THE MORSE-K PROGRAM PACKAGE VERSION 1:

A MONTE-CARLO PROGRAM SYSTEM FOR SMALL COMPUTERS

F.A.R. SCHMIDT, W. BERNNAT

Institute for Nuclear Energy

Universitaet Stuttgart

Dec. 1972

Report # 4-11

Abstract

MORSE-K is a multigroup Monte Carlo program. It can be used for core-, shielding and perturbation calculations both for neutrons and gamma rays. MORSE-K was developed at the Institut für Kernenergetik in Stuttgart, Germany from the ORNL program MORSE. This report describes MORSE-K and presents examples for its various applications.

Translated by Ernest G. Silver, Oak Ridge National Laboratory, Oak Ridge, Tenn. (From the German)

BLANK PAGE

I. INTRODUCTION

Despite the continually increasing capacity of computers, it is desirable to have programs which can perform even complicated calculations with a minimum of core storage and computing time. In the present state of the art they can be obtained only by use of Monte Carlo methods. This version of the code package MORSE-K fulfills the above requirements. Morse combines with these capabilities those of modern Monte Carlo programs [1]. Hence it is possible, in addition to making survey calculations, to perform Monte Carlo calculations in whatever detail is desirable with MORSE-K.

The Multi-group Monte Carlo program MORSE [2], developed in the Neutron Physics Division of ORNL, which was graciously put at our disposal even before its final publication, served as starting point for the development of MORSE-K. Several reasons persuaded us to use MORSE as a basis for our work. They resulted in the following practical properties of MORSE-K. MORSE-K is a multigroup Monte Carlo program. It can use the same cross-section libraries as do transport programs, on the basis of either finite differences or finite elements. This entails a certain limitation in the consideration of cross-section fine structures. However, this limitation is more than offset by the advantages connected with this. First of all, the Monte Carlo user no longer must carry the burden of data preparation alone. He can lean on the experience of others in the production of group data for reactor or shielding calculations. This experience is expressed in systems such as RSYST [3]. MORSE-K can easily be coupled with such systems. Further, it is no longer necessary to distinguish between different reactions. In particular, the production of secondary particles can be very simply included in multigroup data. This means, among other things, that the program is equally suitable for neutron and gamma transport. Additional advantages that are taken over in MORSE-K directly from MORSE can be described by the key words albedo boundary condition, time independence, general geometry package, arbitrary scattering order, weighting mechanisms, and analysis programs for shielding problems.

Version 1 of the program package MORSE-K consists of a basic program with which k_{eff} calculations can be performed for arbitrary three-dimensional geometry by use of flux and collision estimators. In addition, there are special geometry packages (plane, cylinder, sphere, and combinations) and various evaluation programs (area and point detectors for shielding, volume detector for flux integrals and cross-section reduction, correlation package for small effects, plotting program for the geometry test). The basic program requires 18,500 core storage locations. Beyond this a common area, which is needed for storage of cross-sections and geometry data, must be made available. The cross-sections can in part be stored in secondary memories (e.g., ECS in the CDC 6600). A maximum of 350 neutrons can be processed at any one time in one batch. Scattering moments to P_8 are provided for. There is

no pre-programmed limit to the number of groups (upward or downward scattering). The entire program exists in versions for the CDC 6600 and the Siemens 4004 calculators.

The following report describes the individual program sections. Main emphasis is given to the deviations from the initial code, MORSE. The MORSE manual [2] gives information about further details. An input description and a test example for each of the analysis packages are intended to round out the discussions.

II. PROGRAM SYSTEM

The MORSE-K program consists of sub-routine decks for the following tasks: input, generation of random paths, geometry, analysis, and statistics. The core storage used by the input routine is used for neutron storage during the simulation. Before each fundamentally new input the program must be loaded anew. However, small input changes can be carried out through sub-routines written by the user.

In this chapter the standard decks will be described. The majority of their sub-routines have direct predecessors in the MORSE program. Hence the description will be limited to indication of the most significant changes.

The data transfer between individual routines takes place by way of the various common areas. Primarily the structures of blank common were changed. Table 1 gives the named common areas. Table 2 describes the structure of the neutron storage, and Table 3 that of blank common.

1. Input Module

The subroutines of the input module are shown in Table 4. Their status relative to the MORSE code and their main purpose are given. The relations between the sub-routines is shown in Table 5. Table 6 diagrams the sequence of the input. The input description in Chap. III gives the details.

The most far-reaching changes were made in the programs that process the cross-sections. In contrast to the MORSE version referenced here, upward scattering is permitted and the connection with RSYST libraries is permitted. As standard options, there are the capabilities of using different orders of scattering for different materials, of using transport corrections, and of assigning blocks of cross-sections at will to materials. However, the capability for mixing various elements in the program was not developed further. MORSE-K works most effectively when such generation processes take place outside the Monte Carlo calculation (e.g. with RSYST [3]).

A series of parameters was needed for the description of upscatter and the processing of the new cross-section blocks; these were appended to COMMON/LOCSIG. They are described in Table 7.

Table 1. Labeled Common Areas in MORSE-K

Name	Content	Length	Status*	Occurrence
AGECAL	Lifetimes	4	n	BANKR,NBATCH
APOLLO	Adresses and Parameters	100	u	MORSEK, BADMOM, INPUT, SORIN, MORSE BANKR, FBANK, FPROB FSOUR, GETETA, GOMST, GPROB, GSTOR, MSOUR, NXTCC TESTWT, OUTPUT2, OUTPUT3, GETNT, STORNT, SETNT, HELP OUTPT
BANK	Information for collision tape	54	u	MORSEK, INPUT
BNKNMC	Designation of Parameters of collision tape	72	u	MORSEK, INPUT
CHGEOM	Index for geometry package	1	n	MORSEK, INPUT, DIREC, LOOKZ, GEOM, NORML, GOMFLP, JOMIN
DIM1/2/3	Auxiliary storage	27	n	DATIN, FIND, LEGEND, INPUT
FISNEW	Adresses and Data for fission	6	g	MORSEK, INPUT, MORSE, FBANK, FPROB FSOUR, MSOUR, OUTPUT3, SETNT, OUTPT
GEOMA	Regionindex	2	u	MORSEK, GEOM, JOM5 LOOKZ, GOMFLP, JOMIN
GEOMC	Transfer geometry - package random path	13	u	EUCLID, GOMST, GEOM, JOM 5, LOOKZ, NORML, GOMFLP, JOM 14, JOM 15
GEOMH	Hollerith Data	28	u	MSOUR, NORML, GOMFLP, GEM, JOMIN, JOM 11, JOM 16, JOM 17, JOM 12
GEOMT	Entry for JOM 13	1	u	MORSEK, JOM 13, JOMIN

Table 1. Contd.

Name	Content	Length	Status*	Occurrence
GEOM1	NCUESV	1	u	MORSEK, GEOM, NORML, GOMFLP
GEOM4	Present coordinates at block-change	11	u	GEOM, JOM4, JOM5, JM10, NORML, GOMFLP
GEOM7	Present coordinates at section of surface	8	u	JOM5, JOM7
GEOM9	Block addresses	5	u	GEOM, JOM4, JOM5, JOM6, JOM9, JM10, LOOKZ, JOM14, JOM15
GEOM39	Coordinates for determining block	4	u	JOM9, LOOKZ, GEM, JOM13
GEOM56	Block key	4	u	JOM5, JOM6
GEOM77	Direction magnitudes	4	u	JOM5, JOM6, JOM7, LOOKZ, NORML, GOMFLP, JOM14, JOM15.
INOUT	Numbers of the I/O units	2	u	MORSEK, XSEC, GOMFLP, JOM13, JOMIN, JOM16, JOM17, JOM12
JOMIN1	Maximum path in geometry	1	u	MORSEK, GEOM, JOMIN
JOMIN2	Zone number and boundaries	19	u	MORSEK, JM10, LOOKZ JOMIN
JOMIN3	Block numbers, sector information	21	u	MORSEK, JOM4, JOM5, JOM6, JOM9, JM10, LOOKZ, GOMFLP, JOM14
JOMIN8	Adresses of the surfaces	10	u	MORSEK, JOM7, XJOM8, NORML, JOM12
KEANAL	See Table 22	22	n	BANKR, NBATCH, NRUN, STBTCH, STRUN

Table 1 Contd.

Name	Content	Length	Status*	Occurrence
KRES	Parameter(s) for k_{eff} from total run	5	n	OUTPT3,OUTPT
LASTIO	Last quantity of the input area	1	n	ENDINP, SETNT
LOCSIG	Addresses and parameters for cross sections	65	g	MORSEK, INPUT, ANGLES, FIND, GETMUS JNPUT, XSEC, READSG, STORE, LIBTP, GAMGEN COLISN, NSIGTA
MEANS1/2/3	formerly MEANS	18	g	DATIN, BADMOM, ANGLES, Q
MOM1/2	formerly MOMENT	21	g	DATIN, BADMOM, GETMUS, LEGEND, JNPUT, XSEC
NBANK	Beginning of neutron storage	1	n	DATIN, FIND, GETMUS JNPUT, FPROB, FSOUR GETNT, STORNT, SETNT
NCR	Characteristics	31	u	JOM5, JOM6, GEM
NOMIX	Parameter NOMI	1	n	JNPUT, XSEC, READSG
NORMAL	Normal-vector	3	u	ALBEDO, NORML, GOMFLP
NUTRNP	See Table 24 Connection for correlated calculations	11	n	GETNT
NUTRON	Parameters of the current neutron	26	u	MORSEK, INPUT, MORSE, BANKR, FBANK FPROB, FSOUR, GETETA. GOMST, GPROB, GSTOR, MSOUR NXTCOL, TESTWT, ALBEDO, DIREC, OUTPT3 GETNT, STORNT, GETWT OUTPT
PDET	Addresses and data for the detectors	30	n	ENRGYS, SCORIN

Table 1. Contd.

Name	Content	Lengtn	Status*	Occurrence
PERT 1	See Table 23	26		MORSEK, GETNT, MSOUR, OUTPT, OUTPT 3
QAL1/2/3	Early QAL	40	g	DATIN, GETMUS
RES1/2/3	Early RESULT	48	g	DATIN, BADMOM, ANGLES, FIND, JNPUT
RNDOM	Parameter(s) of the random number generator	2	n	MORSEK, INPUT, MORSE, OUTPUT
TITL	Intermediate storage for texts	20	n	INPUT, BIBWQ, XSEC
USER	Addresses and parameters for analysis	23	u	MORSEK, ENRGYS, INPUT
WEIGHT	Addresses and parameters for calculation of weights	6	n	TESTWT, GETWT
XGEM	Addresses for assignment tables geometry media-WQ-media	3	n	MORSEK, INPUT, JNPUT XSEC, GAMGEN, COLISN OUTPUT3, NSIGTA, OUTPUT

* u unchanged
g changed relative to MORSE
n new

Table 2: Neutron Bank (Particle Bank)

Beginning Address	Contents	Meaning
NBANK(1)	ICOMB	
	U	
	V	
	W	
	X	For each parameter NMOST places for a maximum of NMOST neutrons
	Y	
	Z	
	WATE	
	AGE	
	IBLZNT	
NBANK(NFDT) →	X	
	Y	Parameter(s) for first fission neutron
NFDT = 10 · NMOST+1	Z	
	WATEF	
	AGE	
	IG+8000·NAMEX	
	X	Parameters for 2nd neutrons
	Y	
	Parameters for at most NMOST fission neutrons	
	WATEF	
	AGE	
	IG+8000·NAMEX	

IBLZNT = ISIGN(((((NBOUND·64+NZBL)·64+NYBL).64+NXBL)·128+NZ), NSGN)

ICOMB = NREG+2^{6}NMED+2^{16}NAMEX+2^{29}NAME+2^{42}IG

Table 3: Data Areas in Unnamed Common

Beginning addresses	Contents	Length
MXSTR →	Energies	4 · NMTG
	Velocities	
	Source spectrum	
	Weighted source spectrum	
→	Cross sections	
LOCWTS	Average weight in batch	MGPREG
	PATH	MGPREG
	Counter for splitting, Killing und and survival	3 · MGPREG
LOCFWL →	Average weight at the beginning	MGPREG
	Fission weights in batch	2 · MXREG
	Fission weights at the beginning	
LOCEPR →	Weights of the secondary particles	NMGT.MXREG
LOCNSC →	Parameters for weighting of the energy transfer	NMGT.MXREG
	Counter for collisions in groups	NMTG
LOCFSN →	Counter for collisions in groups and geometry-media	NMTG.NGMED
	Fission probabilities	NMTG.MEDIA

Table 3. Contd.

Beginning addresses	Contents	Length
	Fission spectra	NMTG.MFISTP
	Probability for secondary particles	NMTG NGMED
NGEOM →	Geometry data	
NGLAST+1 →	Analysis data	
LMAX+1 →	User data	NLEFT

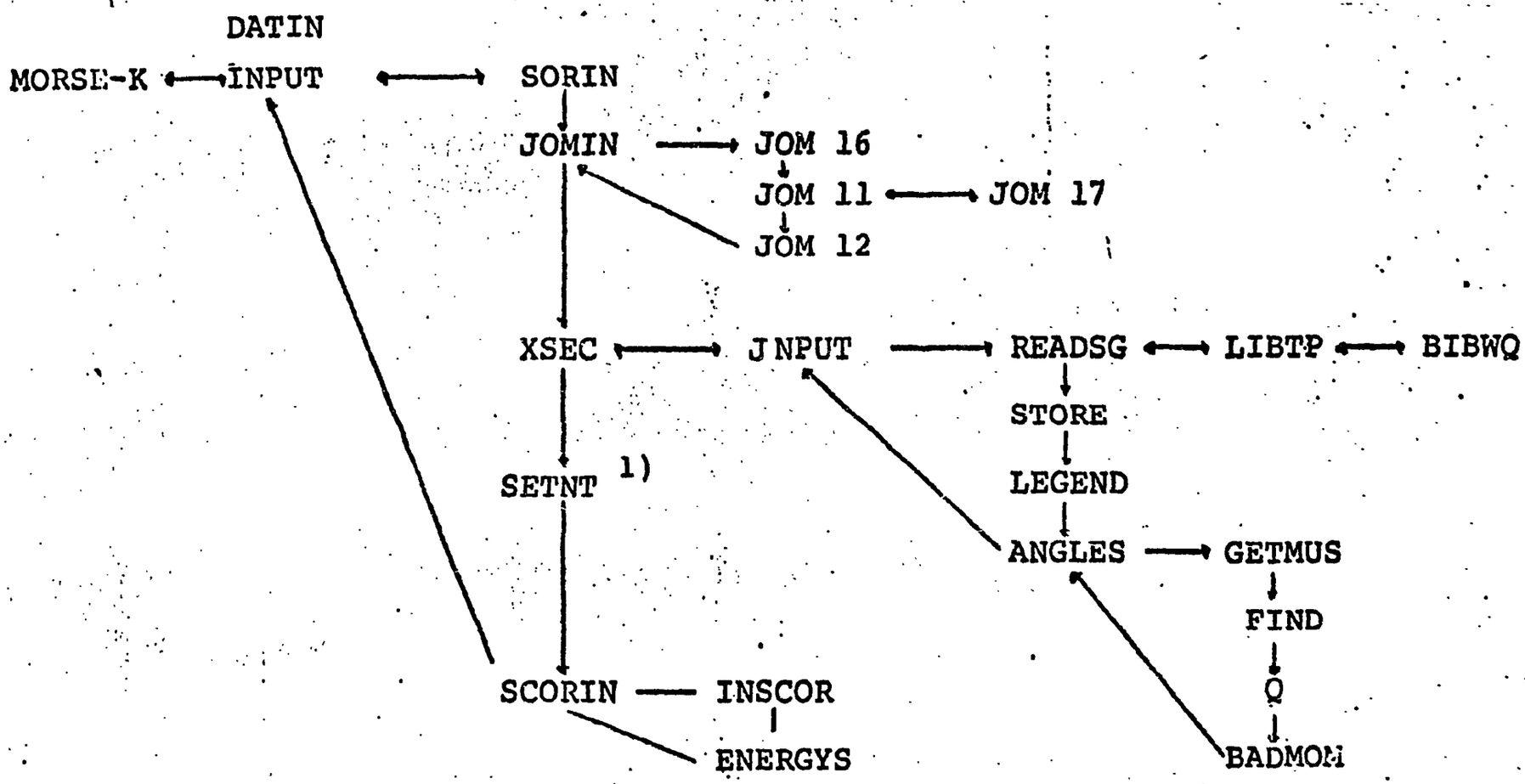
Table 4: The Subroutines for Input to MORSE-K

Name	Status	Function
ANGLES	u	Determination of discrete scattering angles and scattering probabilities by Gaussian quadrature
BADMOM	g	Print-out for bad moments
BIBWQ	n	Reads RSYST data from tape or cards
DATIN	n	Reserves the first place of the neutron storage and places of the auxiliary common. First subroutine
ENDINP	n	Reserves the last place of the neutron storage last subroutine of the input section
ENERGYS	u	Determines boundaries and width of the groups for analysis
FIND	u	Finds the roots of the angle polynomials
GETMUS	u	Calculates moments of the angle polynomials
INPUT	g	Controls the input sequence
INSCOR	u	No function at present
INPUT	g	Controls the cross-section processing
JOMIN	g	Control program for the geometry input.
JOMGEN	n	Reads the input of the general GEOM
JOM11	u	Calculates identification numbers for sectors
JOM12	u	Reads coefficients of the area equations
JOM16	u	Reads boundaries for zones and blocks
JOM17	u	Reads media and Surface assignments
LEGEND	u	Prepares Legendre coefficients for the moment calculations
LIBTP	n	Prepares RSYST data for MORSE-K
Q	u	Calculates polynomials
READSG	g	Controls the reading of the cross sections

Table 4. Contd.

Name	Status	Function
SCORIN	g	Reads the input for the standard analysis
SORIN	g	Reads source spectra and weights
STORE	g	Stores cross-sections in intermediate storage
XSEC	g	Prepares processing and storage of the group data.

Table 5. Relation between Subroutines for Input



1) See Section 2

Table 6: Scheme of Input Sequence

Subroutine	First Card	Last Card	Card contents
INPUT	I1	I5	Problem description
SORIN	I6	I7	Source spectra and weighting
INPUT	I8	I10	Energy boundaries, collision tape random numbers
XSEC	X1	X7	Cross section description
Dependent thereon			
READSG or	X8	X8	Cross sections from cards
LIBTP or	X8-A	X8-A	Identifiers of blocks to be read
BIBWQ and	X8-B	X8-B	RSYST-cross sections from tape or cards
INPUT	X9	X9	Mixture information
INPUT	I11	I19	Parameters for weighting and production of fission and secondary particles
JOMGEN	G A	G Q	General geometry input (May be replaced by another package)
SCORIN	S1	S13	Analysis data

Immediately following, further data may be read in by individual subroutines.

Table 7 New Parameters in COMMON LOCSIG

Parameter	Meaning
IXTAP	Logical Number of the Cross section tape < 0 Tape unblocked = 0 No cross sections on tape > 0 Tape blocked
MCR	Number of blocks to be read from cards < 0 Blocks in RSYST format = 0 No blocks on cards > 0 Blocks in old format
MTP	Number of blocks read from tape
NUS	Number of upscatter groups for secondary particles
NTAB	Maximum length of a column of the cross section blocks
NTABC	Column length to which block is to be reduced (By leaving out activities or by shortening the scattering matrix)
INUS	Total number of upscatter groups
IGSNU INSGU	Starting addresses of the cumulative tables of lengths of individual columns in the cross section blocks

Table 8 lists the structures that can be processed (Reading Programs READSG or BIBWQ). If transport-corrected calculations are to be done (IHT < 0), in the sub-routine LIBTP the cross-sections Σ_T and Σ_{g-g} are replaced by Σ_{TR} and $\Sigma_{g-g}-(\Sigma_T-\Sigma_{TR})$.

There are two possible strategies for processing cross-sections for Monte Carlo calculation. They can be selected by way of the parameter NOMI stored in COMMON/NOMIX. If NOMI ≤ 0, the processing is done the same way as in MORSE. In particular, element cross-sections can be processed into material cross-sections and P_1 development coefficients can be stored. The latter is important, for example, for point detectors. A disadvantage here is the large amount of storage required, especially for large numbers of groups.

In order to avoid this disadvantage, a second processing option was created. It is chosen with NOMI > 0. This requires a special input sequence. The cross-sections must be present as material data, but may still be multiplied by an arbitrary constant (e.g., temperature-dependent density). A separate order of expansion must be given for each material.

The diagram for the cross-section processing sequence is given in Table 9. Table 10 sketches the memory sequence when NOMI ≤ 0 and Table 11 when NOMI > 0.

Immediately before the cross-sections stand some tables with auxiliary parameters for describing the cross-section domains (for instance, length input for the data in the individual groups) and for the matching up of geometry and cross-section materials. The important starting addresses of these regions are listed in common (XGEM). We have:

MXSTR — Starting address of the table that assigns to materials 1 to NGMED one of the NELEM cross-section areas

MSSTR — Starting address of the table of starting addresses of the cross-section regions

NGMED — Largest number with which a material for which cross-sections are read in is designated in the geometry description. The divisions of the common area belonging to this is given in Table 12.

Smaller changes were made in the sub-routines which read in geometry and analysis data. They are discussed in subsections 3 and 4.

The CDC 6600 permits use of Extended Core Storage (ECS). This secondary memory has an access time not much larger than the actual core memory. Hence large data sets can be held there even if they are used frequently. Such data sets are the scattering matrixes of large blocks of cross-sections. They and the associated angles and probabilities may optionally be put in the ECS. This option is selected through the parameter IECS of card X2. The application of this option requires some changes

Table 8: Structure of the Cross Section Blocks read by MORSE-K

Position	Read by READSG MCR > 0	Intermediate storage	Position	Read by BIBWQ MCR < 0 or MTP > 0	Intermediate storage
1	Σ_a	Σ_a	1	Activity 1	
2	$\nu\Sigma_f$	$\nu\Sigma_f$	2	Activity 2	
			⋮	⋮	
			IHT-3	Σ_{tr}	
			IHT-2	Σ_a	Σ_a
			IHT-1	$\nu\Sigma_f$	$\nu\Sigma_f$
IHT = 3	Σ_T	Σ_T	IHT	Σ_T	Σ_T
			IHS-NUS-1	$\sum \Sigma_{g+i\to g}$	
			IHS-NUS	$\Sigma_{g+NUS\to g}$	$\Sigma_{g+NUS\to g}$
			⋮	⋮	⋮
			IHS-1	$\Sigma_{g+1\to g}$	$\Sigma_{g+1\to g}$
IHS	$\Sigma_{g\to g}$	$\Sigma_{g\to g}$	IHS	$\Sigma_{g\to g}$	$\Sigma_{g\to g}$
IHS + 1	$\Sigma_{g-1\to g}$	$\Sigma_{g-1\to g}$	IHS+1	$\Sigma_{g-1\to g}$	$\Sigma_{g-1\to g}$
⋮	⋮	⋮	⋮	⋮	⋮
IHM	$\Sigma_{g-NOS\to g}$	$\Sigma_{g-NOS\to g}$	IHM	$\Sigma_{g-NOS\to g}$	$\Sigma_{g-NOS\to g}$

Table 9: Processing of Cross Sections in MORSE-K by means of JNPUT and Auxiliary Programs

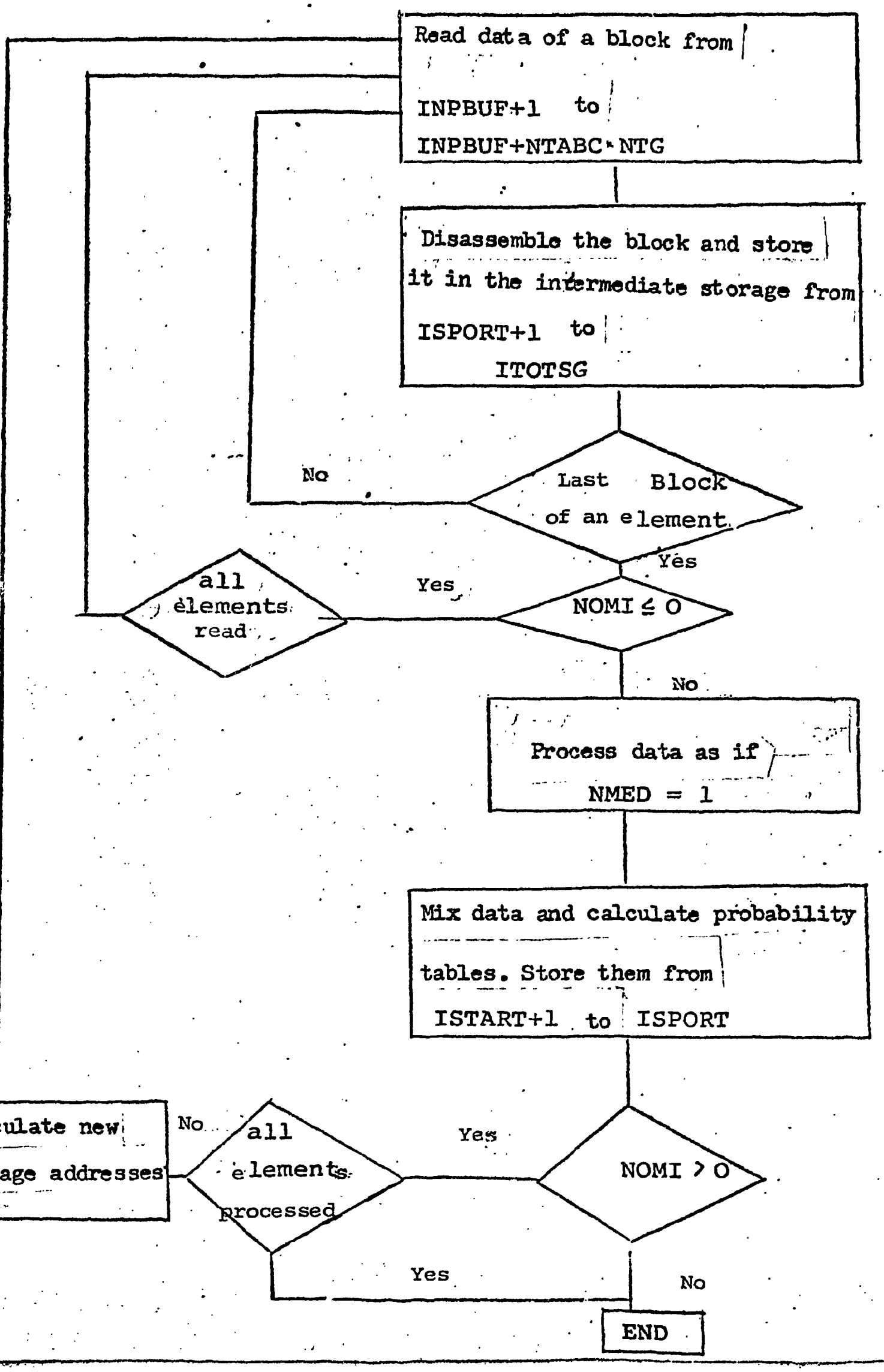


Table 10: Organization of Storage for Cross Sections if NOMI ≤ 0

		Intermediate storage (STORE)		Processing (INPUT)	
Address	Data	Address	Data	Address	Data
INPBUF = ISTART	For one block and all groups data vector	INPBUF		ISTART	For all groups and Medium 1
				ISCCOG	Σ_T
				INABOG	Σ_s
				IGABOG	Σ_s / Σ_T
				IFPORG	Σ_f / Σ_T
				IFNGP	$\nu\Sigma_f / \Sigma_T$
				IFSPOG	$\Sigma_{n\to g}$
				IDSGOG	$\Sigma^n_{g'\to g}$
				IPRBNG	$\Sigma^r_{g'\to g}$
				IPRBGG	$P^n_{g'\to g}$
				ISCANG	$P^r_{g'\to g}$
				ISCAGG	$H^n_{g'\to g}$
				ISTART+ISPORG	$H^r_{g'\to g}$
					Medium 2
					•
					•
					•
		ISPORT ≥ ISTART+INGP*(NTAP+1)	For all elements For all arrangements For each group Scattering cross section from group into other group		•
				ISPORT	Pl coefficients for all media
		ISIGOG	For all elements For all groups Σ_T		⋮
		INFPOG	For all elements For all groups $\nu\Sigma_f$		

ISPORG = NTG+NTG+NTG+NGP+NTS+NGP*NGG
+ NTOT * (2 NSCT+1)

NTOT storage for data of the scattering matrix in the form above

Table 11: Organization of the Cross Section Storage if

NOMI > 0

Read in (READSG)		Intermediate storage (STORE)		Processing (JNPUT)	
Address	Data	Address	Data	Address	Data
	//////		//////	ISTART to ISTARTH ISPORG(1)	Data for Material 1 analogous to Table 10, column 6
INPBUF = ISTART + ISPORG (1)	Material 1 for one block and all groups data vectors				
	//////	ISPORT = INPBUF + NTAB.NMTG	Material 1 Scattering matrix	to ISTART+ ISPORG(2)	Data for Material 2
INPBUF = ISTART + ISPORG (2)	Material 2 9 for one block and all groups data vectors	ISIGOG INFPOG	Σ_T $\nu\Sigma_f$		Data for Material NMED
	//////		////// Material 2		
INPBUF + ISTART + ISPORG (NMED)	~~~ Material NMED		~~~ //////		
			14 Material NMED all coefficients		

ISTART + ISPORG (I) = NSIG (MSSTR+I)

//// unused common area

~~ repeated to NMED-1

Table 12: Storage Areas for Auxiliary Data for Cross Sections

Starting Address	Contents	Length
MXSTR		
	Assignment geometry-cross sections	NGMED
	Blocks for the materials	NELEM
MSSTR	Starting addresses of the material areas	NMED+1
	KFIR	NMED
IRSG	Mixing parameters Card X9	3 • NMIX
INGS	For all groups Starting address of downscatter n	NGP+1
INSG	Starting Address downscatter γ	NGG+1
IGSNU	Starting address upscatter n	NGP
INSGU	Starting address upscatter γ	NGG
ISTART	Material 1	

which were not included in the tables up to now. They will be specially described here since they do not have the same form on all machines. The changes will be comprehensible on the basis of the ECS properties.

ECS variables must be declared separately. They may appear in programs only in the following ways:

(a) in a COMMON statement as elements of an ECS common block
(b) in a CALL or function call as present parameter
(c) in a SUBROUTINE or FUNCTION statement as dummy parameter
(d) in a TYPE ECS statement
(e) in a DIMENSION command

The ECS variables may, at most, be collected in a single COMMON which may contain no other variables.

Data transfer between core memory and ECS goes by way of the subroutine WRITEC (a,b,n) and READEC (a,b,n) where

a = variable or address (array start) in the core memory
b = variable or address in the ECS block
n = integer expression

"n" values will be transferred when either of the two subroutines is called. This causes the areas beginning in the core memory at location "a" and in the ECS at location "b" to be written over. COMMON/EC 1/IECS, INEC, IREC contains the necessary addresses. When

ICES > 0 the starting addresses of the associated ECS areas are stored in blank Common from INEC+1 to INEC+NMED, and hence for all media.

IECS = 0 ECS not needed

ICES < 0 in addition to the starting addresses, an area in blank Common from IREC+1 to IREC+NDSG is kept free, to which the scattering probabilities can be transferred

The ECS option can be used with or without NCMI > 0. For now the use of ECS is intended primarily for NOMI > 0. For this reason the P_L expansion coefficients are not put into ECS in the present version of MORSE-K.

2. Generation of the Random Walks

The random walks of the neutrons are in general generated just as in the MORSE program. Thus the changes made in this part of the program are minor. Table 13 gives a summary of the standard subroutines needed. In addition, the user must supply a few specail programs. These are summarized in Table 14. They will be explained by examples in the analysis packages.

Table 13: Subroutines for Production of Random Walks in MORSE-K

NAME	Status	Function
ALBDO	u	Calculates direction cosines after reflection. Must be fitted to albedo scatterer
COLISN	g	Calculates parameters after collision
DIREC	g	Calculates angle between flight direction and preferred direction for weighting purposes. Must be fitted for special directions. Here for ICHGEM = 1, 2, 4 preferred direction is z-axis, ICHGEM = 3 preferred direction, radius vector.
EUCLID	u	Determines the number of mean free paths between two points
FBANK	g	Stores parameters of the fission neutrons
FPROB	u	Produces fission neutrons
FSOUR	u	Transmits parameters of the fission neutrons to the particle bank
GAMGEN	u	Calculates the energies of secondary particles (except fission)
GETETA	u	Determines flight length from exponential distribution
GETNT	g	Fetches the parameters for the particle to be treated from the particle bank
GETWT	n	Calculates standard weights. Must be replaced for special cases. Here WTAVR is to be read in for each region and each group.
GOMFLP	g	Changes geometry parameters for albedo scattering
GOMST	g	Calculates the coordinates of the next collision point under the assumption of a homogeneous medium for a given flight length
GPROB	u	Controls the production of secondary particles

Table 13, Cont.

NAME	Status	Function
GSTORE	u	Stores parameters of the secondary particles in the particle bank
GTIOUT	g	Determines the energy after a collision using weighting of the scattering energy
HELP	g	Prints error message
MSOUR	u	Controls production of source particles
MORSE	g	Controls production of the random walks
MORSEK	n	Main program. Controls program sequence
NORML	g	Determines the normal to the surface on which albedo scattering takes place
NXTCOL	g	Determines coordinates of the next collision point
OUTPT	g	Calculates average values for source and collision parameters during a batch
RNDM	g	Random number package. The random number generator is written in Assembler, the transformation program in Fortran. XSQR(X1,X2) gives random numbers distributed proportional to x and to (1 - x) in $[0,1\rangle$.
SETNT	g	Prepares particle bank
STORNT	g	Stores parameters in particle bank
TESTWT	g	Checks particle weights against standards

Table 14: Strongly Problem-dependent Subroutines for the Generation of Random Walks in MORSE-K

Name	Calling Programs	Function
BANKR	MORSE	Produces the connections to analysis and statistics programs. Depending on the value of NBNKID various subroutines are called up. if NBNKID = -1 Calls STRUN = -2 Calls STBTCH = 6 Calls ALBDO = 5 Calls RELCOL = -3 Calls NBATCH Inactive call-ups NBNKID = 10 for call-up of TKILL = 9 for call-up of ECUT
	FPROB	Analysis of fission NBNKID = 3
	GSTORE	Analysis of secondary particle production NBNKID = 4 Inactive call-up of GAMLOST by NBNKID =1
	MSOUR	NBNKID = 1 calls SDATA
	NXTCOL	= 7 calls BDRYX
		= 8 calls ESCAPE
	TESTWT	=11 calls RRKILL Inactive call-ups NBNKID =12 Analysis of the weighting = 2 products
NBATCH	BANKR (-3)	Preparation of the results from a batch for analysis and statistics
SOURCE	MSOUR	Production of parameters for source neutrons. Parameters are transmitted by way of parameter brackets (see text)

Table 14, Cont

Name	Calling Programs	Function
STBTCH	BANKR (-2)	Preparation of storage locations for batch statistics and analysis
STRUN	BANKR (-1)	Preparation for the run by set-up of general data
NRUN	MORSE-K	Concluding calculation of average values and their dispersion

Table 15 tries to make clear the logical connections among the subroutines. The most important branch points are indicated.

The parameters of the subroutine SOURCE are the same as in MORSE. For convenience they will be listed here again. The call has the form

```
CALL SOURCE (IG, U, V, W, X, Y, Z, WATE, MED, AG, ISOUR, ITSTR,
                    NGPQT3, DDF, ISBIAS, NMTG)
```

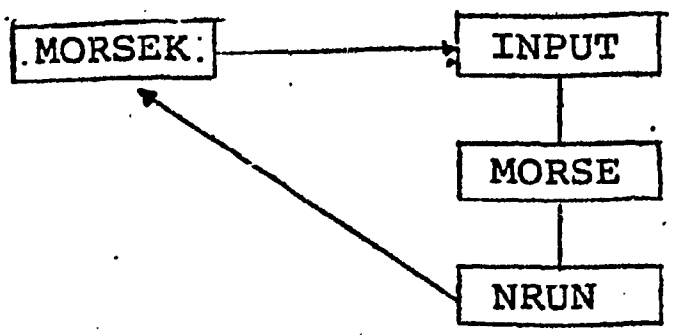
defined as

IG	Energy group of the source particle
U, V, W	Direction cosines of the source particle
X, Y, Z	Spatial coordinates of the source particle
WATE	Weight of the source particle
MED	Medium in which the particle started
AG	Age of the source particle

These parameters may be, but do not have to be, calculated by SOURCE. The following parameters contribute to this:

ISOUR	0	Monoenergetic source in group ISOUR
	0	Source spectrum in blank COMMON (see card 14 of the input)
	$\neq 0$	Particle is a product of a fission (U, V, W isotropic and X, Y, Z coordinates of the fission)
	$= 0$	Particle is a source particle
NGPQT3		Number of groups for which the problem is defined; calculated in INPUT
DDF		Weight correction which normalizes the source spectrum to the number of groups used.

Table 15. Relation between Subroutines for Production of Random Walks



See Table 5

See Table 18

MORSE

SETNT

Preparation for the calculation

BANKR (-1)

40 Preparation for the batch

BANKR (-2)

Print-out of the present random number

MSOUR – SOURCE

FSOUR

LOOKZ

OUTPT (1) (Print-out of the mean source data)

60 GETNT

65 Fit the particle into the system; if the particle cannot be treated then continuation at 165. Otherwise

TESTWT – GETWT

NXTCOL – GETETA – DIREC

EUCLID – GEOM

GOMST

GEOM

NORML

GOMFLP

Table 15, Cont.

Test for

Lifetime [BANKR (10)] Jump to 165

Escape Jump to 165

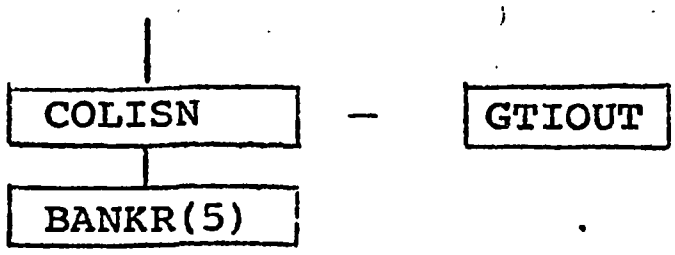
Albedo scattering [ALBDO, BANKR (6)] Jump to 65

Fission [FPROB - FBANK]

Secondary particle [GPROB - GAMGEN]

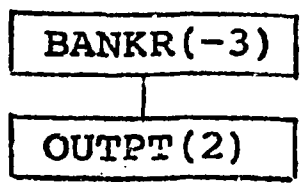
GSTORE

If collision takes place



Jump back to 65

165 If further particles are to be considered then jump to 60



(Print-out of reaction numbers)

If further batches can be considered then jump to 40

Otherwise jump back to MORSEK

ISBIAS 0 No energy weighting; source spectrum begins at (2xNMTG+1) in blank COMMON

0 Energy weighting; source spectrum begins at (3xNMTG+1); WATE must be corrected (see card 14 of the input

NMTG Number of groups (card 13 of the input)

The subroutine GETWT(NUMB) is designed to make it possible to calculate energy-, space-, and possibly also angle-dependent weighting parameters for Russian Roulette and splitting. Weighting functions and weighting parameters must be determined based on the problem. Standards weights WTAVR as function of group and energy are read by INPUT (cards I11 to I13). These suffice for most reactor calculations. In shielding calculations, the weighting function as a rule is strongly space dependent and sometimes direction dependent. The simplest approximation, especially for the calculation of the dose behind thick shields, is an exponential relation of the form

$$\text{WTAVR} = \text{WO(IG,IREG)} \times \exp\left\{\beta \Sigma_T (L - Z)\right\} .$$

The additional constants β and L must be read in, for example, by way of STRUN. Further relations and literature references may be found in reference 4.

A few words remain to be said about the random number generator. It is constructed on the multiplicative congruence method:

$$X_{k+n} = X_k + \lambda^n \pmod P$$

$$\text{RANDOM} = X_{k+n} / P$$

The initial value, X_k, for k = 0 and the multiplier λ can be read in on card I1o. If the parameters there are zero, then they are are determined by the program as

$$\text{RANDIN} = X_o = 6576511432_8$$

$$\text{GENERA} = \quad = 3277244615_8 = 5^{15}$$

The module is determined to a value of $P = 2^{48}$. By the fact that the last bit in X_o was chosen as 0, this corresponds to a significance of 47 bits and a period of 45 bits. The random number sequence can thus also be generated on the CDC 1604 and the IBM 360 (see MORSE manual). The generator has been thoroughly tested by Coveyou [5] and is described in detail by

Schmidt [6]. It is made up of the following commands:

```
        IDENT   RNDM
        ENTRY   FLTRNF
        USE     /RNDOM/
RANDOM  DATA    1717000000065765114328
GENERA  DATA    171700000032772446158
        USE     *
FLTRNF  DATA    0
        SA1     RANDOM
        SA2     GENERA
        DX1     X1*X2
        SA2     UNBIAS
        SB2     X2
        PX6     B2,X1
        SA6     RANDOM
        NX6     B1,X6
        EQ      FLTRNF
UNBIAS  DATA    37178
        END
```

In addition to the random number generator, the random number package includes all the transformation programs of the original package as well as a subroutine XSQR(X1, X2), which generates random numbers that are distributed according to x and 1-x. The use of the random numbers is described in Table 16.

The random number generator for the Siemens 4004 was written by Schweer [12]. It generates the same random number sequences as the CDC generator. Because of the considerably shorter word length of the Siemens 4004 (32 bits), only the first six digits of the random numbers coincide. This corresponds approximately to the number of places that are uncorrelated if pairs of numbers are produced with the generator described. In large calculations, this difference can become noticeable. If only a single parameter of a particle is chosen differently, then, as a rule, other chance chains will result in the following simulations.

Other changes for the Siemens 4004 that are generally required because of the smaller word length are described by Schweer [12].

3. The Geometry Package

The geometry package of MORSE-K contains the five individual decks with which MORSE code can also work. These decks were taken over almost unchanged. They are described in references 2, 7, and 8.

The coupling with the Monte Carlo program is by way of the subroutine JOMIN (input), GEOM (following particles), and LOOKZ (determination of points). These three programs have only control functions in MORSE-K. Depending on the value of the parameter ICHGEM (card I19 of the input), they

Table 16: Use of Random Numbers in MORSE-K

Name	Calling Program	Purpose
FLTRNF	COLISN	1. Determination of new energy 2. Determination of scattering cosine if there is anisotropy
	EPROB	Selection of fission neutrons
	GAMGEN	Determination of the energy of the secondary particle
	GPROB	Selection of secondary particle
	GTIOUT	Weighted determination of the energy after scatter
	TESTW	Determination of the surviving particles, whose weight exceeds the lower limit
	SFLRAF	Transformation according to $[-1, 1)$
	EXPRNF	Exponential distribution by use of rejection technique. At least 2 random numbers are needed.
	POLRN	Sine and cosine of polar angle. Cosine from $[-1, 1)$ at least two random numbers
	AZIRN	Sine and cosine of azimuth angle $\{$Angle from $[0, 2\pi)\}$ at least 2 random numbers
	GTISO	Cosines of a space angle from $[0, 4\pi)$ at least 3 random numbers
	RNMAXF	Maxwell distribution; at least 4 random numbers
	FISRNF	Watt fission spectrum; at least 5 random numbers
	XSQR	Transformation according to x and 1 - x. Two random numbers
EXPRNF	GETETA	Determination of flight length
AZIRN	COLISN	Determination of the azimuth after anisotropic scattering
GTISO	COLISN	Direction after isotropic scattering
	GSTORE	Direction of secondary particles
	MSOUR	Direction of isotropic source particles (For example, fission neutrons)

call up the appropriate subroutines of the packages.

The subroutines are gathered together in Table 17 and explained in keyword fashion. The description of the combinatorial geometry method as given in ref. 7 is included in this report as an appendix.

Geometry subroutines are called by the following programs:

a. INPUT	-	JOMIN
b. EUCLID	-	GEOM
c. GOMST	-	GEOM, NORML, GOMFLP
d. MSOUR	-	LOOKZ

The subroutines NORML and GOMFLP treat albedo-scattering events. They are set up for all geometries. The subroutine DIREC determines the angles between the flight and the preferred directions. The preferred direction is often geometry dependent.

The input for the combinatorial geometry is read by the subroutines JOMCOM and GTVLIN. "Region" and "Media" mean the same thing as in the general geometry package but are structured differently. The concepts "Body" and "Zone" are explained in the appendix, where a description of the possible geometry elements (bodies) is also found.

4. Analysis and Statistics

The outlines of the statistical considerations underlying MORSE-K were developed by Clark and reported in ref. 9. According to this, the transport equation is solved in the form

$$\chi(P)dP = S(P)dP + \int dP'\ \chi(P')\ K(P' \rightarrow P)dP$$

and we solve

$$\lambda = \int F(P')\ \chi(P')dP'$$

using the approximation

$$\tilde{\lambda} = \frac{1}{N} \sum_{i=1}^{N} F(P_i) \qquad P_i \in \chi(P)$$

Table 17: Subroutines in the Geometry Package

Name	Function
General geometry	ICHGEM = 1
GEMGEN	Control program for the determination of the flight path
JOM4	Calculates NCUE when leaving a block
JOM5	Determines medium when NCUE = 0
JOM6	Determines sector for a point
JOM7	Determines NCR (Surface through which the sector is exited)
XJOM8	Calculates for a point the quadratic surface equations (Determines the position of the point relative to the surface)
JOM9	Determines the block in which a point is located
JM1O	Determines the new block after block boundary is crossed
JOM13	Prints error messages
JOM14	Decomposes IBLZN for calculation
JOM15	Forms IBLZN for storage
LOOKGEN	Determines NMED,NREG, and IBLZN for a point
Input programs	
JOMGEN	Control and processing of input
JOM11	Calculates identification of the sectors
JOM12	Reads the coefficients of the surface equations
JOM16	Reads zone and block boundaries
JOM17	Reads MEDIA and SURFACE cards
Plane geometry	ICHGEM = 2
GEMSLB	Determination of the flight path between two points
LOOKSLB	Determination of medium and region of a point
JOMSLB	Input program

Table 17, Cont

Name	Function
Spherical geometry	ICHGEM = 3
GEMSFR	Determination of the flight path between two points
LOOKSFR	Determination of medium and region of a point
JOMSFR	Input program
Cylindrical geometry	ICHGEM = 4
GEMCYL	Control of the flight-path determination
JOM4C	Tests for crossing of z boundary
JOM5C	Tests for crossing of r boundary
JOM6C	Determines region for a point
JOM9C	Determines medium for a point
JOM10C	Determines height interval of a point
LOOKCYL	Determines medium and area for a point
JOMCYL	Input program
Combination geometry	ICHGEM = 5
G1	Follows a particle to the next new zone or to collision point
GG	Calculates the distance between the point and the surfaces of the bodies
LOOKCOM	Calculates the zone of a point
PR	Error routine prints intermediate results
JOMCOM	Controls and processes the input
GENI	Stores input data in unnamed common; prints input
ALBERT	Processes input data for an arbitrary body with six sides (ARB)
GTVLIN	Reads volumes of the regions

where, by the law of large numbers, we have

$$\lim_{N\to\infty} \tilde{\lambda} = \lambda$$

Since in a Monte Carlo calculation N is always finite,

$$\tilde{\lambda} - \lambda = \varepsilon \neq 0$$

The error magnitude is itself a random magnitude. According to the central boundary value theorem, its density should be a normal distribution with zero expectation value and the dispersion

$$\sigma_N^2 = \frac{1}{N}\sigma^2 = \frac{1}{N}\left(\langle x^2\rangle - \langle x\rangle^2\right)$$

A necessary condition for this to hold is that all the sum terms $F(P_i')$ come from the same distribution. This is not self-evident, especially when weighting methods are used. It is therefore tested with the aid of the Batch Concept.

Every Monte Carlo calculation is subdivided into B subcalculations (batches) so that there are N_b source particles simulated in each batch. (We should have approximately $100 \leq N_b \leq 350$.) The batches have weights w_b according to the number of source particles, given by

$$N = N_B = \sum_{b=1}^{B} w_b$$

Then the estimation equation looks like

$$\tilde{\lambda} = \frac{1}{B}\sum_{b=1}^{B}\tilde{\lambda}_b = \frac{1}{N_B}\sum_{b=1}^{B}\frac{w_b}{N_b}\sum_{i=1}^{N_b} F(P_i') \qquad P_i' \in X(P)$$

If a sufficient number of the N_b source particles contribute to $\tilde{\lambda}_b$ ($\tilde{\lambda}_b$ should get a significant contribution from at least 10 particles), then all the $\tilde{\lambda}_b$ stem from the same source distribution. Each batch yields an estimated $\tilde{\lambda}_b$ for its average value. The dispersion may be estimated by considering all the batches:

$$\tilde{\lambda}_b = \frac{1}{N_b} \sum_{i=1}^{N_b} F(P_i)$$

$$\tilde{\lambda} = \frac{1}{N_B} \sum_{b=1}^{B} w_b \tilde{\lambda}_b$$

$$s^2 = \frac{1}{B-1} \sum_{b=1}^{B} (\tilde{\lambda}_b - \tilde{\lambda})^2 = \frac{B-1}{B} G_N^2$$

$\tilde{\lambda}_b$ and $\tilde{\lambda}_b^2$ are calculated after each batch for all magnitudes of interest and summed. At the end of the calculation λ^r and s^2 are then determined from them. This explains the need for the subroutines STRUN, STBTCH, NBTCH, and NRUN, which have already been listed in Table 14. They can be expanded by addition of further subroutines.

The contributions, F, which can be estimated at the event points, P_i, are collected by the subroutine BANKR (NBNKID). The programs that call BANKR as well as the significance of NBNKID are explained in Table 14. Examples for subroutine BANKR are discussed in the separate analysis packages.

Three classes of information may be estimated. For each there exists a package of subroutines. The first gives the basic information for each batch and each run. This includes for each batch the first random number X_o, the average parameter of the source particles, and the distribution of events. Further, at the end of each run, the last random number and several tables are printed out. The tables give information concerning the type of cut-off of the random walks, the distribution of collisions into the energy groups and material regions, and the distribution of splitting and killing. This package is a constituent of each run. In addition, there must be one of the other two packages or one written by the user.

The second package calculates integral values such as k_{eff} and lifetime. (It is not possible to estimate both well if done simultaneously.) Flux and collision estimates are used. From these, an average value is formed:

$$k_{eff} = a\, k_{eff}^{F} + b\, k_{eff}^{C}$$

$$a = \left[s^2 (k_{eff}^{F}) - cov (k_{eff}^{F}, k_{eff}^{S}) \right] / N$$

$$b = \left[s^2 (k_{eff}^{S}) - cov (k_{eff}^{F}, k_{eff}^{S}) \right] / N$$

$$N = s^2(h_{eff}^F) + s^2(h_{eff}^S) - 2\,cov(h_{eff}^F, h_{eff}^S)$$

$$s^2(h_{eff}) = \left[s^2(h_{eff}^F)\, s^2(h_{eff}^S) - cov^2(h_{eff}^F, h_{eff}^S) \right] / N$$

The third package calculates zero-, one-, and two-dimensional magnitudes for various detectors. This package was developed from an early edition of the package SAMBO (Cain in ref. 2).

Table 18 lists the subroutines of the three packages. Table 19 describes the organization of the blank Common needed by package 3.

III. PROGRAM INPUT

The input to any MORSE-K run is made up of five parts, which are each read by different subroutines. They contain information about the particles to be followed (read by INPUT and SORIN), cross-section data, (read by XSEC, INPUT, READSG, LIBTP), the geometry (read by JOMIN, cards designated by G), the detector properties (read by SCORIN), and possibly additional information required by the user (e.g., source data).

User input will be described in the corresponding examples. The MORSE-K input is first read by the subroutine INPUT:

Card I 1 (20 A 4)
arbitrary program identification

Card I 2 (I 5)
NLAST – The number of words in blank common

Card I 3 (15 I 5)
NSTRT – Number of particles in the batch
NMOST – Maximum number of particles which can be stored (350 in MORSE-K Version 1)
NITS – Number of batches
NQUIT – No meaning
NGPQTN – Number of neutron groups to be analyzed
NGPQTG – Number of gamma groups to be analyzed

Table 18: General Subroutines for Analysis and Statistics

Name	Status	Function
PACKAGE 1		
OUTPUT	g	Performs general analysis of the batch data
OUTPT2	g	Prints two-dimensional fields
OUTPT3	g	Performs general analysis of the run
PACKAGE 2		
OUT	n	Prints mean value and dispersion of the k_{eff}
PLOT	n	Plots distribution of the batch results and normal distribution of the end result. Calls up PLOT
STRUN	n	See Table 14
STBTCH	n	
NBATCH	n	
NRUN	n	
PACKAGE 3		
ENDRUN	u	No task
ENRGYS	u	Prepares energy intervals
INSCOR	u	No task
FLUXST	g	Stores batch results for all detectors
REA	n	Carries out regression anlysis (See Chapter VII)
SCORIN	g	Reads input
VAR2	u	Calculation of variances for 2-d fields
VAR3	u	Calculation of variances for 3-d fields
STRUN	n	See Table 14
STBTCH	n	
NBATCH	n	
NRUN	n	

Table 19. Arrangement of the Area for Analysis Data in Blank Common

Address	Name	Length
NGLAST + 1	LABELS	20 * NRESP
LØCRSP	RESP	NRESP * NMTG
LØCXD	EXTR	NEX * NMTG
	XD YD ZD RAD TO FACT	6 * ND
	EXTR	NEXND * ND
LØCIB	IB EP DELE	3 * NE
LØCCØ	CØS	NA
LØCT	T DELT	2 * ND * NT
LØCUD	UD SUD SUD2	3 * ND * NRESP
LØCSD	SD SSD SSD2	3 * ND * NRESP
LØCQE	SUD & SSD Units	20
	QE SQE SQE2	3 * NE * ND
LØCQT	SQE Units	20
	QT SQT SQT2	3 * NT * ND * NRESP
	SQT Units	20

Table 19, cont.

Address	Name	Length
	SQTZW	NK*NE
LOCNKK →	QST SQST SQST2	9*NK*NE
	QNS SQNS SQNS2	
	QAB SQAB SQAB2	
LØCQTE →	QTE SQTE SQTE2	3 * NT * NE * ND
LØCQAE →	SQTE Units	20
	QAE SQAE SQAE2	3 * NA * NE * ND
LMAX →	SQAE Units	20

NMGP – Number of cross-section groups for primary particles

NMTG – Group number of the cross-section blocks

NCOLTP – ≥ 1, a collision tape will be written (requires special BANKR and special analysis programs)

≤ 0, case requires no collision tape

IADJM – > 0, adjoint calculation

≤ 0, forward calculation

MAXTIM – CP time to controlled problem cut-off (sec)

MEDIA – Number of materials for which cross sections are given

MEDALB – Number of Albedo media

$= 0$, no albedo medium in the problem (only material cross sections)

< 0, only albedo medium in problem (no material cross sections will be read)

> 0, transport problem with albedo medium, MEDALB in the geometry

Card I 4 (4I5, 5E10.5)

ISOUR – > 0, monoenergetic source in group ISOUR

≤ 0, source spectrum will be read from cards I 6

NGPFS – Number of source groups leaving group 1

ISBIAS – ≤ 0, No energy weighting of the source

> 0, Weighting of the source spectrum according to card I 7

NOTUSD – Not used

WTSTRT – Initial weight of particles

EBOTN – Lower energy boundary of group NMGP in eV

EBOTG – Lower energy boundary of group NMTG in eV

TCUT – Age to which particles are followed

VELTH – Average particle speed in group NMGP in cm/sec

Card I 5 (7 E 10.4)

XSTRT, YSTRT, ZSTRT — Initial coordinates of source particles

AGSTRT — Age of source particles

UINP, VINP, WINP — Angle cosines of source particles; if all 3 are 0, directions are taken to be isotropic

Card I 6 (7 E 10.4) (needed only when ISOUR ≤ 0)

FS(I) — NGPFS value of normalized, cumulative source spectrum (FS(NGPFS) = 1.0)

I 7 (7 E 10.4) (needed only when ISBIAS > 0

BFS(I) — NGPFS values of the relative significance of space group I

Card I 8 (7 E 10.4)

ENER(I) — Energies (in eV) which correspond to upper group boundaries

Card I 9 describes the collision tape; it may appear only when NCOLTP > 0 and if a corresponding subroutine BANKR has been introduced

Card I 9 (2 I 5, 5 X, 36 I 1, 13 I 1)

NHISTR — Logical number of first tape

NHISMX — Logical number of last tape

NBIND (J), J = 1,36 — A value greater than 0 marks parameters that are to be written on the collision tape (see [2])

NCOLLS(J), J = 1,13 — A value greater than 0 denotes collision types that are to be written on the collision tape (see [2])

Card I 10 (2 Ø 20)

RANDIN - Initial value of pseudo random number sequence

GENERA - Multiplier of random number sequence; both values must begin with 1717. A blank card calls forth an IBM-compatible random number generator. It generates the same pseudo random numbers as the generator in the MORSE code.

The following input cards contain the cross-section information. The sequence read by INPUT is immediately continued. Initially the subroutine XSEC takes over the control of the input. The following cards are expected:

Card X 1 (20 A 4) Arbitrary data description

X 2 (16 I 5)

NGP - Number of groups to which primary particles are assigned

NDS - Number of down-scatters, including in-group scattering (NDS ≤ NGP)

NGG - Number of groups to which secondary particles are assigned

NDSG - Number of down-scatters in the NGG groups (NDSG ≤ NGG)

INGP - Total number of groups of the cross-section blocks (INDS ≤ INGP)

INDS - Total number of down-scatters

NMED - Number of media for which cross sections must be stored

NELEM - Number of elements for which cross sections must be read

NMIX - Number of mixing operations (NELEM x density multiplications)

NCOEF - Maximum allowed number of blocks per element

NSCT - Maximum allowed number of discrete angles (usually NCOEF/2)

ISTAT - ≤ 0 Not defined

> 0 Legendre coefficients will be stored

IECS - = 0 No secondary storage (ECS in the CDC 6600)

> 0 Angles and probabilities will be put into ECS

< 0 Scattering matrixes will also be put into ECS

Card X 3 (16 I 5)

MCR - Number of blocks to be read from cards

> 0 - MCR blocks will be read by READSG

< 0 - |MCR| blocks will be read by BIBWQ

BIBWQ expects IDTF 0 and

RSYST blocks on cards. The structure of the blocks is given by Table 8

MTP - Number of blocks to be read from tape

IXTAP - Logical number of the cross-section tape

CDC Version IXTAP = 17 unblocked

=-17 blocked (or in blocks)

NUS - Number of upscatters of the primary particles

NUSG - Number of upscatters of the secondary particles

NTAB - Length of the blocks to be stored (INDS + INUS + 3 + activities)

NTABC - Length of blocks to be stored (INDS + INUS + 3)

NOMI - > 0 (element x density gives material. Density is read in after the cross-sections for the corresponding element; this saves space! Not possible if ISTAT.GT.0

= 0 Arbitrary composition of the materials from the elements. All elements are read in. The densities follow.

Card X 4 (16 I 5)

IRDSG - > 0 The read-in cross sections are printed out

ISTR	-	> 0	Stored cross-sections are printed out
IFMU	-	> 0	Intermediate results of the calculations are printed out
IMOM	-	> 0	Moments of the angular distribution are printed out
IPRIN	-	> 0	Angles and frequencies are printed out
IPUN	-	> 0	Results of the calculation with false Legendre coefficients are printed out
IDTF	-	> 0	Cross sections have DTF-IV format. They will be multiplied internally by the factor 2 l + 1.
		≤ 0	Cross sections have ANISN format

Card X 5 (16 I 5)

NGMED - Number of media distinguishable in the geometry

Card X 6 (16 I 5)

MEDXS(I) - Number of the cross-section medium which corresponds to the geometry medium (I = 1, NGMED)

Card X 7 (16 I 5)

NSCTT(J) - Number of blocks for cross-section medium J (J = 1, NMED)

The remaining card sequences can be read by various subroutines. The sequence depends on the parameters NOMI, MCR, MTP, and IDTF. If NOMI > 0, then cards for all NMED materials follow: $X8_1$, $X9_1$, ... , $X8_{NMED}$ $X9_{NMED}$. If NOMI ≤ 0, the card sequence becomes $X8_1$, ... $X8_{NMED}$, $X9_1$, ... $X9_{NMIX}$.

The cards X8 can have various formats and meanings. First, it is asked whether cross sections are to be read from cards (MCR > 0) and in which format they are given (IDTF). When MCR > 0 and DTF-IV format is used (IDTF > 0), then

Card X8-A (6 A 8)

HOL(I) - Designation of the block that follows (I = 1.6)

Card X8-B (6 E 12.5)

V(J) - / A block of cross sections, group number INGP, length NTAB

These card sequences are repeated for each block of an element. If MCR > 0 and ANISN format is used (IDTF < 0), then

Card X8 (6 (I2, A1, E9.0))

IN(J) - Repetition factor
T (J) - Repetition sign
} Important only when V(J) = 0

V (J) - Cross section

Cards X8 are read for each group of a block and for each block of an element.

If MCR < 0

Card X8-A (I10, 5I5)

NR - Block identification number

IHM - Length of the block

IGM - Number of groups

IHT - |IHT| position of Σ_T. For IHT < 0, Σ_T is replaced by Σ_{Tr} from position |IHT| -3

IHS - Position for in-group scattering

Card X8-B (6 E 12.5)

V(J) - / A cross-section block of length NTABC with INGP groups

If MTP > 0 and all MCR elements are read, then card X8 is read by BIBWQ:

Card X8 (I 10, 5I5)

NR - Block identification number

IHM - Length of the block

IGM - Number of groups

IHT - > 0 Position of Σ_T

< 0 IHT = - IHT and Σ_T is replaced by the trans-

port cross section Σ_{Tr}. The program expects Σ_{Tr} in position IHT-3. The in-group scattering cross section will be corrected correspondingly

IHS - Position for in-group scattering

For each of the MTP blocks, an X8 card is expected. The blocks themselves are expected in DTF-IV format. Their identification is according to card X8.

Card X9 (2 I 5, E 10.5) read from JNPUT

KM	-	Medium number
KE	-	Element number; element KE is contained in medium KM with a density of RHO. The last element of KM is designated by a negative sign.
RHO	-	Density of the element KE in medium KM

This ends the input of the cross sections. Control is returned to the subroutine INPUT. There the weighting parameters are next read in.

Card I 11 (14 I 5)

NSPLT	-	> 0	Splitting is permitted
NKILL	-	> 0	Russian roulette is permitted
NPAST	-	> 0	Exponential transformation is carried out
NOLEAK	-	> 0	Particles cannot escape from the system; cut-off is possible only by Russian rc lette
IEBIAS	-	> 0	Group transfer is weighted
MXREG	-	> 0	Number of regions for which the weighting parameters are input
		≤ 0	Will be set to 1
MAXGP	-	> 0	Number of groups for which weighting parameters will be input
		≤ 0	Will be set to 1

If NSPLT+NKILL and NPAST are all equal to zero, card I 12 must be left out.

Card I 12 (6I5, 20 X, 2E10.5)

NGP1, NDG, NGP2, NRG1, NDRG, NRG2 — The weighting parameters that follow apply from group NGP1 in steps of NDG to group NGP2 and from region NRG1 in steps of NDRG to region NRG2. If NGP1 is given as equal to zero, the parameters apply for all groups; if NRG1 is given as equal to zero, the parameters apply to all regions.

WTAVE1 — Relative significance of the specified groups in the regions indicated; must be consistent with source weights

PATH — Flight path weighting parameter ($0 \leq PATH \leq 1$)

Card I 13 (6I5, 20 X, 2 E 10.5)

NGP1 — A value less than zero indicates that all cards of type I 12 have been read in

WMUL — WTAVEL x WMUL gives the weight limit for splitting

WDIV — WTAVEL/WDIV gives the weight limit for Russian roulette; if the values are zero, they are set at 6.0

If the group transfer is not to be weighted (IEBIAS $\leq$ 0), the next card to follow will be card I 15.

Card I 14 (7E10.4)

EPROB (IG, NREG) — A set of cards is required for all regions in which IG values of the relative importance of the energy group are given for all groups. New energy-transfer probabilities will be calculated with these values.

$$\overline{\Sigma}_{IG \to IG'} \sim \Sigma_{IG \to IG'} \cdot \text{EPROB (IG', NREG)}$$

Card 15 (14I5)

NSOUR	≤ 0	Source problem
	> 0	Eigenvalue problem, fission neutron source
MFISTP	$= 0$	No fissions permitted
	< 0	-MFISTP fission spectra must be generated by SOURCE
	> 0	MFISTP fission spectra must be read in
NKCALL	> 0	Number of first batch used in k_{eff} calculations. After this batch the source should be finished iterating. In adjoint calculations a spectrum is needed for each medium
NORMF	≤ 0	No eigenvalue is determined
	< 0	Weighting constants and fission weights remain constant from batch to batch
	> 0	At the end of each batch the fission weights are multiplied by the newest estimate of k_{eff}. The relative group weights are changed by the ratio of fission weight to source weight of the last batch.

If fissions do not occur (MFISTP = 0), the next card to follow will be card I 18.

Card I 16 (7E10.4)

FWLO(I) - Average weight of fission neutrons in region I (I = 1, MXREG)

If the energy distribution of the fission neutrons is the same in all media (MFISTP < 0), card I 18 follows

Card I 17 (7E10.4)

FSE(IG, ISPEC) - Fraction of fission neutrons in group IG from the fission spectrum ISPEC (IG = 1, NMTG; ISPEC = 1, MFISTP)

Card I 18 contains the initial weight of the secondary particles in all groups and in each region. If no production of secondary particles is expected, card I 19 follows directly.

Card I 18 (7E10.4)

GWLO(IG, NREG) - Initial weight of the secondary particles in group IG and region NREG. For each region a new set of cards I 18 is used (IG = 1, NGPQTN or NGPQTG; NREG = 1, MXREG) NGPQTG applies to adjoint problems.

Card I 19 (I5)

ICHGEM - Indicates which geometry package is to be used

1	General GEØM. Input control in JOMGEN
2	Plane geometry. Input control in JOMSLB
3	Spherical geometry. Input control in JOMSFER
4	Cylindrical geometry. Input control in JOMCYL
5	Combination geometry. Input control in JOMCOM

After this card, the input control first goes to the geometry package and after that to the analysis package SCØRIN. The necessary input cards are described in the following section. For the geometry input the original English text is taken directly.

The input for the general geometry package is read by the subroutine JOMGEN. It consists of the following cards:

CARD GA (I5, 5X, A6, 1X, A7) Hollerith left adjusted

NSTAT - flag to indicate material media only if 1 and both region and material media if 2.

SEX - sex of the programmer. (Select one from MALE, FEMALE, or blank indicating uncertain.)

STATUS - marital status of programmer.

CARD GB (2A4, A3, 5(E10.5, A1)

DUMMY(3) - hollerith characters not used.

FIN(I) - zone boundaries in increasing order along the X-axis.

BCD(I) - flag to indicate end of input if blank, comma means to continue.

Repeat CARD GB if more than five boundaries along the X axis are needed.

CARD GC - same as CARD GB except for Y axis.

Repeat CARD GC if more than five boundaries along the Y axis are needed.

CARD GD - same as CARD GB except for Z axis.

Repeat CARD GD if more than five boundaries along the Z axis are needed.

CARD GE (A4, A2, 3I5)

BCD1 - hollerith ZONE

BCD2 - dummy

[illegible] specify the zone as being the NXZNOth in the X direction, NYZNOth zone in the Y direction, and NZZNOth zone in the Z direction.

NXZNO -

NYZNO -

NZZNO -

CARD GF(2A4, A3, 5(E10.5, A1)

DUMMY(3) - hollerith characters not used.

FIN(I) - block boundaries in increasing order along the X axis.

BCD(I) - flag to indicate end of input if blank, comma means to continue.

Repeat CARD GF if more than five boundaries along the X axis are needed.

CARD GG - same as CARD GF except for Y axis.

Repeat CARD GG if more than five boundaries along the Y axis are needed.

CARD GH - same as CARD GF except for Z axis.

Repeat CARD GH if more than five boundaries along the Z axis are needed.

CARDS GI to GO describe the geometry for a block and must be included for each block in the zone.

CARD GI (A4, A2, 3I5)

BCD1 - hollerith BLOC

BCD2 - dummy

NXBND - integers which specify the block as being the

NYBND - NXBNDth in the X direction, the NYBNDth in the

NZBND - Y direction, and the NZBND in the Z direction.

CARD GJ (3A4, 10(I5,A1)

NAM2 - hollerith MEDI

DUM(2) - dummy

INP(I) - a list of media sector by sector in the block

BCD(I) - flag to indicate end of input if blank, a comma means to continue.

Continuation with 12(I5,A1) format is permissible.

CARD GK (3A4, 10(I5,A1))

NAM2 - hollerith SURF

DUM(2) - dummy

INP(I) - a list of quadratic surfaces appearing in the block. Numbers must appear in the order the surfaces are described on CARD GQ.

BCD(I) - flag to indicate end of input of blank, a comma means to continue.

Continuation of CARD GK in 12(I5,A1) format is permissible.

CARD GL (A4, A2, 18I3)

S1 - hollerith SECT

DUM - dummy

IND(I) - the designation of each sector which describes the position of the sector relative to quadratic surfaces.

+1: sector is on positive side of surface,

-1: sector is on negative side of surface,

0: surface is not needed to define sector.

There must be a CARD GL for each sector and references to quadratic surfaces must be in same order as they are listed on CARD GQ.

CARD GM (3A4, 10(I5,A1)

NAM2 - hollerith REGI

DUM(2) - dummy

INP(I) - a list of regions sector by sector in the block.

BCD(I) - flag to indicate end of input if blank, a comma means to continue.

Continuation with 12(I5,A1) format is permissible.

CARD GN (3A4, 10(I5,A1))

NAM2 - hollerith SURF

DUM(2) - dummy

INP(I), BCD(I) } same as for CARD GK except for region input instead of material input.

CARD GO(A4, A2, 18I3)

S1 - hollerith SECT

DUM - dummy

IND(I) - same as for CARD GL except for region input instead of material input.

Repeat CARDS GI to GØ for each block.

CARD GP (I5, 16A4, A2)

NØBD - total number of quadratic surfaces in the entire system.

DUM(I) - hollerith characters ignored by the code. (Helpful in identifying input at a later time.)

CARD GQ (4(E10.5, A4, 1X, A1))

CØF(J) - coefficient of the term

BCD1(J) - hollerith indicating which term of the equation. XSQ, YSQ, ZSQ, XZ, YX, YZ, XY, ZX, YZ, X, Y, Z, or blank are the possibilities.

BCD2(J) - a flag which indicates the quadratic equation continues. Any non-blank character ends the field. The next function must start on new card.

Repeat CARDS GQ until all surfaces have been described.

A sample of the input is given in ref /2/

The input for the general geometry package is read by the subroutine JOMSLB. It requires for this purpose the following cards:

CARD GA (I5, E10.5)

MED - medium with Z as lower bound (>0)

Z - lower limit of medium MED.

Repeat CARD GA for all boundaries with the last card containing MED = 0 and the boundary of the system.

CARD GB (E10.5)

Z - lower limit of region boundary. Region numbers are assigned in consecutive order starting with 1 and Z must be in increasing order.

Repeat CARD GB for all region boundaries

End CARD GB input with a blank card. If no region geometry is desired a blank card is required.

CARD GC (4E10.5)

XL - lower boundary of system in X direction.

XU - upper boundary of system in X direction.

YL - lower boundary of system in Y direction.

YU - upper boundary of system in Y direction.

The present version provides room for 25 medium boundaries and 20 region boundaries

The input for spherical geometry is read by the subroutine JOMSFER. It consists of the following cards:

CARD GA (I5, E10.5)

MED - medium number interior to R (>0)

R - outer radius of sphere or spherical shell containing MED.

Repeat CARD GA for all radii.

End CARD GA input with blank card.

CARD GB (E10.5)

R - region radius of sphere or spherical shell containing regions. Region numbers are assigned in consecutive order starting with 1, and R must be in increasing order.

Repeat CARD GB for the number of regions.

End CARD GB input with blank card. If no regions are desired, a blank card must be used to signal no region geometry.

The present version provides for 25 radii.

The input for the cylinder-geometry package is read by the subroutine JOMCYL. It requires the following cards:

CARD GA (I5, 5X,'A8)

NREGIN - flag to indicate material media (=1) or both region and material media (=2).

SEX - sex of programmer.

CARD GB (E10.5)

R - radii of the cylindrical shells describing the material media in ascending order.

Repeat CARD GB until all radii have been input.

End CARD GB input with a blank card.

CARD GC (E10.5, 12I5/8I5)

H - upper height of medium M(I) (>0).

Cylinders assumed to start at H = 0.

M(I) - media for the cylindrical shells for this height.
Repeat CARD GC until all height intervals have been input.
End CARD GC input with a blank card or if there are more than 12 radial intervals, 2 blank cards.
CARD GD (E10.5) omit if NREGIN = 1

RG - radii of the cylindrical shells describing the region geometry in ascending order.

Repeat CARD GD until all region geometry has been input.
End CARD GD with a blank card.
CARD GE (E10.5, 12I5/8I5) omit if NREGIN = 1

HG - upper height of region MG(I).

MG(I) - region numbers for the cylindrical shells for this height.

Repeat CARD GE until all height intervals have been input.
End CARD GE input with a blank card or if there are more than 12 radial intervals, 2 blank cards.

The present version provides room for 25 radii and 25 heights for both region and material description

The input instructions for the combinatorial geometry are:

The combinatorial geometry input data is read by the JOMCOM subroutine, except for the region volumes VNØR(I), which are read by the GTVLIN subroutine whenever IVØPT = 3. For clarity of terminology, the terms "regions" and "media" have essentially the same meaning as in the Ø5R Geometry Package, but are constructed in a different manner. The term "zone" is the same as the "region" as defined in the original combinatorial geometry package. The term "body" has the same meaning as in the original combinatorial geometry package.

CARD CGA (2I5, 10X, 10A6)

IVØPT - option which defines the method by which region volumes are determined; if

IVØPT = 0, volumes set equal to 1.,

IVØPT = 1, concentric sphere volumes are calculated

IVØPT = 2, slab volumes (1-dim.) are calculated, *

IVØPT = 3, volumes are input by card,

IDBG - if IDBG > 0, subroutine PR is called to print results of combinatorial geometry calculations during execution. Use only for debugging,

JTY - alphanumeric title for geometry input (columns 21-80).

CARDS CGB (2X, A3, 1X, I4, 6E10.3)

One set of CGB cards is required for each body and for the END card (see Table B IV). Leave columns 1-6 blank on all continuation cards.

ITYPE - specifies body type or END to terminate reading of body data (for example BOX, RPP, ARB, etc.). Leave blank for continuation cards,

IALP - body number assigned by user (all input body numbers must form a sequence set beginning at 1). If left blank, numbers are assigned sequentially. Either assign all or none of the numbers. Leave blank for continuation cards,

FPD(I) - real data required for the given body

CARDS CGC (2X, A3, I5, 9(A2, I5))

Input zone specification cards. One set of cards required for each input zone, with input zone numbers being assigned sequentially.

*Not operational.

IALP - IALP must be a nonblank for the first card of each set of cards defining an input zone. If IALP is blank, this card is treated as a continuation of the previous zone card,

IALP = END denotes the end of zone description.

NAZ - total number of zones that can be entered upon leaving any of the bodies defined for this input region (some zones may be counted more than once). Leave blank for continuation cards for a given zone. (If NAZ ≤ 0on the first card of the zone card set, then it is set to 5). This is used to allocate blank common,

Alternate IIBIAS(I) and JTY(I) for all bodies defining this input zone.

IIBIAS(I) - specify the "ØR" operator if required for the JTY(I) body,

JTY(I) - body number with the (+) or (-) sign as required for the zone description

CARDS CGD (14I5)

MRIZ(I) - MRIZ(I) is the region number in which the "I^{th}" input zone is contained (I = 1, to the number of input zones). Region numbers must be sequentially defined from 1.

CARDS CGE (14I5)

MMIZ(I) - MMIZ(I) is the medium number in which the "I^{th}" input zone is contained (I = 1, to the number of input zones). Medium numbers must be sequentially defined from 1.

CARDS CGF (7E10.5) (omit if IVØPT ≠ 3)

VNØR(I) - volume of the "I^{th}" region (I = 1 to MXREG, the number of regions).

Note: If ENDRUN is used to obtain collision density and track length per unit volume estimate of fluence, then a data

statement in ENDRUN must give a relationship between region and media. In this case only one medium may be in a region.

The following cards are read by the Standard Analysis Package by way of the subroutine SCØRIN:

Card S 1 (20A4) Arbitrary text for description of the analysis; not stored!

Card S 2 (14I5)

ND – Number of detectors

NNE – Number of energy intervals of the analysis which correspond to primary particles

NE – Total number of energy intervals to be analyzed, $\leq$ NMTG

NT – $> +1$ Number of time intervals. For each detector, a set of NT values is read in

$-1 \leq NT \leq 1$ set at 0

$< -1 -$ NT is the number of time intervals. The same interval arrangement is assumed for all detectors. Only one set is read in.

NA – Number of angle intervals to be analyzed for

NRESP – Number of detector response functions.

NEX – Number of user-required storage areas in blank common of length NMTG. Initial address is LORRSP + NRESP x NMTG

NEXDN – Number of storage areas required by user in blank common of length ND. Initial address LOGXD + 6(ND)

NKK > 0 Flux integrals are calculated for ND regions and also rates, the latter by means of a regression analysis

≤ 0 No meaning

Input Required on CGB Cards for Each Body Type

Card Columns Body Type	ITYPE 3-5	IALP 7-10	Real Data Defining Particular Body 11-20	21-30	31-40	41-50	51-60	61-70	Number of Cards Needed
Box	BØX	IALP is assigned by the user or by the code if left blank.	Vx	Vy	Vz	H1x	H1y	H1z	1 of 2
			H2x	H2y	H2z	H3x	H3y	H3z	2 of 2
Right Parallele-piped	RPP		Xmin	Xmax	Ymin	Ymax	Zmin	Zmax	1
Sphere	SPH		Vx	Vy	Vz	R	-	-	1
Right Circular Cylinder	RCC		Vx	Vy	Vz	Hx	Hy	Hz	1 of 2
			R	-	-	-	-	-	2 of 2
Right Elliptic Cylinder	REC		Vx	Vy	Vz	Hx	Hy	Hz	1 of 2
			R1x	R1y	R1z	R2x	R2y	R2z	2 of 2
Ellipsoid	ELL		V1x	V1y	V1z	V2x	V2y	V2z	1 of 2
			L	-	-	-	-	-	2 of 2
Truncated Right Cone	TRC		Vx	Vy	Vz	Hx	Hy	Hz	1 of 2
			L1	L2	-	-	-	-	2 of 2
Right Angle Wedge	WED		Vx	Vy	Vz	H1x	H1y	H1z	1 of 2
			H2x	H2y	H2z	H3x	H3y	H3z	2 of 2
Arbitrary Polyhedron	ARB		V1x	V1y	V1z	V2x	V2y	V2z	1 of 5
			V3x	V3y	V3z	V4x	V4y	V4z	2 of 5
			V5x	V5y	V5z	V6x.	V6y	V6z	3 of 5
			V7x	V7y	V7z	V8x	V8y	V8z	4 of 5
			Face Descriptions (see note below)						5 of 5
Termination of Body Input Data	END								

Note: Card 5 of the arbitrary polyhedron input contains a four-digit integer for each of the six faces of an ARB body. The format is 6(1X, I4), beginning in col

Card S 3 (6E10.4)

X, Y, Z, VOL, V1, V2 } Identifying numbers for detectors. ND such cards are required. The meaning of the data is determined by the analysis programs. VOL is stored as 1/VOL. The addresses of the data for detector I are

LOCXD+I, LOCXD+ND+1, ..., LOCD + 5*ND + I.

Card S 4 (20A4)

Units of the integral, nondirectional, and total doses (e.g., neutrons)

Card S 5 (20 A4)

Designation of detector I

Card S 6 (7E10.4)

Detector response function for detector I for all groups.

Card pairs S 5 and S 6 are read in for each detector. If NE $\leq$ 1, card S 9 follows here.

Card S 7 (20 A4)

Units of the spectral data

Card S 8 (14I5)

Numbers of the groups in which the lower boundaries of the analysis intervals are located. If NNE $>$ 0, the corresponding number must be exactly NGPQTN. The last number is always NMTG.

If NT $\leq$ 1, card S 11 follows here.

Card S 9 (20 A4)

Units of time distributions

If NE $\leq$ 1, card S 11 follows here.

Card S 10 (20 A4)

Units of the time-dependent spectra

If only one time interval is being considered ($|NT| \leq 1$), card S 12 follows here.

Card S 11 (7E10.4)

XT(I,J) - Upper boundaries of the time intervals in increasing sequence ($J = 1, |NT|$). For each detector a new card series S 11 is required. If $NT < -1$, a single series, which is attributed to all detectors, is sufficient.

If $NA \leq 1$, the regular input ends here.

Card S 12 (20 A4)

Units of the angle spectra

Card S 13 (7E10.4)

XA(I) - Upper boundaries of the angle intervals. The cosine is given. I runs from 1 to NA. Hence XA(NA) is equal to 1.0.

Here ends the regular input. Further input sequences can be implemented by the user. Subroutines which are often used for this purpose are, in the order in which they first appear, ISCOR for additional analysis data, SOURCE for source data, and SDATA for special detector data.

IV. CRITICALITY CALCULATIONS

1. Description

Criticality values are simultaneously estimated with a flux estimator and a collision estimator. The results of both estimations are averaged taking account of their variances. All three results are printed out. Reference 13 gives a description of the estimators.

In addition to the criticality values, the average generation time and lifetime are also determined. It turns out that it is undesirable to determine both types of values from the same neutron histories. In the case at hand, the emphasis was placed on the criticality determination. The lifetimes are merely by-products.

All estimations are done within the batches and as averages over the batch results. It is possible to delay the averaging of the batch results until the estimated onset of the fundamental mode (Card I 15, quantity NKCALL).

2. Test Example

The value of k_{eff} is to be calculated for the axially symmetric reactor shown in Fig. 1 (early model of the Incore Thermionic Reactor [10]). Regions 1 to 6 form the reactor core. They are divided into thermionic and driver zones, which are each subdivided into regions with different moderator concentrations.

Four-group cross sections were used in P_o and P_1 approximation. The calculations were compared with DOT-II [11] calculations. Results are given in Table 20. With the input described here, the results can be described by the k_{eff} value

$$k_{eff}^{comb} = 1.2407 \pm 1.179\ \%$$

Table 20: Comparison of 4-Group Calculations

Method	k_{eff}	σ^z	Calculation Time (min)
DOT 2, PO	1.276		8
DOT 2, P 1	1.244		8
MORSE K, Po	1.273	± 0.006	30
MORSE K, P1	1.237	± 0.005	30
	1.2407	± 0.014	3.2

Two subroutines must be mentioned---the source program SOURCE and the analysis program BANKR.

The program SOURCE generates isotropic source neutrons which may be distributed in x-y-z or r-ϕ-z. The source spectrum must be read in by way of SORIN. SOURCE needs the following input data for determination of the spatial distribution:

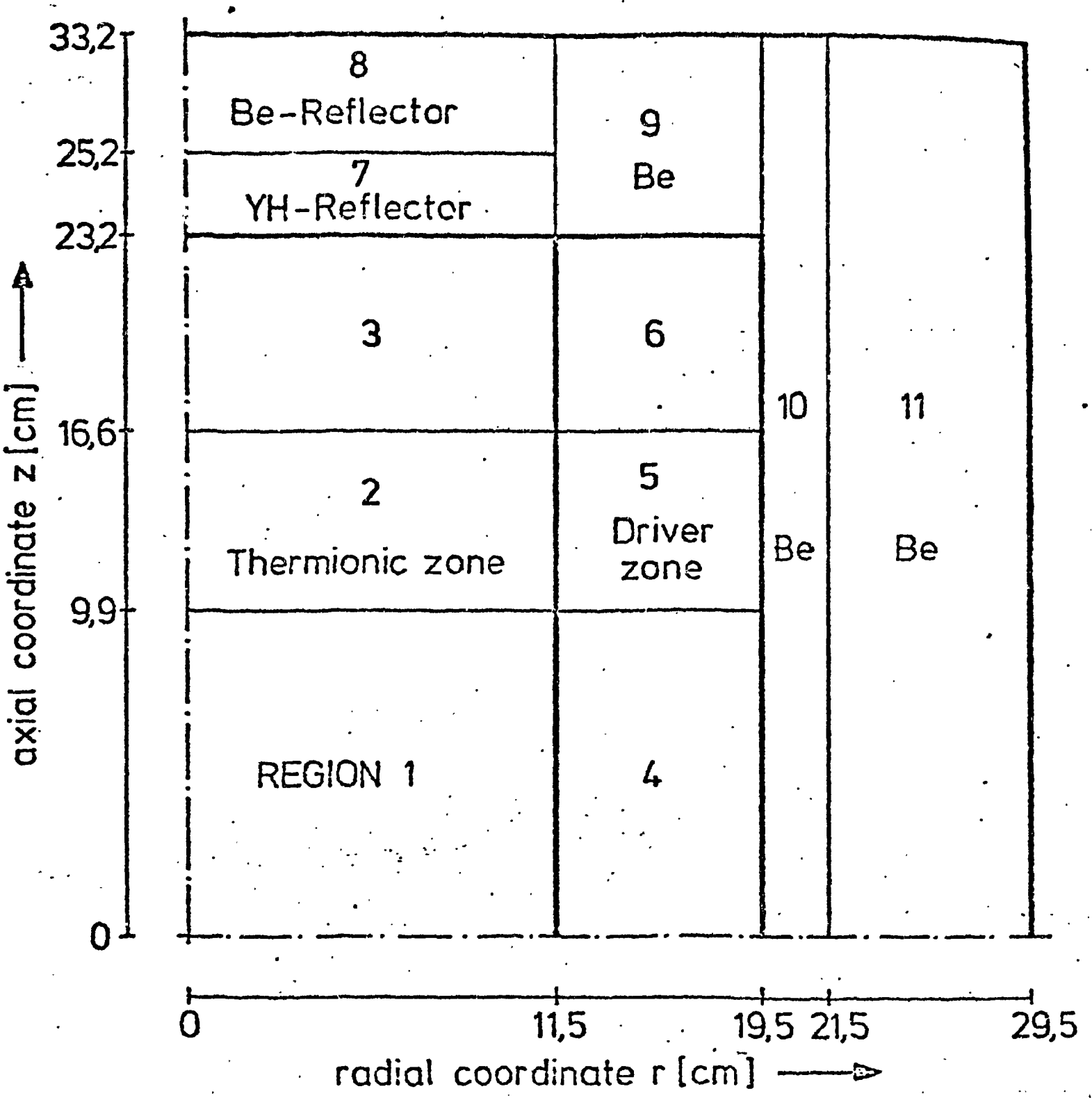


Figure 1 Geometry of the Incore Thermionic Reactor

Card SO 1 (5I5)

NR - Number of r or x interval boundaries $\leq$ 16

NPHI - Number of ϕ or y interval boundaries $\leq$ 16

NZ - Number of z interval boundaries $\leq$ 16

IGEO - $\leq$ 0 x-y-z geometry

> 0 r-φ-z geometry

IDR - Number of source particles whose coordinates and weight are to be printed out

Card SO 2 (7 E 10.3)

R(I) - NR interval boundaries for x or r

Card SO 3 (7 E 10.3)

PHI(I) - NPHI interval boundaries for φ

Card SO 4 (7 E 10.3)

Z1(I) - NZ interval boundaries for z

Card SO 5 (7 E 10.3)

FR(I) - NR-1 values of the density distribution in the r or x space. (In r space the values will be multiplied by $R^2(I+1) - R^2(I)$, in x space by $R(I+1) - R(I)$ to replace the corresponding frequency)

Card SO 6 (7 E 10.3)

FPH(I) - NPHI-1 values of the density distribution in φ or y space. They will be multiplied by $\Delta\varphi$ or Δy, respectively.

Card SO 7 (7E10.3)

FZ(I) - NZ-1 values of the density distribution in z space. (Will be multiplied by Δz)

The program BANKR addresses itself to the following events: (a) source particles, (b) splitting, (c) collision, (d) leaving a material region, (e) albedo scattering, (f) escape of a particle, and (g) killing. The events are processed directly.

BANKR requires the analysis programs of package 1 and package 2 (see Table 18).

The subroutines SOURCE and BANKR are listed below. The common AGECAL contains fission age, age, and their variances. Common KEANAL contains estimations of k_{eff} and their variances according to the flux and collision estimators, as well as their combination. Information is stored for particles, the current batch, and the state of the run.

Table 21: The Parameters of Common KEANAL

Name	Significance
RKFLX	Contribution of the flux estimator for all particles of a batch
VARFLX	Variance of RKFLX
VARCOL	Variance of FTOTL
RKOVAP	Covariance of RKFLX and FTOTL
RKHF	Contribution of the flux estimator for one particle
RKHC	Contribution of the collision estimator for one particle
SWATE	Source weight
FKICSM	k value of collision estimation for a batch
FKIFSM	k value of flux estimation for a batch
FKISM	k value combination for a batch
VAICSM	Variance of FKICSM
VAIFSM	Variance of FKIFSM
VAISM	Variance of FKISM
VAOCSM	k values and variances for the run
VAOFSM	
RKOVSM	
FKOCSM	
FKOFSM	
FNK	
IKCALC	Numbers of the first batch with fundamental mode
IBATCH	Running batch numbers
INITS	Auxiliary variable

SUBROUTINE SOURCE

```
      SUBROUTINE SOURCE(IG,U,V,W,X,Y,Z,WATE,MED,AG,ISOUR,ITSTR,NGPHT3,DD
     1F,ISBIAS,NHTG)
C*****ISOTROPIC SOURCE
C*****SPACE DEPENDENT XYZ OR R PHI Z(IGEO.GT.0)
C*****SOURCE ENERGY BIASING ALLOWED
      COMMON WTS(1)
      COMMON/APOLLO/ XTRA1(28),IO,IN,XTRA2(69)
      DIMENSION R(16),FR(15),Z1(16),PHI(16),FZ(15),FPH(15)
      DATA II /5/
      IF(II)88,88,99
   99  II=0
C     SOURCE INPUT
      READ(IN,1000)NR,NPHI,NZ,IGEO,IDR
      READ(IN,1001)(R(I),I=1,NR)
      READ(IN,1001)(PHI(I),I=1,NPHI)
      READ(IN,1001)(Z1(I),I=1,NZ)
      IF(IGEO.LE.0) WRITE(IO,2000)
 2000 FORMAT(29HTHIS IS A X - Y - Z SOURCE    )
      WRITE(IO,2001)NR,NPHI,NZ,IGEO,IDR
 2001 FORMAT(5H-NR =,I5,9H  NPHI =,I5,7H  NZ =,I5,9H  IGEO =,I5,I7,
     *34H SOURCE PARTICLES WILL BE PRINTED    )
      WRITE(IO,2002) (  R(I),I=1,NR)
      WRITE(IO,2002) (PHI(I),I=1,NPHI)
      WRITE(IO,2002) ( Z1(I),I=1,NZ)
      NR=NR-1
      NZ=NZ-1
      NPHI=NPHI-1
      WRITE(IO,2003)
 2003 FORMAT(44H-SOURCE CONTRIBUTIONS OF SPATIAL INTERVALLS    )
      READ(IN,1001)(FR(I),I=1,NR)
      READ(IN,1001)(FPH(I),I=1,NPHI)
      READ(IN,1001)(FZ(I),I=1,NZ)
      WRITE(IO,2002) ( FR(I),I=1,NR)
      WRITE(IO,2002) (FPH(I),I=1,NPHI)
      WRITE(IO,2002) ( FZ(I),I=1,NZ)
 2002 FORMAT(8E15.4)
        S=0.
      DO 1 I=1,NR
      FR(I)=FR(I)                                *(R(I+1)-R(I))
      IF(IGEO.GT.0) FR(I)=FR(I)*(R(I+1)+R(I))
      S=S+FR(I)
    1 CONTINUE
      FR(1)=FR(1)/S
      IF(NR.EQ.1) GO TO 7991
       DO 2 I=2,NR
      FR(I)=FR(I)/S+FR(I-1)
    2 CONTINUE
 7991 CONTINUE
      S=0.
      DO 3 I=1,NZ
      RX=(Z1(I+1)+Z1(I))*.5
      RX1=(Z1(I+1)-Z1(I))
      FZ(I)=FZ(I)                    *RX1
      S=S+FZ(I)
    3 CONTINUE
```

```
      FZ(1)=FZ(1)/              S
      IF(NZ.EQ.1) GO TO 7992
      DO 4 I=2,NZ
      FZ(I)=FZ(I)/S+FZ(I-1)
    4 CONTINUE
 7992 CONTINUE
      S=0.
      DO 5 I=1,NPHI
      S=S+FPH(I)
     1  *(PHI(I+1)-PHI(I))
    5 CONTINUE
      FPH(1)=FPH(1)/S
      IF(NPHI.EQ.1) GO TO 7994
      DO 6 I=2,NPHI
      FPH(I)=FPH(I)/S+FPH(I-1)
    6 CONTINUE
 7994  CONTINUE
      WRITE(6,1920)
       IC=NMTG+NMTG
      IF(ISBIAS.GT.0) IC=IC+NMTG
   88 CONTINUE
       Q=FLTRNF(B)
      DO 20 I=1,NMTG
      IF(Q-WTS(IC+I))10,10,20
   10 IG=I
      GO TO 30
   20 CONTINUE
   30 IF(ISBIAS.LE.0) GO TO 70
      IF(I-1) 40,60,65
   60 WATE=WATE*WTS(IC-NMTG+1)/WTS(IC+1)
      GO TO 70
   65 WATE=WATE*(WTS(IC-NMTG+I)-WTS(IC-NMTG+I-1))/(WTS(IC+I)-WTS(IC+I-1
     *))
   70 CONTINUE
      IF(ITSTR) 40,50,40
   40 RETURN
   50 CONTINUE
      AR=FLTRNF(AR)
      DO 200 I=1,NR
      IF(FR(I)-AR)200,21,21
   21 K=I
      GO TO 22
  200 CONTINUE
      K=NR
   22 CONTINUE
      RAD=R(K)+(R(K+1)-R(K))*FLTRNF(RX)
      AR=FLTRNF(RX)
         DO 31 I=1,NPHI
      IF(FPH (I)-AR)31,32,32
   32 K=I
      GO TO 33
   31 CONTINUE
      K=NPHI
   33 CONTINUE
      PHI0= PHI(K)+(PHI(K+1)-PHI(K))*FLTRNF(RX)
```

```
      SUBROUTINE  SOURCE



      IF(IGEO.GT.0) PHI0=PHI0*6.283186
   37 AR=FLTRNF(RX)
      DO 43 I=1,NZ
      IF(FZ(I)-AR)43,44,44
   44 K=I
      GO TO 45
   43 CONTINUE
      K=NZ
   45 ZZ=Z1(K)+(Z1(K+1)-Z1(K))*FLTRNF(RX)
      IF(IGEO.GT.0) GO TO 700
      X=RAD
      Y=PHI0
      GO TO 701
  700 CONTINUE
      X=RAD*COS(PHI0)
      Y=RAD*SIN(PHI0)
  701   CONTINUE
      Z=ZZ
      IF(IDR.GT.0) WRITE(6,1919)X,Y,Z,WATE
 1920  FORMAT(22H-INITIAL SOURCE POINTS  ,//,
     15X,*      X          *,5X,*          Y          *,5X,*          Z          *,
     25X,*     WATE        *)
 1919 FORMAT(4(5X,E15.5))
 1001 FORMAT(7E10.3)
 1000 FORMAT(5I5)
      IDR=IDR-1
      RETURN
      END
```

SUBROUTINE BANKR

```
      SUBROUTINE BANKR(NBNKID)
      COMMON /APOLLO/ AGSTRT,DOF,DEADWT(5),ETA,ETATH,ETAUSD,UINP,VINP,
     1 WINP,WTSTRT,XSTRT,YSTRT,ZSTRT,TCUT,XTRA(10),
     2                                  IO,I1,MEDIA,IADJM,ISBIAS,
     3 ISOUR,ITERS,ITIME,ITSTR,LOCNTS,LOCNCL,LOCEPR,LOCNSC,LOCFSN,
     4 MAXGP,MAXTIM,MEDALB,MGPREG,MXREG,NALB,NDEAD(5),NEWNM,NGEOM,
     5 NGPOT1,NGPOT2,NGPOT3,NGPOTG,NGPOTN,NITS,NKCALC,NKILL,NLAST,NMEM,
     6 NNGP,NMOST,NMTG,NOLEAK,NORMF,NPAST,NPSCL(13),NQUIT,NSIGL,NSOUR,
     7 NSPLT,NSTRT,NXTRA(10)
      COMMON/NEUTRON/NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD,
     1WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,BLZNT,BLZON,AGE,OLDAGE
      COMMON/FISNEU/ FTOTL,FWATE,NFISTP,NFISH,NFPT,WATEF
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKTCSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,IBATCH,INITS
      COMMON/AGECAL/ FAGE,AGEL,VFAGE,VLAGE
      COMMON/XGEOM/NXSTR,NSSTR,NGMED
      COMMON WTS(2)
      DIMENSION NSIG(1)
      EQUIVALENCE( WTS(1),NSIG(1))
      DATA NTERM/0/
      DATA RKCOL/0.0/
      DATA PFAGE,PLAGE,FAGE,AGEL,VFAGE,VLAGE/0.0,0.0,0.0,0.0,0.0,0.0/
      DATA OLA/0.0/
      NBNK=  NBNKID
      IF (NBNK) 100,100,140
  100 NBNK=  NBNK + 5
      GO TO (104,103,102,101),NBNK
  101 CALL STRUN
      RETURN
  102 NBAT=  NITS - ITERS
      NSAVE=  NMEM
      CALL STATCH(NBAT)
C     NBAT IS THE BATCH NO. LESS ONE
      RETURN
  103 CALL NBATCH(NSAVE)
C     NSAVE IS THE NO. OF PARTICLES STARTED IN THE LAST BATCH
      RETURN
  104 CALL NRUN(NITS,NQUIT)
C     NITS IS THE NO. OF BATCHES COMPLETED IN THE RUN JUST COMPLETED
C     NQUIT .GT. 1 IF MORE RUNS REMAIN
C           .EQ. 1 IF THE LAST SCHEDULED RUN HAS BEEN COMPLETED
C           IS THE NEGATIVE OF THE NO. OF COMPLETE RUNS, WHEN AN
C                 EXECUTION TIME KILL OCCURS
      RETURN
  140 CONTINUE
      GO TO(1,2 ,20,20,5,30,7,8,20,20,11,20,20),NBNK
C     SOURCE WEIGHT
    1 SWATE=SWATE+WATE
      RETURN
C      SPLITTING
    2  NTERM=NTERM-1
       ...
C     REAL COLLISION
    5 RKHC=RKHC+WATEF
```

SUBROUTINE BANKR

```
      CALL NSIGTA(IGO,MEDOLD,TSIG,PNAB)
      IF(OLA.EQ.0.0) OLA=OLDAGE
      PAB=1.-PNAB
      PLAGE=PLAGE+WTBC*PNAB*(AGE-OLA)
      GO TO 30
C     BOUNDARY CROSSING
    7 CONTINUE
      IF(OLA.EQ.0.0) OLA=OLDAGE
      GO TO 20
C     ESCAPE AND R.R. KILL
    8   NTERM=NTERM+1
C     ADD CONTRIBUTION FROM LAST FLIGHT
      GO TO 30
   30 IF(MEDOLD.EQ.1000)  GO TO 40
      PNUF=WTS(LOCFSN+(MEDOLD-1)*NMTG+IGO)
      RKHF=RKHF+ETAUSD*WTBC*PNUF
      IF(NBNK.EQ.5)PFAGE=PFAGE+ WATEF*(AGE-OLA)
   40 IF(NTERM.LE.0) RETURN
C     TERMINATE HISTORY
      RKCOL=RKCOL+RKHC
      RKFLX=RKFLX+RKHF
      VARCOL=VARCOL+RKHC*RKHC
      RKOVAR=RKOVAR+RKHF*RKHC
      VARFLX=VARFLX+RKHF*RKHF
      FAGE=PFAGE+FAGE
      AGEL=PLAGE+AGEL
      VFAGE=VFAGE+PFAGE*PFAGE
      VLAGE=VLAGE+PLAGE*PLAGE
      PFAGE=0.0
      PLAGE=0.0
      OLA=0.0
      RKHF=0.
      RKHC=0.
       NTERM=0
      RETURN
   11 NTERM=NTERM+1
      GO TO 40
   20 RETURN
      END
```

3. Input

```
11 MATERIALIEN ITR TEST PROBLEM 60K MORSEK
3000
 200  212    3    1    4    0    4    4    0    0  250   11   20
   0    2    0    0       1.0      0.025         0.0      8.E-4   9.78E+5
       5.0         0.0         5.0          0.0          0.0          0.0
       0.9         1.0
 1.05E+7    1.00E+4    4.65E+1    1.0

 4 GRUPPEN WQ ITR
   4    4    0    0    4    4   11   11   11    2    1    0
  18    0    0    0    0    7    7    1
   0    0    0    0    0    0    1
  11
  11    2    3    4    5    6    7    8    9   10    1
   1    2    2    2    2    2    2    1    1    1    2
 P0 ZONE11 ZELLE
  .6106E-05  0.          .4591E+00   .4422E+00  0.          0.
0.           .3018E-04  0.          .6866E+00   .6617E+00   .1873E-01
0.           0.          .1198E-03  0.          .6839E+00   .6479E+00
  .2484E-01  0.          0.          .6841E-03  0.          .6979E+00
  .6968E+00   .3392E-01  0.          0.
   1   -1       1.0
 P0 ZONE2 5.0 ZELLE
  .2476E-02   .3063E-02   .3063E+00   .2775E+00  0.          0.
0.           .1950E-01   .1497E-01   .6845E+00   .5958E+00   .2618E-01
0.           0.          .3574E-01   .3082E-01   .6823E+00   .5452E+00
  .6779E-01   .1174E-03  0.          .8963E-01   .1197E+00   .1055E+01
  .9656E+00   .1014E+00   .1465E-02   .2577E-05
P1 ZONE2 5.0 ZELLE
0.           0.          0.          .7801E-01  0.          0.
0.           0.          0.          0.          .2327E+00   .7507E-02
0.           0.          0.          0.          0.          .2272E+00
  .2499E-01   .2373E-05  0.          0.          0.          0.
  .1616E+00   .3559E-01   .7907E-04   .7509E-08
   2   -1       1.0
```

15 more blocks follow

Next card: Card I 11

```
  0    1    0    0    0   11    4
  1    1    4    1    1   11        1.5       0.09        0.3
 -1
  1   -1    1    1
   1.243     1.243     1.243     1.243     1.243     1.243     0.0
     0.0       0.0       0.0       0.0
  1
```

```
    2       MALE
X-ZONE              -40.    ,       40.
Y-ZONE              -40.    ,       40.
Z-ZONE             -32.5    ,     +32.5
ZONE        1       1       1
X-BLOCK             -40.    ,     40.
Y-BLOCK             -40.    ,     40.
Z-BLOECKE             -32.5,            0.,          9.9,         16.5,         23.1,
      25.5 ,           32.5
BLOCK       1       1       1
MEDIA                 20
REGIONS               11
BLOCK       1       1       2
MEDIA                   1,      4,      10,     11, 1000
SURFACES                1,      2,       3,      4
SECTOR -1   0   0   0
SECTOR +1  -1   0   0
SECTOR   0 +1  -1   0
SECTOR   0   0 +1  -1
SECTOR   0   0   0 +1
REGIONS                 1,      4,      10,     11
SURFACES                1,      2,       3
SECTOR -1   0   0   0
SECTOR +1  -1   0   0
SECTOR   0 +1  -1   0
SECTOR   0   0 +1  -1
BLOCK       1       1       3
MEDIA                   2,      5,      10,     11, 1000
SURFACES                1,      2,       3,      4
SECTOR -1   0   0   0
SECTOR   1 -1   0   0
SECTOR   0 +1  -1   0
SECTOR   0   0 +1  -1
SECTOR   0   0   0 +1
REGIONS                 2,      5,      10,     11
SURFACES                1,      2,       3
SECTOR -1   0   0   0
SECTOR +1  -1   0   0
SECTOR   0 +1  -1   0
SECTOR   0   0 +1  -1
BLOCK       1       1       4
MEDIA                   3,      6,      10,     11, 1000
SURFACES                1,      2,       3,      4
SECTOR -1   0   0   0
SECTOR   1 -1   0   0
SECTOR   0 +1  -1   0
SECTOR   0   0 +1  -1
SECTOR   0   0   0 +1
REGIONS                 3,      6,      10,     11
SURFACES                1,      2,       3
SECTOR -1   0   0   0
SECTOR +1  -1   0   0
SECTOR   0 +1  -1   0
SECTOR   0   0 +1  -1
```

```
BLOCK      1      1      5
MEDIA                7,     9,    10,    11, 1000
SURFACES             1,     2,     3,     4
SECTOR -1   0   0   0
SECTOR  1  -1   0   0
SECTOR  0  +1  -1   0
SECTOR  0   0  +1  -1
SECTOR  0   0   0  +1
REGIONS              7,     9,    10,    11
SURFACES             1,     2,     3
SECTOR -1   0   0   0
SECTOR +1  -1   0   0
SECTOR  0  +1  -1   0
SECTOR  0   0  +1  -1
BLOCK      1      1      6
MEDIA                8,     9,    10,    11, 1000
SURFACES             1,     2,     3,     4
SECTOR -1   0   0   0
SECTOR  1  -1   0   0
SECTOR  0  +1  -1   0
SECTOR  0   0  +1  -1
REGIONS              8,     9,    10,    11
SURFACES             1,     2,     3
SECTOR -1   0   0   0
SECTOR +1  -1   0   0
SECTOR  0  +1  -1   0
SECTOR  0   0  +1  -1
     *       CYLINDRIC SURFACES
          1.XSQ                1.YSQ     -129.96          $
          1.XSQ                1.YSQ     -372.49          $
          1.XSQ                1.YSQ     -453.69          $
          1.XSQ                1.YSQ     -858.49          $
     5     2     4     1   100
0.          6.514        11.55       15.91       19.04
        0.0          1.0
0.0         9.9          16.5        23.1
.1044         .174          1.214      .66241
1.0
0.1044256 0.06552132 0.0602497
```

4. Results---Description of the Output Edit

The results have already been substantially described in the second section of this chapter. Hence here we describe the output edit.

The first page of the output contains the content of input cards I 1 to I 10. The output is self-explanatory. The real and weighted source spectra are given as frequency distributions. The source weights can be calculated from the associated densities.

The second page contains the information from cards X 1 to X 7. They also are self-explanatory. Then the starting addresses of the individual material data are calculated and printed out.

Card X 4 allows us to print out various detailed information concerning the cross sections. The tables of probabilities for scattering, absorption, secondary particle production, and fission are standard. Also given are the probabilities for scattering into other groups. If there is up-scatter from group IG, then scattering to group IG - J is given immediately following. For this case we must have IG+I $\leq$ NMTG and J $<$ NUS.

The information about the last needed storage locations closes the cross-section output. Then follows the content of cards I 11 to I 18. It is self-explanatory.

Depending on the selection of the parameter ICHGEM on card I 19, geometry parameters from various sub-programs are read and printed out. Here, too, the input is self-explanatory. First and last positions of the area in blank common occupied by geometry data are also listed.

If analysis package 3, described in Table 18, is used, the next page will contain the contents of cards S 1 to S 13. Here, too, the output is self-explanatory.

Then follow the print-out of the space remaining in blank common, the time required for the input calculations, and the space available for storage of particle data.

This ends the reproduction of the input. The further output consists of production data, results, and data which the user causes to be listed.

The following is listed automatically for each batch: The initial random number, the average parameters of the source particles, the distribution of events, and the calculation time. In addition, we may have such parameters as source data, reaction rates, or k_{eff} values (in the present case: k_{eff} values, lifetimes, total neutron weight before fission, weight of the fission neutrons, number of fission neutrons. At the end of all the batches, the total time used for simulation is printed.

The further output serves for clarification of the results and their statistical interpretation.

Analysis package 2 collects the batch results and compares them against a normal distribution. Analysis package 3 permits the output edit of the results in the form of one, two, and three parameter fields according to the specifications from cards S 1 to S 13.

The following statistical information concludes the output: cause, number, and weight of the cut-offs of histories, distribution of the scattering events according to both groups and material zones and to groups alone, as well as corresponding distributions of splitting and killing events. The printing of the total time used ends the output of a MORSE-K run.

V. POINT DETECTOR

1. Description

The phase-space coordinates which describe a particle before a collision or source event are used by this estimator to estimate the flux at one or more point detectors. To this end the weight of the particle is multiplied by the probability that after the event the particle will be moving toward the point (will move in $d\Omega$ about Ω), that it will reach the detector without further collision, and that it will there contribute to the measured magnitude. Hence the following quantities must be known or calculated:

a) Angle between flight direction and detector direction

b) Probability for scattering into this direction, i.e., angular distribution of the source particles

c) Probability of scattering into the individual energy groups at the given scattering angle, i.e., energy distribution of the source particles

d) Separation between event point and detector point

e) Number of mean free paths between the event point and the detector point

The quantities a, b, d, and e are constants at each estimation point. The quantity c is variable. We can either calculate expectation values for the contributions to all groups or, on the other hand, we can add in the entire contribution of any energy groups by means of a random number. The first method is chosen in the subroutine for this example.

2. Test Example

The neutron spectrum behind a 45-cm-long LiH cylinder of 19.5 cm radius is calculated. The calculation is done with six groups in P_3 approximation. The detectors are located at the points

X = 0.0	Y = 0.0	Z = 100.0
X = 0.0	Y = 10.0	Z = 100.0
X = 0.0	Y = 0.0	Z = 150.0

The calculations are very slow since, at each collision point, complicated calculations are necessary for the estimation. Hence weighting is needed. The following weighting steps are described:

a) Source weighting and constant parameters for RR
b) Source weighting and fitted parameters for RR
c) Source weighting and continuous weights for RR
d) as in (c) and path length stretching
e) like (d) and non-escape

For this purpose a source subroutine and, for c to e, a new GETWT were needed. Both are taken up in the listings. The parameters for the exponential weighting were determined from the consideration that in the example at hand the transport is determined mainly by particles of the first group. From the literature we find that for exponential weighting the exponential increase in the weighting function is of the form $\beta\Sigma_T$, where β is to be taken between 0.7 and 0.8.

The subroutine GETWT as used for the weightings c to e is

```
      SUBROUTINE GETWT(N)
      COMMON /NUTRON/ NAME,NAVEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDNT,WTBC,IBLZN,BLZON,AGE,OLDAGE
      COMMON/WEIGHT/ NWT,MGP,WT,WL,WA,OMU,WM,WD
      COMMON WTS(1)
      DIMENSION BET(6)
      DATA(BET(I),I=1,6)/0.124,0.180,0.227,3*0.253/
      WA=WTS(IWT)*EXP(-Z*BET(IG))
      WL=WA*0.167
      WT=WA*0.5
      RETURN
      END
```

SUBROUTINE SOURCE

```
      SUBROUTINE SOURCE(IG,U,V,W,X,Y,Z,WATE,M,B,IS,IT,NGP,D,ISR,NMTG)
      COMMON WTS(1)
C     CALCULATION FOR SOURCE AREA
      CALL XSQR(X1,X2)
      A=19.5*X1
      Y=0.0
      Z=FLTRNF(R)
      WATE=0.5
   51 CALL GTISO(U,V,W)
      IF(Z.GT.0.5) GO TO 60
      IF(W.GT.0.) GO TO 50
      R=FLTRNF(R)
      IF(R.GT.0.2) GO TO 51
      WATE=WATE*5.
   50 WATE=WATE*0.6
   60 WATE=WATE*2.
      WATE=D*WATE
      IF(IS) 90,90,95
   95 IG=IS
      RETURN
   90 CONTINUE
      IF(ISB) 100,100,150
  100 IWT=2*NMTG
      GO TO 200
  150 IWT=3*NMTG
  200 R=FLTRNF(R)
      DO 250 I=1,NGP
      IF(R.LE.WTS(I+IWT)) GO TO 300
  250 CONTINUE
  300 IG=I
      IF(ISR) 500,500,350
  350 IF(I-1) 500,400,450
  400 WATE=WATE*WTS(2*NMTG+1)/WTS(3*NMTG+1)
      RETURN
  450 WATE=WATE*(WTS(2*NMTG+I)-WTS(2*NMTG+I-1))/
     1           (WTS(3*NMTG+I)-WTS(3*NMTG+I-1))
  500 RETURN
      END
```

```
      SUBROUTINE BANKR(NBNKID)
C  DO NOT CALL EUCLID FROM BANKR(7)
      COMMON /APOLLO/ AGSTRT,DOF,DEADWT(5),ETA,ETATH,ETAUSD,UINP,VINP,
     1 WINP,WTSTRT,XSTRT,YSTRT,ZSTRT,TCUT,XTRA(10),
     2                                    I0,I1,MEDIA,IADJM,ISBIAS,
     3 ISOUR,ITERS,ITIME,ITSTR,LOCWTS,LOCFWL,LOCEPR,LOCNSC,LOCFSN,
     4 MAXGP,MAXTIM,MEDALB,MGPREG,MXREG,NALB,NDEAD(5),NEMM,NGEOM,
     5 NGPOT1,NGPOT2,NGPOT3,NGPOTG,NGPOTN,NITS,NKCALC,NKILL,NLAST,NMEM,
     6 NMGP,NMOST,NMTG,NOLEAK,NORMF,NPAST,NPSCL(13),NQUIT,NSIGL,NSOUR,
     7 NSPLT,NSTRT,NXTRA(10)
      NBNK= NBNKID
      IF (NBNK) 100,100,140
  100 NBNK= NBNK + 5
      GO TO (104,103,102,101),NBNK
  101 CALL STRUN
      RETURN
  102 NBAT= NITS - ITERS
      NSAVE= NMEM
      CALL STATCH(NBAT)
C   NBAT IS THE BATCH NO. LESS ONE
      RETURN
  103 CALL NBATCH(NSAVE)
C   NSAVE IS THE NO. OF PARTICLES STARTED IN THE LAST BATCH
      RETURN
  104 CALL HELP(4HBANK,1,1,1,1)
      RETURN
  140 GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13) NBNK
C  NBNKID   COLL TYPE    BANKR CALL       NBNKID   COLL TYPE    BANKR CALL
C     1       SOURCE     YES (NSOUR)        2       SPLIT       NO  (TESTW)
C     3       FISSION    YES (FBANK)        4       GAMGEN      NO  (GSTORE
C     5       REAL COLL   YES (NORSE)       6       ALBEDO      YES (NORSE)
C     7       BDRYX       YES (NXTCOL)      8       ESCAPE      YES (NXTCOL
C     9       E-CUT       NO  (NORSE)      10       TIME KILL   NO  (NORSE)
C    11       R R KILL    NO  (TESTW)      12       R R SURV    NO  (TESTW)
C    13       GAMLOST     NO  (GSTORE)
    1 CALL SDATA
      RETURN
    2 RETURN
    3 RETURN
    4 RETURN
    5 CALL RELCOL
      RETURN
    6 RETURN
    7 RETURN
    8 RETURN
    9 RETURN
   10 RETURN
   11 RETURN
   12 RETURN
   13 RETURN
      END
```

```
      SUBROUTINE SDATA
C
C     THIS VERSION IS FOR POINT DETECTORS LOCATED AT (XD,YD,ZD)
C      AND FOR AN ISOTROPIC POINT SOURCE
C
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EBOTN,EBOTG,
     1 TCUT,I0,I1,IADJR,NGPUT1,NGPUT2,NGPUT3,NGPUTG,NGPUTN,NITS,NLAST,
     2 NLEFT,NMSP,NMTG,NSTRT
      COMMON /PDET/ ND,NME,NE,NT,NA,NRESP,NEX,NEAND,NEND,NDNR,NTNR,NTNE,
     1 NAME,NTNDER,NTNEND,NANEND,LOCRSP,LOCXD,LOCIB,LOCCO,LOCT,LOCUD,
     2 LOCSD,LOCNE,LOCRT,LOCOTE,LOCOAE,LMAX,EFIRST,EGTOP,JKS,LOCNKA
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,IBLZN,BLZON,AGE,OLDAGE
      COMMON EN(1)
      IQ=  LOCUD
      DO 5 I=1,ND
      IA=  LOCXD + I
      XE=  EN(IA)
      YE=  EN(IA+ND)
      ZE=  EN(IA+2*ND)
      A=XE-X
      B=YE-Y
      C=ZE-Z
      SD2=A*A+B*B+C*C
      DS=SQRT (SD2)
      MEDIUM=NMED
      N=1
      CALL EUCLID(N,X,Y,Z,XE,YE,ZE,DS,IG,ARG,0,MEDIUM,IBLZN,NREG)
      CON=  WATE * EXP(ARG)/12.56637/SD2
      COS=C/DS
      AGI =  AGE + DS/EN(NMTG+IG)
      ICA=ICK+1
1000  FORMAT(1H ,2I9,5E15.4)
      CALL FLUXST(I,IG,CON,AGI,COS,1)
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RELCOL
C
C     THIS VERSION IS FOR POINT DETECTORS LOCATED AT (XD,YD,ZD)
C
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EBOTN,EBOTG,
     1 TCUT,I1,I1,IADJM,NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
      COMMON /PDET/ ND,NME,NE,NT,NA,NRESP,NEX,NEXND,NEND,NDNR,NTNR,NTNE,
     1 NANE,NTNDNR,NTNEND,NANEND,LOCRSP,LOCXD,LOCIR,LOCCO,LOCT,LOCOD,
     2 LOCSD,LOCDE,LOCOT,LOCQTE,LOCQAE,LMAX,EFIRST,EGTOP,NKK,LOCNKK
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,IBLZN,BLZON,AGE,OLDAGE
      COMMON BL(1)
      DIMENSION NL(1)
      EQUIVALENCE (BL(1),NL(1))
      DATA  NEST/1/,  FNEST/1./
C     NEST * FNEST ARE THE NO. OF ESTIMATES TO BE MADE TO EACH DETECTOR
C     NELDS ISTAT .EQ. 1, NEX .EQ. 1, NEXEND .EQ. 1
      DO 125 I=1,ND
      IA=  LOCXD + I
      XE=  BL(IA)
      YE=  BL(IA+ND)
      ZE=  BL(IA+2*ND)
      A=   XE - X
      B=   YE - Y
      C=   ZE - Z
      SD2=A*A+B*B+C*C
      DS= SQRT(SD2)
      COS=C/DS
      THETA=  (A*UOLD + B*VOLD + C*WOLD)/DS
      IGOLD=  IGO
      IGO=  NGPQT3
      IF(IGO.LE.NGPQT1) IGO=NGPQT1
      IA=  LOCRSP + NRESP*NMTG + 1
      CALL PTHETA(NMED,IGOLD,IGO,THETA,BL(IA),NMTG,-1)
      PSUM=  0.
      IA=  IA - 1
      DO 5 IL=IGOLD,IGO
      IAL=IA+IL
      BL(IAL)=ABS(BL(IAL))
5     PSUM=  PSUM + BL(IAL)
      DO 15 IL=IGOLD,IGO
      IAL=IA+IL
      BL(IAL)=BL(IAL)/PSUM
20    MEDIUM=NMED
      AGED=  AGE + DS/BL(NMTG+IL)
      N=1
      ARG=0.
      CALL EUCLID(N,X,Y,Z,XE,YE,ZE,DS,IL,ARG,0,MEDIUM,IBLZN,NREG)
      CO =  WATE* EXP(ARG)*  BL(IAL)/SD2/FNEST
1000  FORMAT(1H ,2I7,6E15.4)
      CALL FLUXST(I,IL,CON,AGED,COS,0)
15    CONTINUE
      INN=LOCXD+6*ND+I
      NL(IIN)=NL(INN)+1

      RETURN
      END
```

```
      SUBROUTINE PTHETA(IMED,IGOLD,IGQ,THETA,PMU,NMTG,NCO)
C
C     SAMPLE CALL SEQUENCE BY USER
C               COMMON WTS(1)
C                N1    (4+NRESP)*NMTG+1  OR  N1    (5+NRESP)*NMT3+1
C               CALL PTHETA(......,WTS(N1),NMTG)
C       DETERMINE PROBABILITY PER STERADIAN OF SCATTERING THROUGH AN ANGLE
C           THETA WHEN TRANSFERING FROM GROUP IG TO IGQ
C       INCASE OF UPSCATTERING IGQ SHOULD BE .GT. NTG
C       NCO ACTUAL NUMBER OF COEFICIENTS TO BE USED
C       IF .LT. 0 NCO=NCOEF
        COMMON/LOCSIG/ISTART,ISCCOG,INABOG,IGABOG,IFPORG,IFNGPG,IFSPOG,
     1          IDSGOG,IPRBNG,IPRBGG,ISCANG,ISCAGG,ISPORG,ISPORT,INPBUF,
     2          ISIGOG,IMFPOG,IABSOG,ITOTSG,NGP,NDS,NGG,NDSG,INGP,INDS,
     3          NMED,NELEM,NMIX,NCOEF,NSCT,NTS,NTG,NDSNGP,NDSNGG,IADJ,
     4          NME,LOC,INGS,INSG,I1,I0, KKK,IXTAPE,IDEL,ITEML,ITEMG,IRSG,
     5          IRDSG,ISTR,IPRIN,IFMU,IMOM,IDTF,ISTAT,IPUN
     6     ,IXTAP,MCR,MTP,NUS,NUSG,NTAB,NTABC ,INUS,IGSNU,INSGU
        COMMON/XGEN/ MXSTR,MSSTR,NGMED
        COMMON SIGT(2)
        DIMENSION P(10), NSIG(1)
        EQUIVALENCE (SIGT(1),NSIG(1))
        DIMENSION PMU(NMTG)
        IF(NCO.LT.0) NCO=NCOEF
        IF(ISTAT ) 1002,1002,1001
 1002   IF(NCO  -1) 1001,1001,1003
 1003   WRITE(IO,1005) ISTAT,NCOEF,NCO
 1005   FORMAT(1H1, 25HERROR IN PTHETA - ISTAT  , I2, 9H, NCOEF   ,I2)
        CALL HELP(4HPTHT,0,0,0,0)
 1001   IF(IGQ.LE.0)GO TO 100
        GO TO 101
  100   IF(IGOLD.GT.NGP)GO TO 120
  121   IGMD=NGP
        GO TO 110
  120   IGMD=NTG
        GO TO 110
  101   IGMD=IGQ
  110   CONTINUE
C       IF IGQ 0 CONSIDER ALL POSSIBLE DOWNSCATTER GROUPS, OTHERWISE CONSI
C       DER ONLY TO GROUP IGQ
C       THETA IS ANGLE OF SCATTER IN LAB SYSTEM.
C       IGOLD IS INCOMING GROUP
        NED=NSIG(MXSTR+IMED)
        IF(IGOLD.LE.NGP) GO TO 123
  124   IGS=NSIG(MSSTR+NED)+IDSGOG+NSIG(INSG+IGOLD-NGP)
        GO TO 125
  123   IGS=NSIG(MSSTR+NED)+IFSPOG+NSIG(INGS+IGOLD)
  125   CONTINUE
        IF(NCO    .LE.1)GO TO 115
        GO TO 117
  116   DO 118 I=IGOLD,IGMD
  118   PMU(I)= .079577*SIGT(IGS+I-IGOLD+1)
C       .079577 IS 1/4PI
        RETURN
  117   A=THETA
```

```
SUBROUTINE  PTHETA

      P(1)=A
      P(2)= 1.5*A*A-.5
      NMOM=NCO-1
      DO 105 K=3,NMOM
      B=K
  105 P(K)=((2.*(B-1.)+1.)*A*P(K-1)-(B-1.)*P(K-2))/B
C ***
C     WITHIN GROUP SCATTERING IS ASSUMED TO ALWAYS BE NON-ZERO
C ***
      PM=0
      NKND=ISPORT+(MED-1)*NTG*NTS*NCOEF+(IGOLD-1)*NTS
      DO 115 K=1,NMOM
      N1=NKND+NTG*NTS*(K-1)+1
  115 PM=PM + P(K)*SIGT(N1)
      PMU(IGOLD)=  (PM+1.)*.079577*SIGT(IGE+1)
      IGST=IGOLD+1
      DO 106 I=IGST,IGOD
      PM=0
      J=I-IGOLD  +1
      N3=IGE+ J
      DISIG=SIGT(N3)
      IF(DISIG.LE.0.)GO TO 108
  109 DO 107 K=1,NMOM
      N1=NKND+NTG*NTS*(K-1)+J
  107 PM=PM +P(K)*SIGT(N1)
      PMU(I)=  (PM+1.)*.079577*DISIG
  106 CONTINUE
      RETURN
  108 PMU(I)=0.
      RETURN
      END
```

```
SUBROUTINE  STRUN

      SUBROUTINE STRUN
      RETURN
      END
```

Cards I 11 and I 12 read as follows for the individual weighting types:

Weighting a

NSPLT= 1 NKILL= 1 NPAST= 0 NOLEAK= 0

IEBIAS= 0 MXREG= 2 MAXGP= 6

NGP1	NDG	NGP2	NRG1	NDRG	NRG2	WTAVE1	XNU
1	1	1	1	1	2	9.0000E-01	-0.
2	1	2	1	1	2	2.5000E+00	-0.
3	1	6	1	1	2	6.0000E+00	-0.

Weighting b

NSPLT= 1 NKILL= 1 NPAST= 0 NOLEAK= 0

IEBIAS= 0 MXREG= 2 MAXGP= 6

NGP1	NDG	NGP2	NRG1	NDRG	NRG2	WTAVE1	XNU
1	1	1	1	1	1	8.0000E-01	-0.
1	1	6	2	1	2	2.0000E-01	-0.
2	1	2	1	1	1	1.0000E+00	-0.
3	1	6	1	1	1	3.0000E+00	-0.

Weighting c

NSPLT= 1 NKILL= 1 NPAST= 0 NOLEAK= 0

IEBIAS= 0 MXREG= 2 MAXGP= 6

NGP1	NDG	NGP2	NRG1	NDRG	NRG2	WTAVE1	XNU
1	1	1	1	1	2	5.0000E-01	-0.
2	1	2	1	1	2	6.0000E+00	-0.
3	1	3	1	1	2	5.0000E+01	-0.
4	1	6	1	1	2	1.5000E+02	-0.

Weighting d

NSPLT= 1 NKILL= 1 NPAST= 1 NOLEAK= 0

IEBIAS= 0 MXREG= 2 MAXGP= 6

NGP1	NDG	NGP2	NRG1	NDRG	NRG2	WTAVE1	XNU
1	1	1	1	1	2	5.0000E-01	.8200E+00
2	1	2	1	1	2	6.0000E+00	.8200E+00
3	1	3	1	1	2	5.0000E+01	.8200E+00
4	1	6	1	1	2	1.5000E+02	.8200E+00

Weighting e

NSPLT= 1 NKILL= 1 NPAST= 1 NOLEAK= 1

IEBIAS= 0 MXREG= 2 MAXGP= 6

NGP1	NDG	NGP2	NRG1	NDRG	NRG2	WTAVE1	XNU
1	1	1	1	1	2	5.0000E-01	.8200E+00
2	1	2	1	1	2	6.0000E+00	.8200E+00
3	1	3	1	1	2	5.0000E+01	.8200E+00
4	1	6	1	1	2	1.5000E+02	.8200E+00

3. Input

```
      LIH 6GP P3 ZYLINDER 3 PUNKTDETEKTOREN WICHTUNG E
 6000
  200   210    10     1     6     0     6     6     0     0   153     1   777
    0     5     1     1          1.0          0.01          0.0           0.0          1.0
          0.0           0.0           0.0           0.0           0.0           0.0          1.0
0.5           0.25          0.15          0.05          0.05
      0.909         0.076         0.009         0.003         0.003
        6.0           5.0           4.0           3.0           2.0           1.0

LIH 6GP P3
     6     6     0     0     6     6     1     1     1     4     3     1
     4     0     0     0     0     9     9     1
     0     0     0     0     0     0     0
     2
     1     1
     4
 2R        0.0    1603847-7    8047404-8 5R        0.0
 2R        0.0    2124522-7    1034824-7    5797974-8 4R        0.0
 2R        0.0    2742521-7    1210173-7    8469655-9    1497506-8 3R        0.0
 2R        0.0    7467581-7    5600054-7    1505084-7    2430085-8    6776506-9
 2R        0.0
 2R        0.0    1126862-6    9059776-7    1844973-7    1590173-9    300473-09
   855634-10           0.0
 2R        0.0    1857228-6    1141554-6    1859721-7    828298-09    725415-11
   137074-11    390337-12
 3R        0.0    1383202-7 5R        0.0
 3R        0.0    1552742-7    5818425-8 4R        0.0
 3R        0.0    2016727-7    8597831-8    2081349-8 3R        0.0
 3R        0.0    1110005-6    1997495-7    2343770-8    4589186-9 2R        0.0
 3R        0.0    1845514-6    2439137-7    300655-09    328913-10    644030-11
           0.0
 3R        0.0    2424043-6    2390036-7    757274-10    926380-13    101346-13
  198439-14
 3R        0.0    1690739-7 5R        0.0
 3R        0.0    1782997-7    4618013-8 4R        0.0
 3R        0.0    2339293-7    7010063-8    -109737-7 3R        0.0
 3R        0.0    9601901-7    2840997-8    -378202-7    -140346-7 2R        0.0
 3R        0.0    1419741-6    -113565-6    -392136-8    -747787-9    -213451-9
      0.0
 3R        0.0    1420056-6    -123415-6    -206367-8    -18134-10    -34268-11
  -97582-12
 3R        0.0    1593821-7 5R        0.0
 3R        0.0    1570503-7    -115052-9 4R        0.0
 3R        0.0    1981277-7    -508354-7    -428195-7 3R        0.0
 3R        0.0    3677784-7    -189899-6    -618904-7    -142188-7 2R        0.0
 3R        0.0    3551886-7    -362318-6    -104242-8    -114766-9    -22509-09
      0.0
 3R        0.0    3667012-7    -352869-6    -264348-9    -32424-11    -35470-12
  -69454-13
    1    -1          1.0
    1     1     1     1     0     2     6
    1     1     1     1     1     2                                    0.5          0.82
    2     1     2     1     1     2                                    6.0          0.82
    3     1     3     1     1     2                                   50.0          0.82
    4     1     6     1     1     2                                  150.0          0.82
   -1
```

```
    0
    1
    2      MALE
ZONE X              -20.0,       +20.0
ZONE Y              -20.0,       +20.0
ZONE Z              -10.0,       150.0
ZONE        1     1     1
BLOCK X             -20.0,      +20.0
BLOCK Y             -20.0,      +20.0
BLOCK Z             -10.0,        -0.0,       20.0,       45.0,      150.0
BLOCK       1     1     1
MEDIA           1000
REGIONS             1
BLOCK       1     1     2
MEDIA               0,     1
SURFACES            1
SECTOR +1
SECTOR -1
REGIONS             1
BLOCK       1     1     3
MEDIA               0,     2
SURFACES            1
SECTOR +1
SECTOR -1
REGIONS             2
BLOCK       1     1     4
MEDIA           1000
REGIONS             1
     1
         +1.XSQ              +1.YSQ        -380.25      $
LIH ZYLINDER                 3 PUNKTDETEKTOREN
     3     6     6     0     0     1     1     1     0
          0.0         0.0         100.          1.0
          0.0        10.0         100.          1.0
          0.0         0.0         150.0         1.0
FLUX DETEKTOR
 SOURCE BIASING
          1.0         1.0          1.0          1.0          1.0          1.0

     1     2     3     4     5     6
```

4. Results

After about 3 minutes of calculation time, the following total fluxes and spectra resulted at the detectors:

Weighting a

ENERGIES	DETECTOR NO. 1	2	3
6.000E+00	7.162E-08	6.624E-08	2.931E-08
	.222	.206	.192
5.000E+00	3.869E-08	3.895E-08	1.463E-08
	.280	.286	.254
4.000E+00	9.074E-09	8.751E-09	3.304E-09
	.378	.382	.355
3.000E+00	3.319E-11	2.750E-11	1.190E-11
	.850	.903	.844
2.000E+00	3.596E-14	2.486E-14	1.452E-14
	.982	.987	.981
1.000E+00	4.667E-20	3.703E-20	2.089E-20
	.915	.925	.935
1.000E-02			

Weighting b

ENERGIES	DETECTOR NO. 1	2	3
6.000E+00	1.760E-07	1.652E-07	6.530E-08
	.323	.320	.303
5.000E+00	8.267E-08	7.951E-08	2.907E-08
	.338	.352	.319
4.000E+00	5.015E-08	4.544E-08	1.696E-08
	.518	.528	.486
3.000E+00	2.741E-08	2.454E-08	8.594E-09
	.884	.860	.878
2.000E+00	4.373E-09	4.265E-09	1.301E-09
	.837	.813	.823
1.000E+00	3.961E-10	2.903E-10	1.296E-10
	.822	.730	.811
1.000E-02			

weighting c

DETECTOR NO. ENERGIES	1	2	3
6.000E+00			
	6.769E-08	7.029E-08	2.900E-08
	.096	.106	.090
5.000E+00			
	6.644E-08	6.396E-08	2.361E-08
	.358	.345	.337
4.000E+00			
	4.961E-08	5.022E-08	1.616E-08
	.375	.371	.359
3.000E+00			
	2.693E-08	2.627E-08	7.859E-09
	.461	.453	.452
2.000E+00			
	2.236E-08	2.223E-08	6.790E-09
	.431	.439	.428
1.000E+00			
	1.806E-08	9.582E-09	2.897E-09
	.468	.463	.462
1.000E-02			

Weighting d

DETECTOR NO. ENERGIES	1	2	3
6.000E+00			
	1.209E-07	1.162E-07	4.517E-08
	.156	.154	.142
5.000E+00			
	8.682E-08	8.477E-08	2.942E-08
	.235	.237	.223
4.000E+00			
	5.350E-08	5.405E-08	1.748E-08
	.350	.353	.349
3.000E+00			
	6.360E-08	6.009E-08	1.923E-08
	.465	.459	.465
2.000E+00			
	5.689E-08	5.557E-08	1.701E-08
	.464	.462	.460
1.000E+00			
	1.927E-08	1.884E-08	5.551E-09
	.411	.403	.399
1.000E-02			

Weighting / e	DETECTOR ENERGIES V_0	1	2	3
	5.000E+00			
		1.11[illegible]E-07	1.045E-07	4.148E-08
		.159	.156	.159
	5.000E+00			
		9.936E-08	9.291E-08	3.252E-08
		.217	.222	.210
	4.000E+00			
		9.55[illegible]E-08	8.939E-08	2.993E-08
		.171	.167	.169
	3.000E+00			
		6.294E-08	6.003E-08	1.858E-08
		.322	.322	.318
	2.000E+00			
		5.922E-08	5.637E-08	1.747E-08
		.284	.283	.296
	1.000E+00			
		1.033E-08	1.858E-08	5.409E-09
		.353	.369	.363
	1.000E-02			

In order to judge the results, we can compare the shape of the spectra with calculations made for the same arrangement with the program DOT [11] and with the area detector calculation described in the next chapter. Figure 2 shows the results for various z values. They show the same shape as those found with weighting e. The other weightings resulted in occasionally severe under-estimations; not enough particles reached areas important for the result.

VI. AREA DETECTOR

1. Description

This detector allows the estimation of the average flux that corresponds to particles crossing an area. This value is also obtained from one- and two-dimensional SN programs, and hence this detector is especially suitable for comparison with calculations made with different methods.

In the estimation, the particle weights are divided by the cosine of the angle between the particle direction and the normal to the surface and summed. The result is divided by the detector area; this area must be entered as VOL on card S 3.

Difficulties with this detector appear only if the values of cos μ become very small, i.e., when the particle enters almost parallel to the detector surface. Cain has shown [2] how these difficulties can be cleverly avoided. According to this method, we obtain finite variances if, for all $\mu < \varepsilon$, we replace $1/\mu$ by the expectation value estimation $\langle 1/\mu \rangle$. As a rule, the error in this procedure is negligible.

If, in the range $0 < \mu < \xi$, a constant angular flux distribution is assumed, then the μ are distributed according to

$$f(\mu) = 2\mu/\xi^2$$

and we have

$$\langle 1/\mu \rangle = \frac{2}{\xi^2} \int f(\mu) \frac{1}{\mu} d\mu = \frac{2}{\xi}$$

2. Test Example

The irradiation of a head through a collimator was used as a test example. The collimator was simulated by a monoenergetic source with a very small angular opening. The head contains a brain tumor, A picture of the head was produced with the aid of the auxiliary program PICTURE. It is shown in Chap. IX.

The fluxes were estimated in circular rings around the central plane of the tumor. Contributions were recorded from source neutrons and other neutrons which cross this surface. For the sake of simplicity, only six energy groups are considered. They correspond to energies between 15 and 4.06 MeV.

The radii of the circular rings are read on card S 3 with the designation X. On the same card, with the name VOL, the relative surface areas of the rings are entered (in units of π).

```
      SUBROUTINE SOURCE(IG,U,V,W,X,Y,Z,WATE,M,B,IS,IT,NGP,D,ISB,NMTG)
      COMMON WTS(1)
      X=-60.0
      Y=1.5
      Z=20.0
      XU=FLTRNF(R)
      U=1.0-0.0005555*XU
      U1=SQRT(1.0-U*U)
      CALL AZIRN(V1,W1)
      V=V1*U1
      W=W1*U1
      WATE=1.0
      IF(IS) 90,90,95
   95 IG=IS
      RETURN
   90 CONTINUE
      IF(ISB) 100,100,150
  100 NWT=2*NMTG
      GO TO 200
  150 NWT=3*NMTG
  200 R=FLTRNF(R)
      DO 250 I=1,NGP
      IF(R.LE.WTS(I+NWT)) GO TO 300
  250 CONTINUE
  300 IG=I
      IF(ISB) 500,500,350
  350 IF(I-1) 500,400,450
  400 WATE=WATE*WTS(2*NMTG+1)/WTS(3*NMTG+1)
      RETURN
  450 WATE=WATE*(WTS(2*NMTG+I)-WTS(2*NMTG+I-1))/
     1           (WTS(3*NMTG+I)-WTS(3*NMTG+I-1))
  500 RETURN
      END
```

```
      SUBROUTINE BANKR(NBNKID)
C  DO NOT CALL EUCLID FROM BANKR(7)
      COMMON /APOLLO/ AGSTRT,DDF,DEADWT(5),ETA,ETATH,ETAUSD,UINP,VINP,
     1 WINP,WTSTRT,XSTRT,YSTRT,ZSTRT,TCUT,XTRA(10),
     2                                    IO,II,MEDIA,IADJM,ISBIAS,
     3 ISOUR,ITERS,ITIME,ITSTR,LOCWTS,LOCFWL,LOCEPR,LOCNSC,LOCFSN,
     4 MAXGP,MAXTIM,MEDALB,MGPREG,MXREG,NALB,NDEAD(5),NEWNM,NGEOM,
     5 NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NKCALC,NKILL,NLAST,NMEM,
     6 NMGP,NMOST,NMTG,NOLEAK,NORMF,NPAST,NPSCL(13),NQUIT,NSIGL,NSOUR,
     7 NSPLT,NSTRT,NXTRA(10)
      NBNK=  NBNKID
      IF (NBNK) 100,100,140
  100 NBNK=  NBNK + 5
      GO TO (104,103,102,101),NBNK
  101 CALL STRUN
      RETURN
  102 NBAT=  NITS - ITERS
      NSAVE=  NMEM
      CALL STBTCH(NBAT)
C   NBAT IS THE BATCH NO. LESS ONE
      RETURN
  103 CALL NBATCH(NSAVE)
C   NSAVE IS THE NO. OF PARTICLES STARTED IN THE LAST BATCH
      RETURN
  104 CALL HELP(4HBANK,1,1,1,1)
      RETURN
  140 GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13) NBNK
C  NBNKID    COLL TYPE    BANKR CALL        NBNKID    COLL TYPE    BANKR CALL
C     1        SOURCE      YES (MSOUR)        2        SPLIT        NO  (TESTW)
C     3        FISSION     YES (FBANK)        4        GAMGEN       NO  (GSTORE
C     5        REAL COLL    YES (MORSE)       6        ALBEDO       YES (MORSE)
C     7        BDRYX        YES (NXTCOL)      8        ESCAPE       YES (NXTCOL
C     9        E-CUT        NO  (MORSE)      10        TIME KILL    NO  (MORSE)
C    11        R R KILL     NO  (TESTW)      12        R R SURV     NO  (TESTW)
C    13        GAMLOST      NO  (GSTORE)
    1 CALL SDATA
      RETURN
    2 RETURN
    3 RETURN
    4 RETURN
    5 RETURN
    6 RETURN
    7 CALL BDRYX
      RETURN
    8 RETURN
    9 RETURN
   10 RETURN
   11 RETURN
   12 RETURN
   13 RETURN
      END
```

```
      SUBROUTINE BDRYX
C     BDRYX FOR SLAB GEOM
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EROTN,EBOTG,
     1 TCUT,I0,I1,IADJM,NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
      COMMON /PDET/ ND,NNE,NE,NT,NA,NRESP,NFX,NEXND,NEND,NDNR,NTNR,NTNE
     1 ,NANE,NTNDNR,NTNEND,NANEND,LOCRSP,LOCXD,LOCIR,LOCCO,LOCT,LOCUD,
     2  LOCSD,LOCNE,LOCQT,LOCQTE,LOCQAE,LMAX,EFIRST,EGTOP
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,BLANT,BLZON,AGE,OLDAGE
      COMMON BL(1)
      IF(X.GT.+0.001) RETURN
      IF(X.LT.-0.001) RETURN
      YR=Y -1.5
      ZR=Z -20.
      RSQ=YR*YR+ZR*ZR
      IA=LOCXD
      DO 5 I=1,ND
      IA=IA+1
      IF(RSQ.LE.BL(IA))GO TO 20
    5 CONTINUE
      RETURN
   20 COS=U
      ABC = ABS (COS)
      IF (COS) 25,30,25
   25 IF (ABC-1.0001) 35,30,30
   30 CALL HELP(4HBDYX,0,1,0,0)
   35 IF (ABC-0.01) 40,45,45
   40 CON = WATE*200.0
      GO TO 50
   45 CON = WATE/ABC
      CALL FLUXST(I,IG,CON,AGE,U, 0)
   50 RETURN
      END
```

```
      SUBROUTINE SDATA
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EROTN,EROTG,
     1 TCUT,T0,I1,IADJM,NGPOT1,NGPOT2,NGPOT3,NGPOTG,NGPOTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
      COMMON /PDET/ ND,NNE,NE,NT,NA,NRESP,NEX,NEXND,NEND,NDNR,NTNR,NTNE
     1 ,NANE,NTNDNR,NTNEND,NANEND,LOCRSP,LOCXD,LOCIR,LOCCO,LOCT,LOCUD,
     2  LOCSD,LOCQE,LOCQT,LOCQTE,LOCQAE,LMAX,FFIRST,EGTOP
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,NTBC,IBLZT,BLZON,AGE,OLDAGE
      COMMON EN(1)
      COMMON/GEOMC/ DUMMY(10),MARK,NMNDUM(2)
      DATA RAD/512./
      IA=LOCXD
      I=0
      X1=X
      Y1=Y
      Z1=Z
      WTSTO=WATE
      MED=NMED
      IBLZ=IBLZT
      NRG=NREG
   40 I=I+1
      IA=IA+1
      ZI=EN(IA)
      ARGU=0.
   20 CONTINUE
      X2=X1+512.*U
      Y2=Y1+512.*V
      Z2=Z1+512.*W
       MRK=1
      CALL EUCLID(MRK,X1,Y1,Z1,X2,Y2,Z2,RAD,IG,ARG,1,MED,IBLZ,NRG)
      IF(MARK.EQ.-1) GO TO 30
      ARGU=ARGU+ARG
      IF(X1.LT.-0.001) GO TO 20
      IF(X1.LT.+0.001) GO TO 21
      RETURN
   21 YR=Y1-1.5
      ZR=Z1-20.
      RSQ=YR*YR+ZR*ZR
   11 CONTINUE
      IF(RSQ.LE.EN(IA)) GO TO 10
      IA=IA+1
      I=I+1
      GO TO 11
   10 WATE=WATE*EXP(ARGU)
      CON=WATE/ABS(U)
      CALL FLUXST(I,IG,CON,AGE,U,-1)
   30 WATE=WTSTO
      RETURN
      END
```

3. Input

```
Head irradiation
10000
  300   350    30     1     6     0     6     6     0     0   100     3   777
    1     1     0     1        1.0        1.0        0.0        0.0        1.0
          0.0          0.0          0.0          0.0          0.0          0.0          1.0
           6.           5.           4.           3.           2.           1.

  6 GROUP head data
    6     6     0     0     6     6     3     3     3     4     2     0
   12     0     0     0     0    10    10     1
    0     0     0     0     0     0     1
   10
    1     1     3     2     3     1     1     3     2     3
    4     4     4
 skin / P0
   9.6777E-03  0.          9.1277E-02  2.8279E-02  0.          0.
   0.          0.          0.          0.          7.3048E-03  0.
   1.0301E-01  2.6888E-02  9.8961E-03  0.          0.          0.
   0.          0.          4.0106E-03  0.          1.0526E-01  2.0973E-0
   1.6436E-02  5.8344E-03  0.          0.          0.          0.
   2.7365E-03  0.          1.1721E-01  2.6438E-02  2.3683E-02  9.4157E-0
   7.5670E-03  0.          0.          0.          1.3626E-03  0.
   1.3537E-01  3.3814E-02  2.8632E-02  9.8851E-03  7.0559E-03  6.5395E-0
   0.          0.          2.1211E-03  0.          1.6331E-01  3.9901E-0
   3.0901E-02  9.7553E-03  6.2456E-03  7.6221E-03  4.4846E-03  0.
skin / P1
   0.          0.          0.          2.2621E-02  0.          0.
   0.          0.          0.          0.          0.          0.
   0.          2.1292E-02  4.4354E-03  0.          0.          0.
   0.          0.          0.          0.          0.          1.6794E-0
   5.8357E-03  4.1114E-03  0.          0.          0.          0.
   0.          0.          0.          1.9342E-02  8.5225E-03  6.2304E-0
   3.7682E-03  0.          0.          0.          0.          0.
   0.          2.4596E-02  8.4077E-03  7.2410E-03  4.5269E-03  2.5368E-0
   0.          0.          0.          0.          0.          3.0770E-0
   8.4850E-03  7.2448E-03  4.3076E-03  2.6108E-03  1.4630E-03  0.
skin / P2
   0.          0.          0.          1.6664E-02  0.          0.
   0.          0.          0.          0.          0.          0.
   0.          1.5476E-02  4.0495E-03  0.          0.          0.
   0.          0.          0.          0.          0.          1.2663E-0
   5.5134E-03  2.4757E-03  0.          0.          0.          0.
   0.          0.          0.          1.4132E-02  8.5843E-03  4.3250E-0
   1.4239E-03  0.          0.          0.          0.          0.
   0.          1.7525E-02  1.0311E-02  4.3509E-03  1.6754E-03  4.0081E-0
   0.          0.          0.          0.          0.          2.0747E-0
   1.0364E-02  4.4179E-03  1.5415E-03  4.4679E-04 -5.6463E-05  0.
skin / P3
   0.          0.          0.          1.2771E-02  0.          0.
   0.          0.          0.          0.          0.          0.
   0.          1.1883E-02  2.9399E-03  0.          0.          0.
   0.          0.          0.          0.          0.          1.0091E-0
   4.5214E-03  4.1290E-03  0.          0.          0.          0.
   0.          0.          0.          1.1555E-02  6.3959E-03  7.8350E-0
  -8.0001E-04  0.          0.          0.          0.          0.
   0.          1.3719E-02  7.0030E-03  5.1628E-04 -1.0070E-03 -1.2632E-0
   0.          0.          0.          0.          0.          1.4203E-0
   7.8296E-03  6.9970E-04 -1.0381E-03 -1.2598E-03 -1.0272E-03  0.
     1    -1        1.0
```

Six additional blocks follow (bone and brain) as well as the geometric input.

Next card: Card S 1

```
HEAD UNDER COLLIMATOR
  10     6     6     2     0     2     0     0     0
      0.25                                 0.25
      1.00                                 0.75
      2.25                                 1.25
      4.00                                 1.75
      6.25                                 2.25
      9.00                                 2.75
     12.25                                 3.25
     16.00                                 3.75
     20.25                                 4.25
     25.00                                 4.75
NEUTRONS PER PI
FLUX
        1.0         1.0         1.0         1.0         1.0         1.0
RATES FOR BRAIN ACCORDING TO HEHN
 6.37E-09  5.78E-09  5.16E-09  4.77E-09  4.23E-09  4.13E-09
FLUX SPECTRA
    1     2     3     4     5     6
```

4. Results

Fourteen batches with 300 source neutrons each were simulated in 1.38 minutes. The following tables were printed out:

RESPONSES(DETECTOR) NEUTRONS PER PI

DETECTOR	UNCOLL RESPONSE	FSD UNCOLL	TOTAL RESPONSE	FSD TOTAL
1	1.2364E-01	.03912	1.3478E-01	.07907
2	1.1735E-01	.03419	1.3764E-01	.04065
3	1.1446E-01	.02109	1.4721E-01	.03179
4	1.1685E-01	.01669	1.3502E-01	.03209
5	1.4218E-04	.53109	1.7169E-02	.09711
6	0.	0.00000	1.0212E-02	.12095
7	0.	0.00000	8.2929E-03	.08720
8	0.	0.00000	6.6683E-03	.06381
9	0.	0.00000	5.6583E-03	.17058
10	0.	0.00000	3.8958E-03	.20848

RATES FOR BRAIN

RESPONSES(DETECTOR) NEUTRONS PER PI

DETECTOR	UNCOLL RESPONSE	FSD UNCOLL	TOTAL RESPONSE	FSD TOTAL
1	7.8762E-10	.03912	8.3761E-10	.07823
2	7.4750E-10	.03419	8.6503E-10	.04036
3	7.2908E-10	.02109	9.2357E-10	.03287
4	7.4436E-10	.01669	8.4317E-10	.03181
5	9.0567E-13	.53109	9.8236E-11	.09956
6	0.	0.00000	5.7660E-11	.11994
7	0.	0.00000	4.6581E-11	.08444
8	0.	0.00000	3.6595E-11	.07235
9	0.	0.00000	3.1089E-11	.18945
10	0.	0.00000	2.1737E-11	.22216

FLUENCE(ENERGY,DETECTOR) FLUX SPECTRA

DETECTOR NO. ENERGIES	1	2	3	4	5
6.000E+00	1.208E-01	1.281E-01	1.367E-01	1.230E-01	9.236E-03
	.085	.043	.038	.033	.144
5.000E+00	5.369E-03	4.275E-03	3.661E-03	3.757E-03	2.205E-03
	.292	.156	.218	.162	.170
4.000E+00	8.007E-04	1.346E-03	1.456E-03	2.410E-03	1.701E-03
	1.000	.338	.217	.185	.225
3.000E+00	5.803E-04	1.794E-03	2.761E-03	1.893E-03	1.657E-03
	1.000	.351	.245	.229	.216
2.000E+00	3.541E-03	1.299E-03	1.744E-03	2.446E-03	1.870E-03
	.537	.439	.353	.277	.257
1.000E+00	3.709E-03	8.533E-04	9.330E-04	1.555E-03	4.997E-04
	.401	.593	.398	.325	.439

6	7	8	9	10
4.693E-03	3.837E-03	2.456E-03	2.241E-03	1.814E-03
.132	.118	.169	.375	.355
1.859E-03	1.307E-03	1.102E-03	9.381E-04	5.102E-04
.248	.222	.154	.237	.257
1.087E-03	7.230E-04	1.206E-03	4.824E-04	4.307E-04
.156	.307	.183	.337	.283
1.071E-03	1.184E-03	6.708E-04	9.091E-04	3.794E-04
.268	.369	.321	.196	.268
9.948E-04	7.672E-04	6.359E-04	7.359E-04	5.582E-04
.330	.412	.264	.288	.272
5.069E-04	4.745E-04	5.974E-04	3.517E-04	2.035E-04
.389	.289	.349	.495	.410

VII. VOLUME DETECTOR

1. Description

The estimators are identical with those already described in Chap. IV. However, here they are not only used for k_{eff} estimations but generally for the determination of volume-integrated fluxes and rates.

The average cross sections of any region can be determined from the quotient of the two values. Thus it is possible to condense and homogenize cross sections.

Flux and rate are not independent. In order to determine average cross sections, a maximum likelihood estimation (regression analysis) must be carried out. This gives

$$\tilde{\Sigma}_R = \frac{\sum_i \phi_i \Sigma_R \phi_i}{\sum_i \phi_i^2}$$

The density of $\tilde{\Sigma}_R$ is a Student's t distribution. For batch numbers greater than 10 it goes over into a normal distribution.

By this method, the flux estimator is used in the program to calculate values for $\tilde{\Sigma}_\tau, \nu\tilde{\Sigma}_f$ and $\tilde{\Sigma}_a$ ($\tilde{\Sigma}_s = \tilde{\Sigma}_\tau - \tilde{\Sigma}_a$). If (n,2n) reactions occur, $\tilde{\Sigma}_a$ can take on negative values.

2. Test Example

By way of illustration, cross sections for a fuel-rod transport container will be reduced from 16 groups to 6 groups. Figure 2 shows a diagram of such a container (taken from Thomas [14]). (NOTE: There is no Fig. 2 in the German report. Undoubtedly Fig. 3 is meant here.]

The following values were used in the calculation:

S_R = 20 cm water
H = 200 cm
$C_x = C_y$ = 50 cm

D = 12.7 for fuel rod
D = 13.7 for fuel rod cover

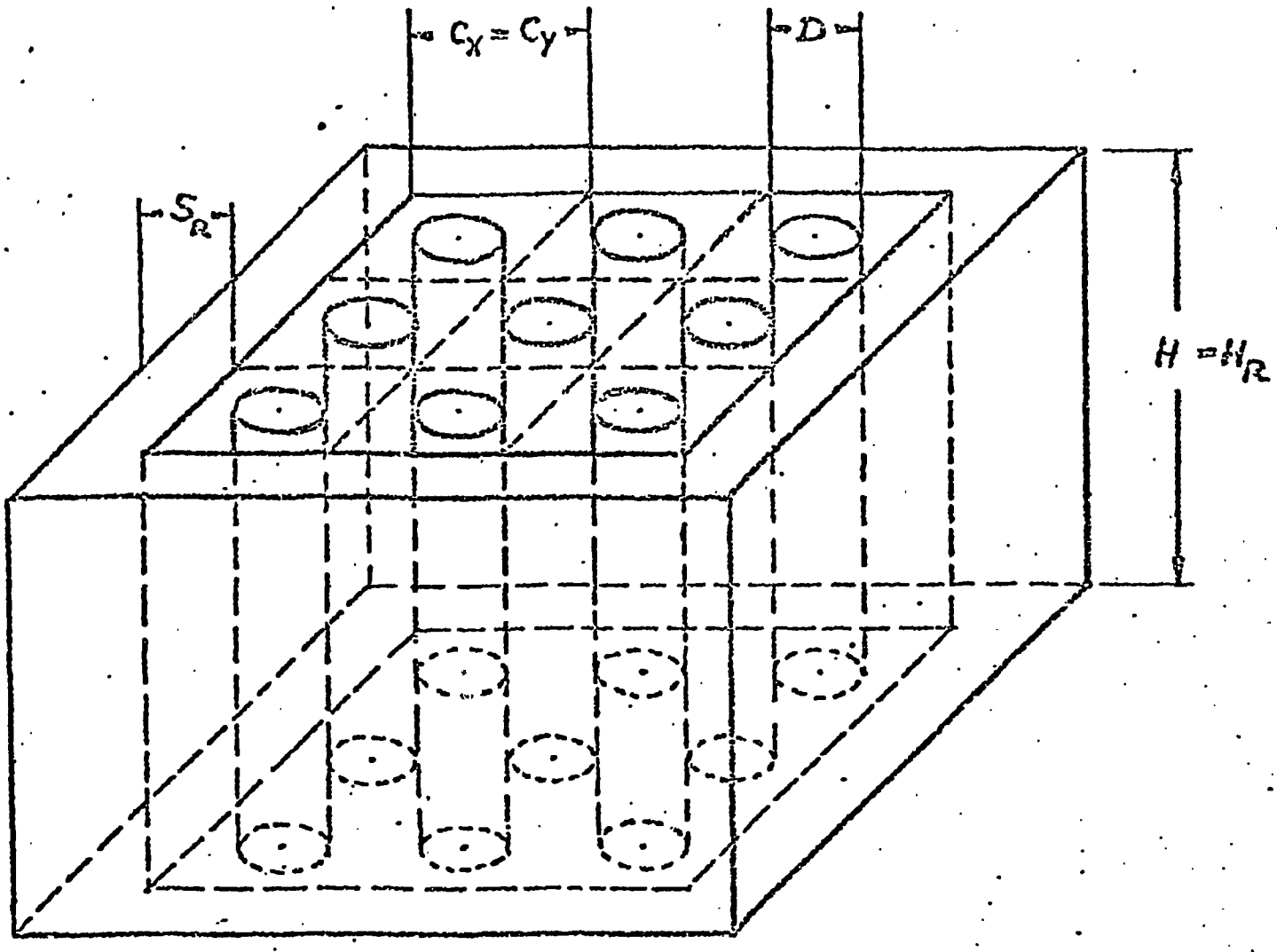


Fig. 3. Diagram of transport container.

The container contained 100 fuel rods, the outermost of which were located directly against the reflector.

```
      SUBROUTINE BANKR(NBNKID)
C  DO NOT CALL EUCLID FROM BANKR(7)
      COMMON /APOLLO/ AGSTRT,DDF,DEADWT(6),ETA,ETATH,ETAUSD,UINP,VINP,
     1 WINP,WTSTRT,XSTRT,YSTRT,ZSTRT,TCUT,XTRA(10),
     2                                    IO,I1,MEDIA,IADJM,ISBIAS,
     3 ISOUR,ITERS,ITIME,ITSTR,LOCWTS,LOCFWL,LOCEPR,LOCNSC,LOCFSN,
     4 MAXGP,MAXTIM,MEDALB,MGPREG,MXREG,NALB,NDEAD(5),NEWNM,NGEOM,
     5 NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NKCALC,NKILL,NLAST,NMEM,
     6 NMGP,NMOST,NMTG,NOLEAK,NORMF,NPAST,NPSCL(13),NQUIT,NSIGL,NSOUR,
     7 NSPLT,NSTRT,NXTRA(10)
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKICSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,NTERM ,INITS
      COMMON/FISNEW/ FTOTL,FWATE,MFISTP,NFISH,NFPT,WATEF
      NBNK=  NBNKID
      IF (NBNK) 100,100,140
  100 NBNK=  NBNK + 5
      GO TO (104,103,102,101),NBNK
  101 CALL STRUN
      RETURN
  102 NBAT=  NITS - ITERS
      NSAVE=  NMEM
      CALL STBTCH(NBAT)
C   NBAT IS THE BATCH NO. LESS ONE
      RETURN
  103 CALL NBATCH(NSAVE)
C   NSAVE IS THE NO. OF PARTICLES STARTED IN THE LAST BATCH
      RETURN
  104 CALL HELP(4HBANK,1,1,1,1)
      RETURN
  140 GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13) NBNK
C  NBNKID    COLL TYPE    BANKR CALL       NBNKID    COLL TYPE    BANKR CALL
C     1        SOURCE      YES (MSOUR)       2        SPLIT        NO  (TESTW)
C     3        FISSION     YES (FBANK)       4        GAMGEN       NO  (GSTORE
C     5        REAL COLL   YES (MORSE)       6        ALBEDO       YES (MORSE)
C     7        BDRYX       YES (NXTCOL)      8        ESCAPE       YES (NXTCOL
C     9        E-CUT       NO  (MORSE)      10        TIME KILL    NO  (MORSE)
C    11        R R KILL    NO  (TESTW)      12        R R SURV     NO  (TESTW)
C    13        GAMLOST     NO  (GSTORE)
    1 CALL SDATA
      RETURN
    2 RETURN
    3 RETURN
    4 RETURN
    5 RKHC=RKHC+WATEF
    6 CALL ALBCOL
C       TREATS ALBEDO AND COLLISION
      RETURN
    7 CALL BDRYX
      RETURN
    8 CALL ESCAPE
      RETURN
    9 RETURN
   10 RETURN
   11 CALL RRKILL
      RETURN
   12 RETURN
   13 RETURN
      END
```

```
      SUBROUTINE SDATA
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EBOTN,EBOTG,
     1 TCUT,IC,I1,IADJM,NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
       COMMON /PDET/ ND,NNE,NE,NT,NA,NRESP,NEX,NEXND,NEND,NONR,NTNR,NTNE
     1 ,NANE,NTNONR,NTNEND,NANEND,LOCRSP,LOCXD,LOCIB,LOCCO,LOCT,LOCUD,
     2  LOCSD,LOCQE,LOCQT,LOCQTE,LOCNAE,LMAX,EFIRST,EGTOP,NKK,LOCNKK
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,BLZNT,BLZON,AGE,OLDAGE
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKICSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,NTERM ,INITS
      DIMENSION CON(4)
      I=NMED
      COS=0.
      CON(1)=WATE
      SWATE=SWATE+WATE
      NTERM=NTERM+1
      CALL FLUXST(I,IG,CON,AGE,COS,-1)
      RETURN
      END


      SUBROUTINE BDRYX
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EBOTN,EBOTG,
     1 TCUT,IC,I1,IADJM,NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
       COMMON /PDET/ ND,NNE,NE,NT,NA,NRESP,NEX,NEXND,NEND,NONR,NTNR,NTNE
     1 ,NANE,NTNONR,NTNEND,NANEND,LOCRSP,LOCXD,LOCIB,LOCCO,LOCT,LOCUD,
     2  LOCSD,LOCQE,LOCQT,LOCQTE,LOCNAE,LMAX,EFIRST,EGTOP,NKK,LOCNKK
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,BLZNT,BLZON,AGE,OLDAGE
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKICSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,NTERM ,INITS
      COMMON WTS(1)
      COMMON/APOLLO/ XT(9),ETAUSD,XTT(31),LOCFSN,NX(22),NMEM
      DIMENSION CON(4)
      IF(MEDOLD-1000) 20,10,20
10    CON(1)=0.0
      CON(2)=0.0
      CON(3)=0.0
      CON(4)=0.0
      GO TO 30
20    I=MEDOLD
      PNUF=WTS(LOCFSN+(I-1)*NMTG+IGO)
      CALL NSIGTA(IGO,MEDOLD,TSIG,PNAB)
      PABS=1.-PNAB
      CON(1)=WTBC*ETAUSD
      CON(2)=CON(1)*PNUF
      CON(3)=CON(1)*PABS
      CON(4)=CON(1)
      CON(1)=CON(1)/TSIG
30    COS=0.0
      RKHF=RKHF+CON(2)
      CALL FLUXST(I,IGO,CON,AGE,COS,0)
      RETURN
      END
```

```
      SUBROUTINE ESCAPE
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EBOTN,EBOTG,
     1 TCUT,I0,I1,IADJN,NGPOT1,NGPOT2,NGPOT3,NGPOTG,NGPOTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
      COMMON /PUET/ NO,NNE,NE,NT,NA,NRESP,NEX,NEXND,NEND,NONR,NTNR,NTNE
     1 ,NAME,NTNONR,NTNEND,NANEND,LOCRSP,LOCXS,LOCIB,LOCCO,LOCT,LOCUD,
     2  LOCSD,LOCQE,LOCQT,LOCQTE,LOCQAE,LMAX,EFIRST,EGTOP,NKK,LOCNKK
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,BLZNT,BLZON,AGE,OLDAGE
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKICSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,NTERM ,INITS
      COMMON WTS(1)
      COMMON/APOLLO/ XT(9),ETAUSD,XTT(31),LOCFSN,NX(22),NMEM
      DIMENSION CON(4)
      IF(MEDOLD-1000) 40,50,40
   50 CON(1)=0.0
      CON(2)=0.0
      CON(3)=0.0
      CON(4)=0.0
      IF(NTERM-NMEM) 10,15,20
   40 I=MEDOLD
      PNUF=WTS(LOCFSN+(I-1)*NMTG+IGO)
      CALL NSIGTA(IGO,MEDOLD,TSIG,PNAB)
      PABS=1.-PNAB
      CON(1)=WTBC*ETAUSD
      CON(2)=CON(1)*PNUF
      CON(3)=CON(1)*PABS
      CON(4)=CON(1)
C*****FLUX ESTIMATOR    **************************
      CON(1)=CON(1)/TSIG
   30 COS=0.0
      RKHF=RKHF+CON(2)
      CALL FLUXST(I,IGO,CON,AGE,COS,0)
      IF(NTERM-NMEM) 10,15,20
   15 NTERM=NTERM-1
      CALL OP
      RKHC=0.
      RKHF=0.
   10 RETURN
   20 CALL HELP(4HNTRM,0,1,0,0)
      RETURN
      END


      SUBROUTINE OP
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKICSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,NTERM ,INITS
      RKFLX=RKFLX+RKHF
      VARCOL=VARCOL+RKHC*RKHC
      VARFLX=VARFLX+RKHF*RKHF
      RKOVAR=RKOVAR+RKHF*RKHC
      RETURN
      END
```

```
      SUBROUTINE ALBCOL
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EBOTN,EBOTG,
     1 TCUT,IC,I1,IADJM,NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
      COMMON /PDET/ ND,NNE,NE,NT,NA,NRESP,NEX,NEXND,NEND,NDNR,NTNR,NTNE,
     1 NANE,NTNDNR,NTNEND,NANEND,LOCRSP,LOCXD,LOCIR,LOCCO,LOCT,LOCUD,
     2  LOCSD,LOCQE,LOCQT,LOCQTE,LOCQAE,LMAX,EFIRST,EGTOP,NKK,LOCNKK
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,BLZNT,BLZON,AGE,OLDAGE
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKICSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,NTERM ,INITS
      COMMON WTS(1)
      COMMON/APOLLO/ XT(9),ETAUSD,XTT(31),LOCFSN,NX(22),NMEM
      DIMENSION CON(4)
      IF(MEDOLD-1000) 20,10,20
   10 CON(1)=0.0
      CON(2)=0.0
      CON(3)=0.0
      CON(4)=0.0
      GO TO 30
   20 I=MEDOLD
      PNUF=WTS(LOCFSN+(I-1)*NMTG+IGO)
      CALL NSIGTA(IGO,MEDOLD,TSIG,PNAB)
      PABS=1.-PNAB
      CON(1)=WTBC*ETAUSD
      CON(2)=CON(1)*PNUF
      CON(3)=CON(1)*PABS
      CON(4)=CON(1)
      CON(1)=CON(1)/TSIG
   30 COS=0.0
      RKHF=RKHF+CON(2)
      CALL FLUXST(I,IGO,CON,AGE,COS,0)
      RETURN
      END



      SUBROUTINE RRKILL
      COMMON /USER/ AGSTRT,WTSTRT,XSTRT,YSTRT,ZSTRT,DFF,EBOTN,EBOTG,
     1 TCUT,IC,I1,IADJM,NGPQT1,NGPQT2,NGPQT3,NGPQTG,NGPQTN,NITS,NLAST,
     2 NLEFT,NMGP,NMTG,NSTRT
      COMMON /PDET/ ND,NNE,NE,NT,NA,NRESP,NEX,NEXND,NEND,NDNR,NTNR,NTNE
     1 ,NANE,NTNDNR,NTNEND,NANEND,LOCRSP,LOCXD,LOCIR,LOCCO,LOCT,LOCUD,
     2  LOCSD,LOCQE,LOCQT,LOCQTE,LOCQAE,LMAX,EFIRST,EGTOP,NKK,LOCNKK
      COMMON /NUTRON/ NAME,NAMEX,IG,IGO,NMED,MEDOLD,NREG,U,V,W,UOLD,VOLD
     1 ,WOLD,X,Y,Z,XOLD,YOLD,ZOLD,WATE,OLDWT,WTBC,BLZNT,BLZON,AGE,OLDAGE
      COMMON/KEANL/RKFLX,VARFLX,VARCOL,RKOVAR,RKHF,RKHC,SWATE,
     1FKICSM,FKIFSM,FKISM,VAICSM,VAIFSM,VAISM,VAOCSM,VAOFSM,RKOVSM,
     2FKOCSM,FKOFSM,FNK,IKCALC,NTERM ,INITS
      COMMON/APOLLO/ XDUMM(64),NMEM
      DIMENSION CON(4)
      IF(NTERM-NMEM) 10,15,20
   15 NTERM=NTERM-1
      CALL OP
      RKHC=1.
      RKHF=0.
   10 RETURN
   20 CALL HELP(4HKILL,0,1,0,0)
      RETURN
      END
```

```
      SUBROUTINE SOURCE(IG,U,V,W,X,Y,Z,WATE,MED,AG,ISOUR,ITSTR,NGPQT3,DD
     1F,ISBIAS,NMTG)
C*****ISOTROPIC SOURCE
C******  GRID GEOM
C******SOURCE ENERGY BIASING ALLOWED
      DIMENSION R(36),XO(36),YO(36),F(36)
      COMMON WTS(1)
      COMMON/APOLLO/ XTRA1(28),IO,IN,XTRA2(70)
      DATA II /5/
      IF(II)88,88,99
   99  II=0
C******SOURCE INPUT
      READ(IN,1000) NR,ZU,ZO,IDR
      READ(IN,1001 )(XO(I),YO(I),R(I),F(I),I=1,NR)
 1000 FORMAT(I5,2E10.5,I5)
 1001 FORMAT(4E10.5)
      WRITE(IO,2000)NR,ZU,ZO
 2000 FORMAT(23H0 SOURCE FOR GRID  ,NR=,I5,4H ZU=,E15.5,4H ZO=,E15.5)
      ZO=ZO-ZU
      S=0.0
      WRITE(IO,1920)
      F( 1)=F( 1)*R( 1)*R( 1)
      DO 1 I1=2,NR
      F(I1)=F(I1)*R(I1)*R(I1) +F(I1-1)
    1 S=  F(I1)
      DO 2 I2=1,NR
    2 F(I2)=F(I2)/S
      WRITE(IO,2001)(XO(I),YO(I),R(I),F(I),I=1,NR)
 2001 FORMAT(9X,2HXO,    13X,2HYO,14X,1HR,14X,1HF,/(4E15.5))
      IC=NMTG+NMTG
      IF(ISBIAS.GT.0) IC=IC+NMTG
   88 CONTINUE
      Q=FLTRNF(B)
      DO 20 I=1,NMTG
      IF(Q-WTS(IC+I))10,10,20
   10 IG=I
      GO TO 30
   20 CONTINUE
   30 IF(ISBIAS.LE.0) GO TO 70
      IF(I-1) 40,60,65
   60 WATE=WATE*WTS(IC-NMTG+1)/WTS(IC+1)
      GO TO 70
   65 WATE=WATE*(WTS(IC-NMTG+I)-WTS(IC-NMTG+I-1))/(WTS(IC+I)-WTS(IC+I-1
     *))
   70 CONTINUE
      IF(ITSTR) 40,50,40
   40 RETURN
   50 CONTINUE
      AR=FLTRNF(AR)
      DO 200 I=1,NR
      IF(F(I)-AR)200,21,21
   21 K=I
      GO TO 22
  200 CONTINUE
      K=NR
```

```
      SUBROUTINE  SOURCE


   22 CONTINUE
      CALL XSQR(R1,R2)
      CALL AZIRN(SIN,COS)
      X=R1*COS+XO(K)
      Y=R1*SIN+YO(K)
      IF(X.GT.Y) GO TO 80
      Z=X
      X=Y
      Y=Z
   80 Z =ZO*FLTRNF(ZR)
      IF(IDR.GT.0) WRITE(6,1919)X,Y,Z,WATE
 1920  FORMAT(22H-INITIAL SOURCE POINTS  ///,
     15X,*       X          *,5X,*        Y         *,5X,*         Z         *,
     25X,*    WATE        *)
 1919 FORMAT(4(5X,E15.5))
      IDR=IDR-1
      RETURN
      END
```

```
      SUBROUTINE STRUN
      COMMON/KEANL/RK(19),IKC(3)
      DO 10 I=1,19
   10 RK(I)=0.0
      RETURN
      END
```

3. Input

```
URANIUM TRANSPORT CASK
6000
 200  250   10    0   16    0   16   16    0    0  250    3   20
   0    6    0    0  1.0      0.02      0.0      1.0          1.0   E+6
 25.0      25.0      25.0      0.0       0.0      0.0          0.0
0.104      0.558     0.901     0.986     0.999    1.0
1.05 E+07 4.0  E+06 1.4  E+064.0   E+05 1.0  E+05 2.15 E+04 4.65 E+03
1.0  E+03 2.15 E+02 4.65 E+01 1.0  E+01 2.15       1.26         1.0
0.425     0.225
16 GROUP DATA WITHOUT SPECIAL WEIGHTING
  16   13    0    0   16   13    3    3    3    1    0    0    0
  -3    0    0    2    0   22   19    1
                                  1
   3
   1    2    3
   1    1    1
        1   22   16   -6   10
```

Three cross-section blocks and the geometry input follow

Next card: Card S 1

```
CROSS SECTION REDUCTION URANIUM/WATER/IRON
  3    6    6    0    0    1    0    0    1
     1.0       1.0       1.0       1.0
     1.0       1.0       1.0       1.0
     1.0       1.0       1.0       1.0
FLUXES
FLUX DETECTORS
     1.0       1.0       1.0       1.0       1.0       1.0       1.0
     1.0       1.0       1.0       1.0       1.0       1.0       1.0
     1.0       1.0
NEUTRONS
  3    6    9   12   14   16
15 0.0      100.0       50
    25.0       25.0     6.35  0.5
    75.0       75.0     6.35  0.5
   125.0      125.0     6.35  0.5
   175.0      175.0     6.35  0.5
   225.0      225.0     6.35  0.5
    75.0       25.0     6.35  1.0
   125.0       25.0     6.35  1.0
   175.0       25.0     6.35  1.0
   225.0       25.0     6.35  1.0
   125.0       75.0     6.35  1.0
   175.0       75.0     6.35  1.0
   225.0       75.0     6.35  1.0
   175.0      125.0     6.35  1.0
   225.0      125.0     6.35  1.0
   225.0      175.0     6.35  1.0
```

4. Results

The results are given separately for the three media (reduction only). The following definitions apply

Detector 1	Uranium
Detector 2	Cover
Detector 3	Steel

The k_{eff} value is unrealistic since in the generation of the data no attention was paid to this calculation.

AVERAGE K .1113E+01

STANDARD DEV. .3315E-01 FOR 10 BATCHES

FLUENCE(ENERGY,DETECTOR) NEUTRONS

DETECTOR NO. ENERGIES	1	2	3
1.050E+07	1.395E-08	5.883E-08	1.115E-07
	.023	.071	.079
4.000E+05	1.866E-07	6.566E-07	1.901E-06
	.042	.036	.094
4.650E+03	9.278E-06	3.573E-05	1.249E-04
	.032	.079	.087
4.650E+01	4.819E-04	1.951E-03	8.472E-03
	.040	.142	.064
1.260E+00	6.493E-03	2.458E-02	1.638E-01
	.082	.148	.107
4.250E-01	1.815E-02	1.798E-01	1.737E+01
	.034	.083	.075
2.000E-02			

CONDENSED ABSORPTION RATES

DETECTOR NO. / ENERGIES	1	2	3
1.050E+07	5.330E-03	4.245E-04	5.901E-04
	.136	.110	.523
4.000E+05	9.756E-03	5.123E-04	-7.945E-06
	.158	.116	-.783
4.650E+03	8.174E-02	3.974E-03	2.588E-04
	.153	.170	.134
4.650E+01	2.142E-01	1.993E-02	2.235E-03
	.160	.082	.099
1.260E+00	2.943E-01	5.508E-02	6.558E-03
	.172	.140	.111
4.250E-01	1.191E+00	1.896E-01	2.048E-02
	.160	.106	.119
2.000E-02			

CONDENSED TOTAL CROSS SECTIONS

DETECTOR NO. / ENERGIES	1	2	3
1.050E+07	4.604E-01	2.827E-01	4.892E-01
	.144	.125	.136
4.000E+05	1.370E+00	3.711E-01	1.535E+00
	.160	.121	.139
4.650E+03	2.095E+00	1.199E+00	2.291E+00
	.153	.137	.109
4.650E+01	2.248E+00	1.530E+00	2.303E+00
	.156	.102	.112
1.260E+00	2.341E+00	1.331E+00	2.513E+00
	.171	.150	.112
4.250E-01	4.445E+00	1.699E+00	4.111E+00
	.157	.100	.119
2.000E-02			

VIII. CORRELATED CALCULATIONS

1. Description

Small differences in estimated values can be calculated satisfactorily by Monte Carlo methods only if the covariance of the neutron random walks for the perturbed and unperturbed systems is positive and of the same order of magnitude as the dispersion of the individual values. This situation can be attained for small local perturbations only if at least partial portions of the neutron histories in the two systems are not independent of one another but are correlated to each other [15]. For this reason the calculation of differences in the multiplication factor caused by small local perturbations, such as geometry, density, or material changes, is performed with correlated Monte Carlo simulations in the MORSE-K system.

The source distribution for the i-th source iteration in the case of multiplying media is determined, for the basic problem (unperturbed case), by the relation

$$Q_f^i(P) = \int k_Q(P' \rightarrow P)\, Q_f^{i-1}(P')\, dP'$$

where $Q_f^i(P)$ is the fission source density of the i-th generation at the phase-space point P. $K_Q(P' \rightarrow P)$ is the transport kernel from the source at P' to fission at P.

For the perturbed system we have, to a first approximation,

$$\tilde{Q}_f^i(P) = \int \tilde{k}_Q(P' \rightarrow P)\, Q_f^{i-1}(P')\, dP'$$

For the multiplication constants for the disturbed and undisturbed systems we have

$$k^i = \frac{\iint K_Q(P' \rightarrow P)\, Q_f^i(P')\, dP\, dP'}{\int Q_f^i(P')\, dP}$$

$$\tilde{k}^i = \frac{\iint \tilde{k}_Q(P' \rightarrow P)\, \tilde{Q}_f^i(P')\, dP\, dP'}{\int \tilde{Q}_f^i(P)\, dP}$$

The difference value is

$$\Delta k^i = \tilde{k}^i - k^i$$

If the neutron histories for the perturbed and unperturbed problems start from the same source distribution Q(P), then $\tilde{k}^i$ cannot be calculated directly but only

$$\hat{k}^i = \frac{\int\int \tilde{k}_Q (P' \to P)\, Q_f^i (P')\, dP\, dP'}{\int Q_f^i (P)\, dP}$$

The approximation $\hat{k}^i = \tilde{k}^i$ is permissible only for the case where Q(P) and Q(P) differ only a little for all values of P. If this approximation is not applicable, the following measures are provided for in MORSE-K.

a) Zone-wise correction of the source distribution. For this procedure the fission zone is subdivided into zones, ΔV_j. This subdivision should be increasingly fine as the perturbation of the source becomes larger. For each of these zones the total contribution to the fission of a neutron which started in ΔV_i is calculated by

$$F_j^i = \frac{\int_{P \in V} \int_{P' \in \Delta V_j} K_Q (P' \to P)\, Q_f^i (P')\, dP'\, dP}{\int_{P' \in \Delta V_j} Q_f^i (P')\, dP'}$$

and

$$\tilde{F}_j^i = \frac{\int_{P \in V} \int_{P' \in \Delta V_j} \tilde{K}_Q (P' \to P)\, \tilde{Q}_f^i (P')\, dP'\, dP}{\int_{P' \in \Delta V_j} \tilde{Q}_f^i (P')\, dP'}$$

If the source perturbation in ΔV_j is very small, or if ΔV_j is small enough that $Q_f^i(P)$ and $\tilde{Q}_f^i(P)$ can be regarded as constant, there is a possibility of determining the $\tilde{F}_j^i$ by use of the unperturbed source distribution $Q_f^i(P)$. The condition for this is

$$\int_{P' \in \Delta V_j} Q_f^i (P')\, dP' \neq 0$$

Using these $\tilde{F}_j^i$ values and the sources in the zones ΔV_j given by

$$\tilde{Q}_j^i = \int_{P' \in \Delta V_j} \tilde{Q}_f^i (P')\, dP'$$

(where $\sim$ designates the disturbed case), the multiplication factor can be determined by

$$\overset{(\sim)}{k}{}^{i} = \frac{\sum_j \overset{(\sim)}{F}{}^{i}_{j}\ \overset{(\sim)}{Q}{}^{i}_{j}}{\sum_j \overset{(\sim)}{Q}{}^{i}_{j}}$$

($\tilde{Q}_f^i(P)$ is determined to a first approximation.)

In the unperturbed case this relation exactly represents the relation for k^{ij}. For the perturbed case it is an approximation which presupposes the validity of the relations for $\tilde{Q}_f^i$ and $\tilde{F}_j^i$.

b) Matrix method. In this method the fission zone is also subdivided into zones. For each zone we calculate the contribution of a neutron starting in zone j to fission in zone 1.

$$\overset{(\sim)}{m}{}^{i}_{jl} = \frac{\int_{P\in\Delta V_l}\int_{P'\in\Delta V_j} \overset{(\sim)}{k}_Q(P'\to P)\ \overset{(\sim)}{Q}{}^{i}_{f}(P')\,dP'\,dP}{\int_{P'\in\Delta V_j} \overset{(\sim)}{Q}{}^{i}_{f}(P')\,dP'}$$

Here the same is true as was true for the equations for the $\tilde{F}_j^i$, and here also the requirement is that the relation

$$\int_{P'\in\Delta V_j} Q_f^i(P')\,dP' \neq 0$$

must be fulfilled.

According to Kaplan [16], the multiplication factor is the lowest eigenvalue of the matrix M, which can be readily seen from the equations

$$Q = \frac{1}{k} M^T Q$$

$$\tilde{Q} = \frac{1}{\tilde{k}} \tilde{M}^T \tilde{Q}$$

with $\overset{(\sim)}{Q}{}^{i}_{j} = \int_{P'\in\Delta V} \overset{(\sim)}{Q}{}^{i}_{f}(P')\,dP'$ $\qquad \tilde{M} = \{\tilde{m}_{jl}\}$

The advantage of this method is that the $\tilde{k}$ value and the $\tilde{Q}_j$ sources are independent of the distribution Q(P) even though the $Q_f^i(P)$ distribution was the

basis of the calculation of the $\tilde{m}_{jl}$. The approximation $\tilde{Q}_f^i$ does not have to be used with this method.

The application of the correlated method in MORSE-K does not require any change in the history simulations. However, extensions of the analysis section are needed. The changes comprise the following points:

(a) Introduction of the parameters IREGB and IIGB for identification of the source region (source medium). With these parameters the analysis of the neutron histories can be made dependent on the source event. The parameter IREGB is needed, for example, in the calculation of the zone-dependent F_j, Q_j, and m_{jl} (IREGB = j).

(b) Special analysis for (ba) neutrons that go from the source point to the first-time arrival in a perturbed region and (bb) neutrons that have previously penetrated a perturbed area. Split neutrons are treated like original neutrons. If the source point lies in the perturbed region, only part bb is carried out. The total contributions are always made up of the contributions from parts ba and bb, where part bb (as explained later) can be treated for several perturbed systems independently of the unperturbed problem. Arbitrarily, many regions described as media can be defined to be perturbed regions.

(c) Saving the parameters X, Y, Z, U, V, W, Random, IG, IREGB, IIGB, NAME, WATE, Age, which fix a specific point of a random walk of neutron NAME if the neutron undergoes its next event in a perturbed region. These parameters are written sequentially on a disk unit (NPTAP 2).

This offers the possibility of performing the simulation part, bb, for a perturbed system based on the source distribution of the unperturbed system. If no perturbation exists, part bb proceeds exactly as in the basic problem. In addition to the data above, results for the basic problem are also written on the same unit (NPTAP 2) in order to be able to determine the deviation in the perturbed system, as compared with the basic (unperturbed) system.

(d) Calculation of the individual values desired. (da) Summation of the contributions to k_{eff} for the perturbed and unperturbed part. Both the collision and flux estimators are applied. (db) Determination of the zone-dependent source for the batch. (dc) Determination of the elements F_j^i for the batch, using flux and collision estimators. (dd) Determination of the matrix elements m_{jl}^i for batch i. In contrast to dc, there is a choice here of either completely suppressing the determination of the matrix element or carrying it out only by means of the flux or collision estimators. The use of both estimators is not excluded, either. (de) Calculation of the differential changes relative to the unperturbed problem, in the case where the perturbed problem is being calculated. (df) Batch statistics on the qualities listed here. The statistical error in the eigenvalue of the matrix is calculated by means of the estimate

$$\delta k = \frac{Q^T \delta M^T Q}{Q^T Q}$$

where $\{\delta m_{ji}\}$ are the calculated statistical errors of the individual matrix elements.

The following input data must be prepared when the correlated method is used. They are read in by the subroutine STRUN; thus the data must follow the geometry description

Card K 1 (6 I 5)

NPTAP 1		Input tape for the perturbed problem (OLD-TAPE)
NPTAP 2		Output tape for the basic problem (NEW-TAPE
NPMED		Number of perturbed areas (media)
IFLM	0/1	No matrix/matrix with flux estimator
ICM	0/1	No matrix/matrix with collision estimator
NFMED		Number of zones ΔV_i (subdivision of the fission zone)

If NFMED = 0, all media with $\nu\Sigma_f \neq 0$ are subdivided as such zones

Card K 2 (14 I 5)

NGMED zone assignments if NFMED $\neq 0$. Here a zone number j must be assigned to each geometry medium such that $1 \leq j \leq$ NFMED if $\nu\Sigma_f \neq 0$. If $\nu\Sigma_f = 0$., then j must be set equal to 0. If no source perturbation is expected, NFMED may be set equal to 1.

Card K 3 (14 I 5)

NPMED. Perturbed regions if NPMED $\neq 0$. NPMED medium numbers are expected. These media are thereby declared to be perturbed media.

NOTE: For the unperturbed problem we must set NPTAP1 = 0, NPTAP 2 = (free UNIT, e.g., Tape 10). For the perturbed problem, set NPTAP1 = (free UNIT, e.g. Tape 10) and NPTAP2 = 0. NPMED, IFLM, ICM, and NFMED must be identical for the perturbed and unperturbed systems.

The following data are put out:

a) Edit of the control parameters, the zone assignments, and the perturbed regions, as well as the last addresses in the core storage. If this address should lie outside the available storage space, the program is interrupted.

b) The number of neutrons that reached a perturbed region (NP ... > 0) and the contributions to k_{eff} are edited by batches.

c) At the end of the RUN, the calculated k or Δ k values, depending on the type of run, as well as the source distribution with its associated statistical errors, are edited.

2. Test Example

The calculation of the reactor from Chapt. IV was selected as the test example. The perturbation consisted of a reduction in the radius of the outer reflector from 29.5 to 25 cm. The k_{eff} values were estimated beginning with batch 3.

The application of the matrix method and of correlated calculations required a series of program changes as well as the preparation of a new analysis package. The most important changes are shown in Table 22. Tables 23 and 24 describe the new named common areas PERT 1 and NUTRNP.

The present version of MORSE-K requires a double start of the program for correlated calculations. For the second start, the entire input must be read again, and this time the changes whose effects are to be calculated are included. (If the input data are stored by means of the RSYST program, the second input can be produced from the first by modification.)

The program package can also be used for the calculation of eigenvalues by the matrix method. (alternative to Chapt. IV). However, in this connection the altered convergence behavior and the increase in the storage requirements are to be noted.

Table 22: Subroutines for Correlated Monte Carlo

Name	Function
BANKR	Is much reduced in its function. The processing of the individual events is taken over by KEANAL.
STRUN	Reads additional input, ~~Inuit~~ winds tapes back, and calculates the starting addresses of the matrices in unnamed Common.
STBTCH	Transmits fission weights and average particle weights for each batch from the unperturbed to the perturbed run.
GETNT	Fetches particle data from Common NBANK. Writes source data from the perturbed region onto intermediate storage and fetches particle data as source information for the perturbed run.
KEANAL	Calculates contributions to the flux, collision and matrix estimators, writes particle data at entrance into the perturbed region ontox the intermediate storage NPTAP 2.
MAT	Carries out matrix integration and writes out results.
NBATCH	Performs statistical calculations at the end of the batch and transfers the data between the perturbed and unperturbed runs.
NRUN	Calculates mean values of the batch contributions.

Table 23: The Parameter of Common Pert 1

Name	Meaning
NPTAP 1	Input tape
NPTAP 2	Output tape
NPMED	Number of perturbed regions
IREGB	Starting medium of particle
NPTERM	Splitting index for neutrons which passed through the perturbed region
IZPERT	Number of particles which reached the perturbed region
LFLM	Starting address of the flux matrix (collision matrix)
LSWB	Starting address of the source vector
LNPMED	Starting address of the assignment vector
LFLMR	Starting address of the batch mean values of the matrices
LSIWR	Starting address of the batch mean values for the source
RKCOLO (2)	Collision estimator contributions, perturbed and unperturbed
RKFLXO (2)	Flux estimator contributions, perturbed and unperturbed
FAGEO (2)	Fission age, perturbed and unperturbed
SWTR	No meaning
IFLM	Matrix with flux estimator
ICM	Matrix with collision estimator
IMAT	0 - no matrix method, 1 - estimation by use of matrix
INDEX	1 - unperturbed calculation, 2 - perturbed calculation
NMO	Index for origin of the neutrons
NFMED	Number of fission zones
LFMED	Starting address of the assignment vector of the NFMED
NFM 2	NFMED * NFMED

Table 24. Parameters of Common NUTRNP

Name	Meaning
XP	x
YP	y
ZP	z
UP	U
VP	v
WP	w
RP	RANDOM
IGP	IG
WATEP	WATE
AGEP	AGE

For the collision before entrance into the perturbed region

3. Input

Because of the trivial nature of the change, the second input requires only the changing of one card of the geometry description. (in order to take account of the thinner reflector). To this are added the cards K1 to K3 for the perturbed and unperturbed runs. They read as follows:

a) Unperturbed

```
****** GEOMETRIE ******
        1.XSO          1.YSO      -858.49          5
****** K1 + K2 ******
    0   17    1    1    1
   11
```

b) Perturbed

```
***** GEOMETRIE *****
        1.XSQ          1.YSQ      625.00      $
***** K1 + K3 *****
  17    0    1    1    1
  11
```

4. Results

After 10 batches, of which seven were used for the estimation, the result in the tenth batch of the perturbed run was:

```
INNER          COLLISION          FLUX          COMBINED
     AVERAGE    .11542E+01     .11639E+01     .11548E+01
FRACT.STANDART DEV.  .56992E+01     .56554E+01     .52640E+01

               FISSION AGE   LIFE TIME
     AVERAGE    .54542E-05     .14520E-06
FRACT.STANDART DEV.  .20606E+02     .27988E+05
OUTER          COLLISION          FLUX          COMBINED
     AVERAGE    .11308E+01     .11352E+01     .11313E+01
FRACT.STANDART DEV.  .18437E+01     .19513E+01     .18408E+01

               FISSION AGE   LIFE TIME
     AVERAGE    .45548E-05     .25938E-06
FRACT.STANDART DEV.  .86923E+01     .43252E+04
FLUXMATRIX
KEFF =  .1118E+01 SOURCE TOTAL =  .1016E+04  5 ITERATIONS

SOURCE DISTRIBUTION

  1   .74115E-01    4   .45853E+00
  2   .48461E-01    5   .22528E+00
  3   .45000E-01    6   .14852E+00
NEW  KEFF =  .1118E+01  FRACT. STANDARD DEV. =  .5702E+01
OLD  KEFF =  .1206E+01  FRACT. STANDARD DEV. =  .6441E+01
DIFFERENCE =  .8802E-01  FRACT. STANDARD DEV. =  .2239E+02
COLLMATRIX
KEFF =  .1121E+01 SOURCE TOTAL =  .1016E+04  5 ITERATIONS

SOURCE DISTRIBUTION

  1   .73338E-01    4   .45657E+00
  2   .49290E-01    5   .22857E+00
  3   .44916E-01    6   .14751E+00
NEW  KEFF =  .1121E+01  FRACT. STANDARD DEV. =  .5564E+01
OLD  KEFF =  .1212E+01  FRACT. STANDARD DEV. =  .6325E+01
```

IX. AUXILIARY PROGRAMS

1. Geometry Plotting Program PICTURE

The program PICTURE consists of three subroutines (PICTURE, MESH, PRINT) and a geometry package. In addition to these, we can load a subroutine GEOCHG (D) with which small modifications can be made to geometry inputs. PICTURE"draws," by way of the printer, designated sections of arbitrary geometric configurations. The plane of the cut is overlaid with a mesh net. The subroutine LOOKZ determines material and region numbers for each mesh point.

A printer symbol can be assigned to each number. A description of the program is also given in reference 6.

1.1 Input Description for the Program PICTURE

Card P 1 Format (I5)

NUSE Option for printing print-out symbols

Material 1 - 9 ≙ 1 - 9
Material 10-35 ≙ A - Z
Material 1000 ≙ U
Material 36-44 ≙ special signs

> 0 NUSE symbols must be put in on card B

P 2 Format (40 A2)

Needed only if NUSE > 0

ATABLE (I) , I = 1, NUSE print-out symbols
Material 1000 ≙ ATABLE (NUSE), otherwise
Material > NUSE ≙ ATABLE (NUSE-1)

For NUSE > 0, the print-scale is doubled.

Card P 3 Format (I 5)

ICHGEM - Geometry Index

1	General geometry
2	Plane geometry
3	Spherical geometry
4	Cylindrical geometry
5	Combinatorial geometry

The geometry input follows according to card P 3.

P 4 Format (2 I 2, 18 A4)

ICNT Control index

< 0	Last picture with the present geometry. Create the next geometry from the present geometry with the subroutine GEOCHG (DUMMY). (Must be read in, e.g., rotation of segnemts.)
= 0	More pictures of the same geometry, or last picture all together.
> 0	Last picture with the present geometry. Next geometry by new geometry input.

IRG Material/Region Index

= 0 Material Geometry will be represented.

≠ 0 Region Geometry will be represented.

TITLE (18) Arbitrary designation of the picture.

Card P 5 Format (6 E 10,5)

XO, YO, ZO	Coordinates of the left upper picture corner
X1, Y1, Z1	Coordinates of the lower right picture corner

Card P 6 Format (6 E 10,5)

XU, YU, ZU	Values proportional to the direction cosines of the left picture edge (starting from XO, YO, ZO)
XV, YV, ZV	Values proportional to the direction cosines of the upper picture edge

Card P 7 Format (2 I 5, 2E 10,5)

NU	Number of print lines
NV	Number of print spaces
DELU	Distance corresponding to two print lines
DELV	Distance corresponding to two print spaces

Not all the information from cards P5 through P7 is needed for the production of a picture. The following combinations make the most sense:

1. Xl, Yl, Zl	are designated. One of the values on card P7 must be given. If a pair is designated(NV or NU or DELV + DELU), the picture will in general be distorted. If the number of print points is given, the latter is disregarded.
2. Xl, Yl, Zl	are not designated. In this case, NU and NV and either DELU and DELV must be given. If both DELU and DELV are given, the picture will in general be distorted.

The further sequence of input cards is determined according to:

INCT <0, the input required for changing the geometry called up by GEOCHG or following routine. Then jump to CARD P4.

$= 0$, jump to CARD P4 if no more pictures are desired.

>0, jump to CARD P3.

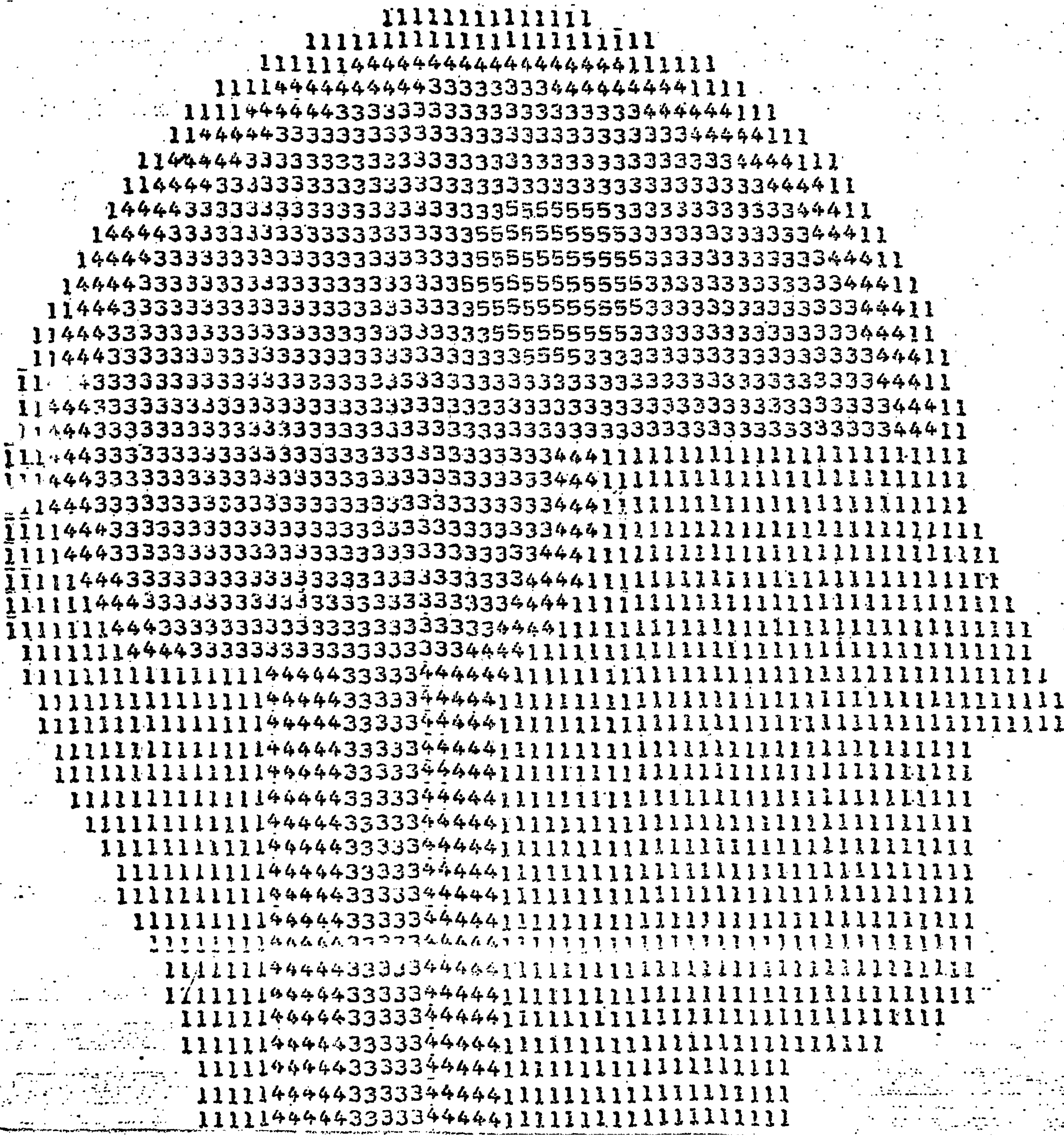
1.2

Test Example for the Program PICTURE

As test example, the head geometry from Sect. IV was printed. The PICTURE input for the picture shown looks like this:

```
      0
  ---------->        GEOMETRY INPUT        <----------
     0  0     HEAD
               0.0      -11.0      26.5      0.0      13.2      0.0
               0.0        0.0      -1.0      0.0       1.0      0.0
               70
```

RESULTS



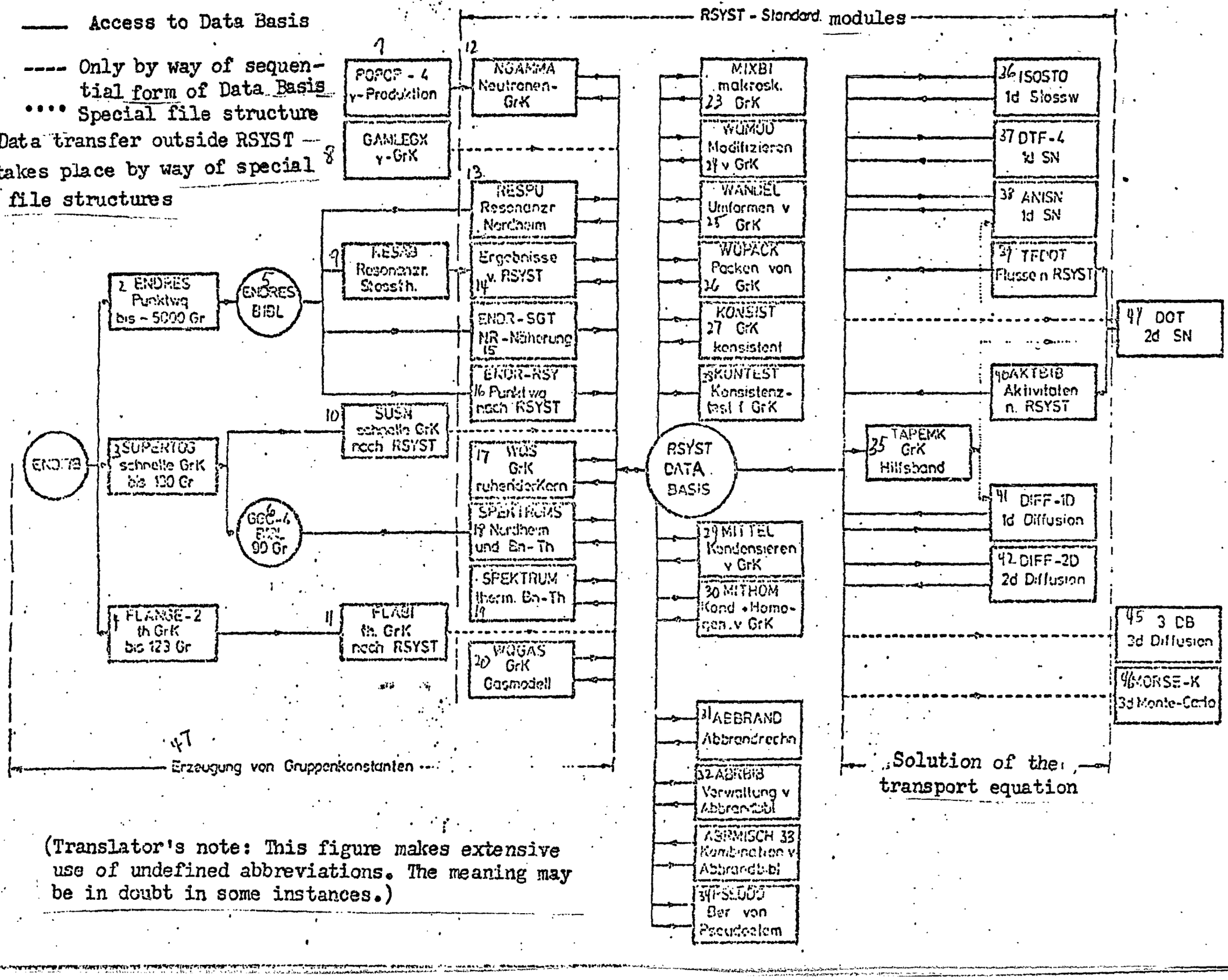
2. Programs for the Production and Processing of Data

Program packages such as MORSE-K should never be used by themselves. Their advantages show up only when they are used in a demanding program environment. Such environments are today delivered almost exclusively by program systems. The reactor-program system RSYST [3] is such a system. The environment delivered by this program system is indicated in Fig. 4 (the state of development of RSYST as of beginning of 1971). The further integration of MORSE-K into RSYST will play a large part in determining Version II of this package.

The following numbers refer to entries on Fig. 4:

2. ENDRES Pointwise cross sections up to 5000 groups
3. SUPERTOG Fast group constants up to 100 groups
4. FLANGE-2 thermal group constants up to 123 groups
5. ENDRES Library
6. GGC-4 Library 99 groups
7. POPOP - 4 Gamma production
8. GAMLEGX Gamma group constants
9. RESAB (?) Resonance calculation, collision ~~thermal~~ theory (?)
10. SUSN Fast group constants according to RSYST
11. FLABI Thermal group constants from RSYST
12. NGAMMA Neutron group constants
13. RESPU Nordheim Resonance calculation
14. Results from RSYST
15. ENDR - SGT NR-approximation
16. ENDR-RSY Pointwise cross sections according to RSYST (?)
17. WQS Group constant nucleus at rest
18. SPECTRUMS Nordheim and Bn-theory
19. SPECTRUM Thermal Bn-theory
20. WQGAS Group Constants, Gas model

Fig. 4 RSYST- Modules and Programs for Reactor Physics Calculations



(Translator's note: This figure makes extensive use of undefined abbreviations. The meaning may be in doubt in some instances.)

23. MIXBI Macroscopic group constants
24. WQMOD Modification of group constants
25. WANDEL Transforming of group constants
26. WQPACK Packing of group constants
27. KONSIST Groupp constants, consistent
28. KONTEST Consistency test for group cross constants
29. MITTEL Condensation of group constants
30. MITHOM Condensation and homogenization of group constants
31. ABBRAND Burnup calculation
32. ABRBIB Management of burnup library
33. ABRMISCH Combination of burnup libraries
34. PSEUDO Calculation of pseudo-elements
35. TAPEMK Group constant auxiliary tape
36. ISOSTO 1-dimension collision method
37. DTF-4 1-dimension S_n
38. ANISN 1-dimension S_n
39. TFDOT Fluxes from RSYST
40. AKTBIB Activities from RSYST
41. DIFF-1D 1-dimension diffusion
42. DIFF-2D 2-dimension diffusion
44. DOT 2-dimension S_n
45. 3 DB 3-dimension diffusion
46. MORSE-K 3-dimension Monte Carlo
47. Production of group constants

Literatur@

1. F.A.R.Schmidt
Status of the Monte Carlo Development
in IAEA-SM-154 paper 70 (1972)

2. E.A.Straker et al.
The MORSE Code - a Multigroup Neutron and Gamma Ray
Monte Carlo Transport Code
ORNL - 45 85 (1970)
sowie
RSIC - CCC -127
und
ORNL - CF-70 - 2 - 31 (1970)

3. R.Rühle et al.
Kontrolle des Reaktorprogrammsystems RSYST-I
IKE-Bericht 4-4 (1972)
und
Dissertation Universität Stuttgart (1973)

4. F.A.R.Schmidt
Biased Angle Selection in Monte Carlo Shielding
Calculations
Nucl. Eng. and Des. 15 (1971) p.2o9

5. R.R.Coveyou and R.D.Macpherson
Fourier Analysis of Uniform Random Number Generators
J.Ass.Comp.Mach 14-1 (1967) p.100

6. F.A.R.Schmidt
Stochastische Methoden der Numerik mit Beispielen aus
der Transporttheorie
unveröffentlicht

7. E.A.Straker et al.
The MORSE Code with Combinatorial Geometry
DNA 2860T (SAI - 72 - 511 - LJ) (1972)
und
RSIC: CCC-203

8. Y.Caseau
Geometric Treatment of Trajectories in the O5R Program
RSIC Report Collection ORNL-tr-1379 (1966)

9. F.H.Clark
Methods and Data for Reactor Shield Calculations
in Adv.in Nuc.Sci.Techn. (Ed.by E.J.Henley and J.Lewins)
Vol. 5 (1969)

10. R.Pruschek et al.
Das Incore Thermionik Reaktor Projekt (ITR)
atw 8/9 (1971) p.472

11. F.R.Mynatt
A User's Manual for DOT
USAEC K-1684 (1969)

12. H.H.Schweer
Techn. Notiz PTB Braunschweig (1973)

13. H.Rief, K.Kschwendt
Reactor Analysis by Monte Carlo
Nucl. Sci.Eng. 30 (1967) p.395

14. W.Thomas, W.Weber
Kritikalitätssicherheit typischer Transport- und
Lagerbehälter für homogenes Spaltmaterial
MRR 105 (1972)

15. W.Bernnat:
Abschätzung der Reaktivität beweglicher Reflektorelemente kompakter Reaktoren mit der Monte-Carlo-Methode
Dissertation Universität Stuttgart (1972)

16. Kaplan, E.L.:
Monte Carlo Methods for Equilibrium Problems in Neutron Multiplication.
UCLR, 5275-T (1958)

APPENDIX

This appendix was taken from reference 7. The permitted types of geometric solids and the common areas are described.

1. Body Types

Combinatorial geometry (CG) describes general three dimensional material configurations by considering unions, differences intersections of simple bodies such as spheres, boxes, cylinders, etc. In effect, the geometric description subdivides the problem space into unique zones.* Each zone is the result of combining one or more of the following geometric bodies.

1. Rectangular Parallelepiped (RPP)
2. Box (An RPP randomly oriented in space)
3. Sphere
4. Right Circular Cylinder
5. Right Elliptic Cylinder
6. Truncated Right Angle Cone
7. Ellipsoid
8. Right Angle Wedge
9. Arbitrary Convex Polyhedron of 4, 5, or 6 sides.

Body types 2-9 may be arbitrarily oriented with respect to the x, y, z coordinate axes used to determine the space. Body 1, a special body described below, must have sides which are parallel to the coordinate axes.

The basic technique for the description of the geometry consists of defining the location and shape of the various zones in terms of

---

* To avoid confusion between importance regions and combinatorial geometry regions, we depart from previous combinatorial geometry descriptions and use the term zone to indicate a combinatorial geometry region which is designated by the variable IR. The term region is reserved for an importance region. Thus the zone index is IR.

the intersections and unions of the geometric bodies. A special operator notation involving the symbols (+), (-), and (ØR) is used to describe the intersections and unions. These symbols are used by the program to construct information relating material descriptions to the body definitions.

If a body appears in a zone description with a (+) operator, it means that the zone being described is wholly contained in the body. If a body appears in a zone description with a (-) operator, it means that the zone being described is wholly outside the body. If the body appears with an (ØR) operator, it means that the zone being described includes all points in the body. In some instances, a zone may be described in terms of subzones lumped together by (ØR) statements. When (ØR) operators are used there are always two or more of them, and they refer to all body numbers following them, either (+) or (-).

Techniques for describing a particular geometry are best illustrated by examples. Consider an object composed of a sphere and a cylinder as shown in Fig. 1. To describe the object, one takes a spherical body (2) penetrated by a cylindrical body (3) (see Fig. 1). If the materials in the sphere and cylinder are the same, then they can be considered as one zone, say zone I (Fig. 1c). The description of zone I would be

$$I = \text{ØR} + 2\ \text{ØR} + 3 \quad .$$

This means that a point is in zone I if it is <u>either</u> inside body 2 <u>or</u> inside body 3.

If different materials are used in the sphere and cylinder, then the sphere with a cylindrical hole in it would be given a different zone number (say J) from that of the cylinder (K).

The description of zone J would be (Fig. 1d):

$$J = +2 - 3 \quad .$$

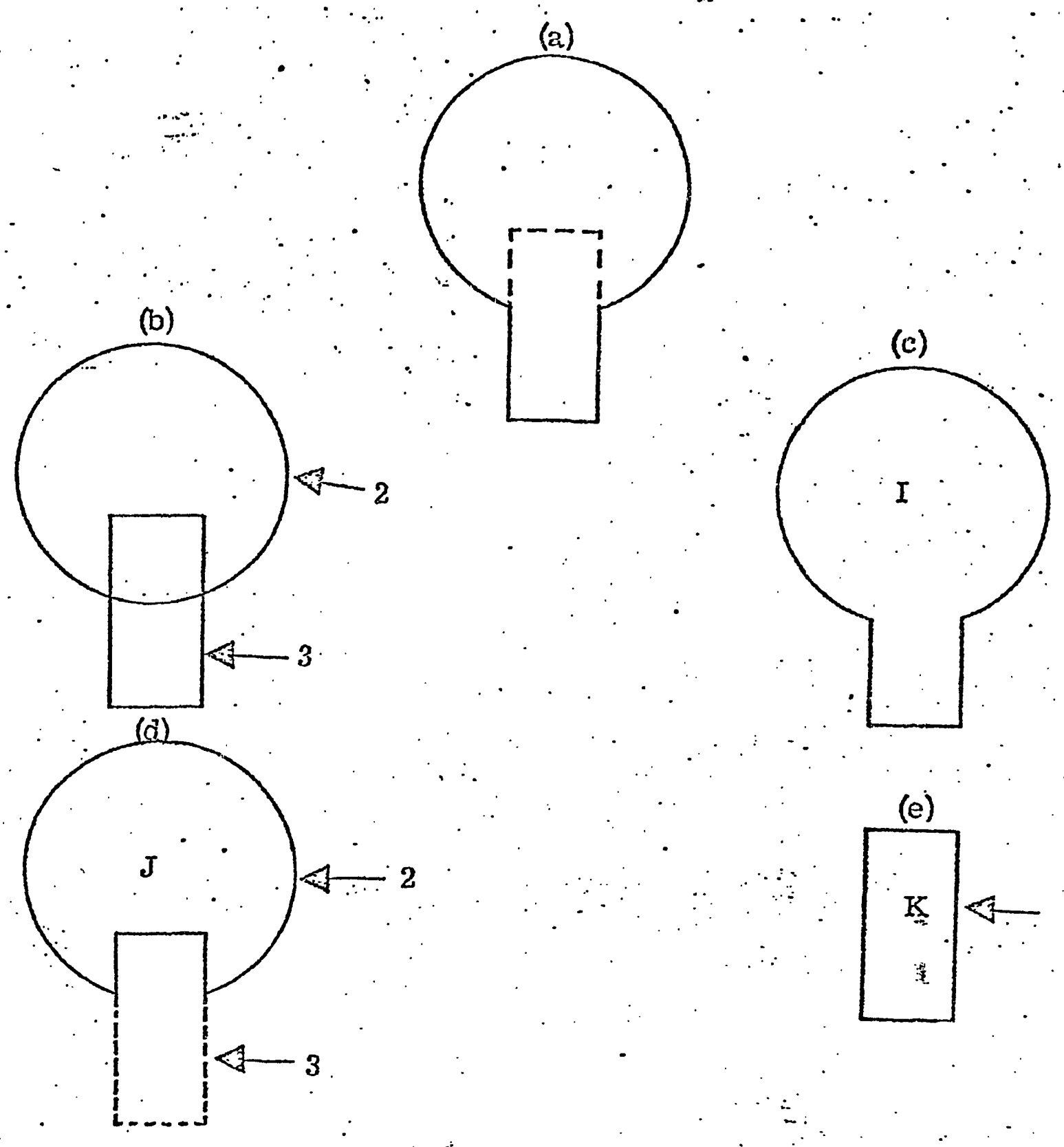


Fig. 1. Examples of combinatorial geometry method.

which are not inside body 3.

The description of zone K is simply (Fig. 1e):

$$K = +3$$

That is, all points in zone K lie inside body 3.

Combinations of more than two bodies and similar zone descriptions could contain a long string of (+), (-), and (ØR) operators. It is important however to remember that every spatial point in the geometry must be located in one and only one zone.

As a more complicated example of the use of the (ØR) operator, consider the system shown in Fig. 2 consisting of the shaded zone A and the unshaded zone B. These zones can be described by the two BØX's, bodies 1 and 3, and the RCC, body 2. The zone description would be

$$A = +1 +2$$

and

$$B = ØR + 3 - 1\ ØR + 3 - 2$$

Notice that the ØR operator refers to all following body numbers until the next ØR operator is reached.

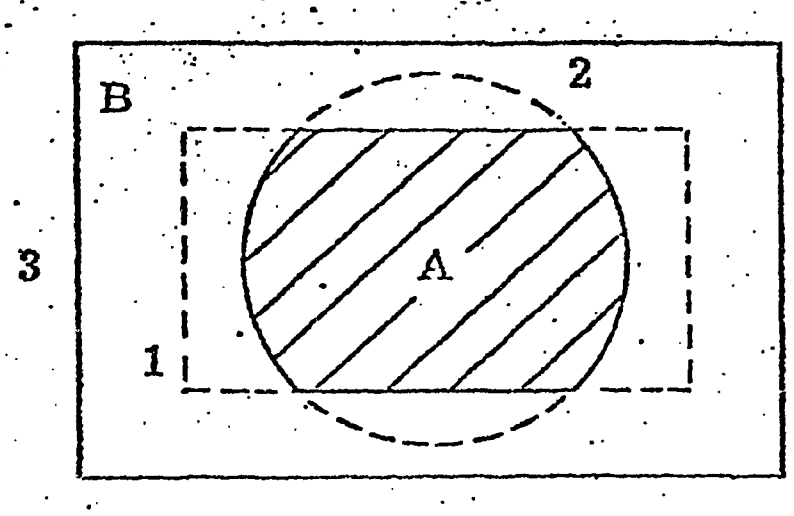


Fig. 2. Use of ØR operators.

The geometry must be specified by establishing two tables. The first table describes the type and location of the set of bodies used in the geometrical description. The second table identifies the physical zones in terms of these bodies. The input routine processes these

tables to put the data in the form required for ray tracing. Because the ray tracing routines cannot track across the outermost body, all of the zones must be within a surrounding external void so that all escaping particles are absorbed. Also no point may be in more than one zone.

The information required to specify each type of body is as follows:

a. Rectangular Parallelepiped (RPP)

Specify the minimum and maximum values of the x, y, and z coordinates which bound the parallelepiped.

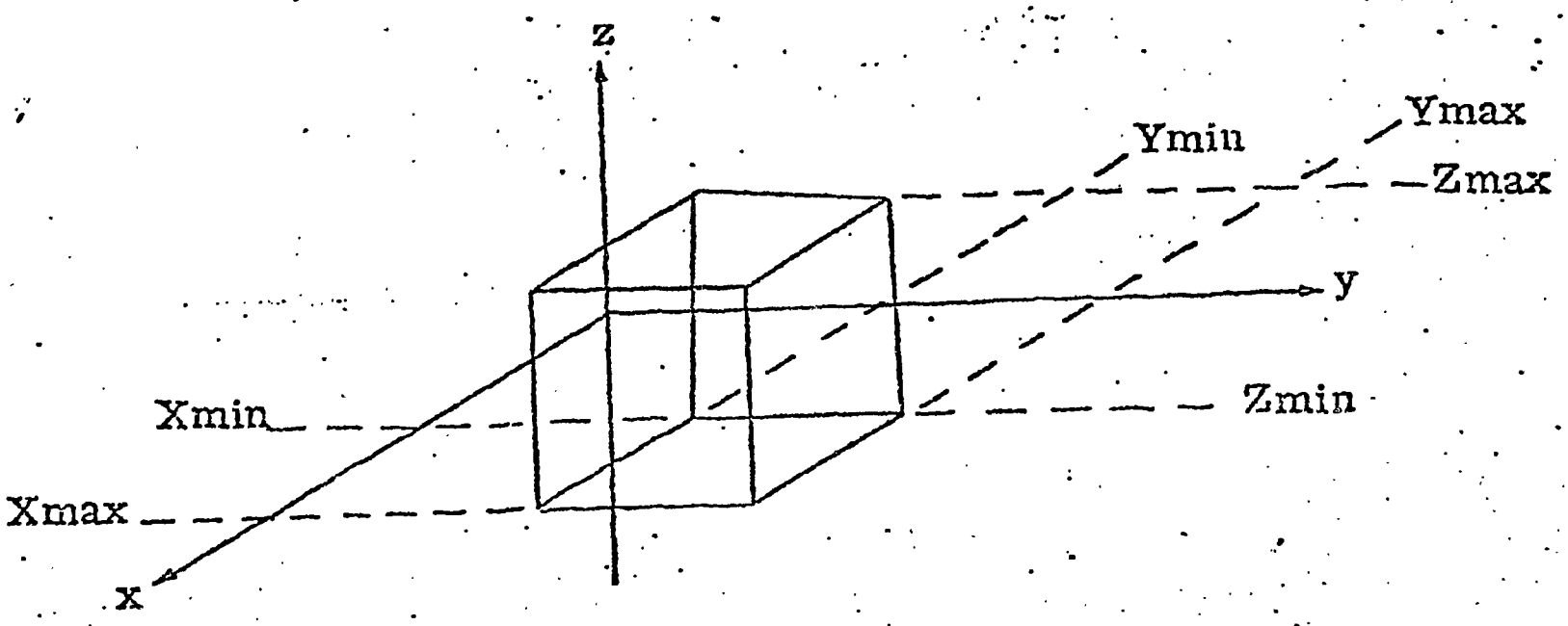


Fig. 3. Rectangular Parallelepiped (RPP).

b) Sphere (SPH)

Specify the vertex V at the center and the scalar, R, denoting the radius.

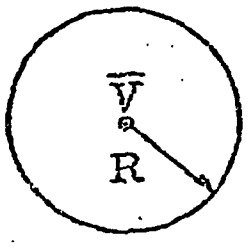


Fig. 4. Sphere (SPH).

c) Right Circular Cylinder (RCC)

Specify the vertex V at the center of one base, a height vector, H, expressed in terms of its x, y, and z components, and a scalar, R, denoting the radius.

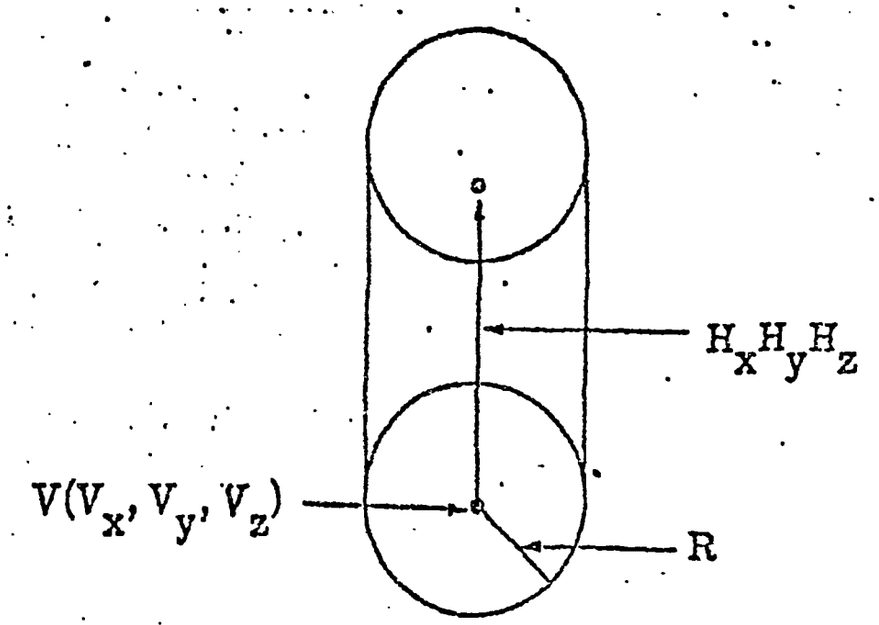


Fig. 5. Right Circular Cylinder (RCC).

d) Right Elliptical Cylinder (REC)

Specify coordinates of the center of the base ellipse, a height vector, and two vectors in the plane of the base defining the major and minor axes. Presently this body is not implemented.

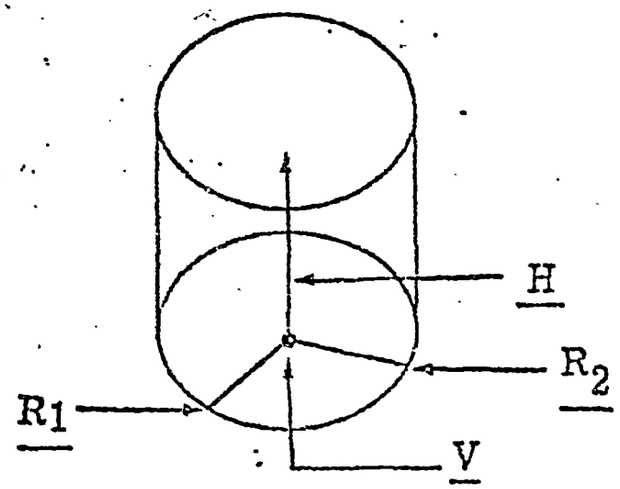


Fig. 6. Right Elliptical Cylinder (REC).

e) Truncated Right Angle Cone (TRC)

Specify a vertex V at the center of the lower base, the height vector, H, expressed in terms of its x, y, z components, and two scalars, R_1 and R_2, denoting the radii of the lower and upper bases.

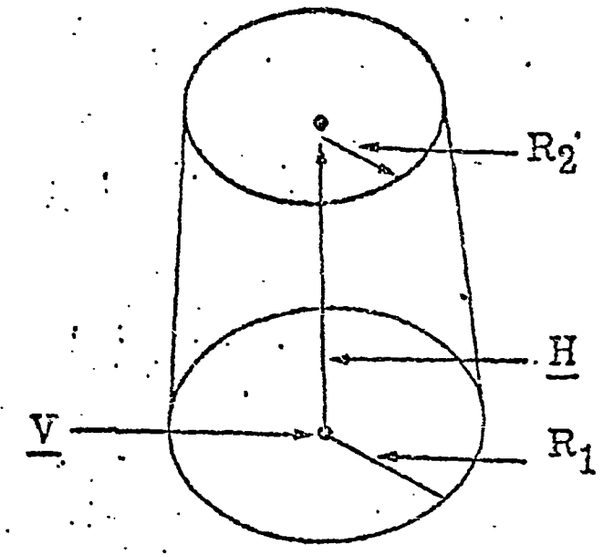


Fig. 7. Truncated Right Angle Cone (TRC).

f) Ellipsoid (ELL)

Specify two vertices, V_1 and V_2, denoting the coordinates of the foci and a scalar, R, denoting the length of the major axis.

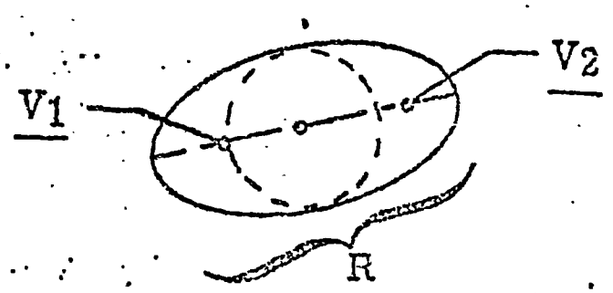


Fig. 8. Ellipsoid (ELL).

g) Wedge (WED)

Specify the vertex V at one of the corners by giving its (x, y, z) coordinates. Specify a set of three mutually perpendicular vectors, a_i, with a_1 and a_2 describing the two legs of the right triangle of the wedge. That is, the x, y, and z components of the height, width, and length vectors are given.

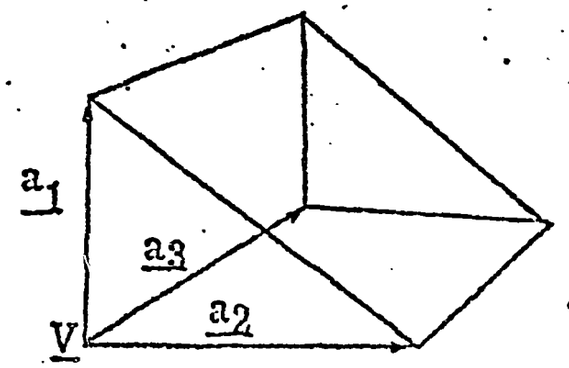


Fig. 9. Right Angle Wedge (WED).

h) Box (BØX)

Specify the vertex V at one of the corners by giving its (x, y, z) coordinates. Specify a set of three mutually perpendicular vectors, a_i representing the height, width, and length of the box, respectively. That is, the x, y, and z components of the height, width, and length vectors are given.

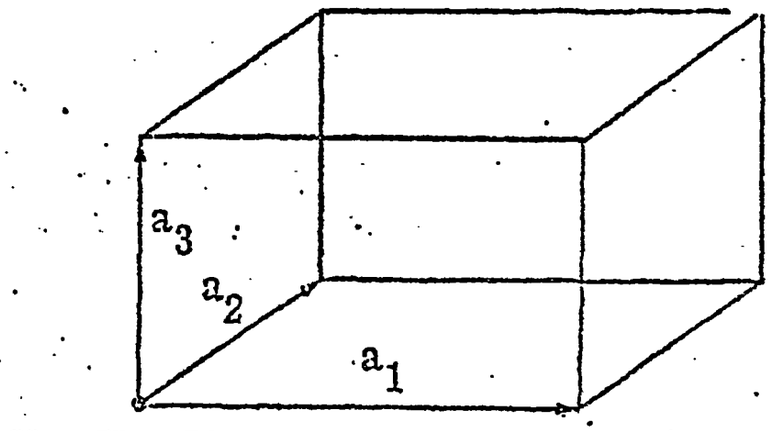


Fig. 10. Box (BØX).

i) Arbitrary Polyhedron (ARB)

Assign an index (1 to 8) to each vertex. For each vertex, give the x, y, z coordinates. Each of the six faces are then described by a four-digit number giving the indices of the four vertex points in that face. For each face these indices must be entered in either clockwise or counterclockwise order.

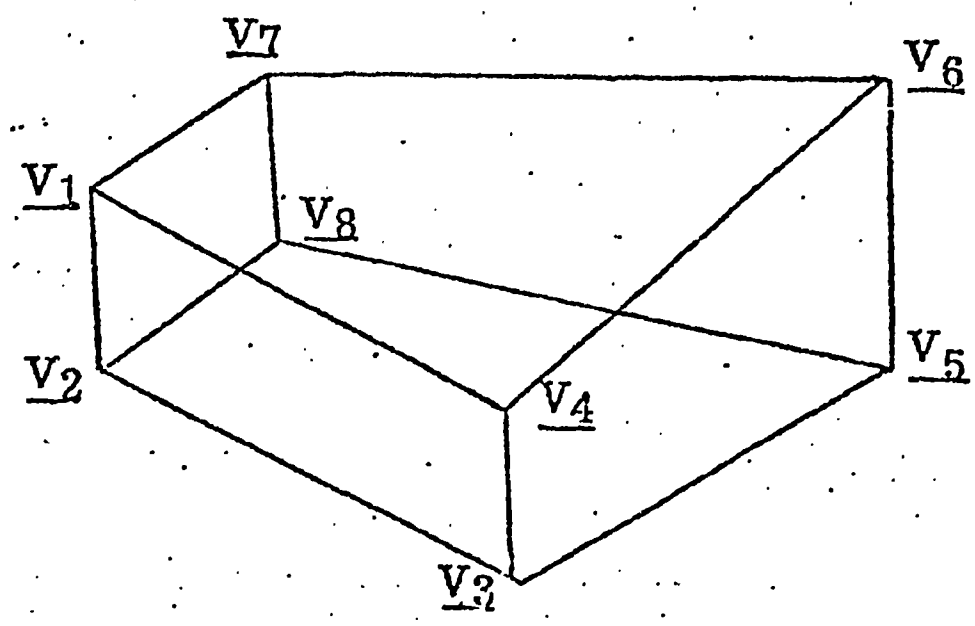


Fig. 11. Arbitrary Polyhedron (ARB).

2. Common Layouts

Tab. 1 Layout of combinatorial geometry data in blank common.

Starting Location	Information	Size
NGEØM=NADD	Length of geometry array	1
KMA	MA Integer array	LTMA
KFPD	FPD Floating point array	LFPD
KLCR	LØCREG Indices to correlate MA array data with code zone data	NUMR
KNBD	NUMBØD Number of bodies for each code zone	NUMR
KIØR	IRØR Indices to correlate input zone to code zone	NUMR
KRIZ	MRIZ Indices to correlate MØRSE region to input zone	IRTRV
KRCZ	MRCZ Indices to correlate MØRSE region to code zone	NUMR
KMIZ	NMIZ Indices to correlate MØRSE media to input zone	IRTRV
KMCZ	NMCZ Indices to correlate MØRSE media to code zone	NUMR
KKR1	KR1 Indices to correlate first code zone to input zones	IRTRV
KKR2	KR2 Indices to correlate last code zone to input zone	IRTRV
KNSR	NSØR Indices of code zones in which source particles have been found	NUMR
KVØL	VNØR Volume of each MØRSE region	NIR
NGLAST		

Tab. 2 Detailed layout of the MA array in blank common.

Position in Blank Common	Information Stored	Size	Description
KMA	Body number 1	7	Body number 1 data (each body requires 7 words of information)
KMA + 1	LØØP for body number 1		
KMA + 2	Body type (ITYPE)		
KMA + 3	LRI		
KMA + 4	LRØ		
KMA + 5			
KMA + 6	Beginning location in FPD of body data		
KMA + 7	Body number 2	7	Same information and order as for body number 1.
KMA + 14			
KMA (L-1)*7	Body number (L)	7	Body number L is the last body.
KMA + L*7	Zone number 1	1	Beginning of code zone information.

Tab. 2 (Cont'd)

Position in Blank Common	Information Stored	Size	Description
	Number of first body in this zone	4	Beginning of information about bodies defining code zone 1. Integer data location is given by: 7*(Body number) - 6.
	Location of integer data for this body		
	First zone to search upon exiting this body		
	Location of next zone to be searched		
	Data on second body in this zone	4	The last two words in each set of body data initiate the "leap frog" process by which the code stores possible zones which can be entered upon exiting this body in that particular zone. These zones are checked by the code when the next zone entered is being determined. If the next zone is not located from this stored data, all zones are searched.
	Data on last body in the zone	4	
	Zone number 2	1	
	Body information		Same information as above except for code zone number 2.
	Zone data of last zone		Code zone information about the last zone input on cards.
KMA + LDATA	Code search information KMA + LTMA-1	2*NAZT	Storage set aside for determining the zone to be searched and where the next zone number is located.

Tab. 3 Detailed layout of the FPD array in blank common.

Position in Blank Common	Information Stored	Size	Description
KFPD	RIN for body 1	1	Path length data for last trajectory in body 1.
KFPD + 1	RØUT for body 1	1	
KFPD + 2	First 6 words of real data for body 1	6	Read from first card of body 1 card set.
KFPD + 8	Remaining words of real data for body 1	N_1	N_1 depends on body type. See Table B IV.
KFPD + 8 + N_1	RIN for body 2	1	Same information as above, but for body 2.
	RØUT for body 2	1	
	Remaining data for body 2 (N_2 words)	N_2	
	Repeat for NUMB bodies.		

NOTE: 8 words are set aside at the end of the FPD array, but are not used.

Tab. 4 Definitions of Variables in Common GØMLØC

Variable	Definition
KMA	Starting location for the array MA containing integer data for each code zone.
KFPD	Starting location for the array FPD containing real data for each code zone.
KLCR	Starting location for the array LØCREG(I) which contains the starting location in the MA array for the I^{th} code zone data.
KNBD	Starting location for the array NUMBØD(I) which contains the number of bodies for the I^{th} code zone.
KIØR	Starting location for the array IRØR(I) which contains the index of the corresponding input zone for the I^{th} code zone.
KRIZ	Starting location for the array MIRZ(I) which contains the index of the MØRSE region corresponding to the I^{th} geometry input zone.
KRCZ	Starting location for the array MRCZ(I) which contains the index of the MØRSE region corresponding to the I^{th} geometry code zone.
KMIZ	Starting location for the array MMIZ(I) which contains the index of the MØRSE media corresponding to the I^{th} geometry input zone.
KMCZ	Starting location for the array MMCZ(I) which contains the index of the MØRSE media corresponding to the I^{th} code zone.
KKR1	Starting location for the array KR1(L) contains the first code zone which was made from the L^{th} input zone.
KKR2	Starting location for the array KR2(L) contains the last code zone which was made from the L^{th} input zone.
KNSR	Starting location for array NSØR which contains the code zones in which source particles have been found.

Tab. 4 (Cont'd)

Variable	Definition
KVØL	Starting location for the array VNØR(I) which contains the volume for MØRSE region (I).
NADD	Starting location for the geometry data length and changed in JØMIN to the total number of words required for geometry data.
LDATA	Length of the integer data in the MA array excluding the words set aside for zone search information.
LTMA	Total length of the MA array.
LFPD	Length of the FPD array
NUMR	Number of code produced zones.
IRTRU	Number of input zones.
NUMB	Number of bodies.
NIR	Number of MØRSE geometry regions.

Tab. 5 Definitions of Variables in Common ØRGI*

Variable	Definition
MARK	Set 1 in G1 if trajectory end point is reached before next intersection. Otherwise set to 0.
DIST0	Distance from point XB(3) to next scattering point. Used in G1 to avoid calculating the next zone if a scattering event occurs before the intersection.
NMEDG	Zone number IR from a LØØKZ call. Stored in BLZNT by MSØUR.

* Note: Variable names are not the same in all routines.

Tab. 6 Definitions of Variables in Common PAREM as found in Combinatorial Geometry

Variable	Definition
XB(3)	Coordinates of the starting point of the present path.
WB(3)	Direction cosines of particle trajectory. Equal to U, V, and W.
IR	Combinatorial zone of present particle position.
WP(3)	Temporary storage of WB(3).
XP(3)	Temporary storage of XB(3).
IDBG	Set non-zero to initialize a debug printout.
IRPRIM	Next region to be entered after a call of G1.
NASC	Body number of last calculated intersection. Set negative to indicate source or collision point not on a body surface.
LSURF	Surface of body NASC where intersection occurred. Positive if particle is entering the body and negative when exiting.
NBØ	Body number and a sign used to define zones. Input in zone description as positive when zone is contained in body and as negative if zone is outside body.
LRI	Entry surface calculated in GG.
LRØ	Exit surface calculated in GG.
RIN	Distance to entry calculated in GG.
RØUT	Distance to exit calculated in GG.
KLØØP	Trajectory index of present path incremented in G1.
LØØP	Index of last trajectory calculated for body NBØ. If LØØP is equal to KLØØP, GG returns immediately with old values of RIN, RØUT, LRI, and LRØ.
ITYPE	Body type of body NBØ (indicates BØX, SPH, etc.).
PINF	Machine infinity.
NØA	Not used.
DIST	Distance from XB(3) to present point.