ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

ORNL/TM-9803

User's Manual for PUTZ: A Point-Kernel Photon Shielding Code

D. T. Ingersoll

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A04; Microfiche A01

ORNL/TM--9803

DE86 011360

ORNL/TM-9803
Distribution Category UC-86

Consolidated Fuel Reprocessing Program

USER'S MANUAL FOR PUTZ:

A POINT-KERNEL PHOTON SHIELDING CODE

D. T. Ingersoll

Engineering Physics and Mathematics Division

Manuscript Completed: February 1986
Date Published: May 1986

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
Martin Marietta Energy Systems, Inc.
Under Contract No. DE-AC05-840R21400
for the
Department of Energy

EBB

TABLE OF CONTENTS

USER'S MANUAL FOR PUTZ: A POINT-KERNEL PHOTON SHIELDING CODE

ABSTRACT

A photon shielding code based on the point-kernel method has been developed. The code, entitled PUTZ, allows the specification of three-dimensional source and body geometries using simple parallelepipeds and annular parallelepipeds, and permits the inclusion of multiple sources and multiple detectors in a single case. A flexible free-form type input format is used, and attenuation data and material buildup factors are contained within the code for selected materials. Versions of the code are available for running on an IBM mainframe computer and an IBM-PC microcomputer (or compatible). A comprehensive user's manual is provided, which describes the basic algorithms, input requirements, output listings, and major limitations.

1. INTRODUCTION

This report describes a newly developed gamma-ray transport code based on the point-kernel method. An initial reaction might be "Why does the world need yet another point-kernel code?" This is a fair question, since the list of documented point-kernel codes is lengthy and the number of undocumented codes is likely enormous. It might be boasted that the answer to this question lies in the superiority of the code's programming or the completeness of the code's features. But actually, the answer lies more in the individuality of the code, and mostly in the philosophy of radiation shield design analysis. Therefore, one needs to digress a bit in order to more soundly justify this code development effort.

1.1 Background

As the design of a system or a facility becomes more detailed and focussed, the nature of the analysis tools used to predict the performance of the design should also become more refined. Initially, the analysis is scoping in nature and is intended to establish general feasibility of the design and to efficiently evaluate potential design alternatives. This is contrasted with the final analysis which should be highly detailed and sufficiently accurate to assure proper performance within established design limits. In this approach, the desirable qualities of the analytic method used early in the design evolution are: (1) the method should be fast so that many parametric calculations can be performed economically, (2) the method should be flexible to accommodate the range of potential designs, and (3) it should be relatively easy to use in order to minimize setup time.

For radiation shielding applications, the first quality listed above usually implies the use of a kernel method. The method is relatively fast, and can yield reasonably accurate solutions for simple shielding problems. However, problems of even modest complexity often violate at least some of the assumptions implicit in the kernel equations and the associated data, and hence introduce uncertainty into the solution. Familiarity with the analytic assumptions and experience in using the method are important for successful application of kernel techniques. The method can also be sufficiently flexible, although this tends to be more a function of how the method is packaged into a working code.

The ease of use of a method is also mostly a function of code packaging. The "user friendliness" of a code means many things including: (1) the amount of input required, (2)

the user's understanding of the input and output quantities, (3) the convenience of the input format, and (4) the degree of special tailoring to the specific class of problems to be solved. When constructing a working computer code, the best solution to these collective considerations will be as varied as the number of code developers. Hence, it is the demand for a fast analytic method in a simple user-friendly package that has promoted the proliferation of kernel-based codes. This same demand has promoted the development of the PUTZ kernel code described below.

1.2 Code Description

The computer code described in this report is named PUTZ (Point-kernel Using Tank Zones). The code's distinguishing feature, reflected in the name, is that it allows for convenient specification of tank-type shield geometries. This feature results from the fact that the code was developed for use in nuclear fuel reprocessing applications, where the majority of the radiation sources are solutions contained within tanks or canisters. Notable features of PUTZ include:

- source and absorber bodies are specified using a simplified type of combinatorial geometry description
- absorber bodies can be conveniently specified as tank-type (annular) regions
- multiple sources and multiple detector points can be specified in a single case
- the maximum-allowed computer memory (for a given region request) is allocated at the time of execution
- the input system is the same as the input system used by more sophisticated radiation shielding codes
- attenuation and buildup data are built into the code for a selected set of materials
- special options permit the printing of data useful for checking input and diagnostics
- multiple cases can be stacked within a single job
- code versions for an IBM mainframe computer and an IBM microcomputer have been tested

Also, reasonable default values of the input are provided to help reduce the amount of input required for common cases. Although an attempt was made to make the code "user friendly," it is important that the user appreciate his responsibility in understanding the assumptions of the point-kernel method, and that he interpret the results accordingly.

1.3 Outline of User's Manual

Chapter 2 of this report describes the basic methodology of PUTZ and presents the more significant equations used in the code. The code's overall architecture is also described. The mass attenuation data and the coefficients used to compute the material buildup factors are also given in Chapter 2. Input requirements for PUTZ are described in Chapter 3, first briefly for the complete list of input parameters, and then in greater detail for many of the parameters. Chapter 4 presents several sample problems which demonstrate the various input options and output edits. Information pertaining to the access and use of the mainframe and micro-computer versions of PUTZ is discussed in Chapter 5. Finally, Chapter 6 discusses some concerns and cautions regarding the implementation of PUTZ, and also identifies some areas for future development. Although all sections provide useful information, familiarity with Chapter 3 is required and an awareness of the first section of Chapter 6 is recommended.

2. PUTZ CODE DESCRIPTION

2.1 Point-Kernel Theory and Putz Equations

The primary equations used by the PUTZ code are quadrature-type extensions to the basic point-kernel equation for the photon flux at a single point due to an isotropic, point source in an infinite uniform medium. This basic relation is given as[1]

$$\phi(R) = \int_E S(E) \frac{e^{-\mu(E)R}}{4\pi R^2} B(\mu(E)R)dE \qquad [1]$$

where

ϕ(R)	=	space-dependent photon flux
S(E)	=	source strength
μ(E)	=	total attenuation coefficient for medium
R	=	distance from source point to flux point
$B(\mu$(E)R)	=	buildup factor for medium

The factor $(4\pi R^2)^{-1}$ is the geometric attenuation factor for a point source, while the $e^{-\mu(E)R}$ factor gives the attenuation of the uncollided flux due to the material between the source and flux points. The $B(\mu(E)R)$ factor represents a correction to the uncollided flux to account for photon scattering in the medium.

For sources which are distributed in energy, the energy variable is treated in a multi-group approach, i.e. the integration over all energy is broken into a sum of integrals over discrete energy ranges (groups). The flux and cross section are assumed to be constant within a group, thus greatly simplifying the equations. This assumption is generally acceptable so long as a sufficient number of groups are used. Extended sources, such as line or volume sources, are treated similarly by dividing the source region into a spatial quadrature and representing the source within a spatial mesh as a point source. The discretized equation for the group photon flux at a single "detector" point is

$$\phi_{ij} = S_{ij} \sum_{k=1}^{N_i} \left[\frac{W_{kj} \cdot B_{kj} \cdot \exp\left(- \sum_{l=1}^{M_k} \mu_{lkj} d_{lk} \right)}{4\pi R_k^2} \right] \qquad [2]$$

where

ϕ_{ij} = photon flux for j^{th} energy group due to source i

S_{ij} = total source strength for j^{th} energy group of source i

N_i = number of quadrature points in source i

M_k = number of materials between k^{th} source cell (within source i) and detector

W_{kj} = weighting factor of k^{th} source cell for j^{th} energy group

B_{kj} = buildup factor for all materials between k^{th} source cell and detector for j^{th} energy group

μ_{lkj} = total attenuation coefficient for l^{th} material between k^{th} source cell and detector for j^{th} energy group

d_{lk} = thickness of l^{th} material between k^{th} source cell and detector

R_k = distance from k^{th} source cell to detector point

The total photon flux for NS sources and for NG energy groups is then given by

$$\phi = \sum_{i=1}^{NS} \sum_{j=1}^{NG} \phi_{ij} \qquad [3]$$

and the total dose rate (in silicon) is given by

$$D(\text{silicon}) = \sum_{i=1}^{NS} \sum_{j=1}^{NG} \phi_{ij}\, \mu_{aj} \frac{(E_j + E_{j-1})}{2} \qquad [4]$$

where

μ_{aj} = energy absorption mass attenuation coefficient for silicon for j^{th} energy group

E_j = upper energy for j^{th} energy group

E_{j-1} = lower energy for j^{th} energy group

The source cell weights, W_{kj} in Eq. 2, are normally computed using the following equation

$$W_{kj} = \frac{V_k}{\sum_{k=1}^{N} V_k} \qquad [5]$$

where V_k is the volume of the k^{th} source cell and N is the number of source cells within the source body. If a volume source is divided into a linear mesh (an input option), then each cell has the same volume and Eq. 5 reduces to $W_{kj} = 1/N$. This is always the case for line sources. Another input option allows the weighting factors to be modified to emphasize the source cells which are closest to the detector position. This approach takes advantage of the fact that the source cells nearest the detector will contribute more to the total detector response than those cells which are further from the detector and must penetrate more of the source body in order to reach the detector point. For this option, the weights are computed using

$$W_{kj} = \frac{e^{-\mu_j x_k} B(\mu_j x_k)}{\sum_{k=1}^{N} [e^{-\mu_j x_k} B(\mu_j x_k)]} \qquad [6]$$

where

μ_j = total attenuation coefficient of source material for j^{th} group

x_k = distance from k^{th} source cell to surface of source in direction of detector

$B(\mu_j x_k)$ = buildup factor

This equation is only applicable for volume sources containing attenuating material.

The buildup factors in the above equations are computed using a Taylor exponential function[2] to fit the dose buildup data. For a function argument of μr, the buildup factor is given by

$$B(\mu r) = Ae^{\alpha_1 \mu r} - (A-1)e^{-\alpha_2 \mu r} \qquad [7]$$

where the A, α_1, and α_2 are coefficients of the fitted function and are built into the PUTZ code for specific materials and energies. When the input energies are different than the energies at which the Taylor coefficients are defined, an interpolation is performed by first computing buildup factors at the nearest energies and then interpolating the logarithm of the buildup factors.

If only one material is traversed between the source and detector points, or if the input option is specified which forces only one material buildup to be used, then Eq. 7 is used directly to calculate the total buildup for the source-to-detector ray. However, if more than one material is traversed, then the proper means of calculating a composite buildup is not obvious. Generally the last material traversed will dominate the buildup effect, but this may not always be the case if the last material is relatively thin or if the preceding material has a higher atomic number.[3] Many empirical equations exist to compute composite buildups; however, no one formula is universally best.

The equation used in PUTZ for calculating composite buildups is a carry-over from PUTZ's predecessor, GHF.[4] The somewhat cumbersome relation is

$$B_{kj} = \frac{\sum_{l=1}^{M_k}\left[B((\mu r)_{kj})\cdot(1-e^{-\mu_{lkj}\cdot d_{lk}})\cdot\exp(-\sum_{n=l+1}^{M_k}\mu_{nkj}d_{nk})\right]}{\left(1-e^{-(\mu r)_{kj}}\right)} \qquad [8]$$

where

$l = 1$ represents the material nearest the source

$l = M_k$ represents the material nearest the detector

$(\mu r)_{kj}$ = total "attenuation distance" for the entire source-to-detector ray given by:

$$(\mu r)_{kj} = \sum_{l=1}^{M_k} \mu_{lkj}\, d_{lk} \qquad [9]$$

All other symbols have been defined previously. Because of the way the limits are defined for the right-most summation in the numerator of Eq. 8, the last term will have an initial value for the index that is greater than the final value. In this case, the code explicitly sets:

$$\exp\left[-\sum_{n=M_k+1}^{M_k}\mu_{nkj}\,d_{nk}\right]=1.0 \qquad [10]$$

There is probably not much more justification in using Eq. 8 than in using the buildup for the last material traversed or some other weighted combination of buildups. For problems of greater importance, it would be prudent to study the sensitivity of the dose response to different buildup assumptions.

2.2 Code Structure

The hierarchy of subroutine calls within the mainframe version of PUTZ is diagrammed in Fig. 1. Differences with the microcomputer version are minor and are discussed in Chapter 5. The MAIN program is a relatively small routine which simply allocates computer memory. Subroutine INPUT establishes the length of most data arrays, sets default values for the input data, performs all input functions via calls to subroutine FIDOS, and prints an edit of the input control arrays. The central routine is CNTRL, which controls the computational functions of PUTZ. CNTRL first defines several scratch arrays and prints an edit of the major input data arrays. The primary loop within PUTZ is then executed, which computes the source cell location, the source-to-detector ray trace, and the ray attenuation factors for each detector. The source cell location is actually computed within subroutine POINTL (for point or line sources) or POINTV (for volume sources). The ray trace is performed within subroutine CROSS, and the attenuation factors are calculated in ATB. Subroutine CNTRL completes the calculation of flux and dose results for each detector and then provides a summary edit of doses computed at all detector points. Program execution is then returned to MAIN for problem termination or the initiation of another case.

The attenuation data and buildup factor coefficients are contained in the BLOCK DATA routine. The mass attenuation data for the allowed 6 materials are listed in Table 1. Table 1 also includes the mass attenuation data for silicon energy absorption, which is used to compute the dose values according to Eq. 4. Table 2 lists the coefficients of the Taylor function used by PUTZ to compute the buildup factors for the 6 materials. All of the attenuation data and buildup factor coefficients were retained from the earlier GHF code,[4] which unfortunately does not further reference the source of the data. This is of some concern, and is an area for future improvement (see Chapter 6).

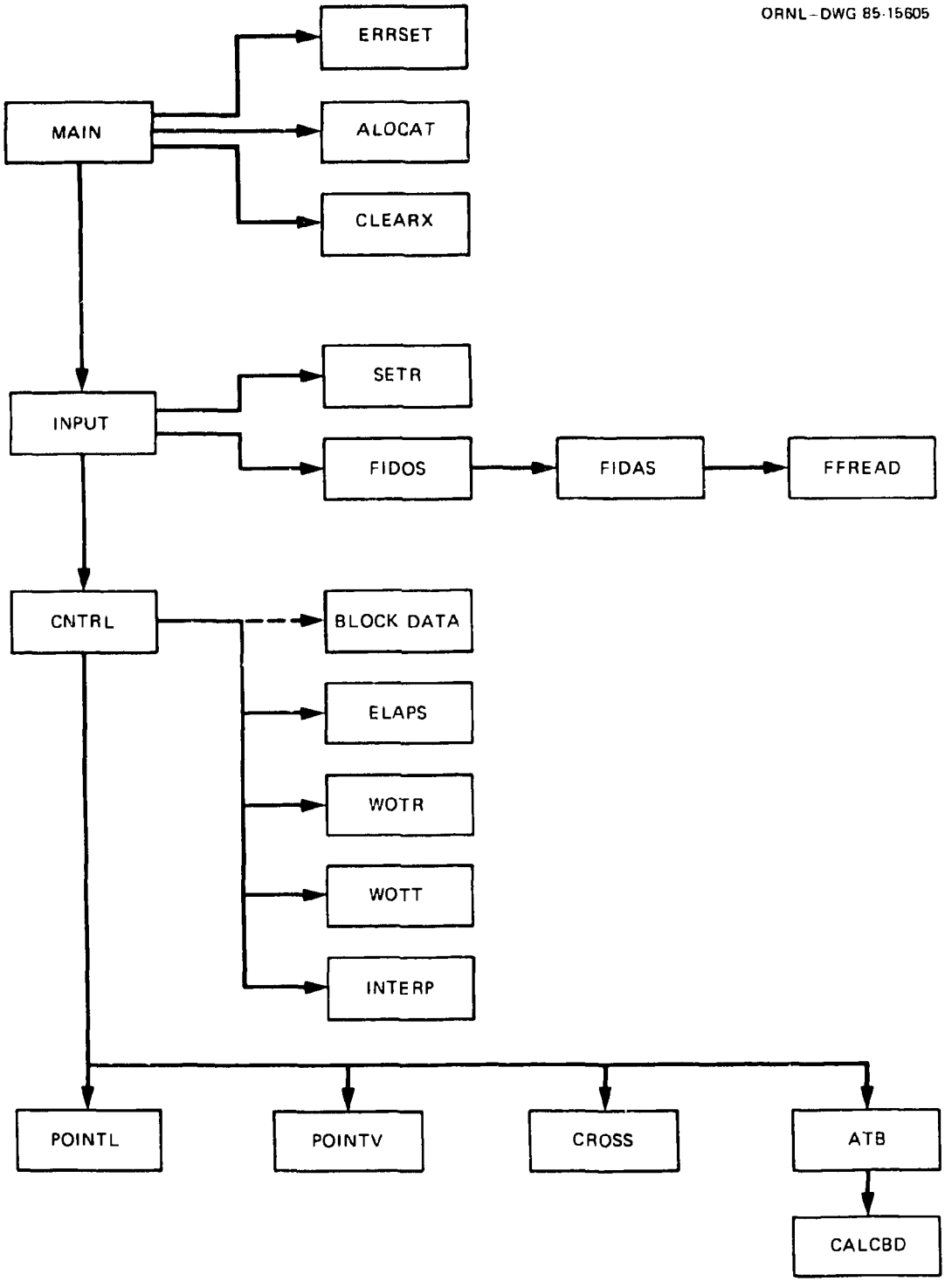

Fig. 1. Subroutine hierarchy for PUTZ code.

TABLE 1. Mass attenuation coefficients for materials available in PUTZ

	Mass attenuation coefficients (cm^2/g)						
Energy (MeV)	Air (1)	Water (2)	Concrete (3)	Iron (4)	Lead (5)	Uranium (6)	Silicon energy absorption
0.05	0.1877	0.2122	0.3306	1.8100	7.2340	10.4300	0.2287
0.07	0.1643	0.1838	0.2147	0.7469	4.0290	4.2810	0.0845
0.10	0.1503	0.1673	0.1663	0.3379	5.2830	1.7110	0.0440
0.13	0.1415	0.1574	0.1501	0.2405	3.1800	2.8650	0.0348
0.15	0.1340	0.1490	0.1384	0.1828	1.9090	2.4860	0.0308
0.20	0.1225	0.1361	0.1242	0.1387	0.9413	1.2200	0.0291
0.30	0.1064	0.1182	0.1068	0.1066	0.3764	0.4821	0.0294
0.40	0.0953	0.1059	0.0954	0.0921	0.2160	0.2711	0.0298
0.50	0.0870	0.0967	0.0870	0.0829	0.1501	0.1839	0.0298
0.60	0.0805	0.0894	0.0804	0.0761	0.1165	0.1394	0.0296
0.80	0.0707	0.0785	0.0706	0.0664	0.0837	0.0964	0.0289
1.00	0.0636	0.0706	0.0634	0.0595	0.0679	0.0760	0.0279
1.25	0.0568	0.0632	0.0567	0.0531	0.0566	0.0616	0.0267
1.50	0.0517	0.0575	0.0517	0.0486	0.0507	0.0543	0.0255
2.00	0.0444	0.0493	0.0446	0.0425	0.0447	0.0471	0.0237
3.00	0.0357	0.0396	0.0363	0.0361	0.0414	0.0435	0.0214
5.00	0.0275	0.0302	0.0287	0.0314	0.0421	0.0443	0.0194

TABLE 2. Constants for Taylor exponential representation of material buildup factors used in PUTZ

"A" PARAMETER

Energy (MeV)	Air	Water	Concrete	Iron	Lead	Uranium
0.05	7.445	7.445	3.923	1.379	1.081	1.055
0.06	6.639	6.639	5.083	1.663	1.097	1.115
0.08	5.603	5.603	18.710	2.214	1.125	1.195
0.10	5.000	5.000	39.160	2.745	1.150	1.238
0.15	4.472	4.472	175.800	4.134	1.174	1.175
0.20	4.353	4.353	142.400	5.139	1.253	1.192
0.30	5.432	5.432	129.100	5.698	1.657	1.468
0.40	6.129	6.129	135.900	6.030	1.996	1.712
0.50	6.561	6.561	104.100	6.221	2.287	1.935
0.60	6.651	6.651	101.600	6.252	2.571	2.195
0.80	6.701	6.701	101.300	6.238	3.001	2.601
1.00	6.641	6.641	101.500	6.160	3.288	2.884
1.50	6.105	6.105	32.810	5.775	3.559	3.213
2.00	5.689	5.689	39.310	5.407	3.520	3.239
3.00	5.014	5.014	13.760	4.686	2.785	2.639
4.00	4.538	4.538	14.220	4.107	2.169	2.094
5.00	4.173	4.173	9.268	3.622	1.661	1.626

"α_1" PARAMETER

Energy (MeV)	Air	Water	Concrete	Iron	Lead	Uranium
0.05	0.31480	0.31480	0.02300	0.00350	0.00040	0.00026
0.06	0.35140	0.35140	0.04230	0.00640	0.00042	0.00029
0.08	0.40100	0.40100	0.03770	0.01410	0.00050	0.00034
0.10	0.42660	0.42660	0.04820	0.02390	0.00061	0.00041
0.15	0.40470	0.40470	0.06060	0.05810	0.00110	0.00064
0.20	0.37480	0.37480	0.07400	0.08910	0.00160	0.00091
0.30	0.28610	0.28610	0.07580	0.11390	0.00250	0.00150
0.40	0.23130	0.23130	0.06870	0.12680	0.00370	0.00231
0.50	0.19400	0.19400	0.06390	0.13130	0.00530	0.00335
0.60	0.16740	0.16740	0.05750	0.12290	0.00800	0.00488
0.80	0.13100	0.13100	0.04650	0.10930	0.01390	0.00846
1.00	0.10750	0.10750	0.03850	0.09850	0.02000	0.01260
1.50	0.07600	0.07600	0.03520	0.07620	0.03300	0.02500
2.00	0.05860	0.05860	0.02470	0.06510	0.04650	0.03840
3.00	0.04120	0.04120	0.02850	0.05930	0.07710	0.06600
4.00	0.03250	0.03250	0.02220	0.05830	0.10600	0.09300
5.00	0.02760	0.02760	0.02670	0.05980	0.13260	0.11850

TABLE 2. Cont'd

"α_2" PARAMETER

Energy (MeV)	Air	Water	Concrete	Iron	Lead	Uranium
0.05	-0.21480	-0.21480	0.21260	0.09650	0.09960	0.09970
0.06	-0.25040	-0.25040	0.23260	0.09760	0.09960	0.09970
0.08	-0.30020	-0.30020	0.04500	0.09020	0.09950	0.09970
0.10	-0.32660	-0.32660	0.00150	0.07610	0.09940	0.09960
0.15	-0.30610	-0.30610	-0.04700	0.03320	0.09890	0.09940
0.20	-0.27480	-0.27480	-0.05640	0.01100	0.09840	0.09910
0.30	-0.19420	-0.19420	-0.05670	-0.00150	0.09840	0.09900
0.40	-0.13630	-0.13630	-0.05210	-0.01880	0.09690	0.09800
0.50	-0.09400	-0.09400	-0.04420	-0.03130	0.09470	0.09670
0.60	-0.06690	-0.06690	-0.03890	-0.00690	0.09190	0.09560
0.80	-0.02770	-0.02770	-0.03200	0.0	0.08600	0.09190
1.00	-0.00750	-0.00750	-0.02460	0.00150	0.08000	0.08740
1.50	0.00346	0.00346	-0.00170	0.01580	0.08080	0.08910
2.00	0.04140	0.04140	-0.00160	0.03490	0.05350	0.06160
3.00	0.05580	0.05580	0.02760	0.04220	0.00004	0.00045
4.00	0.06570	0.06570	0.02320	0.04310	-0.00580	-0.00210
5.00	0.07240	0.07240	0.03280	0.04020	-0.03260	-0.01850

3. INPUT DESCRIPTION

When establishing the input requirements for PUTZ, a compromise was made between the degree of "user friendliness," which tends to imply a minimum amount of input, and problem flexibility, which generally requires a large amount of input. The approach was taken to permit flexible problem specification via numerous input options, but to provide reasonable default values of the input to simplify the requirements for many common problems. In addition, the FIDO input system[5,6] was used to improve input convenience. A brief description of the FIDO input system is given below. A novice user should find this description sufficient to prepare an input file for PUTZ. The more experienced or adventuresome user may refer to Appendix A for a detailed description of the complete set of FIDO input options. These options can be used to greatly reduce the effort of preparing the input file.

3.1 Simplified FIDO Description

The FIDO input system is especially designed to allow easy entering or modifying of large amounts of data. The data are entered in units called "arrays," which are grouped into units called "blocks." A special delimiter is always required to signify the end of each block. An array need not be entered if it is not required as determined by a previous input option, or if the default values of the array are sufficient.

The most flexible input format is called "free-field" input. Each array is initiated with an array identifier which is a one or two digit number followed immediately by a "$$" or a "**". The number identifies the array and the "$$" signifies the entry of free-field integer-type data or the "**" signifies the entry of free-field real-type data. The integer- or real-type data pertaining to that array are then entered with virtually no regard to any format rules or column alignment excepting that:

1. only columns 1-72 can be used,
2. at least one blank must appear between each data entry,
3. a single data entry can not contain any embedded blanks and can not contain more than nine digits, and
4. a single data entry cannot span two cards.

Many other features exist in the FIDO system which permit such things as comments, easy repetition of data, skipping over unchanged data, and even interpolation of data values. These options are discussed in Appendix A. Some of the more common options are used in the sample problems described in Chapter 4.

3.2 PUTZ Input Arrays

The following title card and two data blocks of arrays are required in order to execute a single PUTZ case. Successive cases can be run by beginning with a new title card immediately following the second data block terminator of the preceding case. Default values of the input are given in [ ]'s.

PROBLEM TITLE CARD (80 Column Literal Data)

61$$ Array (10 entries):

[0]	NLS	Number of point and/or line sources.
[0]	NVS	Number of volume sources.
[0]	NABS	Number of absorber bodies (shields).
[1]	INPT	1 - Specify volume sources and absorbers by their center point coordinates and their X, Y, and Z widths, or 2 - Specify volume sources and absorbers by their minimum X, Y, and Z coordinates and their maximum coordinates.
[0]	NG	Number of energy groups in source spectrum.
[1]	NSP	Number of different source spectra.
[1]	NLPT	Maximum number of source cell points in any line source.
[27]	NVPT	Maximum number of source cell points in any volume source.
[1]	LOGM	1 - Use a linear (equal width) division of volume sources, or 2 - use a logarithmic division of volume sources.
[0]	IWT	0 - No effect, or 1 - use attenuation weighting of volume source cells.

62$$ Array (5 entries):

[0]	ND	Number of detector points.
[0]	IDAT	0 - No effect, or 1 - print edit of all attenuation coefficients and buildup factor parameters.
[-1]	IBUF	-1 - Use respective buildup factors for each different material to compute composite buildup, 0 - do not include buildup, or N - use buildup factors for material "N" to compute buildup in all materials.
[0]	IEDT	N - Print additional data useful for diagnostics. N = N1 + N2 + N3, where: N1 = 0 - no effect, or 1 - print volume source cell midpoints N2 = 0 - no effect, or 10 - print source-to-detector ray trace edit N3 = 0 - no effect, or 100 - print ray boundary crossing data
[1]	MATX	Material which is external to all bodies. (1 = Air)

63** Array (3 entries):

[0]	DENX	Density of material which is external to all bodies.
[1]	CONV	Factor which converts from the actual units used for body descriptors and detector coordinates to units of centimeters.
[1]	AGAM	Source normalization in terms of gammas/second corresponding to a unit source intensity.

T (First Data Block Terminator)

Remaining Arrays:

1** (7*(NLS+NVS+NABS) entries)

Enter 7 body descriptors for each line source, volume source, and/or absorber, in that order.

Line sources: X,Y,Z coordinates for first endpoint
X,Y,Z coordinates for second endpoint
tank zone thickness (must be 0.0)

Volume sources: X,Y,Z coordinates for center or minimum point
X,Y,Z widths or coordinates for maximum point
tank zone thickness (must be 0.0)

Absorbers: X,Y,Z coordinates for center or minimum point
X,Y,Z widths or coordinates for maximum point
tank zone thickness

2$$ (NVS+NABS entries)

Material number for each volume source and/or absorber.

3** (NVS+NABS entries)

Density of material for each volume source and/or absorber in units of g/cm^3.

4$$ (NLS+(3*NVS) entries) [1 for line sources; 3 for volume sources]

Number of source cell points in each line source and/or in the X, Y, and Z directions of each volume source.

5** (NLS+NVS entries) [all 1.0's]

Source intensity of each line and/or volume source.

6$$ (NLS+NVS entries) [all 1's]

Spectrum number to be used for each line and/or volume source.

7** (NG+1 entries)

Spectral energy boundaries (in MeV) in ascending order from the lower boundary of first group to the upper boundary of last group.

8** (NSP*NG entries) [all 1.0's]
Energy spectra entered by group for each spectrum. Spectra are normalized to 1.0 within PUTZ.

9** (3*ND entries)
X,Y,Z coordinates for each detector point in same units as body descriptors.

T (Second Data Block Terminator)

3.3 Detailed Explanation of Input

Expanded explanations of several of the input parameters and arrays are given below. The descriptions are given in the same order that the parameters are listed in the preceding section.

NLS (61$$ array): A point source is entered as a special type of line source. In this case, the two endpoint descriptors entered in the 1** array must coincide, and the number of source cell points entered in the 4$$ array for the corresponding source must be 1.

NVS,NABS (61$$ array): Volume sources and absorbers are input as simple rectangular parallelepipeds (RPP). Complex source or shield geometries may require several RPPs to describe the actual geometry. The parameter NVS gives the number of RPPs used to represent all volume sources, and NABS gives the number of RPPs used to construct all shield regions.

INPT (61$$ array): There are two ways to describe a RPP to be used as either a volume source or an absorber. When INPT=1, the X, Y, and Z coordinates of the center of the RPP are entered followed by the corresponding full width of the body in each direction. When INPT=2, the X, Y, and Z coordinates of the minimum and the maximum points of the RPP are input. Hence the value of INPT determines the meaning of the entries in the 1** array. For either INPT option, tank-type bodies are specified by their inside surface.

LOGM (61$$ array): With LOGM=1, each X, Y, and Z dimension of a volume source will be divided into equal-width segments forming a linearly spaced mesh. This option gives a uniform distribution of source cells within the volume source. The LOGM=2 option causes each direction to be divided into a logarithmic mesh oriented

such that the shorter segments are closer to the current detector. This option gives a non-uniform distribution of cells within the volume source with the greater density of cells being located on the detector side of the source. For a given direction, a linear mesh is used if the detector coordinate is between the minimum and maximum bounds of the volume source. A linear mesh is always used for line sources.

IWT (61$$ array): Each source cell within a line source is given equal weight. For a volume source, each cell is normally weighted by the relative volume of the cell (this yields equal weights if LOGM=1). However, if the material contained within a volume source is highly absorbing, then it may be advantageous to preferentially weight the source cells which are nearest the exterior detector. This is because particles emanating from the source cells furthest from the detector will be attenuated by the source material and will not contribute significantly to the detector response. IWT=2 provides this kind of biasing by weighting each cell with the relative amount of attenuation which occurs in the material between the cell and the surface of the source in the direction of the detector. Equations 5 and 6 in Chapter 2 show specifically how the weights are calculated.

IBUF (62$$ array): Three options exist to control how material buildup is computed. If IBUF=0, buildup factors of 1.0 are used (represents no buildup) and the resulting doses are due to only the uncollided flux. For IBUF=N (where "N" is a positive integer ranging from 1 to 6), the buildup factors for material "N" are used to compute the buildup in all materials. If IBUF=-1, then a complex empirical formula (Eq. 8 in Chapter 2) is used to compute buildup. The formula uses buildup factors for each material present in the source-to-detector ray and combines the factors in a weighted method which tends to emphasize the materials located nearest the detector.

IEDT (62$$ array): This parameter allows additional quantities to be printed, which can be useful for checking the correctness of the input specifications or for diagnosing a problem. The parameter is defined in such a way so as to allow three edit options to be independently specified. These options are:

a. Print the coordinates of the mid-point of each source cell. This is useful to assure that a volume source has been divided into the intended number of cells and the proper mesh.

b. Print several quantities which describe the ray trace from the source point to detector point. The following quantities are given for each body crossed by the ray:

IBD: input body number

DTB: distance from source to body

DIB: distance traveled within body

IRDR: order of occurance of body along ray path

After incorporating the gaps between bodies, the following quantities are given in order of occurrence along the ray:

MT: material number

DEN: material density

DIS: thickness of material

c. Print several arrays which describe the ray crossing of each source or absorber body. This information includes:

P: the minimum and maximum coordinate values of the body in the X, Y, and Z directions

D: the distance from the source to the point at which each of the six surfaces of the body are crossed

CORD: the X, Y, and Z coordinates of each of the six body surfaces crossed

IFF,IC: the initial number of acceptable surface crossings and the corresponding surface numbers

IFD,ICK: the final number of acceptable surface crossings and the corresponding surface numbers

All of the above options can be very helpful, especially for locating the cause of some difficulty. However, they should be used cautiously with an understanding of the volume of output which may be produced. To help reduce the amount of output, the last two edits described above are given only for the center source cell. (Note: This will correspond to the geometric center of the total source body only if an odd number of source points is specified.)

Body Descriptors (1 array)**: As discussed above, the source and absorber bodies can be entered in one of two ways as determined by the value of INPT. Line sources are input first and are always specified by the X, Y, and Z coordinates of one end point and then the X, Y, and Z coordinates of the other end point. A zero must also be entered as the tank shell thickness. For a point source (a special line source case), the coordinates for both end points should be the same.

Volume sources and absorbers can be input as the coordinates of the body midpoint followed by the full width of the body in the X, Y, and Z directions (INPT=1), or as the

coordinates of the minimum body corner and the coordinates of the maximum body corner (INPT=2). The seventh entry for each body specifies the thickness of the tank shell. This must be a zero for volume sources, and can be zero or some positive thickness for absorbers. All volume sources must be input before any absorbers.

Material Numbers (2$$ array): There are six allowable materials built into the PUTZ code, which are identified by integer numbers. They are:

1 = air	4 = iron
2 = water	5 = lead
3 = ordinary concrete	6 = uranium

The values are entered in the same order as the body descriptors (excluding line sources) given in the 1$$ array.

Source Cells (4$$ array): These entries specify the number of cells into which each source will be divided. The order of the entries must correspond with the descriptors given in the 1$$ array. Only one entry is required for each line source and three entries are required for each volume source. The default values for the number of cells are 1's for all line sources and 3's for each direction of all volume sources. Note: These values must be consistent with the maximum number of points specified as NLPT and NVPT in the 61$$ array.

Source Spectra (5, 6$$, and 8** arrays)**: Multiple source spectra can be input and assigned to any of the source bodies. The spectra are entered in the 8** array in the following manner: first group in first spectrum, second group in first spectrum, ..., last group in first spectrum, ..., last group in last spectrum. The spectra need not be normalized to unity since this is done within PUTZ. The default values are all 1.0's for the energy spectra. The number of the spectrum to be associated with a given source body should be entered in the 6$$ array in the same order as the source descriptors are given in the 1$$ array. The default spectrum numbers are all 1's. The total source strength for a given energy group in a given source body is computed by multiplying the AGAM factor (63** array) by the source body intensity (5** array) and by the normalized group spectrum value for that body (8** array).

3.4 Multiple Cases

Multiple PUTZ cases can be stacked and executed within a single job by entering a new title card and corresponding input data immediately following the second data block

terminator of the preceding case. There are, however, a few cautions and conveniences which should be noted when running multiple cases. First, the default values of the input parameters are set only prior to reading the input for the first case. Therefore, any default value which has been overridden in the first case will retain the new value until explicitly changed in a subsequent case. Also, if one case fails due to an input error, then subsequent cases may or may not be executed. They will be attempted if the error is detected by the FIDO input scanner (such as entering too few or too many parameters in an array), but will not be attempted if a system error is caused (such as a divide check).

As a convenience, the input arrays are not destroyed between successive cases. This can greatly simplify the input for a subsequent case if the input changes are relatively minor. Since the arrays in the second data block are variable in length, some care must be taken if the length of an array is altered in subsequent cases. The arrays are stored within the code in contiguous locations and in ascending order; therefore, if an array length is changed, all of the higher-numbered arrays are destroyed and must be reentered. For example, if one decided to run two cases in which the only difference was that a greater number of energy groups was used in the second case (thus increasing the length of the 7** and 8** arrays), then arrays 1-6 would not have to be entered in the second case but the 9** array would, even though the detector coordinates from the first case were still desired. The best rule is: "When in doubt, reenter all arrays."

3.5 Dimensional Units

Keeping track of the required dimensional units is always a nuisance, but important. A desirable goal is to deal entirely with metric units; however, some people still prefer to use English units. Draftsmen are frequently the "culprits" since invariably blueprints are specified with inches, feet, etc. In constructing geometry models, one could go to the trouble of converting the blueprint units to metric, but chances are that they will have to be converted back to the original units when the results are reported to the cognizant engineer. This constant conversion is inefficient and increases the likelihood of error. Therefore, provisions were made in PUTZ to allow English or arbitrary units to be used for the body descriptors and detector locations. Otherwise, metric units are required.

In summary, the units required by PUTZ are as follows:

1. Densities (DENX in the 63** array and all entries in the 3** array) are to be in units of g/cm^3.

2. Body descriptors (1** array) and detector coordinates (9** array) can be in any arbitrary units. However, the CONV entry in the 63** array must specify the factor which converts from these units to units of centimeters.

3. The actual units of AGAM (63** array) and the individual source intensities (5** array) are arbitrary, but their product must be in units of inverse seconds.

4. Energy group boundaries (7** array) must be in units of MeV.

5. Energy group spectra (8** array) must be unitless.

If this is all done properly, then the computed photon flux will be in units of $(cm^{2}*s)^{-1}$ and the dose rates will be in rads/hr.

4. SAMPLE PROBLEMS

This chapter lists several sample problems which are useful to demonstrate the features of PUTZ, and which are also helpful in assuring the proper operation of the code. The first sample is actually a collection of four simple cases stacked together. The second sample is a more complex problem with a few tank-type zones, and the third sample is a very large problem with many sources, shields, and detectors.

4.1 Sample 1 - Method Verification

The first sample problem is a collection of four simple cases stacked within a single job. The cases demonstrate how one can use default values and previously defined values to reduce input requirements. Also, the simplicity of the problems makes them relatively convenient for checking the computed doses with hand calculations in order to verify the basic equations in the code. The full input file for Sample 1 is listed in Fig. 2.

The first case models a point source in an infinite homogeneous medium of water. The source is located at the origin of the coordinate system, and the dose is computed at two detector points: one that is on the X-axis (10 cm from the source), and one that is at a random point in space. Seventeen energy groups are used for the source spectrum, which is assumed to be uniform in energy. The seventeen groups were deliberately chosen to correspond with the energy group structure of the attenuation and buildup coefficients which are built into PUTZ. This choice further simplifies the hand calculations used to verify the PUTZ results. The 2$$ and 3** arrays were not entered since they are not used for point or line sources, and the 4$$, 5**, 6$$, and 8** arrays were not entered since the default values were sufficient.

The second case differs from the first only in that material buildup is included and the edit of the attenuation and buildup coefficients is suppressed. Therefore, only the two input parameters corresponding to these options were entered. The "A2" entry in the 62$$ array is explained in Appendix A and means that the following data is to be input starting at the second position in the array. The "E" means that no more entries are to be made in that array. Note that the first "T" terminator can be put on the same line as an array. However, since FIDO will stop reading a line when a "T" is encountered, the second terminator must be on a separate line.

The third case also models a point source, but in an air medium and with the addition of a large concrete slab shield. The location and dimensions of the shield and the locations of the detectors are given in units of feet. The 4$$, 5**, 6$$, and 8** arrays were input explicity,

```
SMP1A:  POINT SOURCE IN "INFINITE" WATER MEDIUM W/O BUILDUP
  O
61$$    1  0  0  1  17  1  1  0  1  0
62$$    2  0  0  1  2
63**    1.0  1.0   1.0
  T
1**
  3R0.0     3R0.0       0.0
7**    0.04 0.06 0.08 0.12 0.14 0.16 0.25 0.35 0.45 0.55
  0.65 0.95 1.05 1.45 1.55 2.45 3.55 6.45
9**    10.0  0.0  0.0
       40.0  50.0  30.0
  T
SMP1B:  POINT SOURCE IN "INFINITE" WATER MEDIUM W/ BUILDUP
62$$    A3  -1  E       T
  T
SMP1C:  POINT SOURCE AND CONCRETE SLAB SHIELD
61$$    1  0  1  1  17  1  1  0  E
62$$    3  0  -1  10  1
63**    0.0  30.48   1.0            T
1**
  3R0.0     3R0.0       0.0
  2.0  0.0  0.0     1.0  10.0  10.0      0.0
2$$    3
3**    2.35
4$$    1          5**    1.0          6$$    1
7**    0.04 0.06 0.08 0.12 0.14 0.16 0.25 0.35 0.45 0.55
  0.65 0.95 1.05 1.45 1.55 2.45 3.55 6.45
8**    F 1.0
9**    4.0 0.0 0.0   4.0 0.0 2.0   4.0 0.0 4.0
  T
SMP1D:  LINE SOURCE AND CONCRETE SLAB SHIELD
61$$    A7  41  E
  T
1**
  0.0  0.0  -5.0     0.0  0.0  5.0   0.0
  2.0  0.0  0.0     1.0  10.0  10.0      0.0
4$$    41
  T
```

Fig. 2. Listing of PUTZ input for first sample problem.

even though they are normally the default values, because the length of the 1** array was changed in this case relative to the preceding cases and this change caused the values in the higher-numbered arrays to be destroyed. Note that several arrays can be entered on the same line.

The final case expands the point source to a line source, which is divided into a quadrature of 41 source cell points. Since the lengths of the arrays were not changed from the previous case, only the new values in the 1** and 4$$ arrays were required.

4.2 Sample 2 - Corner Crossing Test

The input file for the second sample problem is listed in Fig. 3. The problem models three tank-type bodies including one which surrounds a volume source. This particular problem was used extensively during the development of PUTZ in order to test several special situations. A diagram of the geometry projected onto the X-Y plane is given in Fig. 4. The locations of the bodies and the detectors were deliberately chosen to challenge the code's performance for a variety of special detector positions, such as within the source body, within a tank body, or interior to a tank body.

As shown in Fig. 4, several of the rays from the central source cell to the detectors cross at least one body edge or corner. This was done deliberately in order to check PUTZ's performance in this difficult situation. Numerical roundoff and/or logic flaws can cause strange results when a ray happens to cross two body surfaces at exactly the same point. Although this is a common problem in ray tracing codes, the problem is exaggerated in PUTZ since tank bodies contain twice as many corners as solid bodies. The ray trace edit and boundary crossing edit become very useful in debugging corner crossing problems.

4.3 Sample 3 - Large Scale Problem

The third and final sample problem demonstrates how complex an actual case can be. The example includes 22 volume sources (many of which are contained in tanks), 39 shield bodies, and 30 detector points. The input file is listed in Fig. 5. The geometry is roughly a large rectangular room with thick concrete walls and numerous small photon sources distributed throughout the interior of the room. An isometric view of the geometry is shown in Fig. 6. (Note: The computer-generated plot shown in Fig. 6 was produced by the JUNEBUG code[7] for displaying combinatorial geometries. It is not directly compatible with PUTZ geometries, but only minor modifications to the PUTZ body descriptors is required to use with JUNE-

```
SMP2:  TEST TANK SHELL OPTION - VOLM SRC W/ ASSORT'D TANKS
  0
61$$    0  1  5  1  15  1  1  27  1  0
62$$    9  0  -1  11  1
63**    0.0  2.54   1.0            T
1**
  0.0  1.0  0.0      2.0  2.0  1.0   D0.0
  0.0  1.0  0.0      2.0  2.0  1.0   2.0
 12.0  9.5  0.0      6.0 15.0  1.0   0.0
 12.0  9.5  0.0      6.0 15.0  1.0   2.0
 -5.0 -4.0 -5.0      1.0  1.0  1.0   0.0
 -5.0 -4.0 -5.0      1.0  1.0  1.0   0.08
2$$   2  3  2  3  2  4
3**   1.0  2.3  1.0  2.3  1.0  8.0
5**   1.0      6$$   1
7**    0.01  0.02  0.03  0.045  0.07  0.10  0.15
      0.30  0.45  0.70   1.0   1.5   2.0   2.5
       3.0   4.0
8**   F1.0
9**
  20.0  1.0  0.0
  20.0 11.0  0.0
  20.0 21.0  0.0
  10.0 21.0  0.0
  10.0 11.0  0.0
   8.0  9.0  0.0
  16.0 17.0  0.0
   1.5  1.75  0.0
 -8.0  -7.0  -8.0
  T
```

Fig. 3. Listing of PUTZ input for second sample problem.

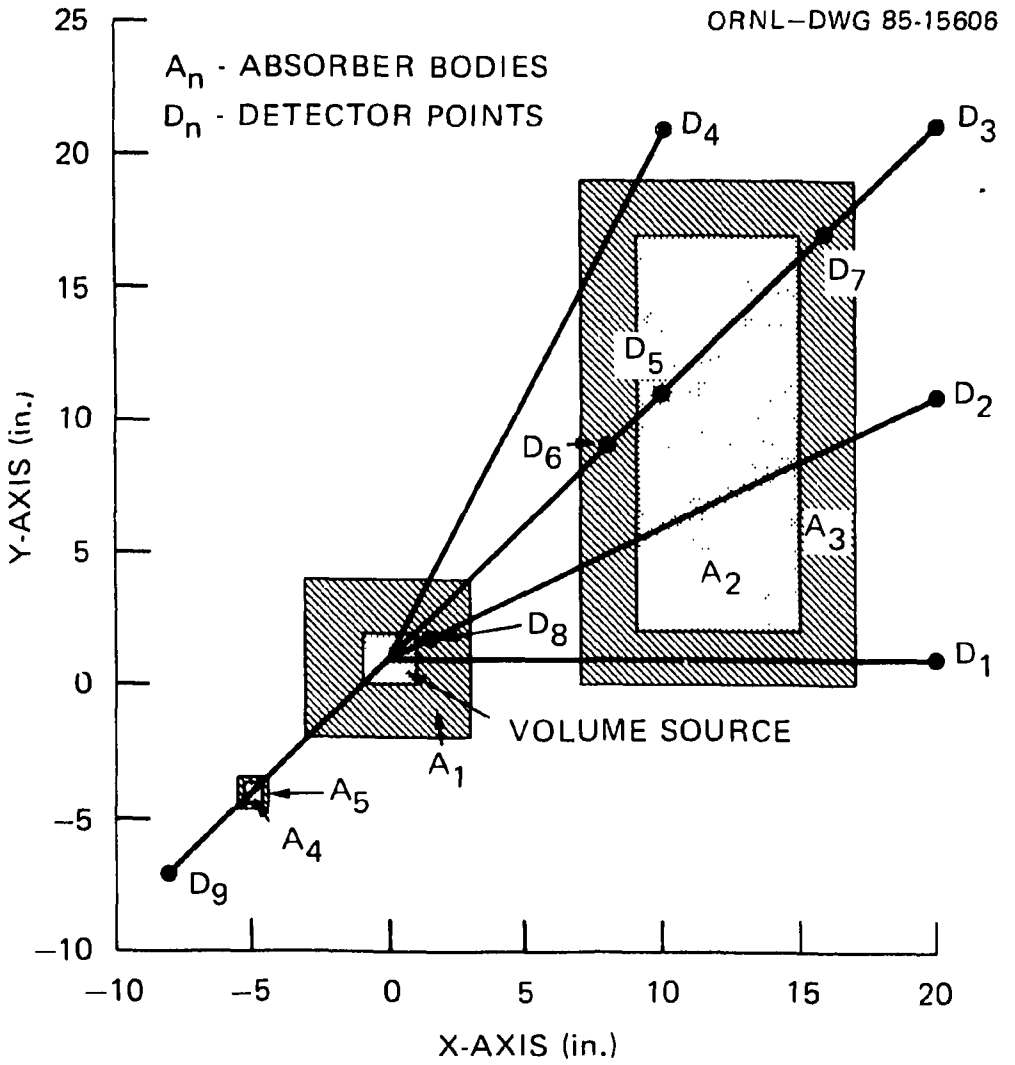

Fig. 4. Diagram of geometry for sample problem 1. (Projected onto X-Y plane).

```
SMP3:  LARGE PROBLEM WITH SEVERAL SOURCES AND DETECTORS
  O
61$$    0  22  39  1  15  2  0  320  1  0
62$$    30  0  -1  0  1
63**    0.0  30.48  1.347+16
  T
1**
   -21.60     14.12     4.68     2.22     2.22      3.6     0.0
   -22.20     15.64     11.00     0.44     0.44      2.0     0.0
   -20.88     15.40     12.00     0.30     0.30      1.5     0.0
   -19.44     15.40D      7.9     1.77     1.77      0.8     0.0
   -19.44     13.40     6.92     0.89     0.89      7.0     0.0
   -21.48     15.40     11.00     0.44     0.44      2.0     0.0
   -14.80     14.08     4.0      2.07     2.07     4.62     0.0
   -12.80     16.08     4.0      2.07     2.07     4.62     0.0
    -6.40     14.60     3.4      3.10     3.10     3.30     0.0
    -6.80     10.0     -2.6      2.5      1.42     1.0      0.0
    -6.80     6.60     -9.6      0.76     0.76     8.0      0.0
    1.60     8.92     -9.6      0.44     0.44     8.0      0.0
   -21.51     0.0     -14.42     7.00     3.50     1.16     0.0
    -7.96    -11.64     -8.16     1.63     1.63     4.42     0.0
    -5.28     -9.08     -8.16     1.18     1.18     7.24     0.0
    6.60     -8.0     -10.20     1.77     1.77     2.30     0.0
    8.0     -10.0     -10.20     1.77     1.77     2.30     0.0
    15.23     14.73     -14.00     3.54     3.54     2.00     0.0
    5.60     -2.40     -10.88     0.74     2.22     6.25     0.0
    41.2      0.0      -8.46     3.72     11.72     2.67     0.0
    41.2      0.0      -5.63     3.72     11.72     3.00     0.0
    41.D2      0.0      -2.06     3.72     11.72     4.13     0.0
   -21.60     14.12     4.68     2.22     2.22      3.6     0.0208
   -22.20     15.64     11.00     0.44     0.44      2.0     0.0208
   -20.88     15.40     12.00     0.30     0.30      1.5     0.0208
   -19.44     15.40      7.9     1.77     1.77      0.8     0.0208
   -19.44     13.40     6.92     0.89     0.89      7.0     0.0208
   -21.48     15.40     11.00     0.44     0.44      2.0     0.0208
   -14.80     14.08     4.0      2.07     2.07     4.62     0.0208
   -12.80     16.08     4.0      2.07     2.07     4.62     0.0208
    -6.40     14.60     3.4      3.10     3.10     3.30     0.0208
    -6.80     10.0     -2.6      2.5      1.42     1.0      0.0208
    -6.80     6.60     -9.6      0.76     0.76     8.0      0.0208
    1.60     8.92     -9.6      0.44     0.44     8.0      0.0208
    -7.96    -11.64     -8.16     1.63     1.63     4.42     0.0208
    -5.28     -9.08     -8.16     1.18     1.18     7.24     0.0208
    6.60     -8.0     -10.20     1.77     1.77     2.30     0.0208
    8.0     -10.0     -10.20     1.77     1.77     2.30     0.0208
```

Fig. 5. *Listing of* PUTZ *input for third sample problem.*

```
    0.0      0.0     40.25     111.0     51.0      4.5      0.0
    -7.50     22.75     -7.5      96.0      5.5     15.0     0.0
    0.0     22.0     12.0     111.0      4.0     24.0      0.0
    0.0     22.75     31.0     111.0      5.5     14.0      0.0
    -12.0    -22.75     -7.5      87.0      5.5     15.0     0.0
    0.0     -22.0      8.0     111.0      4.0     16.0      0.0
    0.0     -22.75     27.0     111.0      5.5     22.0      0.0
   -52.75     0.0     -7.5       5.5     40.0     15.0      0.0
   -52.0      0.0     12.0       4.0     40.0     24.0      0.0
   -52.75     0.0     31.0       5.5     40.0     14.0      0.0
    52.75     0.5     -7.5       5.5     27.0     15.0      0.0
    49.5      0.5     -7.5       1.0     16.0     15.0      0.0
    52.0      0.0     12.0       4.0     40.0     24.0      0.0
    52.75     0.0     31.0       5.5     40.0     14.0      0.0
    -12.0    -19.25    -17.75     87.0     12.5      5.5     0.0
    0.0      0.5     -17.75     111.0     27.0      5.5      0.0
    -7.50     19.75    -17.75     96.0     1D1.5      5.5    0.0
    42.50     14.25     -2.75     15.0     11.5      5.5     0.0
    45.50     11.25    -10.25     15.0      5.5      9.5     0.0
    37.75     17.00    -10.25      5.5      6.0      9.5     0.0
    38.0     -13.75     -2.75     24.0     12.5      5.5     0.0
    38.0     -10.25    -10.25     24.0      5.5      9.5     0.0
    28.75    -16.50    -10.25      5.5      7.0      9.5     0.0
2$$   19R 2  4  6  4   16R 4   23R 3
3**   12R 1.0  2.68  4R 1.0  1.69  1.31  0.74  0.52  0.36
      16R 7.8
      2R 2.47  3.5  2Q3  3R 2.47  3.5   10R 2.47
4$$  3 3 5  3R2  3R2  3 3 2  3R2  2 2 3  3 3 5  3 3 5
     3R4  4 2 2  2 2 10  3R2  10 5 2  3 3 5  2 2 8  2 2 3
     2 2 3  4 4 3  1 3 3  4 16 3  4 16 4  4 16 5
5**   1.01 0.022 0.007 3.4 0.62 0.53 2R26.9 17.31 1.93
  13.24 4.41 79.47 2.33 3R2.02 1.54 0.29 1.33 45.7 1.01
6$$   17R 1  3R 2  1  2
7**    0.01  0.02  0.03  0.045  0.07  0.10  0.15
      0.30  0.45  0.70   1.0   1.5   2.0   2.5
       3.0   4.0
8**   0.2939  0.0683  0.0683  0.0604  0.0429  0.0576  0.0371
    0.0209  0.1036  0.2386  0.0D058  8.56-4  1.67-3  2.74-5
    3.21-6
    0.1436  0.0062  0.0041  0.0058  0.0035  0.0023  0.0029
    0.0012  0.1407  0.6737  0.0132  0.0026  7.16-8  2.18-10
    0.0
```

Fig. 5. Cont'd.

```
9**
      -49.0         0.0        24.0
      -22.0         0.0        24.0
       -8.0         0.0        24.0
        5.0         0.0        24.0
       26.0         0.0        24.0
       41.0         0.0        24.0
      -49.0         0.0        42.5
      -22.0         0.0        42.5
       -8.0         0.0        42.5
        5.0         0.0        42.5
       26.0         0.0        42.5
       41.0         0.0        42.5
      -49.0         0.0       -35.0
      -22.0         0.0       -35.0
       -8.0         0.0       -35.0
        5.0         0.0       -35.0
       26.0         0.0       -35.0
       41.0         0.0       -35.0
      -14.0        15.0        24.0
      -14.0        15.0        42.5
      -14.0        15.0       -35.0
      -55.5         0.0        -9.8
       -7.0       -25.5       -10.5
       7.22       -25.5       -8.53
      41.0        -25.5       -8.2
      55.5          0.0       -8.2
      15.22        25.5      -13.95
      -6.56        25.5        3.3
     -14.44        25.5        3.3
     -18.D05        25.5       11.5
  T
```

Fig. 5. Cont'd.

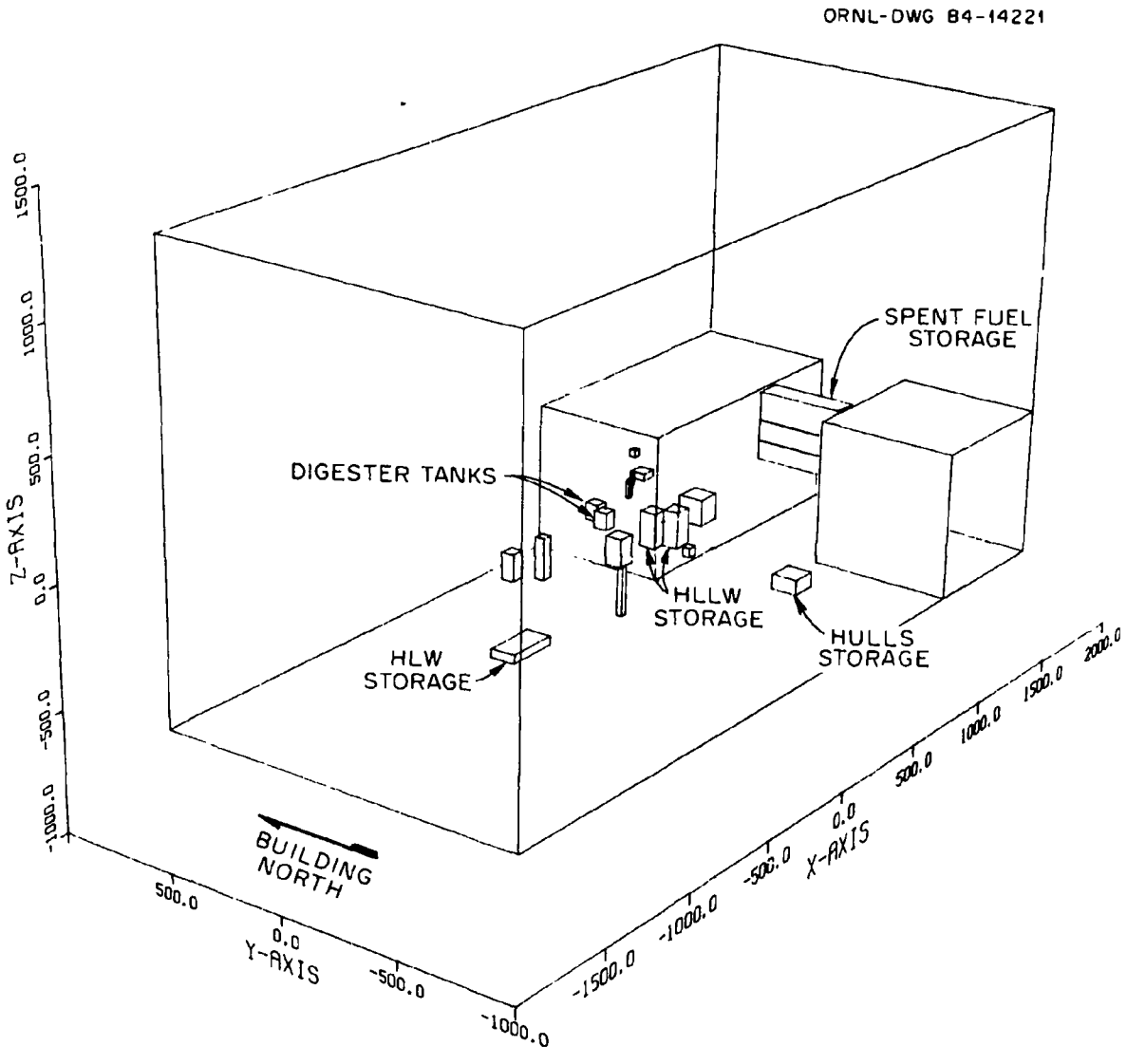

Fig. 6. Perspective view of geometry for sample problem 3. (Provided by S. N. Cramer.)

BUG.) The detector positions are located both within the room and immediately outside the room walls. A total of 23 shield bodies were needed to describe the walls, ceiling and floor of the room because of geometric irregularities in the room design.

The greatest difficulties in setting up such a large problem are the correctness of the geometry model and the proper correspondence between the body descriptors, material numbers, material densities, and source mesh numbers. Careful diagramming of the intended geometry and careful checking of the input are clearly required. One must be especially careful to avoid overlapping bodies. The code would not normally recognize this occurrence and would yield unpredictable (and erroneous) results.

This sample problem was executed on the ORNL IBM 3033 computer and required 15 to 30 seconds of processor time for each detector point. The total CPU time for the job was 12.4 minutes. Approximately 19,000 words (75k bytes) of memory were needed for input and storage arrays.

4.4 Description of Output

A complete listing of output from a PUTZ case is given in Fig. 7. The case is identical to Sample 2 except that only Detector No. 3 is specified and the IEDT input parameter is changed from 11 to 111 in order to demonstrate the full diagnostic print features. The first portion of the output is simply an edit of the input parameters and arrays. The only computation involved in this part of the listing is the calculation of the values given below the title: "SOURCE STRENGTH BY GROUP FOR EACH SOURCE BODY." The source strength for a given energy group in a given source body is computed by multiplying the AGAM factor (63** array) by the source intensity for that body (5** array) and by the normalized group spectrum value for that body (8** array).

If IDAT (62$$ array) had been set equal to 1, then a listing of the mass attenuation data and buildup factor coefficients would have been printed next. These data are given in Tables 1 and 2 in Chapter 2.

The next block of output pertains to the ray tracing and dose calculation for the first detector (the only detector in this example). The coordinates of the mid-points of the first source are given for each of the X, Y, and Z coordinate directions, followed by the bulky edit of boundary-crossing data. These data are with respect to the central souce cell of the first source. Each body is tested for possible soure-to-detector ray crossings. A brief description of the listed values is given below:

```
P*U*T*Z -- POINT-KERNEL USING TANK ZONES
           BY D.TD.INGERSOLL
           VERSION 1.0 -- LAST MODIFIED 6/21/85

SMP5:  TEST TANK SHELL OPTION - VOLM SRC W/ ASSORT'D TANKS

                                                    PRT ON         00
                                                    61$$    0  1  5  1  15  1  1  27  1  0
                                                    62$$    1  0  -1  111 1
   61$ ARRAY      10 ENTRIES READ
                                                    63**    0.0  2.54   1.0           T
   62$ ARRAY       5 ENTRIES READ
   63* ARRAY       3 ENTRIES READ


    0T
SOURCE/BODY PARAMETERS:  (DEFAULTS IN <>S)

 NLS=      0   NO. OF LINE SOURCES  <0>
 NVS=      1   NO. OF V'''   'OURCES  <0>
NABS=      5   NO. OF AB.        (SHIELDS)  <0>
INPT=      1   1/2 - (CENT.  .IDTH)/(MIN,MAX) BODY DESCRIPTORS  <1>
  NG=     15   NO. OF ENERGY GROUPS IN SOURCE SPECTRA  <0>
 NSP=      1   NO. OF DIFFERENT SOURCE SPECTRA  <1>
NLPT=      1   MAXIMUM NO. OF POINTS IN ANY LINE SOURCE  <1>
NVPT=     27   MAXIMUM NO. OF POINTS IN ANY VOLUME SOURCE  <27>
LOGM=      1   1/2 - USE LINEAR/LOG SPACING OF SOURCE MESH  <1>
 IWT=      0   0/1 - NO EFFECT/USE SOURCE ATTENUATION WEIGHTING  <0>


DETEDCTOR AND MISC PARAMETERS:  (DEFAULTS IN <>S)

  ND=      1   NO. OF DETECTORS  <0>
IDAT=      0   0/1 - NO EFFECT/PRINT ATTN & BLDUP DATA  <0>
IBUF=     -1   -1/0/N - BUILDUP FACTOR CONTROL  <-1>
                  -1 --> USE RESPECTIVE BUILDUP FOR EACH MATERIAL
                   0 --> DO NOT INCLUDE ANY BUILDUP
                   N --> USE MATERIAL "N" BUILDUP FOR ALL MATERIALS
IEDT=    111   0/N - NO EFFECT/DIAGNOSTIC EDIT  <0>
                 N = N1 + N2 + N3  WHERE:
                 N1=1   --> EDIT VOLUME SOURCE MESH MID-POINTS
                 N2=10  --> EDIT SOURCE-DETECTOR RAY TRACE
                 N3=100 --> EDIT BOUNDARY CROSSING DATA
MATX=      1   MATERIAL EXTERNAL TO ALL BODIES <1=AIR>


REAL-TYPE PARAMETERS:  (DEFAULTS IN <>S)

DENX=       0.0     DENSITY OF MATERIAL EXTERNAL TO ALL BODIES  <0.0>
CONV=       2.54    CONVERSION FROM DESCRIPTOR UNITS TO CENTIMETERS  <1.0>
AGAM= 1.000E 00     GAMMAS/SEC FOR UNIT SOURCE INTENSITY  <1.0>


**************************
 141056 LOCATIONS AVAILABLE
   1727 LOCATIONS REQUIRED
**************************
```

Fig. 7. Listing of PUTZ output from modified sample problem 2.

```
                                                D          1**
                                                            0.0  1.0  0.0      2.0  2.0  1.0   0.0
                                                            0.0  1.0  0.0      2.0  2.0  1.0   2.0
                                                           12.0  9.5  0.0      6.0 15.0  1.0   0.0
                                                           12.0  9.5  0.0      6.0 15.0  1.0   2.0
                                                           -5.0 -4.0 -5.0      1.0  1.0  1.0   0.0
                                                           -5.0 -4.0 -5.0      1.0  1.0  1.0   0.08
                                                          2$$   2  3  2  3  2  4
    1* ARRAY      42 ENTRIES READ
                                                          3**   1.0  2.3  1.0  2.3  1.0  8.0
    2$ ARRAY       6 ENTRIES READ
                                                          5**   1.0      6$$   1
    3* ARRAY       6 ENTRIES READ
    5* ARRAY       1 ENTRIES READ
                                                          7**    0.01  0.02  0.03  0.045  0.07  0.10  0.15
    6$ ARRAY       1 ENTRIES READ
                                                                 0.30  0.45  0.70   1.0   1.5   2.0   2.5
                                                                  3.0   4.0
                                                          8**   F1.0
    7* ARRAY      16 ENTRIES READ
                                                          9**
    8* ARRAY      15 ENTRIES READ
                                                            20.0 21.0  0.0
                                                            T
    9* ARRAY       3 ENTRIES READ


    0T
BODY DESCRIPTIONS

BODY  TYPE          (X,Y,Z) POSITION                (X,Y,Z) DIMENSION          THICK   MATL   DENSITY
                    -------------------------       -------------------------
  1   VOLM        0.0       1.00      0.0       D   2.00      2.00      1.00     0.0      2      1.000
  2   ABSR        0.0       1.00      0.0           2.00      2.00      1.00     2.00     3      2.300
  3   ABSR       12.00      9.50      0.0           6.00     15.00      1.00     0.0      2      1.000
  4   ABSR       12.00      9.50      0.0           6.00     15.00      1.00     2.00     3      2.300
  5   ABSR       -5.00     -4.00     -5.00          1.00      1.00      1.00     0.0      2      1.000
  6   ABSR       -5.00     -4.00     -5.00          1.00      1.00      1.00     0.08     4      8.00

SOURCE DATA

  SRC     TYPE     INTENSITY     SPEC      NO. OF POINTS

   1      VOLM        1.000        1          3    3    3


 GROUP BOUNDARIES IN MEV

  0.010   0.020   0.030   0.045   0.070   0.100   0.150   0.300   0.450   0.700
  1.000   1.500   2.000   2.500   3.000   4.000


 SOURCE STRENGTH BY GROUP FOR EACH SOURCE BODY

 GRP.    SRC.     1
    1  6.6667E-02
       GRP.     2 THRU GRP.    15 SAME AS ABOVE
TOTAL  1.0000E 00
SMP5:  TEST TANK SHELL DOPTION - VOLM SRC W/ ASSORT'D TANKS
```

Fig. 7. Cont'd.

```
 DETECTOR  1  WITH (X,Y,Z) POSITION:     20.0     21.0      0.0

COORDINATES FOR SOURCE CELL MIDPOINTS

     X DIRECTION:    -0.67     0.0      0.67
     Y DIRECTION:     0.33     1.00     1.67
     Z DIRECTION:    -0.33    -0.00     0.33
BODY   1 DIAGNOSTIC DATA:
      P:      1.000    -1.000          2.000     0.000          0.500    -0.500
      D:      1.414    -1.414      1.414     -1.414**********************
   CORD:      1.000    -1.000      1.000     -1.000**********************
              2.000     0.000      2.000      0.000**********************
             -0.000    -0.000     -0.000     -0.000      0.500     -0.500
 IFF,IC:  4,          1      2      3      4
IFD,ICK:  1,          1
BODY   2 DIAGNOSTIC DATA:
      P:      3.000    -3.000          4.000    -2.000          2.500    -2.500
      D:      4.243    -4.243      4.243     -4.243**********************
   CORD:      3.000    -3.000      3.000     -3.000**********************
              4.000    -2.000      4.000     -2.000**********************
             -0.000    -0.000     -0.000  D  -0.000      2.500     -2.500
 IFF,IC:  4,          1      2      3      4
IFD,ICK:  1,          1
BODY   2 DIAGNOSTIC DATA:
      P:      1.000    -1.000          2.000     0.000          0.500    -0.500
      D:      1.414    -1.414      1.414     -1.414**********************
   CORD:      1.000    -1.000      1.000     -1.000**********************
              2.000     0.000      2.000      0.000**********************
             -0.000    -0.000     -0.000     -0.000      0.500     -0.500
 IFF,IC:  4,          1      2      3      4
IFD,ICK:  1,          1
BODY   3 DIAGNOSTIC DATA:
      P:     15.000     9.000         17.000     2.000          0.500    -0.500
      D:     21.213    12.728     22.627      1.414**********************
   CORD:     15.000     9.000     16.000      1.000**********************
             16.000    10.000     17.000      2.000**********************
             -0.000    -0.000     -0.000     -0.000      0.500     -0.500
 IFF,IC:  2,          1      2
IFD,ICK:  2,          1      2
BODY   4 DIAGNOSTIC DATA:
      P:     17.000     7.000         19.000     0.0D00         2.500    -2.500
      D:     24.042     9.899     25.456     -1.414**********************
   CORD:     17.000     7.000     18.000     -1.000**********************
             18.000     8.000     19.000      0.000**********************
             -0.000    -0.000     -0.000     -0.000      2.500     -2.500
 IFF,IC:  2,          1      2
IFD,ICK:  2,          1      2
BODY   4 DIAGNOSTIC DATA:
      P:     15.000     9.000         17.000     2.000          0.500    -0.500
      D:     21.213    12.728     22.627      1.414**********************
   CORD:     15.000     9.000     16.000      1.000**********************
             16.000    10.000     17.000      2.000**********************
             -0.000    -0.000     -0.000     -0.000      0.500     -0.500
 IFF,IC:  2,          1      2
IFD,ICK:  2,          1      2
BODY   5 DIAGNOSTIC DATA:
      P:     -4.500    -5.500         -3.500    -4.500         -4.500    -5.500
      D:     -6.364    -7.778     -6.364     -7.778**********************
   CORD:     -4.500    -5.500     -4.500     -5.500**********************
             -3.500    -4.500     -3.5D00     -4.500**********************
             -0.000    -0.000     -0.000     -0.000     -4.500     -5.500
```

Fig. 7. Cont'd.

```
 IFF,IC:  0,        1
BODY   6 DIAGNOSTIC DATA:
      P:    -4.420    -5.580         -3.420    -4.580         -4.420    -5.580
      D:    -6.251    -7.891    -6.251    -7.891**********************
   CORD:    -4.420    -5.580    -4.420    -5.580**********************
            -3.420    -4.580    -3.420    -4.580**********************
            -0.000    -0.000    -0.000    -0.000    -4.420    -5.580
 IFF,IC:  0,        1
BODY   6 DIAGNOSTIC DATA:
      P:    -4.500    -5.500         -3.500    -4.500         -4.500    -5.500
      D:    -6.364    -7.778    -6.364    -7.778**********************
   CORD:    -4.500    -5.500    -4.500    -5.500**********************
            -3.500    -4.500    -3.500    -4.500**********************
            -0.000    -0.000    -0.000    -0.000    -4.500    -5.500
 IFF,IC:  0,        1
RAY TRACE FROM CELL NO.  14 IN SOURCE   1:
```

	IBD	DTB	DIB	IRDR	MT	DEN	DIS
						D	
1	1	0.0	1.41	1	2	1.00	1.41
2	2	1.41	2.83	2	3	2.30	2.83
3	3	12.73	8.49	4	1	0.0	5.66
4	4	9.90	2.83	3	3	2.30	2.83
5	4	21.21	2.83	5	2	1.00	8.49
6					3	2.30	2.83
7					1	0.0	4.24

```
CALCULATED FLUXES
GRP.    SRC.     1
   1  1.0334E-22
   2  6.4921E-20
   3  5.7665E-17
   4  1.8837E-13
   5  8.1209E-11
   6  8.6444E-10
   7  5.3735E-09
   8  1.5739E-08
   9  3.1164E-08
  10  5.0291E-08
  11  8.1579E-08
  12  1.1373E-07
  13  1.4216E-07
  14  1.6747E-07
  15  2.0246E-07
TOTAL  8.1090E-07


 DOSE RATE (RADS/HOUR) FOR EACH SOURCE
   1  2.403E-12

ELAPSED TIME FOR DETECTOR:    0 MIN     0.48 SEC
          CUMULATIVE TIME:    0 MIN     0.48 SEC

IHC002I STOP       0
```

Fig. 7. Cont'd.

P: the minimum and maximum coordinate values of the body in the X, Y, and Z directions

D: the distance from the source to the point at which each of the six surfaces of the body are crossed

CORD: the X, Y, and Z coordinates of each of the six body surfaces crossed

IFF,IC: the initial number of acceptable surface crossings and the corresponding surface numbers

IFD,ICK: the final number of acceptable surface crossings and the corresponding surface numbers

The above data are given twice for tank-type bodies (Bodies 2,4,6 in this example) in order to describe the outer and inner surfaces of the body. The IFF and IC data tell which of the surface crossings fall within the bounds of the body. These potentially valid crossings are further checked for special conditions such as the detector being within the body or duplication at corner crossings.

Next, a summary of the ray trace is given including the following information:

IBD: input body number

DTB: distance from source to body

DIB: distance traveled within body

IRDR: order of occurance of body along ray path

After incorporating the gaps between bodies, the following quantities are given in order of occurrence along the ray:

MT: material number

DEN: material density

DIS: thickness of material

If more than one source had been included, then the above data would have been repeated for each source.

The final output data for the first detector is a listing of the calculated gamma-ray flux given by group for each of the sources. The units of the flux are $(cm^2*s)^{-1}$ (assuming that AGAM was properly specified in units of s^{-1} and CONV was specified correctly). Next, the dose rate (in silicon) in units of rads/hr is given for each source. If more than one source had

been included, then a total dose rate summed over all sources would have been printed. Finally, the elapsed execution time for the detector is given followed by the cumulative time for all detectors.

If more than the one detector had been included, then all of the above output would have been repeated for each detector, followed by a final summary table giving the dose rate at each detector due to each source and due to all sources combined.

5. ACCESS AND EXECUTION PROCEDURES

The PUTZ code has been tested on two different computer systems: (1) the ORNL IBM 3033 mainframe computer, and (2) an IBM-PC microcomputer. The two code versions are identical in terms of input data and printed output. The only significant differences are the procedures for executing the program, and of course, the execution speeds. The execution procedures for the two versions are described below. Table 3 compares the execution time of the sample problems described in Chapter 4 when run on an IBM 3033, an IBM-PC with an 8087 numeric coprocessor installed, and an IBM-PC without the 8087 coprocessor. In all cases, the total doses were computed identically to 4 significant figures. The PC with 8087 is a factor of 100 slower than the mainframe for the larger problems. It is difficult to judge the relative speed for the simple problems since the time on the 3033 is extremely short. The PC without the coprocessor is as much as a factor of 18 slower than the same PC with the coprocessor. Obviously, the coprocessor is highly desirable even for problems of modest complexity. The third sample problem was not run on the PC since it was estimated to require approximately 20 hours with the 8087 and roughly 15 days (plus or minus a day) without the 8087.

TABLE 3. Execution Time for PUTZ Sample Problems

	Execution Time (seconds)		
Sample Problem	IBM 3033	PC w/ 8087	PC w/o 8087
Sample 1 - Simple Cases			
Case A	0.01	3.02	5.99
Case B	0.02	3.68	15.05
Case C	0.03	8.30	49.16
Case D	1.00	98.13	1646.06
Sample 2 - Tank Test	3.28	306.92	5524.84
Sample 3 - Large Case	742.19	-	-

5.1 IBM Mainframe Version 1.0/MF

PUTZ was originally developed and tested on the IBM 3033 computers located at ORNL. The code is written in standard FORTRAN language and was compiled using the FORTRAN-H compiler. An executable load module containing PUTZ resides on a private on-line disk and is available for general access. The job control language needed to execute a PUTZ case is:

```
//jobcard
//PUTZ EXEC PGM=PUTZ,REGION=640K
//STEPLIB DD DISP=SHR,DSN=X.DTI46103.UNFOLD
//FT05F001 DD DDNAME=SYSIN
//FTO6F001 DD SYSOUT=A
//SYSIN DD *
      .
      .
  PUTZ Input Deck
      .
      .
//
```

As discussed in Chapter 3, the PUTZ input deck begins with a problem title card followed by the FIDO-type input arrays. Additional cases can be run by simply adding a new title card and arrays immediately after the final block terminator of the first case.

The amount of computer memory required to load the PUTZ program is approximately 90 k-bytes. The amount of memory required for user input and internally-generated data arrays depends on the value of certain input parameters and can be computed using the following equation

$$\begin{aligned} NWORDS = {} & NLS*[ND+(6*NG)+36] + NVS*[ND+(6*NG)+40] + \\ & NABS*[(4*NG)+33] + NG*[NSP+(2*NSPT)+11] + \\ & 4*ND + 5*NSPT + 2 \end{aligned}$$

where:

NLS	=	number of point/line sources
NVS	=	number of volume sources
NABS	=	number of absorber bodies
ND	=	number of detectors
NG	=	number of energy groups
NSP	=	number of different source spectra
NSPT	=	maximum number of source cell points in any line or volume source

The total amount of computer memory needed can then be estimated from

$$\text{MEMORY (k-bytes)} = 90 + (\text{NWORDS}/256)$$

Although MEMORY may be (and should be) less than the REGION specified on the EXEC statement, PUTZ will allocate the full REGION amount of memory at the time of execution. Therefore, the user should not be excessive in the REGION request.

The only I/O transfers which occur are the reading of the input file and the writing of the output file. Therefore, the I/O requirements for PUTZ are always insignificant.

5.2 IBM Microcomputer Version 1.0/PC

The mainframe version of PUTZ was modified and compiled with the Microsoft FORTRAN Compiler Version 3.2. Although coding changes were relatively minor, some changes were made to the subroutine architecture shown in Fig. 1 (Chapter 2). Specifically, the runtime allocation of memory was removed by deleting the call to ALOCAT, and the major container array used to store all input and scratch data was set to a fixed dimension of 16000 words. Also, the ICLOCK system function used in subroutine ELAPS was replaced with a call to an assembler language routine, TICKER.[8] The amount of space needed for input and internal arrays is the same as for the mainframe code, but the amount of space available will always be 16000 words.

PUTZ/PC is executed by inserting the diskette containing the PUTZ program into the default drive and typing "PUTZ" followed by a carriage return. The program responds with title information and prompts the user for an input file name and then an output file name.

The input file must have been previously prepared using any editor or word processor and must be in ASCII format. The data in the file is identical to the "PUTZ Input Deck" indicated in the previous section for the mainframe input. The output file is created by PUTZ and will overwrite any existing file of the same name. A modest amount of information is written to the PC's monitor to let the user know what's happening. Only the total dose (by source) for each detector is provided. The program pauses for a user response at the end of each case before starting subsequent cases.

The output file contains the same output (including printer carriage control) that the mainframe version generates. A simple auxiliary program was written and compiled in IBM BASIC language to facilitate the listing of the output file. The program is run by typing "LISTOFF" followed by a carriage return. LISTOFF prompts the user for the name of the file to be listed, and then prints the file on an Epson-compatible printer. The listing is done in compressed mode (132 column wide) and properly interprets the carriage control characters in column 1 of each record. The compressed print mode is turned off upon completion of the listing.

One note of caution: The more restrictive range of real numbers available on the PC (approximately $10^{\pm 38}$ compared to $10^{\pm 78}$ for the mainframe) can cause problems with underflow and overflow errors. In most cases observed, the errors occured in the calculation of the buildup factors. In this case, the easiest solution is to not use buildup (set IBUF=0).

6. CAUTIONS AND RECOMMENDATIONS

The need for PUTZ was discussed in the first chapter, and the code's versatility and friendliness was described in the next few chapters. Now some of the limitations of point-kernel methods, some of the particular weaknesses of PUTZ, and some of its aspects needing future development must be discussed.

6.1 Concerns and Limitations

It cannot be emphasized enough that the point-kernel method is an approximate method and is intended for scoping analyses or when accuracy is not a major concern. The accuracy of the method is generally assumed to be "order of magnitude." Of course, some simple problems may be more accurate than a factor of 10, perhaps even a factor of 2. One major limitation of the method (and of PUTZ) is that it does not include the contributions of scattered radiation which has scattered from material behind or to the side of the detector. In other words, only the effects of material directly along the line-of-sight between the source and detector are included. This is a particularly worrisome limitation since it will always cause the calculated dose to be lower than the true dose. It is normally preferred that errors in the dose predictions be in the conservative direction, i.e. higher than the true dose. The absence of the backscattered radiation can easily cause the calculated dose to be a factor of 2 to 3 too low if the detector is near a wall or massive item.

Another concern/caution is with respect to the energy-dependent photon flux output by PUTZ. Because of the definition of buildup factors, it is not strictly appropriate to use the photon flux as the final output. Rather, it is an intermediate step in the computation of the total dose. This is especially true when looking at the groupwise flux since the kernel method does not include the transfer of a photon's energy from one group to another. This limitation can cause the computed group flux to be more than a factor of 100 different from the true value. The group flux is output from PUTZ for guidance purposes only, and should be used merely as an indication of which energies tend to dominate the problem.

There are other cautions in the use of PUTZ such as the impact of the source mesh on the dose, and additional inaccuracies associated with multilayer shields. An attempt was made in PUTZ to provide sufficient options so that the user could study these effects and bracket the solution. In the case of the source mesh, the user should try refining the mesh for at least

some cases, and also try the logarithmic mesh option. The problem with multilayer shields results from the approximate treatment of buildup factors and can be investigated by turning off the buildup feature and/or explicitly forcing the use of a single material buildup.

A final caution regarding PUTZ pertains to the very feature which gives the code its name -- the tank zone option. Although this option is convenient, it does add some computational overhead. In essence, the inner ray-tracing routine must search all 12 surfaces of the tank body for valid crossings rather than the usual 6 surfaces of a solid body. This is okay if the tank really does contribute a significant amount of attenuation, but may not be efficient if the tank is quite thin. In this case, it would be better to ignore the tank. Note that ignoring the tank will make the computed dose higher, which is normally conservative and hence acceptable.

6.2 Items for Future Development

There are several items and areas of development which would significantly improve the versatility of PUTZ. First and foremost, the buildup factor data should be updated. The data used in PUTZ are of unknown origin and are likely outdated. Newer data are available[9] which span a wider energy range, and hence would eliminate inaccuracies caused by extrapolating the current data. At the same time that the data are modernized, the Taylor exponential fitting method used by PUTZ should be replaced with a fitting function such as the GP function,[10] which is known to be more accurate.

An addition to PUTZ which would be very useful is a feature or auxiliary program which would graphically plot the source and shield geometry. An adaptation of the JUNEBUG code[7] could be used, although a much simpler code could probably be written owing to the more restrictive rectangular geometries used in PUTZ. Such a feature would greatly facilitate the construction of the geometry model and the presentation of the computed doses.

Also, considerable work could be done to make the microcomputer version of PUTZ more interactive. Ideally, the user could be prompted for individual input parameters, which could then be saved and used later. Even simple on-screen plotting of the geometry model would greatly enhance the code. More monitoring of the progress of the calculation might be useful, and perhaps even some capability to alter some input options during the course of the calculation. Some checking for the existence of the input file and output file is also recommended.

REFERENCES

1. DN. M. Schaeffer, *Reactor Shielding For Nuclear Engineering*, U. S. Atomic Energy Commission, 1973.
2. J. J. Taylor, *Application of Gamma Ray Build-Up Data to Shield Design*, WAPD-RM-217, 1954.
3. A. B. Chilton, J. K. Shultis, and R. E. Faw, *Principles of Radiation Shielding*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.
4. E. J. Allen, S. N. Cramer, J. F. Mincey, and J. V. Pace, *Calculational Studies of Worst-Case Radiation Fields Expected Within Process Cells 1 and 2 forthe Hot Experimental Facility Conceptual Design*, ORNL/TM-7720, 1981.
5. W. W. Engle, Jr., *A Users Manual For ANISN - A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering*, K-1693, 1973.
6. W. A. Rhoades and R. L. Childs, *An Updated Version of the DOT 4 One- and Two-Dimensional Neutron/Photon Transport Code*, ORNL-5851, 1982.
7. M. B. Emmett, L. M. Petrie, and J. T. West, *JUNEBUG-II -- A Three-Dimensional Geometry Plotting Code*, ORNL/NUREG/CSD-2/V2/R2, 1984.
8. W. Claff, Boston Computer Society, published in *BYTE Magazine*, February 1984.
9. D. K. Trubey, "Standard Reference Data for Gamma-Ray Transport in Homogeneous Media," *Proceedings of the Sixth International Conference on Radiation Shielding*, Vol. II, p. 1241, CONF-830538, 1983.
10. Y. Harima, "An Approximation of Gamma-Ray Buildup Factors by Modified Geometrical Progression," *Nucl. Sci. and Engr.*, Vol. 83, p. 299, 1983.

APPENDIX A - FIDO INPUT *

*From Reference 6.

The FIDO input method is especially devised to allow the entering or modifying of large data arrays with minimum effort. Special advantage is taken of patterns of repetition or symmetry wherever possible. The FIDO system was patterned after the input method used with the FLOCO coding system at Los Alamos, and was first applied by Atomics International to the DTF-II[43] code. Since that time, numerous features requested by users have been added, a free-field option has been developed, and the application of FIDO has spread to innumerable codes.

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

Three major types of input are available: fixed-field input, free-field input, and user-field input. Each field includes up to three subfields. Each array is identified by a "*array originator field*" having two subfields:

Subfield 1:	An integer *array identifier* signals the identity of the array to follow.
Subfield 2:	An *array-type indicator*, used as follows:
	"$" fixed-field integer array
	"*" fixed-field real array
	"$$" free-field integer array
	"**" free-field real array
	"U" user-field, format to be read in
	"V" user-field, previous format to be used

These subfields are written together, without blanks. Data are then placed in successive *data fields* until the required number of entries has been accounted for.

In entering data, it is convenient to think of a "index" or "pointer" which is under control of the user, and which specifies the position in the array into which the next data entry is to go. The pointer is always positioned at the first array location by entering the array originator field. The pointer subsequently moves according to the data operator chosen.

Free-field input allows an arbitrary number of fields to be entered in columns 1-72, with fields separated by one or more blanks. A field must begin and end in the same record. In general, a data field has up to three fields:

Subfield 1: The *data numerator*, N_1, an integer.
Subfield 2: The *data operator*, N_2, a character.
Subfield 3: The *data entry*, N_3, an integer or real number.

If subfields 1 and 2 are both used in a field, they must not be separated by blanks. Blanks may always precede subfield 3, however. All entries following a "/" in a given record are ignored.

Single data entries use only N_3, which may be entered with or without a decimal point. The type (integer/real) will be determined by the previous array identifier:

```
1$$  1  2  3          /enter integers 1,2,3
2**  1.  1.1  2       /enter reals 1.0,1.1,2.0
```

An exponential field may be added, with or without the "E" identifier. No imbedded blanks within the subfield are allowed. The subfield may have no more than 9 columns, not including the exponent field. The pointer is advanced by 1 for each entry:

```
          2**   1.E0  .11+1  200-2   /same entries as above
          3**   1234.5678E-3         /8 columns + decimal
BUT NOT   3**   1.23456789           /too many columns
```

Multiple data entries use N_2 and (except "F") N_1 together with N_3 to enter several items with a single field. The type of operation is indicated by N_2.

"R" indicates that data entry N_3 is to be repeated N_1 times. The pointer is advanced by N_1.

"I" indicates linear interpolation. The data numerator, N_1, indicates the number of interpolated points to be supplied. The data entry N_3 is entered, followed by N_1 interpolated entries equally spaced between that value and the data entry found in the third subfield of the next field. The next field may be a single- or multiple-data entry. The pointer is advanced by $N_1 + 1$. The field following an "I" entry is then processed normally, according to its own data operator. The "I" entry is especially valuable for specifying a spatial mesh. In "$" arrays, interpolated values will be rounded to the nearest integer.

"L" indicates logarithmic interpolation. The effect is the same as that of "I" except that the resulting data are evenly separated in log-space. This is limited to positive real numbers, for obvious reasons.

"F" fills the remainder of the array with N_3. N_1 is not used.

For example, the following are equivalent ways of filling an 8-entry array:

```
2$$  1  2  3  4  6 6  8 8
2$$  2I1  4  2R6  F8        /same as above
```

Sequence data entries allow entries to be patterned after data entered by previous fields or previously existing in storage. Fields N_1 and N_2, and N_3 are used, except that if N_1 is omitted, it is taken to be 1.

"Q" is used to repeat sequences of numbers without modification. The length of the sequence is given by the third subfield, N_3. The sequence of N_3 previous entries is to be repeated N_1 times. The pointer is advanced by $N_1 * N_3$. This feature is especially valuable geometry specification.

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

"N" has the same effect as "Q," except that the order of the sequence is reversed each time it is entered.

"M" has the same effect as "N," except that the sign of the sequence is reversed each time it is entered. Options Q, G, and M are valuable in entering directional quadrature sets.

As examples:

```
81**    1 2 3 1Q3       /1,2,3,1,2,3
82**    -3 -2 -1 M3     /-3,-2,-1,1,2,3
83**    3R-2 G3         /-2,-2,-2,2,2,2
```

Zero data entries use subfields N_1 and N_2.

"Z" sets the next N_1 entries to 0. The pointer is advanced by N_1.

As an example:

```
1$$  3Z 1 2              /0,0,0,1,2
```

Pointer-movement data entries move the pointer without changing the data array. Subfields N_1, N_3 or neither may be required.

"S" indicates that the pointer is to skip forward over N_1 positions, leaving those array positions unchanged.

"B" moves the pointer backward N_1 positions.

"A" moves the pointer to the position N_3.

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

For example, given the following sequence of entries, comments indicate the result of reading an array of length 8:

```
1$$   1 2 3 4 5 6 E   /E terminates the array, leaving
                        items 7 and 8 unchanged.
1$$   2 1S 4 A7 7 8   /now we have 2,2,4,4,5,6,7,8
1$$   2I 1 4 1B F8    /1,2,3,8,8,8,8,8
```

Edit fields control printing within the FIDO subroutines.

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

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

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

"/" occurring in column 1 causes the entire record to be ignored as input, but to be printed as a comment in the output stream.

A *block termination field* consists of a field having only "T" in the second subfield.

The *reading of data to an array* is terminated when a new array originator field is supplied, or when the block is terminated by a block termination field. If a incorrect number of positions has been filled, an error edit is given, and a flag is set which will later abort execution of the problem. FIDO then continues with the next array if an array originator was read. The new array originator need not begin a new record. Otherwise, FIDO returns control to the calling program. For example:

```
1$$  1  2  3        2**  F0  T
```

User-field input allows the user to specify the input format. If "U" is specified as the array-type indicator, the FORTRAN format to be used must be supplied in columns 1-72 of the next record. The format must be enclosed by the usual parentheses. The data for the entire array must follow on successive records. The rules of ordinary FORTRAN input as to exponents, blanks, etc., apply. If the array data do not fill the last record, the remainder of the record must be left blank. The user must insure that his format specifies the correct type of data, i.e., real or integer.

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

For example, for an array of 4 entries:

```
10U
(6I2)
1 2 3 4
11V
4 3 2 1
```

would enter integers 1,2,3,4 into the 10th array and 4,3,2,1 into the 11th.

Fixed-field input uses 1 to 6 fields per input record, with fixed, 12-column fields. It is thoroughly described in Reference 1. Because it has been almost entirely replaced by the free-field format, its description is not repeated here.

ORNL/TM-9803
Distribution Category UC-86

DISTRIBUTION

1. L. S. Abbott
2. J. F. Birdwell
3. J. O. Blomeke
4. B. L. Broadhead
5. J. A. Bulcholz
6-8. W. D. Burch
9. D. G. Cacuci
10. R. H. Chapman
11. R. L. Childs
12. S. N. Cramer
13. P. W. Dickson, Jr. (Consultant)
14. W. W. Engle, Jr.
15. M. J. Feldman
16. G. H. Golub (Consultant)
17. N. R. Grant
18. W. S. Groenier
19-23. M. J. Haire
24. D. C. Hampson
25. R. M. Haralick (Consultant)
26-31. D. T. Ingersoll
32. M. V. Keigan
33. H. T. Kerr
34. F. C. Maienschein
35. B. F. Maskewitz
36. B. L. McGill
37. J. A. McEvers
38. S. A. Meacham
39. E. L. Nicholson
40. J. V. Pace III
41. F. L. Peishel
42. W. A. Rhoades
43. R. W. Roussin
44. R. T. Santoro
45. S. L. Schrock
46. M. B. Sears
47. R. W. Sharpe
48. C. O. Slater
49. G. O. Smolen
50. D. Steiner (Consultant)
51. J. G. Stradley
52. D. K. Trubey
53. P. Welesko
54. R. M. Westfall
55. M. E. Whatley
56. O. O. Yarbro
57. H. R. Yook
58. A. Zucker
59-63. EPMD Reports Office
64-65. Laboratory Records, ORNL-RC
66. ORNL Patent Section

EXTERNAL DISTRIBUTION

67. D. E. Bailey, Director, Division of Fuels and Reprocessing, Office of Facilities, Fuel Cycle, and Test Programs, U. S. Department of Energy, Washington, DC 20545

68. F. P. Baranowski, 1110 Dapple Grey Court, Great Falls, VA 22066

69. S. J. Beard, Vice President, Marketing and Uranium Operations, Exxon Nuclear Company, Inc., 600 108th Avenue, N.E., C-00777, Bellevue, WA 98009

70. C. A. Burgess, Westinghouse Hanford Company, P.O. Box 1970-W-B108, Richland, WA 99352

71. R. E. Dahl, Jr., Hanford Engineering Development Laboratory, P.O. Box 1970, Richland, WA 99352

72. M. J. Ohanian, Associate Dean for Research, College of Engineering, 300 Weil Hall, University of Florida, Gainesville, FL 32611

73. J. F. Proctor, Senior Technical Specialist, E. I. du Pont de Nemours and Company, Savannah River Laboratory, Aiken, SC 29801

74. M. J. Rohr, Program Manager, Consolidated Fuel Reprocessing Program, Energy Programs Division, Energy Technology Branch, DOE-ORO, Oak Ridge, TN 37831

75. Office of Assistant Manager for Energy Research and Development, DOE-ORO, Oak Ridge, TN 37831

76-183. Given distribution as shown in TIC-4500 under UC-86, Consolidated Fuel Reprocessing Category