 27, 777 (1977).
8. Cain, V.R., A Study of the Shielding Characteristics of Basic Concrete Structures at the Tower Shielding Facility, ORNL-3464, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1964).
9. Cain, V.R., Clifford, C.E., and Hallard, L.B., Measurements of Radiation Intensities in Vertical Concrete-Lined Holes and an Adjoining Tunnel at the Tower Shielding Facility, ORNL-3513, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1964).
10. Celnik, J., A Simple Albedo Method for Calculating Effects of Single-Legged Ducts, Nuclear Reactor Shielding, p. 688, Roussin, R.W., Abbott, L.S., Bartine, D.E., Editors (1977).
11. Chandrasekhar, S., Radiation Transfer, Dover Publications, Inc., New York, N.Y. (1960).



12. Chapman, J.M., Dose-Attenuation Variation with Incident Gamma-Ray Energy in Two Legged Concrete and Steel Ducts, Nuclear Reactor Shielding, Tech. Report Y-F008-08-05-201, Defense Atomic Support Agency (1966).
13. Chilton, A.B. and Huddleston, C.M., A Semiempirical Formula for Differential Dose Albedo for Gamma Rays on Concrete, Nucl. Sci. Eng. 27, 419-424 (1962); earlier published as a Defense Atomic Support Agency Report (1962).
14. Chilton, A.B., Shultis, J.K., and Faw, R.E., Principles of Radiation Shielding, Chapter 9, Prentice-Hall, Inc., Englewood Cliffs, NJ 07632 (1984).
15. Clifford, C.E., Gamma Shielding Provided by Ducts, Report No. 370. Defense Research Chemical Laboratories, Ottawa, Canada (1962).
16. Clifford, C.E.; Simons, G.G.; Faw, R.E.; and Shultis, J.K., Benchmark Gamma Ray Streaming Experiment, Report No. RRA-T7903 Radiation Research Associates, Fort Worth, Texas (1979).
17. Clifford, C.E.; Muckenthaler, F.J.; Maerker, R.E.; Straker, E.A. and Mynatt, F.R.; Neutron Transport Studies in Ducts and Simple Shields: Comparison of Monte Carlo and Discrete Ordinate Calculations with Experimental Results from the Tower Shielding Facility, Report No. ORNL-TM-1965, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1967).
18. Coleman, W.A., A Determination of Intermediate-Energy Neutron Albedo Data for Concrete Using Monte Carlo, Report No. ORNL-TM-1205, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1965).
19. Coleman, W.A. and Maerker, R.E., Differential Neutron Current Albedos for Concrete in the Incident Energy Range 0.5 eV to 200 keV. II. Results of Monte Carlo Calculations for an Incident Angle of 0 deg., Report No. ORNL-3967, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1966).
20. Coleman, W.A. and Maerker, R.E., Differential Neutron Current Albedos for Concrete in the Incident Energy Range 0.5 eV to 200 keV. III. Results of Monte Carlo Calculations for an Incident Angle of 45 deg., Report No. ORNL-3967, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1966).
21. Coleman, W.A. and Maerker, R.E., Differential Neutron Current Albedos for Concrete in the Incident Energy Range 0.5 eV to 200 keV. IV. Results of Monte Carlo Calculations for an Incident Angle of 60 deg., Report No. ORNL-3967, Oak Ridge National Laboratory, Oak Ridge Tennessee, (1966).



22. Coleman, W.A. and Maerker, R.E., Differential Neutron Current Albedos for Concrete in the Incident Energy Range 0.5 eV to 200 keV. V. Results of Monte Carlo Calculations for an Incident Angle of 75 deg., Report No. ORNL-3967, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1966).
23. Coleman, W.A. and Maerker, R.E., Differential Neutron Current Albedos for Concrete in the Incident Energy Range 0,5 eV to 200 keV. VI. Results of Monte Carlo Calculations for an Incident Angle of 85 deg., Report No. ORNL-3967, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1966).
24. Coleman, W.A., Maerker, R.E., Muckenthaler, F.J., and Stevens, P.N., Calculation of Doubly Differential Current Albedos for Mono-directional Beams of Epicadmium Neutrons Incident on Concrete and Comparison of the Reflected Subcadmium Component with Experimental Results, Nucl. Sci. Eng. 27, 411-422 (1967). Earlier published as Report No. ORNL-TM-1554, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1966).
25. Collins, D.G., A Monte Carlo Multibend Duct Procedure, NARF-62-13T Nuclear Aerospace Research Facility, Fort Worth, Texas (1962).
26. Collins, D.G., French, R.L., Mooney, L.G., Tompkins, K.W., and Wells, M.B., Simplified Methods for Calculating the Penetration of Neutron and Gamma Radiation into Underground Concrete Structures, Report No. RRA-M45, Radiation Research Associates, Inc., Fort Worth, Texas (1964).
27. Corps of Engineers, Huntsville Division (U.S. Army), 100 Man Blast Shelter, Design Analysis Volume 1, March 1983.
28. Doty, D.R., Dose Measurements for Neutron Streaming in Ducts, Technical Report No. R282, U.S. Naval Civil Engineering Laboratory, Port Hueneme, California (1964).
29. Doty, D.R., Differential Neutron Albedo for Finite Slabs, Nucl. Sci. Eng. 27, 478 (1967).
30. Eisenhauer, C.M., Scattering of Cobalt-60 Gamma Radiation in Air Ducts, Tech. Note 74, National Bureau of Standards (1960).
31. French, R.L., Wells, M.B., and Schaeffer, Penetration of Neutron and Gamma Radiation Through the Openings of Underground Structures, Report No. RRA-T41, Radiation Research Associates, Inc., Fort Worth, Texas (1964).
32. French, R.L., Wells, M.B., and Schaeffer, N.M., Neutron and Gamma Ray Penetration into Concrete-Shielded Underground Structures, Report No. RRA-T51, Radiation Research Associates, Inc., Fort Worth, Texas (1965).

33. Gardner, L.B., Neutron and Gamma-Ray Streaming through a Two-Legged, Thick-Wall Steel Duct, Technical Report No. R-558, Defense Atomic Support Agency (1968).
34. Gardner, L.B., Neutron and Gamma-Ray Streaming through Thin-Steel Shelter Entranceways, Technical Report No. R-569, Defense Atomic Support Agency (1968).
35. Gardner, L.B. and Mettler, A.J., Monte Carlo Calculations of Gamma-Ray Streaming through Coplanar and Noncoplanar Ducts, Report No. R412, Defense Atomic Support Agency (1966).
36. Gardner, L.B. and Mettler, A.J., Monte Carlo Calculation of Neutron Streaming through Two-Legged Duct Entranceways, Report No. R379, Defense Atomic Support Agency (1965).
37. Gardner, L.B.; Mettler, A.J.; and Burdick, B., Experimental Determination of Neutron Energy Spectra in Concrete Ducts, Report No. R523, Defense Atomic Support Agency (1967).
38. Hagan, W.K. and Simmons, G.L., Neutron Spectra from Monoenergetic Source Neutrons after Multiple-Reflection between Plane-Parallel Concrete Interfaces, Report No. SAI-013-79-605-LJ, Science Applications, Inc., La Jolla, California (1979).
39. Horton, C.C.; Halliday, D.B.; Lakey, J.R.A., and Harrison, J.R., The Transmission of Thermal Neutrons in a Straight Annular Duct, Report No. SWP/P.27, U.K. Atomic Energy Agency, Harwell (1956).
40. Huddleston, C.M. and Wilcoxson, W., Gamma Ray Streaming Through Ducts, Report No. R289, Defense Atomic Support Agency (1964).
41. Huddleston, C.M., Gamma Ray Streaming through Ducts, Symposium on Protective Structures for Civilian Populations, National Academy of Sciences, Washington, D.C. (1965).
42. Jaeger, R.G. Engineering Compendium on Radiation Shielding, Vol I. Shielding Fundamentals and Methods, Sections 4.4 and 5.4.1, and Chapter 8, Springer-Verlag, New York Inc. (1968).
43. Maerker, R.E. and Muckenthaler, F.J., Calculational and Measurement of the Fast-Neutron Differential Dose Albedo for Concrete, Nucl. Sci. Eng. 22, 455-462 (1965).
44. Maerker, R.E. and Muckenthaler, F.J., Differential Fast-Neutron Albedos for Concrete. I. Descriptions of Monte Carlo Calculation and TSF Experiment and Comparison of Calculated and Experimental Results, Report No. ORNL-3822, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1965).

45. Maerker, R.E. and Muckenthaler, F.J., Measurements and Single-Velocity Calculations of Differential Angular Thermal-Neutron Albedos for Concrete, Report No. ORNL-4090, Oak Ridge National Laboratory (1967). Also Report No. ORNL-TM-1421 (1966).
46. Maerker, R.E. and Muckenthaler, F.J., Monte Carlo Calculations, Using the Albedo Concept, of the Fast-Neutron Dose Rates along the Center Lines of One- and Two-Legged Square Concrete open Ducts and Comparison with Experiment, Nucl. Sci. Eng. 27, 423-432 (1967); earlier published as Oak Ridge National Laboratory, ORNL-TM-1557 (1966).
47. Maerker, R.E. and Muckenthaler, F.J., Calculations, Using the Albedo Concept, of Thermal-Neutron Fluxes along the Center Lines of One-, Two- and Three-Legged Square Concrete open Ducts arising from Thermal Neutrons Entering Duct Mouths: Comparison with Experiment, Report No. ORNL-TM-1758, Oak Ridge National Laboratory (1967).
48. Maerker, R.E. and Muckenthaler, F.J., Neutron Fluxes in Concrete Ducts Due to Incident Epicadmium Neutrons: Calculations and Experiments, Report No. ORNL-TM-1794, Oak Ridge National Laboratory (1967). Also published in Nuclear Science and Engineering 30, 340-354 (1967).
49. Maerker, R.E. and Muckenthaler, F.J., Calculations, Using the Albedo Concept, of Thermal-Neutron Fluxes, Epicadmium-Neutron Fluxes, and Fast-Neutron Dose Rates along the Center Lines of Square Concrete Ducts: Comparison with Experiment, Report No. ORNL-4147, Oak Ridge National Laboratory, (1967).
50. Maerker, R.E. and Cain, V.R., AMC: A Monte Carlo Code Utilizing the Albedo Approach for Calculating Neutron and Capture Gamma-Ray Distributions in Rectangular Concrete Ducts, Report No. ORNL-3964, Oak Ridge National Laboratory, (1967).
51. Maerker, R.E., Claiborne, H.C., and Clifford, C.E., Neutron Streaming in Rectangular Ducts, Oak Ridge National Laboratory, (1966). Later published as Subsection 8.2.5, Engineering Compendium on Radiation Shielding, Vol. 1, Springer-Verlag, New York Inc. (1968).
52. Moore, J.A., Eggen, J.B., and Leopard, F.O., Monte Carlo Procedure for Analysis of Radiation in Ducts (RAID), Report No. AFWL-TR-67-9, Air Force Weapons Laboratory, Fort Worth, Texas (1967).
53. Park, C.M., Agnihotri, C.B., and Silverman, J., Interim Report on Scattering of Gammas through Ducts, Report No. UMNE-2, Dept. of Chemical Engineering, University of Maryland, College Park, MD (1962).



54. Sayedahmed, F.M., Belov, S., Megahid, R., and Hamouda, I., Neutron Transmission through Ducts Perpendicular to Neutron Beam Axis in Water Shields, Report No. U.A.R.A.E.E./Rep.-86, Atomic Energy Post Office, United Arab Republic Atomic Energy Establishment, Cairo, U.A.R. (1969).
55. Schaeffer, N.M. (Editor), Reactor Shielding for Nuclear Engineers, Chapter 7, Available in hardback as TID-25951 from the National Technical Information Service, Springfield VA 22151 (1973).
56. Selph, W.E.; Newell, J.M.; and Bayer, C.E., Analysis of Neutron Streaming through an Underground Hardened Installation, Report No. RRA-T82, Radiation Research Associates, Fort Worth, Texas (1967).
57. Selph, W.E. and Claiborne, H.C., Methods for Calculating Effects of Ducts, Access Ways, and Holes in Radiation Shields, Report No. ORNL-RSIC-20, Oak Ridge National Laboratory (1968).
58. Selph, W.E., Neutron and Gamma-Ray Albedos Report No. ORNL-RSIC-21, Oak Ridge National Laboratory (1968).
59. Shin, K., Murakami, R., Taniuchi, H., Hyodo, T., and Oka, Y., Measurements of Neutron and Gamma-Ray Streaming in a Cavity-Duct System and an Analysis by an Albedo Monte Carlo Nucl. Sci. Eng. 81, 161-171 (1982).
60. Shore, F.J. and Schamberge, R.D., The Transmission of Neutrons through Ducts in Water, Report No. BNL 390 (T-74), Brookhaven National Laboratory, New York (1956).
61. Simmons, G.L., Capture Gamma Ray Currents Emerging from a Type Concrete Barrier, as Functions of Incident Neutron Energy and Direction, Private Communication 6/11/86 in fulfillment of a contract with NBS.
62. Song, Y.T., A Semiempirical Formula for Differential Dose Albedo for Neutrons on Concrete, Report No. TN-589, Naval Civil Engineering Laboratory, Port Hueneme, California (1964).
63. Song, Y.T., Fast-Neutron Streaming through Concrete Ducts; Theoretical Calculations Using the Albedo Concept Compared with Experimental Results, Technical Report No. R-565, Naval Civil Engineering Laboratory, Port Hueneme, California (1968).
64. Song, Y.T., Calculating the Dose Variation of Fast-Neutrons Streaming through an Iron Duct, Technical Report No. R-570, Naval Civil Engineering Laboratory (1968).
65. Song, Y.T.; Huddleston, C.M.; and Chilton, A.B., Neutron Streaming through Ducts, ANS. Trans. 38, 182 (1969).

66. Song, Y.T., Huddleston, C.M., and Chilton, A.B., Neutron Streaming through Ducts--Analysis of Experimental Nucl. Sci. Eng. 44 252-275 (1971).
67. Spencer, L.V., Diaz, J.A., and Moses, E., Neutron Penetration in Cylindrical Ducts, NBS Report 8542, National Bureau of Standards, (1964).
68. Spencer, L.V., and Diaz, J.A., Neutron Penetration in Ducts and Other Cavities, NBS Report 8541, National Bureau of Standards (1964).
69. Spencer, L.V., and Woolf, S., On Neutron Penetration in Duct Systems with Large Cross Section, ANS-SD-7, p. 199, Proceedings of Special Sessions on Gamma-Ray Production and Transport and on Civil Defense Shielding, American Nuclear Society, Chicago, Illinois (1967).
70. Terrell, C.W., Radiation Streaming in Shelter Entraceways, p. 266, Proceedings of the Shielding Symposium, Oct. 31-Nov. 1, U.S. Radiological Defense Laboratory, San Francisco, California (1960).
71. Terrell, C.W., Jerri, A.J., Lyday, R.O., and Sperber, D., Radiation Streaming in Shelter Entraceways, Report No. ARF 1158- 12, Armour Research Foundation, (1960).
72. Terrell, C.W. and Jerri, A.J., Radiation Streaming in Shelter Entraceways, Report No. ARF 1158A01-5, Armour Research Foundation (1961).
73. Terrell, C.W. and Jerri, A.J., Radiation Streaming in Ducts and Shelter Entraceways, Report No. ARF 1158A02-7, Armour Research Foundation (1962).
74. Vanzant, B.W. and DeHart R.C., Penetration of Nuclear into Shafts, Tunnels and Entraceways, Southwest Research Institute, San Antonio, Texas (1966).
75. Vogt, H.G., Intercomparison of Monte Carlo Calculations and Measurements of the Neutron and Gamma-Ray-Streaming in Multi Bent Nucl. Reactor Shielding, p. 857, Science Press, Princeton, N.J. (1977).
76. Woolf, S., private communication of unpublished computations to L.V. Spencer (May 20, 1968).
77. Woolf, S., private communication of unpublished computations to L.V. Spencer (Nov. 22, 1967).
78. Woolf, S. and Spencer, L. V., unpublished computations (1983).

## APPENDIX A. Some Notes on Subprograms

Figure 1 identifies the subprograms of DCTDOS, as well as indicating the flow of control. The list below may also be useful for quick reference. Following the list are brief descriptions.

- a) Control Subprogram:  
DCTDOS
- b) Data Subprograms:  
ENTDAT (storage)  
DATDCT (input data preparation and scaling)  
SRSDET (source and detector data preparation)  
SND (preparation of source and detector weight lists and arrays from basic neutron data)  
SGD (preparation of weight lists and arrays from basic gamma ray data)  
WTLISTS (storage of computed weights in a permanent file called PREWTS, to avoid recomputation with each new run)
- c) Duct Segment Computations:  
DCTISO (isotropic source entering a straight duct segment)  
DCTSLT (mono-obliquity source entering a straight duct segment)  
TURN (bend segment for different source angular parameters)  
BTODR (straight segment following a bend)  
ROOMX (duct input, detector pattern in a room)
- d) Dose Computations:  
DOSX (dose computations for ROOMX, i.e., in a room)  
DOSE (dose computations in, or at the end of a duct)
- e) Auxiliary Subprograms:  
RETRN (makes estimate of backreflection from end of a closed duct)  
XTRAP (a short program for extrapolation of data, mainly for straight duct segments)  
ORDST (prepares approximate orders-of-reflection distributions.)

a) Control Subprogram (DCTDOS). The pattern shown by figure 1 gives a fairly complete summary of the options presented by DCTDOS. For each type of case or problem there is a loop indexed by IC which evaluates a sequence of duct elements. At the conclusion of each problem or case there are options to enter a full set of new data (Run Case), new configuration data only (Problem Case), a change of a straight section (Length), or a new source and/or detector (Weights).

In the File #1, orders 82-85 set up a possible selection of reflected or unreflected component, although usually S=0. Order 94 applies LPIPE segment data to the final segment only. Order 118 gives a new application of weights previously applied in the same run. Order 119



applies new alternative weights near the end of a problem. Orders 124 and 148 develop weights only once for cases in which a segment length is varied. Order 121 develops gamma weights only once if there are at least two bends. The orders from 127 to 132 relate to the computation of dose for the segment preceding a room after a ROOMX dose pattern computation, for comparisons.

b) Data Subprograms.

ENTDAT stores output data for elementary duct segments (DUCTi and BEND arrays). It also stores selected spectral source and dose information in most of the remaining lists. The arrays AWT5 and AMOD represent obliquity incidence data (sidewall and endwall, respectively) for gamma ray sources. These are required by SRSDET to obtain realistic albedo weights for gamma ray problems.

DATDCT is mainly a buffer program in which input describing the configuration is properly scaled and inserted in COMMON storage. Input is summarized in Table 4, and output of DATDCT is summarized in Table A5 below. The ratio PEN, usually equated to zero, can be given a small value to take account of the mean wall depth from which reflection occurs. (We have made almost no use of this option).

The segment eccentricity ECC replaces the duct width and height, and COF is the length scaled by the square root of the cross sectional area. DELR mostly gives a cross sectional area ratio (ARAT) between successive segments. But this may not change, and then OTHR=0 so that DELR=1 (order 230). DELR is also used to provide a required parameter value for IEL = 6 and 8. DZRO also is given the value of a required parameter for IEL = 3 and 6, otherwise parameters of occasional application only.

Table A1. Scaled output produced by DATDCT and recorded with IPRINT = 2. For IEL = 6, ARAT repeats the value for the preceding segment.

IEL	COF	ECC	DELR	DZRO	SCL
	SO	PEN	ZDOOR	RMCase(IEL=8)	
1	L/√A	W/H	ARAT,1	ARAT	√RAT
2	L/√A	W/H	ARAT,1	ARAT	√RAT
3	L/√A	W/H	1	F	√RAT
4,5	L/√A	W/H	ARAT,1	ARAT	√RAT
6	W	H	L	WD	AEX
7	RDLNTH/√A	W/H	ARAT,1	RDLNTH-.01	ARTO
8	WDP	NW-1	NL-1	ARAT	ARTO

All the quantities named in Table A1 have the same interpretation as in Table 4, for the same value for IEL. Note that COF contains mainly scaled lengths, ECC has eccentricity ratios, DELR has reduction factors for constriction.

SCL is a list (in COMMON storage) of scale factors. Little use has been made of  $\sqrt{\text{RAT}}$ , which is ordinarily unity. ARTO is the ratio of entrance area to exit area; if there has been a change, either single or progressive, this scale-factor can indicate the magnitude of the total effect. But the effect is computed segment by segment using ARAT values.

A constriction confined to a segment entrance only is not really accounted for in DCTDOS, although it would not require much change. Note that an increase of segment cross section area will spread the exit flux over a larger area, while a reduction will reduce the total entering flux without any other effect on flux/(exit area).

SRSDET computes source and detector weights. It also makes use of three other programs: WTLISTS provides for more permanent storage of the weights, and SND and SGD apply multiple albedo data to the computation of orders-of-reflection data for neutrons and gamma rays.

Orders 274-278 provide a selection parameter for neutrons, capture gammas, or neutrons plus capture gammas.

Orders from 280 through 327 are directed to computation of orders-of-reflection weights for the gamma-ray case, which depends strongly on the configuration of the first leg and first bend. The estimate here is based on use of simple formulae for angular limits for a source at the centerline point of the entrance to a cylindrical duct (unreflected in the first segment, the CUT limit) and the complementary CUTP angular range which applies mainly to scattered components from the end-surface of first leg. Note again that the unreflected from the first segment (TV0) may be reflected following near-perpendicular incidence from an endwall while the reflected component (TVP) exits the first segment after reflection(s) somewhere in the first leg, here assumed cylindrical for convenience.

The main task for SRSDET is the computation which takes account of spectral changes with multiple reflections, and the effect which this has on the multiple-reflection weights. This requires SND and SGD and their processing of the large tables of multiple-reflection (albedo)

energy data. Hence we give below a more detailed analysis of SND and SGD, which are the heart of SRSDET.

SND Purpose: to generate the arrays WTSD, SWTS, and the list  
ENGAM, for neutrons and capture gammas.

Arguments, Arrays, and Lists:

SGRUPS: Input source list, neutrons.

SWTS: This begins as the source list which, multiplied by a multi-reflection albedo energy array gives dose weight per detector response group-strength.

One should note that WTSD and SWTS are complementary quantities. After constructing each with the orders-of-reflection data, one must still multiply the SWTS by the dose weights in DGRUPS, or correspondingly the WTSD by the spectral weights in SGRUPS, to obtain the dose weights required (see Eqs.9,9').

DGRUPS: Dose weights for the different spectral groups.

ENGAM: This is a list of dose weights for capture gamma rays generated in the duct walls. It is assumed that the ratio of gamma number-current albedo to neutron number-current albedo is .095/.75; and it is assumed that all the gammas are about 1 MeV, with dose weight  $4.5E-10$ . The indicated proportion of gammas/neutrons is assumed to hold for each incident order of reflection, with all orders of reflection contributing, independently of spectral energy. (These assumptions probably are overly simplistic, are untested, and should be reexamined.)

ALBEDO: This is the list of emergent neutron currents for each order of reflection (2nd index) for each energy group (first index), for each initial energy (MEG value). These basic data are included into the program by the READ orders.

Sequence of Computations:

405-416: Preparatory values assigned.

417-421: Begin main loop, first zeroing ALBEDO, which may not always be necessary.

422-426: Read the data and assign the unreflected components.

427-432: Cumulate the contributions to both main arrays, plus ENGAM, for all source energies.

SGD Purpose: for gammas, to generate the arrays WTSD and SWTS.

Arguments, Arrays, and Lists:

All are as described in SND except that ALB replaces ALBEDO of SND, and that VEL1 differs from ENGAM. (One must interpret SGRUPS, DGRUPS, and DWTS, as well as ALB, as applying to gamma rays in this subroutine).

VEL1: This is a weight list for different obliquity angles with which the initiating gamma rays may strike the wall. It is used because the first order of reflection for gamma rays will be significantly different, and will affect all other orders of reflection; hence some account must be taken whether most gamma rays first strike normally, grazingly, or with a cosine distribution.

Sequence of Computations: Closely similar to SND.

451-455: Zero the arrays.

456-462: Begin the source energy loop. KE is the energy index and LOB is the obliquity index. Note that the energies proceed from top down, each represented by the five obliquity cases.

463-467: Read the (mass storage) data and take care of the unreflected component, assumed to be unity and to be incident in proportion to VEL1 obliquity weights.

468-476: Cumulate the two main arrays over all source energies and obliquities.

WTLISTS is called by DCTDOS at the beginning of a run to set up two files, IXFILE and WTFILE. The latter contains precomputed weights for different combinations of IDOSE, ISORS, and LPIPE, currently a maximum of 65 sets. The former contains an index which states whether the set is, in fact, stored for retrieval or should be computed. A second index confirms the location in the file, and a third index gives the number of weights. At the end of DCTDOS a final call for WTFILE stores the file again in PREWTS including any new weight lists that may have been added.

This subroutine is called twice in SRSDET. The first call either accesses precomputed weights or exits the routine because they have not as yet been stored. The second call either exits the routine because they already are included or files a newly prepared set of weights in PREWTS.



c) Duct Segment Computations. These constitute the main working set for solutions. The three basic programs, DCTISO, DCTSLT, and TURN follow essentially the same pattern, namely

- 1) interpolate required data from tables in ENTDAT,
- 2) construct one or more suitable backscattering matrices for the segment (see Eq. 2),
- 3) fold data for the new segment over the input to obtain the output as in Eq. 2, and
- 4) modify, total (to VEL) and possibly print the results.

The program ROOMX is much more complex, because it does not simply focus on transmission of a current, but rather attempts to construct a dose pattern. It treats separately the component entering the room which does not reflect before contributing to the dose. (Let us call this the unreflected component even though it may consist of many earlier reflections.) Note that we have a mean angular parameter for each to assist our description of these two components.

Also given special treatment is the once-reflected component, which is sensitive to the angular and energy properties of the entering current. All other orders of reflection are lumped together and treated with the assumption that these contributions follow a cosine emission law from the room surfaces. The number albedo (.75) is assumed to be modified (reduced) by an escape probability which is taken to be proportional to the ratio of entrance area to half the total wall area.

We believe that these principles are reasonably sound. Tests comparing results with simple neutron Monte Carlo data have been promising. Corresponding tests for gamma ray data have not been carried out; and patterns for gamma ray cases should be less accurate. Other ways to estimate these detector patterns can also be devised and could prove better.

BTODR computes penetration down straight segments which follow bends, and represents the key to successful evaluation of multi-segment ducts. It is a second-generation program.

Consider a bend from the vantage point of a position well-down the following straight section. Mainly visible is the endwall. While albedo from the endwall does not have a strictly isotropic flux upon emergence, the flux does not greatly differ from an isotropic distribution for penetration purposes down a straight section to follow. We approximate this contribution with our DCTISO subprogram. All three other walls of the bend, including the endwall of the first segment, decrease in visibility by a cosine factor with distance down the next segment. We approximate their contribution with our DCTSLT program with the aid of our estimate of the angular parameters  $S_0$ ,  $SP$  of exit from the bend (entrance into the new segment) for components unscattered ( $S_0$ ) or scattered in the bend. It is mainly the  $SP$  from the bend which must be apportioned to the walls which become the new endwall (for DCTISO) and the new sidewalls. The  $S_0$  applies only to one (DCTSLT) component which happens to bypass the bend but usually strikes a wall near the new segment entrance.

The remaining complexities relate to estimates of the number of neutrons (or gammas) which strike the different walls, both reflected in the bend and unreflected, to apportion suitably their relative contributions and mean angular incident at the new segment entrance. Rough symmetry between the exit flux and that striking the endwall-to-be aids in making this estimate.

d) Dose Computation Subprograms. DOSX is complicated by the requirement to evaluate the dose for the pattern used in ROOMX. Otherwise it applies the weight lists from SRSDET to the orders of reflection distributions at the different specified positions.

Both DOSX and DOSE have incorporated several options, but at present the only option used computes an estimated flux dose rather than current dose (using Eq. 10). Both also yield a reduction factor relative to the initial position at the duct entrance.

e) Auxiliary Subprograms. RETRN is of limited application, since it affects the final result only in the general vicinity of a straight segment endwall. It is not a simple program, since it must evaluate what strikes the endwall and subsequent back-attenuation. Possibility



of reduction due to a missing part of the duct sidewalls is ignored.

XTRAP is a short subprogram which has taken different forms with little change in results once the general mathematical form for reduction, i.e., a power law down straight duct segments, is fixed. Present formula is something of a hodgepodge which works after a fashion,

$$Y = \left( AL1 + \frac{AL2}{X^{0.1}} \right) * \frac{1}{X^{XI}} \quad (A1)$$

If the parameters of the fit give  $AL1 < 0$ , a simpler form is used instead, namely

$$Y = (AL1/X^{XI}) \quad (A1')$$

In the usual case, the constant  $XI$  is pre-assigned a value on the basis of  $y \propto X^{-XI-.05}$  for large  $X$ , and  $AL2$ ,  $AL1$  are then computed. Various methods for specifying the constant  $XI$  have been applied. This present approach is based on the assumption that a single term is sufficient for the large- $X$  trend. The constants  $AL1$  and  $L2$  are then determined to fit the distribution well for the largest tabulated values for  $X$ , and give reasonable extrapolated values at distances beyond the tabulation which are not excessively large. The limiting case is most commonly  $Y \rightarrow 0$  for  $X \rightarrow \infty$ . But a finite limit for  $Y$  is not ruled out and can be implemented.

ORDST stands for order-of-reflection distribution. Its purpose is to change a simple power-law distribution fixed by two parameters into a modified distribution which is more flexible and realistic, but which can be readily computed from the same two parameters. More precisely, a power law can be expressed by a single constant times

$$a, a^2, a^3, \dots, a^n, \dots, \quad (A2)$$

The more flexible and applicable form which is developed in this subprogram is an overall factor times

$$\alpha, \alpha^2 + \alpha b, \alpha^3 + \alpha^2 b + \alpha b^2, \dots \quad (A2')$$

Assignment of a value for the parameter  $b$  suggested by Woolf data for elementary segments<sup>76</sup> leads to formulae which give  $\alpha$  and the multiplicative factor in terms of  $a$  and the original strength constant. Such a change is rendered useful because orders-of-reflection distributions in ducts are not simple power laws, and much better descriptions can be obtained with little more added flexibility than this can give.

#### APPENDIX B. Sources of Data

a) S. Woolf Duct Data. The unpublished data of ref. 76 constituted a large number of configurations and sources, as listed in Table B1 below. Two types of directional sources were computed, mono-obliquity sources and continuous sources. The latter were sampled from a normalized distribution of the form

$$f(s, \cos\theta) = 2(1-s)\cos\theta \cdot d(\cos\theta) / [(1+s)^2 - 4s\cos\theta]^{3/2} . \quad (B1)$$

The value  $s=0$  corresponds to the case of incident isotropic flux.

Output orders-of-reflection data were given for the zero'th through the 20'th order of reflection. Albedo data were given similarly. To properly include higher orders the Woolf output also gave values of the Fourier Transform,

$$O(p) = \sum_{n=1}^{\infty} O(n)e^{-np} . \quad (B2)$$

These data turned out to be extremely useful in obtaining simple analytic approximations to the output distributions-in- $n$ .

Only a small portion of the Woolf data were directly used to obtain the ENTDAT data for DUCT0, DUCT1, DUCT2, and BEND, from which the first three parameters of Table 4 interpolate. And as noted in Section VI, corrections were later necessary for an albedo of 0.75 instead of 0.696.

It was early very clear that angular parameters would be necessary in the joining of different segments. But it was not so clear which or how many to use. To attempt perhaps the simplest option first, additional computations were made to provide  $S_0$  and  $SP$ , the (angular average) parameter values listed in the DUCT<sub>i</sub> and BEND arrays, for many of the most important cases.

The Table B1 list of cases were mostly recomputed<sup>77</sup> in order to provide data for  $\langle \sec\theta \rangle$ , and thus for  $S_0$ ,  $SP$ .

Computations were also made for eccentric ducts with rectangular cross sections. The eccentricity ratio  $\epsilon$  is used which equals width/length. Endpoints plus 5 internal planes of a duct of length 6 diameters were computed. Eccentricity ratios  $n$  were .01, .25, .50, .75, and 1.0. Each computation was performed for at least 6  $s$  values. Fortunately the sensitivity to this parameter turned out to be rather low, except for extreme cases. This confirms also that square cross section data has fairly general applicability.

Table B1. Most of the cases listed below were computed with 5,000 to 10,000 histories.

- FINITE, STRAIGHT, OPEN DUCTS, CONTINUOUS ANGULAR INPUT  
8 lengths (0 to 8 diameters), each with 9 input  $s$  values
- FINITE, STRAIGHT, OPEN DUCTS, MONODIRECTIONAL ANGULAR INPUT  
8 lengths (0 to 6 diameters), each with input obliquities  
 $\cos\theta_{in} = .01, .25, .50, \text{ and } .75$ . (These were redone to 8 diameters  
with 30,000 histories.)
- SEMI-INFINITE, STRAIGHT DUCTS, CONTINUOUS ANGULAR INPUT  
13 depths (0 to 6 diameters), each with 6  $s$  values, computation  
of both flux and current.
- OPEN, STRAIGHT, FINITE DUCTS INTERNALLY, CONTINUOUS ANGULAR INPUT  
Midpoints, 1, 2, and 4 diameter lengths, each for 6  $s$  values.  
0.5 diameters depth, 4.5 diameters length, each for 6  $s$  values.  
1.0 " " , 7.0 " " " " " " " " .
- CLOSED, STRAIGHT, FINITE DUCTS, CONTINUOUS ANGULAR INPUT  
4, 6 diameters length, each for 6  $s$  values
- OPEN BEND SEGMENTS, SQUARE CROSS SECTIONS  
1x1, 2x2, 3x2, 2x1.5, 1.5x2, 7  $s$  values (continuous angular input)  
" " " " " " , input perpendicular to entrance.

b) SAI Neutron Multiple-Reflection Data.<sup>38</sup> These tabulated data are currently in floppy-disk files. They correspond to various monoenergetic neutron fluxes which are backscattered many times between two plane-parallel concrete barriers. At their start, the angular distribution is isotropic. With each backscattering, the emergent neutrons are classified into energy groups at and lower than the source energy, in a fairly standard set of 23 energy groups. The concrete composition is that referred to as Type 04, and has density 2.34 g/cm<sup>3</sup> and composition by weight as shown in Table D3 below:

Table B2. Type 04 concrete composition by fractional weight.

ELEMENT	FRACTIONAL WEIGHT	ELEMENT	FRACTIONAL WEIGHT
H	.0056	Si	.3151
O	.4979	S	.0013
Na	.0171	Ca	.0829
Mg	.0026	Fe	.0124
Al	.0457		

The multiple-albedo neutron data are called and used by SND, as previously noted.

c) Gamma Albedo Data.<sup>78</sup> These data were the basis for the multiple-albedo data required for SGD. Original computations by Woolf provided detailed albedo data for single incidence only. A program was then written to apply these data to the space between plane, semi-infinite concrete barriers. Source gamma rays of 17 different energy groups were incident at each of the following obliquity cosines (relative to the perpendicular),

$$\cos\theta_0 = .05, .25, .50, .75, 1.0 \quad .$$

Albedo energy groups and emergent obliquity cosines were then applied a total of 10 times to evaluate the multi-reflection data in ALB.

The emergent obliquity cosines were eventually not applied to the gamma ray cases, with reliance instead on a similarity between low energy gamma and neutron albedo angular distributions.

Our approach to computations for the different duct segment data types requires the application of approximate albedo angular distributions in specialized Monte Carlo studies, plus evaluation of emergent current-dose angular distributions characterized by the S0 and SP parameters. These permit ready estimates of fluxes from currents by the simple rule previously given of

$$\begin{aligned} \text{Flux} &= \frac{\text{flux}}{\text{current}} * \text{current} \\ &\approx \frac{2}{1 + S} * \text{current} \quad . \end{aligned} \quad (10)$$

Thus the emergent flux and flux/current ratio are the main angular data recorded, and are used to link the various duct components together.

#### APPENDIX C. Remarks on Inputs and Outputs with Examples

Figure C1 gives both input and output data for a run with IPRINT=3. To save space let us note that if we had used IPRINT=2 the first page would have been identical, but would have been followed by the last five lines of the 2nd page of these output data to conclude the run.

The second page gives source listing and spectrum at the detector, as well as a breakdown of the DOSE value into contributions from different orders of reflection, termed DFLXN. VELO and VELP differ in that the former is unreflected in the final segment while the latter is reflected in the final segment: both usually have multiple orders of reflection in earlier segments. (Note that the first column gives both an energy group list and a list of orders of reflection (N=n+1).

The four VEL lists correspond to input and/or output for successive duct segments. S0 and SP values below give the angular distribution parameters S for the VELO and VELP components, respectively, which are, however, recorded here only for the last VEL list. Note that for forward currents,  $1 < S < -1$ , with the value +1 corresponding to a current focused along the axis, and with the value -1 corresponding to complete focus sideways, i.e., 90° from the forward direction. The value .000 here in the IC=1 column has no meaning, being a default only.



INPUT

3	2	0	3	3	1.000+000
3	5	3	4	2	3.000+000

0

\*\*\* M-M, 2-LEG, EPITH/TH ORDERS OF REFL., 9/2/86 \*\*\*

0	0.414	0.000	0.000	0.000	0.000
2	3.000	3.000	13.500	0.000	0.000
3	3.000	3.000	0.000	0.000	0.000
5	3.000	3.000	3.500	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000

OUTPUT

\$=====

ISORS	IDOSE	ING	LPIPE	IEL	DIMEN
3	2	0	3	3	1.000E+00
IPRBLM	NEL	IPRINT	JVARY	JDEL	DEL
3	5	3	4	2	3.000E+00

0

\*\*\* M-M, 2-LEG, EPITH/TH ORDERS OF REFL., 9/2/86 \*\*\*

IEL	WIDTH-FT	HEIGHT-FT	LENGTH-FT	OTHER	SELECT
0	.414	.000	.000	.000	.000
2	3.000	3.000	13.500	.000	.000
3	3.000	3.000	.000	.000	.000
5	3.000	3.000	3.500	.000	.000
7	.000	.000	.000	.000	.000
IEL	COF	ECC	DELR	DZRO	SCL
0	.41400	.00000	.00000	.00050	.00000
2	4.50000	1.00000	1.00000	.00000	1.00000
3	.00000	1.00000	1.00000	.00000	1.00000
5	1.16667	1.00000	1.00000	.00000	1.00000
7	.00000	1.00000	1.00000	-.01000	1.00000

NEXT CASE 1.167

DWTS//SWTD//ENGAM

4.124E+00	2.965E+00	2.252E+00	1.739E+00	1.320E+00	9.968E-01	7.457E-01
5.576E-01	4.178E-01	3.131E-01	2.334E-01	1.747E-01	1.302E-01	9.715E-02
7.214E-02	5.317E-02	3.968E-02	2.941E-02	2.207E-02	1.642E-02	1.214E-02

4.124E+00	2.965E+00	2.252E+00	1.739E+00	1.320E+00	9.968E-01	7.457E-01
5.576E-01	4.178E-01	3.131E-01	2.334E-01	1.747E-01	1.302E-01	9.715E-02
7.214E-02	5.317E-02	3.968E-02	2.941E-02	2.207E-02	1.642E-02	1.214E-02

.000E+00	1.690E-10	1.283E-10	9.914E-11	7.523E-11	5.682E-11	4.250E-11
3.178E-11	2.381E-11	1.785E-11	1.331E-11	9.956E-12	7.421E-12	5.537E-12
4.112E-12	3.031E-12	2.262E-12	1.677E-12	1.258E-12	9.360E-13	6.919E-13

NEXT(I)= 0 DOSE= 5.833E+00 REDUX FACTR= 1.000E+00  
 NEXT(I)= 2 DOSE= 8.977E-02 REDUX FACTR= 1.539E-02  
 NEXT(I)= 3 DOSE= 4.889E-02 REDUX FACTR= 8.381E-03  
 NEXT(I)= 5 DOSE= 2.347E-02 REDUX FACTR= 4.024E-03

Fig. C1. Example of data from IPRINT=3.



Fig. C1 (Continued)

ARTO=	1.000E+00				
MEG,N	SOURCE	SPCTRM	DFLXN	VELO	VELP
1	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
3	2.150E-02	2.010E-08	2.365E-04	5.648E-05	5.605E-05
4	8.000E-02	4.532E-07	4.912E-04	8.760E-05	1.226E-04
5	1.405E-01	1.538E-06	6.543E-04	9.016E-05	1.640E-04
6	1.341E-01	4.393E-06	7.069E-04	7.779E-05	1.730E-04
7	2.673E-01	1.074E-05	6.775E-04	6.113E-05	1.594E-04
8	2.646E-01	1.986E-05	6.023E-04	4.550E-05	1.349E-04
9	9.440E-02	4.538E-06	5.095E-04	3.277E-05	1.079E-04
10	3.038E-01	1.757E-05	4.166E-04	2.312E-05	8.306E-05
11	4.430E-01	6.668E-05	3.327E-04	1.610E-05	6.227E-05
12	3.922E-01	2.173E-04	2.613E-04	1.111E-05	4.581E-05
13	4.468E-01	2.341E-04	2.028E-04	7.630E-06	3.325E-05
14	2.464E-01	1.894E-04	1.561E-04	5.220E-06	2.389E-05
15	2.350E-01	4.362E-04	1.193E-04	3.563E-06	1.705E-05
16	2.107E-01	4.200E-04	9.086E-05	2.429E-06	1.211E-05
17	2.099E-01	4.961E-04	6.897E-05	1.654E-06	8.569E-06
18	1.492E-01	3.844E-04	5.223E-05	1.125E-06	6.048E-06
19	1.192E-01	3.246E-04	3.950E-05	7.655E-07	4.261E-06
20	1.497E-01	4.463E-04	2.984E-05	5.206E-07	2.999E-06
21	1.202E-01	3.684E-04	1.609E-05	1.346E-07	1.586E-06
22	9.570E-02	3.763E-04			
23	.000E+00	1.946E-02			

VEL(N)/IC=1	2	3	4	5
1	1.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	1.927E-03	4.540E-04	.000E+00
3	.000E+00	2.163E-03	7.213E-04	1.125E-04
4	.000E+00	1.849E-03	7.729E-04	2.102E-04
5	.000E+00	1.424E-03	6.869E-04	2.541E-04
6	.000E+00	1.042E-03	5.506E-04	2.508E-04
7	.000E+00	7.414E-04	4.152E-04	2.205E-04
8	.000E+00	5.185E-04	3.016E-04	1.803E-04
9	.000E+00	3.589E-04	2.139E-04	1.406E-04
10	.000E+00	2.468E-04	1.495E-04	1.062E-04
11	.000E+00	1.690E-04	1.035E-04	7.837E-05
12	.000E+00	1.154E-04	7.114E-05	5.692E-05
13	.000E+00	7.870E-05	4.873E-05	4.088E-05
14	.000E+00	5.360E-05	3.328E-05	2.911E-05
15	.000E+00	3.648E-05	2.269E-05	2.062E-05
16	.000E+00	2.482E-05	1.546E-05	1.454E-05
17	.000E+00	1.688E-05	1.052E-05	1.022E-05
18	.000E+00	1.148E-05	7.157E-06	7.173E-06
19	.000E+00	7.802E-06	4.868E-06	5.027E-06
20	.000E+00	5.304E-06	3.310E-06	3.519E-06
21	.000E+00	3.606E-06	2.250E-06	1.721E-06
SO=	4.140E-01	4.547E-01	2.680E-01	3.602E-01
SP=	.000E+00	3.532E-01	.000E+00	-7.022E-02

NEXT CASE 4.167

NEXT(I)= 0	DOSE= 5.833E+00	REDUX	FACTR= 1.000E+00
NEXT(I)= 2	DOSE= 8.977E-02	REDUX	FACTR= 1.539E-02
NEXT(I)= 3	DOSE= 4.889E-02	REDUX	FACTR= 8.381E-03
NEXT(I)= 5	DOSE= 1.410E-03	REDUX	FACTR= 2.418E-04

Fig. C1 (Continued)

ARTO= 1.000E+00

MEG,N	SOURCE	SPCTRM	DFLXN	VELO	VELP
1	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
3	2.150E-02	1.234E-09	1.472E-05	9.337E-06	1.242E-06
4	8.000E-02	2.641E-08	2.598E-05	1.277E-05	3.863E-06
5	1.405E-01	9.033E-08	3.287E-05	1.236E-05	6.445E-06
6	1.341E-01	2.550E-07	3.557E-05	1.031E-05	7.994E-06
7	2.673E-01	6.177E-07	3.511E-05	7.945E-06	8.410E-06
8	2.646E-01	1.131E-06	3.268E-05	5.842E-06	8.003E-06
9	9.440E-02	2.576E-07	2.924E-05	4.176E-06	7.136E-06
10	3.038E-01	9.958E-07	2.547E-05	2.931E-06	6.088E-06
11	4.430E-01	3.728E-06	2.175E-05	2.035E-06	5.036E-06
12	3.922E-01	1.187E-05	1.832E-05	1.402E-06	4.073E-06
13	4.468E-01	1.267E-05	1.528E-05	9.613E-07	3.241E-06
14	2.464E-01	1.022E-05	1.264E-05	6.571E-07	2.546E-06
15	2.350E-01	2.334E-05	1.039E-05	4.483E-07	1.981E-06
16	2.107E-01	2.240E-05	8.498E-06	3.055E-07	1.529E-06
17	2.099E-01	2.645E-05	6.924E-06	2.079E-07	1.173E-06
18	1.492E-01	2.049E-05	5.624E-06	1.415E-07	8.950E-07
19	1.192E-01	1.731E-05	4.555E-06	9.623E-08	6.801E-07
20	1.497E-01	2.378E-05	3.681E-06	6.544E-08	5.149E-07
21	1.202E-01	1.967E-05	2.116E-06	3.207E-08	2.768E-07
22	9.570E-02	2.012E-05			
23	.000E+00	1.195E-03			

VEL(N)/IC=1	2	3	4	5
1	1.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	1.927E-03	4.540E-04	.000E+00
3	.000E+00	2.163E-03	7.213E-04	1.058E-05
4	.000E+00	1.849E-03	7.729E-04	1.664E-05
5	.000E+00	1.424E-03	6.869E-04	1.881E-05
6	.000E+00	1.042E-03	5.506E-04	1.831E-05
7	.000E+00	7.414E-04	4.152E-04	1.636E-05
8	.000E+00	5.185E-04	3.016E-04	1.385E-05
9	.000E+00	3.589E-04	2.139E-04	1.131E-05
10	.000E+00	2.468E-04	1.495E-04	9.019E-06
11	.000E+00	1.690E-04	1.035E-04	7.070E-06
12	.000E+00	1.154E-04	7.114E-05	5.475E-06
13	.000E+00	7.870E-05	4.873E-05	4.202E-06
14	.000E+00	5.360E-05	3.328E-05	3.203E-06
15	.000E+00	3.648E-05	2.269E-05	2.429E-06
16	.000E+00	2.482E-05	1.546E-05	1.834E-06
17	.000E+00	1.688E-05	1.052E-05	1.381E-06
18	.000E+00	1.148E-05	7.157E-06	1.037E-06
19	.000E+00	7.802E-06	4.868E-06	7.763E-07
20	.000E+00	5.304E-06	3.310E-06	5.804E-07
21	.000E+00	3.606E-06	2.250E-06	3.088E-07
SO=	4.140E-01	4.547E-01	2.680E-01	7.438E-01
SP=	.000E+00	3.532E-01	.000E+00	2.636E-01

The second case (4.167) follows the same printout pattern.

Figure C2 reproduces the input for all the ORNL neutron duct penetration sources of Figs. 6-9. Only a single run was required because different sources and detectors were cycled at each detector point. On an IBM XT, with the 8087 chip and storage of the ALBEDO, ALB, PREWTS files in the RAM memory, this run required about 5 minutes. It should be noted that the (unit) thermal neutron source required additional normalizing of the data by a factor of very nearly 2 (neutrons/cm<sup>2</sup>) to obtain the experimental source strength. With use of IPRINT = 1, the many cases require about 9 pages of output.

Figure C3 shows the input for the ARF C060 gamma ray experiments. Only a single detector response has been included. Note that in this case the first input data card suffices for 1-leg, 2-leg, and 3-leg cases. Time required was a little over 4 minutes. In every computation about 5/3 minutes is required at the start to set up the PREWTS data file into the arrays consulted in SRSDET. Actual computation time of problems can be better estimated by subtracting this setup interval.

The references include many other interesting cases, including design data such as are in Refs. 3 and 4, additional experimental data from many of the references listed, Monte Carlo studies, and the various semi-analytic and semi-empirical studies. Note that concrete-walled cavities are presently required, or in the case of gamma rays moderately low-Z walls. The program is also written only for near-right-angle bends. Because of the similarity of the <sup>137</sup>Cs ARF results to the <sup>60</sup>Co results we have not included Cesium source computations in this report.

Preliminary applications have also been made to some duct configurations for possible Civil Defense structures. These have not been included in this report because comparison data is lacking.

```

INPUT
  2  2  0  3  1 1.000+000
  2  3  1  2 11 1.333+000
  3
  2  3  4  0  3 1.000+000
  2  3  3  0  3 3.600+005
  2  3  3  1  3 3.600+005
***MAERCKER-MUCKEN, 1-LEG, THERMAL NEUTS, 8/31/86***
  0  0.414  0.000  0.000  0.000  0.000
  2  3.000  3.000  0.000  0.000  0.000
  7  0.000  0.000  0.000  0.000  0.000
  2  2  0  3  1 1.000+000
  2  5  1  4  8 1.333+000
  3
  2  3  4  0  3 1.000+000
  2  3  3  0  3 3.600+005
  2  3  3  1  3 3.600+005
***M+M, 2-LEG, THERMAL FLUX, 8-15-83 ***
  0  0.414  0.000  0.000  0.000  0.000
  2  3.000  3.000 13.500  0.000  0.000
  3  3.000  3.000  0.000  0.000  0.000
  5  3.000  3.000  0.500  0.000  0.000
  7  0.000  0.000  0.000  0.000  0.000
  2  2  0  3  1 1.000+000
  3  7  1  6  6 1.333+000
  3
  3  3  4  0  3 1.000+000
  3  3  3  0  3 3.600+005
  3  3  3  1  3 3.600+005
***MAERCKER-MUCKEN, 3-LEG, THERMAL NEUTS, 7/26/83 ***
  0  0.414  0.000  0.000  0.000  0.000
  2  3.000  3.000 13.500  0.000  0.000
  3  3.000  3.000  0.000  0.000  0.000
  5  3.000  3.000 12.000  0.000  0.000
  3  3.000  3.000  0.000  0.000  0.000
  5  3.000  3.000  0.500  0.000  0.000
  7  0.000  0.000  0.000  0.000  0.000

```

Fig. C2. Input Data for computations of the ORNL duct.

```

INPUT
 5      5      0      3      1 3.319+009
 1      3      1      2     12 3.000-001
 0
*** ARF 1-LEG, 50 CURIE CO-60, RAD DOSE, 9/1/86 ***
 0      0.000      0.000      0.000      0.000      0.000
 1      6.000      6.000      1.800      0.000      0.000
 7      0.000      0.000      0.000      0.000      0.000
-1      5      1      4     10 3.000-001
 0
*** ARF 2-LEG, 50 CURIE CO-60, RAD DOSE, 9/1/86 ***
 0      0.000      0.000      0.000      0.000      0.000
 1      6.000      6.000     10.000      0.000      0.000
 3      6.000      6.000      0.000      0.000      0.000
 5      6.000      6.000      1.000      0.000      0.000
 7      0.000      0.000      0.000      0.000      0.000
-3      7      1      6      6 3.000-001
 0
*** ARF 3-LEG, 50 CURIE CO-60, RAD DOSE, 9/1/86 ***
 0      0.000      0.000      0.000      0.000      0.000
 1      6.000      6.000     10.000      0.000      0.000
 3      6.000      6.000      0.000      0.000      0.000
 5      6.000      6.000      8.000      0.000      0.000
 3      6.000      6.000      0.000      0.000      0.000
 5      6.000      6.000      1.000      0.000      0.000
 7      0.000      0.000      0.000      0.000      0.000

```

Fig. C3. Input Data for computations of the ARF duct.



## APPENDIX D. FORTRAN Listing of DCTDOS

The listing of DCTDOS subprograms which follows can be assembled and run using Microsoft FORTRAN software with a microcomputer. It also requires three additional files, called NBS-TD-G, NBS-TD-N, and PREWTS. The first of these file contains gamma ray multi-reflection albedo data, the second contains neutron multi-reflection data, and the third must have a format acceptable to the subroutine WTLISTS, but is initially full of zeros. All three files are between 45,000 and 56,000 bytes in size. Running times earlier quoted did not include preparatory storage of the files in RAM memory.

For assembly purposes we have divided DCTDOS into four parts roughly equal in size because of a 64,000 byte memory limit. After assembly of each part the object programs have been linked into a run program of size roughly 180,000 bytes.

The subprograms have been little affected by the change from a mainframe to a microcomputer with the exception of ENTDAT, which required a special format.

```

1:*C   DCTDOS IS TO CALCULATE DOSE IN A DUCT-BEND-ROOM CONFIGURATION
2:     COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
3:     X BEND(10,6),SGRUPS(23),DWTS(21),   CRRNTO(9,10),RCRRNT(9,10)
4:     COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
5:     X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
6:     COMMON RCOSO,RCOSP,RCOS,DGRUPS(23),   VELO(9,21),VELP(9,21)
7:     X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
8:     COMMON IDOSE,HNDRSN(23),SEPI(23),   STH(23),SINR(23),THMLN(23)
9:     X,ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
10:    DIMENSION SVELO(22,20),SVELP(22,20),SCL(9),SELECT(9)
11:    DIMENSION IPR(6),ISO(6),IDO(6),NGI(6),ILP(6),DIMI(6)
12:    1 FORMAT(////20X,10H$=====$)
13:    66 FORMAT(I3,I3,I3,I3,I3,1PE10.3,30H   (IPR,ISO,IDO,ING,ILP,DIMEN))
14:    2 FORMAT(F10.3)
15:    3 FORMAT(F10.3,7D13.3)
16:    4 FORMAT(7(1PE10.3))
17:    5 FORMAT(1H )
18:    6 FORMAT(I5,5F10.5)
19:    7 FORMAT(1H1)
20:    8 FORMAT(25H                               NEXT CASE,F7.3)
21:    9 FORMAT(5I5,1PE10.3)
22:    10 FORMAT(55H IPRBLM NEL IPRINT JVARY JDEL DEL           )
23:    11 FORMAT(55H ISORS IDOSE ING LPIPE IEL DIMEN           )
24:    12 FORMAT(55H IEL COF ECC DELR DZRO SCL           )
25:    OPEN(10,FILE='B:NBS-DT-G')
26:    OPEN(11,FILE='B:NBS-DT-N')
27:    OPEN(05,FILE=' ')
28:    OPEN(06,FILE='A:PAPER')
29:    OPEN(07,FILE='B:PREWTS')
30: C   - - - -
31:     IOX=-1
32:     CALL WLISTS(JSTT,JXWT,IOX,NMAX,DWTS,SWTD,ENGAM,IDOSE,ISORS,LP)
33:     CALL ENTDAT
34:     MODG=0
35:     MODN=0
36:     13 READ(5,9) ISORS,IDOSE,ING,LPIPE,IEL,DIMEN
37:     IPRINT=IEL
38:     IF(IPRINT.GT.0) WRITE(6,7)
39:     16 IDEL=0
40:     READ(5,9) IPRBLM,NEL,IPRINT,JVARY,JDEL,DEL
41:     IF(IPRINT.EQ.0) GO TO 50
42:     WRITE(6,1)
43:     WRITE(6,11)
44:     WRITE(6,9) ISORS,IDOSE,ING,LPIPE,IEL,DIMEN
45:     WRITE(6,10)
46:     WRITE(6,9)IPRBLM,NEL,IPRINT,JVARY,JDEL,DEL
47:     50 READ(5,9) JDOS
48:     WRITE(6,9)JDOS
49:     IDOS=0
50:     IF(JDOS.LT.1) GO TO 68
51:     DO 67 I=1,JDOS
52:     READ(5,9) IPR(I),ISO(I),IDO(I),NGI(I),ILP(I),DIMI(I)
53:     IF(IPRINT.EQ.0)GO TO 69
54:     WRITE(6,9) IPR(I),ISO(I),IDO(I),NGI(I),ILP(I),DIMI(I)
55:     69 IPR(JDOS+1)=IPRBLM
56:     ISO(JDOS+1)=ISORS
57:     IDO(JDOS+1)=IDOSE
58:     NGI(JDOS+1)=ING
59:     ILP(JDOS+1)=LPIPE
60:     DIMI(JDOS+1)=DIMEN
61:     67 CONTINUE
62:     68 CALL DATDCT(NEL,SCL,SELECT)
63:     IF(IPRINT.LT.2) GO TO 14
64:     WRITE(6,12)
65:     WRITE(6,6) (NEXT(I),COF(I),ECC(I),DELR(I),DZRO(I),SCL(I),I=1,NEL)
66:     14 IC=1
67:     IF(IPRINT.EQ.0) GO TO 51
68:     WRITE(6,8) COF(JVARY)
69:     51 CONTINUE
70:     VELO(IC,1)=1.
71:     VEL(IC,1)=1.
72:     VELP(IC,1)=0.
73:     RVCT(1)=1.
74:     SO(IC)=COF(1)
75:     DO 15 I=2,21
76:     VELO(IC,I)=0.
77:     VELP(IC,I)=0.

```

```

78:      VEL(IC,I)=0.
79:      15 RVCT(I)=0.
80:      30 IF(NEL-IC) 1000,1000,31
81:      31 IC=IC+1
82:      S=SELECT(IC)
83:      IF(S.EQ.0.0) GO TO 33
84:      VO=1.+0.5*S*(1.-S)
85:      VP=1.-0.5*S*(1.+S)
86:      DO 40 N=1,21
87:      VELO(IC-1,N)=VO*VELO(IC-1,N)
88:      40 VELP(IC-1,N)=VP*VELP(IC-1,N)
89:      33 IEL=NEXT(IC)
90:      SCALE=SCL(IC)
91:      LP=2
92:      RDLNTH=DZRO(NEL)
93:      IF(IC+1-NEL)35,32,32
94:      32 LP=LPIPE
95: C
96:      35 GO TO (100,110,300,500,500,400,200,210), IEL
97: C
98:      100 DLNGTH=COF(IC)
99:      CALL DCTISO(DLNGTH,IC)
100:     GO TO 30
101: C
102:     110 DLNGTH=COF(IC)
103:     CALL DCTSLT(DLNGTH,IC)
104:     GO TO 30
105: C
106:     300 CALL TURN(SEN,IC)
107:     GO TO 30
108: C
109:     400 CALL ROOMX(IC,SVELO,SVELP)
110:     GO TO 30
111: C
112:     500 DLNGTH=COF(IC)
113:     CALL BTODR(DLNGTH,IC)
114:     GO TO 30
115: C
116:     200 IEL=NEL
117:     IF(NEXT(IC-1).EQ.8) GO TO 204
118:     IF(IPRBLM.LT.0) GO TO 205
119:     IF(JDOS.GE.1) GO TO 201
120:     IF(IDOSE.NE.5) GO TO 202
121:     IF(NEL.GE.5) GO TO 202
122:     201 CALL SRSDET(ISORS,ING,MODG,MODN)
123:     GO TO 205
124:     202 IF(IDEL.LT.1) CALL SRSDET(ISORS,ING,MODG,MODN)
125:     GO TO 205
126:     204 IF(IDOS.GT.0)GO TO 205
127:     IC=IC-2
128:     COF(IC)=COF(IC+2)
129:     SCL(IC)=SCL(IC+2)
130:     DELR(IC)=DELR(IC+2)
131:     ECC(IC)=ECC(IC+2)
132:     DZRO(IC)=DZRO(IC+2)
133:     205 CALL DOSE(IC)
134:     IF(JDOS.LT.1) GO TO 220
135:     IDOS=IDOS+1
136:     IPRBLM=IPR(IDOS)
137:     ISORS=ISO(IDOS)
138:     IDOSE=IDO(IDOS)
139:     ING=NGI(IDOS)
140:     LP=ILP(IDOS)
141:     DIMEN=DIMI(IDOS)
142:     206 IF(IDOS.GT.JDOS)GO TO 220
143:     IF(IPRINT.EQ.0)GO TO 207
144:     WRITE(6,66) IPRBLM,ISORS,IDOSE,ING,LP,DIMEN
145:     207 IF(NEXT(IC-1).EQ.8) GO TO 205
146:     GO TO 200
147:     210 IEL=NEL
148:     IF(IDEL.LT.1) CALL SRSDET(ISORS,ING,MODG,MODN)
149:     CALL DOSX(IC,SVELO,SVELP)
150:     IF(JDOS.LT.1) GO TO 212
151:     IDOS=IDOS+1
152:     IPRBLM=IPR(IDOS)
153:     ISORS=ISO(IDOS)
154:     IDOSE=IDO(IDOS)
155:     ING=NGI(IDOS)
156:     LP=ILP(IDOS)
157:     DIMEN=DIMI(IDOS)
158:     IF(IPRINT.EQ.0)GO TO 211
159:     IF (IDOS.GT.JDOS)GO TO 212
160:     WRITE(6,66) IPRBLM,ISORS,IDOSE,ING,LP,DIMEN

```

```

161: 211 IF(IDOS.LE.JDOS) GO TO 210
162: 212 IF(IC.EQ.NEL) GO TO 240
163:     IF(IC.LT.NEL) IC=IC+1
164:     IF(IC.EQ.NEL) GO TO 233
165: 220 IDEL=IDEL+1
166:     IF(IDEL-JDEL) 230,240,240
167: 230 COF(JVARY)=COF(JVARY)+DEL
168:     IF(JDOS.LT.1)GO TO 14
169: 233 IDOS=0
170:     IF(NEXT(IC-1).EQ.8)GO TO 33
171:     GO TO 14
172: 240 IF(IPRBLM**2-4) 16,13,1000
173: C   -   -   -
174: 1000 IOX=1
175:     CALL WTLISTS(JSTT,JSWT,IOX,NMAX,DWTS,SWTD,ENGAM,IDOSE,ISORS,LP)
176:     STOP
177:     END
178:     SUBROUTINE DATDCT(NEL,SCL,SELECT)
179:     COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
180:     X BEND(10,6),SGRUPS(23),DWTS(21), CRRNTO(9,10),RCRRNT(9,10)
181:     COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
182:     X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
183:     COMMON RCOSO,RCOSP,RCOS,DGRUPS(23), VELO(9,21),VELP(9,21)
184:     X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
185:     COMMON IDOSE,HNDRSN(23),SEPI(23), STH(23),SINR(23),THMLN(23)
186:     X,ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
187:     DIMENSION SCL(9),HGHT(9),WDTH(9),XLTH(9),AREA(9),OTHR(9),SELECT(9)
188:     CHARACTER*80 TITLE
189:     2 FORMAT(F10.3)
190:     3 FORMAT(F10.3,7D13.3)
191:     4 FORMAT(7(1PE10.3))
192:     5 FORMAT(1H )
193:     6 FORMAT(I5,5F10.3)
194:     7 FORMAT(1H1)
195:     9 FORMAT(5I5,2F5.2,I5)
196: 10 FORMAT(55H IEL   WIDTH-FT HEIGHT-FT LENGTH-FT OTHER SELECT )
197: 1  FORMAT(A80)
198:     READ(5,1) TITLE
199:     READ(5,6) (NEXT(IC),WDTH(IC),HGHT(IC),XLTH(IC),OTHR(IC),
200:     * SELECT(IC),IC=1,NEL)
201:     IF(IPRINT.EQ.0) GO TO 500
202:     WRITE(6,5)
203:     WRITE(6,1) TITLE
204:     WRITE(6,5)
205:     WRITE(6,10)
206:     WRITE(6,6) (NEXT(IC),WDTH(IC),HGHT(IC),XLTH(IC),OTHR(IC),
207:     * SELECT(IC),IC=1,NEL)
208: 500 CONTINUE
209:     COF(1)=WDTH(1)
210:     PEN=HGHT(1)
211:     ZDOOR=XLTH(1)
212: 53 FORMAT(55H 1=DOOR, PEN WALL (FT), DOOR THICKNESS(FT OF CONCRETE) )
213:     IF(IPRINT.LT.4) GO TO 501
214:     WRITE(6,5)
215:     WRITE(6,53)
216:     WRITE(6,6) NEXT(1),PEN,ZDOOR
217:     WRITE(6,5)
218: 501 CONTINUE
219:     ECC(1)=PEN
220:     DELR(1)=ZDOOR
221:     AREA(1)=0.
222:     DO 25 IC=2,NEL
223:     IEL=NEXT(IC)
224:     IF(IEL-7) 11,30,33
225: 11 AREA(IC)=WDTH(IC)*HGHT(IC)
226:     RAT=(2.*PEN+WDTH(IC))*(2.*PEN+HGHT(IC))/AREA(IC)
227:     SCL(IC)=SQRT(RAT)
228:     ECC(IC)=WDTH(IC)/HGHT(IC)
229:     DZRO(IC)=OTHR(IC)
230:     DELR(IC)=1.
231:     IF(IEL.EQ.3) GO TO 13
232:     DELR(IC)=OTHR(IC)
233:     IF(OTHR(IC).EQ.0.) DELR(IC)=1.
234:     IF(IEL.EQ.6) GO TO 22
235: 13 COF(IC)=XLTH(IC)/SQRT(AREA(IC))
236:     GO TO 25
237: 22 COF(IC)=WDTH(IC)
238:     ECC(IC)=HGHT(IC)
239:     DELR(IC)=XLTH(IC)
240:     DZRO(IC)=OTHR(IC)
241:     SCL(IC)=OTHR(IC+1)
242:     GO TO 25
243: 30 ICP=IC-1

```

```

244:      IF(NEXT(IC-1).EQ.8) ICP=IC-3
245:      COF(IC)=XLTH(IC)/SQRT(AREA(ICP))
246:      SCL(IC)=AREA(2)/AREA(ICP)
247:      IF(NEXT(IC-1).EQ.8) SCL(IC)=SCL(IC-1)
248:      DELR(IC)=DELR(ICP)
249:      ECC(IC)=ECC(ICP)
250:      DZRO(IC)=XLTH(IC)-.01
251:      GO TO 25
252: 33 COF(IC)=WDTH(IC)
253:      ECC(IC)=HGHT(IC)
254:      DELR(IC)=XLTH(IC)
255:      DZRO(IC)=OTHR(IC)/AREA(IC-2)
256:      SCL(IC)=AREA(2)/OTHR(IC)
257:      DZRO(1)=OTHR(1)
258: 25 CONTINUE
259: 1000 RETURN
260:      END
261:      SUBROUTINE SRSDET(ISORS,ING,MODG,MODN)
262:      COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
263: X BEND(10,6),SGRUPS(23),DWTS(21), CRRNTO(9,10),RCRRNT(9,10)
264:      COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
265: X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),COL,CO2,VEL(9,21),IPRINT
266:      COMMON RCOSO,RCOSP,RCOS,DGRUPS(23), VELO(9,21),VELP(9,21)
267: X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
268:      COMMON IDOSE,HNDRSN(23),SEPI(23),STH(23),SINR(23),THMLN(23)
269: X ,ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
270:      DIMENSION SWTD(21),TEMP(5)
271: 1 FORMAT(1H )
272: 2 FORMAT(55H DWTS//SWTD//ENGAM )
273: 3 FORMAT(7(1PE10.3))
274:      CNG=ING
275:      CG=CNG**2
276:      CN=1.+(CNG-CG)/2.
277:      CN=CN*DIMEN
278:      CG=CG*DIMEN
279: C      - - - -
280:      JSTT=0
281:      IOX=0
282:      CALL WTLISTS(JSTT,JXWT,IOX,NMAX,DWTS,SWTD,ENGAM,IDOSE,ISORS,LP)
283:      IF(IPRINT.GT.2)GO TO 100
284:      IF(JSTT.EQ.0)GO TO 15
285: C      - - - -
286: 50 IF(IDOSE.LT.5) GO TO 100
287:      IAG=ISORS-3
288:      IF(IAG.LT.1) GO TO 100
289:      XL=COF(2)
290:      CUT=0.5/SQRT(XL**2+0.25)
291:      CUTP=SQRT(1.-CUT**2)
292:      DO 97 J=1,5
293:      AMOD(3,J)=0.
294: 97 AMOD(4,J)=0.
295:      DO 98 J=1,5
296:      FJ=J
297:      IF(CUT.GT.AMOD(1,J)) GO TO 99
298: 98 CONTINUE
299: 99 NCUT=FJ+.1
300:      DO 980 J=1,5
301:      FJ=J
302:      IF(CUTP.GT.AMOD(1,J)) GO TO 990
303: 980 CONTINUE
304: 990 NCUTP=FJ+.1
305:      FR=(AMOD(2,NCUT)-CUT)/(AMOD(2,NCUT)-AMOD(1,NCUT))
306:      FRP=(AMOD(2,NCUTP)-CUTP)/(AMOD(2,NCUTP)-AMOD(1,NCUTP))
307:      DO 96 J=1,NCUT
308: 96 AMOD(3,J)=AWTS(IAG,J)
309:      DO 960 J=1,NCUTP
310: 960 AMOD(4,J)=AWTS(IAG+2,J)
311:      AMOD(3,NCUT)=FR*AMOD(3,NCUT)
312:      AMOD(4,NCUTP)=FRP*AMOD(4,NCUTP)
313:      SD=0.
314:      ED=0.
315:      DO 102 J=1,5
316:      SD=SD+AMOD(3,J)
317: 102 ED=ED+AMOD(4,J)
318:      DO 103 J=1,5
319:      AMOD(3,J)=AMOD(3,J)/(SD+1.E-7)
320: 103 AMOD(4,J)=AMOD(4,J)/(ED+1.E-7)
321:      TOT=0.
322:      TVP=VELP(2,2)+VELP(2,3)+VELP(2,4)+VELP(2,5)+VELP(2,6)+VELP(2,7)
323:      TVO=VELO(2,1)
324:      IF(IEL.LT.4) TVO=0.
325:      DO 200 J=1,5

```



```

326:      TEMP(J)=TVO*AMOD(4,J)+TVP*AMOD(3,J)
327:  200 TOT=TOT+TEMP(J)
328:      DO 201 J=1,5
329:  201 VEL1(J)=TEMP(6-J)/TOT
330:      IF(IPRINT.LT.4) GO TO 100
331:      WRITE(6,1)
332:      WRITE(6,2)
333:      WRITE(6,3) (AMOD(3,J),J=1,5)
334:      WRITE(6,3) (AMOD(4,J),J=1,5)
335:      WRITE(6,3) TVO,TVP,TOT,TOT,(VEL1(J),J=1,5)
336:      WRITE(6,2)
337:  498 WRITE(6,1)
338:  100 DO 12 MEG=1,23
339:      GO TO (105,110,115,120,125),ISORS
340:  105 SGRUPS(MEG)=SINR(MEG)
341:      GO TO 4
342:  110 SGRUPS(MEG)=STH(MEG)
343:      GO TO 4
344:  115 SGRUPS(MEG)=SEPI(MEG)
345:      GO TO 4
346:  120 SGRUPS(MEG)=SGAM(MEG)
347:      GO TO 4
348:  125 SGRUPS(MEG)=SCOBLT(MEG)
349:  4 GO TO (205,210,215,220,225),IDOSE
350:  205 DGRUPS(MEG)=SNYNEU(MEG)
351:      GO TO 12
352:  210 DGRUPS(MEG)=1.
353:      GO TO 12
354:  215 DGRUPS(MEG)=HNDRSN(MEG)
355:      GO TO 12
356:  220 DGRUPS(MEG)=THMLN(MEG)
357:      GO TO 12
358:  225 DGRUPS(MEG)=DGAM(MEG)
359:  12 CONTINUE
360:      IF(IDOSE.LT.5) GO TO 13
361:      NMAX=10
362:      MMAX=17
363:      DO 14 N=1,21
364:      DWTS(N)=0.
365:      SWTD(N)=0.
366:  14 ENGAM(N)=0.
367:      CALL SGD(SGRUPS,SWTS,WTSD,DGRUPS,VEL1,MODG)
368:      GO TO 11
369:  13 CALL SND(SGRUPS,SWTS,WTSD,DGRUPS,ENGAM,MODN)
370:      NMAX=21
371:      MMAX=23
372:  11 IF(JSTT.EQ.0)GO TO 15
373:      DO 19 N=1,NMAX
374:      DWTS(N)=0.
375:      SWTD(N)=0.
376:      DO 19 M=1,MMAX
377:      SWTD(N)=SWTD(N)+SWTS(M,N)*DGRUPS(M)
378:      DWTS(N)=DWTS(N)+WTSD(M,N)*SGRUPS(M)
379:  19 CONTINUE
380:  15 CALL WTLISTS(JSTT,JXWT,IOX,NMAX,DWTS,SWTD,ENGAM,IDOSE,ISORS,LP)
381:      DO 20 N=1,NMAX
382:      IF(IDOSE.EQ.5) DWTS(N)=SWTD(N)
383:      DWTS(N)=CN*DWTS(N)+CG*ENGAM(N)
384:  20 CONTINUE
385:  C      -      -      -
386:  25 IF(IPRINT.LT.2) GO TO 1000
387:      WRITE(6,1)
388:      WRITE(6,2)
389:      WRITE(6,3) (DWTS(N),N=1,NMAX)
390:      WRITE(6,1)
391:      WRITE(6,3) (SWTD(N),N=1,NMAX)
392:      WRITE(6,1)
393:      WRITE(6,3) (ENGAM(N),N=1,NMAX)
394:      WRITE(6,1)
395:  1000 RETURN
396:      END
397:      SUBROUTINE SND(SGRUPS,SWTS,WTSD,DGRUPS,ENGAM,MODN)
398:  $LARGE ALBDO
399:      DIMENSION SGRUPS(23),SWTS(23,21),WTSD(23,21),DGRUPS(23),
400:      * ALBDO(23,21,23),ENGAM(21)
401:      CHARACTER*80 TITLE
402:      1 FORMAT(A80)
403:  7010 FORMAT(I5)
404:  7020 FORMAT(8(1PE10.3))
405:      IF(MODN.EQ.1) GO TO 12
406:      IUT=11
407:      REWIND IUT
408:      READ(IUT,1) TITLE

```

```

409:      IF(IPRINT.LT.4) GO TO 12
410:      WRITE(6,1) TITLE
411: 12 FG=.095/.75*4.5E-10
412:      DO 25 N=1,21
413:      ENGAM(N)=0.
414:      DO 25 M=1,23
415:      SWTS(M,N)=0.
416: 25 WTSD(M,N)=0.
417:      DO 350 MEG=1,23
418:      IF(MODN.EQ.1) GO TO 35
419:      DO 5 M=1,20
420:      DO 5 J=1,23
421:      5 ALBDO(J,M,MEG)=0.
422:      3 READ(IUT,7010) ISORS
423:      DO 30 J=ISORS,23
424: 30 READ(IUT,7020) (ALBDO(J,M,MEG),M=1,20)
425: 35 SWTS(ISORS,1)=1.*SGRUPS(ISORS)
426:      WTSD(ISORS,1)=1.*DGRUPS(ISORS)
427:      DO 100 N=2,21
428:      DO 100 M=ISORS,23
429:      SWTS(M,N)=SWTS(M,N)+ALBDO(M,N-1,MEG)*SGRUPS(ISORS)
430:      ENGAM(N)=ENGAM(N)+ALBDO(M,N-1,MEG)*FG*SGRUPS(ISORS)
431: 100 WTSD(ISORS,N)=WTSD(ISORS,N)+ALBDO(M,N-1,MEG)*DGRUPS(M)
432: 350 CONTINUE
433:      IF(MODN.EQ.0) CLOSE(11)
434:      MODN=1
435:      RETURN
436:      END
437:      SUBROUTINE SGD(SGRUPS,SWTS,WTSD,DGRUPS,VEL1,MODG)
438: $LARGE ALB
439:      DIMENSION SGRUPS(23),SWTS(23,21),WTSD(23,21),DGRUPS(23),
440:      * ALB(17,9,85),VEL1(5)
441:      CHARACTER*80 TITLE
442:      1 FORMAT(A80)
443:      2 FORMAT(2I5)
444:      3 FORMAT(1X,7(1PE9.3),2(1PE8.2))
445:      IF(MODG.EQ.1) GO TO 12
446:      IUT=10
447:      REWIND IUT
448:      READ(IUT,1) TITLE
449:      IF(IPRINT.LT.4)GO TO 12
450:      WRITE(6,1) TITLE
451: 12 CONTINUE
452:      DO 25 M=1,23
453:      DO 25 N=1,21
454:      SWTS(M,N)=0.
455: 25 WTSD(M,N)=0.
456:      DO 350 K=1,85
457:      IF(MODG.EQ.1) GO TO 35
458:      KE=(K-1)/5
459:      LOB=K-5*KE
460:      DO 5 M=1,9
461:      DO 5 J=1,17
462:      5 ALB(J,M,K)=0.
463:      READ(IUT,2) IK,IE
464:      DO 30 J=IE,17
465: 30 READ(IUT,3) (ALB(J,M,K),M=1,9)
466: 35 SWTS(IE,1)=SWTS(IE,1)+VEL1(LOB)*SGRUPS(IE)
467:      WTSD(IE,1)=WTSD(IE,1)+VEL1(LOB)*DGRUPS(IE)
468:      DO 100 N=2,10
469:      DO 100 M=IE,17
470:      IF(N.LE.2) GO TO 40
471:      IF(M.GT.16) GO TO 40
472:      IF(N.GT.M-10) GO TO 100
473: 40 CONTINUE
474:      SWTS(M,N)=SWTS(M,N)+ALB(M,N-1,K)*SGRUPS(IE)*VEL1(LOB)
475: 100 WTSD(IE,N)=WTSD(IE,N)+ALB(M,N-1,K)*DGRUPS(M)*VEL1(LOB)
476: 350 CONTINUE
477:      IF(MODG.EQ.0) CLOSE(10)
478:      MODG=1
479:      RETURN
480:      END
481:      SUBROUTINE ORDST(RHO,A,AL,ORLIST)
482:      DIMENSION TEMP(21,21),BLST(21),ORLIST(21)
483:      AB=A*RHO
484:      B=(A-AB)/(1.-AB*(1.36788-.36788*A))
485:      ALB=(1.-B)*(1.-AB)*AL/RHO/(1.-A)
486:      BLST(1)=1.
487:      TEMP(1,1)=1.
488:      DO 10 I=2,21
489:      BLST(I)=B*BLST(I-1)
490: 10 TEMP(1,I)=TEMP(1,I-1)*AB
491:      TEMP(1,1)=0.

```

```

492:      DO 20 J=2,21
493:      DO 20 I=J,21
494:      20 TEMP(J,I)=TEMP(J-1,I-1)
495:      DO 30 I=1,21
496:      ORLIST(I)=0.
497:      DO 30 J=1,I
498:      30 ORLIST(I)=ORLIST(I)+TEMP(J,I)*BLST(J)
499:      DO 40 I=1,21
500:      40 ORLIST(I)=ORLIST(I)*ALB
501:      RETURN
502:      END
503:      SUBROUTINE DCTISO(DLNGTH,IC)
504:      COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
505:      X BEND(10,6),SGRUPS(23),DWTS(21),CRRNTO(9,10),RCRRNT(9,10)
506:      COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
507:      X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
508:      COMMON RCOSO,RCOSP,RCOS,DGRUPS(23),VELO(9,21),VELP(9,21)
509:      X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
510:      COMMON IDOSE,HNDRSN(23),SEPI(23),STH(23),SINR(23),THMLN(23)
511:      X,ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
512:      DIMENSION XI(5),ORLIST(21),ESC(21)
513:      DO 4 I=1,5
514:      4 XI(I)=0.
515:      1 FORMAT(1H )
516:      2 FORMAT(I3,11(1PE10.2))
517:      3 FORMAT(I3,7E13.3)
518:      IF(DELR(IC).LT.0.) GO TO 131
519:      DO 130 J=1,21
520:      130 ESC(J)=DELR(IC)
521:      GO TO 133
522:      131 ESC(1)=1.
523:      DO 132 J=2,21
524:      132 ESC(J)=ESC(J-1)*(-DELR(IC))
525:      133 CONTINUE
526:      DLNGTH=DLNGTH/SCALE
527:      EC=ECC(IC)
528:      ECCO=-.06*LOG(EC)**2*DLNGTH/SQRT(3.182+DLNGTH**2)
529:      ECCP=-.109*LOG(EC)**2
530:      ECCO=EXP(ECCO)
531:      ECCP=EXP(ECCP)
532:      XI(1)=2.
533:      XI(4)=2.
534:      IF(DLNGTH.GE.8.) GO TO 10
535:      LX=(2.*DLNGTH+1.)
536:      PWR=(DLNGTH-DIAMS(LX))/(DIAMS(LX+1)-DIAMS(LX))
537:      DO 15 KX=1,5
538:      Y2=DUCTO(LP,LX+1,KX)
539:      Y1=DUCTO(LP,LX,KX)
540:      RCRRT(IC,KX)=0.
541:      15 CRRNTO(IC,KX)=Y1*(Y2/Y1)**PWR
542:      GO TO 13
543:      10 DO 12 KX=1,5
544:      AO=DUCTO(LP,17,KX)
545:      A2=DUCTO(LP,13,KX)
546:      CALL XTRAP(AO,A2,XI(KX),DLNGTH,AXTR)
547:      12 CRRNTO(IC,KX)=AXTR
548:      13 CONTINUE
549:      AL=(-.35+CRRNTO(IC,2))/(.5+DLNGTH**3)
550:      A=-.30+CRRNTO(IC,3)
551:      RHO=.95
552:      RMTX(1,1)=CRRNTO(IC,1)
553:      CALL ORDST(RHO,A,AL,ORLIST)
554:      DO 19 I=2,21
555:      RMTX(1,I)=ORLIST(I)
556:      DO 19 J=2,I
557:      19 RMTX(J,I)=RMTX(J-1,I-1)
558:      20 SO(IC)=1.-CRRNTO(IC,4)
559:      SP(IC)=-1.+2.*CRRNTO(IC,5)
560:      VELO(IC,1)=RMTX(1,1)*RVCT(1)*ESC(1)*ECCO
561:      DO 31 J=2,21
562:      VELO(IC,J)=RMTX(J,J)*RVCT(J)*ESC(1)*ECCO
563:      VELP(IC,J)=0.
564:      IP=J-1
565:      DO 30 I=1,IP
566:      30 VELP(IC,J)=VELP(IC,J)+RMTX(I,J)*RVCT(I)*ESC(J-I+1)
567:      VELP(IC,J)=VELP(IC,J)*ECCP
568:      31 CONTINUE
569:      DO 32 J=1,21
570:      VEL(IC,J)=VELO(IC,J)+VELP(IC,J)
571:      32 RVCT(J)=VEL(IC,J)
572:      DLNGTH=DLNGTH*SCALE
573:      IF(IPRINT.LT.4) GO TO 1000

```

```
574:    40 WRITE(6,1)
575:      WRITE(6,3) IC ,DLNGTH, (CRRNTO(IC,KX),KX=1,5)
576:      WRITE(6,3) IC, CRRNTO(IC,1),A,AL,SO(IC),SP(IC)
577:      WRITE(6,2) IEL, (VELO(IC,J),J=1,11)
578:      IND=1
579:      WRITE(6,2) IND, (VELP(IC,J),J=1,11)
580:      IND=2
581:      WRITE(6,2) IND, (RVCT(J),J=1,11)
582:  1000 RETURN
583:      END
584:
```

\*

```

1:*   SUBROUTINE DCTSLT(DLNGTH,IC)
2:   COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
3:   X BEND(10,6),SGRUPS(23),DWTS(21),CRRNTO(9,10),RCRRNT(9,10)
4:   COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
5:   X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
6:   COMMON RCOSO,RCOSP,RCOS,DGRUPS(23),VELO(9,21),VELP(9,21)
7:   X VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
8:   COMMON IDOSE,HNDRSN(23),SEPI(23),STH(23),SINR(23),THMLN(23)
9:   X ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
10:  DIMENSION RMTX1(21,21),RMTX2(21,21),VO1(21),VO2(21),VP1(21),
11:  X VP2(21),XI(5),ORLIST(21),ESC(21)
12:  DO 4 I=1,5
13:  4 XI(I)=0.
14:  1 FORMAT(1H )
15:  2 FORMAT(I3,11(1PE10.2))
16:  3 FORMAT(I3,9E12.4)
17:  IF(DELR(IC).LT.0.) GO TO 131
18:  DO 130 J=1,21
19:  130 ESC(J)=DELR(IC)
20:  GO TO 133
21:  131 ESC(1)=1.
22:  DO 132 J=2,21
23:  132 ESC(J)=ESC(J-1)*(-DELR(IC))
24:  133 CONTINUE
25:  DLNGTH=DLNGTH/SCALE
26:  CO1=.75/(1.+SO(IC-1))-0.5
27:  CO2=1.-CO1
28:  U=.5*(1.+SO(IC-1))
29:  ZM=U/SQRT(1.-U**2)
30:  VALO=1.-DLNGTH/ZM
31:  IF(VALO.LT.0.) VALO=0.
32:  7 EC=ECC(IC)
33:  ECCO=-.06*LOG(EC)**2*DLNGTH/SQRT(3.182+DLNGTH**2)
34:  ECCP=-.109*LOG(EC)**2
35:  ECCO=EXP(ECCO)
36:  ECCP=EXP(ECCP)
37:  IF(DLNGTH.GE.8.) GO TO 10
38:  LX=(2.*DLNGTH+1.)
39:  PWR=(DLNGTH-DIAMS(LX))/(DIAMS(LX+1)-DIAMS(LX))
40:  DO 15 KX=1,5
41:  Y2=DUCT1(LP,LX+1,KX)
42:  Y1=DUCT1(LP,LX,KX)
43:  RCRRT(KX,1)=Y1*(Y2/Y1)**PWR
44:  Y2=DUCT2(LP,LX+1,KX)
45:  Y1=DUCT2(LP,LX,KX)
46:  15 RCRRT(KX,2)=Y1*(Y2/Y1)**PWR
47:  GO TO 13
48:  10 DO 12 KX=1,5
49:  AO=DUCT1(LP,17,KX)
50:  A2=DUCT1(LP,13,KX)
51:  CALL XTRAP(AO,A2,XI(KX),DLNGTH,AXTR)
52:  RCRRT(KX,1)=AXTR
53:  AO=DUCT2(LP,17,KX)
54:  A2=DUCT2(LP,13,KX)
55:  CALL XTRAP(AO,A2,XI(KX),DLNGTH,AXTR)
56:  12 RCRRT(KX,2)=AXTR
57:  13 CONTINUE
58:  AL1=(-.35+RCRRT(2,1))/(.5+DLNGTH**3)
59:  AL2=(-.35+RCRRT(2,2))/(.5+DLNGTH**3)
60:  A1=-.3+RCRRT(3,1)
61:  A2=-.3+RCRRT(3,2)
62:  RHO=.85
63:  RMTX1(1,1)=VALO
64:  RMTX2(1,1)=VALO
65:  CALL ORDST(RHO,A1,AL1,ORLIST)
66:  DO 17 I=2,21
67:  17 RMTX1(1,I)=ORLIST(I)
68:  CALL ORDST(RHO,A2,AL2,ORLIST)
69:  DO 19 I=2,21
70:  RMTX2(1,I)=ORLIST(I)
71:  DO 19 J=2,I
72:  RMTX1(J,I)=RMTX1(J-1,I-1)
73:  19 RMTX2(J,I)=RMTX2(J-1,I-1)
74:  20 RCO1=2./RCRRT(4,1)
75:  RCO2=2./(1.+RCRRT(4,2))
76:  RCP1=1./RCRRT(5,1)
77:  RCP2=1./RCRRT(5,2)
78:  VO1T=1.0101E-9
79:  VO2T=2.002E-9
80:  VP1T=1.0101E-9
81:  VP2T=2.0202E-9
82:  DO 31 J=1,21

```



```

83:      VO1(J)=RMTX1(J,J)*VELO(IC-1,J)*ESC(1)
84:      VO2(J)=RMTX2(J,J)*VELO(IC-1,J)*ESC(1)
85:      VP1(J)=0.
86:      VP2(J)=0.
87:      JP=J-1
88:      IF(J-1) 28,28,29
89:      DO 30 I=1,JP
90:      VP1(J)=VP1(J)+RMTX1(I,J)*VELO(IC-1,I)*ESC(J-I+1)
91:      VP2(J)=VP2(J)+RMTX2(I,J)*VELO(IC-1,I)*ESC(J-I+1)
92:      VO1T=VO1T+VO1(J)
93:      VO2T=VO2T+VO2(J)
94:      VP1T=VP1T+VP1(J)
95:      VP2T=VP2T+VP2(J)
96:      31 CONTINUE
97:      DO 32 J=1,21
98:      VELO(IC,J)=CO1*VO1(J)+CO2*VO2(J)
99:      32 VELP(IC,J)=CO1*VP1(J)+CO2*VP2(J)
100:     DO 33 J=1,21
101:     VELO(IC,J)=VELO(IC,J)*ECCO
102:     VELP(IC,J)=VELP(IC,J)*ECCP
103:     VEL(IC,J)=VELO(IC,J)+VELP(IC,J)
104:     33 RVCT(J)=VEL(IC,J)
105:     SO(IC)=-1.+2.*(CO1*VO1T+CO2*VO2T)/(CO1*VO1T*RCO1+CO2*VO2T*RCO2)
106:     SP(IC)=-1.+2.*(CO1*VP1T+CO2*VP2T)/(CO1*VP1T*RCP1+CO2*VP2T*RCP2)
107:     DLNGTH=DLNGTH*SCALE
108:     IF(IPRINT.NE.4) GO TO 1000
109:     40 WRITE(6,1)
110:     WRITE(6,3) IC ,DLNGTH,(RCRRNT(KX,1) ,KX=1,5)
111:     WRITE(6,3) IC ,DLNGTH,(RCRRNT(KX,2) ,KX=1,5)
112:     WRITE(6,3) IC,AL1,A1,AL2,A2,CO1,SO(IC),SP(IC),CO2
113:     WRITE(6,3) IC,VO1T,VO2T,VP1T,VP2T,RCO1,RCO2,RMTX1(1,1),RMTX2(1,1)
114:     WRITE(6,2) IEL,(VELO(IC,J),J=1,11)
115:     WRITE(6,2) IEL,(VELP(IC,J),J=1,11)
116:     IND=13
117:     WRITE(6,2) IND,(VO1(J),J=1,21,2)
118:     WRITE(6,2) IND,(VO2(J),J=1,21,2)
119:     1000 RETURN
120:     END
121:     SUBROUTINE XTRAP(AO,A2,XI,FA,AINT)
122:     SAVE=XI
123:     FO=8.
124:     F2=6.
125:     XI=LOG(A2/AO)/LOG(FO/F2)-.05
126:     DL=0.1
127:     FOL=FO**DL
128:     F2L=F2**DL
129:     FAL=FA**DL
130:     DO=AO*FO**XI*FOL
131:     D2=A2*F2**XI*F2L
132:     AL1=(DO-D2)/(FOL-F2L)
133:     IF(AL1) 10,10,5
134:     5 AL2=(FOL*D2-F2L*DO)/(FOL-F2L)
135:     AINT=(AL1+AL2/FAL)/FA**XI
136:     GO TO 1000
137:     10 XI=LOG(A2/AO)/LOG(FO/F2)
138:     AL1=(AO*FO**(XI+1.)-A2*F2**(XI+1.))/(FO-F2)
139:     AINT=AL1/FA**XI
140:     XI=SAVE
141:     1000 RETURN
142:     END
143:     SUBROUTINE TURN(SEN,IC)
144:     COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP.
145:     X BEND(10,6),SGRUPS(23),DWTS(21), CRRNTO(9,10),RCRRNT(9,10)
146:     COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
147:     X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
148:     COMMON RCOSO,RCOSP,RCOS,DGRUPS(23), VELO(9,21),VELP(9,21)
149:     X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
150:     COMMON IDOSE,HNDRSN(23),SEPI(23), STH(23),SINR(23),THMLN(23)
151:     X,ENGAM(21),DIMEN,SCOBTLT(23), AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
152:     DIMENSION RX1(13),RINF(13),ORLIST(21),ESC(21)
153:     DATA RX1/.88,.86,.84,.82,.79,.74,.69,.62,.54,.44,.26,.01,.001/
154:     DATA RINF/.86,.83,.80,.76,.71,.66,.58,.48,.37,.24,.01.
155:     * .001,.0001/
156:     IF(DELR(IC).LT.0.) GO TO 131
157:     DO 130 J=1,21
158:     130 ESC(J)=DELR(IC)
159:     GO TO 133
160:     131 ESC(1)=1.
161:     DO 132 J=2,21
162:     132 ESC(J)=ESC(J-1)*(-DELR(IC))
163:     133 CONTINUE
164:     CO=0.
165:     CP=0.

```

```

166:      AZFAC=1.-DZRO(IC)
167:      CP1=0.
168:      CP2=0.
169:      RCOSO=2./(1.+SO(IC-1))
170:      RCOSP=2./(1.+SP(IC-1))
171:      DO 305 I=1,21
172:      CO=CO+VELO(IC-1,I)
173:      CP=CP+VELP(IC-1,I)
174:      VELO(IC,I)=0.
175:      VELP(IC,I)=0.
176: 305 RVCT(I)=VELO(IC-1,I)
177:      RCOSO=(CO*RCOSO+CP*RCOSP)/(CO+CP)
178:      RCOSP=RCOSO
179:      RCOS=RCOSO
180:      IX=1
181: 134 SEN=2./(RCOS+1.E-6)-1.
182: C
183:      RCSS=COF(IC)
184:      KS=(SEN*10.+1.)
185:      IF(KS.LT.1) KS=1
186:      RKS=KS
187:      PWR=(SEN*10.+1.00001-RKS)
188:      C1=RX1(KS)*(RX1(KS+1)/RX1(KS))**PWR
189:      CINF=RINF(KS)*(RINF(KS+1)/RINF(KS))**PWR
190:      BD=(1.-C1)/(C1-CINF)
191:      BN=CINF*BD
192:      RX=RCSS**1.5
193:      RDUX=(1.+BN*RX)/(1.+BD*RX)
194: C
195:      1 FORMAT(1H )
196:      2 FORMAT(I3,11(1PE10.2))
197:      3 FORMAT(I3,7E13.3)
198:      LS=(5.*SEN+3.)
199:      DO 15 KX=1,5
200:      Y2=BEND(LS+1,KX)
201:      Y1=BEND(LS,KX)
202:      PWR=(SEN-SINVAL(LS))/(SINVAL(LS+1)-SINVAL(LS)+.00001)
203: 15 CRRNTO(IC,KX)=Y1*(Y2/Y1)**PWR
204:      AL=CRRNTO(IC,2)
205:      A=CRRNTO(IC,3)
206:      SO(IC)=CRRNTO(IC,4)*SQRT(2.-SEN)-1.
207:      SP(IC)=2.*CRRNTO(IC,5)-1.
208:      IF(IX.GT.0) SO1=SO(IC)
209:      IF(IX.GT.0) SP1=SP(IC)
210:      IF(IX.LT.0) SO2=SO(IC)
211:      IF(IX.LT.0) SP2=SP(IC)
212:      RHO=.75
213:      RMTX(1,1)=CRRNTO(IC,1)
214:      CALL ORDST(RHO,A,AL,ORLIST)
215:      DO 20 I=2,21
216:      RMTX(1,I)=ORLIST(I)
217:      DO 19 J=2,I
218: 19 RMTX(J,I)=RMTX(J-1,I-1)
219: 20 CONTINUE
220:      VELO(IC,1)=VELO(IC,1)+RMTX(1,1)*RVCT(1)*ESC(1)
221:      IF(IX.GT.0) CO1=VELO(IC,1)
222:      IF(IX.LT.0) CO2=VELO(IC,1)
223:      DO 31 J=2,21
224:      VELO(IC,J)=VELO(IC,J)+RMTX(J,J)*RVCT(J)*ESC(1)
225:      IP=J-1
226:      DO 30 I=1,IP
227:      DVP=RMTX(I,J)*RVCT(I)*ESC(J-I+1)*RDUX
228: 30 VELP(IC,J)=VELP(IC,J)+DVP
229:      IF(IX.GT.0) CO1=CO1+VELO(IC,J)
230:      IF(IX.GT.0) CP1=CP1+VELP(IC,J)
231:      IF(IX.LT.0) CO2=CO2+VELO(IC,J)
232:      IF(IX.LT.0) CP2=CP2+VELP(IC,J)
233: 31 CONTINUE
234: C
235:      IF(IX.LT.0) GO TO 325
236:      RENORM=1.+(1.-AZFAC)*CO1/(CO+CP*1.E-5)
237:      DO 315 J=1,21
238:      VELO(IC,J)=VELO(IC,J)*AZFAC
239:      VELP(IC,J)=VELP(IC,J)*RENORM
240: 315 RVCT(J)=VELP(IC-1,J)
241:      CO1=CO1*AZFAC
242:      CP1=CP1*RENORM
243:      RCOS=RCOSP
244:      IX=IX-2
245:      GO TO 134
246: C
247: 325 CONTINUE

```

```

248:      SO(IC)=-1.+(1.+SO1)*(1.+SO2)*CO2/(CO1*(SO2-SO1)+CO2*(1.+SO1))
249:      SP(IC)=-1.+(1.+SP1)*(1.+SP2)*CP2/(CP1*(SP2-SP1)+CP2*(1.+SP1))
250:      DO 32 J=1,21
251:      VEL(IC,J)=VELO(IC,J)+VELP(IC,J)
252:      32 RVCT(J)=VELP(IC,J)
253:      IF(IPRINT.NE.4) GO TO 1000
254:      40 WRITE(6,1)
255:      WRITE(6,3) IC,SEN,(CRRNTO(IC,KX),KX=1,5)
256:      WRITE(6,3) IC ,RMTX(1,1),AL,A,SO(IC),SP(IC)
257:      WRITE(6,2) IEL,(VELO(IC,J),J=1,11)
258:      IND=1
259:      WRITE(6,2) IND,(VELP(IC,J),J=1,11)
260:      IND=2
261:      WRITE(6,2) IND,(RVCT(J),J=1,11)
262:      1000 RETURN
263:      END
264:      SUBROUTINE BTODR(DLNGTH,IC)
265:      COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
266:      X BEND(10,6),SGRUPS(23),DWTS(21), CRRNTO(9,10),RCRRNT(9,10)
267:      COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
268:      X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),COL,CO2,VEL(9,21),IPRINT
269:      COMMON RCOSO,RCOSP,RCOS,DGRUPS(23), VELO(9,21),VELP(9,21)
270:      X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
271:      COMMON IDOSE,HNDRSN(23),SEPI(23),STH(23),SINR(23),THMLN(23)
272:      X,ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
273:      DIMENSION STRVEL(22),TVLO(22),TVLP(22),SIDE(22,22),HACK(22,22),
274:      X VPB(22),VPS(22),VOS(22),VOB(22)
275:      1 FORMAT(1H )
276:      2 FORMAT(I3,11(1PE10.2))
277:      3 FORMAT(I3,9E12.4)
278:      C
279:      STRSO=SO(IC-1)
280:      DLBND=1.
281:      DISTNS=DLNGTH+DLBND
282:      DO 10 I=1,21
283:      10 STRVEL(I)=VELO(IC-1,I)
284:      TVO=0.
285:      TVP=0.
286:      C
287:      C DIRECT COMPONENT
288:      CALL DCTSLT(DLNGTH,IC)
289:      DO 20 I=1,21
290:      TVLO(I)=VELO(IC,I)
291:      TVLP(I)=VELP(IC,I)
292:      TVO=TVO+VELO(IC,I)
293:      20 TVP=TVP+VELP(IC,I)
294:      RCO=TVO*2./(1.001+SO(IC))
295:      RCP=TVP*2./(1.001+SP(IC))
296:      C
297:      C SIDE COMPONENT
298:      DO 200 I=2,21
299:      200 VELO(IC-1,I)=0.
300:      VELO(IC-1,1)=1.
301:      VELP(IC,1)=0.
302:      SO(IC-1)=0.5
303:      CALL DCTSLT(DISTNS,IC)
304:      DO 205 I=1,20
305:      205 SIDE(1,I)=(VELP(IC,I+1)-.54*VELP(IC,I))/ .1162
306:      DO 210 I=2,21
307:      IP=I-1
308:      DO 207 J=1,IP
309:      207 SIDE(I,J)=0.
310:      DO 210 J=I,21
311:      210 SIDE(I,J)=SIDE(I-1,J-1)
312:      VPS(1)=0.
313:      VOS(1)=0.
314:      DO 220 J=2,20
315:      VOS(J)=SIDE(J,J)*(VELP(IC-1,J)-.2055*STRVEL(J-1))
316:      VPS(J)=0.
317:      JP=J-1
318:      DO 220 I=1,JP
319:      VPS(J)=VPS(J)+SIDE(I,J)*VELP(IC-1,I)
320:      IF(I.GT.1) VPS(J)=VPS(J)-SIDE(I,J)*.2055*STRVEL(I-1)
321:      220 CONTINUE
322:      SVO=0.
323:      SVP=0.
324:      DO 230 I=2,20
325:      TVLO(I)=TVLO(I)+.829*VOS(I)
326:      TVLP(I)=TVLP(I)+.829*VPS(I)
327:      SVO=SVO+.829*VOS(I)
328:      230 SVP=SVP+.829*VPS(I)
329:      TVO=TVO+SVO
330:      TVP=TVP+SVP

```

```

331:      RCO=RCO+SVO*2. / (1.001+SP(IC))
332:      RCP=RCP+SVP*2. / (1.001+SP(IC))
333: C
334: C BACK COMPONENT
335:      DO 300 I=2,21
336:      VELO(IC-1,I)=0.
337: 300 RVCT(I)=0.
338:      RVCT(1)=1.
339:      VELO(IC-1,1)=1.
340:      CALL DCTISO(DISTNS,IC)
341:      BVP=0.
342:      BVO=0.
343:      DO 305 I=1,21
344:      BVO=BVO+VELO(IC,I)
345:      BVP=BVP+VELP(IC,I)
346: 305 BACK(1,I)=VEL(IC,I)
347:      RCOB=2. / (1.001+SO(IC))
348:      RCPB=2. / (1.001+SP(IC))
349: C
350:      SO(IC-1)=-0.5
351:      CALL DCTSLT(DISTNS,IC)
352:      BVPM=0.
353:      DO 310 I=2,21
354:      BVPM=BVPM+VELP(IC,I)
355: 310 BACK(1,I)=(BACK(1,I)-1.37*VELP(IC,I)) / .205
356:      BACK(1,1)=BACK(1,1) / .205
357:      RCSPB=BVP*RCPB-1.4*BVPM*2. / (1.001+SP(IC))
358:      BVP=BVP-1.4*BVPM
359:      RCSPB=RCSPB/BVP
360: C
361:      DO 315 I=2,21
362:      IP=I-1
363:      DO 312 J=1,IP
364: 312 BACK(I,J)=0.
365:      DO 315 J=I,21
366: 315 BACK(I,J)=BACK(I-1,J-1)
367:      VOB(1)=0.
368:      VPB(1)=0.
369:      DO 320 J=2,21
370:      VOB(J)=BACK(J,J)*(VELP(IC-1,J)+STRVEL(J-1))
371:      VPB(J)=0.
372:      JP=J-1
373:      DO 320 I=1,JP
374:      IF(I.GT.1) VPB(J)=VPB(J)+BACK(I,J)*STRVEL(I-1)
375: 320 VPB(J)=VPB(J)+BACK(I,J)*VELP(IC-1,I)
376:      SVO=0.
377:      SVP=0.
378:      DO 330 I=1,21
379:      SVO=SVO+.171*VOB(I)
380:      SVP=SVP+.171*VPB(I)
381:      VELO(IC,I)=TVLO(I)+.171*VOB(I)
382:      VELP(IC,I)=TVLP(I)+.171*VPB(I)
383: 330 VEL(IC,I)=VELO(IC,I)+VELP(IC,I)
384:      TVO=TVO+SVO
385:      RCO=RCO+SVO*RCOB
386:      TVP=TVP+SVP
387:      RCP=RCP+SVP*RCSPB
388:      SO(IC)=-1.+2.*TVO/RCO
389:      SP(IC)=-1.+2.*TVP/RCP
390:      VELP(IC,1)=0.
391:      VEL(IC,1)=VELO(IC,1)
392:      DO 50 I=1,21
393:      VELO(IC-1,I)=STRVEL(I)
394: 50 RVCT(I)=VEL(IC,I)
395:      SO(IC-1)=STRSO
396:      IF(IPRINT.NE.4) GO TO 1000
397: 41 WRITE(6,1)
398:      WRITE(6,3) IC,SO(IC),SP(IC),TVO,RCO,TVP,RCP,VELO(IC,2),VELP(IC,2)
399:      WRITE(6,2) IEL,(VELO(IC,J),J=1,21,2)
400:      WRITE(6,2) IEL,(VELP(IC,J),J=1,21,2)
401:      IND=17
402:      WRITE(6,2) IND,(TVLO(J),J=1,21,2)
403:      WRITE(6,2) IND,(TVLP(J),J=1,21,2)
404: 1000 RETURN
405:      END

```

\*



```

1:* SUBROUTINE ROOMX(IC,SVELO,SVELP)
2: COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP.
3: X BEND(10,6),SGRUPS(23),DWTS(21), CRRNTO(9,10),RCRRNT(9,10)
4: COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
5: X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
6: COMMON RCOSO,RCOSP,RCOS,DGRUPS(23), VELO(9,21),VELP(9,21)
7: X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
8: COMMON IDOSE,HNDRSN(23),SEPI(23), STH(23),SINR(23),THMLN(23)
9: X,ENGAM(21),DIMEN,SCOBLT(23), AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
10: DIMENSION D(20),DL(20),W(20),OVEL(22,20),PVEL(22,20),SVELO(22,20),
11: * SVELP(22,20)
12: 1 FORMAT(1H )
13: 2 FORMAT(I3,11(1PE10.2))
14: 3 FORMAT(11(1PE10.2))
15: WTH=COF(IC)
16: HT=ECC(IC)
17: XL=DELR(IC)
18: AEX=SCALE
19: AEXR=DZRO(IC+1)
20: IF(AEXR.GT.1.) AEXR=1.
21: WD=DZRO(IC)
22: RMCASE=DZRO(1)
23: WDP=COF(IC+1)
24: NW=(ECC(IC+1)+1.01)
25: NL=(DELR(IC+1)+1.01)
26: NT=NW*NL
27: DELW=WDP/ECC(IC+1)
28: DELL=XL/DELR(IC+1)
29: TN=SQRT(1.-SO(IC-1)**2)/SO(IC-1)
30: A=SP(IC-1)*(1.+SO(IC-1))/(1.+SP(IC-1))/SO(IC-1)
31: RR=AEX/(WTH*HT+HT*XL+XL*WTH)
32: AL=.75
33: ALR=AL*(1.-RR/2.)
34: TNR=SQRT(HT*WTH/3.14159)/XL
35: C POSITION 1,N=0 (UNREFLECTED, IN FRONT OF DUCT)
36: OVEL(1,1) =2./(1.+SO(IC-1))*AEXR
37: PVEL(1,1)=2./(1.+SP(IC-1))*AEXR
38: C POSITIONS 2 THRU NT, N IS 2 OR GREATER
39: DO 11 I=2,NT
40: OVEL(3,I)=(1.+SO(IC-1))*RR*ALR**2
41: PVEL(3,I)=(1.+SP(IC-1))*RR*ALR**2
42: DO 11 J=4,21
43: OVEL(J,I)=ALR*OVEL(J-1,I)
44: 11 PVEL(J,I)=ALR*PVEL(J-1,I)
45: C POSITION 1, N IS 2 OR GREATER
46: DO 12 J=3,21
47: OVEL(J,1)=.5*OVEL(J,2)
48: 12 PVEL(J,1)=.5*PVEL(J,2)
49: C POSITIONS 2 THRU NW, N=0 (UNREFLECTED)
50: DO 8 I=2,NW
51: OVEL(1,I)=0.
52: 8 PVEL(1,I)=0.
53: C POSITIONS (NW+1) THRU NT,N=0 (UNREFLECTED)
54: DO 13 I=1,NL
55: FI=(I-1)
56: DO 13 J=1,NW
57: FJ=(J-1)
58: K=(I-1)*NW
59: DL(K+J)=DELL*FI
60: W(K+J)=DELW*FJ
61: 13 D(K+J)=SQRT(DL(K+J)**2+W(K+J)**2 +.0001)
62: K=NW+1
63: DO 25 I=K,NT
64: FR=(DL(I)*TN-W(I)+.5*WD)/WD
65: IF(FR) 16,16,17
66: 16 FR=0.
67: 17 IF(FR-1.) 19,18,18
68: 18 FR=1.
69: 19 SAF=SQRT(1.+AEX/3.14159/D(I)**2)
70: SAF=DL(I)/D(I)*(1.-1./SAF)
71: IF(SAF-(1.-SO(IC-1)))20,20,21
72: 21 SAF=1.-SO(IC-1)
73: 20 OVEL(1,I)=FR*SAF/(1.-SO(IC-1))
74: PVEL(1,I)=(1.+SP(IC-1))*(A*OVEL(1,I)/(1.+SO(IC-1))+(1.-A)*SAF)
75: 25 CONTINUE
76: C MODIFY FOR BEND INTO ROOM CASES
77: IF(RMCASE) 100,51,150
78: C ENTERING UNREFLECTED MOVING TOWARDS DETECTORS
79: 100 K=NW+1
80: DO 102 I=K,NT,NW
81: OVEL(1,I)=0.
82: DO 102 J=2,NW

```



```

83: 102 OVEL(1,I+J-1)=2.*OVEL(1,I+J-1)
84: GO TO 51
85: C ENTERING UNREFLECTED MOVING TOWARDS WALL
86: 150 DO 152 I=1,NT
87: TEMP=DL(I)
88: DL(I)=W(I)+(WTH-WDP)
89: W(I)=TEMP
90: 152 D(I)=SQRT(DL(I)**2+W(I)**2+.0001)
91: TEMP=NW
92: NW=NL
93: NL=TEMP
94: K=NW+1
95: DO 154 I=K,NT
96: 154 OVEL(1,I)=0.
97: GO TO 32
98: C ONCE-REFLECTED COMPONENT (N=1)
99: 51 SPX=1.
100: RMX=0.
101: IF(TNR.GT.TN) GO TO 30
102: SPX=0.
103: RMX=1.
104: 30 NLHF=(NL/2+1)
105: DO 31 I=1,NLHF
106: KI=(I-1)*NW
107: KIR=(NL-I)*NW
108: DO 31 J=1,NW
109: TEMP=DL(J+KI)
110: DL(J+KI)=DL(J+KIR)
111: 31 DL(J+KIR)=TEMP
112: DO 34 I=1,NT
113: 34 D(I)=SQRT(DL(I)**2+W(I)**2+.0001)
114: K=NT-NW+1
115: DO 33 J=1,NT
116: IF(J.LT.K) GO TO 35
117: OVEL(2,J)=OVEL(1,J)*AL*2.*DL(J)/D(J)+RMX*(1.+SO(IC-1))*AL*RR/2.
118: PVEL(2,J)=PVEL(1,J)*AL*2.*DL(J)/D(J)+RMX*(1.+SP(IC-1))*AL*RR/2.
119: GO TO 33
120: 35 SAFF=SQRT(1.+AEX/3.1415926/D(J)**2)
121: JAFP=DL(J)/D(J)*(1.-1./SAFF)
122: OVEL(2,J)=(1.+SO(IC-1))*AL*(SPX*SAFF+RMX*RR)
123: PVEL(2,J)=(1.+SP(IC-1))*AL*(SPX*(A*SAFF/(1.+SO(IC-1)))+(1.-A)*RR)
124: * +RMX*RR)
125: 33 CONTINUE
126: GO TO 57
127: C RMCASE=1. (N=1)
128: 32 OVEL(2,1)=AL*(1.+SO(IC-1))
129: PVEL(2,1)=AL*(1.+SP(IC-1))
130: DO 36 J=2,NT
131: SAFF=SQRT(1.+AEX/3.1415926/D(J)**2)
132: JAFP=DL(J)/D(J)*(1.-1./SAFF)
133: OVEL(2,J)=AL*(1.+SO(IC-1))*SAFF
134: 36 PVEL(2,J)=AL*(1.+SP(IC-1))*SAFF
135: TEMP=NW
136: NW=NL
137: NL=TEMP
138: C RENORMALIZATION
139: 57 DO 55 I=1,NT
140: DO 55 J=1,21
141: IJ=I+J
142: IF(IJ.EQ.2) GO TO 55
143: OVEL(J,I)=OVEL(J,I)*OVEL(1,1)
144: PVEL(J,I)=PVEL(J,I)*PVEL(1,1)
145: 55 CONTINUE
146: C MATRIX GENERATION AND MULTIPLICATION(S)
147: DO 70 I=1,NT
148: DO 58 J=1,21
149: SVELP(J,I)=0.
150: 58 SVELO(J,I)=OVEL(1,I)*VELO(IC-1,J)+PVEL(1,I)*VELP(IC-1,J)
151: DO 65 IP=2,21
152: DO 65 J=2,IP
153: 65 SVELP(IP,I)=SVELP(IP,I)+OVEL(J,I)*VELO(IC-1,IP+1-J)+PVEL(J,I)*
154: * VELP(IC-1,IP+1-J)
155: 70 CONTINUE
156: IF(IPRINT.NE.4) GO TO 1000
157: 40 WRITE(6,1)
158: WRITE(6,2) IC,AEX,AEXR,WD,RMCASE,DELW,DELL,TN,TNR,A,RR,ALR
159: WRITE(6,1)
160: WRITE(6,3) (SVELO(J,1),J=1,21,2)
161: WRITE(6,3) (SVELO(J,2),J=1,21,2)
162: WRITE(6,3) (SVELO(J,NT),J=1,21,2)
163: WRITE(6,3) (OVEL(J,1),J=1,10)
164: WRITE(6,1)
165: WRITE(6,3) (SVELP(J,1),J=1,21,2)

```

```

166: WRITE(6,3) (SVELP(J,2),J=1,21,2)
167: WRITE(6,3) (SVELP(J,NT),J=1,21,2)
168: WRITE(6,3) (PVEL(J,1),J=1,10)
169: WRITE(6,1)
170: 1000 RETURN
171: END
172: SUBROUTINE DOSX(IC,SVELO,SVELP)
173: COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
174: X BEND(10,6),SGRUPS(23),DWTS(21), CRRNTO(9,10),RCRRNT(9,10)
175: COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
176: X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
177: COMMON RCOSO,RCOSP,RCOS,DGRUPS(23), VELO(9,21),VELP(9,21)
178: X VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
179: COMMON IDOSE,HNDRSN(23),SEPI(23), STH(23),SINR(23),THMLN(23)
180: X,ENGAM(21),DIMEN,SCOBLT(23), AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
181: DIMENSION SVELO(22,20),SVELP(22,20),PRAT(23,2),ORAT(20,2),
182: * DRAT(20,2),SPCTRM(23),DOSE(20),DFLXN(21)
183: 1 FORMAT(1H )
184: 2 FORMAT(5H IEL=,I2,5H IC=,I2)
185: 3 FORMAT(8H DIR/ICD)
186: 4 FORMAT(8H REF/ICD)
187: 5 FORMAT(8H TOT/ICD)
188: 6 FORMAT(8H DIR/IFD)
189: 7 FORMAT(8H REF/IFD)
190: 8 FORMAT(30H DOSE/ENTRANCE DOSE W(FT)=,F7.2,9H L(FT)=,F7.2)
191: 10 FORMAT(9(1PE12.4))
192: 11 FORMAT(8H RCOSIN=,1PE10.3,7H CDIN=,1PE10.3,7H DIN=,1PE10.3,
193: X7H EXF=,1PE10.3)
194: 12 FORMAT(55H) MEG,N SOURCE SPCTRM DFLXN VELO VELP )
195: 13 FORMAT(5H DOSE)
196: 14 FORMAT(55H VEL(N)/IC=1 2 3 4 5 )
197: 15 FORMAT(I5,7(1PE10.3))
198: 16 FORMAT(5H SO= ,7(1PE10.3))
199: 17 FORMAT(5H SP= ,7(1PE10.3))
200: RCOSIN=2./(.1.+COF(1))
201: 20 CDIN=DWTS(1)
202: IF(CDIN.EQ.0.) CDIN=1.
203: NW=(ECC(IC)+1.01)
204: NL=(DELR(IC)+1.01)
205: NT=NW*NL
206: NMAX=21
207: IF(IDOSE.NE.5) GO TO 122
208: NMAX=10
209: 122 DO 100 MEG=1,23
210: SPCTRM(MEG)=0.
211: U=1.
212: DO 100 N=1,NMAX
213: SPCTRM(MEG)=SPCTRM(MEG)+(SVELO(N,1)+SVELP(N,1))*SWTS(MEG,N)/U
214: 100 U=U*.696
215: C
216: DO 50 I=1,NT
217: ORAT(I,1)=0.
218: PRAT(I,1)=0.
219: U=1.
220: DO 40 NREF=1,NMAX
221: ORAT(I,1)=ORAT(I,1)+SVELO(NREF,I)*DWTS(NREF)/U
222: PRAT(I,1)=PRAT(I,1)+SVELP(NREF,I)*DWTS(NREF)/U
223: DFLXN(NREF)=(SVELP(NREF,1)+SVELO(NREF,1))*DWTS(NREF)/U
224: 40 U=U*.696
225: DRAT(I,1)=PRAT(I,1)+ORAT(I,1)
226: DOSE(I)=DRAT(I,1)
227: 50 CONTINUE
228: C
229: IF(NEXT(1)) 101,101,102
230: 102 ZD=DELR(1)
231: EXF=(.6715+ZD*(.924+.565*ZD))/(.0685+ZD*(.5445+.137*ZD))
232: CDIN=CDIN*EXP(ZD*EXF)
233: 101 DIN=CDIN*RCOSIN
234: C
235: DO 120 I=1,NT
236: ORAT(I,2)=ORAT(I,1)/DIN
237: PRAT(I,2)=PRAT(I,1)/DIN
238: DRAT(I,2)=DRAT(I,1)/DIN
239: ORAT(I,1)=ORAT(I,1)/CDIN
240: PRAT(I,1)=PRAT(I,1)/CDIN
241: 120 DRAT(I,1)=DRAT(I,1)/CDIN
242: IF(IPRINT.EQ.0) GO TO 1000
243: IF(IPRINT.LE.2) GO TO 128
244: WRITE(6,2) IEL,IC
245: 105 WRITE(6,1)
246: WRITE(6,12)
247: JC=IC-2

```

```

248: DO 109 I=1,21
249: WRITE(6,15) I, SGRUPS(I), SPCTRM(I), DFLXN(I), VELO(JC, I), VELP(JC, I)
250: 109 CONTINUE
251: MEG=22
252: WRITE(6,15) MEG, SGRUPS(MEG), SPCTRM(MEG)
253: MEG=23
254: WRITE(6,15) MEG, SGRUPS(MEG), SPCTRM(MEG)
255: WRITE(6,1)
256: WRITE(6,1)
257: WRITE(6,14)
258: DO 113 N=1,21
259: WRITE(6,15) N, (VEL(ID, N), ID=1, JC)
260: 113 CONTINUE
261: WRITE(6,16) (SO(ID), ID=1, JC)
262: WRITE(6,17) (SP(ID), ID=1, JC)
263: WRITE(6,1)
264: IF(IPRINT.LT.4) GO TO 128
265: WRITE(6,11) RCOSIN, CDIN, DIN, EXF
266: WRITE(6,3)
267: WRITE(6,10) (ORAT(I,1), I=1, NT)
268: WRITE(6,4)
269: WRITE(6,10) (PRAT(I,1), I=1, NT)
270: WRITE(6,1)
271: WRITE(6,5)
272: DO 125 J=1, NW
273: WRITE(6,10) (DRAT(I+J-1,1), I=1, NT, NW)
274: 125 CONTINUE
275: WRITE(6,6)
276: WRITE(6,10) (ORAT(I,2), I=1, NT)
277: WRITE(6,7)
278: WRITE(6,10) (PRAT(I,2), I=1, NT)
279: WRITE(6,1)
280: 128 WRITE(6,8) COF(IC), DELR(IC-1)
281: WRITE(6,1)
282: DO 130 J=1, NW
283: WRITE(6,10) (DRAT(I+J-1,2), I=1, NT, NW)
284: 130 CONTINUE
285: WRITE(6,1)
286: WRITE(6,13)
287: DO 135 J=1, NW
288: WRITE(6,10) (DOSE(I+J-1), I=1, NT, NW)
289: 135 CONTINUE
290: 1000 RETURN
291: END
292: 'L'0
293: SUBROUTINE DOSE(IC)
294: COMMON DIAMS(18), DUCT0(3,17,6), DUCT1(3,17,6), DUCT2(3,17,6), IEL, LP,
295: X BEND(10,6), SGRUPS(23), DWTS(21), CRRNTO(9,10), RCRRT(9,10)
296: COMMON RVCT(21), RMTX(21,21), SO(9), SP(9), DZRO(9), NEXT(9), COF(9),
297: X ECC(9), DELR(9), SNYNEU(23), SINVAL(10), CO1, CO2, VEL(9,21), IPRINT
298: COMMON RCOSO, RCOSP, RCOS, DGRUPS(23), VELO(9,21), VELP(9,21)
299: X ,VEL1(5), SCALE, SWTS(23,21), WTS(23,21)
300: COMMON IDOSE, HNDRSN(23), SEPI(23), STH(23), SINR(23), THMLN(23)
301: X, ENGAM(21), DIMEN, SCOBLT(23), AWTS(6,5), SGAM(23), DGAM(23), AMOD(5,5)
302: DIMENSION VELX(21), SPCTRM(23), DFLXN(21), DELO(9), DELP(9),
303: X DOUT(9), REDUXT(9), REDUXD(9)
304: 1 FORMAT(1H )
305: 2 FORMAT(1H1)
306: 3 FORMAT(I5.7(1PE11.3))
307: 4 FORMAT(55H IC DELP DELO DELX RCOSO RCOSP RCOSX )
308: 5 FORMAT(55H NEXT(JC) CDIN CDOUT DIN DOUT REDUXT(JC) REDUXC )
309: 7 FORMAT(9H NEXT(I)=,I2,7H DOSE=,1PE10.3,14H REDUX FACTR=,1PE10.3)
310: 8 FORMAT(I5.5(1PE10.3))
311: 9 FORMAT(55H MEG,N SOURCE SPCTRM DFLXN VELO VELP )
312: 10 FORMAT(I5.4(1PE10.3),I10.4(1PE10.3))
313: 11 FORMAT(55H VEL(N)/IC=1 2 3 4 5 )
314: 12 FORMAT(I5,10(1PE10.3))
315: 13 FORMAT(5H SO= ,10(1PE10.3))
316: 14 FORMAT(5H SP= ,10(1PE10.3))
317: 15 FORMAT(7H ARTO= ,1PE10.3)
318: RCOSO=2./(1.+SO(IC-1))
319: RCOSP=2./(1.+SP(IC-1))
320: RCOSIN=2./(1.+COF(1))
321: CALL RETRN(VELX,SX,IC)
322: RCOSX=2./(1.+SX)
323: CDIN=0.
324: JC=IC-1
325: DO 19 I=1, JC
326: DELO(I)=0.
327: 19 DELP(I)=0.
328: DELX=0.
329: CDR1=0.
330: NMAX=21

```

```

331:      IF(IDOSE.LT.5) GO TO 20
332:      NMAX=10
333:      20 CDIN=DWTS(1)
334:      IF(CDIN.EQ.0.) CDIN=1.
335:      U=1.
336:      CRD1=CRD1+VEL(IC-1,2)*DWTS(2)/U
337:      DO 100 NREF=1,NMAX
338:      DO 99 I=1,JC
339:      DELO(I)=DELO(I)+VELO(I,NREF)*DWTS(NREF)/U
340:      99 DELP(I)=DELP(I)+VELP(I,NREF)*DWTS(NREF)/U
341:      DELX=DELX+VELX(NREF)*DWTS(NREF)/U
342:      100 U=U*.696
343:      DELX=DELX*SCALE
344:      DO 213 M=1,23
345:      SPCTRM(M)=0.
346:      U=1.
347:      V=1.
348:      DO 212 N=1,NMAX
349:      DFLXN(N)=(VELO(JC,N)*RCOSO+VELX(N)*RCOSX)*V/U
350:      SPCTRM(M)=SPCTRM(M)+(SWTS(M,N)*1.E10)*DFLXN(N)/V
351:      V=V*.75
352:      212 U=U*.696
353:      213 SPCTRM(M)=SPCTRM(M)/1.E10
354:      IF(NEXT(1)) 101,101,102
355:      C
356:      102 ZD=DELR(1)
357:      EXF=(.6715+ZD*(.924+.565*ZD))/(.0685+ZD*(.5445+.137*ZD))
358:      CDIN=CDIN*EXP(ZD*EXF)
359:      C
360:      101 DIN=CDIN*RCOSIN
361:      CDOUT=DELO(JC)+DELX
362:      DO 103 I=1,JC
363:      DOUT(I)=DELO(I)*2./(1.+SO(I))+DELP(I)*2./(1.+SP(I))
364:      REDUXT(I)=DOUT(I)/CDIN
365:      103 REDUXD(I)=DOUT(I)/DIN
366:      REDUXC=CDOUT/CDIN
367:      IF(IPRINT.EQ.0) GO TO 1000
368:      IF(IPRINT.EQ.1)GO TO 106
369:      DO 104 I=1,JC
370:      WRITE(6,7) NEXT(I),DOUT(I),REDUXD(I)
371:      104 CONTINUE
372:      GO TO 107
373:      106 WRITE(6,7) NEXT(JC),DOUT(JC),REDUXD(JC)
374:      107 IF(NEXT(JC+2).EQ.8) GO TO 1000
375:      WRITE(6,1)
376:      IF(IPRINT.GT.2) GO TO 105
377:      GO TO 1000
378:      105 WRITE(6,1)
379:      WRITE(6,15) SCALE
380:      WRITE(6,9)
381:      DO 109 I=1,21
382:      WRITE(6,8) I,SGRUPS(I),SPCTRM(I),DFLXN(I),VELO(JC,I),VELP(JC,I)
383:      109 CONTINUE
384:      MEG=22
385:      WRITE(6,8) MEG,SGRUPS(MEG),SPCTRM(MEG)
386:      MEG=23
387:      WRITE(6,8) MEG,SGRUPS(MEG),SPCTRM(MEG)
388:      WRITE(6,1)
389:      WRITE(6,11)
390:      DO 113 N=1,21
391:      WRITE(6,12) N,(VEL(ID,N),ID=1,JC)
392:      113 CONTINUE
393:      WRITE(6,13) (SO(ID),ID=1,JC)
394:      WRITE(6,14) (SP(ID),ID=1,JC)
395:      110 WRITE(6,1)
396:      IF(IPRINT.LE.3)GO TO 1000
397:      WRITE(6,4)
398:      WRITE(6,3) IC,DELP,DELO,DELX,RCOSO,RCOSP,RCOSX
399:      WRITE(6,5)
400:      WRITE(6,3) NEXT(JC),CDIN,CDOUT,DIN,DOUT(JC),REDUXT(JC),REDUXC
401:      1000 RETURN
402:      END

```

\*



```

1:*   SUBROUTINE RETRN(VELX,SX,IC)
2:   COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,
3:   X BEND(10,6),SGRUPS(23),DWTS(21),CRRNTO(9,10),RCRRNT(9,10)
4:   COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),
5:   X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT
6:   COMMON RCOSO,RCOSP,RCOS,DGRUPS(23),VELO(9,21),VELP(9,21)
7:   X VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)
8:   COMMON IDOSE,HNDRSN(23),SEPI(23),STH(23),SINR(23),THMLN(23)
9:   X ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)
10:  DIMENSION VELX(21),VELPX(21),SVELP(21),SVELO(21),XVELO(21),
11:  X XVELP(21),SVEL(21)
12: C
13: 1  FORMAT(1H )
14: 2  FORMAT(I3,11(1PE10.2))
15: 3  FORMAT(I3,7E13.3)
16:  KSUP=DZRO(IC)
17:  DO 8 IR=1,21
18:  SVEL(IR)=VEL(IC-1,IR)
19:  VELPX(IR)=VELP(IC-1,IR)
20: 8  VELX(IR)=VELP(IC-1,IR)
21:  SX=SP(IC-1)
22:  SXP=SP(IC-1)
23:  IF(KSUP) 50,50,10
24: 10 RDLNTH=COF(IC)
25:  IF(RDLNTH.GT.5.) GO TO 50
26:  DLNGTH=COF(IC-1)
27:  IF(RDLNTH.LT.DLNGTH+1.) GO TO 50
28:  IF(RDLNTH.GT.DLNGTH+5.) GO TO 50
29:  TOTO=0.
30:  TOTP=0.
31:  DO 12 NREF=1,21
32:  TOTO=TOTO+VELO(IC-1,NREF)
33: 12 TOTP=TOTP+VELP(IC-1,NREF)
34:  RC=(TOTO*RCOSO+TOTP*RCOSP)/(TOTO+TOTP)
35:  IF(RC-1.)13,13,14
36: 13 RC=1.01
37: C
38: 14 STOTO=0.
39:  STOTP=0.
40:  XTOTO=0.
41:  XTOTP=0.
42:  SSP=0.
43:  XSP=0.
44:  XSO=0.
45:  SO(IC+1)=0.
46:  SP(IC+1)=0.
47: 26 DO 27 IR=1,21
48:  RVCT(IR)=VEL(IC-2,IR)
49:  SVELO(IR)=VELO(IC-1,IR)
50: 27 SVELP(IR)=VELP(IC-1,IR)
51:  SSO=SO(IC-1)
52:  SSP=SP(IC-1)
53:  IEL=NEXT(IC-1)
54:  LP=2
55:  GO TO (100,110,300,500,500),IEL
56: 100 CALL DCTISO(RDLNTH,IC-1)
57:  GO TO 30
58: 110 CALL DCTSLT(RDLNTH,IC-1)
59:  GO TO 30
60: 300 DLNGTH=0.
61: 500 IF(IEL.GE.4) CALL BTOD(RDLNTH,IC-1)
62: C
63: 30 RISO=RDLNTH-DLNGTH
64:  RSLT=RISO-.75
65:  LP=3
66:  RVCT(1)=0.
67:  DO 35 IR=1,20
68: 35 RVCT(IR+1)=0.53*VEL(IC-1,IR)
69:  CALL DCTISO(RISO,IC)
70:  DO 40 IR=1,21
71:  XVELO(IR)=VELO(IC,IR)
72:  XVELP(IR)=VELP(IC,IR)
73:  IF(IR.GT.20) GO TO 41
74: 40 VELO(IC-1,IR+1)=.06*VEL(IC-1,IR)
75: 41 XSO=SO(IC)
76:  XSP=SP(IC)
77:  SO(IC-1)=0.
78:  CALL DCTSLT(RSLT,IC)
79: C
80:  DO 45 IR=1,21
81:  VELO(IC-1,IR)=SVELO(IR)
82:  VELP(IC-1,IR)=SVELP(IR)

```



```

83:      VEL(IC-1,IR)=SVEL(IR)
84:      VELX(IR)=XVELO(IR)+XVELP(IR)+VELO(IC,IR)+VELP(IC,IR)+VELPX(IR)
85:      XTOTO=XTOTO+XVELO(IR)
86:      XTOTP=XTOTP+XVELP(IR)
87:      STOTO=STOTO+VELO(IC,IR)
88:      45 STOTP=STOTP+VELP(IC,IR)
89:      SO(IC-1)=SSO
90:      SP(IC-1)=SSP
91:      TOTX=TOTP+STOTO+STOTP+XTOTO+XTOTP
92:      RCOS=2.*(TOTP/(1.+SXP)+XTOTO/(1.+XSO)+XTOTP/(1.+XSP)
93:      X +STOTO/(1.+SO(IC))+STOTP/(1.+SP(IC)))/TOTX
94:      SX=2./RCOS-1.
95: C
96:      IF(IPRINT.NE.4) GO TO 50
97:      47 WRITE(6,1)
98:      WRITE(6,2) IEL,(XVELO(IR),IR=1,21,2)
99:      WRITE(6,2) IC,(VELPX(IR),IR=1,21,2)
100:     WRITE(6,2) IC,(XVELP(IR),IR=1,21,2)
101:     WRITE(6,2) IC,(VELX(IR),IR=1,21,2)
102:     WRITE(6,3) IC,SX,SSP,XSO,XSP,SO(IC),SP(IC),DLNGTH
103:     WRITE(6,3) IC,TOTP,XTOTO,XTOTP,STOTO,STOTP,TOTX,RDLNTH
104:     WRITE(6,1)
105:     50 RETURN
106:     END
107:     SUBROUTINE WTLISTS(JSTT,JXWT,IOX,NMAX,DWTS,SWTD,ENGAM,IDOSE,
108:     X ISORS,LP)
109:     DIMENSION DWTS(21),ENGAM(21),SWTD(21),IXFILE(65,3),WTFILE(65,21,3)
110:     4 FORMAT(3I5,3I12)
111:     5 FORMAT(3I5)
112:     8 FORMAT(5(1PE12.4))
113: C - - - -SETS UP AND READS PRECALCULATED WEIGHTS FROM A FILE (WTSFILE)
114:     ITOT=65
115:     15 IF(IOX) 20,17,100
116:     17 IF(IXALT) 80,1000,50
117: C (MAIN PROGRAM: IOX=q1)
118: C - - - - SET UP 3 INDEX FILES WITH THE INTERNAL WTFILES
119:     20 DO 40 I=1,ITOT
120:         READ(7,5) (IXFILE(I,J),J=1,3)
121:         DO 40 IX=1,21
122:             READ(7,8) (WTFILE(I,IX,J),J=1,3)
123:         40 CONTINUE
124:         IOX=0
125:         IXALT=1
126:         REWIND 7
127: C - - WRITE(4,4) (IXFILE(12,J), J=1,3)
128: C - - WRITE(4,8) (WTFILE(12,2,J), J=1,3)
129:         GO TO 1000
130: C - - - - WRITE BACK THE INFORMATION INTO PREWTS
131:     100 DO 110 I=1,ITOT
132:         WRITE(7,5) (IXFILE(I,J),J=1,3)
133:         DO 110 NX=1,21
134:             WRITE(7,8) (WTFILE(I,NX,J),J=1,3)
135:         110 CONTINUE
136:         REWIND 7
137:         GO TO 1000
138: C (IOX VALUE OUTPUT DOESN'T MATTER)
139: C - - - - NEXT EITHER READ THE OLD WEIGHTS OR STORE NEW ONES
140: C (JSTT=q1, CALC THEN WRITE; JSTT=0=ISTT, READ THEN SKIP)
141:     50 IXALT=-IXALT
142:     JXWT=IDOSE+5*ISORS+20*LP-25
143:     IF(JSTT) 1000,51,1000
144:     51 JSTT=IXFILE(JXWT,2)
145:     IF(JSTT.NE.0) GO TO 1000
146:     52 JMAX=IXFILE(JXWT,3)
147:     DO 55 NX=1,JMAX
148:         DWTS(NX)=WTFILE(JXWT,NX,1)
149:         SWTD(NX)=WTFILE(JXWT,NX,2)
150:     55 ENGAM(NX)=WTFILE(JXWT,NX,3)
151:     NMAX=JMAX
152:     GO TO 1000
153: C - - - - WRITE WEIGTHS JUST COMPUTED
154:     80 IXALT=-IXALT
155:     IF(JSTT.EQ.0) GO TO 1000
156:     IXFILE(JXWT,2)=0
157:     IXFILE(JXWT,3)=NMAX
158:     DO 85 NX=1,NMAX
159:         WTFILE(JXWT,NX,1)=DWTS(NX)
160:         WTFILE(JXWT,NX,2)=SWTD(NX)
161:         WTFILE(JXWT,NX,3)=ENGAM(NX)
162:     85 CONTINUE
163:     1000 RETURN
164:     END
165:     SUBROUTINE ENTDAT

```

```
166: COMMON DIAMS(18),DUCTO(3,17,6),DUCT1(3,17,6),DUCT2(3,17,6),IEL,LP,  
167: X BEND(10,6),SGRUPS(23),DWT(21), CRRNTO(9,10),RCRRNT(9,10)  
168: COMMON RVCT(21),RMTX(21,21),SO(9),SP(9),DZRO(9),NEXT(9),COF(9),  
169: X ECC(9),DELR(9),SNYNEU(23),SINVAL(10),CO1,CO2,VEL(9,21),IPRINT  
170: COMMON RCOSO,RCOSP,RCOS,DGRUPS(23), VELO(9,21),VELP(9,21)  
171: X ,VEL1(5),SCALE,SWTS(23,21),WTSD(23,21)  
172: COMMON IDOSE,HNDRSN(23),SEPI(23), STH(23),SINR(23),THMLN(23)  
173: X,ENGAM(21),DIMEN,SCOBLT(23),AWTS(6,5),SGAM(23),DGAM(23),AMOD(5,5)  
174: DIMENSION TPX1(17), TPX2(17), TPX3(51), TPX4(51), TPX5(17),  
175: X TPX6(51), TPX7(17), TPX8(51), TPX9(51), TP10(17), TP11(51),  
176: X TP12(17), TP13(51), TP14(51), TP15(17), TP16(51), TP17( 8),  
177: X TP18( 8), TP19( 8), TP20( 8), TP21( 8), TP22( 8), TP23(23),  
178: X TP24(23), TP25(23), TP26(23), TP27(20), TP28(10), TP29(23),  
179: X TP30(23), TP31(23), TP32(23), TP33(23)  
180: C  
181: DATA TPX1 /0..5,1..1.5,2..2.5,3..3.5,4..4.5,5..5.5,  
182: X 6..6.5,7..7.5,8./  
183: DO 101 IL=1,17  
184: 101 DIAMS(IL)=TPX1(IL)  
185: C  
186: DATA TPX2 /1..413..205..111..0685..0481..0354,  
187: X .0292..0226..0191..0155..0127..0108..0091..0077..0065..0055/  
188: DATA TPX3 /.35..56..66..71..74..75..76,  
189: X .753..745..73..715..698..675..655..635..615..60..35..488..595,  
190: X .675..72..747..76..753..745..73..715..698..675..655..635..615,  
191: X .60..52..555..61..725..77..79..804..80..79..775..758..733..71,  
192: X .685..66..64..62/  
193: DATA TPX4 /.30..59..75..85..93..98,1,02,  
194: X 1.04,1.05,1.05,1.05,1.05,1.05,1.05,1.05,1.05,1.05,1.05,1.05,1.05,1.05,  
195: X .91..975,1,02,1,05,1,06,1,07,1,08,1,08,1,08,1,08,1,08,1,08,1,08,  
196: X 1,08,.74,.825,.90,.965,1,015,1,05,1,06,1,07,1,08,1,08,1,08,1,08,  
197: X 1,08,1,08,1,08,1,08,1,08/  
198: DATA TPX5 /1..517..285..154..097..063..046..037,  
199: X .028..027..019..016..013..0108..009..0074..0062/  
200: DATA TPX6 /.265..38..48..55..605..65..69,  
201: X .725..75..78..80..823..84..85..86..87..88..265..365..46..53..58,  
202: X .625..66..693..72..745..765..785..8..82..83..84..85..40..40..43,  
203: X .49..54..584..62..65..675..70..72..74..75..768..78..79..80/  
204: IDX=0  
205: DO 106 J=1,3  
206: DO 106 IL=1,17  
207: IDX=IDX+1  
208: IF(J-1) 106,104,105  
209: 104 DUCTO(1,IL,1)=TPX2(IDX)  
210: DUCTO(1,IL,4)=TPX5(IDX)  
211: 105 DUCTO(J,IL,2)=TPX3(IDX)  
212: DUCTO(J,IL,3)=TPX4(IDX)  
213: DUCTO(J,IL,5)=TPX6(IDX)  
214: 106 CONTINUE  
215: C  
216: DATA TPX7 /2..1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
217: X 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
218: DATA TPX8 /.35..62..52..51..51..512..518,  
219: X .523..533..548..562..58..6..622..645..665..685..35..59..515..51,  
220: X .51..512..518..523..533..548..562..58..6..622..645..665..685,  
221: X .615..56..52..51..51..512..518..523..533..548..562..58..6..622,  
222: X .645..665..685/  
223: DATA TPX9 /.3..61..83..945,1,00,1,04,1,06,  
224: X 1.065,1.067,1.064,1.06,1.04,1.02,1.00,.98,.96,.94,.3,.68,.865,  
225: X .965,1.025,1.06,1.08,1.084,1.08,1.07,1.05,1.04,1.02,1.00,.98,.96,  
226: X .94,.68,.79,.905,1..1,07,1,11,1,125,1,13,1,12,1,11,1,1,1,07,1,05,  
227: X 1,03,1,01,.99,.97/  
228: DATA TP10 /.5..5..5..5..5..5..5..5..5..5,  
229: X .5..5..5..5..5/  
230: DATA TP11 /.35..48..58..64..69..735..766,  
231: X .79..81..83..843..85..86..866..872..878..884..33..44..533..59..64  
232: X .68..71..733..75..765..775..78..786..791..795..8..805..35..42,  
233: X .485..545..587..625..653..68..7..716..728..737..745..753..754,  
234: X .758..76/  
235: IDX=0  
236: DO 111 J=1,3  
237: DO 111 IL=1,17  
238: IDX=IDX+1  
239: IF(J-1) 111,109,110  
240: 109 DUCT1(1,IL,1)=TPX7(IDX)  
241: DUCT1(1,IL,4)=TP10(IDX)  
242: 110 DUCT1(J,IL,2)=TPX8(IDX)  
243: DUCT1(J,IL,3)=TPX9(IDX)  
244: DUCT1(J,IL,5)=TP11(IDX)  
245: 111 CONTINUE  
246: C  
247: DATA TP12 /2..1,517,1,124,1,1,1,1,1,1,1,1,1,1,1,1,
```

```

248: X 1..1..1..1..1..1..1..1/
249: DATA TP13 /.35..565..77..81..77..71..663.
250: X .63..606..6..61..62..635..657..68..7..72..35..49..685..78..76.
251: X .712..663..63..606..6..61..62..635..657..68..7..72..5..61..8.
252: X .825..804..74..685..657..64..64..643..655..67..69..715..745..764/
253: DATA TP14 /.3..56..72..845..94..1.01..1.04.
254: X 1.07..1.085..1.10..1.10..1.09..1.077..1.06..1.04..1.02..1.00..3..62..836.
255: X .91..97..1.01..1.04..1.07..1.085..1.10..1.10..1.09..1.077..1.06..1.04..1.02.
256: X 1.00..767..83..907..966..1.005..1.04..1.06..1.08..1.095..1.10..1.10.
257: X 1.09..1.077..1.06..1.04..1.02..1.00/
258: DATA TP15 /.5..5..5..5..5..5..5..5..5..5..5.
259: X .5..5..5..5..5/
260: DATA TP16 /.65..57..54..568..645..73.
261: X .78..78..78..78..78..78..78..78..78..78..78..78..78..78..78..485..456..468..513.
262: X .59..655..675..675..675..675..675..675..675..675..675..675..675..675..675.
263: X .41..41..43..476..536..584..607..613..62..62..62..62..62..62..62.
264: X .62..62/
265:   IDX=0
266:   DO 116 J=1,3
267:   DO 116 IL=1,17
268:   IDX=IDX+1
269:   IF(J-1) 116,114,115
270: 114 DUCT2(1,IL,1)=TP12(IDX)
271:   DUCT2(1,IL,4)=TP15(IDX)
272: 115 DUCT2(J,IL,2)=TP13(IDX)
273:   DUCT2(J,IL,3)=TP14(IDX)
274:   DUCT2(J,IL,5)=TP16(IDX)
275: 116 CONTINUE
276: C
277:   DO 21 K=1,4,3
278:   DO 20 J=2,3
279:   DO 20 IL=1,17
280:   DUCTO(J,IL,K)=DUCTO(1,IL,K)
281:   DUCT1(J,IL,K)=DUCT1(1,IL,K)
282:   DUCT2(J,IL,K)=DUCT2(1,IL,K)
283: 20 CONTINUE
284: 21 CONTINUE
285: C
286:   DATA TP17 /-.4..-2.0..0.2..0.4..0.6..0.8..1./
287:   DATA TP18 /.152..169..189..213..243..290..361..450/
288:   DATA TP19 /.241..248..257..266..275..277..279..293/
289:   DATA TP20 /.41..41..41..41..41..41..41..41/
290:   DATA TP21 /.988..983..980..1.105..955..9487..1.039..1./
291:   DATA TP22 /.5..5..5..5..5..5..5..5/
292:   DO 22 IL=1,8
293:   SINVAL(IL)=TP17(IL)
294:   BEND(IL,1)=TP18(IL)
295:   BEND(IL,2)=TP19(IL)
296:   BEND(IL,3)=TP20(IL)
297:   BEND(IL,4)=TP21(IL)
298:   BEND(IL,5)=TP22(IL)
299: 22 CONTINUE
300: C
301:   DATA TP23 /0..0..00215..08..1405..1341..2673..2646..0944
302:   * .3038..443..3922..4468..2464..235..2107..2099..1492..1192..1497.
303:   * .1202..0957..0./
304:   DATA TP24 /0..0..0..0..0..0..0..0..0..0..0..0..0..0.
305:   * 0..0..0..0..0..0..0..0..0..0..1./
306:   DATA TP25 /1.898E+6..2.204E+6..3.777E+6..7.885E+6..1.05E+7.
307:   * 1.023E+7..1.027E+7..1.351E+7..5.54E+6..2.316E+7..4.544E+7..1.054E+8.
308:   * 2.215E+8..2.E+8..1.74E+8..1.549E+8..1.538E+8..1.071E+8..8.454E+7.
309:   * 1.009E+8..7.236E+7..6.108E+7..5.411E+7/
310:   DATA TP26 /2.326E+8..7.325E+7..8.596E+7..4.877E+7..5.477E+7.
311:   * 4.258E+7..5.011E+7..6.085E+7..4.762E+7..6.542E+7..7.11E+7..7.11E+7.
312:   * 9.493E+7..1.813E+8..5.777E+8..8.534E+8..1.398E+8..6*0./
313:   DATA TP33 /0..0..0..0..0..0..0..0..0..1.0..15*0./
314:   DO 26 M=1,23
315:   SEPI(M)=TP23(M)
316:   STH(M)=TP24(M)
317:   SINR(M)=TP25(M)
318:   SGAM(M)=TP26(M)
319:   SCOBLT(M)=TP33(M)
320: 26 CONTINUE
321: C
322:   DATA TP27 /.4359..324..1566..0722..0113..0..1..
323:   * 3*0..1..25..25..15..0..1..3*0./
324:   DATA TP28 /.9..65..4..15..0..1..9..65..4..15/
325:   IDX=0
326:   DO 28 I=1,4
327:   DO 28 J=1,5
328:   IDX=IDX+1
329:   AWT5(I,J)=TP27(IDX)
330:   IF(I-2) 27,27,28

```

```
331: 27 AMOD(I, J)=TP28 (IDX)
332: 28 CONTINUE
333: C
334: DATA TP29 /7.E-9,7.E-9,7.08E-9,6.72E-9,6.03E-9,
335: *5.43E-9,4.83E-9,4.48E-9,4.33E-9,4.23E-9,3.96E-9,3.3E-9,1.73E-9,
336: * 7.E-10,6.5E-10,6.07E-10,6.72E-10,5.35E-10,3.88E-10,3.42E-10,
337: * 3.27E-10,3.22E-10,3.2E-10/
338: DATA TP30 /5.46E-9,5.13E-9,4.84E-9,4.61E-9,4.44E-9,
339: * 4.13E-9,4.01E-9,3.39E-9,3.15E-9,3.09E-9,2.64E-9,1.97E-9,1.12E-9,
340: * 0.229E-9,9*0./
341: DATA TP31 /22*0..1./
342: DATA TP32 /1.66E-9,1.46E-9,1.27E-9,1.08E-9,8.75E-10,
343: * 7.35E-10,6.44E-10,5.3E-10,4.45E-10,3.5E-10,2.56E-10,2.15E-10,
344: * 1.77E-10,1.22E-10,6.6E-11,3.9E-11,8.37E-11,6*0./
345: DO 32 M=1,23
346: SNYNEU(M)=TP29(M)
347: HNDRSN(M)=TP30(M)
348: THMLN(M)=TP31(M)
349: 32 DGAM(M)=TP32(M)
350: RETURN
351: END
352:
353:
```

\*



U.S. DEPT. OF COMM. <b>BIBLIOGRAPHIC DATA SHEET</b> (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 87-3534	2. Performing Organ. Report No.	3. Publication Date MARCH 1987
4. TITLE AND SUBTITLE <p style="text-align: center;">DCTDOS: NEUTRON AND GAMMA PENETRATION IN COMPOSITE DUCT SYSTEMS</p>			
5. AUTHOR(S) <p style="text-align: center;">L. V. Spencer</p>			
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		7. Contract/Grant No.	8. Type of Report & Period Covered
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)			
10. SUPPLEMENTARY NOTES  <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)  <p style="text-align: center;">This paper describes computer methods for estimating neutron and gamma ray fluence rate, dose, and even spectral features due to penetration through a series of duct segments. The procedure links together data for individual segments -- straight sections and bends -- in arbitrary combinations; and the resulting composite can include computations for a room at the end, if there is one. This particular method was developed for rapid estimates in the protection problems against nuclear weapons; but the concepts which are employed are more broadly applicable.</p>			
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) computer software; gamma-ray albedos; gamma-ray penetration; multiple gamma-ray albedos; multiple neutron albedos; neutron penetration; radiation in ducts; weapons gamma-ray shielding; weapons neutron shielding			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.  <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		14. NO. OF PRINTED PAGES <p style="text-align: center;">90</p>	15. Price <p style="text-align: center;">\$13.95</p>





